# Evolution von ontologiebasierten Mappings in den Lebenswissenschaften

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

DISSERTATION

zur Erlangung des akademischen Grades

Doctor Rerum Naturalium
(Dr. rer. nat.)
im Fachgebiet Informatik

vorgelegt von
Diplom-Bioinformatikerin Anika Groß
geboren am 18. November 1982 in Erfurt

Die Annahme der Dissertation haben empfohlen:

1. Prof. Dr. Erhard Rahm (Universität Leipzig)
2. Prof. Dr. Ulf Leser (Humboldt-Universität zu Berlin)

Die Verleihung des akademischen Grades erfolgt mit Bestehen der Verteidigung am 05. März 2014 mit dem Gesamtprädikat *magna cum laude*.

# Danksagung

Ich bedanke mich ganz besonders bei meinem Betreuer Prof. Dr. Erhard Rahm, der mir die Möglichkeit gab, zu promovieren. In zahlreichen Gesprächen und Diskussionen unterstützte er mich mit vielen hilfreichen Hinweisen und Vorschlägen, meine Ideen und Ergebnisse zu reflektieren und weiterzuentwickeln. Insbesondere halfen mir Prof. Rahms Anmerkungen stets, meine Entwürfe für wissenschaftliche Publikationen zu verbessern und schließlich zu veröffentlichen.

Ich bin sehr dankbar für die Finanzierung meiner Mitarbeiterstelle während meiner Promotion durch die Abteilung Datenbanken des Instituts für Informatik der Universität Leipzig, das Interdisziplinäre Zentrum für Bioinformatik (IZBI) in Leipzig und das „eScience“-Forschungsnetzwerk Sachsen der Hochschule für Technik, Wirtschaft und Kultur (HTWK) in Leipzig.

Ich danke all meinen Kollegen in der Abteilung Datenbanken für das gute Arbeitsklima und die schönen gemeinsamen Erlebnisse. Allen voran möchte ich mich ganz besonders bei Michael Hartung bedanken, mit dem ich während meiner gesamten Promotion zusammen arbeiten und forschen durfte. Ich danke Michael sehr für all seine guten Ratschläge, unsere vielen Gespräche und einfach die tolle Zusammenarbeit. Außerdem möchte ich Toralf Kirsten und Andreas Thor für viele hilfreiche Diskussionen während der letzten Jahre danken. Kay Prüfer und Janet Kelso vom Max-Planck-Institut für evolutionäre Anthropologie in Leipzig danke ich für ihre hilfreichen Kommentare und ihre Unterstützung während unserer gemeinsamen Arbeit zu funktionalen Analysen. Außerdem danke ich Julio Cesar Dos Reis und Cédric Pruski vom Public Research Centre Henri Tudor in Luxemburg für die erfolgreiche, kollaborative Arbeit zur Adaptierung von Mappings. Andrea Hesse sowie Gabriele Queck und Fabian Schmidt danke ich für ihre freundliche Unterstützung bei zahlreichen organisatorischen und administrativen Aufgaben.

Ich bedanke mich sehr bei allen fleißigen Lesern dieser Arbeit für ihre hilfreichen Vorschläge zur Verbesserung und Korrektur. Von Herzen danke ich meinen Eltern, meinem Bruder und meinen Großeltern für ihre uneingeschränkte Unterstützung. Meinem Freund Marko danke ich so sehr, weil er einfach da ist und immer Verständnis hat.

Leipzig, den 9. Dezember 2013 Anika Groß

# Zusammenfassung

Im Bereich der *Lebenswissenschaften* steht eine große und wachsende Menge heterogener Datenquellen zur Verfügung, welche häufig in quellübergreifenden Analysen und Auswertungen miteinander kombiniert werden. Um eine einheitliche und strukturierte Erfassung von Wissen sowie einen formalen Austausch zwischen verschiedenen Applikationen zu erleichtern, kommen *Ontologien* und andere strukturierte Vokabulare zum Einsatz. Sie finden Anwendung in verschiedenen Domänen wie der Molekularbiologie oder Chemie und dienen zumeist der *Annotation* realer Objekte wie z. B. Gene oder Literaturquellen. Unterschiedliche Ontologien enthalten jedoch teilweise überlappendes Wissen, so dass die Bestimmung einer Abbildung (*Ontologiemapping*) zwischen ihnen notwendig ist. Oft ist eine manuelle Mappingerstellung zwischen großen Ontologien kaum möglich, weshalb typischerweise automatische Verfahren zu deren Abgleich (*Matching*) eingesetzt werden. Aufgrund neuer Forschungserkenntnisse und Nutzeranforderungen entwickeln sich die Ontologien kontinuierlich weiter. Die Evolution der Ontologien hat wiederum Auswirkungen auf abhängige Daten wie beispielsweise Annotations- und Ontologiemappings, welche entsprechend aktualisiert werden müssen. Im Rahmen dieser Arbeit werden neue Methoden und Algorithmen zum Umgang mit der *Evolution ontologiebasierter Mappings* entwickelt. Dabei wird die generische Infrastruktur *GOMMA* zur Verwaltung und Analyse der Evolution von Ontologien und Mappings genutzt und erweitert.

Zunächst wurde eine vergleichende *Analyse der Evolution* von Ontologiemappings für drei Subdomänen der Lebenswissenschaften durchgeführt. Ontologien sowie Mappings unterliegen teilweise starken Änderungen, wobei die Evolutionsintensität von der untersuchten Domäne abhängt. Insgesamt zeigt sich ein deutlicher Einfluss von Ontologieänderungen auf Ontologiemappings. Dementsprechend können bestehende Mappings infolge der Weiterentwicklung von Ontologien ungültig werden, so dass sie auf aktuelle Ontologieversionen migriert werden müssen. Dabei sollte eine aufwendige Neubestimmung der Mappings vermieden werden. In dieser Arbeit werden zwei generische Algorithmen zur (semi-) automatischen *Adaptierung* von Ontologiemappings eingeführt. Ein Ansatz basiert auf der Komposition von Ontologiemappings, wohingegen der andere Ansatz eine individuelle Behandlung von Ontologieänderungen zur Adaptierung der Mappings erlaubt. Beide Verfahren ermöglichen die Wiederverwendung unbeeinflusster, bereits bestätigter Mappingteile

und adaptieren nur die von Änderungen betroffenen Bereiche der Mappings. Eine Evaluierung für sehr große, biomedizinische Ontologien und Mappings zeigt, dass beide Verfahren qualitativ hochwertige Ergebnisse produzieren.

Ähnlich zu Ontologiemappings werden auch ontologiebasierte *Annotationsmappings* durch Ontologieänderungen beeinflusst. Die Arbeit stellt einen generischen Ansatz zur Bewertung der Qualität von Annotationsmappings auf Basis ihrer Evolution vor. Verschiedene Qualitätsmaße erlauben die Identifikation glaubwürdiger Annotationen beispielsweise anhand ihrer *Stabilität* oder Herkunftsinformationen. Eine umfassende Analyse großer Annotationsdatenquellen zeigt zahlreiche Instabilitäten z. B. aufgrund temporärer Annotationslöschungen. Dementsprechend stellt sich die Frage, inwieweit die Datenevolution zu einer Veränderung von abhängigen Analyseergebnissen führen kann. Dazu werden die Auswirkungen der Ontologie- und Annotationsevolution auf sogenannte funktionale Analysen großer biologischer Datensätze untersucht. Eine Evaluierung anhand verschiedener Stabilitätsmaße erlaubt die Bewertung der Änderungsintensität der Ergebnisse und gibt Aufschluss, inwieweit Nutzer mit einer signifikanten Veränderung ihrer Ergebnisse rechnen müssen.

Darüber hinaus wird GOMMA um *effiziente Verfahren* für das *Matching sehr großer Ontologien* erweitert. Diese werden u. a. für den Abgleich neuer Konzepte während der Adaptierung von Ontologiemappings benötigt. Viele der existierenden Match-Systeme skalieren nicht für das Matching besonders großer Ontologien wie sie im Bereich der Lebenswissenschaften auftreten. Ein effizienter, *kompositionsbasierter Ansatz* gleicht Ontologien indirekt ab, indem existierende Mappings zu Mediatorontologien wiederverwendet und miteinander kombiniert werden. Mediatorontologien enthalten wertvolles Hintergrundwissen, so dass sich die Mappingqualität im Vergleich zu einem direkten Matching verbessern kann. Zudem werden generelle Strategien für das *parallele Ontologie-Matching* unter Verwendung mehrerer Rechenknoten vorgestellt. Eine größenbasierte Partitionierung der Eingabeontologien verspricht eine gute Lastbalancierung und Skalierbarkeit, da kleinere Teilaufgaben des Matchings parallel verarbeitet werden können. Die Evaluierung im Rahmen der *Ontology Alignment Evaluation Initiative* (OAEI) vergleicht GOMMA und andere Systeme für das Matching von Ontologien in verschiedenen Domänen. GOMMA kann u. a. durch Anwendung des parallelen und kompositionsbasierten Matchings sehr gute Ergebnisse bezüglich der Effektivität und Effizienz des Matchings, insbesondere für Ontologien aus dem Bereich der Lebenswissenschaften, erreichen.

# Inhaltsverzeichnis


**I Einleitung** **11**

**1 Einführung** **13**
1.1 Motivation . . . . . 13
1.2 Wissenschaftlicher Beitrag . . . . . 21
1.3 Aufbau der Arbeit . . . . . 24

**2 Verwandte Arbeiten** **27**
2.1 Ontologieevolution . . . . . 27
2.2 Evolution von schema- und ontologiebasierten Mappings . . . . . 30
2.3 Schema- und Ontologie-Matching . . . . . 40

**3 Grundlagen** **57**
3.1 Modelle . . . . . 57
3.2 GOMMA . . . . . 65

**II Evolution von Ontologiemappings** **71**

**4 Analyse der Evolution von Ontologiemappings** **73**
4.1 Motivation . . . . . 73
4.2 Modell für Ontologie- und Mappingänderungen . . . . . 74
4.3 Analyse der Mappingevolution . . . . . 79
4.4 Zusammenfassung . . . . . 84

**5 Evolutionsbasierte Bewertung von Ontologiemappings** **85**
5.1 Motivation . . . . . 85
5.2 Grundlagen . . . . . 87
5.3 Stabilitätsmaße . . . . . 88
5.4 Evaluierung . . . . . 90

5.5 Zusammenfassung . . . . . 95

**6 Adaptierung von Ontologiemappings 97**
6.1 Motivation . . . . . 97
6.2 Generelles Szenario . . . . . 98
6.3 Kompositionsbasierte Adaptierung . . . . . 99
6.4 Diff-basierte Adaptierung . . . . . 103
6.5 Evaluierung . . . . . 112
6.6 Zusammenfassung . . . . . 115

**III Evolution von Annotationsmappings 117**

**7 Evolution und Qualität von Annotationen 119**
7.1 Motivation . . . . . 119
7.2 Annotationsevolution - Modelle und Maße . . . . . 122
7.3 Bewertung der Annotationsstabilität . . . . . 126
7.4 Evaluierung . . . . . 128
7.5 Zusammenfassung . . . . . 134

**8 Einfluss der Ontologieevolution auf funktionale Analysen 135**
8.1 Motivation . . . . . 135
8.2 Methoden . . . . . 137
8.3 Ergebnisse und Diskussion . . . . . 143
8.4 Zusammenfassung . . . . . 149

**IV Matching großer Ontologien 151**

**9 Kompositionsbasiertes Matching 153**
9.1 Motivation . . . . . 153
9.2 Mappingkomposition . . . . . 155
9.3 Evaluierung . . . . . 160
9.4 Zusammenfassung . . . . . 163

**10 Paralleles Ontologie-Matching 165**
10.1 Motivation . . . . . 165
10.2 Parallelisierungsstrategien . . . . . 167

10.3 Verteilte Infrastruktur zum parallelen Ontologie-Matching . . . . . 173
10.4 Evaluierung . . . . . 174
10.5 Zusammenfassung . . . . . 178

**11 Evaluierung von GOMMA im Rahmen der OAEI 2012 179**
11.1 OAEI 2012 . . . . . 179
11.2 Präsentation des GOMMA-Systems . . . . . 180
11.3 Evaluierungsergebnisse . . . . . 185
11.4 Zusammenfassung . . . . . 191

**V Zusammenfassung und Ausblick 193**

**12 Zusammenfassung und Ausblick 195**
12.1 Zusammenfassung . . . . . 195
12.2 Ausblick . . . . . 199

**VI Anhang 203**

**A Einfluss der Ontologieevolution auf funktionale Analysen 205**
A.1 Evolution von GO . . . . . 205
A.2 Pseudocode zur Berechnung der Konzeptregionen und Regionenstabilität . . . . . 207
A.3 Evolutionsanalysen der realen Datensätze . . . . . 208
A.4 Simulierte Datensätze . . . . . 211

**B Evaluierung im Rahmen der OAEI 2012 215**

**Literatur 219**

# Teil I

# Einleitung

# 1 Einführung

## 1.1 Motivation

Heutzutage steht eine enorm große und wachsende Menge an Datenquellen zur Verfügung. Beispielsweise ist die Datenbankkollektion des *„Nucleic Acid Research“* Journals in den letzten 20 Jahren stetig gewachsen und umfasst im Jahr 2013 rund 1500 Datenbanken aus dem Bereich der Molekularbiologie [52]. Um quellübergreifende Auswertungen und Analysen zu ermöglichen, muss das Wissen verschiedener Quellen kombiniert werden [116]. Dies ist von besonderem Interesse für Wissenschaft und Forschung, weil dadurch wertvolle, neue Erkenntnisse gewonnen werden können. Zudem sollten Nutzer auf möglichst vollständiges Wissen zu einem Thema zugreifen können. Es ist beispielsweise sinnvoll, anhand der Symptome eines Patienten Therapievorschläge zu generieren. Dazu werden u. a. Datenquellen mit Informationen über diesen sowie andere Patienten, Krankheiten, Medikamente und Therapieformen benötigt. Jedoch können Daten unterschiedlicher Quellen sehr heterogen sein, da z. B. Synonyme oder Homonyme verwendet werden. Um Wissen in einer einheitlichen, strukturierten Form zu erfassen und einen formalen Wissensaustausch zwischen verschiedenen Anwendungen zu erleichtern, werden häufig standardisierte Vokabulare wie *Ontologien* verwendet. Eine Ontologie besteht aus formal spezifizierten Konzepten (Kategorien) und bildet ein gemeinsames Vokabular zur Repräsentation allgemeinen oder domänenspezifischen Wissens, z. B. über Krankheiten. Häufig dienen Ontologien als Metadaten der einheitlichen, semantischen Beschreibung (*Annotation*) von Objekten der realen Welt (Instanzen). Beispielsweise können Patientenakten mit den Konzepten einer Krankheitsontologie annotiert werden,

***Annotationsmapping***

| Proteine | Molekulare Funktionen |
|---|---|
| *Insulin* | *hormone activity* |
| *Insulin* | *protein binding* |
| *Tapasin* | *protein binding* |
| *Somatostatin* | *hormone activity* |
| ... | ... |

***Ontologiemapping***

| Anatomie des Menschen | Anatomie der Maus |
|---|---|
| *extremities* | *Limbs* |
| *head* | *Head and Neck* |
| *neck* | *Head and Neck* |
| *tailbone* | *Tail* |
| ... | ... |

Abbildung 1.1: Beispiele ontologiebasierter Mappings.

anstatt individuell einen unstrukturierten Text zur diagnostizierten Krankheit zu hinterlegen. Im Bereich der Molekularbiologie ist z. B. die Annotation von Genen oder Proteinen mit den Konzepten einer Ontologie zur Beschreibung ihrer molekularen Funktionen weit verbreitet (siehe Abbildung 1.1, links). Die Menge der Verknüpfungen von Instanzen einer Datenquelle mit den Konzepten einer Ontologie wird auch als *Annotationsmapping* bezeichnet. Annotationen können zum einen manuell durch Experten erstellt und zum anderen automatisch generiert werden. Beispielsweise übertragen einige Verfahren Annotationen von bereits annotierten Genen auf bisher nicht oder kaum annotierte Gene anhand der Ähnlichkeit ihrer Sequenzen (z. B. [86, 28]). Andere automatische Verfahren extrahieren Vorschläge für neue Annotationen aus der verfügbaren Literatur (z. B. [36, 67]).

Eine weitere Form ontologiebasierter Mappings bilden *Ontologiemappings*. Verschiedene Ontologien derselben Domäne beinhalten oft ähnliches und überlappendes Wissen. Beispielsweise listet die *OBOFoundry*[1] [164] circa 40 Anatomieontologien auf. Häufig existiert jedoch wenig Koordination zwischen den verantwortlichen Gruppen und Konsortien, die Ontologien entwickeln, so dass die Erstellung einer Abbildung (Mapping) zwischen ähnlichen Konzepten notwendig ist. Ontologiemappings enthalten eine Menge semantischer Beziehungen (*Korrespondenzen*) zwischen den Konzepten verschiedener Ontologien. Abbildung 1.1 (rechts) veranschaulicht beispielhaft Korrespondenzen zwischen Konzepten zweier Anatomieontologien. Oft ist der Aufwand einer manuellen Mappingbestimmung sehr hoch, so dass (semi-) automatische Methoden zum Abgleich der Ontologien zum Einsatz kommen [152, 49, 153]. Ontologiemappings sind für verschiedene Anwendungen nützlich. So ist ein Mapping zwischen Anatomieontologien der Maus und des Menschen für vergleichende Analysen hilfreich, wenn Erkenntnisse aus Experimenten im Modellorganismus Maus auf entsprechende anatomische Strukturen des Menschen übertragen werden sollen [18]. Weiterhin sind Ontologiemappings notwendig, um eine integrierte Ontologie wie die speziesübergreifende *Uber Anatomy Ontology* (Uberon) [135] durch Zusammenführen einzelner speziesspezifischer Ontologien zu erstellen. Solch eine integrierte Ontologie dient als Referenzontologie für den Wissensaustausch innerhalb einer Domäne. Allgemein verbessern ontologiebasierte Mappings die semantische Vernetzung der Quellen, helfen Heterogenitäten zwischen verschiedenen Datenquellen zu überwin-

[1] The Open Biological and Biomedical Ontologies Foundry: `http://www.obofoundry.org`, Stand 01.12.2013

den und erlauben deren Integration. Dadurch wird Nutzern und Anwendungen eine quellübergreifende Suche, Datenverarbeitung und Analyse ermöglicht.

Nach Gruber stellen Ontologien die explizite Spezifikation einer Konzeptualisierung dar [68]. Der Begriff „Ontologie“ umfasst ein großes Spektrum an Terminologien unterschiedlicher Ausdrucksstärke [114]. Dieses reicht von einfachen, kontrollierten Vokabularen, über Thesauri und Taxonomien, bis hin zu komplexen Ontologien, die logische Bedingungen wie Disjunktheit von Konzepten spezifizieren. Ontologien haben insbesondere in den Lebenswissenschaften enorm an Bedeutung gewonnen [19, 112]. Dort weisen sie meist eine graphartige Struktur auf, in welcher die Konzepte über gerichtete Beziehungen einer bestimmten Semantik miteinander verknüpft sind. Die häufigsten Beziehungstypen bilden *is-a* für Subklassen- und *part-of* für Teil-Ganzes-Beziehungen. Diese bilden zusammen mit den Konzepten einen gerichteten, azyklischen Graphen (*engl. directed acyclic graph*, DAG). Darüber hinaus existieren weitere domänenspezifische Beziehungstypen wie „proximal to“, welcher angibt wie nahe z. B. zwei Organe im Körper beieinander liegen.

Im Bereich der Lebenswissenschaften werden typischerweise biologische Objekte (wie Gene oder Proteine), Experimente, Publikationen oder elektronische Patientenakten (*engl.: electronic health records, EHR*) mit Konzepten einer Ontologie annotiert. Die weitverbreitete Nutzung von Ontologien spiegelt sich in der hohen Anzahl verfügbarer Ontologien wider. So stehen derzeit auf den Plattformen *OBOFoundry* und *BioPortal*[2] [141, 182] mehr als 350 verschiedene Ontologien aus dem Bereich der Lebenswissenschaften zur Verfügung. Eine der meist genutzten Ontologien ist die *Gene Ontology* (GO) [7]. GO umfasst Wissen zu biologischen Prozessen, molekularen Funktionen und zellulären Komponenten und dient der standardisierten Beschreibung der Eigenschaften von Genen und Proteinen. GO-Annotationen werden häufig in sogenannten funktionalen Analysen (*Term Enrichment* - Analysen) [176] verwendet, beispielsweise zur Identifikation signifikant überrepräsentierter Eigenschaften innerhalb einer Menge von Genen. Eine weitere sehr bekannte Ontologie ist der *National Cancer Institute Thesaurus* (NCIT) [162], der zur Annotation von Daten im Bereich der Krebsforschung genutzt wird. *SNOMED Clinical Terms* (SCT) [43] kommt hauptsächlich zur Annotation elektronischer Patientenakten (*engl.: electronic health records, EHR*) zum Einsatz. Hingegen werden die *Medical Subject Headings* (MeSH) [120] zur einheitlichen Beschreibung und Klassifikation von Publikationen in der bekannten Publikationsdatenbank *PubMed*[3] verwendet. Andere Ontologien umfassen Wissen zu chemischen Entitäten (*Chemical entities of biological interest*, ChEBI [39]), zur Anatomie verschiedener Spezies wie der Maus (*Adult Mouse Anatomy Ontology*, MA [84]) oder zu Spezies-Taxonomien wie z. B. die *Fly taxonomy* (FT) [69]. Ontologien in den Lebenswissenschaften sind teilweise sehr groß, d. h. sie enthalten viele Konzepte sowie Beziehungen zwischen den Konzepten. Beispielsweise umfasst GO mit seinen drei Subontologien ≈40.000

[2]NCBO BioPortal: `http://bioportal.bioontology.org/`;Stand 01.12.2013
[3]PubMed: http://pubmed.org

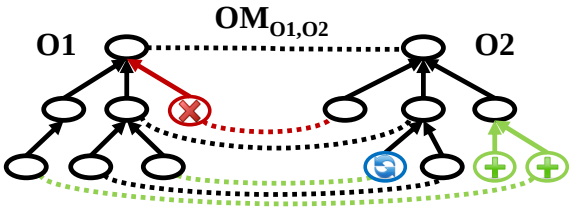



Abbildung 1.2: Ontologie- und Mappingevolution.

Konzepte, die durch mehr als 70.000 Beziehungen miteinander verknüpft sind. NCIT umfasst derzeit über 90.000 (100.000) Konzepte (Beziehungen). Mit seinen mehr als 300.000 Konzepten und über 1 Million Beziehungen ist SCT die derzeit größte verfügbare Ontologie in den Lebenswissenschaften. Der Metathesaurus des *Unified Medical Language System* (UMLS) [16] führt mehr als 100 biomedizinische Ontologien zusammen und bietet somit eine mächtige integrierte Datenquelle.

Typischerweise entwickeln sich Ontologien über die Zeit, d. h., sie werden regelmäßig angepasst (*Ontologieevolution*) [83, 74]. Insbesondere ist es von Interesse stets, den aktuellen Wissensstand einer Domäne zu repräsentieren. Beispielsweise liefern Experimente und Analysen neue Forschungserkenntnisse, die in die Ontologien eingepflegt werden sollen. Außerdem kann es zur Änderung einer Ontologie kommen, um u. a. veränderte Nutzeranforderungen zu unterstützen, initiale Designfehler zu beheben oder die Umsetzung bestimmter Richtlinien zur Umstrukturierung und Reorganisation zu realisieren [83, 74]. Im Bereich der Lebenswissenschaften veröffentlichen Gruppen und Konsortien, die Ontologien verwalten, regelmäßig und teilweise häufig neue Versionen. So gibt GO täglich eine neue Version frei. NCIT und UMLS veröffentlichen meist monatlich bzw. halbjährlich neue Versionen. Typischerweise enthalten die neuen Versionen verbessertes und erweitertes Wissen wie zusätzliche Konzepte, Beziehungen oder Attributwerte wie neue Synonyme. Allerdings wird auch existierendes Wissen in der Ontologie überarbeitet oder sogar gelöscht. So können einige Konzepte zu einem allgemeineren Konzept zusammengeführt werden (*engl.: merge*), wenn das Detailwissen der einzelnen Konzepte nicht mehr benötigt wird oder redundante Konzepte erkannt wurden.

Die Evolution der Ontologien hat Auswirkungen auf abhängige Mappings, die die Ontologien nutzen. Abbildung 1.2 zeigt beispielhaft die Evolution zweier Ontologien $O1$ und $O2$ sowie ein Mapping $OM_{O1,O2}$, das die beiden Ontologien verknüpft. In $O1$ wurde ein Konzept gelöscht (rot) und in $O2$ wurden zwei Konzepte hinzugefügt (grün). Ein weiteres Konzept in $O2$ wurde beispielsweise durch Veränderung des Konzeptnamens überarbeitet (blau). Diese Änderungen haben Auswirkungen auf Korrespondenzen (gestrichelte Linien), die zwischen den betroffenen Konzepten bestehen, und erfordern entsprechende Anpassungen. Im Beispiel führt die Konzeptlöschung zur Löschung einer Korrespondenz (rot gestrichelt), wohingegen eine Konzepthinzufügung und die Überarbeitung eines Konzepts zur Hinzufügung von zwei neuen Korrespondenzen (grün gestrichelt) führen.

Zuvor bestimmte ontologiebasierte Mappings können also ungültig werden, wenn die beteiligten Ontologien Änderungen unterliegen. Um die Mappings stets gültig bezüglich der aktuellen Ontologieversion(en) zu halten, ist es notwendig, diese entsprechend anzupassen. Automatisch generierte Ontologie- und Annotationsmappings könnten durch erneutes Anwenden des zugrunde liegenden Algorithmus aktualisiert werden, falls neue Ontologieversion(en) veröffentlicht werden. Dieses Vorgehen ist für manuell bestimmte, qualitativ hochwertige Mappings nicht sinnvoll. Eine vollständige, automatische Neuberechnung würde zu einer niedrigeren Qualität bezüglich der Genauigkeit (*engl.: Precision*) und Abdeckung (*engl.: Recall*) der Mappings führen. Eine manuelle Aktualisierung veralteter Mappings ist hingegen sehr aufwendig, da die zugrunde liegenden Ontologien, sowie die Mappings selbst, sehr groß sein können. Daher ist es sinnvoll, möglichst einen hohen Anteil bereits existierender Mappings wiederzuverwenden (z. B. schwarze Korrespondenzen in Abbildung 1.2) und nur die von Änderungen beeinflussten Mappingteile anzupassen. Die Anpassung von ontologiebasierten Mappings infolge der Ontologieevolution wird auch als *Mappingadaptierung* bzw. *Mappingmigration* bezeichnet. Ziel einer Mappingadaptierung ist es, möglichst automatisch zu bestimmen, welche Mappingteile wiederverwendet werden können und darüber hinaus Vorschläge zur Anpassung der von Änderungen betroffenen Teile eines Mappings zu generieren. Domänenexperten brauchen dann nur einen kleinen Teil des Mappings zu überprüfen und können aus einer Liste von Vorschlägen korrekte Korrespondenzen auswählen.

Im Folgenden werden einige Anforderungen an Verfahren zur Adaptierung ontologiebasierter Mappings diskutiert. Diese werden im Kapitel 2.2.3 (*Verwandte Arbeiten*) zum Vergleich existierender Verfahren sowie zur Einordnung und Abgrenzung dieser Arbeit herangezogen. Verfahren zur Adaptierung ontologiebasierter Mappings sollten folgende Anforderungen erfüllen:

- *Mappingqualität:* Ziel der Adaptierung ontologiebasierter Mappings ist es, eine hohe Mappingqualität zu erreichen. Dazu müssen zum einen korrekte Korrespondenzen (hohe Precision) und zum anderen ein vollständiges Mapping (hoher Recall) bestimmt werden. Ein wichtiges Qualitätskriterium ist die Konsistenz der migrierten Mappings bezüglich der neuen Ontologieversion(en). Adaptierungsverfahren sollten anhand realer Ontologie- und Mappingversionen hinsichtlich der Qualität der erzeugten Mappings evaluiert werden.

- *Konsistenz:* Die Adaptierung soll ein konsistentes Mapping erzeugen, das ausschließlich Korrespondenzen zu gültigen Konzepten der neuen Ontologieversion(en) enthält. Falls ungültige (z. B. gelöschte) Konzepte an Korrespondenzen beteiligt sind, entstehen Inkonsistenzen, die u. a. durch die Entfernung der betroffenen Korrespondenzen aufgelöst werden können.

- *Einbeziehen neuer Konzepte*: Um ein vollständiges Mapping zu erhalten, sollte ein Adaptierungsverfahren Ontologieerweiterungen (z. B. Konzepthinzu-

fügungen) berücksichtigen und gegebenenfalls Korrespondenzen zu hinzugefügten Ontologiebereichen erzeugen. Beispielsweise können (semi-) automatische Verfahren zum Abgleich von Ontologien eingesetzt werden, um Vorschläge für Korrespondenzen zu neuen Konzepten zu generieren.

- *Reduktion des manuellen Aufwands und Involvierung von Nutzern:* Aufgrund der Größe der Ontologien und Mappings sollte eine Adaptierung mit möglichst geringem manuellen Aufwand realisiert werden. Ziel ist es daher, möglichst große Teile eines von Ontologieänderungen betroffenen Mappings wiederzuverwenden und von Änderungen betroffene Bereiche eines Mappings automatisch zu identifizieren und entsprechend zu adaptieren. Um eine verbesserte Mappingqualität zu erreichen, sollten Nutzer in die Mappingadaptierung involviert sein. (Semi-)automatische Verfahren versuchen, zum einen den manuellen Adaptierungsaufwand zu reduzieren und zum anderen eine Verifikation und Korrektur von Korrespondenzen durch Nutzer zu unterstützen.

- *Unterstützung von semantischen Mappings:* Während der Migration von Ontologiemappings sollte die Semantik der Korrespondenzen berücksichtigt werden, um ein möglichst ausdrucksstarkes Mapping zu erzeugen. Häufig enthalten Ontologiemappings Korrespondenzen mit einer Äquivalenzsemantik, jedoch können auch andere Beziehungstypen wie z. B. „is-a“ auftreten. Insbesondere eine Adaptierung infolge komplexer Änderungen wie das Zusammenfassen mehrerer Konzepte zu einem Konzept (*merge*) erfordert spezielle Methoden, die die Semantik der Korrespondenzen berücksichtigen.

Neben dem Einfluss durch Ontologieevolution können existierende, ontologiebasierte Mappings weiteren Änderungen unterliegen. Beispielsweise können manuell gepflegte Mappings unvollständig sein, weshalb Experten an deren Vervollständigung arbeiten. So kann eine neue Korrespondenz zu einem Ontologiemapping hinzufügt werden, auch wenn die beteiligten Konzepte bereits seit längerem in der Ontologie vorliegen. Zudem erstellen Kuratoren regelmäßig neue Annotationen, weil z. B. experimentell nachgewiesen wurde, dass ein bestimmtes Gen eine bestimmte Funktion hat. Mappings unterliegen also Änderungen, die von der Ontologieevolution sowohl abhängig als auch unabhängig sind. Insgesamt wird die Veränderung der Mappings als *Mappingevolution* bezeichnet.

Für die Bestimmung von Ontologiemappings zwischen bisher nicht verknüpften Ontologien sowie für die (semi-) automatische Mappingadaptierung kommen automatische Verfahren zum Abgleich der Ontologien zum Einsatz (*Ontologie-Matching*, *engl.: ontology matching*) [49]. Das Hauptziel des Ontologie-Matchings ist es, qualitativ hochwertige Ontologiemappings bezüglich Precision und Recall zu produzieren. Darüber hinaus ist die Performanz von Match-Verfahren wichtig [161, 152]. Zum einen sind geringe Laufzeiten insbesondere für interaktive Applikationen wichtig, so dass Nutzer keine langen Wartezeiten in Kauf nehmen müssen. Zum anderen sollen die Match-Methoden skalierbar sein. Letzteres ist insbesondere für sehr große

biomedizinische Ontologien wie NCIT oder SCT von Interesse. Es existieren bereits zahlreiche Systeme zum Matching von Ontologien oder Schemas. Häufig fokussieren unterschiedliche Systeme auf bestimmte Aspekte wie eine hohe Mappingqualität, eine geringe Laufzeit oder domänenspezifische Verfahren. Die *Ontology Alignment Evaluation Initiative* (OAEI)[4] hat es sich zur Aufgabe gemacht, Systeme für das Matching von Schemas bzw. Ontologien vergleichend zu evaluieren, um deren Stärken und Schwächen bewerten zu können. Dabei werden u. a. die Qualität der Ergebnisse und die Laufzeit der Systeme analysiert. OAEI bietet Match-Aufgaben aus verschiedenen Bereichen wie z. B. den Abgleich von Vokabularen aus den Sozialwissenschaften (*Library Track*), Anatomieontologien (*Anatomy Track*) oder Ontologien zur Organisation von Konferenzen (*Conference Track*). Während der OAEI 2011.5 wurde zum ersten Mal der sogenannte *Large BioMed Track* durchgeführt. Ziel des Tracks ist es, Mappings zwischen den besonders großen biomedizinischen Ontologien NCIT, SCT sowie dem *Foundational Model of Anatomy* (FMA) [158] zu finden. Die OAEI 2012-Evaluierung[5] zeigte, dass nur wenige Systeme in der Lage sind diese sehr großen Ontologien abzugleichen. Daher ist es von Bedeutung, insbesondere skalierbare Match-Verfahren zu entwickeln [152].

Diese Arbeit beschäftigt sich mit der Evolution von ontologiebasierten Mappings sowie skalierbaren Methoden zur automatischen Bestimmung von Mappings hauptsächlich im Bereich der Lebenswissenschaften. Die Lebenswissenschaften sind ein sehr dynamisches Forschungsgebiet, so dass sich die dort verwendeten Ontologien sowie abhängige Mappings fortlaufend ändern. Die Frequenz von Ontologieänderungen wie Hinzufügungen und Löschungen von Konzepten und Beziehungen kann in verschiedenen Ontologien sowie innerhalb verschiedener Teile einer Ontologie stark variieren [76]. Nutzer möchten also wissen, inwieweit die Ontologieevolution ontologiebasierte Mappings beeinflusst. Verwendet ein Nutzer automatische Verfahren zur Erstellung von Mappings, interessiert ihn die Stabilität der verschiedenen Mappings, um Rückschlüsse auf die Robustheit der Verfahren bezüglich der Ontologieevolution zu ziehen. Falls die Ontologieevolution Auswirkungen auf abhängige Mappings hat, benötigen Nutzer aktuell gültige Versionen ihrer Mappings. Wenn z. B. eine neue Version einer in UMLS integrierten Ontologie erscheint, müssen die Mappings zu anderen UMLS-Teilen angepasst werden. Eine neue Mappingversion sollte qualitativ hochwertig sein und mit möglichst geringem manuellen Aufwand erstellt werden. Insbesondere für große Ontologien benötigt der Nutzer skalierbare, automatische Verfahren z. B. zur Bestimmung von Korrespondenzen zwischen hinzugefügten Ontologiebereichen. Darüber hinaus verwenden Nutzer Ontologien sowie assoziierte Mappings in weiterführenden Analysen. Beispielsweise nutzt ein Biologe die GO sowie GO-basierte Annotationen, um funktionale Analysen für eine von ihm erforschte Gruppe von Genen durchzuführen. Wenn nun neuere Eingabeversionen verfügbar sind, stellt sich die Frage, ob und wie sich Ergebnisse durch Verwendung

[4] `http://oaei.ontologymatching.org/`

[5] `www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2012/results2012`

der neuen Versionen verändern würden. Für den Forscher ist es also interessant, ob sich die „biologische Aussage“ seiner früheren Analyseergebnisse über die Zeit signifikant verändert oder nicht. Außerdem können Kuratoren das Wissen über die Ontologie- und Annotationsevolution in ihre Arbeit einbeziehen, um z. B. Nutzer davon in Kenntnis zu setzen, inwieweit geplante Änderungen semantisch bedeutsam für abhängige Anwendungen sein können. Aus den dargestellten Nutzerinteressen lassen sich die drei folgenden, in dieser Arbeit behandelten Themen ableiten:

**Analyse der Mappingevolution:** Bisher wurde die Evolution ontologiebasierter Mappings insbesondere in den Lebenswissenschaften nur wenig untersucht (siehe Kapitel 2.2). Beispielsweise ist bisher nicht bekannt, wie sich Ontologiemappings verändern und inwieweit die Mappings durch Ontologieevolution beeinflusst werden. Ontologiemappings können auf unterschiedliche Weise bestimmt bzw. berechnet werden. Es ist unklar, inwieweit verschiedene Match-Verfahren mehr oder weniger stabile Mappings produzieren. Ebenso liefert eine Analyse der Stabilität von Annotationen Hinweise zu deren Qualität. Umfassende Änderungen in Annotationsmappings können ebenso wie Ontologieänderungen starken Einfluss auf die Ergebnisse abhängiger Analysen und Experimente haben. Daher ist es von Interesse, den Einfluss der Ontologieevolution auf ontologiebasierte Mappings sowie auf weiterführende Analysen zu untersuchen. Die gewonnenen Erkenntnisse helfen dann einzuschätzen, ob zuvor bestimmte Mappings und Analyseergebnisse noch glaubwürdig und valide oder bereits veraltet sind. Wenn die Evolution der Ontologien einen starken Einfluss auf abhängige Mappings hat, ist eine Anpassung der Mappings notwendig.

**(Semi-) automatische Mappingadaptierung:** Eine weitestgehend automatische Adaptierung ontologiebasierter Mappings vermeidet die aufwendige Neubestimmung der Mappings, wenn sich die zugrunde liegenden Ontologien ändern. Vielmehr ist es sinnvoll, die stabilen Teile des alten Mappings wiederzuverwenden. Dies ist insbesondere für manuell bestimmte Mappings von Interesse, da somit qualitativ hochwertige Korrespondenzen erhalten bleiben. Die Migration ontologiebasierter Mappings ist nicht trivial, da komplexe Ontologieänderungen wie z. B. die Aufspaltung eines Konzepts in mehrere Konzepte (*engl. split*) auftreten können. Eine frühere Korrespondenz zu einem solchen Konzept müsste zu einer oder mehreren neuen Korrespondenzen abgeändert werden. Unterschiedliche Ontologieänderungen erfordern eventuell unterschiedliche Handlungen zur korrekten Adaptierung der Korrespondenzen. Bisher existieren nur wenige Arbeiten, die eine (semi-) automatische Adaptierung der Mappings anstreben (siehe Kapitel 2.2). Frühere Ansätze berücksichtigen beispielsweise nicht den Einfluss komplexer Änderungsarten oder die Einführung neuer Korrespondenzen infolge hinzugefügter Konzepte.

**Skalierbares Ontologie-Matching:** Bisher nicht verknüpfte Ontologien oder Ontologieteile können automatisch durch Verfahren des Ontologie-Matchings be-

stimmt werden, um den manuellen Aufwand zu reduzieren. Dies ist insbesondere für sehr große Ontologien von Interesse. Typischerweise ist es notwendig, das kartesische Produkt bezüglich der Größe zweier zu vergleichenden Ontologien zu berechnen (quadratische Komplexität). Viele der existierenden Match-Systeme skalieren jedoch nicht für das Matching der extrem großen Ontologien im Bereich der Lebenswissenschaften (siehe Kapitel 2.3). Daher ist es sinnvoll, effiziente Methoden zur Bestimmung qualitativ hochwertiger Ontologiemappings zu entwickeln. Beispielsweise können existierende Mappings wiederverwendet und miteinander kombiniert werden, um somit auf einem „indirekten Weg“ neue Mappings zu kreieren, ohne die Ontologien vollständig abgleichen zu müssen. Zudem ermöglicht die heutige Verfügbarkeit von Mehrkernprozessoren und gleichzeitige Nutzbarkeit mehrerer Rechner eine parallele Ausführung des Match-Prozesses, indem die Rechenlast durch Partitionierung der Match-Aufgabe in Teilaufgaben zerlegt und auf mehrere Rechenknoten verteilt wird. Außerdem ist es sinnvoll den Suchraum der Anzahl zu vergleichender Konzeptpaare zu reduzieren, um nicht das gesamte kartesische Produkt zweier Ontologien auswerten zu müssen.

## 1.2 Wissenschaftlicher Beitrag

Die Beiträge dieser Arbeit nutzen und erweitern die generische Infrastruktur GOMMA (**G**eneric **O**ntology **M**atching and Mapping **Ma**nagement) [102], die der Verwaltung und Analyse der Evolution von Ontologien sowie ontologiebasierten Mappings dient. GOMMA nutzt ein generisches Repository zur Versionsverwaltung und umfasst u. a. Komponenten zur Bestimmung von Ontologie- und Mappingänderungen sowie zum Matching von Ontologien. Die Infrastruktur fokussiert ursprünglich auf Applikationen im Bereich der Lebenswissenschaften. Der wissenschaftliche Beitrag dieser Dissertation besteht aus den folgenden Arbeiten im Bereich der Evolution von Ontologie- und Annotationsmappings sowie skalierbaren Techniken des Ontologie-Matchings:

**Evolution von Ontologiemappings:** Um zunächst die Evolution von Ontologiemappings und den Einfluss von Ontologieänderungen auf abhängige Mappings analysieren zu können, werden verschiedene Metriken zur Bestimmung von Evolutionsintensitäten in Ontologien und Ontologiemappings eingeführt. Darauf aufbauend erfolgt eine vergleichende Analyse zur Evolution automatisch generierter Mappings unter Verwendung verschiedener Match-Algorithmen für drei Anwendungsgebiete in den Lebenswissenschaften. Die Metriken unterstützen Nutzer, die Evolution von Ontologiemappings besser zu verstehen und helfen zu entscheiden, ob ein Mapping noch glaubwürdig ist oder eine aufwendige Anpassung durchgeführt werden muss. Eine weitere Studie hat

die evolutionsbasierte Bewertung von Ontologiemappings zum Ziel. Auf Basis historischer Korrespondenzähnlichkeiten werden Stabilitätswerte als zusätzliches Bewertungskriterium für automatisch generierte Korrespondenzen berechnet. Um den manuellen Aufwand einer Anpassung von Ontologiemappings zu minimieren, werden zwei Ansätze zur (semi-)automatischen Adaptierung eingeführt. Ein Algorithmus nutzt das Prinzip der Mappingkomposition (*kompositionsbasierte Adaptierung*). Alternativ passt der *Diff-basierte Adaptierungsalgorithmus* Mappings anhand unterschiedlicher Strategien zur individuellen Behandlung verschiedener Änderungsoperationen an. Beide Techniken basieren auf der Wiederverwendung nicht beeinflusster Korrespondenzen und adaptieren nur den beeinflussten Mappingteil. Die Evaluierung der Qualität erfolgt für Mappingversionen zwischen drei großen Ontologien aus den Lebenswissenschaften und zeigt, dass Ontologiemappings weitestgehend automatisch adaptiert werden können. Um eine manuelle Mappingmigration sowie die korrekte Adaptierung komplexer Fälle zu unterstützen, können zudem Vorschläge zur Verifikation durch Experten generiert werden.

**Evolution von Annotationsmappings:** Es wird ein Annotationsmodell eingeführt, das Informationen zur Herkunft der Annotationen bezüglich ihrer Erstellungsmethode sowie zur Evolution der Annotationen einbezieht. Darauf aufbauend können Maße zur Bewertung der Stabilität von Annotationen definiert werden. Diese dienen als Kriterien zur Bewertung der Annotationsqualität, wodurch glaubwürdige Annotationen zur Verwendung in weiterführenden Analysen identifiziert werden können. Die Evaluierung für GO-Annotationen in Swiss-Prot und Ensembl belegt eine signifikante Evolution von Annotationsmappings. Daraus ergibt sich die Frage, welche Auswirkungen die Evolution auf abhängige Anwendungen haben kann. Dies wird für die in den Lebenswissenschaften weit verbreiteten funktionalen Analysen von Gen- oder Proteinmengen empirisch getestet. Um den Einfluss der Ontologie- und Annotationsevolution auf funktionale Analysen zu untersuchen, werden zwei Maße zur Bewertung der Stabilität von Analyseergebnissen präsentiert. Für die Evaluierung werden funktionale Analysen für zwei reale sowie 50 simulierte Datensätze unter Verwendung verschiedener Eingabeversionen durchgeführt. Anschließend werden Ontologie- sowie Annotationsänderungen im Zusammenhang mit der Evolution der Ergebnisse betrachtet. Die Untersuchung zeigt, dass die Evolution von Ontologien und Annotationen durchaus Auswirkungen auf die Ergebnisse funktionaler Analysen hat, jedoch sind die Ergebnisse inhaltlich relativ stabil, so dass sich ihre biologische Aussage meist nicht grundsätzlich ändert.

**Skalierbares Matching großer Ontologien:** Zum Matching großer Ontologien werden zwei skalierbare Methoden vorgestellt. Ein *kompositionsbasierter Match-Ansatz* beruht auf der indirekten Bestimmung von Ontologiemappings durch Komposition bereits existierender Ontologiemappings. Der Ansatz ist besonders vielversprechend, wenn Mappings zu zentralen, semantisch reich-

haltigen Mediatorontologien einer betrachteten Domäne (*engl. hub*) wiederverwendet werden, da dies wertvolles Hintergrundwissen liefern kann. Neben einer hohen Effizienz kann so auch eine verbesserte Match-Qualität im Vergleich zum herkömmlichen (direkten) Ontologie-Matching erreicht werden. Weiterhin werden skalierbare Ansätze zum *parallelen Ontologie-Matching* unter Verwendung mehrerer Rechenknoten präsentiert. Dazu zählen die Inter- und Intra-Matcher-Parallelität sowie die Realisierung der parallelen Ausführung für element- und strukturbasierte Match-Verfahren. Abschließend erfolgt eine umfassende Evaluierung zum Ontologie-Matching mit GOMMA im Rahmen der OAEI 2012. Neben dem kompositionsbasierten und parallelen Matching kommt eine Methode zur Reduktion des Suchraums zum Matching besonders großer Ontologien zum Einsatz. Als generisches Werkzeug konnte GOMMA erfolgreich an allen geforderten Aufgaben in verschiedenen Domänen teilnehmen. Im Vergleich zu anderen Systemen erzielte GOMMA sehr gute Ergebnisse bezüglich der Match-Qualität und benötigten Laufzeiten. Insbesondere konnte GOMMA die beste Match-Qualität im *Anatomy* und *Library Track* erreichen.

Die in der vorliegenden Arbeit dargestellten Ergebnisse wurden bereits als begutachtete Beiträge bei internationalen Konferenzen, Workshops, oder Journals publiziert. Die generische Infrastruktur GOMMA wurde 2011 im *Journal of Biomedical Semantics* publiziert. GOMMA wurde im Rahmen dieser Dissertation in den folgenden Arbeiten zur Evolution ontologiebasierter Mappings eingesetzt und erweitert:

- Ein evolutionsbasierter Ansatz zur Bewertung von Ontologiemappings wurde 2009 auf der *BTW*-Konferenz präsentiert [174]. 2012 wurde eine Fallstudie zur Evolution von Ontologiemappings im Bereich der Lebenswissenschaften auf dem internationalen *EvoDyn*-Workshop im Rahmen der *ISWC* vorgestellt [66]. Die Algorithmen zur (semi-) automatischen Adaptierung von Ontologiemappings wurden auf der internationalen *DILS*-Konferenz 2013 präsentiert [60].

- Auf der *DILS*-Konferenz 2009 wurde die Arbeit zur Analyse der Evolution und Qualität von Annotationsmappings vorgestellt [61]. Die Studie zum Einfluss der Ontologie- und Annotationsevolution auf funktionale Analysen erschien 2012 in *Bioinformatics* [65].

- GOMMA's Match-Komponente wurde im Rahmen dieser Dissertation um Verfahren zum skalierbaren Matching großer Ontologien erweitert. 2011 wurde der Ansatz zum kompositionsbasierten Ontologie-Matching auf der internationalen Konferenz *ICBO* präsentiert [63]. Der Ansatz sowie die Infrastruktur zum parallelen Ontologie-Matching wurden auf der *DILS*-Konferenz 2010 vorgestellt [62]. GOMMA's Match-Komponente wurde im Rahmen der OAEI 2012 vergleichend mit anderen Match-Systemen evaluiert und auf dem zugehörigen *Ontology Matching*-Workshop präsentiert [64].

## 1.3 Aufbau der Arbeit

Die nachfolgende Arbeit gliedert sich in drei Hauptteile. Im ersten Teil werden verwandte Arbeiten diskutiert und Grundlagen beschrieben:

**Kapitel 2** gibt eine Einführung zur Evolution von Ontologien. Anschließend werden Arbeiten zur Evolution und Adaptierung von schema- und ontologiebasierten Mappings sowie zum Schema- und Ontologie-Matching diskutiert und von den in dieser Arbeit vorgestellten Verfahren abgegrenzt.

**Kapitel 3** führt die in dieser Arbeit verwendeten Modelle für Ontologien, Instanzen sowie für Ontologie- und Annotationsmappings ein. Außerdem wird das System GOMMA mit seinen Komponenten zum Matching von Ontologien und zur Bestimmung des Diffs zwischen Ontologieversionen vorgestellt.

Der zweite Teil – Evolution von Ontologiemappings – stellt zunächst eine vergleichende Analyse zur Evolution von Ontologiemappings vor. Weiterhin wird eine Studie zur Bewertung von Ontologiemappings mittels Stabilität präsentiert. Anschließend werden die zwei Ansätze zur Adaptierung von Ontologiemappings vorgestellt. Dieser Teil der Arbeit gliedert sich in die folgenden Kapitel:

**Kapitel 4** umfasst eine vergleichende Analyse zur Evolution von automatisch generierten Mappings unter Verwendung verschiedener Match-Algorithmen. Es werden das verwendete Evolutionsmodell sowie verschiedene Metriken eingeführt. Die Evaluierung betrachtet Mappings zwischen bekannten Ontologien aus drei Bereichen der Lebenswissenschaften für den Zeitraum 2006-2010.

**Kapitel 5** umfasst eine Studie zur Bewertung von Ontologiemappings auf Basis historischer Informationen. Es werden zwei Maße zur Berechnung der Stabilität von Korrespondenzen vorgestellt. Die Evaluierung untersucht eine instanzbasierte Match-Strategie für Mappings zwischen zwei Subontologien der GO.

**Kapitel 6** stellt zwei Ansätze zur (semi-)automatischen Adaptierung von Ontologiemappings vor. Zunächst werden die notwendigen Operatoren sowie der Algorithmus zur kompositionsbasierten Adaptierung vorgestellt. Anschließend werden der Diff-basierte Adaptierungsalgorithmus und mögliche Strategien zur individuellen Behandlung der Änderungen diskutiert. Die Evaluierung der Qualität erfolgt unter Verwendung von aus dem UMLS extrahierten Mappingversionen zwischen drei großen Ontologien aus den Lebenswissenschaften.

Der dritte Teil – Evolution von Annotationsmappings – stellt zunächst eine Studie zur Evolution und Qualität von Annotationen vor. Anschließend wird der Einfluss von Ontologie- und Annotationsevolution auf abhängige, funktionale Analysen untersucht. Der Teil umfasst folgende zwei Kapitel:

**Kapitel 7** stellt eine Studie zur Evolution und Qualität von Annotationsmappings vor. Zunächst werden das Evolutions- und Qualitätsmodell für Annotationen sowie Maße zur Bewertung der Annotationsstabilität präsentiert. Der Ansatz wird für funktionale GO-Protein-Annotationen in den zwei Datenquellen Ensembl und Swiss-Prot vergleichend evaluiert.

**Kapitel 8** untersucht den Einfluss der Ontologie- und Annotationsevolution auf funktionale Analysen. Dazu werden zwei Maße zur Bewertung der Stabilität von Analyseergebnissen präsentiert. Für die Evaluierung wurden funktionale Analysen für zwei reale sowie 50 simulierte Datensätze unter Verwendung von Ontologie- und Annotationsversionen im Zeitraum 2003-2010 durchgeführt.

Im vierten Teil – Matching großer Ontologien – werden zunächst zwei Ansätze zum effizienten Matching großer Ontologien präsentiert. Dazu zählt zum einen eine kompositionsbasierte Match-Strategie und zum anderen das parallele Matching von Ontologien. Anschließend wird eine umfassende Evaluierung zum Ontologie-Matching mit dem System GOMMA für verschiedene Domänen im Rahmen der OAEI gezeigt. Der Teil beinhaltet die folgenden drei Kapitel:

**Kapitel 9** führt eine Methode zur indirekten Bestimmung von Ontologiemappings durch Komposition zuvor bestimmter Ontologiemappings ein. Es werden die verwendeten Operatoren sowie der kompositionsbasierte Match-Ansatz vorgestellt. Die Evaluierung untersucht die Qualität sowie Laufzeiten anhand einer Match-Aufgabe im Bereich Anatomie.

**Kapitel 10** präsentiert verschiedene Ansätze zum parallelen Ontologie-Matching unter Verwendung mehrerer Rechenknoten. Dabei werden Inter- und Intra-Matcher-Parallelität sowie die Realisierung einer parallelen Ausführung für element- und strukturbasierte Match-Verfahren dargestellt. Anschließend wird die verteilte Infrastruktur zum parallelen Ontologie-Matching vorgestellt. Die Evaluierung bietet eine Analyse der verschiedenen Verfahren bezüglich ihrer Ausführungszeit und zeigt die Skalierbarkeit der Ansätze für große Ontologien.

**Kapitel 11** zeigt eine umfassende Evaluierung zum Ontologie-Matching mit dem System GOMMA im Rahmen der OAEI 2012. Neben den in den vorherigen Kapiteln eingeführten effizienten Match-Methoden wird ein Verfahren zur Reduktion des Suchraums zum Matching besonders großer Ontologien vorgestellt. Die Evaluierung umfasst Ergebnisse zur Match-Qualität und Laufzeit von GOMMA und weiteren teilnehmenden Systemen für die sechs Teilaufgaben der OAEI 2012.

Abschließend erfolgt eine Zusammenfassung der Ergebnisse der Dissertation sowie ein Ausblick auf zukünftige Arbeiten. Im Anhang werden weitere Details zu einigen Arbeiten präsentiert.

# 2

# Verwandte Arbeiten

Dieses Kapitel gibt zunächst eine Einführung zur Evolution von Ontologien (Kapitel 2.1). Dabei sind insbesondere Algorithmen zur Bestimmung von Unterschieden zwischen verschiedenen Versionen sowie fortgeschrittene Analysemöglichkeiten von Interesse, da diese im weiteren Verlauf dieser Arbeit verwendet werden. Anschließend werden bisherige Arbeiten zur Evolution und Adaptierung von schema- und ontologiebasierten Mappings vorgestellt und von den in dieser Arbeit vorgestellten Ansätzen abgegrenzt (Kapitel 2.2). Verwandte Arbeiten aus dem Bereich des Schema- und Ontologie-Matchings werden in Kapitel 2.3 diskutiert. Nach einer allgemeinen Einführung zu Verfahren des Ontologie-Matchings werden insbesondere skalierbare Match-Verfahren sowie Arbeiten zum Matching biomedizinischer Ontologien vorgestellt und von den in dieser Arbeit präsentierten Methoden abgegrenzt.

## 2.1 Ontologieevolution

Ontologien sowie andere Schemaarten wie z. B. relationale Datenbankschemas, konzeptionelle ER- oder UML-Modelle oder XML-Schemas werden regelmäßig an neue oder veränderte Anforderungen angepasst. Gründe für die Weiterentwicklung bzw. Evolution sind u. a. notwendige Fehlerkorrekturen, neue Erkenntnisse in einer Domäne oder die Bereitstellung neuer Funktionalitäten. Häufig werden neue Konzepte und Beziehungen zu den Ontologien hinzugefügt, jedoch werden auch veraltete Konzepte und Beziehungen aus den Ontologien entfernt. Die Änderungen im Schema oder in einer Ontologie müssen korrekt und effizient, d. h. mit möglichst geringem

manuellen Aufwand, auf abhängige Komponenten wie Instanzen, Datenbanksichten, Anwendungen und Mappings propagiert werden, um diese konsistent und aktuell zu halten. Die Artikel [53] und [83] geben einen Überblick zu Arbeiten im Bereich der Ontologieevolution und -versionierung. Hartung et al. [83] diskutieren insbesondere die Evolution von relationalen Schemas, XML-Schemas und Ontologien. Im Rahmen dieser Dissertation werden Vorarbeiten aus dem Bereich der Ontologieevolution verwendet, um Auswirkungen auf ontologiebasierte Mappings zu untersuchen und Änderungen zu propagieren. Dabei wird insbesondere die automatische Bestimmung eines Diff-Evolutionsmappings (Diff) zur Erkennung von Unterschieden zwischen verschiedenen Ontologieversionen benötigt (siehe Kapitel 3.1.3).

Arbeiten zur Evolution von Ontologien befassen sich u. a. mit der Versionierung von Ontologien [105, 106], der Entwicklung und Bearbeitung von Ontologien [167, 166], der kollaborativen Ontologieentwicklung [138] sowie der Bestimmung von Unterschieden (Diff) zwischen verschiedenen Versionen [140, 143, 75]. Klein et al. [105] stellen das OntoView-System zur Versionierung von Ontologien vor. OntoView unterstützt insbesondere Ontologieentwickler, verschiedene Versionen zu verwalten und die Beziehungen zwischen den verschiedenen Ontologieversionen zu spezifizieren. Dazu diskutieren sie Methoden zur Änderungserkennung zwischen verschiedenen Versionen insbesondere im Kontext des *Semantic Web*. Sogenannte nicht-logische Änderungen („Non-logical changes“) umfassen einfache Attributänderungen eines Konzepts wie z. B. eine Änderung des Konzeptnamens. Änderungen der logischen Definition („Logical definition changes“) betreffen hingegen u. a. Änderungen der Ontologiestruktur. Stojanovic et al. [167] formalisieren den Evolutionsprozess, indem Strategien zur eindeutigen Behandlung von kritischen Ontologieänderungen vorgeschlagen werden. Der KAON Prototyp (*Karlsruhe Ontology and Semantic Web Tool suite*) [179] stellt dazu eine graphische Nutzerschnittstelle zur schrittweisen Entwicklung und Bearbeitung von Ontologien zur Verfügung. Der vorgestellte Evolutionsprozess [167] erfolgt in sechs Phasen und wird für jede Änderung durchlaufen. Zunächst entscheidet sich der Entwickler, eine bestimmte Änderung in der Ontologie vorzunehmen (*Change Capturing*), welche anschließend in eine formale Repräsentation überführt wird (*Change Representation*). Dabei werden elementare Änderungen (z. B. Löschungen oder Einfügungen von Konzepten) und komplexe Änderungen (z. B. Zusammenfassen von Konzepten) unterschieden. Anschließend werden mögliche, aus der Änderung resultierende Inkonsistenzen identifiziert und behoben (*Semantics of Change*). So müssen z. B. Subkonzepte eines gelöschten Konzepts ebenfalls gelöscht oder innerhalb der Ontologie verschoben werden. Die geplanten Änderungen werden realisiert und protokolliert (*Change Implementation*), jedoch findet dabei keine Versionierung der Ontologie statt. Die anschließende *Change Propagation* -Phase sieht eine Propagierung der Änderungen in abhängige Anwendungen und Ontologien vor. Dies wird durch die rekursive Anwendung des Evolutionsprozess z. B. in einer abhängigen Ontologie umgesetzt. Abschließend können Änderungen überprüft und gegebenenfalls zurückgenommen werden (*Change Validation*).

Die Arbeit in [139] schlägt eine umfangreichere Liste einfacher und komplexer Änderungsoperationen zur Beschreibung der Unterschiede zwischen Ontologieversionen vor. Dazu zählen u. a. das Aufspalten (*split*) und Zusammenfassen von Konzepten (*merge*). Das Protégé-System [138] erlaubt die kollaborative Entwicklung von Ontologien. Dazu erfolgt eine zentrale Ontologieverwaltung sowie die Unterstützung konkurrierender Änderungen durch mehrere Entwickler (synchrone Entwicklung). Alternativ erlaubt eine asynchrone Entwicklung die lokale Änderung der Ontologie, wobei Änderungen verschiedener Nutzer später zusammengefasst und Konflikte behoben werden müssen. Es besteht die Möglichkeit, regelmäßig Versionen der Ontologie zu erstellen oder die vorgenommenen Änderungen in einem *Change Log* zu protokollieren. Änderungen können zudem durch Experten kontrolliert werden. Für den Fall, dass verschiedene Versionen einer Ontologie jedoch kein Log der Änderungen vorliegt, kann die Menge der Änderungsoperationen (Diff-Evolutionsmapping) zwischen zwei Versionen semi-automatisch bestimmt werden. Zur Berechnung eines Diff verwendet Protégé den PromptDiff-Algorithmus [140]. Der Fixpunktalgorithmus vergleicht zwei Versionen schrittweise und stoppt, wenn keine neuen Änderungen mehr gefunden werden. Zur Bestimmung der Änderungen (z. B. *add*, *delete*, *split*, *merge*) kommen verschiedene Match-Verfahren zum Einsatz.

Papavassiliou et al. [143] stellen ein Framework zur Bestimmung von sogenannten *High-Level*-Änderungen in RDF/S-Ontologien vor. Der Algorithmus erfasst zunächst die einfachen Unterschiede in Form von hinzugefügten und gelöschten RDF-Tripeln. Anhand einer Menge von Bedingungen und der optionalen Anwendung von heuristischen Matchern werden iterativ zusammengesetzte Änderungen identifiziert, so dass die Menge der einfachen Änderungen reduziert wird.

In eigener Vorarbeit wurde der COnto-Diff-Algorithmus [75] entwickelt. COnto-Diff bestimmt ein kompaktes, ausdrucksstarkes Diff-Evolutionsmapping. Der Algorithmus nutzt ein Match-Mapping zwischen Ontologieversionen, um zunächst einfache Basisänderungen (z. B. Hinzufügung / Löschung von Konzepten / Beziehungen) zu bestimmen. Durch Anwendung einer Regelmenge werden diese schrittweise zu komplexen Änderungsoperationen (z. B. Aufspalten / Zusammenfassen von Konzepten) aggregiert. Komplexe Änderungen und verbleibende Basisänderungen bilden ein kompaktes Diff-Evolutionsmapping, welches Experten unterstützt, bisherige Änderungen zwischen Versionen einfacher zu erfassen, um beispielsweise die Weiterentwicklung der Ontologie effizient zu realisieren. COnto-Diff wird im Rahmen dieser Arbeit zur Migration veralteter Mappings auf aktuelle Ontologieversionen verwendet. Der Algorithmus sowie die bestimmten Änderungsoperationen werden im Grundlagenkapitel (Kapitel 3.2.2) genauer vorgestellt.

Um die Evolution von Ontologien über längere Zeiträume zu erfassen und besser nachvollziehen zu können, haben sich verschiedene Arbeiten mit deren Analyse und Präsentation beschäftigt. Aufgrund des hohen Forschungsinteresses und der rasanten Entwicklung ändern sich Ontologien insbesondere in den Lebenswissenschaften teilweise sehr stark. Typischerweise werden biomedizinische Ontologien wie die

GO [55, 115] kontinuierlich bearbeitet und regelmäßig als neue Version veröffentlicht. Für GO existieren mehrere Arbeiten zur Evolutionsanalyse (z. B. [183, 146]) und Visualisierung von Änderungen (z. B. [144]). Eine umfassende Evolutionsanalyse von 16 Ontologien in den Lebenswissenschaften (z. B. GO, NCIT, MA) [81] zeigte die stetige Aktualisierung und ein signifikantes Wachstum der Ontologien über einen Zeitraum von vier Jahren. Das dazu vorgestellte Framework unterstützt u. a. die Erkennung von Änderungen zwischen verschiedenen Ontologieversionen wie Hinzufügungen und Löschungen von Konzepten. Im Kontext der Ontologieevolution wurden in eigenen Vorarbeiten verschiedene Werkzeuge entwickelt. Das System OnEX[6] [80] ermöglicht Nutzern, Online-Evolutionsanalysen für Ontologien aus den Lebenswissenschaften durchzuführen. Quantitative Statistiken geben einen Überblick zur Entwicklung der Ontologien über die Zeit (z. B. zum Ontologiewachstum und zur Anzahl bestimmter Änderungen). Darüber hinaus kann der Nutzer relevante Konzepte im Detail bezüglich früherer Änderungen analysieren. Der in [76] vorgestellte Algorithmus ermöglicht die Identifikation stabiler und instabiler Ontologieregionen durch Aggregation und Propagierung von Änderungen zwischen aufeinanderfolgenden Versionen. Stabile Regionen wurden seit längerem nicht bearbeitet, wohingegen instabile Regionen häufigen Änderungen unterliegen. Die Analyse-Funktionalitäten stehen dem Nutzer über die Online-Webapplikation REx[7] [29] zur Verfügung. Der Regionenalgorithmus [76] wird im Rahmen dieser Arbeit als Analysewerkzeug in Kapitel 8 eingesetzt. Der COnto-Diff-Algorithmus wurde insbesondere zur Bestimmung von Diff-Evolutionsmappings zwischen Versionen biomedizinischer Ontologien eingesetzt. Nutzer können aggregierte Änderungen zwischen Versionen ihrer Wahl über die Webapplikation CODEX[8] [78] einsehen.

## 2.2 Evolution von schema- und ontologiebasierten Mappings

Ein Schemamapping (Ontologiemapping) umfasst die semantischen Beziehungen bzw. Korrespondenzen zwischen den Elementen (Konzepten) eines Quell- und eines Zielschemas (einer Quell- und einer Zielontologie). Derartige Mappings müssen infolge von Schema- bzw. Ontologieevolution „regeneriert“ werden. Eine vollständige Neubestimmung der Mappings ist ressourcenintensiv. Insbesondere besteht für komplexe Schemas oder sehr große Ontologien ein hoher manueller Aufwand. Darüber hinaus ist unklar, inwieweit ein auf Basis neuer Schema- oder Ontologieversionen erstelltes Mapping noch das bisher korrekte Wissen des alten Mappings widerspiegelt. Wichtige, bereits bestätigte Informationen könnten verloren gehen. Dies gilt sowohl für eine manuelle Neubestimmung als auch für eine automatische Mapping-

[6] http://www.izbi.de/onex
[7] http://www.izbi.de/rex
[8] http://www.izbi.de/codex

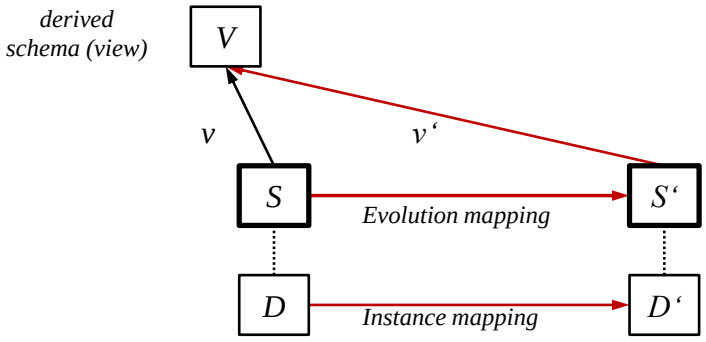



Abbildung 2.1: Szenario zur Schemaevolution (in Anlehnung an [83]).

erstellung durch Schema- oder Ontologie-Matching. Daher ist es sinnvoll möglichst große Teile des alten Mappings wiederzuverwenden. Im Folgenden werden zunächst einige relevante Ansätze aus dem Bereich der Schemaevolution und Adaptierung von Schemamappings vorgestellt (Kapitel 2.2.1). Anschließend werden Arbeiten zur Evolution und Adaptierung ontologiebasierter Mappings (Kapitel 2.2.2) diskutiert.

### 2.2.1 Schemaevolution und Adaptierung von Schemamappings

Schemaevolution stellt ein klassisches Problem des Metadaten-Managements dar [14]. Änderungen in Datenbankschemas führen zu Inkonsistenzen in abhängigen Daten und Anwendungen wie beispielsweise Instanzen oder Sichten (*Views*). Entsprechend dem Übersichtartikel [83] zeigt Abbildung 2.1 ein typisches Szenario der Schemaevolution. Wenn sich das Schema $S$ einer Datenbank mit Instanzen $D$ zu Schema $S'$ ändert, müssen die Instanzen entsprechend angepasst werden. Schemaänderungen von $S$ nach $S'$ werden in einem Evolutionsmapping beschrieben. Dieses Evolutionsmapping kann ein Schemamapping zwischen dem Quellschema $S$ und dem Zielschema $S'$ zur Beschreibung gleicher oder ähnlicher Konzepte sein. Alternativ kann ein Evolutionsmapping die Unterschiede (Diff) zwischen $S$ und $S'$ z. B. in Form einfacher oder komplexer Änderungsoperationen beschreiben. Ebenso können die notwendigen Änderungen zur Migration der Instanzen in einem Instanzmapping (von $D$ nach $D'$) hinterlegt werden. Änderungen im Schema $S$ haben außerdem Auswirkungen auf abhängige Anwendungen und müssen folglich propagiert werden. Anstelle einer aufwendigen Adaptierung von Applikationen aufgrund von Schemaänderungen ist es sinnvoll, stabile Views zu nutzen. Eine Anwendung basierend auf dem View $V$ sollte unbeeinflusst von Änderungen von $S$ nach $S'$ bleiben. Dazu muss der View $V$ erhalten bleiben, indem nur das View-Mapping $v$ entsprechend adaptiert wird ($v'$). Um das View-Mapping $v'$ ($M_{V,S'}$) zu erhalten, kann $v$ ($M_{V,S}$) mit einem Schemamapping zwischen $S$ und $S'$ ($M_{S,S'}$) kombiniert werden. Dazu kann beispielsweise der im Rahmen des Model Management [14, 131, 13] definierte `compose`-Operator ($\circ$)

verwendet werden, welcher die Zusammenfassung zweier sukzessiver Mappings zu einem Mapping erlaubt: ($M_{V,S'} = M_{V,S} \circ M_{S,S'}$).

Model Management ist ein generischer Ansatz zur leichteren Erstellung und Wartung metadaten-intensiver Applikationen [14, 131, 13]. Das Framework schlägt verschiedene generische Schema- und Mapping-Operatoren wie z. B. `match`, `compose` oder `merge` vor. Diese erlauben die Manipulation von Modellen (*models*) wie z. B. Datenbankschemas und Ontologien sowie Mappings zwischen diesen Modellen. Die Operatoren können u. a. helfen, die aus der Schemaevolution resultierenden Probleme zu lösen [131]. Die wesentlichen Operatoren für Modelle ($A$, $B$, $C$) und Mappings zwischen diesen Modellen (z. B. $M_{A,B}$) sind entsprechend [14, 131]:

- `match`: Erstellung eines Mappings $M_{A,B}$ zwischen zwei Modellen $A$ und $B$.
- `compose`: Kombination zweier Mappings $M_{A,B}$ und $M_{B,C}$ zu einem Mapping $M_{A,C}$.
- `merge`: Zusammenfassen zweier Modelle $A$ und $B$ zu einem dritten Modell $C$ unter Verwendung eines Mappings $M_{A,B}$.
- `extract`: Für ein Modell $A$ und ein Mapping $M_{A,B}$ zu einem anderen Modell $B$ wird der Teil des Modells $A$, der an dem betrachteten Mapping $M_{A,B}$ beteiligt ist, zurückgegeben.
- `diff`: Wie `extract`, jedoch wird der Teil von $A$ zurückgegeben, der nicht an $M_{A,B}$ beteiligt ist[9].

Wenige Arbeiten befassen sich explizit mit der Adaptierung von Mappings infolge von Schemaevolution [178, 184]. Wenn sich Schemas weiterentwickeln, sollte eine vollständige Neubestimmung der abhängigen Mappings vermieden werden. Ziel ist es, die Originalmappings möglichst für die Erstellung einer angepassten, konsistenten Mappingversion wiederzuverwenden. Velegrakis et al. [178] verfolgen einen inkrementellen Ansatz zur Migration von Schemamappings infolge von Schemaevolution. Existierende Mappings werden wiederverwendet und schrittweise, entsprechend der im Schema vorgenommenen Änderungen, angepasst. Dabei werden einfache Änderungsoperationen an Schemaelementen wie Löschungen, Umbenennungen, Verschiebungen und das Kopieren von Elementen berücksichtigt. Um ein konsistentes Mapping zu erhalten, werden im Falle einer Elementlöschung beeinflusste Korrespondenzen aus abhängigen Mappings entfernt. Für Umbenennungen, Verschiebungen und das Kopieren von Elementen müssen beeinflusste Korrespondenzen aktualisiert werden, indem beispielsweise eine Namensänderung entsprechend propagiert wird. Die Hinzufügung von Schemaelementen wird in diesem Ansatz nicht behandelt, da

[9] Der im Rahmen des Model Management definierte `diff`-Operator entspricht nicht dem $diff$ im Sinne der Berechnung von Unterschieden zwischen verschiedenen Ontologieversionen.

Elementhinzufügungen keine Inkonsistenzen im Mapping hervorrufen. Für die Evaluierung wurden manuelle Änderungen an kleinen Schemas (16-159 Elemente) vorgenommen. Von den Änderungen betroffene Mappings wurden mit dem inkrementellen Ansatz adaptiert, wodurch der manuelle Aufwand zur Bestimmung eines gültigen Mappings infolge von Schemaänderungen deutlich reduziert werden konnte.

Alternativ zu dem änderungsbasierten Ansatz stellen Yu et al. [184] einen kompositionsbasierten Ansatz zur Mappingadaptierung vor. Dabei wird ein Schemamapping zwischen der alten und neuen Schemaversion zur Repräsentation der Schemaevolution bestimmt. Durch Komposition dieses Mappings mit dem ursprünglichen Schemamapping wird das gültige Mapping bezüglich der neuen Schemaversion(en) erstellt. Im Gegensatz zum vorherigen änderungsbasierten Ansatz wird hier eine mappingbasierte Repräsentation der Schemaevolution verfolgt. Die Autoren argumentieren, dass dieser Ansatz allgemeingültiger und präziser ist, insbesondere in Bezug auf die Migration der Daten des alten Schemas zum neuen Schema (Instanzmigration). Der änderungsbasierte Adaptierungsalgorithmus beruht zudem auf einer vordefinierten, endlichen Menge von Änderungen und muss für jede primitive Änderung ausgeführt werden. Eine Evaluierung für kleine Schemas (87-265 Elemente) und einige manuell eingeführte Schemaänderungen zeigt eine Reduzierung des Nutzeraufwands für den kompositionsbasierten Ansatz gegenüber einer Neubestimmung durch Schema-Matching und die dadurch erforderliche manuelle Korrektur. Insbesondere wenn sich nur wenige Schemaelemente ändern, können große Mappingteile wiederverwendet werden und Experten müssen nur adaptierte Bereiche kontrollieren. Ziel der kompositionsbasierten Adaptierung war es zudem, eine korrekte Komposition zur Migration der Instanzen bereitzustellen.

Im Kontext der Schemaevolution und Mappingadaptierung spielt neben `compose` der `inverse`-Operator eine wichtige Rolle. Ein Übersichtsartikel [50] zeigt wie die Mappingoperatoren `compose` und `inverse` zur Adaptierung von Schemamappings genutzt werden können. Die Autoren führen eine ausführliche theoretische Diskussion zur Semantik, zu Algorithmen und zu Implementierungen von Komposition und Inversion. Beispielsweise muss es in einem komplexen Workflow zur Datenintegration möglich sein, einmal vorgenommene Änderungen wieder rückgängig zu machen, um z. B. eine alte Version wiederherzustellen. Problematisch ist dabei insbesondere die Migration von Instanzen. Es ist nicht in jedem Fall möglich, die exakte Inversion (d. h. eine vollständige und korrekte Wiederherstellung der alten Daten) zu bilden. Falls Informationen im Schema bzw. Mapping verloren gehen, können zugeordnete Instanzen nicht einfach rekonstruiert werden. Daher kommen zur Wiederherstellung der Daten verschiedene Annäherungen an das Inverse (z. B. *Quasi-Inverse*) zum Einsatz.

Darüber hinaus existieren Arbeiten, die sich mit der Migration von Anfragen an veraltete Schemas auf neue veränderte Schemas beschäftigen (*Query Migration*). Für Mappings ist eine Adaptierung notwendig, wenn sich das Quell- und/oder Zielschema ändert. Im Gegensatz zu Mappings hängen Anfragen nur von einem Schema

ab und müssen bei dessen Änderung angepasst werden. Prinzipiell können jedoch ähnliche Verfahren wie zur Adaptierung von Schemamappings eingesetzt werden. Die *PRISM workbench* [34] ist ein System, das die Migration von Anfragen auf Schemas zum Ziel hat. PRISM nutzt Techniken aus dem Bereich des *Query Rewriting* und Mappingoperationen wie Komposition und Inversion, um eine Migration der Anfragen zu realisieren.

Die Arbeiten auf dem Gebiet der Schemaevolution und der dadurch notwendigen Adaptierung von Schemamappings können ebenfalls für die Migration von Ontologiemappings nützlich sein. Schemas und Ontologien unterscheiden sich u. a. bezüglich der Rolle der Instanzen [83, 74]. Einige Ontologien beinhalten Instanzen, trennen diese jedoch nicht klar von den eigentlichen Ontologiekonzepten. Andere Ontologien beschreiben Instanzen, die außerhalb der Ontologie verwaltet werden (Annotationen). Zudem existieren für einige Ontologien keine nutzbaren Instanzdatensätze. Beispielsweise stellen die Instanzen der Konzepte einer Anatomieontologie die konkreten Ausprägungen der anatomischen Bereiche im Körper eines Menschen dar. Ontologien und Instanzen werden also häufig nicht zusammen verwaltet. Stattdessen nutzen viele Anwendungen (z. B. ontologiebasierte Annotationen oder Anfragen) die Ontologien, so dass eine Anpassung dieser abhängigen Applikationen an neue Ontologieversionen notwendig ist. Insbesondere in den Lebenswissenschaften werden Ontologien sehr stark genutzt und häufig neue Ontologieversionen veröffentlicht, so dass gerade in dieser Domäne viele abhängige Mappings und Anwendungen existieren, die infolge von Ontologieänderungen adaptiert werden müssen.

### 2.2.2 Evolution und Adaptierung von ontologiebasierten Mappings

Einige Arbeiten beschäftigten sich zunächst mit einer Analyse der Evolution ontologiebasierter Mappings. In [81] wurde neben der Evolution von Ontologien auch die Evolution von Annotations- und Ontologiemappings betrachtet. Dabei wurde ein starker Zuwachs an GO-Annotationen in der Datenquelle Ensembl [91] beobachtet. Zudem zeigte sich ein relativ hoher Anteil an Annotationslöschungen aufgrund zahlreicher Instanzlöschungen in Ensembl. Eine Analyse der Evolution automatisch generierter Ontologiemappings zwischen GO-Subontologien zeigte insbesondere Instabilitäten für instanzbasierte Match-Verfahren auf. Die Studie verwendete quantitative Metriken bezüglich der Mappinggröße und -abdeckung, untersuchte jedoch nicht die Auswirkungen der Ontologieevolution auf abhängige, ontologiebasierte Mappings. Dos Reis et al. [44, 45] präsentieren das DyKOSMap-Framework, welches u. a. die Adaptierung von Mappings zwischen biomedizinischen Ontologien zum Ziel hat. Anhand einiger Beispiele diskutiert die Arbeit [44] die Rolle verschiedener Ontologieänderungsoperationen für die Adaptierung von Mappings sowie Techniken zur Wiederherstellung von Ontologiemappings infolge der Ontologieevolution. Eine initiale

Analyse für die biomedizinischen Ontologien NCIT und MedDRA (*Medical Dictionary for Regulatory Activities*) [23] zeigt, dass nur ein Teil der Ontologieänderungen Auswirkungen auf abhängige Mappings hat. Die Autoren schlussfolgern, dass es nicht sinnvoll ist, Mappings infolge von Evolution regelmäßig neu zu berechnen. Stattdessen sollten Techniken für eine möglichst automatisierte, intelligente Migration entwickelt werden. Die Fallstudie motiviert die Notwendigkeit der Mappingadaptierung insbesondere in den Lebenswissenschaften, bietet jedoch noch keine konkrete Strategie zur (semi-) automatischen Migration von Ontologiemappings.

Kondylakis et al. [107] untersuchen die Evolution von Ontologien, um die Migration von ontologiebasierten Anfragen (*query migration*) zu unterstützen. Ziel ist es, Entwicklern zu helfen, invalide Anfragen zu identifizieren und diese gegebenenfalls manuell zu adaptieren. Dazu werden Sequenzen vorangegangener Ontologieänderungen für beeinflusste, potenziell invalide Anfragen ermittelt, so dass deren Aktualisierung erleichtert und die dafür benötigte Zeit reduziert wird. Der Ansatz unterstützt außerdem Ontologiedesigner, vergangene Modellierungsentscheidungen besser nachzuvollziehen. Die Autoren beschränken die Anwendbarkeit des Ansatzes auf valide RDF/S-Wissensdatenbanken und identifizieren den Einfluss der Evolution von RDF/S-Ontologien auf SPARQL-Anfragen. Zur automatischen Bestimmung von Unterschieden zwischen Ontologieversionen wird der von den Autoren veröffentlichte Algorithmus in [143] zur Bestimmung einfacher und komplexer Änderungsoperationen genutzt. Aus der Menge der Änderungsoperationen werden jene extrahiert, welche Auswirkungen auf Anfragen haben („*Affected Queries Detection*“ (AQD)-Modul). Für Änderungen, die Anfragen beeinflussen, werden dann sogenannte Änderungspfade (*change path*) berechnet. Ein Änderungspfad ist eine Sequenz von Änderungsoperationen, die in der Vergangenheit (zwischen der ersten und letzten betrachteten Version) aufgetreten sind. In der Evaluierung wurden u. a. Versionen der GO über ein halbes Jahr sowie GO-basierte Anfragen der AmiGO-Plattform [26] untersucht. Die Untersuchung zeigte, dass durch eine steigende Anzahl von Änderungen ein höherer Anteil der untersuchten Anfragen beeinflusst wird. Die benötigte Zeit zur Erstellung der Änderungspfade stieg linear im Vergleich zur Anzahl zu prüfender Änderungen. Der Ansatz bietet Entwicklern eine Unterstützung zur manuellen Migration ontologiebasierter Anfragen, indem Hintergrundwissen zur Änderungshistorie beeinflusster Anfragen zur Verfügung gestellt wird. Konkrete Vorschläge zur Adaptierung invalider Anfragen werden nicht generiert.

Khattak et al. [100] stellen einen Ansatz zur teilweisen Neuberechnung von Ontologiemappings vor, wenn diese durch Änderungen in den zugrunde liegenden Ontologien beeinflusst werden. Zur Bestimmung von Ontologieänderungen wird ein *Change History Log* (CHL) [101] genutzt. Dieser umfasst Basis-Änderungsoperationen (*create*, *update*, *delete*) für verschiedene Ontologiekomponenten (z. B. *ClassChange*, *PropertyChange*). Geänderte Elemente der Quell- bzw. Zielontologie werden jeweils durch Anwendung eines Match-Verfahrens (siehe Kapitel 2.3) mit der gesamten anderen aktuellen Ontologieversion abgeglichen. Anschließend erfolgt eine Aktualisie-

rung des veralteten Mappings, wobei alle veralteten Korrespondenzen entfernt und neu berechnete hinzugefügt werden. Somit werden zwar Korrespondenzen zu neuen Konzepten erstellt, jedoch werden nur von der Evolution unbeeinflusste Korrespondenzen wiederverwendet und viele der existierenden Korrespondenzen verworfen. Die Methode differenziert nicht zwischen verschiedenen einfachen (z. B. Attributänderung) oder komplexen (z. B. Aufspaltung von Konzepten) Änderungsarten, sondern „überschreibt“ sämtliche von einer Änderung betroffenen Korrespondenzen. In der Evaluierung werden Mappings zwischen den Ontologien (u. a. zwischen MA und NCIT) unter Verwendung verschiedener Match-Systeme (z. B. Falcon [87], TaxoMap [72]) automatisch berechnet. Anstelle realer Versionen der Ontologien werden manuell 25 Änderungen vorgenommen. Diese umfassen hauptsächlich Hinzufügungen, jedoch werden die Änderungen nicht im Detail aufgelistet und erläutert. Mappings zwischen den geänderten Ontologieversionen werden durch die Match-Systeme sowie deren Erweiterung um die vorgestellte Methode berechnet. Im Vergleich zur vollständigen Neuberechnung der Mappings erreicht das Matching der geänderten Ontologiebereiche (teilweise Neuberechnung) geringere Laufzeiten. Allerdings produzieren beide Methoden unterschiedliche Ergebnisse, wobei die Qualität (Korrektheit) der berechneten Mappings nicht evaluiert wird.

Martins et al. [125] schlagen einen Ansatz zur Evolution von Ontologiemappings vor, wobei mögliche Inkonsistenzen in Mappings, abhängig von der zuvor angewandten Evolutionsstrategie in der Ontologie, aufgelöst werden sollen. Die Autoren unterscheiden elementare Änderungen an Ontologiemappings (z. B. die Hinzufügung oder Löschung von Attributwerten des Quell- oder Zielkonzepts) und zusammengesetzte Mappingänderungen (z. B. Änderung von Attributwerten des Quell- oder Zielkonzepts). Der vorgeschlagene Mappingevolutionsprozess versucht für jede von Änderungen betroffene Korrespondenz, den zuvor angewandten Evolutionsprozess der Ontologie zu identifizieren. Wenn mindestens eine Strategie ermittelt wird, wird eine Menge von Mappingänderungen durchgeführt. Die Arbeit diskutiert den Evolutionsprozess in der Ontologie selbst und einem davon betroffenen Mapping beispielhaft anhand einer Konzeptlöschung. Entsteht ein verwaistes Ontologiekonzept, wird dieses unterhalb des Wurzelkonzepts neu eingehängt („reconnect orphaned concept to root"). Eine ungültige Korrespondenz zu dem gelöschten Konzept wird aus dem Mapping entfernt oder zum Elternkonzept des gelöschten Konzepts umgehangen. Da der Evolutionsprozess beispielhaft diskutiert wird, bleibt unklar, ob in jedem Fall ein konsistentes Mapping bezüglich der neuen Ontologieversion(en) erzeugt wird. Für die Evaluierung haben Experten vier Mappingversionen zwischen Ontologien von jeweils 15-25 Konzepten erstellt. Für den Vergleich der automatisch adaptierten mit den manuell erstellten Mappingversionen werden keine konkreten Evaluationsergebnisse bezüglich der Qualität der erzeugten Mappings beschrieben.

Das System OnEX [80] erlaubt die Migration von Annotationen zwischen biologischen Objekten (z. B. Genen) und Ontologiekonzepten infolge der Veröffentlichung einer neueren Version. Dazu werden zunächst Unterschiede zwischen der alten und

neuen Ontologieversion bestimmt. Die Adaptierung einer Annotation erfolgt, falls das beteiligte Ontologiekonzept gelöscht, auf veraltet („obsolet“) gesetzt oder mit einem anderen Konzept (*fuse*) zusammengefasst wurde. Der Nutzer erhält eine Übersicht der beobachteten Änderungen sowie eine Liste der durch diese Änderungen beeinflussten Annotationen. Auf Basis dieser Informationen kann er die veralteten Annotationen auf die neue Ontologieversion migrieren. Im Falle einer Konzeptlöschung werden die abhängigen Annotationen ebenfalls gelöscht. Annotationen zu obsoleten Konzepten können gelöscht oder aktualisiert werden. Beispielsweise veröffentlicht GO für obsolete Konzepte ein Mapping zu alternativen, gültigen Konzepten, die OnEX dem Nutzer gegebenenfalls präsentiert. Wenn ein Konzept mit einem anderen Konzept zusammengefasst wird, können abhängige Annotationen gelöscht oder zu dem neuen Zielkonzept migriert werden. OnEX ermöglicht die Migration ungültiger Annotationen infolge einiger „informationsreduzierender“ Änderungen, unterstützt jedoch keine weiteren Änderungsarten wie Hinzufügungen oder das Aufspalten von Konzepten. Die Arbeit präsentiert die praktische Anwendung der Annotationsmigration für Nutzer, allerdings wurden die verwendeten Änderungsoperationen und Adaptierungsstrategien nicht formal und detailliert vorgestellt.

### 2.2.3 Zusammenfassung und Abgrenzung der eigenen Arbeit

Ein Großteil der existierenden Ontologien und Schemas wird an neue oder veränderte Anforderungen angepasst. Ontologieänderungen haben Auswirkungen auf abhängige Daten und Applikationen wie z. B. Ontologie- und Annotationsmappings. Diese können somit ungültig werden und müssen entsprechend angepasst bzw. aktualisiert werden. Die Nutzung von Ontologien und ontologiebasierten Mappings ist insbesondere im Bereich der Lebenswissenschaften weit verbreitet. Die intensive Erforschung der Domäne führt zu zahlreichen neuen Erkenntnissen und somit zur regelmäßigen Erweiterung und Überarbeitung der Ontologien. Nutzer von Ontologien und ontologiebasierten Anwendungen in den Lebenswissenschaften müssen dementsprechend häufig mit der Veröffentlichung neuer Ontologieversionen umgehen.

Im Gegensatz zur Evolution von Ontologien wurde die Evolution und Adaptierung von ontologie- bzw. schemabasierten Mappings bisher nur wenig untersucht. Erste Analysen [81, 44] verdeutlichen, dass ontologiebasierte Mappings im Bereich der Lebenswissenschaften Änderungen unterliegen, jedoch ist nicht bekannt, wie und durch welche Änderungen die Mappings beeinflusst werden. Aus diesen Gründen untersucht diese Dissertation die Evolution von Ontologie- und Annotationsmappings, insbesondere im Bereich der Lebenswissenschaften. Dazu werden zunächst verschiedene Maße eingeführt, die Aufschluss über die Stabilität und Glaubwürdigkeit von Ontologiemappings infolge der Evolution der zugrunde liegenden Ontologien geben (Kapitel 4, 5). Dabei wird u. a. der Einfluss verschiedener Match-Verfahren auf die Evolutionsintensität der Mappings analysiert. Zudem werden die Evolution von Annotationsmappings (Kapitel 7) sowie Auswirkungen der Ontologie- und

Annotationsevolution auf funktionale Analysen großer biologischer Datensätze untersucht (Kapitel 8). Ziel der Arbeiten ist es, Nutzer zu unterstützen, u. a. die Auswirkungen der Ontologieevolution auf abhängige ontologiebasierte Mappings und Analyseergebnisse einschätzen und bewerten zu können.

Aufgrund der Evolution von Ontologien besteht die Notwendigkeit, Mappings auf gültige Ontologieversionen zu migrieren. Ein wichtiger Beitrag dieser Arbeit sind zwei Ansätze zur (semi-) automatischen Adaptierung ontologiebasierter Mappings (Kapitel 6). Tabelle 2.1 fasst die Eigenschaften existierender Verfahren basierend auf den in Kapitel 1 diskutierten Anforderungen zur Adaptierung von Mappings zusammen und vergleicht diese mit den Ansätzen der vorliegenden Arbeit.

Der eigene kompositionsbasierte Ansatz verwendet ein Ontologiemapping zwischen Ontologieversionen und kombiniert dieses mit dem veralteten Mapping. Hingegen nutzt das eigene Diff-basierte Adaptierungsverfahren ein komplexes Diff-Evolutionsmappings zur Mappingmigration. Beide Ansätze versuchen möglichst große Teile zuvor bestimmter Mappings wiederzuverwenden. Bisherige Verfahren versuchen ebenfalls bestehende Korrespondenzen zu erhalten, jedoch unterscheidet sich der Grad der Wiederverwendung teilweise deutlich. Beispielsweise nutzen Khattak et al. [100] ein Diff-Evolutionsmapping, um von Änderungen betroffene Korrespondenzen zu löschen und anschließend alle geänderten Ontologiebereiche automatisch mit der anderen Ontologie abzugleichen. Dadurch entsteht zwar ein aktuelles Mapping, jedoch ist dessen Qualität fragwürdig. Bisherige Ansätze betrachten zumeist nur ein Evolutionsmapping für einfache Änderungen und berücksichtigen nicht die Auswirkungen komplexer Änderungsoperationen wie *merge* oder *split*. Kondylakis et al. [107] nutzen zwar ein komplexes Diff-Evolutionsmapping, jedoch unterstützen sie Nutzer nur durch das Aufzeigen von Ontologieänderungen und bieten keine konkreten Vorschläge für eine Migration ungültiger, ontologiebasierter Anfragen. Die meisten der existierenden Ansätze erzeugen ein konsistentes Mapping bezüglich der neuen Ontologieversionen, verzichten jedoch auf die Erstellung neuer Korrespondenzen zu hinzugefügten Konzepten. Im Gegensatz dazu wird in dieser Arbeit die Erstellung eines konsistenten und möglichst vollständigen Mappings verfolgt. Dazu werden Vorschläge für Korrespondenzen zu neuen Konzepten generiert, welche durch Nutzer verifiziert werden können. Insbesondere der Diff-basierte Ansatz bietet individuelle Strategien zur Adaptierung infolge verschiedener komplexer Ontologieänderungen wie *merge* und *split*. Beide Verfahren berücksichtigen zudem die Semantik (z. B. Äquivalenz, „more general“-Beziehung) der erzeugten Korrespondenzen, wohingegen existierende Ansätze eine einheitliche Mappingsemantik unterstellen. Bisher erfolgte eine Evaluierung der Adaptierungsverfahren zumeist anhand einiger manueller Änderungen in teilweise sehr kleinen Ontologien bezüglich der Zeitersparnis gegenüber einer Neubestimmung der Mappings. Im Gegensatz dazu wird in dieser Arbeit insbesondere die Qualität der adaptierten Mappings anhand realer Mappings zwischen Versionen sehr großer Ontologien wie z. B. NCIT und SCT evaluiert.

| | Velegrakis et al. 2003 [178] | Yu and Popa 2005 [184] | Kondylakis et al. 2012 [107] | Khattak et al. 2012 [100] | Martins and Silva 2009 [125] | Hartung et al. 2009 [80] (OnEX) | Eigene Ansätze (Kap. 6) |
|---|---|---|---|---|---|---|---|
| **Beschreibung** | Änderungs-basierte Adaptierung | Kompositions-basierte Adaptierung | Bestimmung invalider Anfragen | Neuberechnung für alle geänderten Ontologieteile | Anwendung der Ontologie-evolutionsstrategie | Migration via GUI für vordefinierte Ontologien | Kompositions- und Diff-basierte Adaptierung |
| **Eingabe** | | | | | | | |
| **Zu adaptier. Mapping** | Schemamapping | Schemamapping | Ontologiebasierte Anfragen | Ontologiemapping | Ontologiemapping | Ontologiebasiertes Annotationsmapping | Ontologiemapping |
| **Genutztes Evolutions-mapping** | einfacher Diff | Schemamapping | komplexer Diff | einfacher Diff | einfacher Diff | einfacher Diff | Ontologiemapping oder komplexer Diff |
| **Mapping-konsistenz** | ja | ja | nutzerabhängig | ja | ? | ja | ja |
| **Einbeziehen neuer Konzepte** | nein | nein | nein | ja | nein | nein | ja |
| **Nutzer-Involvierung** | (semi-) automatisch | (semi-) automatisch | manuell | automatisch | (semi-) automatisch | (semi-) automatisch | (semi-) automatisch |
| **Semantische Mappings** | Äquivalenz | Äquivalenz | - | Äquivalenz | Äquivalenz | - | Äquivalenz, *more / less general* |
| **Evaluierung** | | | | | | | |
| **Schema-/ Ontologie-größe** | 16 -159 Elemente | 87 -265 Elemente | $\leq$ 28.000 Konzepte | $\leq$ 42.000 Konzepte | 15-20 Konzepte | $\leq$ 97.000 Konzepte | $\leq$ 319.000 Konzepte |
| **Schema-/ Ontologie-evolution** | manuelle Änderungen | manuelle Änderungen | Change Log | manuelle Änderungen | manuelle Änderungen | Ontologie-versionen | Ontologie-versionen |
| **Qualität** | nein | nein | nein | nein | nein | nein | ja |

Tabelle 2.1: Vergleich der eigenen Ansätze mit bestehenden Verfahren zur Adaptierung von Mappings.

## 2.3 Schema- und Ontologie-Matching

Methoden des Schema- und Ontologie-Matchings werden u.a. während der Adaptierung ontologiebasierter Mappings zur Erzeugung neuer Korrespondenzen benötigt. Im Folgenden werden zunächst grundlegende Verfahren des Schema- und Ontologie-Matchings vorgestellt. Dabei werden komplexe Match-Strategien (*Match Workflows*), semantische Ontologiemappings und einige Evaluierungsverfahren vorgestellt. Im Anschluss werden Methoden zum skalierbaren Ontologie-Matching diskutiert. Diese umfassen Verfahren zur Reduktion des Suchraums und zum parallelen Matching sowie Ansätze basierend auf der Wiederverwendung und Komposition existierender Mappings. Weiterhin werden Match-Ansätze und -Systeme aus dem Bereich der Lebenswissenschaften diskutiert. Abschließend erfolgt die Abgrenzung der in dieser Arbeit vorgestellten Match-Verfahren von existierenden Ansätzen.

### 2.3.1 Verfahren des Schema- und Ontologie-Matchings

Schema- bzw. Ontologie-Matching-Verfahren dienen der Bestimmung von Korrespondenzen zwischen den Konzepten zweier Schemas (z.B. relationale Datenbank- oder XML-Schemas) bzw. Ontologien. Typischerweise wird ein Quellschema auf ein Zielschema abgebildet. Die Menge der identifizierten Korrespondenzen bildet ein Schema- bzw. Ontologiemapping (*engl. schema mapping, ontology mapping, ontology alignment*). Neben einem paarweisen Abgleich der Schemas / Ontologien (*engl. 2-way/pairwise matching*) existieren einige holistische Ansätze (*engl. n-way/holistic matching*), die mehrere Schemas ganzheitlich abgleichen, um z.B. ein gemeinsames integriertes Mediatorschema zu erstellen [85, 168]. In dieser Arbeit wird ausschließlich das paarweise Schema- und Ontologie-Matching betrachtet. Automatische Verfahren kommen zum Einsatz, um den manuellen Aufwand einer Mappingerstellung zu reduzieren. Insbesondere für sehr große Ontologien kann ein manueller Abgleich sehr zeitaufwendig, fehleranfällig und teilweise nicht realisierbar sein, weshalb eine Automatisierung sinnvoll ist. Automatische Verfahren allein genügen jedoch nicht, um Ontologiemappings von hoher Qualität zu erstellen. Die generierten Korrespondenzen müssen durch Experten verifiziert und vervollständigt werden, weshalb häufig der Begriff des semi-automatischen Matchings verwendet wird.

Der Bereich des Schema- und Ontologie-Matchings wurde in den letzten Jahren intensiv erforscht, so dass bereits zahlreiche Algorithmen und Prototypen existieren. Die Arbeiten [153, 160, 49] geben einen Überblick zur Thematik. Rahm und Bernstein [153] publizierten eine erste Klassifikation für Schema-Matching-Ansätze und unterscheiden grundsätzlich individuelle Ansätze und kombinierte Verfahren. Individuelle Match-Ansätze werden in metadaten- und instanzbasierte Verfahren unterteilt. Metadatenbasierte Verfahren nutzen Informationen des Schemas bzw. der Ontologie, wobei element (konzept)- und strukturbasierte Ansätze unterschie-

den werden. Elementbasierte Verfahren vergleichen Konzepte u. a. auf Basis der linguistischen Ähnlichkeit ihrer konzept-assoziierten Informationen wie beispielsweise Namen, Synonyme, Datentypen oder Beschreibungen. Dabei kommen verschiedene String-basierte Ähnlichkeits- oder Distanzfunktionen zum Einsatz. Beispielsweise basiert die Levenshtein-Distanz (auch Editierdistanz) [117, 70] auf der minimalen Anzahl notwendiger Editieroperationen, um einen String $S_1$ in einen anderen String $S_2$ zu überführen. Token-basierte Verfahren zerlegen die zu vergleichenden Strings $S_1$ und $S_2$ zunächst in Teil-Strings (Token). So bildet $n$-Gram für einen String alle Teil-Strings der Länge $n$ (z. B. Trigram = 3-Gram) [49]. Die Ähnlichkeit zweier Strings $S_1$ und $S_2$ wird anschließend anhand der Ähnlichkeit ihrer Tokenmengen $T_1$ und $T_2$ mithilfe typischer Mengenoperationen und Maße wie beispielsweise Jaccard ($\frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$) [92, 122] oder Dice ($2 \cdot \frac{|T_1 \cap T_2|}{|T_1| + |T_2|}$) [122, 27] berechnet. Strukturbasierte Verfahren bestimmen die Ähnlichkeit von Konzepten, indem beispielsweise Informationen über den lokalen Kontext eines Konzepts wie Kinder-, Eltern- oder Geschwisterkonzepte in das Matching einbezogen werden. Weiterhin nutzen Graph-Matching-Algorithmen, wie der iterative Fixpunkt-Algorithmus *Similarity Flooding* [132] die Ontologiehierarchie um Ähnlichkeitswerte auf benachbarte Konzepte zu propagieren. Im Gegensatz zu metadatenbasierten Ansätzen gleichen instanzbasierte Verfahren zwei Konzepte anhand der Ähnlichkeit ihrer zugeordneten Instanzen ab, wodurch wertvolle Hinweise zur Bedeutung der Schemaelemente gewonnen werden können [153]. Instanzbasierte Verfahren können u. a. ergänzend eingesetzt werden, falls die im Schema vorhandenen Informationen nicht ausreichen, um automatisch Korrespondenzen zu identifizieren. So können Kategorien verschiedener Produktkataloge anhand gemeinsam zugeordneter Produkte abgeglichen werden [175]. In den Lebenswissenschaften werden Annotationen genutzt, um Korrespondenzen zwischen Konzepten der GO-Subontologien anhand der Überlappung ihrer zugeordneten Gene zu bestimmen [104]. Um eine Ähnlichkeit zweier Konzepte anhand ihrer Instanzmengen zu bestimmen, kann ähnlich zu Tokenmengen z. B. das Dice-Maß eingesetzt werden.

Es existieren folglich viele verschiedene Match-Verfahren, welche mehr oder weniger gut für verschiedene Szenarien geeignet sein können. Oft kann ein einzelner Matcher kein zufriedenstellendes Ergebnis produzieren. Hingegen kann die Kombination mehrerer individueller Verfahren eine erhöhte Match-Qualität erzielen, da sich Verfahren gegenseitig ergänzen können. Der Übersichtsartikel [153] unterscheidet dazu hybride und zusammengesetzte (composite) Matcher. Ein hybrider Matcher integriert mehrere Match-Kriterien in einem Matcher. Beispielsweise berechnet COMAs *NamePath*-Matcher [41] die Ähnlichkeit zweier Elemente auf Basis der Ähnlichkeit ihrer „hierarchischen Namen“, so dass sowohl Elementnamen als auch die Schemastruktur einbezogen werden. Zunächst werden Elementnamen auf einem Pfad vom betrachteten Element zur Wurzel konkateniert, bevor die linguistische Ähnlichkeit der Namenspfade berechnet wird. Zusammengesetzte Matcher kombinieren hingegen die Ergebnisse verschiedener unabhängig ausgeführter Matcher und können flexibel aus einem Repertoire individueller Matcher zusammengesetzt werden.

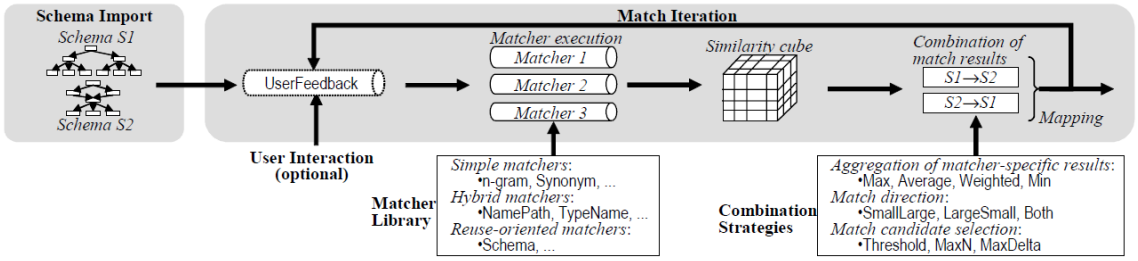



Abbildung 2.2: Match Workflow in COMA (Abbildung aus [41]).

### Match Workflows

Viele Match-Systeme unterstützen verschiedene Matcher und bieten somit die Möglichkeit, diese zu kombinieren. Abbildung 2.2 zeigt die typische Verarbeitung eines *Match Workflows* in COMA [41], die eine flexible Kombination mehrerer Matcher erlaubt. Zunächst werden zwei Eingabeschemas $S1$ und $S2$ konvertiert und in das interne Format (graphartige Struktur) überführt. Das anschließende Matching kann mehrere Iterationen umfassen, um das Mapping schrittweise zu verbessern. Jede Iteration umfasst einen optionalen Schritt zur Integration von Nutzerentscheidungen (*Feedback*), die Ausführung verschiedener Matcher sowie die Kombination der einzelnen Match-Ergebnisse. Geeignete Matcher können aus einer Matcher-Bibliothek (*Matcher Library*) ausgewählt werden. COMA bietet individuelle (*Simple*) und hybride Matcher (*Hybrid*) sowie Matcher basierend auf der Wiederverwendung existierender Mappings (*Reuse-oriented*). Jeder Matcher bestimmt ein Zwischenergebnis, bestehend aus einer Menge von Korrespondenzen zwischen den Eingabe-Schemas. Dabei ist jeder Korrespondenz ein Ähnlichkeitswert zwischen 0 und 1 zugeordnet. Das Ergebnis der Matcher-Ausführung für $k$ Matcher, $m$ $S1$-Elemente und $n$ $S2$-Elemente ist ein $k \times n \times m$-Würfel bestehend aus Ähnlichkeitswerten. Anschließend sollen die Ergebnisse der individuellen Matcher-Ausführung (Ähnlichkeitswerte im Würfel) zu einem Ergebnis kombiniert werden. Dazu werden zunächst für jede Korrespondenz die Matcher-spezifischen Ähnlichkeitswerte zu einem kombinierten Ähnlichkeitswert aggregiert. Dabei kann u. a. der Durchschnitt (*Average*) der Ähnlichkeitswerte oder auch der maximale / minimale (*Max* / *Min*) Ähnlichkeitswert über alle Matcher gebildet werden. Das Ergebnis der Aggregation ist eine $n \times m$-Matrix, die je einen kombinierten Ähnlichkeitswert für jedes mögliche Paar von Schemaelementen enthält. Anschließend wird eine Selektionsstrategie angewendet, um das Ergebnis zu filtern und die möglichst besten Korrespondenzen auszuwählen. Die einfachste Form der Selektion ist die Auswahl aller Korrespondenzen mit einem Ähnlichkeitswert oberhalb eines bestimmten Grenzwerts (*Threshold*). Alternativ können die jeweils $n$ besten Korrespondenzen für jedes $S1$-Element ausgewählt werden (*MaxN*). Beispielsweise würde die Anwendung von $Max1$ ($n = 1$) zu einem 1:1-Mapping führen, d. h. jedem $S1$-Element wird genau ein $S2$-Element zugeordnet (und umgekehrt). Die *MaxN*-Selektion mit $n > 1$ ist besonders sinnvoll, um Nutzer in der interaktiven Phase eine Korrespondenz für jedes Konzept aus mehreren Match-Kandidaten (1:n-Mapping)

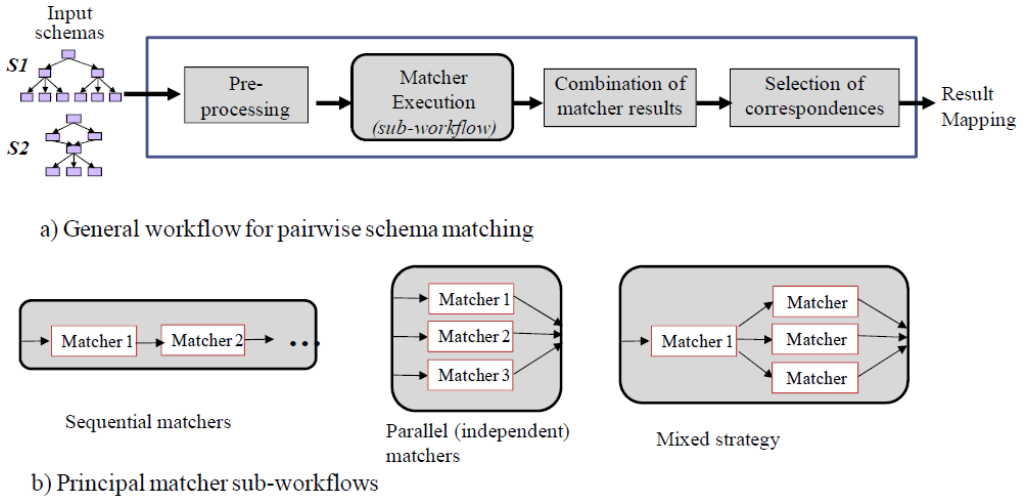



Abbildung 2.3: Match Workflows (Abbildung aus [152]).

auswählen zu lassen. Die *MaxDelta*-Selektion verfolgt ein ähnliches Ziel, wählt jedoch für jedes $S1$-Element die Korrespondenz mit dem besten (maximalen) Ähnlichkeitswert sowie alle Korrespondenzen zu diesem $S1$-Element mit einem Ähnlichkeitswert innerhalb eines gewissen Toleranzbereichs *delta*. Somit werden mehrere Match-Kandidaten ausgegeben, falls Korrespondenzen mit dem gleichen oder beinahe gleichen Ähnlichkeitswert für ein $S1$-Element vorhanden sind.

COMA ermöglicht also, ähnlich wie viele andere Match-Systeme, die Ausführung komplexer Match Workflows. Allgemein besteht ein Match Workflow aus verschiedenen Phasen (Abbildung 2.3a). Zunächst findet eine Vorverarbeitung (*Preprocessing*) der Ontologieelemente statt. Dazu zählen u. a. die Normalisierung von Strings (z. B. die Entfernung von Trennzeichen), die Analyse von Schema-Charakteristika oder die Tokenisierung und Indexierung von Konzeptattributen. Anschließend findet die Ausführung des eigentlichen Matchings statt, das typischerweise aus mehreren Matchern zusammengesetzt ist (Sub-Workflow). Verschiedene Matcher können sequenziell oder unabhängig voneinander ausgeführt werden (Abbildung 2.3b) [152, 49]. Bei der sequenziellen Ausführung bildet die Ausgabe eines Matchers die Eingabe des nächsten Matchers, d. h. die Matcher hängen voneinander ab. Im Gegensatz dazu sind parallele Matcher unabhängig voneinander. Des Weiteren können die Matcher in einer Mischform aus sequenziellen und parallelen Matchern kombiniert werden. Unter Verwendung der Zwischenergebnisse finden verschiedene Nachbearbeitungsschritte (*Postprocessing*) statt. Die Ergebnisse der einzelnen Matcher müssen aggregiert werden und verschiedene Selektionsstrategien dienen der Auswahl möglichst korrekter Korrespondenzen (siehe vorheriger Abschnitt zu COMA).

Neben typischen Selektionsstrategien können weitere fortgeschrittene Techniken z. B. zur semantischen Verifikation der Korrespondenzen zum Einsatz kommen. Das System ASMOV [94] prüft ein Mapping bezüglich verschiedener Arten von Inkonsisten-

zen und entfernt gegebenenfalls widersprüchliche Korrespondenzen. Beispielsweise sollen sogenannte *„CrissCross“*-Korrespondenzen vermieden werden. Zwei Korrespondenzen $(a1, b1)$ und $(a2, b2)$ zwischen den Konzepten zweier unterschiedlicher Ontologien $O1$ und $O2$ $(a1, a2 \in O1 \wedge b1, b2 \in O2)$ sind nicht konsistent zueinander (nicht verifizierbar), falls $a2$ ein Kind von $a1$ ist $(a2 < a1)$, $b1$ jedoch ein Kind von $b2$ ist $(b1 < b2)$ (oder umgekehrt). ASMOV sammelt nicht verifizierte Korrespondenzen in einer Liste, prüft Gründe für eine Eliminierung und wiederholt die semantische Verifikation gegebenenfalls in weiteren Iterationen. Darüber hinaus existieren zahlreiche weitere Arbeiten, die das Ontologiemapping im Zusammenhang mit den Eingabeontologien unter Verwendung von *Reasoning*-Methoden auf Inkonsistenzen untersuchen und diese eliminieren bzw. reparieren (z. B. [130, 181, 98]).

### Semantische Ontologiemappings

Die meisten Match-Systeme bestimmen Äquivalenzmappings, d. h. sie identifizieren ausschließlich Gleichheitsbeziehungen zwischen den Konzepten zweier Ontologien ($a = b$, $a \in O1$, $b \in O2$). Jedoch existieren auch andere Beziehungstypen wie *is-a* oder *part-of*. Diese sind nützlich, wenn die zu vergleichenden Ontologien ein unterschiedliches Level an Detailinformationen besitzen. Wenn die Quellontologie sehr detaillierte Informationen enthält, die Zielontologie hingegen die Thematik eher generisch beschreibt, existieren tendenziell weniger Äquivalenzbeziehungen zwischen beiden Ontologien. Da die Konzepte der Zielontologie allgemeiner sind als jene der Quellontologie, ist es sinnvoll, ein semantisch reichhaltigeres Mapping mit weiteren Beziehungstypen wie z. B. *is-a* oder *part-of* zu bestimmen. Darüber hinaus sind semantisch ausdrucksstarke Mappings hilfreich für Applikationen wie beispielsweise das Zusammenführen (Merging) von Ontologien [157, 156]. Nur wenige Ansätze verfolgen neben der Bestimmung von Äquivalenzbeziehungen die Bestimmung von *is-a*- oder *part-of*-Beziehungen (z. B. [165, 58, 59, 72]). Die verschiedenen Ansätze nutzen unterschiedliche Notationen bzw. Bezeichnungen wie z. B. *is-a*, *less general* oder Subsumptionsbeziehung. Die Autoren des semantischen Match-Systems S-Match [58, 59] bestimmen Äquivalenzbeziehungen (=), *less general* ($\sqsubseteq$), *more general* ($\sqsupseteq$), *Mismatch* bzw. Disjunktheits- ($\perp$) und *overlapping* / Schnittmengen ($\sqcap$)-Beziehungen. Zusätzlich ordnen sie die Beziehungen entsprechend ihrer Bindungsstärke vom stärksten zum schwächsten Beziehungstyp: $=$, $\sqsubseteq$, $\sqsupseteq$, $\perp$, $\sqcap$, wobei $\sqsubseteq$ und $\sqsupseteq$ die gleiche Bindungsstärke haben. Der Ansatz in [6] verfolgt eine semantische Anreicherung von Äquivalenzmappings, die zuvor durch ein Standard-Match-System bestimmt wurden. Dabei werden neben Gleichheitsbeziehungen auch *is-a*- und *part-of*-Beziehungen bestimmt. Die vorgestellte Methode kann flexibel auf existierende Ontologiemappings angewendet werden. Dadurch entsteht ein semantisch ausdrucksstarkes Mapping, das für weitere Applikationen nützlich ist. Semantische Mappings sollen in dieser Arbeit im Kontext der Mappingadaptierung berücksichtigt werden, insbesondere wenn komplexe Änderungen wie *split* oder *merge* auftreten.

### Evaluierung

Bellahsene et al. [12] geben einen aktuellen Überblick zur Evaluierung von Schema- und Ontologie-Matching. Im Folgenden soll nur ein kurzer Überblick über die in der Arbeit verwendeten Maße und Kriterien gegeben werden. Im Bereich des Schema- und Ontologie-Matchings wird typischerweise die Qualität der Ergebnisse automatischer Match-Verfahren unter Verwendung der Maße *Precision*, *Recall* und *F-Measure* untersucht. Dazu muss ein korrektes Mapping (Goldstandard, Referenzmapping) zur Verfügung stehen. Durch Vergleich des korrekten und des automatisch bestimmten Mappings können richtig positive (*true positive, TP*), falsch positive (*false positive, FP*) und falsch negative (*false negative, FN*) Korrespondenzen bestimmt werden.

Die *Precision* (Genauigkeit) gibt den Anteil der korrekt identifizierten gegenüber allen identifizierten Korrespondenzen an:

$Precision = \frac{|TP|}{|TP|+|FP|}$

Der *Recall* (Abdeckung) gibt den Anteil der korrekt identifizierten Korrespondenzen gegenüber allen Korrespondenzen des Referenzmappings an:

$Recall = \frac{|TP|}{|TP|+|FN|}$

Das F-Measure bildet das harmonische Mittel aus Precision und Recall:

$F - Measure = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$

Weiterhin können Aspekte wie die Laufzeit oder der Speicherverbrauch von Match-Verfahren untersucht werden. Geringe Laufzeiten sind u. a. wichtig in interaktiven Anwendungen, wenn Mapping-Designer bei der manuellen Mappingerstellung unterstützt werden sollen [152, 12]. Zudem können Reasoning-Methoden eingesetzt werden, um zu testen, ob ein Mapping fehlerfrei ist, d. h. keine Inkonsistenzen oder Widersprüche enthält. Dazu kann eine Ontologie aus den zwei Ausgangsontologien und dem dazwischen bestehenden Mapping zusammengefasst und bezüglich der Erfüllbarkeit (*Satisfiability*) der Konzepte überprüft werden. Zur Evaluierung kann dann ein Ratio aus dem Anteil nicht erfüllbarer Konzepte im Verhältnis zur Größe der zusammengefassten Ontologie (*Unsatisfiability Measure*) [129] dienen.

Die Organisatoren der *Ontology Alignment Evaluation Initiative* (OAEI)[10] führen seit 2004 jährlich vergleichende Evaluierungen von Systemen zum Ontologie- und Schema-Matching durch. OAEI bietet dazu verschiedene Benchmarks aus unterschiedlichen Domänen wie Sozialwissenschaften (*Library Track*), Konferenzorganisation (*Conference Track*) oder Anatomie (*Anatomy Track*). Ziel ist es, Stärken und Schwächen der einzelnen Systeme u. a. bezüglich der Qualität der erzielten Match-Ergebnisse und der Laufzeit der Systeme [48] aufzuzeigen. Darüber hinaus kommen z. B. innerhalb des *Large BioMed Tracks* spezielle Maße wie das *Unsatisfiability Measure*[11] zum Einsatz.

[10] `http://oaei.ontologymatching.org`

[11] `http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2012/results2012`

### 2.3.2 Skalierbares Ontologie-Matching

Trotz der umfangreichen Forschung auf dem Gebiet des Schema- und Ontologie-Matchings, haben viele Systeme Probleme, sehr große Ontologien zu verarbeiten. Insbesondere in den Lebenswissenschaften existieren Ontologien, welche mehrere zehntausende Konzepte (z. B. GO, NCIT, FMA) oder gar mehrere hunderttausende Konzepte (SCT) umfassen. Ziel automatischer Match-Verfahren ist es, Mappings hoher Qualität (Effektivität) in einer vertretbaren Laufzeit (Effizienz) zu bestimmen. Jedoch wird das Finden korrekter Korrespondenzen durch den enorm großen Suchraum (quadratisch bezüglich der Größe der Ontologien) erschwert. Es ist also wichtig, dass Match-Systeme auch für sehr große Ontologien skalieren [152]. Im Rahmen der OAEI-Evaluierungen haben sich die Systeme in den letzten Jahren bezüglich der erreichten Qualität und Laufzeit verbessert [48]. Viele Systeme haben jedoch weiterhin Probleme, sehr große Match-Aufgaben wie den 2012 eingeführten *Large BioMed Track* zum Abgleich sehr großer biomedizinischer Ontologien, zu absolvieren [3]. Häufig halten Match-Systeme Zwischenergebnisse (Würfel der Ähnlichkeitswerte) vollständig im Hauptspeicher, was beim Matching sehr großer Ontologien zum Abbruch des Match-Prozesses oder zu einer deutlichen Erhöhung der Ausführungszeit (z. B. durch *Swapping*) führen kann. Somit ist die Verbesserung der Performanz trotz der bisherigen Fortschritte ein wichtiges Ziel im Bereich des Schema- und Ontologie-Matchings [152, 145].

Für ein skalierbares Matching ist es sinnvoll, die benötigten Hauptspeicher- und CPU-Anforderungen gering zu halten [152]. Linguistische Match-Techniken können effiziente Verfahren zur Berechnung von String-Ähnlichkeiten einsetzen. So kann eine Tokenisierung und Indexierung von Konzeptattributen, die im Matching genutzt werden, als Vorverarbeitungsschritt ausgeführt werden (z. B. [109, 97]). Strukturelle Match-Verfahren können u. a. durch Vorberechnung der Konzeptvorgänger, -nachfolger oder -pfade optimiert werden [5], so dass nicht für jede strukturelle Anfrage erneut die Hierarchie traversiert und die gesamte Ontologiestruktur im Hauptspeicher gehalten werden muss. Eine strukturelle Indexierung kann durch ein Verfahren wie die *range compression* [1] erreicht werden. Dabei werden Konzepte nummeriert und zum Wertebereich (Intervall) ihrer Nachfolgerkonzepte assoziiert. Neben derartigen Optimierungen diskutiert Rahm in [152] vier aussichtsreiche Richtungen für das skalierbare Matching großer Ontologien (*large-scale matching*):

- Reduktion des Suchraums (durch partitionsbasiertes Matching und frühes Verwerfen unähnlicher Konzeptpaare)
- Paralleles Matching
- Selbst-konfigurierende Match Workflows
- Wiederverwendung existierender Match-Ergebnisse

Im Folgenden werden insbesondere relevante Ansätze zur Reduktion des Suchraums, zum parallelen Matching und zur Wiederverwendung von Mappings diskutiert.

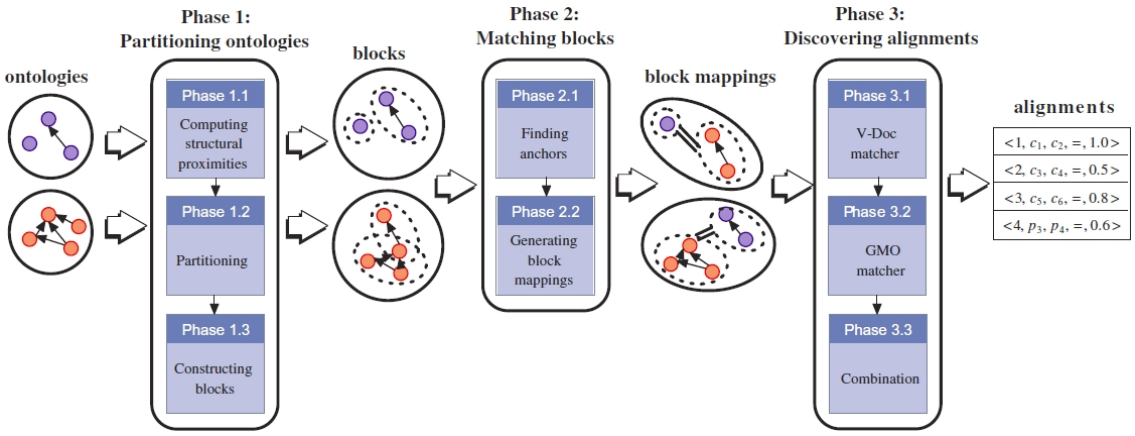



Abbildung 2.4: Partitionierung und Matching in Falcon (aus [88]).

### Verfahren zur Reduktion des Suchraums

Grundsätzlich hat das Matching von Ontologien eine quadratische Komplexität, da alle Konzepte der Quellontologie mit allen Konzepten der Zielontologie verglichen werden (kartesisches Produkt bezüglich Ontologiegröße). Typischerweise überlappen nur bestimmte Regionen zweier Ontologien, so dass ein Großteil der Vergleiche nicht zu korrekten Korrespondenzen führt. Daher ist es sinnvoll, zunächst Partitionen zu identifizieren, die aller Wahrscheinlichkeit nach überlappende Informationen enthalten. Partitionsbasierte Ansätze zerlegen das Match-Problem in Teilprobleme, indem die Eingabeontologien zunächst partitioniert und anschließend partitionsweise verglichen werden. Um den Suchraum zu reduzieren, ist es sinnvoll, jede Partition der Quellontologie nur mit einer / wenigen Partitionen der Zielontologie zu vergleichen. Dadurch kann sich die Effizienz des Matchings bezüglich der Laufzeit und Speicheranforderungen signifikant verbessern. Zudem kann eine Verbesserung der Precision erreicht werden, wenn Vergleiche, die ohnehin zu falsch positiven Korrespondenzen führen würden, von vornherein vermieden werden.

Einer der ersten Ansätze zur *Partitionierung* ist das fragmentbasierte Schema-Matching in COMA++ [8, 42]. In der ersten Phase werden Schemafragmente (Sub-Schemas) manuell bestimmt. Anschließend werden ähnliche Fragmente durch ein "leichtgewichtiges" Matching, z. B. durch Abgleich der jeweiligen Wurzeln der Sub-Schemas, identifiziert. Jedes der ähnlichen Fragmentpaare wird unabhängig voneinander abgeglichen und entstandene Teilmappings werden zum finalen Mapping zusammengeführt.

Das Ontologie-Matching-System Falcon [87] verfolgt ebenfalls einen partitionsbasierten Ansatz [88] (siehe Abbildung 2.4). Zunächst werden die Ontologien mithilfe einer strukturellen Clustering-Methode in Blöcke (*blocks*) zerlegt. Diese Blöcke werden anschließend anhand sogenannter Anker (*anchors*) abgeglichen, um ähnliche Blockpaare (*block mappings*) zu identifizieren. Anker sind Korrespondenzen zwischen sehr

ähnlichen Elementen, die anhand einer linguistischen Ähnlichkeit unter Verwendung von Elementnamen und -kommentaren durch Abgleich der vollständigen Ontologien vorberechnet werden. Später werden nur ähnliche Blöcke, die einen gemeinsamen Anker haben, unter Verwendung einer komplexeren Match-Strategie verglichen. Die Autoren geben an, dass die Vorberechnung der Anker circa die Hälfte der Ausführungszeit ihres Algorithmus benötigt. Das System TaxoMap verfolgt einen ähnlichen Ansatz, partitioniert jedoch zunächst eine Ontologie und versucht anschließend entsprechende Partitionen in der zweiten Ontologie zu identifizieren [72, 71]. Wenn eine der beiden Ontologien wesentlich größer und ausdrucksstärker ist als die andere, bestimmen Zhong et al. [190] die Subontologie (Partition) in der größeren Ontologie, die am ähnlichsten zu der kleineren Ontologie ist. Dazu werden Ähnlichkeitswerte unter Verwendung eines Name-Matcher sowie eines WordNet[12]-basierten Matchers für das kartesische Produkt bezüglich der Ontologiegrößen berechnet. Unter Auswertung der Nachbarkonzepte ähnlicher Konzepte werden zusammenhängende Subgraphen identifiziert und zu einer Subontologie zusammengefügt. Diese wird anschließend mit der zweiten Ontologie unter Verwendung einer aufwendigeren Methode wie z. B. *Similarity Flooding* [132] abgeglichen.

Der Anchor-Flood-Algorithmus [159] nutzt Anker als Startpunkt, um schrittweise Konzepte in deren struktureller Umgebung (Nachbarschaft) abzugleichen. Anker werden dabei als gegeben vorausgesetzt, d. h. deren Bestimmung ist nicht Teil des Algorithmus. Das Matching neu hinzukommender Korrespondenzen wird solange ausgeführt, bis keine Korrespondenzen mehr identifiziert werden oder alle Konzepte betrachtet wurden. Die Ausgabe des Algorithmus ist eine Menge von Mappings zwischen Teilgraphen der Ontologien, sogenannten Segmenten (Partitionen). Das System LogMap [97] berechnet zunächst effizient exakte Anker-Korrespondenzen mithilfe zuvor erstellter lexikaler und struktureller Indizes. Unter Nutzung der Anker, des strukturellen Index und Horn-Klauseln detektiert LogMap „nicht erfüllbare“ Klassen und führt direkt eine Mappingreparatur durch. Die Autoren geben an, dass die Methode gegebenenfalls unvollständig im Vergleich zu klassischen Reasoning-Methoden ist, jedoch eine gute Annäherung bietet und sehr effizient ist (linear bezüglich der Ontologiegröße). Anschließend werden in der strukturellen Umgebung der verbleibenden Anker neue Korrespondenzen unter Verwendung einer String-basierten Match-Methode berechnet. Die Mappingreparatur und das Matching werden iterativ fortgeführt bis alle Anker und neu hinzukommenden Korrespondenzen abgearbeitet wurden. LogMap vermeidet die Auswertung des kartesischen Produkts der Eingabeontologien und skaliert somit für sehr große Ontologien. Zudem bereinigt es logische Inkonsistenzen, wodurch die Precision verbessert werden kann.

Eine weitere Möglichkeit zur Reduktion des Suchraums ist die frühe Eliminierung sehr unähnlicher Konzeptpaare. Methoden mit diesem Ziel werden auch als „*Blocking*“-Methoden [11] bezeichnet und kommen insbesondere für das Matching von Instanzen bzw. Objekten (*engl. object matching, record linkage, entity reso-*

[12] WordNet: http://wordnet.princeton.edu/

*lution, duplicate detection*) zum Einsatz, da in diesem Bereich der Suchraum oft weit größer ist als für Ontologie-Matching-Probleme. Der „*Quick Ontology Mapping*“ (QOM)-Ansatz [47] verfolgt eine Reduktion des quadratische Suchraums, in dem nur ein Teil aller möglichen Konzeptpaare zweier Ontologien als Kandidaten für das eigentliche Matching ausgewählt werden. Dazu werden verschiedene Heuristiken wie die Auswahl von Konzepten anhand ähnlicher Namen (*label*) unter Verwendung einer sortierten Namensliste angewendet. Außerdem werden strukturelle Ähnlichkeiten der Konzepte ausgenutzt, um Konzeptpaare als Kandidaten zu identifizieren. Die Autoren zeigen, dass die Komplexität von QOM für Ontologien der Größe $n$ $O(n \cdot log(n))$ beträgt, anstelle der üblichen $O(n^2)$-Komplexität vieler Match-Algorithmen. Die empirische Evaluierung zeigt, dass geringe Verluste bezüglich der Qualität der berechneten Ontologiemappings in Kauf genommen werden müssen. Peukert et al. [149] schlagen eine regelbasierte Optimierungsstrategie für Match Workflows vor. Dabei dienen Filteroperatoren der Eliminierung unähnlicher Konzeptpaare aus Zwischenergebnissen in sequentiellen Match Workflows. Matcher arbeiten nur auf der Ausgabe eines zuvor ausgeführten Matchers (d. h. in einem reduzierten Suchraum). Zudem werden Match Workflows effizient umgeschrieben (*rewrite*), so dass parallele Workflows bestehend aus unabhängigen Match-Verfahren sequentialisiert werden, was den Einsatz der Filteroperatoren auch für parallele Match Workflows ermöglicht.

### Paralleles Matching

*Parallelisierung* ist eine weitere Möglichkeit, die Ausführung einer bestimmten Aufgabe zu beschleunigen. Eine parallele Datenverarbeitung hilft beispielsweise, die Antwortzeiten für Datenbankanfragen drastisch zu reduzieren [151]. Parallelisierung kann auch für die Beschleunigung von Ontologie-Matching-Aufgaben genutzt werden, allerdings existieren auf diesem Gebiet bisher nur wenige Arbeiten. Die Arbeit in [171] diskutiert erste Ideen zur parallelen Ausführung von Ontologie-Matching im Bereich *„Urban computing“* wie z. B. Verkehrsmanagement. Die Autoren schlagen die Zerlegung einer in der Domäne bevorzugten Ontologie in mehrere Teile vor, die parallel auf verschiedenen Rechenknoten mit mehreren anderen Ontologien abgeglichen werden, um die bevorzugte Ontologie zu erweitern. Eine Partitionierungsstrategie, das angewandte Match-Verfahren und eine Realisierung in Form einer Systemarchitektur oder Evaluierung werden nicht vorgestellt.

Mit V-Doc+ [185] wurde ein *MapReduce*-basierter Ansatz zum parallelen Ontologie-Matching mittels TF-IDF[13] vorgestellt. MapReduce ist ein generisches Framework für parallele Berechnungen auf mehreren Rechenknoten [38]. Der V-Doc+Algorithmus besteht aus drei Phasen, welche jeweils als MapReduce-Prozess realisiert sind: (1) Erstellung von Beschreibungen (Descriptions) für jede Entität (Klasse, Eigenschaft, Instanz), (2) Sammeln von Informationen der Nachbarentitäten und (3) Matching virtueller Dokumente mittels TF-IDF. Die beiden ersten Phasen dienen der Extrak-

[13] *engl. Term Frequency–Inverse Document Frequency*

tion der virtuellen Dokumente aus RDF. Dabei müssen u. a. Beschreibungen iterativ von Nachbarknoten aus RDF-Triplen extrahiert werden. Die teilweise notwendigen Rekursionen werden durch k-faches Wiederholen eines MapReduce-Prozesses realisiert. Um in der eigentlichen Matching-Phase unnötige Vergleiche einzusparen, wird ein Blocking-Schritt durchgeführt. Die Autoren nutzen eine Partitionierung anhand der auftretenden Worte in den virtuellen Dokumenten. Dabei werden die wichtigsten Worte anhand eines *scores* ausgewählt und für die Partitionierung genutzt (Map-Phase). Alle Entitäten zu einem Wort werden an einen Rechenknoten (Reducer) geschickt, der die Ähnlichkeitsberechnung zwischen allen zugeordneten Entitäten durchführt. Unterschiedliche Wortfrequenzen führen zu Lastbalancierungsproblemen, so dass die Autoren spezifische Reducer zur Verarbeitung der Tasks für die häufigsten Worte einsetzen. Die Evaluierung anhand zweier realer Match-Probleme zeigt, dass die Laufzeit verbessert werden kann und geringe Qualitätsverluste in Kauf genommen werden müssen.

#### Skalierbare wiederverwendungsbasierte Match-Verfahren

Ein weiterer Ansatz zur Verbesserung der Effizienz und Effektivität von Schema- und Ontologie-Matching-Ansätzen ist die *Wiederverwendung* (*reuse*) bereits existierender Mappings [153]. Die Wiederverwendung von Mappings wurde bereits im Zusammenhang mit der Evolution von Mappings diskutiert, um ein veraltetes Mapping auf neue Ontologieversionen zu migrieren (Kapitel 2.2). Zum Matching von Ontologien ist insbesondere die Komposition existierender Mappings zur indirekten Berechnung von Ontologiemappings von Interesse. Dazu werden Mappings von der Quell- und Zielontologie zu einer Zwischenontologie benötigt. Diese kann wertvolles Hintergrundwissen enthalten, wodurch neue Korrespondenzen im Vergleich zu einem direkten Matching gewonnen werden können. Neben einer verbesserten Match-Qualität kann eine Verbesserung der Laufzeit erreicht werden, da durch eine effiziente Kombination existierender Mappings die Evaluierung des kartesischen Produkt bezüglich der Ontologiegröße vermieden werden kann. Um Mappings wiederverwenden zu können, wird eine Infrastruktur zur Verwaltung zuvor bestimmter Mappings benötigt [152]. Im Bereich der Lebenswissenschaften existieren z. B. Repositories mit Mappingsammlungen wie BioPortal [141], das Mappings zwischen zahlreichen biomedizinischen Ontologien bereitstellt.

Im Rahmen des Model Management wurden bereits verschiedene Schema- und Mapping-Operatoren, wie `compose`, `extract`, `match`, `diff` oder `merge`, vorgeschlagen [131] (siehe 2.2). Der `compose`-Operator ($\circ$) bietet die Möglichkeit, zwei Mappings zu einem zu kombinieren und eignet sich daher für indirekte Match-Ansätze mit dem Ziel der Mapping-Wiederverwendung. Das Konzept des `compose`-Operators wurde im Bereich des Schema-Matchings zum indirekten Matching von Schemas auf Basis der Wiederverwendung von Mappings angewendet (*Reuse-Matcher* in COMA und COMA++) [8, 41]. Der sogenannte *MatchCompose*-Operator führt eine effiziente Kombination von zwei oder mehr Schemamappings durch. Beispielsweise kann ein

Mapping $M_{S1,S3}$ zwischen zwei Schemas $S1$ und $S3$ durch Kombination der bereits existierenden Mappings zu einem Schema $S2$ ($M_{S1,S2}$ und $M_{S2,S3}$) erhalten werden: $M_{S1,S3} = M_{S1,S2} \circ M_{S2,S3}$. Dabei wird die Transitivitätseigenschaft ausgenutzt: wenn eine Korrespondenz $(c, d)$ ($c \in S1$, $d \in S2$) und eine Korrespondenz $(d, e)$ ($e \in S3$) existieren, dann existiert auch eine Korrespondenz $(c, e)$. Die Eigenschaft kann insbesondere für Gleichheitsbeziehungen ausgenutzt werden. Zur Anwendung des Reuse-Matchers durchsucht COMA sein Repository nach allen anwendbaren Mappingpfaden, die beide Schemas verbinden. Die Evaluierung des Reuse-Matchers zeigte gute Ergebnisse bezüglich der Effizienz und Effektivität [41, 40].

Zur Lösung eines Match-Problems kann es zudem sinnvoll sein, einen Schemateil zu identifizieren, der noch nicht durch ein Mapping abgedeckt ist, um für diesen im weiteren Verlauf des Match Workflows gegebenenfalls Korrespondenzen zu identifizieren. In dem Model Management System Rondo [133] wurde der Operator `extract` eingeführt. Nach der Definition in [131] gibt `extract` den Teil eines Models bzw. Schemas zurück, das nicht an einem Mapping beteiligt ist: $\langle S1_e, M_{S1,S1_e} \rangle$ = `extract`$(S1, M_{S1,S2})$. Für ein Schema $S1$ und dessen Mapping $M_{S1,S2}$ zu einem weiteren Schema $S2$ gibt der `extract`-Operator den bisher nicht in $M_{S1,S2}$ abgedeckten $S1$-Schemateil $S1_e$ sowie dessen Verknüpfung zum Originalschema $M_{S1,S1_e}$ zurück.

Das Korpus-basierte Match-Verfahren in [123] nutzt eine Wissensbasis bestehend aus Schemamappings, welche in früheren Match-Aufgaben erstellt wurden. Diese Wissensbasis dient als (domänenspezifischer) Korpus, der im Matching wiederverwendet werden soll. Zwei zu vergleichende Schemaelemente sind ähnlich, falls beide den gleichen Elementen im Korpus entsprechen. Die Autoren nutzen ein maschinelles Lernverfahren, um Korrespondenzen zwischen den zu vergleichenden Schemas und dem Korpus zu identifizieren. Um eine Entscheidung zu treffen, ob die Schemaelemente einem Korpuselement $c$ gleichen, wird für jedes $c$ ein Modell basierend auf verschiedenen Matchern gelernt. Der Ansatz erfordert einen relativ hohen Aufwand zum Lernen und Anwenden der Modelle, insbesondere wenn sehr große Schemas abgeglichen werden und ein großer Korpus zur Verfügung steht [152].

In [126] werden Top-Level-Ontologien, d. h. sehr generische Ontologien, als „semantische Brücken“ zum Ontologie-Matching verwendet. Die Autoren nutzen einen `compose`-Operator zur Kombination von Mappings zwischen der Top-Level-Ontologie und den zu vergleichenden Ontologien. Sehr große Top-Level-Ontologien wie z. B. die „Suggested Upper Merged Ontology“ (SUMO) führten zu einer Verbesserung der Precision und des Recalls für das Matching verschiedener Domänenontologien.

Es existieren nur wenige Arbeiten zum indirekten Ontologie-Matching bzw. zur Mappingkomposition im Bereich der Lebenswissenschaften. Die Arbeit in [187] schlägt die indirekte Erstellung eines Mappings zwischen zwei Ontologien $O1$ und $O2$ unter Verwendung zweier direkter Mappings ($M_{O1,R}$,$M_{O2,R}$) zu einer Referenzontologie $R$ vor. Insbesondere verwenden die Autoren Mappings zu FMA, um das

Mapping zwischen MA und dem Anatomieteil des NCIT zu bestimmen. Sie nutzen dabei die transitive Eigenschaft ähnlich zum `compose`-Operator: wenn für ein FMA-Konzept $C_F$ sowohl eine Korrespondenz zu einem MA-Konzept $(C_M, C_F)$ als auch zu einem NCIT-Konzept $(C_N, C_F)$ besteht, dann werden die beiden Konzepte $C_M$ und $C_N$ automatisch einander zugeordnet $(C_M, C_N)$. In der Studie konnte die Qualität bezüglich F-Measure nicht evaluiert werden, da kein Referenzmapping verfügbar war. Allerdings zeigen die Experimente, dass das indirekte Matching einen Großteil der Korrespondenzen des direkten Matchings identifizieren kann und zusätzlich einige weitere Korrespondenzen liefert. Tordai et al. [177] präsentieren eine empirische Analyse der Mappingkomposition unter Ausnutzung der Transitivitätseigenschaft von Äquivalenzmappings. Sie verwenden dazu u. a. eine Sammlung von Korrespondenzen aus BioPortal [141], ohne dabei verschiedene Mediatorontologien zu unterscheiden. Die Autoren analysieren den Beitrag einzelner Korrespondenzen zum finalen Mapping, indem sie den Einfluss von Konzeptnamen und -synonymen näher untersuchen. In der Evaluierung wurden aus ≈570.000 einzelnen Korrespondenzen ≈600.000 kombinierte Korrespondenzen erstellt, wobei die jeweiligen Quell- und Zielontologien nicht unterschieden wurden. Anhand einer Stichprobe wurde eine Precision der Ergebnisse von 92% ermittelt, wohingegen der Recall nicht untersucht wurde. Die Studie zeigt, dass durch Komposition einige neue Korrespondenzen gefunden werden können, die das direkte Matching nicht erzeugt.

### 2.3.3 Ontologie-Matching in den Lebenswissenschaften

In den vorherigen Abschnitten wurden bereits einige Systeme und Ansätze diskutiert, die Ontologien aus dem Bereich Lebenswissenschaften abgleichen. Im Folgenden werden dazu einige weitere, relevante Arbeiten vorgestellt.

Es existieren einige Studien, die das Matching von Anatomieontologien betrachten (z. B. [186, 134, 18, 188, 189]). Beispielsweise präsentiert die Arbeit in [189] den Vergleich zweier unterschiedlicher Ansätze zum Matching von FMA und GALEN. Beide Ansätze nutzen eine Kombination aus lexikalen und strukturellen Match-Techniken. Der erste Ansatz nutzt zusätzlich domänenspezifisches Hintergrundwissen, wohingegen der zweite Ansatz generisch ist und ebenso in anderen Domänen eingesetzt werden kann. Die Ergebnisse beider Ansätze überlappen in großen Teilen, jedoch identifizieren beide Ansätze korrekte Korrespondenzen, die der andere Ansatz nicht finden konnte. Somit könnten die beiden Methoden jeweils voneinander profitieren. Zudem haben viele Systeme am OAEI *Anatomy Track* teilgenommen oder das dort verwendete Anatomie-Referenzmapping zwischen MA und dem Anatomieteil des NCIT verwendet (z. B. SAMBO [111], ASMOV [94], Falcon [87], AgreementMaker [32], GOMMA [102], LogMap [97], YAM++ [137], ServOMap [9]).

Neben dem Matching von Anatomieontologien ist auch das Matching anderer biomedizinischer Ontologien wie z. B. GO von Interesse. In [17] verlinken die Autoren

die GO mit chemischen Entitäten in ChEBI unter Verwendung von Konzeptnamen und Synonymen. Der adaptive Ansatz in [155] bezieht Domänenexperten in den Match Workflow ein, um verschiedene Vokabulare zur funktionellen Beschreibung von Genen und Proteinen abzugleichen. Tan et al. [170] nutzen einen instanzbasierten Ansatz, um u. a. ein Mapping zwischen GO mit der Signal-Ontologie SigO [169] zu erstellen. Neben lexikalen und strukturellen Match-Techniken kommt ein instanzbasiertes Verfahren unter Ausnutzung biomedizinischer Literaturquellen zum Einsatz. Dabei wird eine Korrespondenz zwischen zwei Konzepten erstellt, falls diese eine gewisse Überlappung bezüglich der ihnen zugeordneten wissenschaftlichen Artikel aufweisen. Ein weiterer instanzbasierter Ansatz nutzt u. a. die Zuordnung von Proteinen zu Konzepten in GO und OMIM [73] aus, um Korrespondenzen zwischen biologischen Prozessen und genetischen Krankheiten zu identifizieren. Die LOOM-Methode [56] gleicht normalisierte Namen und Synonyme der Konzepte ab und unterstellt beinahe exakte String-Ähnlichkeit. Die Ergebnisse zeigen eine sehr gute Precision, jedoch liegt der Recall bei 60%, d. h. es werden eher unvollständige Mappings produziert. Aus den Ergebnissen schlussfolgern die Autoren, dass bereits einfache lexikale Match-Verfahren ausreichen, um Mappings guter Qualität zwischen biomedizinischen Ontologien zu berechnen.

Die Wiederverwendung von Mappings auf Basis der Komposition zeigte bereits eine Möglichkeit, Hintergrundwissen während des Matchings auszunutzen. Viele Systeme beziehen einfache Synonym-Wörterbücher oder eine lexikale Wissensbasis wie WordNet in das Matching ein (z. B. [59, 8, 190, 119]). Das Match-System SAMBO [111] verwendet Domänenwissen aus dem UMLS-Metathesaurus, um das Matching biomedizinischer Ontologien zu unterstützen. SAMBO nutzt dabei die Ähnlichkeit von Konzepten der Quell- und Zielontologie in UMLS aus. Für die Experimente wurde der UMLS-Metathesaurus während des Matchings angefragt. Als Ergebnis erhält SAMBO UMLS-Konzepte, welche den gesuchten Term eines Quell- oder Zielkonzepts enthalten. Den Konzepten wird ein Ähnlichkeitswert auf Basis des Grads der Übereinstimmung der Anfrageergebnisse zugewiesen.

Die Arbeit in [46] nutzt Wissen aus zuvor manuell erstellten Mappings für neue Ontologie-Matching-Aufgaben. Die Autoren gruppieren Konzepte der Quellontologie, die mit dem gleichen Konzept der Zielontologie verknüpft sind (n:1-Mappings). Derartige Cluster werden für mehrere Zielontologien erstellt und verglichen. Falls ein Cluster stabil ist, wird diese Information für das Matching zu einer neuen Zielontologie wiederverwendet. Eine Fallstudie [4] erstellt Mappings zwischen flachen Vokabularen zu einer semantisch reichhaltigeren Ontologie, die Hintergrundwissen für das Matching liefert.

Das Match-System AgreementMaker [32] wurde für die OAEI 2011-Teilnahme um die Verwendung einer Mediatorontologie während des Matchings erweitert [33]. Durch Ausnutzen der Uberon-Ontologie konnte AgreementMaker verbesserte Ergebnisse bezüglich der Qualität im *Anatomy task* im Vergleich zum Vorjahr erreichen. Als Erweiterung des Ansatzes wird in [148] ein Thesaurus aus Synonymen der

zu vergleichenden Ontologien sowie weiterer externer Ontologien aufgebaut. Dieser Thesaurus wird dann im anschließenden lexikalen Matching genutzt. Eine weitere Arbeit [31] setzte ebenfalls Uberon als Referenzontologie für das Matching der Anatomieontologien des OAEI ein.

### 2.3.4 Zusammenfassung und Abgrenzung der eigenen Arbeit

Es existieren bereits zahlreiche Arbeiten zum (semi-) automatischem Schema- und Ontologie-Matching. Verschiedene Evaluationen zeigten, dass zum einen lexikale / linguistische Match-Verfahren eine wichtige Rolle für die Qualität der Match-Ergebnisse spielen. Zum anderen ist eine Kombination verschiedener metadaten- und instanzbasierter Matcher in komplexeren Workflows vielversprechend. Zudem zeigte sich, dass der Einsatz domänenspezifischen Hintergrundwissens sowie die Wiederverwendung und Kombination bestehender Mappings die Match-Qualität und -Laufzeit verbessern können. Durch die enorme Größe einiger Ontologien sind insbesondere skalierbare Match-Verfahren von Interesse. Einige Arbeiten versuchen unnötige Vergleiche einzusparen, indem möglichst nur überlappende Ontologiebereiche (Partitionen) miteinander verglichen werden. Andere Ansätze streben eine frühe Eliminierung unähnlicher Konzeptpaare an, um den Suchraum zu verkleinern. Bisherige Evaluierungen im Rahmen der OAEI zeigten, dass viele der verfügbaren Systeme nicht in der Lage sind, besonders große Ontologien abzugleichen.

In dieser Arbeit wird das System GOMMA [102] zur automatischen Berechnung von Ontologiemappings eingesetzt. GOMMA wurde ursprünglich für die Verwaltung und Analyse biomedizinischer Ontologie- und Mappingversionen entwickelt. Im Rahmen dieser Dissertation wir die *Match*-Komponente von GOMMA um das kompositionsbasierte Matching zur Wiederverwendung existierender Mappings (Kapitel 9) und die Parallelisierung des Ontologie-Matchings durch Partitionierung der Eingabeontologien (Kapitel 10) erweitert. GOMMA's *Match*-Komponente wurde im Rahmen der OAEI 2012 evaluiert. Die Ergebnisse (Kapitel 11) zeigen, dass GOMMA im Vergleich zu anderen Systemen sehr gute Ergebnisse bezüglich der Effizienz und Qualität des Matchings insbesondere für Ontologien aus den Lebenswissenschaften erreicht.

Die Wiederverwendung und Komposition bereits existierender Mappings wurde bisher im Bereich des Schema-Matchings untersucht [41, 40]. Wenige Arbeiten haben ein indirektes Ontologie-Matching durch Kombination von Ontologiemappings realisiert [187, 177, 126]. Im Gegensatz zu bisherigen Ansätzen, untersucht diese Arbeit die Komposition von Mappings (Kapitel 9) über mehrere verschiedene Zwischenontologien (*intermediate ontologies*), die sich gegenseitig ergänzen können. Insbesondere zentrale Mediatorontologien (*engl. „hubs“*) im Bereich der Lebenswissenschaften liefern wertvolles Hintergrundwissen, wodurch eine Verbesserung bezüglich der Match-Qualität erreicht werden kann. Das indirekte Ontologie-Matching wird

zudem um eine neuartige *extendMatch*-Operation zur Vervollständigung des kombinierten Mappings erweitert. Im Anschluss an den hier vorgestellten kompositionsbasierten Ansatz wurden einige Papiere veröffentlicht [33, 148, 31], die ebenfalls Referenzontologien (z. B. UMLS, Uberon) zur Verbesserung der Match-Qualität für biomedizinische Ontologien einsetzen.

Existierende Fragmentierungs- bzw. Partitionierungsstrategien (z. B. [88, 159]) zur Verbesserung der Effizienz und Skalierbarkeit des Ontologie-Matchings bergen die Gefahr, dass korrekte Korrespondenzen eliminiert werden, wodurch sich insbesondere der Recall reduzieren kann. Hingegen bietet eine Parallelisierung des Matchings einen generischen, skalierbaren Ansatz, ohne die Match-Qualität zu verringern. Im Rahmen dieser Arbeit wird der erste Ansatz zum parallelen Ontologie-Matching unter Ausnutzung mehrerer Rechenknoten vorgestellt. Zum Zeitpunkt der in Kapitel 10 präsentierten Arbeit (2010) existierte kein System, das das parallele Matching von Ontologien realisierte. Erstmals werden verschiedene Strategien zur parallelen Ausführung unabhängiger Matcher (Inter-Matcher-Parallelisierung) sowie kleinerer Match-Teilaufgaben durch eine einfache größenbasierte Partitionierung der Eingabeontologien (Intra-Matcher-Parallelisierung) vorgestellt. Die Partitionierung ist für element- sowie verschiedene struktur- und instanzbasierte Match-Verfahren anwendbar und ermöglicht eine gute Lastbalancierung. 2012 wurde die verteilte Ausführung von Ontologie-Matching unter Verwendung von MapReduce realisiert [185]. Die Autoren diskutieren Probleme bezüglich der Lastbalancierung und erreichen insgesamt einen schlechteren Speedup als GOMMA.

Für GOMMA's OAEI-Teilnahme wurde das System zudem um eine einfache, aber hilfreiche *Blocking*- bzw. Partitionierungsstrategie insbesondere für „asymmetrische" Match-Probleme erweitert. Ziel ist die Isolation einer besonders relevanten Subontologie, die anstelle der vollständigen Ontologie abgeglichen wird. Nach einer Vorverarbeitung werden Konzeptnamen und Synonyme indexiert und dadurch effizient abgeglichen. Im Gegensatz dazu evaluieren existierende Ansätze (z. B. [190]) häufig das kartesische Produkt bezüglich der Ontologiegröße, bevor die eigentliche Partitionierungsstrategie eingesetzt wird.

# 3 Grundlagen

In diesem Kapitel werden zunächst die Modelle definiert und erläutert, die in den weiteren Kapiteln der Arbeit verwendet werden (Kapitel 3.1). Es werden die Modelle für Ontologien, Ontologiemappings und Annotationsmappings sowie deren Versionierung vorgestellt. Darüber hinaus wird das in der Arbeit genutzte System GOMMA mit seinen Komponenten zum Ontologie-Matching und zur Diff-Berechnung vorgestellt (Kapitel 3.2).

## 3.1 Modelle

### 3.1.1 Ontologiemodell

#### Ontologie

Eine Ontologie $O = (C, A, R)$ besteht aus einer Menge von Konzepten $C$, welche Attribute aus $A$ besitzen und durch Beziehungen aus $R$ miteinander verbunden sind. Ein Konzept $c \in C$ wird durch Attribute aus $A$ detaillierter beschrieben. Ein Attribut $a = (c, a_{name}, a_{value})$ aus $A$ eines Konzepts $c$ besitzt einen Namen $a_{name}$ und einen Attributwert $a_{value}$. Jedes Konzept wird durch ein *ID*-Attribut eindeutig definiert. In Ontologien im Bereich der Lebenswissenschaften wird das *ID*-Attribut häufig als *Accession Number* (*Accession*) bezeichnet. Daneben existieren weitere Attribute wie z. B. der bevorzugte Name (Label), weitere Synonyme, eine Definition oder eine nähere Beschreibung. Synonyme sind ein Beispiel für mehrwertige Attribute, d. h. ein Konzept kann mehrere Synonyme bzw. alternative Namen besitzen. Beispielsweise

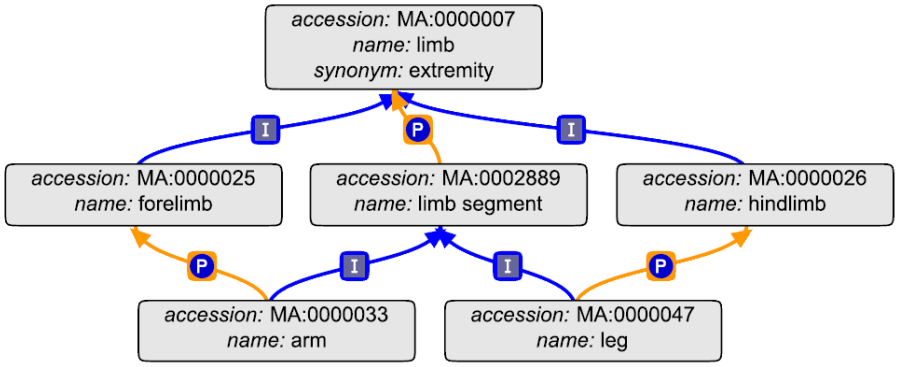



Abbildung 3.1: Ausschnitt der MA-Ontologie (generiert mit OBO-Edit), I = *is-a*-Beziehung, P = *part-of*-Beziehung.

könnte das Konzept MA:0000007 in Abbildung 3.1 ein weiteres Synonym wie '*fore / hindlimb*' aufweisen. In Ontologien im Bereich der Lebenswissenschaften wird häufig ein spezielles *obsolete*-Attribut genutzt, um anzugeben, ob ein Konzept ungültig bzw. veraltet ist und nicht mehr verwendet werden sollte.

Zwischen den Konzepten einer Ontologie bestehen Beziehungen. Eine gerichtete Beziehung $r = (c_s, type, c_t) \in R$ zwischen einem Quellkonzept $c_s$ und einem Zielkonzept $c_t$ hat einen Beziehungstyp *type*, der die Semantik der Beziehung spezifiziert. Der häufigste Beziehungstyp ist *is-a*, welcher eine Spezialisierung des Quellkonzepts gegenüber dem Zielkonzept darstellt, d. h. $c_s$ ist ein $c_t$. Weiterhin existieren zahlreiche andere Beziehungstypen wie z. B. *part-of* für Teil-Ganzes-Beziehungen. Typischerweise bilden Konzepte mit ihren *is-a* und *part-of* Beziehungen einen gerichteten, azyklischen Graphen (*engl.: directed acyclic graph*, DAG). Beziehungen anderen Typs (außer *is-a* und *part-of*) reichern die Ontologie semantisch an, bilden jedoch nicht zwingend einen DAG. Eine Ontologie kann beispielsweise inverse Beziehungstypen wie *inverse-is-a* oder *has-part* enthalten, so dass Zyklen entstehen. Eine Beispiel für eine domänenspezifische Beziehung ist die in GO verwendete *regulates*-Beziehung[14], welche die direkte Auswirkung eines Prozesses auf einen anderen Prozess darstellt. In einem DAG ist Mehrfachvererbung möglich, d. h. ein Konzept kann *is-a*-Beziehungen zu mehreren Eltern- bzw. Vorgängerkonzepten haben. Spezielle Wurzelkonzepte (*root(s)*) besitzen keine Vorgängerkonzepte und stellen die obersten Konzepte in der Ontologiehierarchie dar. Falls eine Ontologie mehrere Wurzelkonzepte besitzt, kann optional eine virtuelle Wurzel als deren einziges Vaterkonzept eingeführt werden. Weiterhin haben die Blattkonzepte (Blätter) einer Ontologie keine Nachfolger- bzw. Kinderkonzepte.

Entsprechend der eingeführten Notation kann beispielsweise die *Adult Mouse Anatomy* Ontologie mit dem Ausdruck $MA = (C, A, R)$ beschrieben werden. Das Konzept MA:0000001 ('*mouse anatomical entity*') bildet deren Wurzel (*root*). Die mithilfe von

[14] http://www.geneontology.org/GO.ontology.relations.shtml#regulates

OBO-Edit[15] [37] erzeugte Abbildung 3.1 zeigt einen Ausschnitt der MA. Beispielsweise umfasst die Menge der Attribute $A$ des Konzepts '*limb*' eine eindeutige ID (*accession*), einen Namen sowie ein zusätzliches Synonym:

- (MA:0000007, *accession*, MA:0000007),
- (MA:0000007, *name*, limb),
- (MA:0000007, *synonym*, extremity).

Beispiele für verschiedene Beziehungstypen zwischen Konzepten bilden die *is-a*-Beziehungen der Konzepte '*arm*' (MA:0000033) und '*leg*' (MA:0000047) zu '*limb segment*' (MA:0002889):

- (MA:0000033, *is-a*, MA:0002889) bzw.
- (MA:0000047, *is-a*, MA:0002889)

und deren zusätzliche *part-of*-Beziehungen zu:

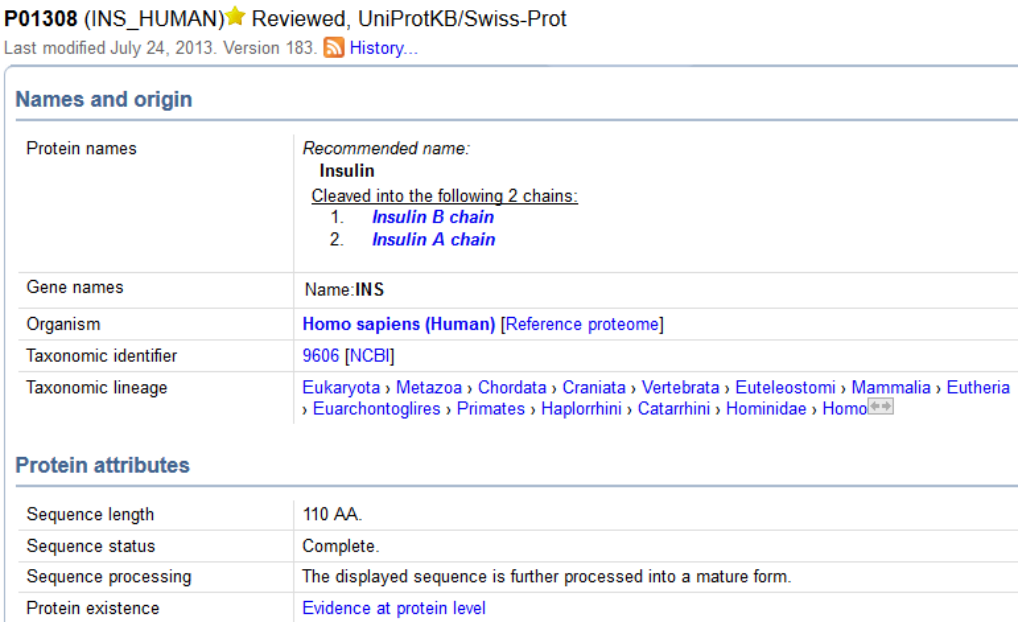
- '*forelimb*' (MA:0000033, *part-of*, MA:0000025) bzw.
- '*hindlimb*' (MA:0000047, *part-of*, MA:0000026).

### Ontologieversionen

Eine Ontologie kann in einer oder mehreren Versionen $v = 1, \ldots, n$ vorliegen: $O^v = (C^v, A^v, R^v, t)$. In dieser Arbeit wird ein lineares Versionierungsschema angenommen. Jede Ontologieversion $O^v$ ist ab dem Zeitpunkt $t$ ihrer Veröffentlichung bis zur Herausgabe einer neuen Version $O^{v'}$ zum Zeitpunkt $t'$ gültig. Alle Objekte $C^v$, $A^v$ und $R^v$ der Ontologieversion $O^v$ gelten ebenfalls ab dem Zeitpunkt $t$, bis eine neuere Version zum Zeitpunkt $t'$ veröffentlicht wird ($t < t'$). Jede Version $O^i$ hat also genau eine Vorgängerversion $O^{i-1}$ und genau eine Nachfolgeversion $O^{i+1}$, mit Ausnahme der ersten bzw. letzten Version, welche jeweils keine Vorgänger- bzw. Nachfolgeversion besitzen. Häufig werden neue Ontologieversionen in regelmäßigen Abständen veröffentlicht. Das GO-Konsortium veröffentlicht täglich eine GO-Version, wohingegen der NCIT monatlich aktualisiert wird.

Die Konzepte und Beziehungen des zuvor gezeigten Ontologieausschnitts waren in der Version $MA^{1.210}$ vom März 2013 gültig. Die Konzepte MA:0000007, MA:0000025, MA:0000026, MA:0000033 und MA:0000047 sowie deren Beziehungen untereinander existierten bereits seit der ersten Version vom August 2005 ($MA^{1.1}$). Hingegen wurde das Konzept '*limb segment*' (MA:0002889) sowie die *is-a*-Beziehungen von '*arm*' (MA:0000033) und '*leg*' (MA:0000047) zu MA:0002889 erst im September 2008 in Version $MA^{1.188}$ hinzugefügt.

[15] http://oboedit.org/



**P01308** (INS_HUMAN) Reviewed, UniProtKB/Swiss-Prot

Last modified July 24, 2013. Version 183. History...

**Names and origin**

| | |
|---|---|
| Protein names | *Recommended name:*<br>**Insulin**<br>Cleaved into the following 2 chains:<br>1. ***Insulin B chain***<br>2. ***Insulin A chain*** |
| Gene names | Name:**INS** |
| Organism | **Homo sapiens (Human)** [Reference proteome] |
| Taxonomic identifier | 9606 [NCBI] |
| Taxonomic lineage | Eukaryota › Metazoa › Chordata › Craniata › Vertebrata › Euteleostomi › Mammalia › Eutheria › Euarchontoglires › Primates › Haplorrhini › Catarrhini › Hominidae › Homo |

**Protein attributes**

| | |
|---|---|
| Sequence length | 110 AA. |
| Sequence status | Complete. |
| Sequence processing | The displayed sequence is further processed into a mature form. |
| Protein existence | Evidence at protein level |

Abbildung 3.2: UniProtKB/Swiss-Prot Eintrag für Insulin (Ausschnitt).

### 3.1.2 Instanzmodell

Neben Ontologien existieren Instanzquellen $IS = (I, IA)$, die aus einer Menge von Instanzen (Entitäten) $I$ und deren Instanzattributen $IA$ bestehen. Instanzen im Bereich der Lebenswissenschaften sind z. B. Gene, Proteine, Publikationen, Experimente oder Patientenakten, die typischerweise mit Ontologiekonzepten annotiert werden (siehe Kapitel 3.1.3). Eine Instanz $i \in I$ wird durch Attribute aus $IA$ näher beschrieben. Entsprechend des Ontologiemodells besitzt ein Instanzattribut $ia = (i, ia_{name}, ia_{value})$ der Instanz $i$ einen Attributnamen ($ia_{name}$) und -wert ($ia_{value}$). Innerhalb einer Quelle $IS$ besitzt jede Instanz ein eindeutiges ID-Attribut ($Accession$). Im Bereich der Lebenswissenschaften sind Instanzen häufig über alle Quellen hinweg eindeutig mit dieser $Accession$ identifizierbar, wodurch eine quellübergreifende Referenzierung der Instanzen ermöglicht wird. Weitere Attribute wie ein Name oder eine Beschreibung liefern zusätzliche Informationen. Beispielsweise hat der Eintrag *P01308* für humanes Insulin in *UniProtKB/Swiss-Prot* (Abbildung 3.2)[16] Protein- und Gennamen sowie weitere Attribute wie die Protein-Sequenzlänge.

Verfügbare Instanzquellen z. B. zur Verwaltung von Proteinen oder Genen (Swiss-Prot, Ensembl, ...) veröffentlichen regelmäßig neue Versionen. Die Version einer Instanzquelle $IS^w = (I^w, IA^w, t)$ ist ebenso wie eine Ontologiequelle ab dem Zeitpunkt $t$ gültig, bis eine neuere Version zum Zeitpunkt $t'$ veröffentlicht wird ($t < t'$). Dabei wird ebenfalls ein lineares Versionierungsschema angenommen.

[16] http://www.uniprot.org/uniprot/P01308

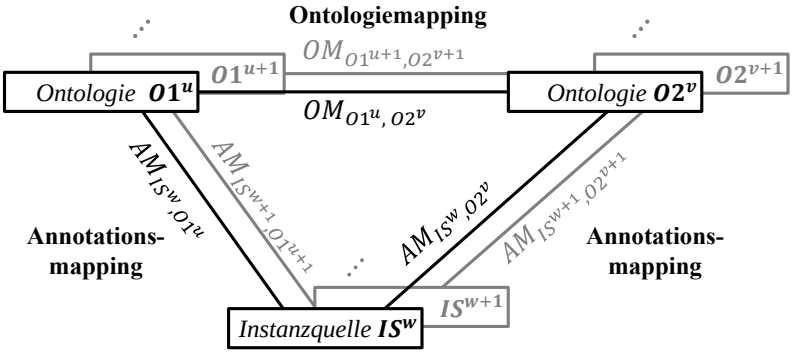



Abbildung 3.3: Mappingmodell.

### 3.1.3 Mappingmodell

Ein Mapping umfasst eine Menge von Korrespondenzen zwischen den Objekten zweier Datenquellen. Entsprechend der Art der Datenquellen werden verschiedene Mappings unterschieden. Im Rahmen dieser Arbeit werden Ontologiemappings (Mappings zwischen Ontologien) und ontologiebasierte Annotationsmappings (Mappings zwischen Instanzquellen und Ontologien) betrachtet. Weiterhin kann ein Diff-Evolutionsmappings zwischen verschiedenen Ontologieversionen existieren. Die drei Mappingarten werden in den folgenden Abschnitten genauer vorgestellt und definiert. Abbildung 3.3 gibt einen Überblick zu dem verwendeten Mappingmodell.

#### Annotationsmappings

Annotationen dienen der einheitlichen Beschreibung von Objekten der realen Welt. Im Rahmen dieser Arbeit werden ontologiebasierte Annotationen betrachtet. Diese spielen insbesondere in den Lebenswissenschaften eine wichtige Rolle. Dort werden u. a. biologische Objekte wie Gene und Proteine, Patientenakten oder biomedizinische Literaturquellen mit den Konzepten verschiedener Ontologien beschrieben. So dient die Annotation von Genen bzw. Proteinen mit Konzepten der *Gene Ontology* (GO) der Beschreibung ihrer molekularen Funktionen oder der biologischen Prozesse, in die sie involviert sind. Instanzquellen wie Swiss-Prot [21], GOA [10] oder Ensembl [91] stellen z. B. Annotationen für Gene und Proteine zur Verfügung.

Ein Annotationsmapping $AM$ umfasst eine Menge von Annotationen (bzw. Assoziationen / Korrespondenzen) zwischen den Instanzen $I$ einer Instanzquelle $IS$ und den Konzepten $C$ einer Ontologie $O$. Das Mappingmodell in Abbildung 3.3 zeigt zwei Ontologien $O1$ und $O2$ sowie eine Instanzquelle $IS$. Die Instanzquelle $IS$ ist jeweils durch ein Annotationsmapping mit $O1$ und $O2$ verknüpft. Da für Instanzquellen sowie für Ontologien typischerweise verschiedene Versionen veröffentlicht werden, existieren ebenso mehrere Versionen eines Annotationsmappings. Im Mappingmodell sind jeweils zwei Mappingversionen zwischen $IS$ und $O1$

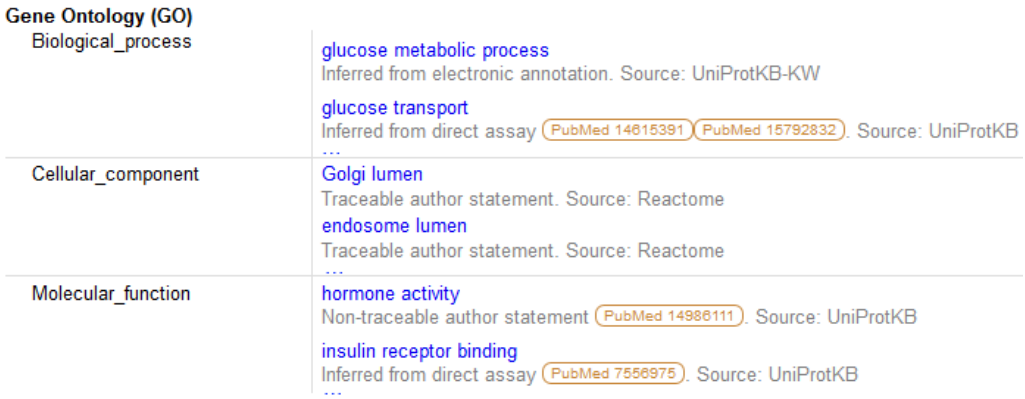



Abbildung 3.4: GO-Annotationen für Insulin in UniProtKB/Swiss-Prot (Ausschnitt)

($AM_{IS^w,O1^u}$ und $AM_{IS^{w+1},O1^{u+1}}$) sowie zwischen $IS$ und $O2$ abgebildet ($AM_{IS^w,O2^v}$ und $AM_{IS^{w+1},O2^{v+1}}$). Annotationsmappings werden wie folgt definiert:

$$AM_{IS^w,O^v} = \{(i,c)|i \in IS^w, c \in O^v\}$$

Dabei verknüpft eine einzelne Annotation $a = (i,c)$ eine Instanz $i \in IS^w$ einer Instanzquellversion und ein Konzept $c \in O^v$ einer Ontologieversion.

Annotationen können zusätzlich mit einer Menge von Merkmalen beschrieben werden. Beispielsweise können Informationen über die *Herkunft* einer Annotation, wie z. B. deren Erstellungsmethode, Aufschluss über die Glaubwürdigkeit der Annotation geben (siehe Kapitel 7). Abbildung 3.4 zeigt einen Teil der GO-Annotationen für Insulin in der Datenquelle UniProtKB/Swiss-Prot. Insulin ist an dem biologischen Prozess 'Glukosetransport' ('*glucose transport*') beteiligt und hat die molekulare Funktion 'Hormonaktivität' ('*hormone activity*'). Beide Annotationen sind jeweils mit zusätzlichen Attributen wie Literaturverweisen, ihrer Ursprungsquelle oder einem sogenannten *Evidence Code*[17] ('*Inferred from direct assay*', '*Non-traceble author statement*') versehen. Das GO-Konsortium fordert Kuratoren dazu auf, GO-Annotationen mit Evidence Codes zu versehen. Dadurch bekommen Nutzer Hinweise zur Herkunft der Annotation, d. h., auf Basis welcher Methode (z. B. Art von Experiment oder Analyse) die beschriebene Assoziation zwischen einem biologischen Objekt (z. B. Gen) und einem GO-Konzept nachgewiesen wurde.

### Ontologiemappings

Ein Ontologiemapping $OM_{O1,O2}$ verbindet die Konzepte zweier unterschiedlicher Ontologien $O1/O2$ durch sogenannte Korrespondenzen. Dabei wird die Quellontologie $O1$ auf die Zielontologie $O2$ abgebildet. Im Rahmen dieser Arbeit werden jedoch nicht nur Ontologiemappings zwischen je einer bestimmten Ontologieversion von $O1$ und $O2$ betrachtet. Da sich Ontologien verändern und regelmäßig neue Versionen

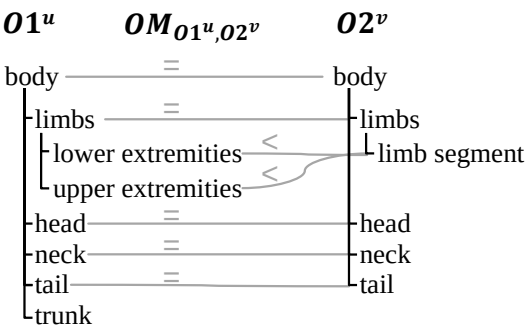
[17] http://www.geneontology.org/GO.evidence.shtml

veröffentlicht werden, existieren auch mehrere Versionen eines Ontologiemappings. Abbildung 3.3 zeigt zwei Versionen eines Ontologiemappings zwischen den Ontologieversionen $O1^u$ und $O2^v$ ($OM_{O1^u,O2^v}$) sowie deren Nachfolgeversionen $O1^{u+1}$ und $O2^{v+1}$ ($OM_{O1^{u+1},O2^{v+1}}$). Das Modell bezieht eine gleichzeitige Änderung der Quell- und Zielontologie ein, deckt aber ebenso den einfacheren Fall ab, wenn sich nur eine der beiden Ontologien ändert. Ontologiemappings werden wie folgt definiert:

$OM_{O1^u,O2^v} = \{(c_1, c_2, sim, m, semType, status) | c_1 \in O1^u, c_2 \in O2^v, sim \in [0,1],$
$m \in Method, semType \in \{=, >, <, \approx\}, status \in \{handled, toverify\}\}$

Eine Korrespondenz $corr = (c_1, c_2, sim, m, semType, status)$ verknüpft ein Konzept $c_1 \in O1^u$ einer Quellontologieversion und ein Konzept $c_2 \in O2^v$ einer Zielontologieversion. Der Grad der Ähnlichkeit *sim* gibt die Stärke der Verbindung zweier Konzepte an. Dabei zeigen Werte nahe oder gleich 1 (0) an, dass zwei Konzepte sehr ähnlich (unähnlich) sind. Der Ähnlichkeitswert wird mit einer bestimmten Methode $m$ (z. B. *manuell, Name-Matcher, ...*) bestimmt. Typischerweise wird manuell erstellten Korrespondenzen der Ähnlichkeitswert 1 zugeordnet. Da die manuelle Bestimmung von Ontologiemappings sehr ressourcenintensiv bzw. teilweise (aufgrund der Größe der Ontologien) nicht realisierbar ist, kommen häufig (semi-) automatische Match-Verfahren (siehe Kapitel 2.3) zum Einsatz. Automatische Verfahren ordnen jeder Korrespondenz einen *sim*-Wert von 0 bis 1 zu. Die Verbindung zwischen Konzepten kann verschiedene semantische Typen (*semType*) aufweisen. Häufig werden Gleichheitsbeziehungen zwischen Konzepten bestimmt ('=', *equals*). Jedoch existieren auch andere Beziehungstypen wie z. B. die *more general* ('>') oder *less general* ('<') Semantik, die jeweils angeben, ob das Quellkonzept $c_1$ allgemeiner oder weniger allgemein als das Zielkonzept $c_2$ ist. In diesem Modell werden *is-a*- und *part-of*-Beziehungen vorerst nicht unterschieden. Alternativ kann eine genauere und feingranularere Semantik gewählt werden. Falls der Beziehungstyp einer Korrespondenz nicht genau spezifiziert werden kann, wird die weniger strikte *is related*-Semantik ('≈') verwendet. Die Symbole '=', '<', '>' und '≈' und dienen als Abkürzung der betrachteten Beziehungstypen. Zusätzlich kann einer Korrespondenz ein Status zugeordnet werden. Dieser drückt aus, ob eine automatisch erstellte Korrespondenz bereits bestätigt ist bzw. geprüft wurde (*handled*) oder noch durch Experten verifiziert (*to verify*) werden muss. Die Korrespondenzattribute *semType* und *status* werden insbesondere im Kapitel 6 während der Adaptierung von Ontologiemappings genutzt, um den semantischen Korrespondenztyp und manuell zu überprüfende Korrespondenzen zu kennzeichnen.

Es soll auch die Inversion von Ontologiemappings unterstützt werden, um ein Mapping $OM_{O2^v,O1^u}$ aus $OM_{O1^u,O2^v}$ zu erhalten. Es wird ein `inverse`-Operator verwendet, der jede Korrespondenz des Mappings $OM_{O1^u,O2^v}$ wie folgt invertiert: $(c_1, c_2, sim, semType, status) \mapsto (c_2, c_1, sim, newSemType, status)$, wobei $c_1 \in O1^u$ und $c_2 \in O2^v$ ist. Dabei wird die Reihenfolge der korrespondierenden Konzepte $c_1$ und $c_2$ getauscht. Die Ähnlichkeits- (*sim*) und Statuswerte (*status*) bleiben unverändert. Der semantische Typ einer Korrespondenz (*semType*) wird entsprechend




**Korrespondenzen in $OM_{O1^u,O2^v}$**

(body, body, 1.0, *manual*, = , *handled*)
(limbs, limbs, 1.0, *manual*, = , *handled*)
(lower extremities, limb segment, 1.0, *manual*, < , *handled*)
(upper extremities, limb segment, 1.0, *manual*, < , *handled*)
(head, head, 1.0, *manual*, = , *handled*)
(neck, neck, 1.0, *manual*, = , *handled*)
(tail, tail, 1.0, *manual*, = , *handled*)

Abbildung 3.5: Beispiel eines Mappings zwischen zwei Ontologien $O1$ und $O2$.

der folgenden Regeln angepasst: $=\mapsto=$, $<\mapsto>$, $>\mapsto<$ und $\approx\mapsto\approx$. Der `inverse`-Operator wird während der Adaptierung von Ontologiemappings (Kapitel 6) und für das kompositionsbasierte Ontologie-Matching (Kapitel 9) benötigt.

Abbildung 3.5 zeigt beispielhaft ein Mapping $OM_{O1^u,O2^v}$ zwischen je einer Version zweier unterschiedlicher Ontologien $O1^u$ und $O2^v$. Das Mapping wurde manuell bestimmt, so dass alle Korrespondenzen eine Ähnlichkeit von 1, die Bestimmungsmethode $m$=*manual* und den Status *handled* aufweisen. Fünf Korrespondenzen stellen Gleichheitsbeziehungen (z. B. (*body*, *body*, ...), (*tail*, *tail*,...)) dar, so dass deren semantischer Typ '=' ist. Obere und untere Extremitäten ('*upper/lower extremities*') sind hingegen jeweils Gliedmaßensegmente ('*limb segment*'), so dass die *less-general*-Semantik zugewiesen wird. Im Rahmen dieser Arbeit wird die Semantik von Korrespondenzen einbezogen, da dies beispielsweise für die Adaptierung von Ontologiemappings eine wichtige Rolle spielt. Jedoch ist die automatische Bestimmung der Korrespondenzsemantik kein Forschungsziel dieser Arbeit. Die Berechnung von Korrespondenzen mit Gleichheitssemantik kann durch Verwendung eines Match-Systems wie z. B. GOMMA [102] (siehe Kapitel 3.2) erfolgen. Bereits existierende Mappings können dann entweder manuell oder durch Anwendung spezieller Verfahren semantisch angereichert werden (z. B. [58, 6]).

Neben Ontologiemappings zwischen verschiedenen Ontologien, können auch Ontologiemappings zwischen zwei unterschiedlichen Versionen der gleichen Ontologie (z. B. $OM_{O,O'}$) bestimmt werden (Evolutionsmapping). Ein Ontologiemapping zwischen verschiedenen Versionen einer Ontologie beinhaltet eine Menge semantischer Korrespondenzen zwischen den Konzepten der beiden Ontologieversionen, d. h. es erfasst die Übereinstimmungen der beiden Versionen. Alternativ kann ein Diff-Evolutionsmappings zur Erfassung der Unterschiede zwischen zwei Ontologieversionen bestimmt werden.

### Diff-Evolutionsmappings

Ein Diff-Evolutionsmapping $diff(O, O')$ beinhaltet eine Menge von Änderungsoperationen zwischen einer Ontologieversion $O$ und einer anderen Ontologieversion

$O'$. Im Gegensatz zu einem Ontologiemapping werden in einem Diff explizit Unterschiede in Form von Änderungen wie Hinzufügungen oder Löschungen anstelle von Korrespondenzen zwischen ähnlichen Konzepten erfasst. Sowohl Ontologie- als auch Diff-Evolutionsmappings zwischen Ontologieversionen werden u. a. verwendet, um ein veraltetes Ontologiemapping auf aktuelle Ontologieversionen zu migrieren.

Zur Bestimmung von Diff-Evolutionsmappings zwischen Ontologieversionen stehen bereits verschiedene Algorithmen zur Verfügung (siehe Kapitel 2.1). Es können einfache und komplexere Änderungsoperationen unterschieden werden. Einfache Änderungen beziehen sich typischerweise auf ein Konzept, ein Attribut oder eine Beziehung, welche hinzugefügt, gelöscht oder geändert werden können. Komplexere Änderungen beziehen sich hingegen auf mehrere Objekte, beispielsweise das Zusammenführen mehrerer Konzepte zu einem Konzept (*merge*) oder die Aufspaltung eines Konzepts in mehrere neue Konzepte (*split*). Angenommen Abbildung 3.5 zeigt ein Mapping zwischen zwei verschiedenen Versionen der gleichen Ontologie (also $OM_{O1^u,O1^w}$ statt $OM_{O1^u,O2^v}$), dann ist die Operation $delC(trunk)$ ein Beispiel für eine Konzeptlöschung (einfache Änderungsoperation). Ein Beispiel für eine komplexe Änderungsoperation bildet die Zusammenfassung der Konzepte '*lower extremities*' und '*upper extremities*' zu '*limb segment*': $merge(\{lower\ extremities, upper\ extremities\}, limb\ segment)$). Im Rahmen dieser Arbeit wird zur Bestimmung eines Diff-Evolutionsmappings GOMMA's *Diff*-Komponente COnto-Diff [75] verwendet, die im Kapitel 3.2.2 näher vorgestellt wird.

## 3.2 GOMMA

GOMMA (**G**eneric **O**ntology **M**atching and Mapping **Ma**nagement)[18] [102] bietet eine umfassende Infrastruktur zur Verwaltung und Analyse der Evolution von Ontologien und Mappings in den Lebenswissenschaften. Es nutzt ein generisches Repository, um Ontologieversionen und verschiedene Mappings einheitlich und effizient zu verwalten. Zudem bietet GOMMA verschiedene Funktionalitäten zum Matching von Ontologien sowie zur Bestimmung von Änderungen zwischen verschiedenen Versionen. Diese werden von Analysewerkzeugen wie beispielsweise OnEX (Ontology Evolution Explorer) [80] und REx (Region Evolution Explorer) [29, 76] genutzt.

Abbildung 3.6 gibt einen Überblick zu den verschiedenen Komponenten des Systems. GOMMA besteht aus drei Ebenen: einem Repository, funktionellen Komponenten sowie verschiedenen Applikationen. Das Repository ermöglicht die zentrale, einheitliche und effiziente Verwaltung von Ontologie- und Instanzquellversionen sowie Versionen von Ontologie- und Annotationsmappings. Das zugrunde liegende Versionierungsmodell basiert auf den vorgestellten Modellen für Ontologie- und Instanzversionen (Kapitel 3.1). GOMMA speichert die Elemente einer Datenquelle (Ontologiekonzepte, -attribute und -beziehungen bzw. Instanzen) einmalig ab [103].

[18] `http://dbs.uni-leipzig.de/GOMMA`

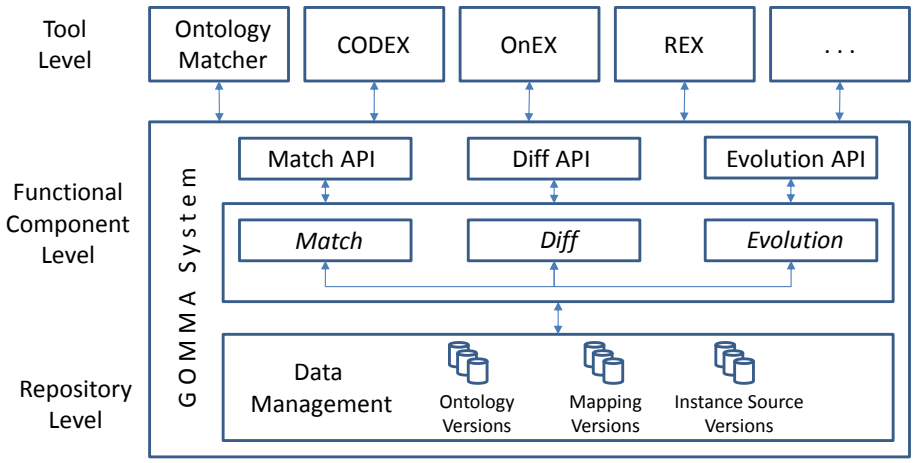



Abbildung 3.6: Überblick der komponentenbasierten GOMMA-Infrastruktur.

Dadurch wird vermieden, dass unveränderte Elemente redundant für jede neue Version abgelegt werden. Um anzugeben, in welchem Zeitraum ein Element gültig ist, wird ihm eine Lebenszeit in Form eines Start- ($t_{start}$) und Enddatums ($t_{end}$) zugeordnet. Da jede Quellversion mit einem Versionsdatum $t$ assoziiert ist, kann GOMMA den Stand einer Version durch Selektion der zum Zeitpunkt $t$ gültigen Elemente ($t_{start} \leq t \leq t_{end}$) rekonstruieren. GOMMA stellt verschiedene Importfunktionalitäten zur Verfügung, um Ontologien, Instanzen und Mappings zu integrieren. Beispielsweise werden typische Ontologieformate wie OBO und OWL [127] unterstützt. Außerdem kann auf öffentliche Ontologie-Archive wie beispielsweise das Gene Ontology CVS-Repository[19] zugegriffen werden. Mappings können aus verschiedenen CSV-, XML- oder RDF-Formaten importiert und in diese Formate exportiert werden.

Die in GOMMA verwalteten Ontologien, Instanzen und Mappings werden durch die drei funktionellen Komponenten *Match*, *Diff* und *Evolution* genutzt. Alle drei Komponenten verwenden eine zentrale Komponente, um auf die im GOMMA-Repository verwalteten Ontologie-, Instanz- und Mappingversionen zuzugreifen. Die *Match*-Komponente dient der automatischen Berechnung von Ontologiemappings, indem die semantische Ähnlichkeit zwischen Ontologiekonzepten bestimmt wird (siehe Kapitel 3.2.1). Die *Diff*-Komponente dient der Bestimmung von Diff-Evolutionsmappings zwischen Versionen von Ontologie- und Instanzquellen. Sie umfasst Funktionalitäten zur Bestimmung einfacher sowie komplexerer Änderungsoperationen (siehe COnto-Diff Kapitel 3.2.2). Die berechneten Ontologie- und Evolutionsmappings können im GOMMA-Repository abgespeichert werden. Die *Evolution*-Komponente ermöglicht die Evolutionsanalyse für Ontologie- sowie Instanzquellen, indem deren Änderungshistorie statistisch ausgewertet wird.

[19] http://geneontology.org/GO.downloads.ftp.cvs.shtml#cvs

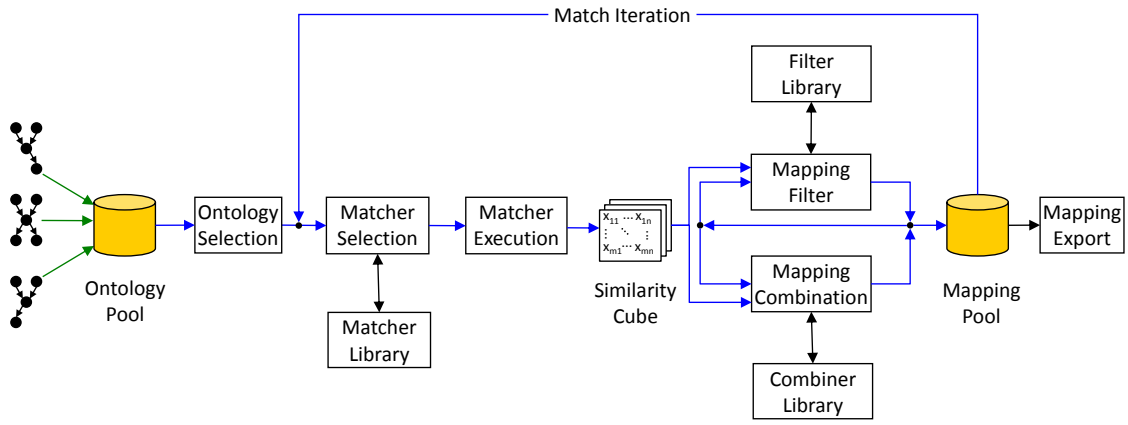



Abbildung 3.7: GOMMA Match Workflow.

Die obere Ebene der GOMMA-Infrastruktur besteht aus verschiedenen Applikationen, die auf die drei Kernfunktionalitäten *Match*, *Diff* und *Evolution* über komponentenspezifische Schnittstellen (APIs) zugreifen. *Ontology Matcher* nutzt hauptsächlich die *Match*-Komponente. CODEX [78] bestimmt hingegen Diff-Evolutionsmappings, bestehend aus einfachen und komplexen Änderungsoperationen durch Verwendung der *Diff*-Komponente. Die Applikationen OnEX (Ontology Evolution Explorer) [80] und REx (Region Evolution Explorer) [29, 76] nutzen die *Diff*- sowie die *Evolution*-Komponente. OnEX ermöglicht eine detaillierte Untersuchung von Änderungen in Ontologien in den Lebenswissenschaften. REx bietet hingegen die Möglichkeit, über längere Zeiträume die Stabilität verschiedener Ontologieregionen zu bestimmen und auszuwerten. REx nutzt einen Algorithmus [76] zur Bestimmung änderungsintensiver und stabiler Regionen innerhalb eines bestimmten Zeitintervalls.

### 3.2.1 Ontologie-Matching

GOMMA ermöglicht das Matching insbesondere großer Ontologien, wie sie in den Lebenswissenschaften vorkommen. Das System bietet eine umfassende Sammlung metadaten- und instanzbasierter Match-Verfahren (*Matcher Library*) sowie Verfahren zur Mappingkombination beziehungsweise -aggregation (*Combiner Library*) und Selektion (*Filter Library*). Abbildung 3.7 zeigt einen typischen GOMMA Match Workflow. Der Ansatz ist durch das Schema- und Ontologie-Matching-System COMA++ inspiriert [41], welches die kombinierte Ausführung mehrerer Match-Algorithmen unterstützt. Die abzugleichenden Ontologien werden zunächst in das zentrale Repository importiert. Abhängig von der Anwendung können die Ontologien auch importiert und direkt abgeglichen werden, ohne diese im Repository abzulegen. Anschließend werden verschiedene Matcher aus der *Matcher Library* ausgewählt und ausgeführt. Beispielsweise bestimmt der linguistische Name/Synonym-Matcher

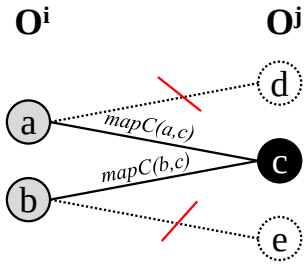
(*NameSyn*) die Trigram-Ähnlichkeit zweier Konzepte auf Basis ihrer Konzeptnamen und -synonyme. Der strukturbasierte *NamePath*-Matcher bezieht hingegen die Namen aller Elternkonzepte auf dem Pfad zur Wurzel in die Ähnlichkeitsberechnung ein. Der strukturbasierte *Context*-Matcher vergleicht den Kontext zweier Konzepte, in dem die Konzeptnamen sowie die Namen der Eltern- und Kinderkonzepte berücksichtigt werden. Ein instanz- (bzw. annotations-) basierter Matcher berechnet die Ähnlichkeit zweier Ontologiekonzepte unter Einbeziehung ihrer Instanzen. Der Matcher nutzt ein Annotationsmapping, um gemeinsame Instanzen zweier Konzepte zu bestimmen, und wendet eine Ähnlichkeitsfunktion wie Dice oder Jaccard zur Berechnung des *sim*-Werts an. Jeder einzelne Matcher generiert ein Zwischenergebnis, das jeweils eine Matrix der berechneten Ähnlichkeitswerte ($0 \leq sim \leq 1$) repräsentiert (*Similarity Cube*). Um den Speicherbedarf zu reduzieren können unnötige Korrespondenzen mit niedrigen Ähnlichkeitswerten frühzeitig verworfen werden, d. h. GOMMA muss nicht den gesamten *Similarity Cube* erhalten.

Die Zwischenergebnisse werden dann entsprechend einer Kombinationsstrategie zu einer Ergebnismatrix aggregiert. Dazu dienen übliche Mengenoperationen wie Vereinigung, Schnittmenge und Differenz, aber auch andere Ansätze wie beispielsweise ein Mehrheitsentscheid, der eine Korrespondenz akzeptiert, sofern diese durch die Mehrheit der angewendeten Matcher bestimmt wurde. Kombinationsoperatoren werden außerdem durch eine Aggregationsstrategie (z. B. Durchschnitt, Maximum, Minimum) konfiguriert, um für jede Korrespondenz einen kombinierten Ähnlichkeitswert aus den Ähnlichkeitswerten der einzelnen Matcher zu bestimmen. Weiterhin werden Filter- bzw. Selektionsstrategien ausgeführt, um die möglichst besten Korrespondenzen auszuwählen. Dazu zählen einfache Techniken wie *Threshold* sowie komplexere Strategien wie z. B. MaxN oder MaxDelta (siehe Kapitel 2.3.1). Zudem kann ein Match-Prozess in GOMMA gegebenenfalls iterativ ausgeführt werden, um ein Ergebnis schrittweise zu verfeinern. Das finale Mapping kann im GOMMA-Repository abgespeichert und/oder exportiert werden. Die automatisch generierten Korrespondenzen können durch Experten verifiziert und korrigiert werden. Der modulare Aufbau von GOMMA's *Match*-Komponente ermöglicht eine einfache Erweiterung des Systems um weitere Match-, Aggregations- und Selektionsstrategien.

Im Rahmen dieser Arbeit wird GOMMA's *Match*-Komponente um einen kompositionsbasierten Match-Ansatz (Kapitel 9) und die parallele Ausführung verschiedener unabhängiger Matcher (Kapitel 10) erweitert. Zudem wird die *Match*-Komponente in Teil II und IV zur Bestimmung von Ontologiemappings genutzt.

### 3.2.2 COnto-Diff

COnto-Diff [75] ermöglicht die Bestimmung eines semantisch ausdrucksstarken Diff-Evolutionsmappings zwischen Ontologieversionen und ist der wesentliche Bestandteil der *Diff*-Komponente in GOMMA. Ein Diff zwischen Ontologieversionen ist beispielsweise hilfreich, um die Weiterentwicklung von Ontologien effizient umset-




a,b∈O$^i$ ∧ c∈O$^j$ ∧ a≠b ∧ *mapC*(a,c) ∧ *mapC*(b,c)
∧ ∄ d(d∈O$^j$ ∧ *mapC*(a,d) ∧ c≠d)
∧ ∄ e(e∈O$^j$ ∧ *mapC*(b,e) ∧ c≠e)
→ **create**[*merge*({a},c), *merge*({b},c)],
**eliminate**[*mapC*(a,c), *mapC*(b,c)]

Abbildung 3.8: Beispiel einer *merge*-Regel in COnto-Diff.

zen zu können. Während der Ontologieentwicklung werden Änderungen teilweise gar nicht oder in Form einer einfachen Liste dokumentiert. Dies ist insbesondere für große Ontologien unübersichtlich und kaum nutzbar für menschliche Anwender wie beispielsweise Ontologieentwickler und -kuratoren. Mithilfe eines Diffs können Ontologieentwickler nachvollziehen, wie sich die Ontologie bisher verändert hat. In kollaborativen Projekten zur Entwicklung von Ontologien ist es u. a. wichtig zu wissen, welche Änderungen bereits umgesetzt wurden und welche Änderungen noch ausstehen. Darüber hinaus unterstützt ein Diff zwischen Ontologieversionen die (semi-) automatische Adaptierung abhängiger Daten wie Ontologie- und Annotationsmappings, wenn diese auf aktuelle Versionen migriert werden müssen.

Der COnto-Diff-Algorithmus bestimmt zunächst ein Ontologiemapping $OM_{O^i,O^j}$ zwischen der alten ($O^i$) und der neuen Ontologieversion ($O^j$, $i < j$), welches z. B. mithilfe der *Match*-Komponente von GOMMA erstellt werden kann. Auf Basis des Ontologiemappings und der beiden Versionen werden anschließend sogenannte Basis-Änderungsoperationen bestimmt. Diese beziehen sich auf ein einzelnes Konzept, eine Beziehung oder ein Attribut und umfassen einfache Modifikationen wie Hinzufügungen (*add*), Löschungen (*del*) und Änderungen bzw. Abbildungen (*map*). Die Menge der bestimmten Basis-Änderungsoperationen bilden ein Diff-Evolutionsmapping $diffBasic_{O^i,O^j}$. Um dieses einfache Diff-Evolutionsmapping semantisch anzureichern, wird ein regelbasierter Ansatz zur iterativen Aggregation der einfachen Änderungsoperationen in eine kleinere, kompaktere Menge von komplexen Änderungsoperationen verfolgt. Komplexe Änderungsoperationen sind z. B. die Aufspaltung eines Konzept in mehrere Konzepte (*split*), die Zusammenfassung mehrerer Konzepte zu einem Konzept (*merge*) oder die Hinzufügung bzw. Löschung größerer Subgraphen (*addSubGraph*, *delSubGraph*). Das Ergebnis aller Regelanwendungen ist ein kompaktes Diff-Evolutionsmapping $diffCompact_{O^i,O^j}$.

Abbildung 3.8 zeigt eine der notwendigen Regeln zur Erstellung einer komplexen Operation $merge(\{a,b\},c)$ ($a,b \in O^i$, $c \in O^j$). Die zwei einfachen Änderungsoperationen $mapC(a,c)$ und $mapC(b,c)$ stellen eine Abbildung der Quellkonzepte $a$ bzw. $b$ in der alten Version $O^i$ auf das Zielkonzept $c$ in der neuen Version $O^j$ dar. Wenn die Konzepte $a$ und $b$ an keiner weiteren *mapC*-Operation zu einem anderen Zielkonzept $d$ bzw. $e$ aus $O^j$ (wobei $c \neq d \wedge c \neq e$) beteiligt sind, werden zwei Opera-

| **Änderungsoperation** | **Beschreibung** |
|---|---|
| *addC(c), delC(c)* | Hinzufügung/Löschung eines Konzepts *c* |
| *toObsolete(c), revokeObsolete(c)* | Setzen/Aufheben des „veraltet-Status“ eines Konzepts *c* |
| *substitute(c, c‘)* | Ersetzen eines Konzepts *c* durch ein anderes Konzept *c‘* |
| *split(s, T)* | Aufspalten eines Quellkonzepts *s* in mehrere Zielkonzepte *T* |
| *merge(S, t)* | Zusammenführen mehrerer Quellkonzepte *S* in ein Zielkonzept *t* |
| *addSubGraph(c_root,C_Sub)* | Einfügen eines Subgraphen mit Wurzel *c_root* und den Konzepten *C_Sub* |
| *delSubGraph(c_root,C_Sub)* | Löschen eines Subgraphen mit Wurzel *c_root* und den Konzepten *C_Sub* |
| *addR(r), delR(r)* | Hinzufügung/Löschung einer Beziehung *r* |
| *move(c, P, P‘)* | Verschiebung eines Konzepts *c* von seinen Elternkonzepten *P* zu anderen Elternkonzepten *P‘* |
| *addA(a), delA(a)* | Hinzufügung/Löschung eines Attributs *a* |
| *chgAttValue(c, att, v1, v2)* | Ändern des Attributwertes *v1* von Attribut *att* des Konzepts *c* zu Wert *v2* |

Tabelle 3.1: COnto-Diff-Änderungsoperationen.

tionen $merge(\{a\}, c)$ und $merge(\{b\}, c)$ geschlussfolgert (**create**). Zudem werden die einfachen Änderungsoperationen $mapC(a, c)$ und $mapC(b, c)$ aus dem Evolutionsmapping entfernt (**eliminate**). Anhand einer weiteren Regel aggregiert COnto-Diff $merge(\{a\}, c)$ und $merge(\{b\}, c)$ zu $merge(\{a, b\}, c)$.

COnto-Diff ist modular aufgebaut und bietet daher einen hohen Grad an Flexibilität. Die Match-Phase ist unabhängig von der Bestimmung des Diffs, wodurch individuell geeignete, domänenspezifische Verfahren zur Bestimmung des Ontologiemappings angewandt werden können. Beispielsweise nutzt COnto-Diff derzeit für Ontologien im Bereich der Lebenswissenschaften die *Match*-Komponente von GOMMA, um einen exakten Abgleich auf Basis der Konzept-*Accessions* und den *NameSyn*-Matcher auszuführen. Darüber hinaus können die zur Diff-Bestimmung angewendeten Regeln leicht adaptiert und erweitert werden. Tabelle 3.1 zeigt die Menge der in dieser Arbeit verwendeten Änderungsoperationen. Dazu zählt beispielsweise die für die Lebenswissenschaften typische Änderungsoperation *toObsolete*. $toObsolete(c)$ gibt an, dass der Status eines Konzepts auf veraltet gesetzt wurde. Weiterhin werden Konzepthinzufügungen ($addC$), -löschungen ($delC$) und -ersetzungen ($substitute$) sowie komplexere Konzeptänderungen wie $split$, $merge$ und die Hinzufügung / Löschung ganzer Subgraphen einbezogen ($addSubGraph$ / $delSubGraph$). Änderungen in der Struktur der Ontologie umfassen die Hinzufügung ($addR$) und Löschung ($delR$) von Beziehungen sowie die Verschiebung eines Konzepts innerhalb der Hierarchie ($move$). Attributänderungen beziehen die Hinzufügung / Löschung eines Attributs ($addA$ / $delA$) oder die Modifikation eines Attributwerts ($chgAttValue$) ein. Die Menge der in dieser Arbeit mithilfe von COnto-Diff identifizierten Änderungsoperationen bildet das Diff-Evolutionsmapping $diff_{O,O'}$ zwischen zwei Ontologieversionen $O$ und $O'$. Diff-Evolutionsmappings werden insbesondere in Teil II (Evolution von Ontologiemappings) und III (Evolution von Annotationsmappings) dieser Arbeit genutzt.

# Teil II

# Evolution von Ontologiemappings

# 4

# Analyse der Evolution von Ontologiemappings

## 4.1 Motivation

Ontologien unterliegen regelmäßigen Änderungen, um stets den aktuellen Wissensstand ihrer Domäne zu repräsentieren. Neu veröffentlichte Versionen enthalten u. a. hinzugefügte Konzepte, Beziehungen und Attributwerte. Allerdings werden auch bereits bestehende Informationen überarbeitet oder sogar gelöscht. Zuvor bestimmte Ontologiemappings können durch derartige Ontologieänderungen ungültig werden [44]. Um weiterhin gültig und somit nutzbar zu sein, müssen veraltete Mappings neu bestimmt oder angepasst werden. Beispielsweise beruht das Referenzmapping des OAEI *Anatomy Track* auf sechs Jahre alten Ontologieversionen[20]. Zwar wurde das Referenzmapping über die Zeit korrigiert und verbessert, jedoch erfolgte keine Migration auf aktuellere MA- und NCIT-Versionen. Somit ist unklar, wie gut die Mappingqualität bezüglich aktueller Ontologieversionen ist.

Die Evolution von Ontologiemappings wurde bisher, insbesondere im Bereich der Lebenswissenschaften, kaum untersucht (siehe verwandte Arbeiten in Kapitel 2.2). Beispielsweise ist bisher nicht klar, wie und wie stark sich Mappings zwischen bekannten Ontologien der Lebenswissenschaften verändern. Es ist zu erwarten, dass Ontologieänderungen einen Einfluss auf abhängige Mappings haben, jedoch betrifft

[20] Das aktuelle OAEI-*Anatomy Track*-Referenzmapping basiert auf den 2007 gültigen MA- und NCIT-Versionen.

dies wahrscheinlich nur einen Teil der Ontologieänderungen. Um den manuellen Aufwand zu minimieren, werden Ontologiemappings häufig automatisch durch Match-Verfahren (siehe Kapitel 2.3) bestimmt. In Bezug auf die Evolution ist dabei allerdings unklar, inwieweit verschiedene Match-Verfahren mehr oder weniger stabile Mappings produzieren. Instabilität von Ontologiemappings deutet an, dass die zugrunde liegenden Ontologien stark überarbeitet werden, was u. a. auf die intensive Erforschung und Weiterentwicklung einer Domäne und eine entsprechende Anpassung und Verbesserung des in der Ontologie repräsentierten Wissens zurückzuführen ist. Je nachdem wie viele Bereiche einer Ontologie ein Match-Verfahren einbezieht, können Änderungen geringere oder stärke Auswirkungen auf die generierten Mappings haben. Eine Analyse der Ontologie- und Mappingevolution hilft u. a. zu verstehen, ob ein Ontologiemapping noch aktuell ist oder eine Adaptierung aufgrund von Änderungen in den Ontologien realisiert werden muss.

Dieses Kapitel umfasst eine Fallstudie zur Untersuchung der Evolution von Ontologien und Mappings für verschiedene Subdomänen der Lebenswissenschaften. Basierend auf den vorherigen Fragestellungen werden folgende Beiträge behandelt:

- Aufbauend auf dem in Kapitel 3.1 eingeführten Modell, wird ein generelles Versionierungsschema vorgestellt, das die Untersuchung der Evolution von Ontologien und Ontologiemappings erlaubt (Kapitel 4.2).
- Es werden ein generisches Modell sowie Maße zur Erfassung von Änderungen vorgestellt, um die Änderungsintensität in Ontologien und Mappings vergleichen zu können. Zudem erlaubt das Modell den Einfluss der Ontologieevolution auf abhängige Mappings zu analysieren, um beispielsweise herauszufinden, welche Ontologieänderungen zur Hinzufügung oder Löschung von Korrespondenzen führen (Kapitel 4.2).
- Es erfolgt eine Evaluierung für drei Szenarien aus den Lebenswissenschaften. Dabei wird untersucht wie sich die Ontologien und Mappings in verschiedenen Domänen entwickeln. Die Evolution automatisch bestimmter Ontologiemappings wird dabei für verschiedene Match-Verfahren vergleichend analysiert (Kapitel 4.3).

## 4.2 Modell für Ontologie- und Mappingänderungen

In diesem Kapitel werden ein Versionierungsschema und das verwendete Modell für Ontologie- und Mappingänderungen beschrieben. Zudem werden Änderungsfaktoren zur Bewertung der Evolutionsintensität zwischen aufeinanderfolgenden Ontologie- und Mappingversionen eingeführt. Ein weiteres Maß dient als Indikator, inwieweit

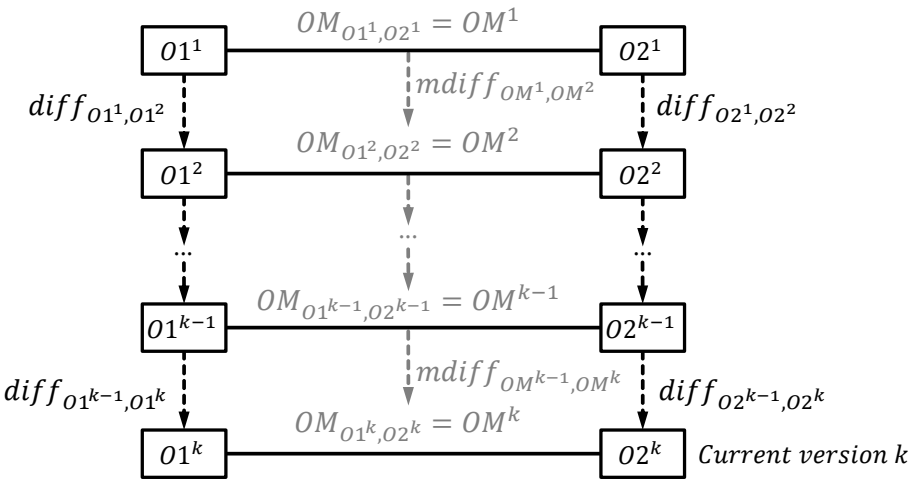



Abbildung 4.1: Allgemeines Versionierungsschema für aufeinanderfolgende Ontologie- und Mappingversionen.

die Evolution von Ontologien Auswirkungen auf Veränderungen in Ontologiemappings hat.

Zunächst wird das generelle Versionierungsschema für Ontologien und Mappings basierend auf den in Kapitel 3.1 definierten Ontologie- und Mappingmodellen eingeführt (siehe Abbildung 4.1). Dementsprechend ist eine Ontologieversion $O^v = (C^v, R^v, A^v, t)$ eine Momentaufnahme der Ontologie $O$, die zu einem bestimmten Zeitpunkt $t$ veröffentlicht wurde. Zur Vereinfachung wird im Modell anstelle von Veröffentlichungsdaten eine aufsteigende Nummerierung der Versionen ($v = 1, 2, \ldots$) verwendet. Es existiert eine Abfolge von Versionen ($v = 1 \ldots k$) für zwei unterschiedliche Ontologien $O1$ und $O2$, die jeweils durch ein Ontologiemapping $OM_{O1,O2}$ miteinander verknüpft sind. Zur Vereinfachung werden Ontologiemappings $OM^v = OM_{O1^v,O2^v}$ nur zwischen Ontologieversionen bestimmt, die zum gleichen Zeitpunkt gültig waren und somit die gleiche Versionsnummer aufweisen ($O1^v$ und $O2^v$). Die Unterschiede zwischen zwei Ontologie- und Mappingversionen werden jeweils als $diff(O^v, O^{v+1})$ und $mdiff(OM^v, OM^{v+1})$ bezeichnet und in den folgenden Abschnitten näher beschrieben.

### 4.2.1 Ontologieänderungen

Das Modell basiert auf den in Kapitel 3.2.2 vorgestellten Änderungsoperationen des COnto-Diff-Algorithmus [75]. Für diese Studie wird der $diff(O^v, O^{v+1})$ jeweils zwischen zeitlich aufeinanderfolgenden Ontologieversionen $O^v$ und $O^{v+1}$ unter Verwendung von COnto-Diff berechnet. COnto-Diff bestimmt eine Menge von Basis- und komplexen Änderungsoperationen, deren Anwendung die alte Version ($O^v$) in die neue Version ($O^{v+1}$) überführt.

| **Typ** | Informationserweiterung | Informationsreduktion | Informationsüberarbeitung |
|---|---|---|---|
| **Änderungs-operation** | *addC(c)* | *delC(c)* | *split(s, T)* |
| | *addSubGraph(c_root,C_Sub)* | *delSubGraph(c_root,C_Sub)* | *merge(S, t)* |
| | *addR(r)* | *delR(r)* | *substitute(c, c')* |
| | *addA(a)* | *delA(a)* | *move(c, P, P')* |
| | *revokeObsolete(c)* | *toObsolete(c)* | *chgAttValue(c, att, v1, v2)* |

Tabelle 4.1: Kategorisierung der COnto-Diff-Änderungsoperationen in drei Gruppen.

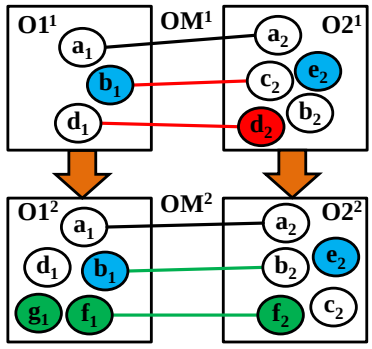
Tabelle 4.1 kategorisiert die betrachteten Änderungsoperationen in drei Gruppen. Die Evolution wird für aufeinanderfolgende Versionen $O^v \mapsto O^{v+1}$ betrachtet. Die erste Gruppe umfasst informationserweiternde Operationen, die zur Hinzufügung von Informationen in $O^v$ führen (z. B. neue Konzepte, Beziehungen und Attributwerte). Die zweite Gruppe (Informationsreduktion) enthält Änderungsoperationen, die Informationen aus $O^v$ entfernen, wie beispielsweise die Löschung von Konzepten, Beziehungen und Attributwerten. Alle weiteren Operationen wie *split* und *merge* überarbeiten bestehendes Wissen in der Ontologie (Informationsüberarbeitung), d. h. sie stellen weder eine reine Erweiterung noch eine reine Reduktion der Ontologieinformationen dar.

Um eine quantitative Analyse der Änderungen durchführen zu können, werden die Konzepte der jeweils betrachteten Ontologieversionen $O^v$ und $O^{v+1}$ auf Basis ihrer Änderungsoperationen in die folgenden Mengen eingeteilt. Jedes Konzept ist eindeutig über eine *Accession Number* identifiziert und wird genau einer Gruppe zugeordnet. Es gibt Konzepte, die nur in $O^v$ auftreten, in beiden Versionen enthalten sind oder nur in $O^{v+1}$ auftreten.

- ***Extension set:*** $Ext(O^{v \mapsto v+1})$ = Menge von Konzepten in $O^v \cup O^{v+1}$, die ausschließlich zu informationserweiternden Änderungsoperationen assoziiert sind.
- ***Reduction set:*** $Red(O^{v \mapsto v+1})$ = Menge von Konzepten in $O^v \cup O^{v+1}$, die ausschließlich zu informationsreduzierenden Änderungsoperationen assoziiert sind.
- ***Revision set:*** $Rev(O^{v \mapsto v+1})$ = Menge von Konzepten in $O^v \cup O^{v+1}$, die zu mindestens einer Änderungsoperation assoziiert sind, aber weder zu $Ext$ noch zu $Red$ gehören. Jedes $Rev$-Konzept unterlag also entweder einer Informationsüberarbeitung oder sowohl erweiternden als auch reduzierenden Änderungsoperationen im gleichen Versionsübergang.

Alle weiteren Konzepte bleiben unverändert, d. h. sie wurden nicht durch Änderungen beeinflusst. Abbildung 4.2 zeigt beispielhaft die Evolution zweier Ontologien $O1$ und $O2$ und eines Mappings zwischen diesen beiden Ontologien ($OM_{O1,O2}$).

(a)




(b)

| Ontologies / Mapping | | *Ext* | *Red* | *Rev* |
|---|---|---|---|---|
| | | $\{f_1, g_1\} \cup \{f_2\}$ | $\emptyset \cup \{d_2\}$ | $\{b_1\} \cup \{e_2\}$ |
| *Add* | $\{(b_1,b_2), (f_1,f_2)\}$ | 2/3 | 0 | 1/2 |
| *Del* | $\{(b_1,c_2), (d_1,d_2)\}$ | 0 | 1 | 1/2 |

Abbildung 4.2: (a) Beispielevolution für zwei Ontologien und ein Mapping für den Versionsübergang $v = 1 \mapsto 2$. Die Konzepte $b_1$ und $e_2$ wurden überarbeitet (blau), $d_2 \in O2$ wurde gelöscht (rot) und $g_1$, $f_1$ und $f_2$ wurden hinzugefügt (grün). Änderungen des Mappings zwischen $O1$ und $O2$ umfassen zwei Korrespondenzhinzufügungen $((b_1, b_2), (f_1, f_2))$ und zwei Korrespondenzlöschungen $((b_1, c_1), (d_1, d_2))$. (b) Matrix zum Einfluss der Ontologie- auf die Mappingänderungen.

Beispielsweise werden im Versionsübergang von $O2^1$ nach $O2^2$ drei Änderungen durchgeführt: Hinzufügung des Konzepts $f_2$ ($addC$), Löschung des Konzepts $d_2$ ($delC$) und die Änderung eines Attributwerts von $e_2$ ($chgAttValue$). Die drei Konzepte werden jeweils den drei Gruppen $Ext$, $Red$ und $Rev$ zugeordnet: $Ext(O2^{1\mapsto 2}) = \{f_2\}$, $Red(O2^{1\mapsto 2}) = \{d_2\}$ und $Rev(O2^{1\mapsto 2}) = \{e_2\}$. Alle anderen $O2$-Konzepte im Beispiel (Abbildung 4.2) bleiben unverändert.

Die Größe der drei Konzeptmengen $Ext$, $Red$ und $Rev$ erfasst den Änderungsgrad während der Evolution von $O^v$ zu $O^{v+1}$ quantitativ. Auf Basis von $Ext$, $Red$ und $Rev$ wird der Ontologieänderungsfaktor ($Ontology\ Change\ Ratio = OCR$) wie folgt definiert:

$$OCR(O^{v\mapsto v+1}) = \frac{|Ext(O^{v\mapsto v+1}) \cup Red(O^{v\mapsto v+1}) \cup Rev(O^{v\mapsto v+1})|}{|O^v \cup O^{v+1}|}$$

Der Ontologieänderungsfaktor für $O2$ im Beispiel aus Abbildung 4.2 ist folglich $OCR(O2^{1\mapsto 2}) = |\{f_2, d_2, e_2\}|/|\{a_2, b_2, c_2, d_2, e_2, f_2\}| = 0.5$.

### 4.2.2 Mappingänderungen

Zur Bestimmung von Änderungen in Mappings wird ein einfacher Mapping-Diff $mdiff(OM^v, OM^{v+1})$ berechnet. Für zwei aufeinanderfolgende Mappingversionen $OM^v$ und $OM^{v+1}$ erfasst der Mapping-Diff Hinzufügungen neuer Korrespondenzen ($Add$) sowie Löschungen bisher gültiger Korrespondenzen ($Del$). Zusätzlich könnte eine Veränderung des Ähnlichkeitswerts einer Korrespondenz zwischen verschiedenen Mappingversionen berücksichtigt werden. In Kapitel 5 wird die Historie und

Stabilität von Korrespondenzen bezüglich Änderungen des Ähnlichkeitswerts näher betrachtet. Für diese Studie werden geänderte Korrespondenzen in die zwei folgenden Gruppen eingeteilt:

- ***Addition set:*** $Add(OM^{v\mapsto v+1}) = OM^{v+1} \backslash OM^{v}$
- ***Deletion set:*** $Del(OM^{v\mapsto v+1}) = OM^{v} \backslash OM^{v+1}$

Alle anderen Korrespondenzen treten in beiden Mappingversionen auf und werden somit als unverändert angesehen. Basierend auf diesen beiden Mengen, wird der Mappingänderungsfaktor (*Mapping Change Ratio* = $MCR$) wie folgt definiert:

$$MCR(OM^{v\mapsto v+1}) = \frac{|Add(OM^{v\mapsto v+1}) \cup Del(OM^{v\mapsto v+1})|}{|OM^{v} \cup OM^{v+1}|}$$

Im Beispiel in Abbildung 4.2 gibt es zwei neue Korrespondenzen ($Add(OM^{1\mapsto 2}) = \{(b_1, b_2), (f_1, f_2)\}$) und zwei gelöschte Korrespondenzen $(b_1, c_2)$ und $(d_1, d_2)$. Da es eine unveränderte Korrespondenz $(a_1, a_2)$ gibt, beträgt der Mappingänderungsfaktor 0.8 ($MCR(OM^{1\mapsto 2}) = 4/5$).

### 4.2.3 Einfluss von Ontologie- auf Mappingänderungen

Um zu bestimmen inwieweit Ontologieänderungen Mappingänderungen beeinflussen bzw. auslösen, ist es sinnvoll die verschiedenen Ontologie- und Mappingänderungen im Zusammenhang zu betrachten. Dazu werden die drei definierten Gruppen von Konzeptänderungen (*Ext*, *Red*, *Rev*) mit den beiden Arten von Korrespondenzänderungen (*Add*, *Del*) verknüpft. Daraus ergeben sich sechs verschiedene Indikatoren, die zur Analyse der Mappingevolution eingesetzt werden (siehe Kapitel 4.3).

Nicht alle Ontologieänderungen führen tatsächlich zu Änderungen in einem abhängigen Mapping. Der Einflussfaktor (*Impact Ratio* = $IR$) gibt den Anteil geänderter Konzepte an, die an geänderten Korrespondenzen beteiligt sind und somit Einfluss auf Änderungen im Mapping haben. $IR$ ist kein Indikator für die Anzahl der geänderten Korrespondenzen. Das Maß wird für jede Menge geänderter Ontologiekonzepte $O_{Ch}$ (*Ext*, *Red*, oder *Rev*) und geänderter Mappingkorrespondenzen $OM_{Ch}$ (*Add* oder *Del*) wie folgt definiert:

$$IR(O_{Ch}, M_{Ch}) = \frac{|\{c \in O_{Ch} | \exists c' : (c,c') \in M_{Ch} \vee (c',c) \in M_{Ch}\}|}{|O_{Ch}|}$$

Um beispielsweise den Anteil additiver Ontologieänderungen zu erfassen, der zu neuen Korrespondenzen führt, wird der Einflussfaktor für $O_{Ch} = Ext(O1^{1\mapsto 2}) \cup Ext(O2^{1\mapsto 2})$ und $OM_{Ch} = Add(OM^{1\mapsto 2})$ bestimmt. Im Beispiel in Abbildung 4.2 erscheinen zwei ($f_1$, $f_2$) von insgesamt drei *Ext*-Konzepten in der Menge der hinzugefügten Korrespondenzen. Änderungen an diesen zwei Konzepten haben einen

Einfluss auf das abhängige Mapping. Der Einflussfaktor $IR(Ext, Add)$ beträgt in diesem Fall $\frac{2}{3}$. Es ist zu erwarten, dass erweiternde Änderungen ($Ext$) meist zu Korrespondenzhinzufügungen führen, wohingegen reduzierende Änderungen ($Red$) eher Korrespondenzlöschungen auslösen.

## 4.3 Analyse der Mappingevolution

Im Folgenden wird die Evolution von Ontologien und Mappings für verschiedene Subdomänen der Lebenswissenschaften analysiert. Dabei erfolgt u. a. ein Vergleich der Mappingevolution für verschiedene Match-Verfahren und eine Analyse zum Einfluss von Ontologieänderungen auf Mappingänderungen.

### 4.3.1 Datensätze und Konfigurationen

In der Evaluierung werden drei verschiedene Mapping-Szenarien aus dem Bereich der Lebenswissenschaften betrachtet:

- Anatomie (*Anatomy*): Mapping zwischen MA und dem Anatomieteil des NCIT (NCITa)
- Molekularbiologie (*Molecular Biology*): Mapping zwischen den zwei GO-Subontologien Molekulare Funktionen (MF) und Biologische Prozesse (BP)
- Chemie (*Chemistry*): Mapping zwischen ChEBI und NCIT

Für jede Eingabeontologie werden die im Juni und Dezember gültigen Versionen zwischen 2006 und 2010 verwendet (10 Versionen: 2006-06 bis 2010-12). Ontologiemappings basieren auf den zu einem Zeitpunkt gültigen Ontologieversionen und werden automatisch mit GOMMA generiert. Änderungen in Mappings resultieren ausschließlich aus Ontologieänderungen, da automatisch berechnete Mappings auf Basis sich ändernder Ontologieversionen verwendet werden. Es existieren demzufolge keine von der Ontologieevolution unabhängigen Mappingänderungen, wie sie durch Experten in manuell verifizierten Mappings vorgenommen werden können. Zur Berechnung der Ontologiemappings werden die folgenden metadatenbasierten Matcher eingesetzt:

- *Name*: Berechnung der String-Ähnlichkeit zwischen den Namen zweier Konzepte
- *NameSyn*: Berechnung der maximalen String-Ähnlichkeit zwischen den Namen und Synonymen zweier Konzepte

| | ontologies | | *Name 0.6* | | *NameSyn 0.6* | | *NameSyn 0.8* | | *Context 0.6* | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\|C^{2006\text{-}06}\|$ | growth | $\|M^{2006\text{-}06}\|$ | growth | $\|M^{2006\text{-}06}\|$ | growth | $\|M^{2006\text{-}06}\|$ | growth | $\|M^{2006\text{-}06}\|$ | growth |
| ***Anatomy*** | 8,806 | 1.1 | 1,496 | 1.1 | 1,636 | 1.1 | 1,264 | 1.1 | 1,272 | 1.0 |
| ***Molecular Biology*** | 18,974 | 1.6 | 852 | 1.1 | 1,531 | 1.7 | 251 | 1.6 | 465 | 1.6 |
| ***Chemistry*** | 69,005 | 1.7 | 1,353 | 3.9 | 3,242 | 3.2 | 1,930 | 3.7 | 277 | 6.1 |

Tabelle 4.2: Größe und Wachstum der Ontologien und Mappings. Die Tabelle zeigt die Anzahl der Konzepte ($|C^{2006-06}|$) und Anzahl der Mapping-Korrespondenzen ($|M^{2006-06}|$) in der ersten betrachteten Version, wobei $|C|$ die Summe der Quell- und Zielontologiekonzepte für jedes Szenario angibt. *growth* bezieht sich jeweils auf das Wachstum von der ersten (2006-06) zur letzten (2010-12) betrachteten Version.

- *Context*: Berechnung der String-Ähnlichkeit für den Konzeptkontext (konkatenierter String aus Eltern-, Kinder- und Konzeptnamen) zweier Konzepte

Die String-Ähnlichkeit wird jeweils auf Basis von Trigram (n-Gram, n=3) und dem Dice-Maß berechnet. In dieser Untersuchung liegt der Fokus auf der Evolutionsanalyse von Ontologiemappings, so dass hier nicht die Qualität der Mappings im Vordergrund steht. Die Auswahl der Match-Verfahren basiert auf früheren Studien, in welchen das Matching unter Ausnutzung von Konzeptnamen und Synonymen (*NameSyn*) sehr gute Ergebnisse insbesondere für Anatomieontologien erzielen konnte ([56, 63], siehe Kapitel 9 für [63]). Zusätzlich werden der *Name*- und *Context*-Matcher ausgeführt, um verschiedene metadatenbasierte Verfahren vergleichend analysieren zu können. Um möglichst genaue Ergebnisse (d. h. eine gute Precision) zu erreichen, werden die wahrscheinlichsten Korrespondenzen, die einen bestimmten Grenzwert überschreiten, ausgewählt. Für diese Studie wird ein einheitlicher Grenzwert von 0,6 für alle drei Match-Verfahren verwendet. Für *NameSyn* wird zusätzlich der striktere Grenzwert 0,8 untersucht. Zusätzlich werden für jedes Konzept nur die Korrespondenzen mit dem höchsten Ähnlichkeitswert sowie jene in einer kleinen Delta-Umgebung ausgewählt (MaxDelta-Selektion [41]).

### 4.3.2 Ontologie- und Mappingevolution

Tabelle 4.2 gibt einen Überblick zu den Größen der Ontologien ($|C|$) und Mappings ($|M|$) sowie deren Wachstum (*growth*) zwischen Juni 2006 und Dezember 2010. $|C|$ repräsentiert hier die kombinierte Ontologiegröße und gibt die Anzahl aller Konzepte in der Quell- und Zielontologie für jedes der drei Szenarien an. Das Wachstum (*growth*) bezieht sich auf die Änderung der Ontologie ($|C|$)- bzw. Mappinggröße ($|M|$) von der ersten zur letzten betrachteten Version:

$growth(|C|, \text{2006-06} \mapsto \text{2010-12}) = \frac{|C^{2006-06}|}{|C^{2010-12}|}$ bzw.

$growth(|M|, \text{2006-06} \mapsto \text{2010-12}) = \frac{|M^{2006-06}|}{|M^{2010-12}|}$.

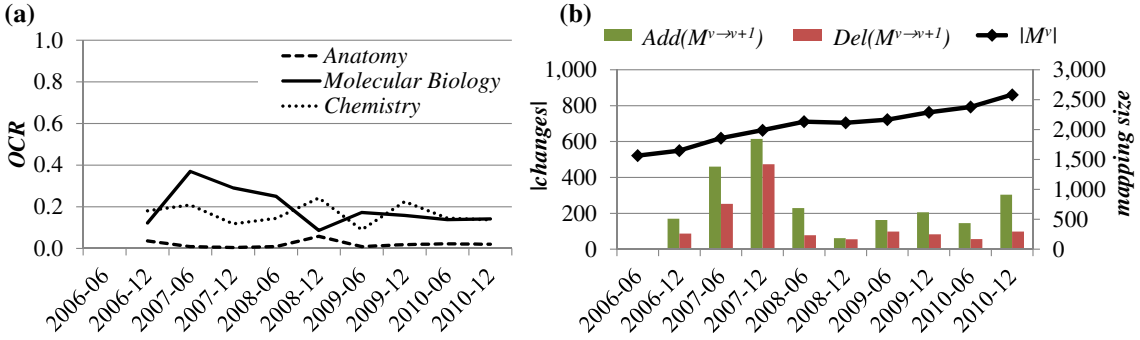



Abbildung 4.3: (a) Ontologieänderungsfaktoren ($OCR$). (b) Mappingevolution für den *NameSyn 0,6*-Matcher, *Molecular Biology*.

Im Anatomie-Szenario steigt die kombinierte Quell- und Zielontologiegröße leicht um den Faktor 1,1 auf fast 10.000 Konzepte. Hingegen wachsen die Ontologien für Molekularbiologie und Chemie um 60-70% auf ≈30.000 und ≈120.000 Konzepte. Für zwei der drei Szenarien (Anatomie und Molekularbiologie) entspricht das Wachstum der Mappings ungefähr jenem der Ontologien. Im Gegensatz dazu wachsen die Mappings im Chemie-Szenario wesentlicher stärker (Faktor 3-6) als die Ontologien (Faktor 1,7). Im Vergleich zur Ontologiegröße sind die Mappings teilweise sehr klein (insbesondere für den *Context*-Matcher im Chemie-Szenario), so dass ein starkes Mappingwachstum begünstigt wird. *Context* produziert eher wenige Korrespondenzen, da neben lokalen Konzeptinformationen auch Informationen aus Eltern- und Kinderkonzepten genutzt werden.

Die Verwendung eines höheren Grenzwerts für *NameSyn* führt zu kleineren Mappings. Insbesondere im Molekularbiologie-Szenario reduziert sich die Abdeckung der Quell- und Zielontologie durch das Mapping erheblich. Die Abdeckung einer Ontologie durch ein Mapping ist interessant in Bezug auf den Einfluss der Ontologieevolution. Wenn sich beispielsweise Ontologieregionen ändern, in welchen keine Korrespondenzen liegen, resultieren daraus kaum Mappingänderungen. Im Gegensatz dazu ändern sich Mappings sehr viel stärker, wenn diese sehr stark veränderliche Ontologieteile abdecken.

Abbildung 4.3(a) zeigt die Ontologieänderungsfaktoren ($OCR$) (siehe Kapitel 4.2.3) zwischen aufeinanderfolgenden Versionen des fünfjährigen Betrachtungszeitraums[21]. Verglichen mit den anderen beiden Domänen gab es in der Anatomiedomäne nur wenige Änderungen. Im Molekularbiologie-Szenario tritt insbesondere 2007 eine hohe Änderungsrate (fast 40%) auf. Ab 2008 sinkt der $OCR$ und ähnelt jenem des Chemie-Szenarios (circa 20%). Abbildung 4.3(b) stellt die Ergebnisse zur Mappingevolution für *NameSyn 0,6* im Bereich Molekularbiologie detaillierter dar. Insgesamt dominieren Korrespondenzhinzufügungen, so dass die finale Mappinggröße auf mehr als 2.500 Korrespondenzen anwächst. Allerdings gab es auch eine beträchtliche Anzahl

[21] Der erste $OCR$ errechnet sich aus dem Übergang von Version 2006-06 zu Version 2006-12.

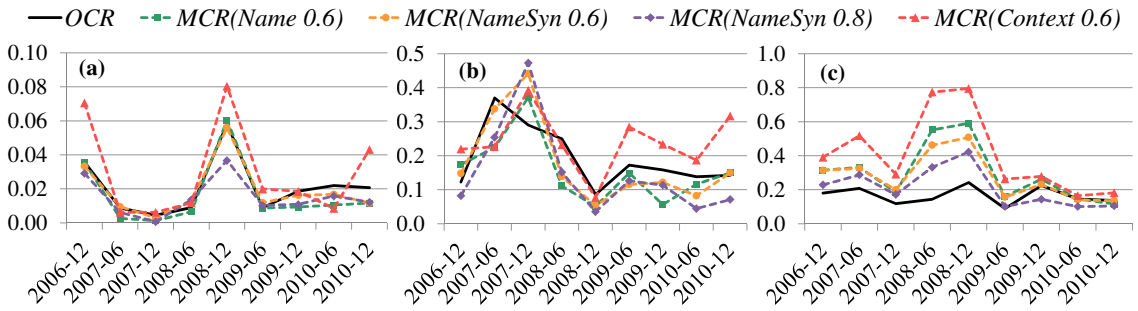



Abbildung 4.4: Ontologie ($OCR$)- und Mappingänderungsfaktoren ($MCR$) für drei Subdomänen der Lebenswissenschaften (a) *Anatomy*, (b) *Molecular Biology*, (c) *Chemistry*.

Löschungen. Beispielsweise werden Ende 2007 im Vergleich zur Vorversion beinahe 500 Korrespondenzen nicht mehr produziert, was auf massive Überarbeitungen von GO-BP und GP-MF im Jahr 2007 zurückzuführen ist. Das Beispiel verdeutlicht, dass durchaus starke Mappingänderungen auftreten können.

### 4.3.3 Vergleich der Match-Verfahren

Um die Mappingstabilität für verschiedene Match-Verfahren vergleichend zu analysieren, wird eine mögliche Korrelation zwischen Ontologie- und Mappingänderungen untersucht. Dazu werden die Ontologie- und Mappingänderungsfaktoren für die drei Szenarien und vier Match-Verfahren im Beobachtungszeitraum berechnet (Abbildung 4.4 a-c). Im Anatomie-Szenario, haben sich die Ontologien sowie die Mappings nur leicht verändert (siehe Wertebereich der y-Achse). Im Gegensatz dazu gibt es für die beiden anderen Szenarien einen überraschend hohen Anteil Mappingänderungen (10-80%). Insbesondere für Anatomie und Molekularbiologie zeigt sich eine deutliche Korrelation zwischen den Ontologieänderungsfaktoren (schwarze, durchgezogene Linien) und den Mappingänderungsfaktoren (farbige, gestrichelte Linien). Die *Name*- und *NameSyn*-Mappings sind insgesamt stabiler als die *Context*-Mappings. Insbesondere im Chemie-Szenario haben sich 2008 80% des *Context*-Mappings verändert. Die Verwendung des *Context*-Matchers führt zu einer höheren Instabilität der Mappings, da sich zusätzlich Änderungen der Eltern- und Kinderkonzepte auswirken. Beispielsweise ändert sich der Kontext eines Konzepts durch dessen Verschiebung von einem Elternknoten zu einem anderen. Im Molekularbiologie-Szenario ändern sich in Version 2007-12 insbesondere die *NameSyn*-Mappings, auch wenn das Maximum der Ontologieevolution bereits in der vorherigen Version (2007-06) lag. GO-BP und GO-MF wurden 2007 nacheinander bearbeitet, so dass die jeweiligen Anpassungen in verschiedenen Versionen veröffentlicht wurden. Erst die Kombination der Änderungen in beiden Sub-Ontologien führte 2007-12 zu den zahlreichen Mappingänderungen.

| | ***Ext*** | | | ***Red*** | | | ***Rev*** | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\|O_{Ch}\|$ | *IR* (*Ext*, ***Add***) | *IR* (*Ext*, ***Del***) | $\|O_{Ch}\|$ | *IR* (*Red*, ***Add***) | *IR* (*Red*, ***Del***) | $\|O_{Ch}\|$ | *IR* (*Rev*, ***Add***) | *IR* (*Rev*, ***Del***) |
| ***Anatomy*** | 95 | 18.7% | 0.1% | 7 | 0.0% | 7.8% | 89 | 6.8% | 4.1% |
| ***Molecular Biology*** | 2,359 | 4.6% | 0.7% | 223 | 2.4% | 8.8% | 2,209 | 3.5% | 2.1% |
| ***Chemistry*** | 8,377 | 11.7% | 1.2% | 366 | 3.5% | 5.3% | 6,441 | 8.1% | 4.0% |

Tabelle 4.3: Einfluss der Ontologieänderungen (*Ext*, *Red*, *Rev*) auf Mappingänderungen (*Add*, *Del*) für *NameSyn 0.6*. Anzahl Konzeptänderungen $|O_{Ch}|$ und prozentualer *Impact Ratio* $IR(O_{Ch}, M_{Ch})$ (Durchschnittwerte für alle Versionsübergänge).

### 4.3.4 Einfluss der Ontologie- auf Mappingänderungen

Tabelle 4.3 zeigt den Einfluss der Ontologieänderungen auf Mappingänderungen. Es werden die Ergebnisse für *NameSyn 0,6* als Durchschnittswert aller Versionsübergänge präsentiert. Die Tabelle zeigt die Anzahl geänderter Ontologiekonzepte ($|O_{Ch}|$) für jede Gruppe von Ontologieänderungen (*Ext*, *Red*, *Rev*) und deren Anteil (*Impact Ratio IR* als Prozentwert), der zu Hinzufügungen (*Add*) oder -löschungen (*Del*) im Mapping führte. Es wird deutlich, dass ein hoher Anteil (>80%) der Ontologieerweiterungen, -reduktionen und -überarbeitungen keinen Einfluss auf die Ontologiemappings hat. Dies begründet sich in der geringen Abdeckung der Ontologien durch die Mappings. Änderungen in Regionen der Ontologie, die nicht im Mapping abgedeckt sind, führen eher nicht zu Mappingänderungen. Der Einfluss der Ontologieänderungen hängt u.a. davon ab, welche Ontologieinformationen ein Match-Verfahren nutzt, um eine Korrespondenz im Mapping aufzunehmen.

Die erweiternden Ontologieänderungen (*Ext*) bilden den größten Anteil der Ontologieänderungen. Sie führen hauptsächlich zu Korrespondenzhinzufügungen und resultieren in allen drei Domänen eher selten in Korrespondenzlöschungen. *Red*-Konzepte führen eher zu Korrespondenzlöschungen jedoch auch zu einigen Korrespondenzhinzufügungen. Dies könnte durch spezifische Charakteristika der Match-Verfahren verursacht werden. Wenn beispielsweise ein Synonym eines Konzepts gelöscht wird, so dass die darauf basierende Korrespondenz verloren geht, kann dieses Konzept zu einem anderen Konzept verknüpft werden, so dass eine neue Korrespondenz hinzukommt. Dies gilt insbesondere für Selektionsverfahren wie *MaxDelta* oder *MaxN*, welche die jeweils beste(n) Korrespondenz(en) für ein Konzept auswählen. Somit kann eine Synonymlöschung im gleichen Evolutionsschritt sowohl zu einer Korrespondenzlöschung als auch zu einer Korrespondenzhinzufügung führen. Überarbeitete Konzepte (*Rev*) lösen sowohl Hinzufügungen (*Add*) als auch Löschungen (*Del*) von Korrespondenzen aus. Dies ist naheliegend, da Konzeptüberarbeitungen in einem Evolutionsschritt sowohl erweiternd als auch reduzierend sein können (z. B. eine Attributhinzufügung und -löschung). Absolut betrachtet lösen überarbeitende Änderungen mehr Korrespondenzlöschungen aus als die reduzierenden Änderungen. So sind z. B. im Chemie-Szenario circa 250 der Konzeptüberarbeitungen (siehe *Rev*: 4% von ≈6.400) jedoch nur circa 20 reduzierende Änderungen (siehe *Red*: 5,3% von ≈370) für Korrespondenzlöschungen (*Del*) verantwortlich. Konzeptüberarbeitungen

sind insgesamt für einen großen Teil der Mappingänderungen verantwortlich, wohingegen Löschungen eher eine geringere Rolle spielen. Die Analyse bietet eine aggregierte, überblicksartige Sicht auf den Einfluss von Ontologieänderungen auf Mappingänderungen. In Kapitel 6 werden Änderungsoperationen individuell betrachtet, um von Änderungen betroffene Korrespondenzen zu adaptieren.

## 4.4 Zusammenfassung

In diesem Kapitel wurde die Evolution von Ontologiemappings für drei verschiedene Domänen der Lebenswissenschaften (Anatomie, Molekularbiologie und Chemie) unter Verwendung verschiedener Match-Verfahren untersucht. Basierend auf dem eingeführten Modell und verschiedenen Maßen wurden die Evolutionsintensität der Ontologien und Mappings sowie der Einfluss von Ontologieänderungen auf Mappingänderungen analysiert.

Die Molekularbiologie- und Chemie-Ontologien wurden stark erweitert und überarbeitet, wohingegen die Anatomie-Ontologien relativ stabil sind. Dies ist nachvollziehbar, da die Anatomie verschiedener Spezies bereits seit langem gut erforscht ist, wohingegen die Molekularbiologie und Chemie derzeit intensiv erforschte Domänen sind. Da typischerweise bereits existierende Ontologieinformationen erweitert oder überarbeitet werden, gibt es eher wenige Löschungen für alle drei Domänen. Generell hat die Ontologieevolution einen deutlichen Einfluss auf Ontologiemappings, die auf Basis typischer metadatenbasierter Match-Verfahren bestimmt wurden. Insbesondere der strukturelle *Context*-Matcher produzierte relativ instabile Ergebnisse, wohingegen Mappings basierend auf dem *Name*- und *NameSyn*-Matcher vergleichsweise stabil sind. Wie erwartet lösen Ontologieerweiterungen eher Korrespondenzhinzufügungen aus, wohingegen reduzierende Änderungen eher zu Korrespondenzlöschungen führen. Konzeptüberarbeitungen finden relativ häufig statt und führen sowohl zu Hinzufügungen als auch zu Löschungen von Korrespondenzen. Abhängig von der Mappinggröße bzw.- abdeckung und dem verwendeten Match-Verfahren zeigen die Ergebnisse eine deutliche Korrelation zwischen Ontologie- und Mappingänderungen.

# 5

# Evolutionsbasierte Bewertung von Ontologiemappings

## 5.1 Motivation

Die bisherigen Untersuchungen zeigten, dass Ontologieänderungen Einfluss auf abhängige Ontologiemappings haben. Automatisch generierte Ontologiemappings hängen zudem von weiteren im Matching verwendeten Sekundärdatenquellen wie z. B. Thesauri, Instanzquellen und deren Assoziationen zu Ontologien (Annotationen) ab. Ähnlich zu Ontologien unterliegen Sekundärdatenquellen ebenfalls regelmäßigen Änderungen. Infolge der Evolution von Ontologien und Sekundärquellen können Ontologiemappings relativ instabil sein, d. h. die Ähnlichkeit zwischen zwei Konzepten kann über mehrere Versionen signifikant variieren. Abbildung 5.1 zeigt beispielhaft die Historie der Konzeptähnlichkeit zweier Korrespondenzen über 21 Versionen. Das betrachtete Mapping verknüpft die zwei GO-Subontologien molekulare Funktionen (MF) und biologische Prozesse (BP). Ein biologischer Prozess „besteht“ aus mehreren Funktionen und eine Funktion kann an mehreren Prozessen „beteiligt“ sein[22]. Folglich werden zwischen BP und MF n:m-Korrespondenzen anstelle von Äquivalenzbeziehungen bestimmt. Das Mapping wurde durch ein instanzbasiertes Match-Verfahren generiert, wobei die Ähnlichkeit zweier Konzepte anhand ihrer Assoziationen zu gleichen Instanzen (z. B. Gene, Proteine) bestimmt wird (siehe [104]). Beide Korrespondenzen haben einen relativ hohen Ähnlichkeitswert von

[22] Siehe auch `http://www.geneontology.org/GO.process.guidelines.shtml`:
„A biological process is a recognized series of events or molecular functions.“

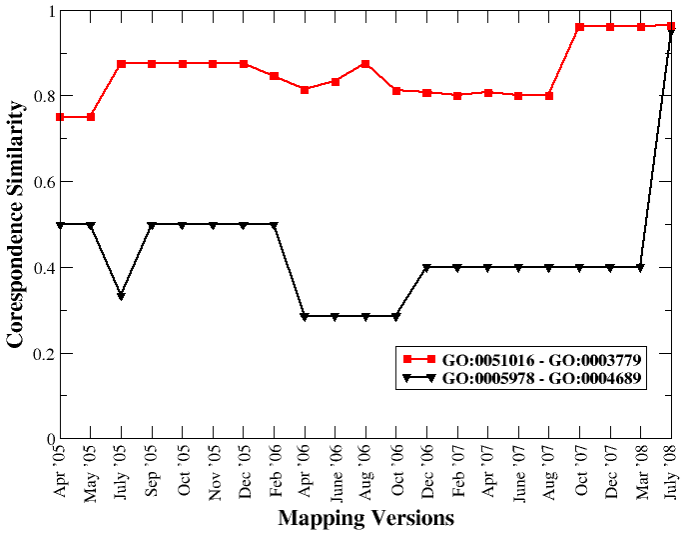



Abbildung 5.1: Historie von Ähnlichkeitswerten zweier GO-Korrespondenzen.

0,95 in der letzten Mappingversion, jedoch unterscheiden sie sich signifikant in ihrer Historie. Die rote Korrespondenz zwischen GO:0051016 ('*barbed-end actin filament capping*') und GO:0003779 ('*actin binding*') existiert kontinuierlich auf einem höheren Ähnlichkeitslevel und ist somit stabiler als die schwarze Korrespondenz zwischen GO:0005978 ('*glycogen biosynthetic process*') und GO:0004689 ('*phosphorylase kinase activity*'). Da der Ähnlichkeitswert der schwarzen Korrespondenz von 0,4 auf 0,95 springt, ist ihre Korrektheit fragwürdig und sollte entsprechend verifiziert werden. Die Beobachtung, dass Konzeptähnlichkeiten über die Zeit derartig schwanken, begründet sich hauptsächlich in der Evolution der Quellen, von welchen die Mappings abhängen. Dazu zählen Änderungen u. a. in Ontologien, Änderungen in Instanzquellen (und anderen Sekundärquellen) sowie Änderungen der Annotationen zwischen Instanzen und Ontologiekonzepten [81].

Ziel automatischer Match-Verfahren ist es, Mappings von hoher Qualität zu bestimmen. Dazu hat sich beispielsweise die Kombination verschiedener Match-Verfahren (z. B. [8], siehe Kapitel 2.3.1) bewährt. Bisherige Ansätze berücksichtigen jedoch nicht die Ontologieevolution, da sie nur Informationen der spezifischen, zu vergleichenden Ontologieversionen verwenden. Ziel ist es, die Stabilität von Korrespondenzen anhand einer Historie von Ähnlichkeitswerten zu berechnen und diese zur Bestimmung robuster Mappings zu nutzen. Instabile Korrespondenzen sollten einer manuellen Validierung unterzogen werden. Im Unterschied zum vorherigen Kapitel wird hier die Stabilität einzelner Korrespondenzen und nicht die Stabilität bzw. Evolutionsintensität eines gesamten Mappings betrachtet. Neben den in Kapitel 2.3.1 diskutierten verwandten Arbeiten, existieren relevante Ansätze, die die Evolution von Assoziationsregeln im Bereich des *Data Mining* untersuchen. Die Identifikation von Trends und Änderungen an Assoziationsregeln ist z. B. für wirtschaftliche Anwendungen wichtig, um auf veränderte Kundenanforderungen reagieren zu können.

In [2] wurde vorgeschlagen, Assoziationsregeln in verschiedenen Zeitintervallen zu beobachten, in dem Änderungen der „support"- und „confidence"-Werte der Regeln betrachtet werden. Ein weiterer Ansatz [121] sieht vor, wesentliche Unterschiede in Assoziationsregeln zu identifizieren. Ähnlich zu Mappingkorrespondenzen verbinden Assoziationsregeln ebenfalls Mengen von Elementen, um semantische Beziehungen zu beschreiben. Jedoch versuchen *Data Mining*-Ansätze, Regeln zu identifizieren, die sich signifikant in ihrer Evolution von anderen Regeln unterscheiden. Im Gegensatz dazu fokussiert der hier vorgestellte Ansatz auf die Unterscheidung stabiler und instabiler Korrespondenzen innerhalb eines Ontologiemappings, um Experten bei einer Verifikation automatisch generierter Korrespondenzen zu unterstützen.

Dieses Kapitel umfasst die folgenden Beiträge:

- Es wird ein generischer Ansatz zur evolutionsbasierten Bewertung automatisch generierter Ontologiemappings vorgestellt. Dieser ist unabhängig von dem zur Berechnung der Ähnlichkeitswerte verwendeten Match-Verfahren. Zusätzlich zu den bereits vermerkten Ähnlichkeitswerten der Korrespondenzen, soll jede Korrespondenz mit der Stabilität bezüglich ihrer historischen Änderungen annotiert werden. Folglich können Korrespondenzen nicht nur durch ihre Ähnlichkeitswerte sondern auch anhand der berechneten Stabilitätswerte klassifiziert und evaluiert werden.
- Es werden zwei Stabilitätsmaße definiert, die die Evolution der Ähnlichkeitswerte einer gegebenen Korrespondenz quantifizieren. Die „*Average Stability*"' betrachtet Änderungen in den verschiedenen Evolutionsschritten, wohingegen die „*Weighted Maximum Stability*" die Stabilität bezüglich des aktuellen Ähnlichkeitswerts ermittelt (Kapitel 5.3).
- Die Evaluierung präsentiert erste Ergebnisse am Beispiel eines instanzbasierten Match-Verfahrens im Bereich der Lebenswissenschaften. Die Stabilitätsmaße werden verwendet, um Korrespondenzen eines Mappings in verschiedene Gruppen wie „*accepted*", „*candidates*" oder „*questionable*" zu klassifizieren (Kapitel 5.4).

## 5.2 Grundlagen

Entsprechend dem in Kapitel 4.2 eingeführten Versionierungsmodell werden Ontologiemappings $OM^i = OM_{O1^i,O2^i}$ zwischen zwei Ontologieversionen $O1^i$ und $O2^i$, die zum gleichen Zeitpunkt $t$ gültig waren und die gleiche Versionsnummer $i = 1 \ldots n$ aufweisen, verwendet. Falls eine Mappingversion $OM^i$ von einer weiteren Sekundärdatenquelle $D$ abhängt, wird die zum gleichen Zeitpunkt gültige Version $D^i$ genutzt.

Im Folgenden werden zwei Konzepte $a$ und $b$ jeweils aus den Ontologien $O1^i$ und $O2^i$ sowie ein Match-Verfahren $m$ betrachtet. Um den Ansatz allgemein anwendbar

zu halten, werden keine Annahmen über das verwendete Match-Verfahren getroffen. Zur Berechnung der Ähnlichkeit nutzt ein Matcher neben den Konzepten $a$ und $b$ selbst (z. B. deren Name) auch weitere Informationen wie die Ontologiestruktur oder assoziierte Instanzen in einer Quellversion $D^i$.

Die durch ein Match-Verfahren $m$ bestimmte Ähnlichkeit zwischen zwei Konzepten $a$ und $b$ wird als $sim(a, b, m|O1^i, O2^i)$ mit $a \in O1^i$ und $b \in O2^i$ bezeichnet. Da während der Ontologieevolution u. a. Konzepte gelöscht und hinzugefügt werden, ist es nicht gegeben, dass ein Konzept $a$ in allen Versionen $O1^i$ $(i = 1 \ldots n)$ auftritt. Dementsprechend wird die *Korrespondenzähnlichkeit* $sim_i(a, b, m)$ zwischen zwei Konzepten bezüglich einer Version $i$ definiert.

$$sim_i(a, b, m) = \begin{cases} sim(a, b, m|A_i, B_i), \text{if } a \in A_i \wedge b \in B_i \\ 0, \text{ otherwise} \end{cases} \in [0, 1]$$

Wenn mindestens eines der beiden Konzepte nicht in der betrachteten Version auftaucht, wird die Korrespondenzähnlichkeit $sim_i(a, b, m)$ auf den Minimalwert 0 gesetzt. Diese Definition erlaubt eine Ähnlichkeitsberechnung für Korrespondenzen über verschiedene Versionen.

## 5.3 Stabilitätsmaße

Um die Historie verschiedener Korrespondenzen bewerten und vergleichen zu können, werden zwei Stabilitätsmaße definiert. Diese aggregieren die verschiedenen Ähnlichkeitswerte für jede Korrespondenz über mehrere Versionen. Beide Maße beruhen auf dem Ähnlichkeitswert der aktuellen Version $n$ sowie den Ähnlichkeitswerten der $k > 0$ vorherigen Versionen. Jedoch wird die maximale Anzahl $kmax$ der zuvor verfügbaren Versionen durch die Korrespondenz $(a, b)$ und das Match-Verfahren $m$ limitiert. Es können nur Versionen ab dem Zeitpunkt betrachtet werden, zu dem zwei Konzepte $a$ und $b$ zum ersten Mal gemeinsam auftraten. Zudem wird $kmax$ auf die erste Version beschränkt, für die $sim_i(a, b, m) > 0$ gilt. Es muss also die erste Version bestimmt werden, in der das Match-Verfahren $m$, für die Korrespondenz $(a, b)$, einen Ähnlichkeitswert $> 0$ berechnet. Ziel ist es, den initialen „Sprung“ von 0 auf einen positiven Ähnlichkeitswert nicht in die Berechnung der Stabilität einzubeziehen, da dies nicht als Instabilität bestraft werden soll. Dementsprechend wird $kmax$ in den späteren Stabilitätsdefinitionen genutzt und ist wie folgt definiert:

$$kmax_n(a, b, m) = \max_{k=1\ldots n-1}(\{k|sim_{n-k}(a, b, m) > 0\})$$

Diese Definition ist nur dann wohldefiniert, falls mindestens eine Korrespondenz mit $sim_i(a, b, m) > 0$ innerhalb der vorherigen $k$ Versionen existiert. Allerdings ist diese Beschränkung nicht relevant, da Korrespondenzen mit $sim_i(a, b, m) = 0$ *für alle* $i < n$ keine historischen Informationen beinhalten und somit auch nicht mit einem Stabilitätswert annotiert werden können.

Im Folgenden werden zwei Stabilitätsmaße (1) auf Basis des Durchschnitts und (2) auf Basis eines gewichteten Maximums der Ähnlichkeitswerte vorgestellt. Die *Average Stability* erfasst die durchschnittliche Schwankung der Ähnlichkeitswerte in den letzten $k$ Evolutionsschritten und ist für eine Korrespondenz $(a, b)$ wie folgt definiert:

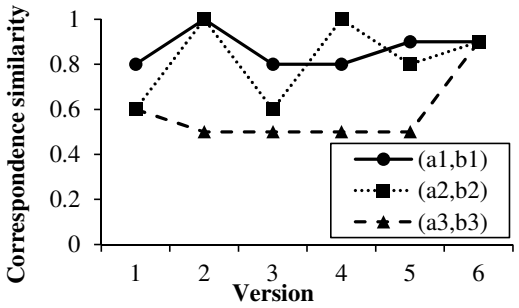
$$stabAvg_{n,k}(a,b,m) = \begin{cases} 1 - \frac{1}{k} \cdot \sum_{i=n-k}^{n-1} |sim_{i+1}(a,b,m) - sim_i(a,b,m)|, \text{ if } k \leq kmax_n(a,b,m) \\ stabAvg_{n,kmax_n(a,b,m)}(a,b,m), \text{ otherwise} \end{cases} \in [0,1]$$

Als Indikator für instabile Korrespondenzen, erfasst das Maß sowohl kleine als auch große Unterschiede der durch $m$ bestimmten Korrespondenzähnlichkeit. Folglich wird $(a, b)$ als stabil betrachtet, wenn nur wenige, eher kleine Unterschiede der Ähnlichkeit während der Evolution auftreten. Zu diesem Zweck werden die absoluten Differenzen der Korrespondenzähnlichkeiten aufeinanderfolgender Versionen für alle Evolutionsschritte von $n-k$ bis $n$ aufsummiert. Diese Summe wird mit der absoluten Anzahl Evolutionsschritte ($k$) normalisiert. Da jeder der $k$ Evolutionsschritte eine Änderung der Ähnlichkeit zwischen 0 und 1 beiträgt, wird für die Stabilität ebenso ein Wert zwischen 0 und 1 ausgegeben. Abschließend wird diese normalisierte Summe von 1 abgezogen, um einen *Average Stability*-Wert von 1 (0) für die perfekte Stabilität (vollständige Instabilität) zu erhalten.

Für eine Korrespondenz $(a, b)$ in der aktuellen Version $n$ und ein Match-Verfahren $m$ wird die *Weighted Maximum Stability* unter Verwendung des gewichteten Maximums für die letzten $k$ Evolutionsschritte wie folgt definiert:

$$stabWM_{n,k}(a,b,m) = \begin{cases} 1 - \max_{i=1\ldots k} \left[ \frac{|sim_n(a,b,m) - sim_{n-i}(a,b,m)|}{i} \right], \text{ if } k \leq kmax_n(a,b,m) \\ stabWM_{n,kmax_n(a,b,m)}(a,b,m), \text{ otherwise} \end{cases} \in [0,1]$$

Die *Weighted Maximum Stability* bestimmt, wie nah vergangene Ähnlichkeitswerte der aktuellen Ähnlichkeit $sim_n(a, b, m)$ einer Korrespondenz $(a, b)$ sind. Die Bewertung der Stabilität in den letzten $k$ Evolutionsschritten fokussiert also auf Version $n$. Für eine Version $n-i$ wird der Abstand $sim_{n-i}(a, b, m)$ zur aktuellen Ähnlichkeit $sim_n(a, b, m)$ betrachtet. Die Distanz wird durch die Anzahl der Evolutionsschritte $i$ normalisiert (bzw. gewichtet). Folglich haben Veränderungen in späteren Versionen einen größeren Einfluss als Veränderungen in früheren Versionen. Mit $stabWM$ können also Trends unterschieden werden, inwieweit die Korrespondenzähnlichkeit über die Zeit konstant ist, leicht ansteigt/abfällt oder in der nahen Vergangenheit plötzlich auf ein anderes Niveau springt. Die Auswahl des maximalen Werts über alle betrachteten Versionen $i$ gibt die größte Abweichung an und bestimmt die Stabilität. Der berechnete Maximalwert wird wiederum von 1 abgezogen, so dass vollständige Stabilität (Instabilität) dem Wert 1 (0) entspricht. Eine perfekte Stabilität ($stabWM = 1$) ergibt sich genau dann, wenn alle früheren Ähnlichkeiten $sim_i$ einer Korrespondenz $(a, b)$ mit der aktuellen Ähnlichkeit $sim_n$ übereinstimmen. Im Gegensatz dazu wird vollständige Instabilität ($stabWM = 1$) erreicht, falls $|sim_n - sim_{n-1}| = 1$ ist, d. h. die Korrespondenzähnlichkeit änderte sich im letzten Evolutionsschritt ($n - 1 \rightarrow n$) von 0 auf 1 oder umgekehrt.




| | $stabAvg_{6,5}$ | $stabWM_{6,5}$ |
|---|---|---|
| $(a_1,b_1)$ | 0.9 =1 - (0.2+0.2+0+0.1+0)/5 | $0.95 = 1 - \max\left(\frac{0}{1}; \mathbf{\frac{0.1}{2}}; \frac{0.1}{3}; \frac{0.1}{4}; \frac{0.1}{5}\right)$ |
| $(a_2,b_2)$ | 0.7 =1 - (0.4+0.4+0.4+0.2+0.1)/5 | $0.9 = 1 - \max\left(\mathbf{\frac{0.1}{1}}; \frac{0.1}{2}; \frac{0.3}{3}; \frac{0.1}{4}; \frac{0.3}{5}\right)$ |
| $(a_3,b_3)$ | 0.9 =1 - (0.1+0+0+0+0.4)/5 | $0.6 = 1 - \max\left(\mathbf{\frac{0.4}{1}}; \frac{0.4}{2}; \frac{0.4}{3}; \frac{0.4}{4}; \frac{0.3}{5}\right)$ |

Abbildung 5.2: Berechnung der Stabilitätswerte für drei Beispielkorrespondenzen.

Abbildung 5.2 (oben) zeigt beispielhaft die Evolution der Ähnlichkeitswerte dreier Korrespondenzen. Alle Korrespondenzen haben in der aktuellen Version ($n = 6$) einen Ähnlichkeitswert von 0,9, verhalten sich jedoch in den $k = 5$ vorherigen Versionen unterschiedlich. Abbildung 5.2 (unten) zeigt die Berechnung der beiden Stabilitätsmaße $stabAvg_{6,5}$ und $stabWM_{6,5}$. Die erste Korrespondenz ist über alle Versionen hinweg sehr stabil und erreicht somit hohe Stabilitätswerte. Im Gegensatz dazu unterliegt die zweite Korrespondenz starken Schwankungen, die im letzten Versionsübergang nachlassen. Dadurch ist die *Average Stability* eher niedrig, wohingegen die Korrespondenz eine hohe *Weighted Maximum Stability* erreicht. Der Ähnlichkeitswert der dritten Korrespondenz gleicht sich in den ersten fünf Versionen sehr, erhöht sich jedoch deutlich von Version 5 zu Version 6. Dieses Verhalten spiegelt sich in einer hohen *Average Stability* und gleichzeitig niedrigen *Weighted Maximum Stability* wider. Der letzte Evolutionsschritt trägt hauptsächlich zur niedrigen *Weighted Maximum Stability* bei. Das Beispiel illustriert, dass beide Stabilitätsmaße unterschiedliche Aspekte der Stabilität innerhalb eines gewissen Zeitraums quantifizieren. Die Nützlichkeit und Anwendbarkeit der Stabilitätsmaße zeigt sich in der folgenden Evaluierung.

## 5.4 Evaluierung

In diesem Abschnitt werden die Ergebnisse einer ersten Evaluierung des hier vorgestellten Ansatzes zur evolutionsbasierten Bewertung von Ontologiemappings gezeigt. Dazu werden die BP-und MF-Subontologien der GO [55] betrachtet. Die beiden

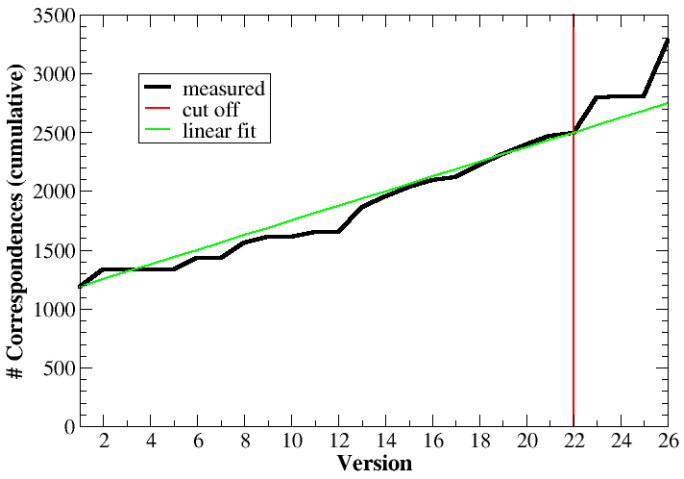



Abbildung 5.3: Kumulative Häufigkeit von Korrespondenzen bezüglich ihres Auftretens in einer Version.

Ontologien umfassten im April 2008 jeweils 15.131 und 8.827 Konzepte. Weiterhin werden GO-Annotationen aus Ensembl [91] genutzt. Im Juli 2008 umfasste Ensembl u. a. 46.704 Proteine, welche 80.705 (100.195) Annotationen zu BP (MF) haben. Es werden je 26 Versionen der Quellen zwischen 2004 und 2008 verwendet.

Zur Bestimmung mehrerer Versionen eines Ontologiemappings wurde ein zuvor erfolgreich angewendetes instanzbasiertes Match-Verfahren [175, 104] eingesetzt. Die Ähnlichkeit zwischen zwei Konzepten wird aus der Überlappung ihrer assoziierten Instanzen errechnet. Das Ähnlichkeitsmaß $sim_{min-3}$ fordert, dass jede Korrespondenz zwischen zwei Konzepten in mindestens drei Instanzen überlappen muss. Insgesamt führte das instanzbasierte Match-Verfahren zu 3.280 Korrespondenzen zwischen MF und BP in der letzten betrachteten Version. Alle Korrespondenzen weisen mindestens eine Ähnlichkeit $sim_{26}(a, b, min - 3)$ von 0,8 auf.

### 5.4.1 Quantitative Statistiken

In einer ersten Analyse wird die Existenz von Korrespondenzen in verschiedenen Versionen untersucht. Interessant ist dabei, wie viele Korrespondenzen aus Version 26 ebenfalls in den vorherigen (1-25) Versionen vorhanden waren. Abbildung 5.3 illustriert die kumulative Häufigkeit der Korrespondenzen bezüglich ihres ersten Auftretens (minimale Versionsnummer). Circa 1.200 der 3.280 Korrespondenzen der letzten Version waren bereits in der ersten Version vorhanden. Zwischen Version 1 und 22 zeigt sich ein beinahe linearer Anstieg (siehe Abbildung 5.3: *linear fit*). Anschließend steigt der Anteil der Korrespondenzen, die auch in der letzten Version enthalten sind, deutlich an. Ungefähr 76% aller Korrespondenzen (2.497) existieren mindestens seit Version 22 oder früher. Die restlichen 24% kamen erst in der

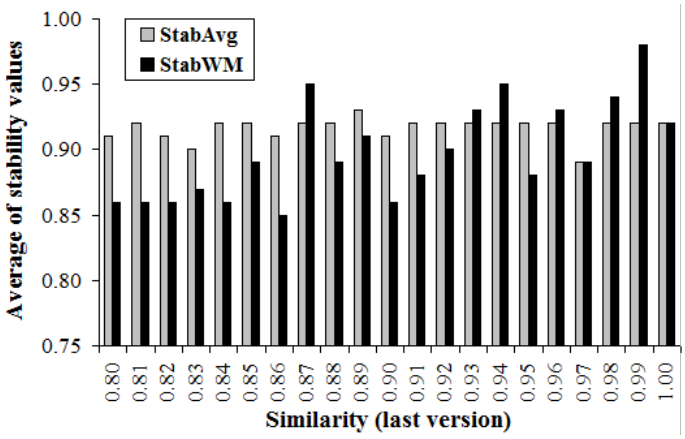



Abbildung 5.4: *Average Stability* gruppiert nach Korrespondenzen mit einem bestimmten Ähnlichkeitswert in der letzten Version.

jüngeren Vergangenheit hinzu. Für Korrespondenzen, die nur in wenigen Versionen auftreten, ist die Signifikanz der Stabilitätsinformation limitiert. Daher werden im Folgenden nur die 2.497 Korrespondenzen betrachtet, die in Version 22 oder früher zum ersten mal auftraten.

Zur Bewertung der *Average Stability* wird mit $k = 25$ die vollständige Historie betrachtet ($stabAVG_{26,25}$). Dies spiegelt die Langzeitstabilität einer Korrespondenz wider. Der Wert $k = 25$ impliziert nicht, dass alle Korrespondenzen in allen 25 vorherigen Versionen auftauchen müssen. Durch die Verwendung von $kmax$ in der Definition der Stabilitätsmaße wird sichergestellt, dass $stabAVG_{26,25}$ für alle 2.497 Korrespondenzen wohldefiniert ist. Für die *Weighted Maximum Stability* wird $k = 4$ verwendet ($stabWM_{26,4}$), so dass die Kurzzeitstabilität bzw. Entwicklungen der kürzeren Vergangenheit einer Korrespondenz bewertet werden können.

Die neuen Maße sollen einen zusätzlichen Nutzen für die Annotation von Match-Ergebnissen bringen. Daher soll die statistische Unabhängigkeit der Stabilitätsmaße im Vergleich zum Ähnlichkeitsmaß ($sim_{26}$) untersucht werden. Zu diesem Zweck werden für die letzte Version Gruppen von Ähnlichkeiten (jeweils der Größe 0,01 ) zwischen 0,8 und 1 unterschieden. Für jede der Gruppen werden die durchschnittlichen $stabAVG_{26,25}$- und $stabWM_{26,4}$-Werte berechnet (siehe Abbildung 5.4). Die Werte für $stabAVG_{26,25}$ ($stabWM_{26,4}$) reichen von 0,89 bis 0,93 (0,85 bis 0,98), haben einen Mittelwert von 0,92 (0,9) und zeigen keinen klaren Trend bzw. keine spezielle Ordnung. Dies zeigt, dass die Stabilitätsmaße statistisch unabhängig vom berechneten Ähnlichkeitswert des Match-Verfahrens sind und somit hilfreich für eine Klassifikation der Korrespondenzen sein können.

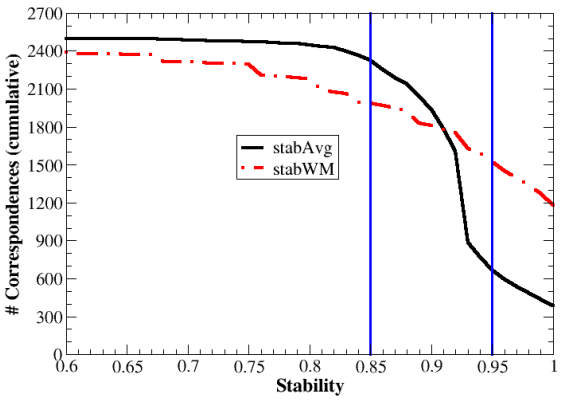



Abbildung 5.5: Kumulative Häufigkeit von Korrespondenzen für $stabAVG_{26,25}$ und $stabWM_{26,4}$.

### 5.4.2 Klassifikation der Korrespondenzen

Stabilitätsmaße können neben der Ähnlichkeit zur Bewertung und Klassifikation automatisch berechneter Korrespondenzen genutzt werden. Dies erlaubt eine detailliertere Auswertung der Glaubwürdigkeit einer Korrespondenz, wodurch die manuelle Verifikation unterstützt wird. Die folgende Evaluierung präsentiert beispielhaft eine mögliche Anwendung der Maße zur Unterstützung einer manuellen Entscheidung bezüglich der Korrespondenzkorrektheit. Dazu werden Korrespondenzen in verschiedene Gruppen eingeteilt.

Für jedes Maß ($stabAVG_{26,25}$, $stabWM_{26,4}$, $sim_{26}$) werden ein oberer ($t_{high}$) und ein unterer ($t_{low}$) Grenzwert verwendet. Korrespondenzen mit einem Wert größer als $t_{high}$ gelten als die besten Korrespondenzen bezüglich des Kriteriums, wohingegen Korrespondenzen zwischen $t_{high}$ und $t_{low}$ bzw. niedriger als $t_{low}$ jeweils als mittelmäßig bzw. unzureichend eingestuft werden.

Um die Grenzwerte für die Stabilitätsmaße zu setzen, werden zunächst $stabAVG_{26,25}$ und $stabWM_{26,4}$ analysiert. Abbildung 5.5 zeigt die kumulative Anzahl der Korrespondenzen bezüglich der zwei Stabilitätsmaße zwischen 0,6 und 1. Für beide Maße weist eine verhältnismäßig kleine Anzahl aller Korrespondenzen Stabilitätswerte von circa 0,85 oder kleiner auf. Circa 6% (20%) aller Korrespondenzen weisen Werte unter 0,85 für $stabAVG_{26,25}$ ($stabWM_{26,4}$) auf. Dies spiegelt sich in der stabilen, kumulativen Korrespondenzanzahl zwischen 0,6 und 0,85 wider. Jedoch unterscheidet sich das Verhalten von $stabAVG_{26,25}$ und $stabWM_{26,4}$ für Stabilitätswerte größer 0,85. Während die Kurve für $stabAVG_{26,25}$ zwischen 0,85 und 0,95 deutlich abfällt (von 2.330 auf 669), sinkt die kumulative Häufigkeit für $stabWM_{26,4}$ nur leicht (von 1.985 auf 1.528). Für $stabAVG_{26,25}$ ($stabWM_{26,4}$) erreichen 377 (1.178) Korrespondenzen perfekte Stabilität. Dementsprechend werden die unteren Grenzwerte $t_{stabAVG,low}$ und $t_{stabWM,low}$ auf 0,85 und die oberen Grenzwerte $t_{stabAVG,high}$ und $t_{stabWM,high}$ auf 0,95 gesetzt (als Linien in Abbildung 5.5 veranschaulicht). Auf Basis der Erfahrung

| $sim_{26}$ > 0.9 / $sim_{26}$ ≤ 0.9 | stabWM > 0.95 | 0.95 ≥ stabWM ≥ 0.85 | 0.85 > stabWM | Σ |
|---|---|---|---|---|
| stabAvg > 0.95 | 424 / 44 | 37 / 55 | 11 / 25 | 596 |
| 0.95 ≥ stabAvg ≥ 0.85 | 863 / 96 | 203 / 212 | 235 / 125 | 1734 |
| 0.85 > stabAvg | 17 / 5 | 13 / 16 | 85 / 31 | 167 |
| Σ | 1449 | 536 | 512 | 2497 |

Abbildung 5.6: Anzahl der durch $stabAvg_{26,25}$, $stabWM_{26,4}$ und $sim_{26}$ klassifizierten Korrespondenzen.

früherer Match-Aufgaben, die $sim_{min-3}$ nutzten, wird der obere Grenzwert für $sim_{26}$ ($t_{sim,high}$) auf 0,9 gesetzt. Zuvor wurde bereits implizit $t_{sim,low} = 0,8$ angewendet (siehe Kapitel 5.4.1).

Die Grenzwerteinstellungen sind spezifisch für das hier betrachtete Szenario und können für andere Match-Aufgaben, z. B. für andere Ontologien, Sekundärquellen oder Match-Verfahren abweichen. Beispielsweise hängt die Häufigkeit der Ontologieänderungen von der Domäne ab und beeinflusst somit die Grenzwerte der Stabilitätsmaße. Zudem kann die Wahl der $k$ betrachteten vorherigen Versionen vom Ausmaß der Änderungen abhängen. Beispielsweise könnten nur Hauptversionen (*major releases*) und nicht die Zwischenversionen (*minor releases*) einer Ontologie berücksichtigt werden. Eine automatische Konfiguration der verschiedenen Parameter ist wünschenswert und Ziel zukünftiger Arbeiten.

Abbildung 5.6 zeigt eine Klassifikation der Korrespondenzen bezüglich der zwei Stabilitätsmaße und des Ähnlichkeitswerts in der letzten Version. Die Einteilung veranschaulicht beispielhaft die Anwendbarkeit des hier vorgestellten Ansatzes unter Verwendung der beschriebenen Grenzwertkonfigurationen. Allgemein sinkt die Glaubwürdigkeit der Korrespondenzmengen von oben links nach unten rechts. Zum besseren Verständnis werden die Ergebnisse in drei Gruppen eingeteilt: I (weiß), II (hellgrau) and III (dunkelgrau). Korrespondenzen der Gruppe I umfassen 54,8% (1.368) aller Korrespondenzen und haben voraussichtlich die beste Qualität, da sie für mindestens zwei der Kriterien sehr gute und für keines der Kriterien unzureichende Werte aufweisen. Dementsprechend können diese Korrespondenzen akzeptiert werden (*accepted*). Die zweite Gruppe (II) deckt 15,3% (382) der Korrespondenzen ab. Diese erreichen bezüglich der Maße hauptsächlich mittlere bis hohe Werte und können als Kandidaten (*candidates*) eingeordnet werden. Gruppe III umfasst schließlich 29,9% (747) der Korrespondenzen, die überwiegend unzureichende Werte für mindestens eines der Kriterien aufweisen und somit als fragwürdig (*questionable*) klassifiziert werden.

Diese ersten Evaluierungsergebnisse zeigen, dass die vorgestellten Stabilitätsmaße eine präzisere Einschätzung der Glaubwürdigkeit und Korrektheit automatisch ge-

nierter Korrespondenzen unterstützen können. Um eine Klassifikation der Korrespondenzen in verschiedene Gruppen zu ermöglichen, wurden für die Maße obere und untere Grenzwerte eingesetzt. Anhand einer solchen Klassifikation kann der in Kapitel 3.1.3 eingeführte Status einer Korrespondenz angepasst werden. Als *candidates* eingestufte Korrespondenzen können den Status *to verify* erhalten, wohingegen *questionable*-Korrespondenzen verworfen werden sollten, da diese sehr instabil sind. Die als *accepted* klassifizierten Korrespondenzen sind glaubwürdig, da diese bereits über einen langen Zeitraum hohe Ähnlichkeitswerte aufweisen. Sie könnten den Status *handled* erhalten. Allerdings sollten automatisch generierte Korrespondenzen grundsätzlich durch Experten überprüft werden. Solch eine manuelle Verifikation kann durch die vorgestellten Stabilitätsmaße und Klassifikationskriterien unterstützt werden.

Die hier vorgestellten Maße zur Bewertung der Stabilität einzelner Korrespondenzen wurden später zusätzlich für metadatenbasierte Match-Verfahren evaluiert [90]. Dabei konnte gezeigt werden, dass beispielsweise ein *Name*-Matcher wesentlich stabilere Ergebnisse als das hier untersuchte instanzbasierte Verfahren erzeugt. Instanzbasierte Match-Verfahren hängen neben der Evolution der Ontologien zusätzlich von der Evolution der verwendeten Instanzen und Annotationen ab und produzieren daher tendenziell instabilere Ergebnisse.

## 5.5 Zusammenfassung

Es wurde ein evolutionsbasierter Ansatz zur Bewertung von Ontologiemappings vorgestellt. Dieser nutzt zwei Stabilitätsmaße, die die Historie der Ähnlichkeitswerte von Korrespondenzen verwenden. Die *Average Stability* berücksichtigt Änderungen zwischen allen betrachteten aufeinanderfolgenden Versionen, wohingegen die *Weighted Maximum Stability* die Stabilität bezüglich der Ähnlichkeitswerte in der letzten Version eines Ontologiemappings untersucht. Die vorgestellten Stabilitätsmaße können neben der Ähnlichkeit genutzt werden, um die Korrespondenzen eines Ontologiemappings zu beurteilen und zu bewerten. Weiterhin können Korrespondenzen mithilfe der Maße klassifiziert, d. h. in verschiedene Gruppen wie z. B. *accepted*, *candidates* und *questionable* eingeteilt werden. Bisherige Match-Verfahren beziehen keine historischen Informationen zu Korrespondenzen ein und können durch den hier vorgestellten Ansatz ergänzt werden. Der Ansatz ist generisch, da er unabhängig vom verwendeten Match-Verfahren auf jedes automatisch generierte Ontologiemapping angewendet werden kann. Die Stabilitätsmaße können individuell (applikationsspezifisch) angepasst werden. Erste Evaluierungsergebnisse im Bereich der Lebenswissenschaften zeigen die Anwendbarkeit des Ansatzes zur Klassifikation von Korrespondenzen eines automatisch generierten Ontologiemappings.

# 6

# Adaptierung von Ontologiemappings

## 6.1 Motivation

Die bisherigen Untersuchungen zeigten, dass die Evolution von Ontologien Einfluss auf abhängige Ontologiemappings hat. Da Mappings infolge von Ontologieänderungen ungültig werden können, müssen sie entsprechend angepasst werden, um konsistent bezüglich neu veröffentlichter Ontologieversionen zu sein. Beispielsweise erfordert eine neue Ontologieversion in BioPortal [141] oder UMLS [16] die Adaptierung assoziierter Mappings, so dass Nutzer und Applikationen auf aktuellen Mappingversionen arbeiten können. In diesem Kapitel werden verschiedene Methoden zur weitestgehend automatischen Adaptierung von Ontologiemappings vorgestellt. Ziel ist es, eine aufwendige Neubestimmung des gesamten Mappings zu vermeiden und stabile bzw. unveränderte Mappingteile wiederzuverwenden. Die Migration von Ontologiemappings ist insbesondere für komplexe Änderungen, wie die Aufspaltung eines Konzepts in mehrere Konzepte, nicht trivial. In diesem Fall muss eine frühere Korrespondenz zu dem noch nicht gespaltenen Konzept angepasst werden, so dass eine oder mehrere neue Korrespondenzen entstehen. Jede Art von Ontologieänderung erfordert möglicherweise unterschiedliche Aktionen, um ein abhängiges Ontologiemapping zu aktualisieren. Bisher existieren nur wenige Forschungsarbeiten zur Adaptierung von ontologiebasierten Mappings (siehe Kapitel 2.2.2). Die meisten Ansätze berücksichtigen nicht den Einfluss unterschiedlicher Ontologieänderungen oder basieren auf einer eingeschränkten Menge von Änderungsoperationen. Beispielsweise wurden die Aufspaltung existierender Konzepte (*split*) oder die Erstellung neuer Konzepte (*addC*), die jeweils zu neu-

en Korrespondenzen führen können, bisher nicht betrachtet. Darüber hinaus sollte während der Adaptierung der semantische Typ von Korrespondenzen (z. B. Äquivalenz, „*less / more general*“) berücksichtigt werden, anstatt wie bisherige Verfahren eine einheitliche Semantik für alle Korrespondenzen (z. B. nur Äquivalenz) zu unterstellen. Es ist zudem sinnvoll, adaptierte Korrespondenzen gegebenenfalls für eine manuelle Verifizierung zu markieren. Bisherige Arbeiten zur Adaptierung von Schema- und Ontologiemappings führten keine Evaluierung bezüglich der Mappingqualität durch, so dass unklar ist, inwieweit möglichst korrekte und vollständige Mappings erzeugt werden. Dieses Kapitel umfasst folgende Beiträge:

- Es wird ein *kompositionsbasierter Ansatz* zur Adaptierung von Ontologiemappings vorgestellt. Dieser nutzt das Prinzip der Mappingkomposition, indem das veraltete Ontologiemapping mit einem Ontologiemapping zwischen der alten und neuen Ontologieversion kombiniert wird (Kapitel 6.3).

- Ein alternativer, *Diff-basierter Ansatz* nutzt ein Diff-Evolutionsmapping, das eine Menge von Änderungen umfasst. Der Ansatz verwendet eine Sammlung sogenannter *Change Handler*, um verschiedene änderungsspezifische Mappinganpassungen vorzunehmen (Kapitel 6.4).

- Es erfolgt eine Evaluierung beider Ansätze bezüglich der Qualität (Precision, Recall, F-Measure) der adaptierten Ontologiemappings. Dazu werden existierende Mappingversionen zwischen sehr großen biomedizinischen Ontologien aus UMLS extrahiert. Die Ergebnisse zeigen, dass Ontologiemappings weitestgehend automatisch adaptiert werden können. Zudem können verschiedene Mappinganpassungen für unterschiedliche Arten von Ontologieänderungen vorgeschlagen werden, so dass eine manuelle Reparatur von Mappings infolge von Ontologieevolution vereinfacht wird (Kapitel 6.5).

## 6.2 Generelles Szenario

Dieses Kapitel basiert auf dem im Grundlagenkapitel 3.1 eingeführten Modell für Ontologien und Ontologiemappings sowie deren Versionen. Abbildung 6.1 zeigt das hier verwendete generelle Szenario für zwei Ontologien $O1$ und $O2$ sowie deren neuere Versionen $O1'$ und $O2'$. Ein Mapping $OM_{O1,O2}$ verbindet die alten Versionen der beiden Ontologien. Die Aufgabenstellung ist nun die neue Mappingversion $OM_{O1',O2'}$ zu bestimmen. Um dies möglichst automatisch zu realisieren, werden Evolutionsmappings zwischen den alten und neuen Ontologieversionen verwendet. Zwei Ontologiemappings $OM_{O1,O1'}$ und $OM_{O2,O2'}$ enthalten jeweils Korrespondenzen zwischen den Konzepten der alten ($O1$ / $O2$) und neuen ($O1'$ / $O2'$) Ontologieversion. Mappings zwischen verschiedenen Ontologieversionen können beispielsweise durch Anwendung eines Match-Verfahrens generiert werden. Der hier vorgestellte

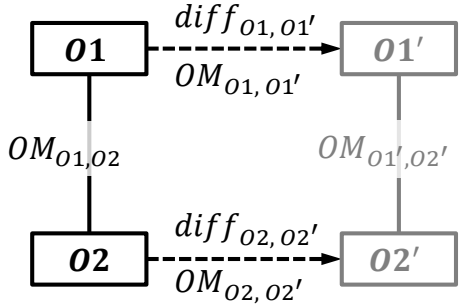



Abbildung 6.1: Szenario zur Evolution von Ontologien und Ontologiemappings.

kompositionsbasierte Ansatz nutzt die Mappings $OM_{O1,O2}$, $OM_{O1,O1'}$ und $OM_{O2,O2'}$, um das adaptierte Mapping $OM_{O1',O2'}$ durch Mappingkomposition zu erstellen.

Alternativ können Diff-Evolutionsmappings ($diff_{O1,O1'}$ und $diff_{O2,O2'}$) zwischen den alten und neuen Ontologieversionen verwendet werden. Diese umfassen alle Änderungen, die während der Evolution von $O1$ nach $O1'$ und $O2$ nach $O2'$ auftraten. Ein Diff-Evolutionsmapping kann durch einen Diff-Algorithmus wie PromptDiff [140] oder COnto-Diff [75] bestimmt werden. Die verwendeten Änderungen sind im Grundlagenkapitel in Tabelle 3.1 aufgelistet. Der hier präsentierte Diff-basierte Ansatz nutzt die Diff-Evolutionsmappings $diff_{O1,O1'}$ und $diff_{O2,O2'}$, um das adaptierte Mapping $OM_{O1',O2'}$ zu erstellen.

Um ein möglichst vollständiges Ontologiemapping bezüglich der neuen Ontologieversionen zu erstellen, sollen zusätzlich neue Korrespondenzen bestimmt werden, falls die zugrunde liegenden Ontologien um neue Konzepte erweitert wurden. Da eine manuelle Mappingbestimmung für sehr große Ontologien aufwendig oder kaum realisierbar ist, kommen (semi-) automatische Match-Verfahren zum Einsatz. Dazu wird die bereits zuvor eingesetzte Match-Strategie basierend auf den Namen und Synonymen der Konzepte (*NameSyn*) verwendet (siehe Kapitel 4).

## 6.3 Kompositionsbasierte Adaptierung

Dieser Abschnitt stellt den kompositionsbasierten Mappingadaptierungsansatz vor. Die Stärke des Ansatzes liegt in der Wiederverwendung eines zuvor bereits validierten Mappings, um somit eine aufwendige Neuberechnung bereits bestätigter Korrespondenzen zu vermeiden. Typischerweise beschränken sich Änderungen auf einen relativ kleinen Teil der Ontologien, so dass der größte Teil des neuen Mappings voraussichtlich leicht bestimmt werden kann.

Das Beispiel in Abbildung 6.2 dient der Veranschaulichung und zeigt die Evolution einer Anatomieontologie ($O2 \mapsto O2'$). Änderungen in $O2$ erfordern die Adaptierung des Mappings $OM_{O1,O2}$. Insbesondere müssen einige frühere Korrespondenzen ge-

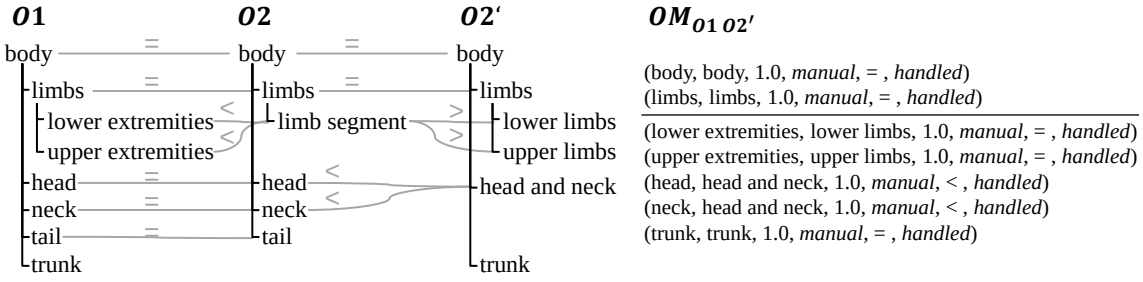



Abbildung 6.2: Beispiel zur Mappingevolution.

löscht und neue Korrespondenzen hinzugefügt werden. Das korrekt adaptierte Mapping $OM_{O1,O2'}$ ist rechts im Beispiel dargestellt. Die Korrespondenzen oberhalb der Linie bleiben unverändert im Vergleich zu $OM_{O1,O2}$. Korrespondenzen unterhalb der Linie werden entsprechend der Evolution von $O2$ angepasst (z. B. (*lower extremeties*, *lower limbs*), (*upper extremeties*, *upper limbs*)) oder neu erstellt (*trunk*, *trunk*). Die Korrespondenz (*tail*, *tail*) muss entfernt werden, da das Konzept '*tail*' in der neuen Version $O2'$ nicht mehr auftaucht. Der kompositionsbasierte Ansatz strebt eine Adaptierung durch Komposition des veralteten Ontologiemappings $OM_{O1,O2}$ mit dem Mapping $OM_{O2,O2'}$ an. Außerdem soll geprüft werden, ob hinzugefügte Konzepte zu neuen Korrespondenzen führen.

Die Komposition von zwei Mappings $OM_{A,B}$ und $OM_{B,C}$ generiert ein Mapping $OM_{A,C}$ zwischen den Ontologien $A$ und $C$. Entsprechend der im Rahmen des Model Management eingeführten Operatoren (siehe Kapitel 2.2.1), wird ein Operator zur Komposition von Ontologiemappings wie folgt definiert:

$$OM_{A,C} = \texttt{compose}\ (OM_{A,B}, OM_{B,C}) = OM_{A,B} \circ OM_{B,C} =$$
$$\{(c_1, c_2, aggSim(sim_1, sim_2), \texttt{getNewType}(semType_1, semType_2),$$
$$\texttt{getNewStatus}(semType_1, semType_2)) |$$
$$c_1 \in A, c_2 \in C, b \in B : \exists (c_1, b, sim_1, semType_1, status_1) \in OM_{A,B} \wedge$$
$$\exists (b, c_2, sim_2, semType_2, status_2) \in OM_{B,C}\}$$

Die Erstellung einer Korrespondenz $(c_1, c_2)$ in $OM_{A,C}$ erfordert die Existenz zweier Korrespondenzen $(c_1, b)$ und $(b, c_2)$, wobei $c_1$ und $c_2$ zu dem gleichen Konzept $b \in B$ assoziiert sind. Die Attributwerte der neuen Korrespondenz werden aus den Werten der beiden zu verknüpfenden Korrespondenzen gebildet. Der neue Ähnlichkeitswert ($aggSim$) wird durch Aggregation der Ähnlichkeiten $sim_1$ und $sim_2$, z. B. durch Berechnung des Durchschnitts, bestimmt.

Zudem soll der semantische Typ der beiden Korrespondenzen $(c_1, b)$ und $(b, c_2)$ ($semType_1$ und $semType_2$) kombiniert werden, um den semantischen Typ für die Korrespondenz $(c_1, c_2)$ zu bestimmen (Operation getNewType). Außerdem soll der Status von $(c_1, c_2)$ festgelegt werden (Operation getNewStatus). Um semantische Korrespondenztypen zu kombinieren und den Korrespondenzstatus zu bestimmen,

| *semType2* \ *semType1* | = | < | > | ≈ |
|---|---|---|---|---|
| = | = | < | > | ≈ |
| < | < | < | ≈ | ≈ |
| > | > | ≈ | > | ≈ |
| ≈ | ≈ | ≈ | ≈ | ≈ |

Abbildung 6.3: Regeln zur Kombination semantischer Typen (getNewType) und Bestimmung des neuen Korrespondenzstatus (getNewStatus).

wird eine Menge von Kombinationsregeln vorgeschlagen (siehe Abbildung 6.3). Beispielsweise führt die Kombination von '=' und '<' zu dem neuen semantischen Typ '<'. Die Grundidee ist, dass der semantische Typ mit der niedrigeren Bindungsstärke den neuen semantischen Typ bestimmt. Basierend auf der Definition der *semantischen Relation* in [58] hat '=' eine höhere Bindungsstärke als '<' und '>', welche wiederum stärker sind als '≈'. Die semantischen Typen '<' und '>' haben die gleiche Bindungsstärke, so dass der neue semantische Typ nicht mithilfe von Regeln aus deren Kombination bestimmt werden kann (graue Zellen in Abbildung 6.3). Falls anhand der Regeln der semantische Type '≈' bestimmt wird, wird grundsätzlich der Status *to verify* vergeben, da in diesem Fall zwingend ein Nutzer die Korrespondenz sowie ihren semantischen Typ überprüfen muss. Für alle anderen Kombinationen wird der Korrespondenzstatus auf *handled* gesetzt. Diese Regeln sind ein erster Vorschlag, um während der Mappingadaptierung mit dem Problem der Kombination semantischer Korrespondenzen und Änderungsoperationen umgehen zu können.

**Algorithmus 1:** CompAdapt($OM_{O1,O2}$,$OM_{O1,O1'}$,$OM_{O2,O2'}$)

**Input** : Ontologiemappings $OM_{O1,O2}$, $OM_{O1,O1'}$, $OM_{O2,O2'}$
**Output** : Adaptiertes Ontologiemapping $OM_{O1',O2'}$

1 $OM_{O1',O1} \leftarrow$ `inverse`($OM_{O1,O1'}$);
2 $OM_{O1',O2} \leftarrow$ `compose`($OM_{O1',O1}$,$OM_{O1,O2}$);
3 $OM_{O1',O2'} \leftarrow$ `compose`($OM_{O1',O2}$,$OM_{O2,O2'}$);
4 **return** $OM_{O1',O2'}$;

Algorithmus 1 (CompAdapt) zeigt die kompositionsbasierte Mappingadaptierung für den allgemeineren Fall, dass sich beide Ontologien weiterentwickeln ($O1 \mapsto O1'$, $O2 \mapsto O2'$). Die Eingabe des Algorithmus umfasst das alte Ontologiemapping $OM_{O1,O2}$ sowie die beiden Mappings $OM_{O1,O1'}$ und $OM_{O2,O2'}$. Zunächst wird das inverse Mapping $OM_{O1',O1}$ erstellt (Zeile 1) und mit $OM_{O1,O2}$ kombiniert, so dass zwischenzeitlich ein Mapping von $O1'$ nach $O2$ (Zeile 2) entsteht. Dieses wird weiter mit $OM_{O2,O2'}$ kombiniert, um das adaptierte Mapping $OM_{O1',O2'}$ zwischen $O1'$ und $O2'$ zu erhalten (Zeile 3). Falls nur eine der beiden Ontologien Änderungen unterliegt, muss nur eine der beiden Kompositionen durchgeführt werden. Wenn sich $O1$

zu $O1'$ weiterentwickelt, werden die beiden ersten Schritte ausgeführt. Dementsprechend wird nur `compose` ($OM_{O1,O2}$,$OM_{O2,O2'}$) ausgeführt, falls sich $O2$ zu $O2'$ entwickelt. Die Eingabe des CompAdapt-Algorithmus für das Beispiel in Abbildung 6.2 umfasst das alte Ontologiemapping $OM_{O1,O2}$ und das Ontologiemapping zwischen den $O2$-Versionen ($OM_{O2,O2'}$). $OM_{O1,O2}$ enthält folgende Korrespondenzen:

(*body*, *body*, 1.0, *manual*, = , *handled*),
(*limbs*, *limbs*, 1.0, *manual*, = , *handled*),
(*lower extremities*, *limb segment*, 1.0, *manual*, < , *handled*),
(*upper extremities*, *limb segment*, 1.0, *manual*, < , *handled*),
(*head*, *head*, 1.0, *manual*, = , *handled*),
(*neck*, *neck*, 1.0, *manual*, = , *handled*),
(*tail*, *tail*, 1.0, *manual*, = , *handled*)

und $OM_{O2,O2'}$ umfasst die Korrespondenzen:

(*body*, *body*, 1.0, *manual*, = , *handled*),
(*limbs*, *limbs*, 1.0, *manual*, = , *handled*),
(*limb segment*, *lower limbs*, 1.0, *manual*, > , *handled*),
(*limb segment*, *upper limbs*, 1.0, *manual*, > , *handled*),
(*head*, *head and neck*, 1.0, *manual*, < , *handled*),
(*neck*, *head and neck*, 1.0, *manual*, < , *handled*).

Die Ausgabe des kompositionsbasierten Adaptierungsalgorithmus CompAdapt ist das Mapping $OM_{O1,O2'}$ bestehend aus folgenden Korrespondenzen:

(*body*, *body*, 1.0, *CompAdapt*, = , *handled*),
(*limbs*, *limbs*, 1.0, *CompAdapt*, = , *handled*),
(*lower extremities*, *lower limbs*, 1.0, *CompAdapt*, $\approx$ , *to verify*),
(*upper extremities*, *upper limbs*, 1.0, *CompAdapt*, $\approx$ , *to verify*),
(*lower extremities*, *upper limbs*, 1.0, *CompAdapt*, $\approx$ , *to verify*),
(*upper extremities*, *lower limbs*, 1.0, *CompAdapt*, $\approx$ , *to verify*),
(*head*, *head and neck*, 1.0, *CompAdapt*, < , *handled*),
(*neck*, *head and neck*, 1.0, *CompAdapt*, < , *handled*).

Aus der Komposition der Korrespondenzen über das Konzept '*limb segment*' in der mittleren Ontologie $O2$ resultierenden vier Korrespondenzen (je zwischen '*upper / lower extremities*' und '*upper / lower limbs*'). Für diese Korrespondenzen kann keine automatische Entscheidung zum semantischen Typ getroffen werden (Kombination von '>' und '<'), weshalb sie den Status *to verify* und den semantischen Typ '$\approx$' erhalten und somit manuell überprüft werden sollten. Neben sechs richtigen Korrespondenzen identifiziert CompAdapt also die zwei falschen Korrespondenzen (*lower extremities*, *upper limbs*) und (*upper extremities*, *lower limbs*). Später zeigt sich, dass der alternative Diff-basierte Ansatz in Kapitel 6.4 besser mit derartigen Situationen umgehen kann. Die Korrespondenz (*tail*, *tail*) wird korrekterweise nicht in das adaptierte Mapping aufgenommen, da '*tail*' in $O2'$ nicht mehr existiert und dementsprechend keine Korrespondenz von '*tail*' in $O1$ zu einem Konzept in $O2'$ existiert.

Die Komposition der Mappings allein reicht jedoch nicht aus, um neue Korrespondenzen infolge von Konzepthinzufügungen (z. B. '*trunk*' in $O2'$) zu bestimmen. Um diesem Defizit entgegenzuwirken, wird im Algorithmus CompAdaptMatch zusätzlich ein Match-Schritt angewendet:

**Algorithmus 2:** CompAdaptMatch($OM_{O1,O2}$,$OM_{O1,O1'}$,$OM_{O2,O2'}$,$O1$,$O1'$,$O2$,$O2'$)

**Input** : Ontologiemappings $OM_{O1,O2}$, $OM_{O1,O1'}$, $OM_{O2,O2'}$, Ontologieversionen $O1$,$O1'$,$O2$,$O2'$

**Output** : Adaptiertes Ontologiemapping $OM_{O1',O2'}$

1 $OM_{O1',O2'} \leftarrow$ `CompAdapt(`$OM_{O1,O2}$,$OM_{O1,O1'}$,$OM_{O2,O2'}$`)`;
2 $Add_{O1} \leftarrow O1' \backslash O1$;
3 $Add_{O2} \leftarrow O2' \backslash O2$;
4 $OM_{O1',O2'} \leftarrow OM_{O1',O2'} \bigcup$ `match(`$Add_{O1}$,$O2'$`)` $\bigcup$ `match(`$O1'$,$Add_{O2}$`)`;
5 **return** $OM_{O1',O2'}$;

Nach der Mappingadaptierung (Zeile 1) werden für beide Ontologien hinzugefügte Konzepte ($Add_{O1}$, $Add_{O2}$) identifiziert (Zeile 2–3). Zur Übersicht wird hier die einfache Notation der Mengendifferenz ($O1' \backslash O1$ bzw. $O2' \backslash O2$) verwendet, welche Konzepte der neuen Ontologieversion ($O1'$ bzw. $O2'$) zurückgibt, die nicht in der alten Version ($O1$ bzw. $O2$) enthalten sind. Um neue Korrespondenzen zu identifizieren, werden hinzugefügte Konzepte jeweils mit der anderen neuen Ontologieversion abgeglichen (Zeile 4) und zu dem adaptierten Mapping hinzugefügt. Falls sich nur eine der Ontologien ändert, werden lediglich die neuen Konzepte der geänderten Ontologie mit der anderen Ontologie abgeglichen. Im Beispiel wird das Konzept '*trunk*' zu $O2'$ hinzugefügt, was zu einer neuen, korrekten Korrespondenz (*trunk*, *trunk*) im adaptierten Mapping führt.

## 6.4 Diff-basierte Adaptierung

Die Diff-basierte Adaptierung von Ontologiemappings betrachtet individuelle Ontologieänderungen sowie sogenannte *Change Handler* zur Adaptierung des Ontologiemappings. Dieser modulare Ansatz ermöglicht eine flexible Erfassung unterschiedlicher Änderungstypen und bietet verschiedene automatische oder interaktive Ansätze zur Mappingadaptierung. Beispielsweise würde unter Verwendung der kompositionsbasierten Adaptierung eine Konzeptlöschung zur Löschung aller beeinflussten Korrespondenzen führen. Hingegen kann ein *Change Handler* versuchen die Korrespondenz zu erhalten, indem diese zu einem Konzept in der näheren Umgebung migriert wird. Darüber hinaus können *Change Handler* gegebenenfalls die Verifikation der vorgeschlagenen Mappingänderungen durch einen Experten festlegen.

Zunächst wird die Diff-basierte Mappingadaptierung für den Fall betrachtet, wenn nur eine der beiden Ontologien Änderungen unterliegt (Kapitel 6.4.1). Anschlie-

ßend stellt Kapitel 6.4.2 die verschiedenen *Change Handler* sowie deren Ansätze zur Mappingadaptierung vor. In Kapitel 6.4.3 wird zudem der allgemeine Fall diskutiert, dass beide Ontologien Änderungen unterliegen. Auch wenn der vorgestellte Ansatz für verschiedene Diff-Verfahren anwendbar ist, wird hier die Nutzung des COnto-Diff-Algorithmus [75] angenommen. COnto-Diff eignet sich, um ein Diff-Evolutionsmapping zwischen zwei Versionen zu bestimmen und erzeugt typische Änderungsoperationen wie *merge*, *substitute*, *split*, *addC* oder *delC* (siehe Tabelle 3.1). Für das Beispiel in Abbildung 6.2 enthält das Diff-Evolutionsmapping $diff_{O2,O2'}$ die folgenden Änderungsoperationen:

- Aufspalten eines Konzepts: *split*(*limb segment*,{*lower limbs*, *upper limbs*}),
- Zusammenführen zweier Konzepte: *merge*({*head*, *neck*}, *head and neck*),
- Löschung eines Konzepts: *delC*(*tail*),
- Hinzufügung eines Konzepts: *addC*(*trunk*).

### 6.4.1 Adaptierungsalgorithmus - eine geänderte Ontologie

Die Eingabe des DiffAdapt-Algorithmus (Algorithmus 3) umfasst das zu adaptierende Ontologiemapping ($OM_{O1,O2}$), die zwei Versionen der Quellontologie $O1$ und $O1'$, deren Diff ($diff_{O1,O1'}$) und die Zielontologie $O2$. *Change Handler* werden entsprechend der angegebenen Reihenfolge in der *Change Handler*-Liste $CH$ angewendet.

**Algorithmus 3:** DiffAdapt($OM_{O1,O2}$,$diff_{O1,O1'}$,$O1$,$O1'$,$O2$,$CH$)

**Input** : Ontologiemapping $OM_{O1,O2}$, $diff_{O1,O1'}$, Ontologieversionen $O1$, $O1'$, $O2$, Geordnete *Change Handler*-Liste $CH$
**Output** : Adaptiertes Ontologiemapping $OM_{O1',O2}$

```
1 OM_infl ← getInfluencedCorrs(OM_O1,O2, diff_O1,O1', CH);
2 OM_O1',O2 ← OM_O1,O2 \ OM_infl; //Wiederverwenden des unbeeinflussten
  Mappingteils
3 foreach ch ∈ CH do
4     diffPart_O1,O1' ← diff_O1,O1'.filter (ch.getHandledOperations());
5     ch.handleChg(OM_infl, diffPart_O1,O1', O1, O1', O2);
6 OM_O1',O2 ← OM_O1',O2 ∪ OM_infl;
7 return OM_O1',O2;
```

Zunächst werden alle Korrespondenzen identifiziert, die durch Änderungen im gegebenen Diff beeinflusst werden. Dazu wird für jede Korrespondenz in $OM_{O1,O2}$ geprüft, ob das $O1$-Konzept einer Änderung im $diff_{O1,O1'}$ unterlag (Zeile 1). Dabei werden nur Änderungen berücksichtigt, die in der Liste zu behandelnder Änderungsoperationen vorkommen (in $CH$ spezifiziert). Alle beeinflussten Korrespondenzen in $OM_{infl}$ werden mit dem Status *to verify* initialisiert, da sie eventuell durch Nutzer verifiziert werden müssen. Im Gegensatz dazu, werden unbeeinflusste Korrespondenzen mit dem Status *handled* wiederverwendet, indem diese direkt zum neuen

Mapping $OM_{O1',O2}$ (Zeile 2) hinzugefügt werden. Im Beispiel in Abbildung 6.2 ändern sich die $O2$-Konzepte '*body*' und '*limbs*' nicht, so dass zwei Korrespondenzen direkt wiederverwendet werden können:

(*body*, *body*, 1.0, *DiffAdapt*, = , *handled*),
(*limbs*, *limbs*, 1.0, *DiffAdapt*, = , *handled*).

Korrespondenzen in $OM_{infl}$ können ebenfalls wiederverwendet werden, müssen jedoch aufgrund einer Änderung angepasst werden. Im Beispiel in Abbildung 6.2 umfasst $OM_{infl}$ aufgrund der *split*-, *merge*-, und *delC*-Operationen die Korrespondenzen (*lower extremities*, *limb segment*), (*upper extremities*, *limb segment*), (*head*, *head*), (*neck*, *neck*) und (*tail*, *tail*). Das Mapping $OM_{infl}$ wird anschließend entsprechend der spezifizierten *Change Handler* $ch \in CH$ iterativ adaptiert (Zeile 3–5). Für jeden *Change Handler* werden relevante Änderungsoperationen des Diffs selektiert ($diffPart_{O1,O1'}$). Während der Behandlung der Änderungen (Zeile 5) entfernt jeder *Change Handler* veraltete Korrespondenzen aus $OM_{infl}$ und fügt gegebenenfalls neue Korrespondenzen zu $OM_{infl}$ hinzu. Mögliche Strategien zur Änderungsbehandlung werden anschließend in Kapitel 6.4.2 erläutert. Abhängig von der verwendeten Methode sind die resultierenden Korrespondenzen bereits vollständig bearbeitet (Status *handled*) oder müssen später durch Experten verifiziert werden (Status *to verify*). Abschließend werden die bereits wiederverwendeten Korrespondenzen in $OM_{O1',O2}$ mit dem adaptierten Mappingteil $OM_{infl}$ vereinigt und als Ergebnis zurückgegeben (Zeile 6–7).

Im Beispiel in Abbildung 6.2 wurde die Zielontologie $O2$ geändert. Für diesen Fall muss zunächst das inverse Mapping von $OM_{O1,O2}$ gebildet werden:
$OM_{O2,O1} \leftarrow \texttt{inverse}(OM_{O1,O2})$.
Die DiffAdapt-Eingabe umfasst dann das Ontologiemapping $OM_{O2,O1}$, das Diff-Evolutionsmapping zwischen der alten und neuen $O2$-Version ($diff_{O2,O2'}$), die Versionen der Quellontologien ($O2$, $O2'$), die Zielontologie $O1$ und die *Change Handler*-Liste $CH$. Für diesen Fall ist die DiffAdapt-Ausgabe das adaptierte Ontologiemapping $OM_{O2',O1}$, das erneut invertiert wird, um $OM_{O1,O2'}$ zu erhalten.

### 6.4.2 Behandlung von Änderungen

Um geeignete Ansätze zur Mappingadaptierung realisieren zu können, existiert ein *Change Handler* für jede Art von Ontologieänderung. Die *Change Handler* können leicht angepasst und erweitert werden, um beispielsweise neue Adaptierungsstrategien einzubinden, Nutzer-Feedback anzufordern oder auf neue Änderungsarten zu reagieren. Im Folgenden werden Strategien zur Adaptierung von Ontologiemappings diskutiert. Verschiedene Abbildungen illustrieren wesentliche Adaptierungsstrategien für die Änderungsarten *merge* und *substitute* (Abbildung 6.4), *delC* (Abbildung 6.5) sowie *split* (Abbildung 6.6) und zeigen wie Korrespondenzen (schwarze, gestrichelte Linien) aus $OM_{O1,O2}$ entsprechend der Evolution von $O1$ nach $O1'$ adaptiert werden.

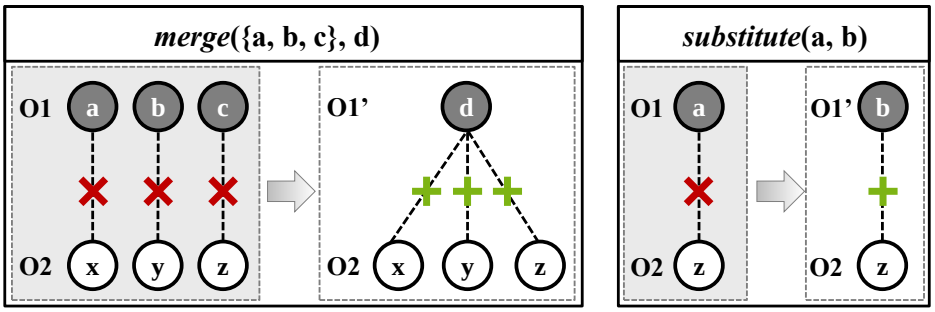



Abbildung 6.4: Adaptierungsstrategien für *merge* und *substitute*.

Die Liste der *Change Handler* für die Ausführung von Diff-Adapt umfasst: $CH_{merge}$, $CH_{delC}$, $CH_{toObsolete}$, $CH_{substitute}$, $CH_{split}$, $CH_{addC}$ und $CH_{revokeObsolete}$. COnto-Diff gewährleistet, dass jedes Konzept einer Ontologie nur an einer der betrachteten Änderungen beteiligt ist, weshalb die Reihenfolge der *Change Handler*-Anwendung hier keine Rolle spielt. Falls andere oder weitere Änderungsoperationen berücksichtigt werden, können Abhängigkeiten zwischen den *Change Handlern* bestehen, so dass ihre Ausführungsreihenfolge eine Rolle spielen kann. Für diesen Fall ist es sinnvoll zunächst Korrespondenzen zu behandeln, die von informationsreduzierenden Änderungsoperationen wie *merge* und *delC* betroffen sind, bevor Handlungen infolge von Informationserweiterungen (z. B. $CH_{split}$, $CH_{addC}$) durchgeführt werden. Anschließend können weitere $CH$ z. B. für Mappinganpassungen infolge einfacher Änderungen (z. B. Attributänderungen) durchgeführt werden.

Abbildung 6.4 (links) zeigt die Adaptierungsstrategie infolge einer *merge*-Operation. Dabei fasst $merge(\{a, b, c\}, d)$ die Quellkonzepte $a, b, c \in O1$ zu einem Zielkonzept $d \in O1'$ zusammen. Der *merge*-Handler ($CH_{merge}$) migriert alle Korrespondenzen, die zu einem der Quellkonzepte der *merge*-Operation assoziiert sind, zum Zielkonzept der *merge*-Operation in $O1'$. Dazu wird jede Korrespondenz aus $OM_{O1,O2}$, die zu einem Quellkonzept einer *merge*-Operation assoziiert ist, aus dem Mapping entfernt (z. B. $(a, x)$). Die jeweilige neue Korrespondenz zum Zielkonzept der *merge*-Operation (z. B. $(d, x)$) wird entsprechend hinzugefügt.

Algorithmus 4 zeigt die vom *merge*-Handler angewendete Strategie. Für jede Korrespondenz *corr* (Zeile 1) und jede *merge*-Operation $\in Merge$ (Zeile 2) prüft der Algorithmus, ob das $O1$-Konzept von *corr* (*corr*.`srcID()`) identisch zu einem der Quellkonzepte $S$ der aktuellen *merge*-Operation ist (Zeile 5–6). Wenn dies zutrifft, wird die beeinflusste Korrespondenz angepasst (Zeile 7-10). Dabei wird das ursprüngliche $O1$-Konzept durch das Zielkonzept der *merge*-Operation $t \in O1'$ ersetzt. Das $O2$-Konzept von *corr* (*corr*.`trgID()`) bleibt unverändert. Abschließend wird die alte Korrespondenz *corr* aus dem Mapping entfernt und die adaptierte Korrespondenz *newCorr* zu $OM$ hinzugefügt (Zeile 11).

**Algorithmus 4:** *merge*-Handler($OM$,$Merge$,$O1$,$O1'$,$O2$)

**Input** : Ontologiemapping $OM$, $diffPart_{O1,O1'}$ $Merge$, Ontologieversionen $O1$, $O1'$, $O2$

```
1  foreach corr ∈ OM do
2      foreach merge ∈ Merge do
3          S ← merge.getSourceIDs();
4          t ← merge.getTargetID();
5          foreach s ∈ S do
6              if s = corr.srcID() then
7                  newType ← getNewType(corr.getType(), <);
8                  newStatus ← getNewStatus(corr.getType(), <);
9                  newCorr ← createCorr(t, corr.trgID(), corr.getSim(),
10                                                          newType, newStatus);
11                 OM.remove(corr).add(newCorr);
```

Zudem unterstützt der *merge*-Handler die Adaptierung des semantischen Typs der hinzugefügten Korrespondenz. Bisher stellt COnto-Diff keine semantischen Typen für Änderungsoperationen zur Verfügung. Dennoch soll während der Kombination von Korrespondenzen ein geeigneter semantischer Typ für jede Änderungsoperation angenommen werden. Beispielsweise gilt für $merge(\{a, b, c\}, d)$ typischerweise, dass die Konzepte $a$, $b$, $c$ weniger allgemein sind ('$<$') als $d$. Somit kann '$<$' mit dem semantischen Typ der alten Korrespondenz ($=, <, >, \approx$) kombiniert werden, um den neuen semantischen Typ zu bestimmen. Der neue semantische Korrespondenztyp (getNewType) und Status der Korrespondenz (Operation getNewStatus) werden anhand der in Kapitel 6.3 beschriebenen Kombinationsregeln ermittelt (siehe Abbildung 6.3).

Im Anatomiebeispiel in Abbildung 6.2 werden '*head*' und '*neck*' zum Konzept '*head and neck*' zusammengefasst (*merge*({*head*, *neck*}, *head and neck*)). Dementsprechend werden alle Korrespondenzen, die zuvor mit '*head*' oder '*neck*' assoziiert waren, dem neuen Konzept '*head and neck*' zugeordnet:

(*head*, *head and neck*, 1.0, *DiffAdapt*, $<$ , *handled*),
(*neck*, *head and neck*, 1.0, *DiffAdapt*, $<$ , *handled*).

Für *substitute*$(a, b)$ wird eine ähnliche Adaptierungsstrategie wie für *merge* angewendet (Abbildung 6.4 (rechts)). Das Konzept $a \in O1$ wird durch $b \in O1'$ ersetzt. Da $a$ Teil einer Korrespondenz zu $z$ in $O2$ ist, wird die Korrespondenz zwischen $a$ und $z$ entfernt und eine neue Korrespondenz von $b$ nach $z$ hinzugefügt. Es wird angenommen, dass der semantische Typ von *substitute* $a = b$ ist. Dieser wird entsprechend der Regeln mit dem ursprünglichen semantischen Typ der Korrespondenz kombiniert.

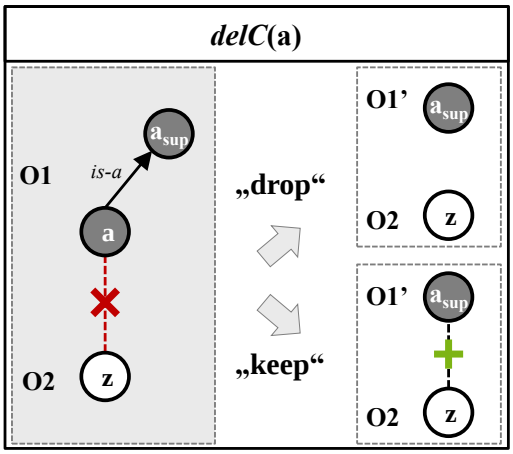



Abbildung 6.5: Adaptierungsstrategien für *delC*.

Im Falle einer Konzeptlöschung werden zwei verschiedene Adaptierungsstrategien betrachtet (siehe Abbildung 6.5). Eine naheliegende Möglichkeit ist die Entfernung aller Korrespondenzen, die gelöschte Konzepte aus $O1$ nutzen („drop“-Strategie). Entsprechend wird infolge von $delC(a)$ in Abbildung 6.5 die Korrespondenz $(a, z)$ gelöscht (siehe „drop“). Anstelle einer vollständigen Löschung, können Korrespondenzen (falls möglich) auf ihre Elternkonzepte übertragen werden („keep“-Strategie). Dementsprechend werden Korrespondenzen entfernt, die das gelöschte Konzept enthalten, jedoch werden neue „allgemeinere“ (*more general*) Korrespondenzen erzeugt. Infolge einer Konzeptlöschung ($delC(a)$) bildet das direkte Elternkonzept des gelöschten Konzepts ($a_{sup}$) das Quellkonzept der neuen Korrespondenz $(a_{sup}, z)$ (siehe „keep“ in Abbildung 6.5). Im Fall von Mehrfachvererbung kann die Korrespondenz auf alle Eltern übertragen werden. Der semantische Typ ergibt sich aus dem semantischen Typ der alten, gelöschten Korrespondenz $(a, z)$ sowie der *is-a*-Beziehung zum Elternkonzept $(a < a_{sup})$ in $O1$. Der Status von Korrespondenzen, die mit der „keep“-Strategie migriert werden, wird grundsätzlich auf *to verify* gesetzt, da ein Nutzer prüfen muss, ob die Vorschläge korrekt sind und eine sinnvolle semantische Anreicherung des Mappings darstellen. Im Beispiel in Abbildung 6.2 würde die „drop“-Strategie aufgrund der Löschung von '*tail*' zur Entfernung der Korrespondenz (*tail*, *tail*) führen. Die „keep“-Strategie führt hingegen zu der neuen Korrespondenz (*tail*, *body*, 1.0, *DiffAdapt*, $<$ , *to verify*). Diese Korrespondenz ist nicht falsch. Allerdings enthält eine „weniger allgemein“-Beziehung eines Konzepts zum Wurzelkonzept der anderen Ontologie keine zusätzliche Information, die das Mapping ausdrucksstärker macht (hier aufgrund des kleinen Beispiels). Die „keep“-Strategie könnte zukünftig erweitert werden, z. B. indem geprüft wird, ob eine *less* / *more general* - Beziehung zu einem Konzept vorgeschlagen wird, das noch keine Äquivalenzbeziehung besitzt. Für Konzepte, die als veraltet eingestuft werden (*toObsolete*-Änderung), werden die gleichen Adaptierungsstrategien wie für Konzeptlöschungen angewendet.

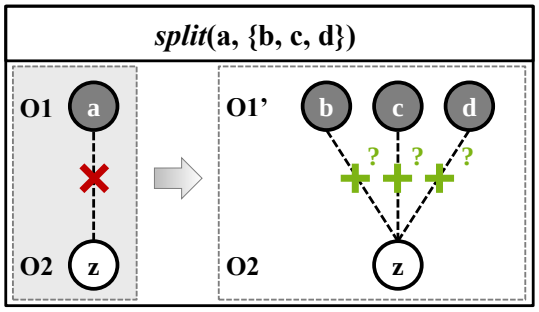



Abbildung 6.6: Adaptierungsstrategie für *split*.

Die Adaptierung von Korrespondenzen, die durch *split* beeinflusst wurden, gestaltet sich komplex (siehe Abbildung 6.6). Beispielsweise löst $split(a, \{b, c, d\})$ die Aufspaltung des Quellkonzepts $a \in O1$ in mehrere Zielkonzepte $b$, $c$, $d \in O1'$ aus. Korrespondenzen, die von einer *split*-Operation betroffen sind, werden entfernt (im Beispiel $(a, z)$). Es werden zwei verschiedene Strategien zur Hinzufügung neuer Korrespondenzen betrachtet. Zunächst können alle möglichen Korrespondenzkombinationen zwischen den aufgespaltenen Zielkonzepten $b$, $c$, $d$ und dem unveränderten Konzept der betroffenen Korrespondenz ($z \in O2$) erstellt werden („take all" - Strategie; in Abbildung 6.6 alle mit + gekennzeichneten Korrespondenzen). Dieses Vorgehen ähnelt einer Komposition der veralteten Korrespondenz mit Korrespondenzen zwischen der alten und neuen Ontologieversion (siehe Kapitel 6.3). Alternativ kann die Adaptierung infolge von *split* die beste(n) Korrespondenz(en) auswählen („take best" - Strategie). Dazu wird für jede von einer *split*-Operation betroffene Korrespondenz (z. B. $(a, z)$) ein lokales Matching der aufgespaltenen Konzepte ($b$, $c$, $d \in O1'$) mit dem unveränderten Konzept der alten Korrespondenz ($z \in O2$) durchgeführt. Anhand der ermittelten Ähnlichkeitswerte wird die beste(n) Korrespondenz(en) mit der höchsten Ähnlichkeit ausgewählt. Beispielsweise könnte dann nur eine, der mit + gekennzeichneten Korrespondenzen in Abbildung 6.6, ausgewählt werden. Für den Abgleich der Konzepte kann eine übliche Match-Strategie verwendet werden.

Die infolge einer *split*-Operation adaptierten Korrespondenzen erhalten einen semantischen Typ entsprechend der Regeln in Abbildung 6.3. Es wird angenommen, dass für *split* der Typ '>' gilt ($a > b$, $a > c$, $a > d$), welcher mit dem semantischen Typ der alten Korrespondenz kombiniert wird. Sämtliche infolge von *split* adaptierten Korrespondenzen sind nur Vorschläge, so dass grundsätzlich der Status *to verify* vergeben wird und Nutzer über die Gültigkeit der Korrespondenzen entscheiden müssen.

Im Beispiel in Abbildung 6.2 wurde '*limb segment*' in '*lower limbs*' und '*upper limbs*' aufgespalten, wovon die zwei Korrespondenzen (*lower extremities*, *limb segment*) und (*upper extremities*, *limb segment*) betroffen sind. Unter Verwendung der „take all" - Strategie, werden dem Nutzer alle vier möglichen Kombinationen zwischen '*lower / upper extremities*' und '*lower / upper limbs*' präsentiert. Im Gegen-

satz dazu kann mithilfe der „take best“ - Strategie für jede betroffene Korrespondenz die korrekt migrierte Korrespondenz identifiziert werden. Beispielsweise führt die Anwendung von *NameSyn*[23] jeweils zu einem Ähnlichkeitswert von ≈0,1 für die falschen Korrespondenzen ((*lower extremities*, *upper limbs*), (*upper extremities*, *lower limbs*)), wohingegen die korrekten Korrespondenzen einen Wert von ≈0,3 erreichen:

(*lower extremities*, *lower limbs*, 0.3, *DiffAdapt*, ≈ , *to verify*),
(*upper extremities*, *upper limbs*, 0.3, *DiffAdapt*, ≈ , *to verify*).

Für alle Konzepthinzufügungen und *revokeObsolete*-Operationen in $O1'$ wird ein automatisches Matching mit der gesamten Zielontologie $O2$ durchgeführt. Es kann eine sehr restriktive Selektion von Korrespondenzen (z. B. *MaxDelta*-Selektion, hoher Ähnlichkeitsgrenzwert) angewendet werden, so dass Nutzer nur die besten Ergebnisse verifizieren müssen und viele falsch positive Korrespondenzen vermieden werden. Alternativ kann die Auswahl der korrekten Korrespondenzen weniger restriktiv sein. Dadurch wird ein hoher Recall erreicht und die Selektion dem Nutzer überlassen. Der Status der neu vorgeschlagenen Korrespondenzen wird auf *to verify* gesetzt. Der semantische Typ wird gegebenenfalls durch das Match-Verfahren festgelegt oder kann zunächst auf '≈' gesetzt und später durch einen Nutzer bestimmt werden. Im Beispiel in Abbildung 6.2 enthält $diff_{O2,O2'}$ die Hinzufügung des Konzepts '*trunk*', das mit $O1$ unter Verwendung des *NameSyn*-Verfahrens abgeglichen wird. Durch eine restriktive Auswahl des besten Ergebnisses wird

(*trunk*, *trunk*, 1.0, *DiffAdapt*, ≈ , *to verify*)

identifiziert, so dass DiffAdapt nach der Anwendung aller *Change Handler* das korrekte und vollständige Ontologiemapping $OM_{O1,O2'}$ für das Beispiel erzeugt.

COnto-Diff identifiziert auch Änderungen von Attributwerten (z. B. *addA*, *delA*) sowie strukturelle Änderungen (z. B. *move*). Eine einfache Strategie derartige Änderungen zu behandeln, ist die Entfernung der beeinflussten Korrespondenzen und ein anschließendes Matching der veränderten Konzepte (aus $O1'$) mit der gesamten neuen Ontologie ($O2'$). Alternativ könnte das Matching nur im lokalen Kontext der veränderten Korrespondenzen stattfinden, um den Suchraum kleiner zu halten und falsch positive Ergebnisse zu vermeiden. Es ist jedoch anzunehmen, dass die Qualität der adaptierten Mappings deutlich sinkt, wenn alle Korrespondenzen, die durch Attribut- oder Strukturänderungen beeinflusst wurden, entfernt und gegebenenfalls durch neue, automatisch generierte Korrespondenzen ersetzt werden. Dementsprechend ist es nicht zu empfehlen, diese Strategien für manuell erstellte Korrespondenzen anzuwenden, so dass mehr verifizierte Korrespondenzen erhalten bleiben. Alternativ könnten die durch Attribut- oder Strukturänderungen beeinflussten Korrespondenzen identifiziert und mit *to verify* markiert werden. Somit können Experten prüfen, ob die Änderungen einen Einfluss auf die Gültigkeit der Korrespondenzen haben und eine Adaptierung notwendig ist.

[23] Trigram-Ähnlichkeit für die Konzeptnamen/-synonyme unter Verwendung von Dice.

### 6.4.3 Adaptierungsalgorithmus - zwei geänderte Ontologien

Falls beide Ontologien Änderungen unterliegen, kann das Ontologiemapping durch zweimalige Anwendung des DiffAdapt-Algorithmus (Algorithmus 3) migriert werden:

**Algorithmus 5:** DiffAdaptBoth($OM_{O1,O2}$,$diff_{O1,O1'}$,$diff_{O2,O2'}$,$O1$,$O1'$,$O2$,$O2'$,$CH$)

**Input** : Ontologiemapping $OM_{O1,O2}$, $diff_{O1,O1'}$, $diff_{O2,O2'}$, Ontologieversionen $O1$, $O1'$, $O2$, $O2'$, geordnete *Change Handler*-Liste $CH$

**Output** : Adaptiertes Ontologiemapping $OM_{O1',O2'}$

1 $OM_{O1',O2} \leftarrow$ `DiffAdapt(`$OM_{O1,O2}, diff_{O1,O1'}, O1, O1', O2, CH$`)`;
2 $OM_{O2,O1'} \leftarrow$ `inverse(`$OM_{O1',O2}$`)`;
3 $OM_{O2',O1'} \leftarrow$ `DiffAdapt(`$OM_{O2,O1'}, diff_{O2,O2'}, O2, O2', O1', CH$`)`;
4 **return** `inverse(`$OM_{O2',O1'}$`)`;

Die Eingabe des DiffAdaptBoth-Algorithmus (Algorithmus 5) ähnelt der Eingabe des DiffAdapt-Algorithmus, jedoch umfasst sie je zwei Versionen für beide Eingabeontologien ($O1/O1'$, $O2/O2'$) sowie zwei Diff-Mappings ($diff_{O1,O1'}$, $diff_{O2,O2'}$). Zunächst wird das gegebene Ontologiemapping bezüglich der Änderungen in der Quellontologie ($O1$) adaptiert, um $OM_{O1',O2}$ zu erhalten. Um das Mapping bezüglich Änderungen in der Zielontologie $O2$ zu adaptieren, wird DiffAdapt ein zweites Mal aufgerufen. Als Eingabe werden das inverse Mapping $OM_{O2,O1'}$, der Diff der Zielontologie $diff_{O2,O2'}$ und die $O1'$-Version (Zeile 3) benötigt. Abschließend wird das Mapping erneut invertiert und zurückgegeben (Zeile 4).

Wenn sich beide Ontologien verändern, kann es Korrespondenzen geben, deren Quell- und Zielkonzepte gleichzeitig Änderungen unterliegen. Wenn beispielsweise beide Konzepte einer Korrespondenz in mehrere Konzepte aufgespalten werden, könnten bei sequentieller, unabhängiger Behandlung falsche Ergebnisse produziert werden. Ein mögliches Szenario ist in Abbildung 6.7 dargestellt. Unter zweimaliger Verwendung der „take all“ - Strategie werden zu viele Korrespondenzen produziert (das „lokale kartesische Produkt“). Im Gegensatz dazu könnte die „take best“ - Strategie zu einer falschen Selektion von (*lower extremities*, *limbs*) im ersten Schritt führen, so dass nach dem zweiten Adaptierungsschritt nur ein unvollständiges Ergebnis (*lower extremities*, *lower limbs*) gefunden wird.

Um mit derartigen Kollisionen umgehen zu können, wird vorgeschlagen, diese Änderungen im Zusammenhang, vor der eigentlichen Adaptierung zu bearbeiten. Zu-

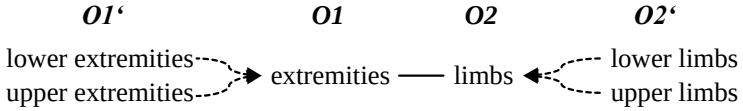



Abbildung 6.7: Beispiel für kollidierende Änderungen, wenn zwei Ontologien Änderungen unterliegen.

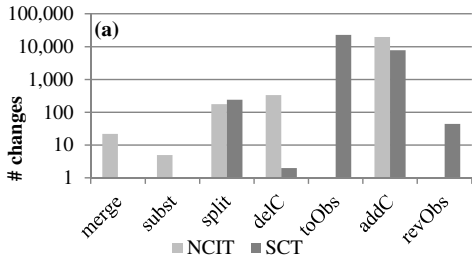
nächst können Korrespondenzen mit kollidierenden Änderungen identifiziert und im Eingabemapping behandelt werden, bevor anschließend DiffAdaptBoth ausgeführt wird. Insbesondere ist es zu empfehlen möglicherweise kritische Änderungskombinationen wie *split–split*, *merge–split* und *substitute–split* zu prüfen und deren Migration in einem Schritt durchzuführen.

## 6.5 Evaluierung

Zur Evaluierung der vorgeschlagenen Mappingadaptierungsverfahren, werden drei sehr große Ontologien aus den Lebenswissenschaften genutzt: *SNOMED-CT* (SCT), *NCI Thesaurus* (NCIT) und *Foundational Model of Anatomy* (FMA). Als integrierte Datenquelle führt der Metathesaurus des UMLS über 100 biomedizinische Ontologien zusammen. Konzepte verschiedener Ontologien werden aufeinander abgebildet und zusammengefasst, so dass in UMLS Mappings zwischen verschiedenen Ontologien verfügbar sind. Die Extraktion der zwei Mappings NCIT–FMA und SCT–NCIT mit je zwei Versionen aus den Jahren 2009 und 2012 erfolgte entsprechend der in [96] verwendeten Methode zur Mappingextraktion aus UMLS. Die Mappingversionen aus 2009 werden mit den beiden Algorithmen adaptiert und somit auf die in 2012 gültigen Ontologieversionen migriert. Die UMLS-Mappings aus 2012 werden als Referenzmappings genutzt, um die Qualität der automatisch adaptierten Mappings bestimmen zu können. Die UMLS-Referenzmappings können jedoch nur als „Silberstandard“ angesehen werden, d. h. sie sind nicht vollständig und werden weiter durch Kuratoren korrigiert. Dabei werden auch Korrespondenzen modifiziert, die keinerlei Ontologieänderungen unterlagen. In dieser Evaluierung werden derartige Korrespondenzen aus den Mappings entfernt, da sich diese nicht infolge von Ontologiemodifikationen geändert haben und dementsprechend nicht während der Adaptierung detektiert werden können. Zur Bewertung der Qualität der adaptierten Mappings bezüglich der neuen Referenzmappings (aus 2012) werden Precision, Recall und F-Measure berechnet. Im Folgenden werden zunächst die verwendeten Datensätze analysiert (Kapitel 6.5.1). Anschließend wird die Qualität der vorgeschlagenen Adaptierungsverfahren evaluiert (Kapitel 6.5.2).

### 6.5.1 Ontologie- und Mappinganalyse

Abbildung 6.8 gibt einen Überblick zur Anzahl der Änderungen in den betrachteten Ontologie- (a) und Mappingversionen (d) sowie zur Versionsgröße der Ontologien (b) und Mappings (c). Zwischen 2009 und 2012 blieb FMA vollständig stabil, wohingegen NCIT und SCT starken Änderungen unterlagen. Neben einigen *merge*-Operationen (22 in NCIT) gab es eine beträchtliche Anzahl (∼180 (240)) *split*-Operationen in NCIT (SCT). In SCT wurden mehr als 22.000 Konzepte auf obsolet gesetzt, wohingegen NCIT zwischen 2009 und 2012 stark erweitert wurde (∼20.000 Konzepthinzufügungen).




| (b) | FMA | NCIT | SCT |
|---|---|---|---|
| $\|C_{2009}\|$ | 62,285 | 63,655 | 310,121 |
| $\|C_{2012}\|$ | 62,285 | 84,132 | 318,502 |

| (c) | NCIT-FMA | SCT-NCIT |
|---|---|---|
| $\|M_{2009}\|$ | 2,294 | 19,971 |
| $\|M_{2012}\|$ | 2,408 | 22,732 |

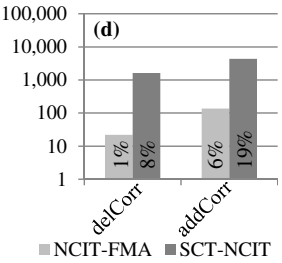



Abbildung 6.8: (a) Ontologieänderungen (b) Ontologiegröße (c) Mappinggröße (d) Mappingänderungen.

Die NCIT–FMA-Mappingversion aus 2009 ist vergleichsweise klein (∼2.300) im Gegensatz zu SCT–NCIT mit circa 20.400 Korrespondenzen (Abbildung 6.8c). In dem betrachteten Zeitraum ist das NCIT–FMA-Mapping um ∼5% und das SCT–NCIT-Mapping sogar um 14% gewachsen. Zudem wurde das SCT–NCIT-Mapping stärker durch Änderungen beeinflusst. 8% der Korrespondenzen wurden aus dem alten Mapping gelöscht und 19% wurden zum neuen Mapping hinzugefügt. Dementsprechend hat NCIT–FMA einen höheren Anteil unveränderter Korrespondenzen und wird dadurch vermutlich leichter als SCT–NCIT zu adaptieren sein.

### 6.5.2 Ergebnisse der Mappingadaptierung

Abbildung 6.9 zeigt die Qualität der adaptierten NCIT–FMA (links) und SCT–NCIT (rechts) Mappings. Um einschätzen zu können, wie hoch der Beitrag der untersuchten Mappingadaptierungsverfahren ist, wird der Anteil unbeeinflusster Korrespondenzen im adaptierten Mapping eingetragen (*Unaff*-Mapping). Die gepunktete und gestrichelte Linie heben jeweils Recall ($Rec_{unaff}$) und F-Measure ($F\text{-}Meas_{unaff}$) des *Unaff*-Mappings hervor. Es werden die Ergebnisse des kompositionsbasierten Ansatzes (CA) und dessen Erweiterung um das Matching neuer Konzepte (CA+m) verglichen. Außerdem wird die Diff-basierte Adaptierung (DA) unter Verwendung der wesentlichen *Change Handler* $CH_{merge}$, $CH_{substitute}$, $CH_{split}$ („take best“), $CH_{delC}$ und $CH_{toObsolete}$ („drop“) sowie deren Erweiterung um $CH_{addC}$ und $CH_{revokeObsolete}$ (DA+C) verwendet. Der Diff-basierte Ansatz ist flexibel und kann leicht um weitere *Change Handler* z.B. für Attribut- und Strukturänderungen erweitert werden. In dieser Evaluierung hatte die einheitliche Behandlung von Attribut- und Strukturänderungen einen negativen Einfluss auf die Mappingqualität, so dass diese *Change Handler* hier nicht eingesetzt werden. In zukünftigen Arbeiten sollte eine differenzierte Behandlung dieser Änderungen durch erweiterte Strategien betrachtet werden.

In den untersuchten Fällen ist die Qualität des *Unaff*-Mappings sehr hoch, da 94% bzw. 80% der NCIT–FMA- bzw. SCT–NCIT-Mappings nicht von Änderungen beeinflusst wurden und direkt wiederverwendet werden können. Für die Adaptierung des

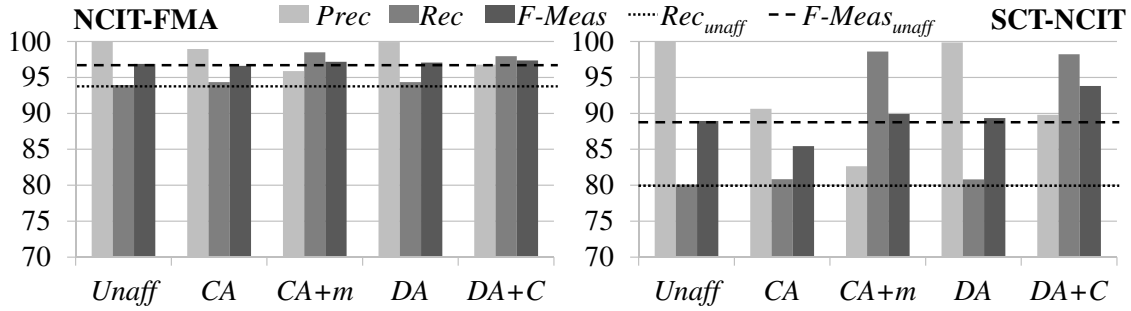



Abbildung 6.9: Ergebnisse der Adaptierung bezüglich der Mappingqualität.

relativ stabilen NCIT–FMA-Mappings produzieren alle betrachteten Ansätze ähnlich gute Ergebnisse mit sehr hohem F-Measure. Die Adaptierung des SCT–NCIT-Mappings ist schwieriger und hilft somit die relative Effektivität der vorgeschlagenen Ansätze besser zu unterscheiden. Verglichen zur Qualität des *Unaff*-Mappings, liefert der kompositionsbasierte Ansatz CA ungenauere Ergebnisse und erhöht den Recall kaum. Dies ist dadurch bedingt, dass CA alle möglichen Kombinationen der existierenden Korrespondenzen einbezieht und keine weitere Selektion stattfindet. Ein zusätzliches Matching hinzugefügter Konzepte (CA+m) führt für SCT–NCIT zu einer signifikanten Verbesserung des Recalls um 18.6%. Typischerweise resultiert ein automatisches Matching in einer Reduktion der Precision, so dass sich insgesamt, verglichen zu *Unaff*, nur ein leicht erhöhtes F-Measure ergibt. CA+m stellt bereits eine gute Strategie dar, um ein konsistentes und möglichst vollständiges Mapping zu erzeugen. Die generierten Korrespondenzen sollten anschließend manuell überprüft werden, um mögliche falsch positive Korrespondenzen zu eliminieren.

Für SCT–NCIT übertreffen die Diff-basierten Ansätze (DA) deutlich die kompositionsbasierten Ansätze. Aufgrund der individuellen Behandlung von Änderungen, kann neben der reinen Wiederverwendung unbeeinflusster Korrespondenzen (*Unaff*), der Recall bei hoher Precision weiter leicht erhöht werden (um ∼0,7%). Dies entspricht ∼160 zusätzlichen korrekten (TP) Korrespondenzen im adaptierten SCT–NCIT-Mapping. Insgesamt produziert DA+C die Mappings mit der höchsten Qualität, da hier zusätzliche *Change Handler* zum Einsatz kommen. Insbesondere findet die Methode weitere korrekte Korrespondenzen für hinzugefügte Konzepte, was zu einer signifikanten Verbesserung des Recalls führt. Der Recall von DA+C und CA+m ist ähnlich, jedoch bleibt die Precision für DA+C wesentlich höher, so dass das F-Measure von DA+C insgesamt besser ist (∼94% anstatt ∼90%). Unter Verwendung eines niedrigeren Schwellwerts als der angewendete strikte Wert von 1.0 könnte der Recall weiter verbessert werden. In diesem Fall können Experten die Auswahl der korrekten Korrespondenzen aus den empfohlenen Korrespondenzen in DA+C übernehmen, wodurch insgesamt eine sehr hohe Qualität des finalen, adaptierten Mappings erreicht werden kann. Falls die UMLS-Referenzmappings unvollständig sind, könnten Korrespondenzen, die derzeit als falsch positiv klassifiziert werden, helfen

die Mappings zu vervollständigen. Beispielsweise produzierte die Anwendung der „keep“- anstelle der „drop“-Strategie für veraltete und gelöschte Konzepte ausschließlich zusätzliche, falsch positive Korrespondenzen bezüglich der verwendeten Referenzmappings. Es ist jedoch zu erwarten, dass mindestens ein Teil der ermittelten semantischen Korrespondenzen korrekt ist. Diese müssen durch Experten verifiziert werden, da im verwendeten Referenzmapping keine semantischen Korrespondenzen z. B. vom Typ '<' oder '>' enthalten sind.

Basierend auf diesen Ergebnissen ist die folgende semi-automatische Adaptierung von Ontologiemappings zu empfehlen:

1. Zunächst soll mithilfe des DA-Ansatzes ein bezüglich der neuen Ontologieversionen konsistentes Mapping ermittelt werden.
2. Anschließend sollen weitere Strategien wie DA+C angewendet werden, um neue Korrespondenzen vorzuschlagen und so ein möglichst vollständiges Mappings zu erhalten.
3. Basierend auf den Adaptierungsergebnissen und Vorschlägen sollen Experten das Mapping weiter vervollständigen und Korrespondenzen mit dem Status *to verify* bestätigen, verwerfen oder anpassen.

## 6.6 Zusammenfassung

Die Weiterentwicklung von Ontologien kann zu ungültigen Ontologiemappings führen. In diesem Kapitel wurden ein kompositions- sowie ein Diff-basierter Algorithmus zur Anpassung von Ontologiemappings infolge von Ontologieevolution vorgestellt. Beide Ansätze können unbeeinflusste Korrespondenzen aus existierenden Mappings wiederverwenden und auf (semi-) automatische Weise nur die Teile anpassen, welche Änderungen unterlagen. Der kompositionsbasierte Ansatz ist konzeptionell einfacher, jedoch reicht er bereits für nur leicht geänderte Ontologien aus. Der Diff-basierte Ansatz ist hingegen mächtiger, da verschiedene änderungsspezifische Ansätze zur Mappingadaptierung unterstützt werden. Es wurden u. a. Strategien zur Mappingadaptierung infolge komplexer Änderungen wie *split* und *merge* vorgestellt. Beide Ansätze erlauben zudem eine Adaptierung des semantischen Typs von Korrespondenzen und unterstützen Experten bei der Verifikation der vorgeschlagenen Korrespondenzen. Die für sehr große, biomedizinische Ontologien durchgeführte Evaluierung bestätigte die Effektivität der präsentierten Ansätze. Beide Verfahren profitieren vom Matching neuer Konzepte, so dass möglichst vollständige Mappings (mit einem verbesserten Recall) produziert werden.

# Teil III

# Evolution von Annotationsmappings

# 7

# Evolution und Qualität von Annotationen

## 7.1 Motivation

Eine der wichtigsten Anwendungen von Ontologien ist die Annotation von Objekten der realen Welt (Instanzen) mit den Konzepten verschiedener Ontologien. Ontologiebasierte Annotationen spielen insbesondere in den Lebenswissenschaften eine wichtige Rolle. Dort werden u. a. Eigenschaften von Genen und Proteinen mit den Konzepten der *Gene Ontology* (GO) beschrieben. Datenquellen wie Swiss-Prot [21] und Ensembl [91] stellen GO-Annotationen zur Verfügung. Die Menge von Annotationen zwischen den Instanzen einer Instanzquelle (z. B. Swiss-Prot) und den Konzepten einer Ontologie (z. B. GO) wird als *Annotationsmapping* bezeichnet. Annotationen werden für verschiedene Anwendungen wie funktionale Analysen von Genexpressionsdaten (z. B. [15, 22]), Analysen biologischer Netzwerke [35] oder instanzbasiertes Ontologie-Matching [104] eingesetzt. Zudem werden existierende Annotationen genutzt, um Vorschläge für neue funktionale Annotationen bisher nicht oder unvollständig annotierter biologischer Objekte zu generieren (z. B. [95, 86, 67]).

Die Ergebnisse dieser Applikationen hängen signifikant davon ab, welche Annotationen verwendet wurden und sind somit auf eine gute Qualität der Annotationen z. B. hinsichtlich ihrer Korrektheit oder Vollständigkeit angewiesen. Eine Bewertung der Annotationsqualität gestaltet sich schwierig, da sie anwendungsabhängig sein kann und letztlich durch einen Nutzer beurteilt werden sollte. Ein wichtiger Qualitätsaspekt ist die Stabilität von Annotationen, da grundlegende Änderungen in Annotationsmappings frühere Ergebnisse beeinflussen oder sogar ungültig machen können. In einer ersten Untersuchung wurde bereits gezeigt, dass sich Annotationen sowie

| Ensembl ID | GO ID | $v_{48}$ | $v_{49}$ | $v_{50}$ | $v_{51}$ | $v_{52}$ |
|---|---|---|---|---|---|---|
| ENSP00000344151 | GO:0015808 (L-alanine transport) | IDA | IDA | IDA | IDA | IDA |
| ENSP00000230480 | GO:0005615 (extracellular space) | TAS | TAS | IDA | TAS | IEA |
| ENSP00000352999 | GO:0006915 (apoptosis) | IDA | - | - | - | IDA |

Abbildung 7.1: Evolution von GO-Annotationen über fünf Ensembl-Versionen ($v_{48} - v_{52}$ = Dez.2007-Dez.2008).

die zugrunde liegenden Ontologien und Instanzen über die Zeit verändern [81]. Neben Annotationshinzufügungen wurden auch zahlreiche Löschungen detektiert. Eine evolutionsbasierte Untersuchung von Annotationen erlaubt eine Bewertung der Annotationsstabilität und unterstützt Nutzer bei der Beurteilung der Glaubwürdigkeit der Annotationen. So ist z. B. eine seit längerem stabile Annotation glaubwürdiger als eine Annotation, die zwischenzeitlich gelöscht wurde. Darüber hinaus gibt die zur Erstellung einer Annotation verwendete Methode Hinweise zur Annotationsqualität. Nutzer können besser einschätzen wie glaubwürdig (oder biologisch fundiert) eine Annotation ist, wenn sie wissen, durch welche Art von Experiment diese bestätigt wurde. Die Relevanz der Erstellungsmethode wird durch die steigende Nutzung sogenannter *Evidence Codes*[24] (EC) bekräftigt. *Evidence Codes* geben Auskunft zur Erstellungsmethode bzw. Herkunft (*engl. provenance, lineage*) [20, 24] von GO-Annotationen. Während einige Annotationen experimentell bestätigt sind, basieren andere auf automatischen Verfahren, die z. B. Sequenzhomologien (z. B. [86, 28]) oder Textextraktionsmethoden (z. B. [36, 67]) einbeziehen. *Evidence Codes* können durch verschiedene Anwendungen genutzt werden, um z. B. auf bestimmte Teilmengen wie nur manuell oder automatisch erstellte Annotationen zu fokussieren.

Abbildung 7.1 veranschaulicht die Evolution einiger GO-Protein-Annotationen über fünf Ensembl-Versionen. Die erste Annotation (ENSP00000344151, GO:0015808) war durchgehend mit dem unveränderten *Evidence Code* IDA (*inferred from direct assay*) verfügbar, d. h. sie ist experimentell bestätigt und stabil. Im Gegensatz dazu wurde die zweite Annotation zu ENSP00000230480 von TAS (*traceable author statement*) über IDA auf IEA (*inferred from electronic annotation*) geändert. Die häufige Überarbeitung der Herkunftsinformation weist auf eine reduzierte Glaubwürdigkeit der Annotation hin. Die dritte Annotation (Zeile 3) ist ebenfalls instabil und weniger glaubwürdig, da sie zwischenzeitlich nicht verfügbar war ($v_{49} - v_{51}$).

Die Evolution, Stabilität und Herkunft von Annotationen wurden in früheren Arbeiten kaum untersucht. Neben den bereits diskutierten verwandten Arbeiten zur Evolution ontologiebasierter Mappings (siehe Kapitel 2.2.2) sind für diese Studie Vorarbeiten zur Datenqualität insbesondere von Annotationen in den Lebenswissenschaften relevant. Allgemein wurden Daten- und Informationsqualität hauptsächlich im Kontext der Datenintegration [136, 154] betrachtet. Im Bereich der Lebenswissen-

[24] http://www.geneontology.org/GO.evidence.shtml

schaften beschäftigten sich einige Arbeiten mit der Qualität von GO-Annotationen u. a. bezüglich *Evidence Codes* [25, 99, 172]. Die Fallstudie in [25] bewertet die Qualität von Annotationen unter Verwendung einiger intuitiv definierter Qualitätskennzahlen für ECs und zeigt deskriptive und vergleichende Statistiken für Annotationen verschiedener Modell-Eukaryoten. Jones et al. [99] untersuchen eine Methode zur Bewertung der Fehlerrate manuell überprüfter Annotationen, die ursprünglich auf Basis von Sequenzähnlichkeiten erstellt wurden (*Evidence Code* ISS = *Inferred from Sequence or Structural Similarity*). Der Ansatz vergleicht ISS-Annotationen mit anderen manuell geprüften Annotationen und detektiert eine verhältnismäßig hohe Fehlerrate für die ISS-Annotationen in der GOSeqLite-Datenbank. In [172] empfehlen die Autoren die Nutzung von ECs als Indikator für die Glaubwürdigkeit von Annotationen. Weiterhin zeigen sie Statistiken zur Verteilung von homologiebasierten, literaturbasierten und anderen Annotationen für verschiedene Spezies. Die bisherigen Ansätze zur Qualität von GO-Annotationen verdeutlichen die Relevanz der Herkunftsinformationen, betrachten jedoch keine historischen Informationen wie beispielsweise die Stabilität von Annotationen.

In dieser Arbeit wird ein generisches Evolutionsmodell vorgestellt, das eine mehrdimensionale Analyse von Annotationsmappings bezüglich der Stabilität und Herkunft der Annotationen erlaubt. Das Modell berücksichtigt Änderungen von Instanzen und Ontologien sowie Modifikationen der Annotationen selbst. Ziel ist es, Nutzer bei einer Qualitätsbewertung bezüglich der Glaubwürdigkeit von Annotationen zu unterstützen. Die hier präsentierten Methoden zur Bewertung von Annotationen sind insbesondere für Anwender und Applikationen im Bereich der Lebenswissenschaften nützlich. So können z. B. Algorithmen Informationen zur Annotationsstabilität ausnutzen, um robustere bzw. glaubwürdigere Ergebnisse zu erhalten.

Das Kapitel umfasst die folgenden Beiträge:

- Das im Grundlagenkapitel 3.1.3 eingeführte Annotationsmodell wird erweitert, so dass verschiedene Qualitätstaxonomien berücksichtigt werden können. Zudem werden verschiedene Maße zur quantitativen Erfassung von Annotationsänderungen vorgestellt (Kapitel 7.2).

- Aufbauend auf dem Modell werden evolutionsbasierte Qualitätsmaße zur Bewertung von Annotationen definiert. Dabei werden die Stabilität und Herkunft von Annotationen ausgenutzt, um die Identifikation glaubwürdiger Annotationen zu unterstützen (Kapitel 7.3).

- Es wird eine vergleichende Evaluierung der Annotationsevolution für die zwei großen Datenquellen Swiss-Prot und Ensembl durchgeführt. Die vorgestellten Maße erlauben die Untersuchung typischer Annotationsänderungen. Zudem können aktuelle Annotationen bezüglich ihrer Stabilität und Herkunft klassifiziert werden (Kapitel 7.4).

## 7.2 Annotationsevolution - Modelle und Maße

Die Stabilität von Annotationsmappings wird durch Änderungen der beteiligten Ontologien und Instanzdatenquellen sowie Änderungen der Assoziationen zwischen Instanzen und Ontologien beeinflusst. Es wird angenommen, dass einzelnen Annotationen verschiedene Merkmale zugeordnet sind, die als Qualitätsindikatoren dienen können. Die assoziierten Werte werden vordefinierten Qualitätstaxonomien entnommen. Zunächst wird das in Kapitel 3.1.3 eingeführte Annotationsmodell erweitert, so dass Qualitätsmerkmale unterstützt werden. Anschließend werden Änderungsoperationen für Annotationen sowie Maße zur Quantifizierung der Annotationsevolution eingeführt.

### 7.2.1 Annotations- und Qualitätsmodell

Entsprechend der in Kapitel 3.1.3 eingeführten Definition verknüpft ein Annotationsmapping eine spezifische Version einer Instanzquelle mit einer spezifischen Version einer Ontologie. Zusätzlich können Annotationsmappings zu Qualitätstaxonomien assoziiert sein, welche die Qualität einzelner Annotationsassoziationen durch verschiedene Kriterien wie z. B. deren Herkunft oder Stabilität spezifizieren. Für diese Studie werden Annotationsmappings wie folgt definiert:

$AM_{IS^w,O^v,Q} = \{(i, c, \{q\}) | i \in IS^w, c \in O^v, q \in Q\}$

Eine Annotation $a = (i, c, \{q\})$ verknüpft eine Instanz $i \in IS^w$ einer Instanzquellversion und ein Konzept $c \in O^v$ einer Ontologieversion. Dabei werden einzelne Annotationen zusätzlich mit einer Menge von Qualitätsmerkmalen $Q = \{Q_1, \ldots, Q_m\}$ beschrieben. Einzelne Qualitätsmerkmale $q$ können numerische Werte annehmen oder aus einer vordefinierten Qualitätstaxonomie entnommen werden. Qualitätstaxonomien repräsentieren vordefinierte Kriterien zur einheitlichen Charakterisierung der Qualität beispielsweise *Evidence Codes* für Herkunftsinformationen oder verschiedene Stabilitätsindikatoren. Für jede Qualitätstaxonomie soll maximal ein Qualitätsindikator einer Annotation $a$ zugeordnet werden. Die Qualitätsbewertung wird einer Annotation üblicherweise bei der initialen Erstellung zugewiesen. Das Beispiel in Abbildung 7.1 zeigt jedoch, dass sich diese Bewertung über die Zeit verändern kann, da z. B. neue Informationen zu einer Annotation verfügbar sind.

Eine Qualitätstaxonomie repräsentiert bestimmte Qualitätskriterien und besteht aus einer Menge vordefinierter Qualitätsterme $\{q_1, \ldots, q_n\}$, welche in einer *is-a*-artigen Hierarchie angeordnet sein können. Für den allgemeinen Fall wird ein Qualitätsterm $q = (q', type)$ mit einem Namen $q$, durch einen Typ *type* und einen optionalen Superterm $q'$ definiert. Jeder Qualitätsterm hat maximal einen Superterm. Ein Qualitätsterm ohne Superterm stellt die Wurzel der Qualitätstaxonomie dar. Qualitätsterme können die verschiedenen Typen „instanziierbar“ oder „abstrakt“ haben.

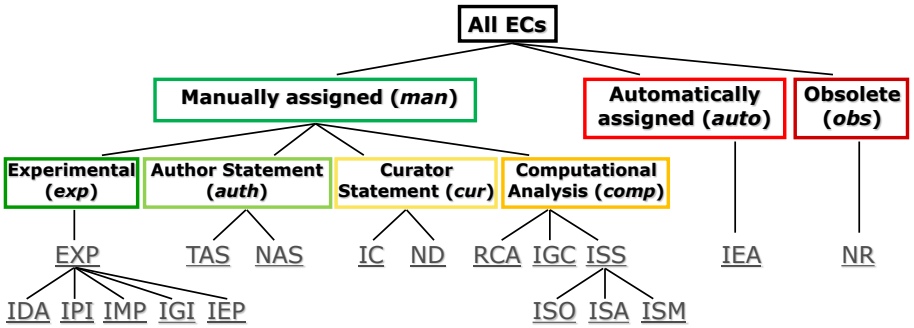



Abbildung 7.2: *Evidence Code*-Taxonomie, Stand Februar 2009.

Instanziierbare Qualitätsterme sind für die Bewertung einer Annotation verfügbar, wohingegen abstrakte Terme instanziierbare Terme aggregieren. Zur Vereinfachung wird für diese Studie angenommen, dass sich Qualitätstaxonomien nicht verändern, auch wenn in der Realität Modifikationen stattfinden können.

Es werden zwei Arten von Qualitätsindikatoren verwendet, um (1) den Herkunftstyp und (2) die Stabilität von Annotationen zu spezifizieren. Für Herkunftsinformationen sollen die existierenden *Evidence Codes* [25] für GO-Annotationen verwendet und analysiert werden. Abbildung 7.2 zeigt die *EC-Qualitätstaxonomie* auf Basis der im Februar 2009 verfügbaren *Evidence Codes*. Die Taxonomie umfasst drei wesentliche Gruppen: manuell erstellte (*'Manually assigned'* - *man*), automatisch erstellte (*'Automatically assigned'* - *auto*) und veraltete (*'Obsolete'* - *obs*) Annotationen. Manuell erstellte Annotationen wurden durch ein Experiment oder einen Experten bestätigt und werden weiter in *exp*, *auth*, *cur* und *comp* unterteilt. Im Gegensatz dazu sind *auto*-Annotationen (*IEA*) nicht verifiziert und basieren auf Algorithmen, welche z. B. Sequenzhomologien oder Swiss-Prot-*Keyword*-Mappings ausnutzen. Für die Bewertung von Annotationen auf Basis ihrer Stabilität werden zunächst numerische Werte bestimmt, die dann auf einzelne Kategorien abgebildet werden, so dass die Nutzung und Evaluierung erleichtert wird. Die hier verwendete Qualitätstaxonomie für die Stabilität besteht aus nur zwei Termen, um stabile (*stable*) von instabilen (*unstable*) Annotationen entsprechend ihrer Historie unterscheiden zu können. Beispielsweise kann eine automatisch generierte, stabile Annotation zwischen einer Instanz $i$ und einem Ontologiekonzept $c$ als $a = (i, c, \{IEA, stable\})$ beschrieben werden. Die eingeführten Qualitätstaxonomien werden in der Evaluierung in Kapitel 7.4 verwendet.

In den Lebenswissenschaften folgt die Versionierung von Annotationsmappings typischerweise dem Versionierungsschema der Instanzdatenquelle, d. h. eine neue Instanzquellversion enthält möglicherweise veränderte Annotationen und kann eine aktuelle oder ältere Ontologieversion referenzieren. Auf der anderen Seite wird

[25] `http://www.geneontology.org/GO.evidence.shtml` Es wurden die im Februar 2009 verfügbaren *Evidence Codes* verwendet. Seitdem hat das GO-Konsortium zusätzliche *Evidence Codes* definiert, jedoch wurde keiner der hier verwendeten *Evidence Codes* entfernt.

eine neue Ontologieversion typischerweise nicht in Zusammenhang mit der neuen Version eines abhängigen Annotationsmappings veröffentlicht. Aufeinanderfolgende Versionen einer Instanzquelle können also auf die gleiche Ontologieversion verweisen. Dementsprechend können Versionen von Annotationsmappings mit $AM^w = AM_{IS^w,O^v,Q}$ abgekürzt werden.

### 7.2.2 Änderungsoperationen

Um eine Untersuchung der Evolution von Annotationen zu ermöglichen, werden verschiedene Änderungsoperationen unterschieden. Grundsätzlich treten Hinzufügungen (*add*), Änderungen (*chg*) und Löschungen (*del*) von Annotationen auf. Annotationsänderungen können durch Änderungen der zugrunde liegenden Ontologien oder Instanzen beeinflusst werden. Insbesondere führt eine Weiterentwicklung von Ontologien und Instanzen zu einer entsprechenden Propagierung der Änderungen auf Annotationen. In dieser ersten Untersuchung zur Annotationsevolution werden zunächst Löschungen sowie *merge*-, *substitute*- und *toObsolete*-Änderungen für Ontologiekonzepte und Instanzen berücksichtigt. Für die Ermittlung von Änderungen in Ontologien kann z. B. COnto-Diff (siehe Kapitel 3.2.2) eingesetzt werden. Um Instanzänderungen zu bestimmen, kann z. B. ein *Change Log* der Instanzdatenquelle genutzt werden. Unter Berücksichtigung der Änderungen in Ontologien und Instanzquellen werden folgende Änderungsoperationen für Annotationen unterschieden:

| | |
|---|---|
| $add$ | Hinzufügung einer neuen Annotation |
| $del_{ont}$ / $del_{ins}$ | Löschung einer Annotation aufgrund der Änderung eines Ontologiekonzepts / Instanzobjekts |
| $del_{ann}$ | Löschung einer existierenden Annotation ohne Konzept- oder Instanzänderung |
| $chg_{ont}$ / $chg_{ins}$ | Veränderung einer Annotation aufgrund der Änderung eines Ontologiekonzepts / Instanzobjekts |
| $chg_{qual}$ | Veränderung eines Qualitätsindikators einer Annotation |

In Kapitel 4.2.2 wurde bereits ein einfacher Mapping-Diff ($mdiff$) zur Bestimmung von Korrespondenzhinzufügungen und -löschungen bezüglich zweier Versionen eines Ontologiemappings erstellt. Ebenso können für zwei Versionen eines Annotationsmappings einfache Annotationshinzufügungen (*add*) und -löschungen (*del*) erfasst werden. Da Annotationsänderungen durch die Evolution der beteiligten Instanzen und Ontologien ausgelöst werden können, sind einige der einfachen Annotationsänderungen möglicherweise auf komplexere Änderungen ($del_{ont}$, $del_{ins}$, $chg_{ont}$, $chg_{ins}$) zurückzuführen. Insbesondere wird für jede Annotationslöschung geprüft, ob die beteiligte Instanz oder das beteiligte Ontologiekonzept einer Änderung unterlag. Falls dies zutrifft, wird die Löschoperation detaillierter als $del_{ont}$ oder $del_{ins}$ beschrieben. Zudem werden Annotationen unabhängig von Änderungen der zugrunde liegenden Ontologien und Instanzen gelöscht ($del_{ann}$), z. B. wenn ein Experte entscheidet, ei-

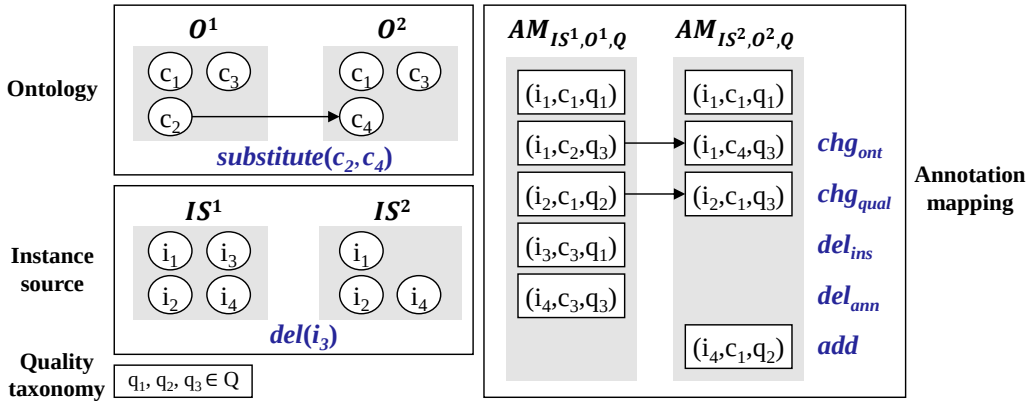



Abbildung 7.3: Beispielszenario zur Annotationsevolution.

ne falsche Annotation zu entfernen. Für alle $del_{ont}$- und $del_{ins}$-Operationen, wird weiter geprüft, ob eine vermeintlich gelöschte Annotation möglicherweise erhalten blieb und z. B. infolge einer *merge*- oder *substitute*-Operation migriert wurde. Falls dies der Fall ist, wird die betroffene Annotationsänderung als $chg_{ont}$ oder $chg_{ins}$ (statt $del_{ont}$ oder $del_{ins}$) bezeichnet. Falls eine $chg_{ont}$- oder $chg_{ins}$-Operation zuvor als einfache Annotationshinzufügung ermittelt wurde, wird die entsprechende *add*-Operation entfernt. Zudem werden $chg_{qual}$-Änderungen erfasst, indem für alle Annotationen der neuen Version geprüft wird, ob diese einer Änderung der Qualitätsindikatoren unterlagen. Dies geschieht z. B., wenn eine automatisch generierte Annotation später experimentell bestätigt wird.

Abbildung 7.3 veranschaulicht einige Annotationsänderungen für einen Evolutionsschritt zwischen zwei Versionen einer Ontologie ($O^1$, $O^2$), einer Instanzqelle ($IS^1$, $IS^2$) und eines Annotationsmappings ($AM_{IS^1,O^1,Q}$, $AM_{IS^2,O^2,Q}$). Zur Übersicht wird auf die Darstellung der Hierarchien für die Ontologie $O$ und Qualitätstaxonomie $Q$ verzichtet. Es werden beispielhaft je eine Ontologie- und Instanzänderung betrachtet. Das Konzept $c_2 \in O^1$ wird durch $c_4 \in O^2$ ersetzt (*substitute*). Diese Ontologieänderung führt zur Anpassung einer Annotation ($chg_{ont}$: $(i_1, c_2, q_3) \rightarrow (i_1, c_4, q_3)$). Zudem ändert sich in der neuen Version die Qualitätsinformation der Annotation $(i_2, c_1)$ von $q_2$ auf $q_3$ ($chg_{qual}$). Die Löschung der Instanz $i_3$ löst eine Annotationslöschung aus ($del_{ins}$: $(i_3, c_3, q_1)$). Eine weitere Annotation $(i_4, c_3, q_3)$ wird gelöscht, ohne dass dies auf eine Ontologie- oder Instanzänderung zurückgeführt werden kann. Zudem wird eine neue Annotation hinzugefügt $(i_4, c_1, q_2)$.

Das vorgestellte Evolutionsmodell für Annotationen soll insbesondere in der Evaluierung (7.4) eingesetzt werden, um u. a. verschiedene Ursachen von Annotationsänderungen (z. B. Annotationslöschung aufgrund einer Instanzänderung) zu beurteilen. Aufbauend auf dem Modell kann die Ermittlung von komplexen Mappingänderungen zukünftig genauer untersucht werden, um z. B. einen ausdrucksstarken Diff zwischen verschiedenen Versionen eines Annotationsmappings zu bestimmen.

### 7.2.3 Maße zur Quantifizierung der Annotationsevolution

Die Annotationsevolution soll bezüglich der zuvor definierten Änderungsoperationen (Kapitel 7.2.2) quantitativ bewertet werden. Dazu werden Annotationsänderungen für einen Versionsübergang $w_i \rightarrow w_j$ ermittelt. Entsprechende Maße erfassen die Anzahl von Annotationshinzufügungen ($Add$), die Anzahl von Annotationslöschungen ($Del_{ont}$, $Del_{ins}$, $Del_{ann}$) sowie die Anzahl von Annotationsänderungen ($Chg_{ont}$, $Chg_{ins}$, $Chg_{qual}$) für den betrachteten Versionsübergang.

Zudem sollen in der Evaluierung einfache Maße wie die Mappinggröße oder das Wachstum eines Annotationsmappings zwischen zwei betrachteten Versionen ($growth$, entsprechend Kapitel 4.3.2) ermittelt werden. Darüber hinaus soll untersucht werden, wie sich Annotationen mit verschiedenen Qualitätsmerkmalen über die Zeit entwickeln. Beispielsweise wird erfasst, welche Qualitätsgruppen (Annotationen mit einem bestimmten Qualitätsterm $q$) starken Änderungen unterliegen oder relativ stabil bleiben. Dazu werden die folgenden quantitativen Maße definiert:

| | |
|---|---|
| $\lvert AM^{w_i} \rvert$ | Anzahl der Annotationen in Version $w_i$ des Annotationsmappings $AM$ |
| $\lvert AM^{w_i,q} \rvert$ | Anzahl der Annotationen mit Qualität $q$ in Version $w_i$ von $AM$ |
| $\frac{\lvert AM^{w_i,q} \rvert}{\lvert AM^{w_i} \rvert}$ | Relativer Anteil der Annotationen mit Qualität $q$ im Vergleich zur Gesamtanzahl der Annotationen in Version $w_i$ |

## 7.3 Bewertung der Annotationsstabilität

Um Nutzer bei der Beurteilung der Glaubwürdigkeit von Annotationen zu unterstützen, soll eine Bewertung von Annotationen bezüglich ihrer Stabilität ermöglicht werden. Zur Ermittlung der Stabilität einzelner Annotationen wird zunächst die Historie einer Annotation $a = (i, c)_n$ einer Version $v_n$ wie folgt definiert:

$$h((i,c)_n) = ((i,c)_0, (i,c)_1, \ldots, (i,c)_n) | 0 \leq j < n : (i,c)_j \rightarrow (i,c)_{j+1}$$

Eine Annotation $(i,c)_{j+1}$ in Version $v_{j+1}$ entwickelte sich aus $(i,c)_j$ der vorherigen Version. Dabei ändert sich $(i,c)$ z. B. aufgrund der Ersetzung einer Instanz oder bleibt unverändert. Falls eine Annotation infolge einer Löschung oder vor ihrer initialen Erstellung nicht in einer Version vorkommt, wird ein Nullwert verwendet. Die Berechnung erfolgt für alle Versionen eines vordefinierten Betrachtungszeitraums $p$ (z. B. im letzten Jahr). Auf Basis der Historie $h$ einer Annotation $a$ können verschiedene Maße zur Evolution von $a$ innerhalb von $p$ definiert werden. Zunächst wird das Alter einer Annotation (bezüglich der Anzahl von Versionen) wie folgt beschrieben:

$$a_{age} = (n - fo) + 1$$

Dabei ist $n$ die Versionsnummer der aktuellen Version ($v_n$). $fo$ ist die Nummer der Version, in der die Annotation $a$ zum ersten Mal innerhalb $p$ auftrat.

Zusätzlich wird die Anzahl der Versionen innerhalb $p$ gezählt, in welchen eine Annotation $a$ vorkam ($a_{present}$). Dieses Maß ignoriert alle Versionen des Annotationsmappings, bevor $a$ initial zum Annotationsmapping hinzugefügt wurde. Auf Basis von $a_{age}$ und $a_{present}$ wird das einfache Maß *Existence Stability* definiert, das die relative Existenz einer einzelnen Annotation $a$ misst:

$$stab_{exis}(a) = a_{present}/a_{age}$$

Dieses Maß ähnelt der in Kapitel 5 definierten *Average Stability* zur Erfassung der durchschnittlichen Änderung der Ähnlichkeitswerte automatisch generierter Korrespondenzen zwischen Ontologiekonzepten. Annotationen weisen keinen Ähnlichkeitswert auf, weshalb hier lediglich die Existenz oder Abwesenheit einer Annotation seit ihrem ersten Auftreten erfasst wird. Zur Evaluierung von Änderungen der Qualität von Annotationen innerhalb $p$ wird eine erweiterte Historie $h_Q$ für eine Annotation bezüglich eines Qualitätsindikators (z. B. Herkunft) verwendet:

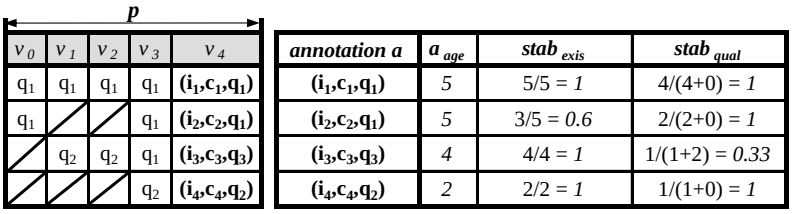
$$h_Q((i,c,q)_n) = ((i,c,q)_0, (i,c,q)_1, \ldots, (i,c,q)_n) | 0 \leq j < n : (i,c,q)_j \rightarrow (i,c,q)_{j+1}$$

Die erweiterte Historie $h_Q$ beinhaltet die Werte der betrachteten Qualitätsindikatoren bezüglich einer Qualitätstaxonomie $Q$. Es ist anzumerken, dass die Betrachtung von Änderungen der Qualitätsmerkmale innerhalb der Annotationshistorie nur für manche Kriterien nützlich ist. Die Evaluierung soll auf Änderungen der Herkunftsinformationen von Annotationen fokussieren, z. B. wenn sich der Evidence Code einer Annotation infolge neuer experimenteller Ergebnisse verändert. Qualitätsänderungen werden gezählt, indem die Anzahl der Versionen in der Historie von $a$ bestimmt wird, für die eine Änderungen der Qualitätsinformation ($a_{changed}$) stattfand. Im Gegensatz dazu spezifiziert $a_{unchanged}$ die Anzahl der Versionen, in denen eine solche Modifikation nicht stattfand. Für den Vergleich der Qualitätsänderungen werden Versionen, in welchen eine Annotation temporär gefehlt hat, nicht betrachtet. Unter Verwendung der Anzahl $a_{unchanged}$ und $a_{changed}$ wird das Stabilitätsmaß *Quality Stability* für eine einzelne Annotation $a$ wie folgt definiert:

$$stab_{qual}(a) = a_{unchanged}/(a_{unchanged} + a_{changed})$$

$stab_{qual}$ bewertet die Häufigkeit von Qualitätsänderungen einer Annotation. Die ermittelten Stabilitätswerte liegen für beide Maße zwischen 0 und 1. Eine perfekte Stabilität von 1 signalisiert, dass eine Annotation seit ihrem ersten Auftreten kontinuierlich vorhanden war (perfekte *Existence Stability*) oder keine Qualitätsänderungen aufweist (perfekte *Quality Stability*). Ein niedriger Wert zeigt hingegen Instabilität an. Die Maße werden in der Evaluierung angewendet, um Annotationen bezüglich ihrer Stabilität zu untersuchen.

Das Beispiel in Abbildung 7.4 veranschaulicht die vorgeschlagenen Maße für vier Beispielannotationen. Es wird ein Beobachtungszeitraum von fünf Versionen ($v_0 \ldots v_4$) betrachtet. Für jede Version ist der Qualitätsterm einer Annotation dargestellt. Durchgestrichene Zellen geben an, dass eine Annotation in den jeweiligen Versionen nicht verfügbar war. Die vier Historien von $(i_1, c_1, q_1)$, $(i_2, c_2, q_1)$, $(i_3, c_3, q_3)$



| p | | | | |
|---|---|---|---|---|
| $v_0$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
| $q_1$ | $q_1$ | $q_1$ | $q_1$ | **($i_1$,$c_1$,$q_1$)** |
| $q_1$ | | | $q_1$ | **($i_2$,$c_2$,$q_1$)** |
| | $q_2$ | $q_2$ | $q_1$ | **($i_3$,$c_3$,$q_3$)** |
| | | | $q_2$ | **($i_4$,$c_4$,$q_2$)** |

| *annotation a* | $a_{age}$ | $stab_{exis}$ | $stab_{qual}$ |
|---|---|---|---|
| **($i_1$,$c_1$,$q_1$)** | *5* | 5/5 = *1* | 4/(4+0) = *1* |
| **($i_2$,$c_2$,$q_1$)** | *5* | 3/5 = *0.6* | 2/(2+0) = *1* |
| **($i_3$,$c_3$,$q_3$)** | *4* | 4/4 = *1* | 1/(1+2) = *0.33* |
| **($i_4$,$c_4$,$q_2$)** | *2* | 2/2 = *1* | 1/(1+0) = *1* |

Abbildung 7.4: Historie und Ergebnisse der Maße für vier Beispielannotationen.

und $(i_4, c_4, q_2)$ in Version $v_4$ weisen unterschiedliche Charakteristika auf. Annotation $(i_1, c_1, q_1)$ wurde in $v_0$ eingeführt ($a_{age} = 5$) und hat eine perfekte Stabilität für $stab_{qual}$ und $stab_{exis}$. Im Gegensatz dazu war die ebenfalls seit fünf Versionen verfügbare Annotation $(i_2, c_2, q_1)$ zwischenzeitlich nicht existent ($v_1$, $v_2$), so dass die *Existence Stability* auf 0,6 reduziert ist. $(i_3, c_3, q_3)$ war durchgängig in vier Versionen von $p$ verfügbar, unterlag jedoch zwei Qualitätsänderungen, so dass ihr $stab_{qual}$-Wert niedrig ist (0,33). Die vierte Annotation $(i_4, c_4, q_2)$ weist eine perfekte Stabilität für beide Maße auf, jedoch ist diese Annotation relativ neu ($a_{age} = 2$), da sie zum ersten Mal in $v_3$ auftrat.

## 7.4 Evaluierung

Die Evaluierung umfasst eine vergleichende Analyse der Evolution von Annotationen in den zwei großen Annotationsdatenquellen Ensembl [91] und Swiss-Prot [21]. Beide Quellen annotieren u. a. Proteine mit Konzepten der *Gene Ontology*. In Kapitel 7.4.1 erfolgt eine Analyse zur Evolution von Annotationen bezüglich verschiedener Herkunftsarten (unter Verwendung von *Evidence Codes*). Zudem wird untersucht, wie Instanz (Protein) - und Ontologieänderungen auf Annotationen propagiert werden. In Kapitel 7.4.2 werden die Stabilitätsindikatoren für Annotationen beider Quellen ausgewertet.

### 7.4.1 Analyse der Annotationsherkunft

Für die Untersuchung werden Versionen der Datenquellen Swiss-Prot (Sp) und Ensembl (E) zwischen März 2004 und Dezember 2008 verwendet. In diesem Zeitraum hat Swiss-Prot (Ensembl) 14 (28) Hauptversionen veröffentlicht (Versionsnummer 43-56 bzw. 25-52). Beide Quellen stellen zahlreiche Annotationen für verschiedene Spezies zur Verfügung. Swiss-Prot umfasst hauptsächlich manuell geprüfte Annotationen während Ensembl auf die automatische Generierung und Integration von Daten fokussiert. Es werden funktionale Annotationen humaner Proteine zu Konzepten der *Gene Ontology* (GO) betrachtet. Für diese Analyse werden die drei GO-Subontologien zu biologischen Prozessen, molekularen Funktionen und zellulären

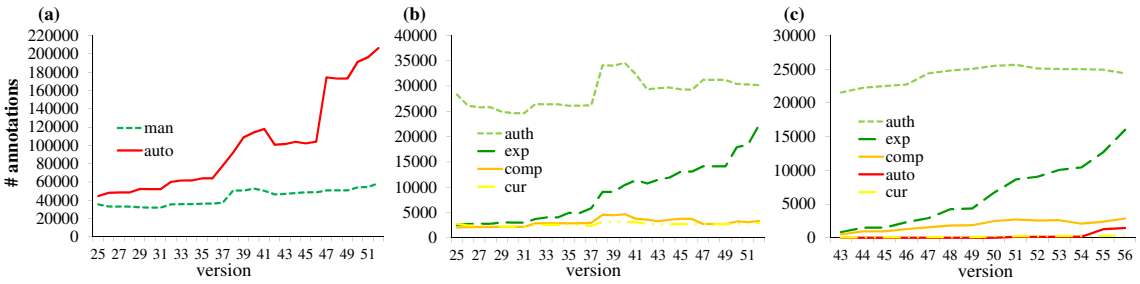



Abbildung 7.5: Evolution von Annotationen in verschiedenen EC-Gruppen
(a) Manuell vs. automatisch erstellte Annotationen (Ensembl),
(b) „Subklassen“ der manuell erstellten Annotationen (Ensembl),
(c) Alle Annotationen (Swiss-Prot).

Komponenten nicht separat betrachtet, d. h. GO wird als eine Ontologie angesehen. Swiss-Prot versucht jeweils aktuelle GO-Versionen zu nutzen, wohingegen Ensembl-Versionen teilweise auf älteren GO-Versionen beruhen.

Abbildung 7.5 zeigt die Entwicklung der Anzahl von GO-Annotationen in den verschiedenen Gruppen der EC-Taxonomie (siehe Abbildung 7.2) für Ensembl und Swiss-Prot. Mit ≈78% von insgesamt 265.000 Annotationen in der letzten Version umfasst Ensembl überwiegend automatisch generierte Annotationen (Abbildung 7.5(a)). Weiterhin gab es ein starkes Wachstum automatisch erstellter Annotationen innerhalb der vier betrachteten Jahre (*growth* =4,6). Zwischen Version 40 und 42 erfolgte zudem eine erhebliche Anzahl Löschungen, so dass die Gesamtanzahl der Annotationen gesunken ist. Die Anzahl manuell erstellter Annotationen ist im Gegenzug nur leicht (um Faktor 1,7) gestiegen. Abbildung 7.5(b) gibt einen detaillierten Einblick in die Entwicklung der manuell erstellten Annotationen in Ensembl. Insbesondere experimentell validierte Annotationen zeigen ein deutliches Wachstum (*growth* =8,9). Auf Aussagen von Autoren basierende Annotationen sind relativ häufig verfügbar jedoch ist ihre Gesamtanzahl nur leicht anstiegen (*growth* =1,1). Im Gegensatz dazu existieren durchgehend verhältnismäßig wenige *cur*- und *comp*-Annotationen.

Abbildung 7.5(c) veranschaulicht die Evolution von Annotationen in Swiss-Prot. Die hauptsächlich manuell gepflegte Datenquelle umfasst in der letzten betrachteten Version ≈45.000 Annotationen. Im Gegensatz zu Ensembl enthält Swiss-Prot nur sehr wenige automatisch erstellte Annotationen (1.440), welche auch erst in den späteren Versionen auftreten. Der Hauptteil von Swiss-Prot umfasst *auth*-Annotationen (≈24.000 in Version 56). Diese Anzahl sinkt leicht seit Version 51. Im Gegenzug ist die Anzahl der *exp*-Annotationen deutlich auf circa 16.000 angestiegen (*growth* =18,5). Insgesamt umfasst Swiss-Prot hauptsächlich manuell erstellte Annotationen, die eine kontinuierlich stabile Evolution ohne nennenswerte Schwankungen aufweisen.

| | | *Add* | *Chg* | | | *Del* | | |
|---|---|---|---|---|---|---|---|---|
| | | | $Chg_{ins}$ | $Chg_{ont}$ | $Chg_{qual}$ | $Del_{ann}$ | $Del_{ins}$ | $Del_{ont}$ |
| **Sp** | **abs. (%)** | 32,613 (53%) | 18,214 (30%) | | | 10,502 (17%) | | |
| | | - | 16,106 | 56 | 2,052 | 8,511 | 1,369 | 622 |
| **E** | **abs. (%)** | 391,771 (60%) | 47,805 (8%) | | | 208,585 (32%) | | |
| | | - | 4,310 | 171 | 43,324 | 145,209 | 60,788 | 2,588 |

Tabelle 7.1: Anzahl (und Prozentsatz) der Änderungsoperationen in Swiss-Prot (Sp) und Ensembl (E), aggregierte Werte über alle Versionen.

Tabelle 7.1 fasst die Anzahl der seit März 2004 vorgenommenen Änderungsoperationen für Swiss-Prot und Ensembl zusammen. Für Ontologien wurden zunächst *delC*-, *merge*-, *substitute*- und *toObsolete*-Änderungen berücksichtigt. Zur Bestimmung von Instanzänderungen wurden die Instanzen verschiedener Versionen einer Quelle basierend auf ihrer *Accession* verglichen. Komplexere Änderungen wie z. B. Ersetzungen (*substitute*) oder das Zusammenfassen (*merge*) von Proteinen wurden mithilfe von Evolutionsinformationen der Anbieter ermittelt. Swiss-Prot stellt die Historie von Proteinen, z. B. Änderungen der *Accession*, über die Weboberfläche zur Verfügung (*Entry History*). Ensembl vermerkt Änderungen wie das Zusammenfassen oder Ersetzen von Proteinen in einem Log.

Die Mehrheit der Annotationsänderungen in Ensembl (60%) und Swiss-Prot (53%) sind Hinzufügungen. Allerdings tritt auch eine überraschend hohe Anzahl Löschungen und Änderungen auf, die vermutlich aus einigen grundlegenden Umstrukturierungen, wie der Einführung neuer *Accessions*, resultieren. In Swiss-Prot sind circa 30% aller Änderungsoperationen Annotationsänderungen (*Chg*), die hauptsächlich durch Veränderungen an Instanzen verursacht wurden ($Chg_{ins}$). Die betroffenen Annotationen bleiben also häufig erhalten, anstatt gelöscht zu werden. Im Gegensatz dazu sind Annotationsänderungen in Ensembl hauptsächlich von Änderungen der Qualitätsterme bzw. *Evidence Codes* ($Chg_{qual}$) betroffen. In beiden Quellen haben Ontologieänderungen einen vergleichsweise geringen Einfluss auf Annotationen. Dies wird u. a. dadurch verursacht, dass Annotationen innerhalb der Instanzquellen verwaltet werden, wohingegen Ontologien unabhängig von den Instanzen entwickelt werden. Die Anzahl der Löschungen ist in beiden Datenquellen nicht unerheblich. Insbesondere in Ensembl sind 32% aller Änderungen Annotationslöschungen.

Als nächstes soll die Verteilung der Änderungsoperationen *Add*, *Chg* und *Del* für verschiedene EC-Gruppen untersucht werden (siehe Tabelle 7.2). In Swiss-Prot ist circa die Hälfte aller hinzugefügten Annotationen experimentell validiert. Ein Drittel der neuen Annotationen sind *auth*-Annotationen. *Chg*- und *Del*-Operationen treten mit jeweils 83% und 70% hauptsächlich für *auth*-Annotationen auf, was auf eine erhöhte Instabilität für Annotationen mit dieser Herkunftsinformationen hinweist. Ensembl führt hingegen hauptsächlich Hinzufügungen und Löschungen automatisch generierter Annotationen durch (jeweils 81% und 75% aller *Add*- und *Del*-Änderungen). Von *Chg*-Änderungen sind im wesentlichen *auto*- und *auth*-

| | Swiss-Prot | | | | | | Ensembl | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Add* | | *Chg* | | *Del* | | *Add* | | *Chg* | | *Del* | |
| *exp* | 15,751 | 48.2% | 1,830 | 10.0% | 1,784 | 17.0% | 25,979 | 6.6% | 5,826 | 12.2% | 7,575 | 3.6% |
| *auth* | 11,307 | 34.6% | 15,177 | 83.3% | 7,350 | 70.0% | 34,046 | 8.7% | 16,381 | 34.3% | 29,148 | 14.0% |
| *cur* | 339 | 1.0% | 65 | 0.4% | 73 | 0.7% | 6,362 | 1.6% | 300 | 0.6% | 6,318 | 3.0% |
| *comp* | 3,730 | 11.4% | 1,107 | 6.1% | 1,214 | 11.6% | 6,734 | 1.7% | 5,720 | 12.0% | 4,362 | 2.1% |
| *auto* | 1,541 | 4.7% | 35 | 0.2% | 81 | 0.8% | 316,979 | 80.9% | 18,344 | 38.4% | 157,632 | 75.6% |
| *obs* | 0 | 0.0% | 0 | 0.0% | 0 | 0.0% | 1,826 | 0.5% | 1,234 | 2.6% | 3,550 | 1.7% |
| **sum** | 32,668 | | 18,214 | | 10,502 | | 391,926 | | 47,805 | | 208,585 | |

Tabelle 7.2: Verteilung der *Add*, *Chg*, *Del*-Operationen in verschiedenen EC-Gruppen in Ensembl und Swiss-Prot.

Annotationen betroffen. Insgesamt tritt die Evolution existierender Annotationen hauptsächlich für *auto*- und *auth*-Annotationen auf.

Im Folgenden sollen die Änderungen der Herkunftsinformationen (EC) detaillierter betrachtet werden. Dies soll Einblick geben, welche neuen *Evidence Codes* ausgewählt werden, um die Beschreibung der Annotationsqualität in neuen Versionen zu verbessern. Die Tabelle 7.3 fasst EC-Änderungen in Swiss-Prot und Ensembl für die betrachteten Versionen zwischen März 2004 und Dezember 2008 zusammen. Die Tabellenzellen geben an, wie viele Annotationen von (*from*, Zeilen) einer EC-Gruppe zu (*to*, Spalten) einer anderen EC-Gruppe geändert wurden. Dabei wurden die EC-Zuordnungen der Annotationen entsprechend der Taxonomie in Abbildung 7.2 zu den abstrakten Gruppen *exp*, *auth*, *cur*, *comp*, *auto* und *obs* aggregiert. Beispielsweise werden Änderungen von ISS nach TAS als „from *comp* to *auth*“ und Änderungen von IPI nach IDA als „from *exp* to *exp*“ zusammengefasst. Annotationsänderungen in Swiss-Prot treten hauptsächlich für *auth* - Annotationen auf (72%). Neu zugewiesene ECs stammen meistens aus der *exp* - Gruppe (66%). Dies zeigt einen Fortschritt in der Entwicklung von Annotationen, da experimentell belegte Annotationen (*exp*) gegenüber jenen, die lediglich auf der Aussage eines Autors beruhen (*auth*), bevorzugt werden. In Ensembl führt der enorme Anteil automatisch generierter Annotationen zu einem etwas anderen Bild. Nur der Anteil neu vergebener *auto* - und *exp*-*Evidence Codes* steigt im Vergleich zu vorher an (jeweils von 13% auf 16% und von 37% auf 43%). Für alle anderen Gruppen, insbesondere für *auth* - Annotationen verringert sich der Anteil infolge von Änderungen der EC-Informationen (von 35%

| Swiss-Prot | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *from / to* | *exp* | *auth* | *cur* | *comp* | *auto* | *obs* | sum | % |
| *exp* | 147 | 24 | 0 | 42 | 1 | 0 | **214** | 10% |
| *auth* | 1,121 | 270 | 34 | 165 | 0 | 0 | **1,590** | 72% |
| *cur* | 7 | 9 | 0 | 3 | 0 | 0 | **19** | 1% |
| *comp* | 160 | 197 | 7 | 0 | 0 | 0 | **364** | 16% |
| *auto* | 16 | 4 | 0 | 1 | 0 | 0 | **21** | 1% |
| *obs* | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0% |
| **sum** | **1,451** | **504** | **41** | **211** | **1** | **0** | **2,208** | |
| **%** | 66% | 23% | 2% | 10% | 0% | 0% | | |

| Ensembl | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *from / to* | *exp* | *auth* | *cur* | *comp* | *auto* | *obs* | sum | % |
| *exp* | 896 | 413 | 11 | 1,259 | 2,966 | 3 | **5,548** | 13% |
| *auth* | 1,592 | 798 | 73 | 1,038 | 11,901 | 23 | **15,425** | 35% |
| *cur* | 21 | 27 | 0 | 16 | 182 | 0 | **246** | 1% |
| *comp* | 1,280 | 1,206 | 26 | 0 | 3,101 | 0 | **5,613** | 13% |
| *auto* | 3,311 | 10,169 | 228 | 2,329 | 0 | 116 | **16,153** | 37% |
| *obs* | 79 | 391 | 9 | 12 | 725 | 0 | **1,216** | 3% |
| **sum** | **7,179** | **13,004** | **347** | **4,654** | **18,875** | **142** | **44,201** | |
| | 16% | 29% | 1% | 11% | 43% | 0% | | |

Tabelle 7.3: *Evidence Code*-Änderungen in Swiss-Prot und Ensembl.

auf 29%). Die meisten EC-Änderungen traten (in beiden Richtungen) zwischen *auto*- und *auth*-Annotationen auf, was auf eine hohe Instabilität dieser EC-Gruppen hinweist.

### 7.4.2 Stabilitätsanalyse

Zusätzlich zu den Herkunfts- bzw. *Evidence Code*-Informationen, soll die Stabilität der Annotationen analysiert werden. Die berechneten Stabilitätswerte werden auf die Kategorien einer einfachen Qualitätstaxonomie abgebildet (Kapitel 7.2.1). Für die Stabilitätskriterien $stab_{exis}$ und $stab_{qual}$ wird ein minimaler Grenzwert von 0,9 für stabile Annotationen (*stable*) angewendet. Annotationen mit niedrigeren Stabilitätswerten gelten als instabil (*unstable*). Folglich muss eine stabile Annotation in mehr als 90% aller Versionen seit ihrem ersten Auftreten enthalten sein. Ebenso gilt, dass sich für maximal 10% der Versionsübergänge die Qualitätsinformation bezüglich des *Evidence Codes* einer Annotation ändern darf. Für eine aggregierte Sicht wird ein kombiniertes Stabilitätsmaß genutzt, das für jede Annotation $a$ den kleineren der beiden Stabilitätswerte auswählt ($stab_{comb}(a) = min(stab_{qual}(a), stab_{exis}(a))$). Eine Annotation mit einem niedrigen $stab_{comb}$-Wert weist also mindestens einen schlechten Stabilitätswert auf. Annotationen mit den *Evidence Codes* NR (*not recorded*) und ND (*no biological data available*) werden hier nicht betrachtet. Zudem soll das Alter einer Annotation berücksichtigt werden, um z. B. eine differenzierte Betrachtung der Stabilität für kürzlich hinzugefügte oder seit langem bestehende Annotationen zu ermöglichen. Seit einem halben Jahr existierende Annotationen gelten als neu (*novel*). Zwischen einem halben und anderthalb Jahren alte Annotationen werden als mittel-alt (*middle*) klassifiziert. Annotationen, die seit mehr als anderthalb Jahren verfügbar sind, gelten als alt (*old*).

Tabelle 7.4 zeigt Klassifikationsergebnisse für beide Datenquellen. Die insgesamt circa 45.000 (263.000) Annotationen in Swiss-Prot (Ensembl) werden bezüglich der drei Kriterien Herkunft (Spalten), Stabilität (Zeilen) und Alter (Zeilengruppen) klassifiziert. Für jede EC-Gruppe wird die Anzahl der stabilen (*stable*) und instabilen (*unstable*) Annotationen angegeben. Graue Zeilen markieren den aggregierten $stab_{comb}$-Wert. Swiss-Prot umfasst anteilig mehr alte Annotationen (72%) als Ensembl (49%). Im Gegenzug führt die Erstellung automatischer Annotationen in Ensembl zu einem relativ hohen Anteil (24%) neuer Annotationen. Trotz des hohen Anteils alter Annotationen werden nur 4% der Swiss-Prot-Annotationen als instabil klassifiziert. In Ensembl sind dagegen 13% der Annotationen instabil (bezüglich $stab_{comb}$). In anderen Worten, Swiss-Prot (Ensembl) umfasst zu 96% (87%) stabile Annotationen.

Bezüglich der Stabilitätskriterien zeigt sich für beide Quellen, dass neuere (*novel*) und mittel-alte (*middle*) Annotationen aufgrund ihrer kurzen Historie nur selten als instabil klassifiziert werden. Daher werden im Folgenden insbesondere ältere Annotationen (*old*) hinsichtlich ihrer Stabilität analysiert. In Swiss-Prot folgt die

| Swiss-Prot | | exp | | auth | | cur | | comp | | auto | | sum | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | stable | unstable | stable | unstable | stable | unstable | stable | unstable | stable | unstable | stable | unstable |
| old | $\|stab_{exis}\|$ | 7,980 | 84 | 22,064 | 169 | 184 | 0 | 1,651 | 16 | 96 | 1 | **31,975** | **270** |
| | $\|stab_{qual}\|$ | 6,965 | 1,099 | 21,913 | 320 | 160 | 24 | 1,599 | 68 | 96 | 1 | **30,733** | **1,512** |
| | $\|stab_{comb}\|$ | 6,905 | 1,159 | 21,760 | 473 | 160 | 24 | 1,589 | 78 | 96 | 1 | **30,510** | **1,735** |
| middle | $\|stab_{exis}\|$ | 2,306 | 0 | 1,107 | 0 | 36 | 0 | 364 | 0 | 35 | 0 | **3,848** | **0** |
| | $\|stab_{qual}\|$ | 2,266 | 40 | 1,101 | 6 | 36 | 0 | 362 | 2 | 35 | 0 | **3,800** | **48** |
| | $\|stab_{comb}\|$ | 2,266 | 40 | 1,101 | 6 | 36 | 0 | 362 | 2 | 35 | 0 | **3,800** | **48** |
| novel | $\|stab_{exis}\|$ | 5,655 | 0 | 1,054 | 0 | 115 | 0 | 845 | 0 | 1,308 | 0 | **8,977** | **0** |
| | $\|stab_{qual}\|$ | 5,637 | 18 | 1,054 | 0 | 115 | 0 | 844 | 1 | 1,308 | 0 | **8,958** | **19** |
| | $\|stab_{comb}\|$ | 5,637 | 18 | 1,054 | 0 | 115 | 0 | 844 | 1 | 1,308 | 0 | **8,958** | **19** |

| Ensembl | | exp | | auth | | cur | | comp | | auto | | sum | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | stable | unstable | stable | unstable | stable | unstable | stable | unstable | stable | unstable | stable | unstable |
| old | $\|stab_{exis}\|$ | 9,473 | 641 | 22,421 | 1,024 | 238 | 15 | 1,715 | 198 | 71,082 | 21,392 | **104,929** | **23,270** |
| | $\|stab_{qual}\|$ | 8,774 | 1,340 | 20,488 | 2,957 | 190 | 63 | 1,170 | 743 | 89,115 | 3,359 | **119,737** | **8,462** |
| | $\|stab_{comb}\|$ | 8,415 | 1,699 | 19,700 | 3,745 | 184 | 69 | 1,079 | 834 | 68,440 | 24,034 | **97,818** | **30,381** |
| middle | $\|stab_{exis}\|$ | 3,378 | 9 | 4,244 | 9 | 67 | 0 | 470 | 7 | 62,136 | 1,818 | **70,295** | **1,843** |
| | $\|stab_{qual}\|$ | 3,062 | 325 | 3,949 | 124 | 60 | 7 | 354 | 303 | 63,245 | 709 | **70,670** | **1,468** |
| | $\|stab_{comb}\|$ | 3,057 | 330 | 3,942 | 311 | 60 | 7 | 353 | 124 | 61,442 | 2,512 | **68,854** | **3,284** |
| novel | $\|stab_{exis}\|$ | 8,808 | 0 | 2,492 | 0 | 157 | 0 | 942 | 0 | 49,909 | 0 | **62,308** | **0** |
| | $\|stab_{qual}\|$ | 8,650 | 215 | 2,425 | 35 | 149 | 8 | 885 | 32 | 49,608 | 301 | **61,717** | **591** |
| | $\|stab_{comb}\|$ | 8,650 | 158 | 2,425 | 67 | 149 | 8 | 885 | 57 | 49,608 | 301 | **61,717** | **591** |

Tabelle 7.4: Klassifikation der Annotationen in Swiss-Prot und Ensembl bezüglich Herkunft, Stabilität und Alter; $stab > 0{,}9$ (weiß), $stab \leq 0{,}9$ (grau).

Mehrheit der instabilen Annotationen aus EC-Änderungen ($|stab_{qual}| \approx 1.500$), wohingegen nur wenige Annotationen eine reduzierte *Existence Stability* aufweisen ($|stab_{exis}| = 270$). Bezüglich $stab_{exis}$ stammen die meisten Annotationen aus der *auth*-Gruppe. Insgesamt sind jedoch die meisten instabilen Annotationen in Swiss-Prot vom Typ *exp*. Dies stimmt mit der vorherigen Beobachtung zu EC-Änderungen überein, dass die *Evidence Codes* der Swiss-Prot-Annotationen hauptsächlich zu „experimentell validiert“ (*exp*) verändert wurden. Derartige Instabilitäten stellen eine Verbesserung der Herkunftsinformationen dar, so dass betroffene Annotationen an Glaubwürdigkeit hinzugewinnen. In Ensembl werden Instabilitäten hauptsächlich durch die temporäre Abwesenheit von Annotationen verursacht ($|stab_{exis}| \approx 23.000$). Die Mehrheit der instabilen Annotationen (bezüglich $stab_{comb}$) tritt für automatisch generierte (*auto*) Annotationen (79%, $\approx 24.000$) sowie *auth*-Annotationen (12%, $\approx 3.700$) auf, wodurch die zuvor bereits beobachtete, hohe Instabilität dieser EC-Gruppen bestätigt wird.

Der hier vorgestellte Ansatz zur Bewertung von Annotationen ist insbesondere für Datenquellen wie Ensembl hilfreich, die sehr viele nicht verifizierte, automatisch generierte Annotationen enthalten. Der Ansatz erlaubt die Identifikation glaubwürdiger bzw. weniger glaubwürdiger Annotationen anhand verschiedener Kriterien wie der Herkunft oder Stabilität der Annotationen. Die verwendeten Maße $stab_{exis}$ und $stab_{qual}$ stellen orthogonale Methoden dar, so dass Annotationen bezüglich der Kriterien unterschiedlich klassifiziert werden können. Nutzer können Submengen der Annotationen extrahieren, indem z. B. nur experimentell validierte und bezüglich $stab_{exis}$ und $stab_{qual}$ stabile Annotationen ausgewählt werden. Beispielsweise kann

eine Annotation als glaubwürdig gelten, wenn diese insgesamt stabil (bezüglich $stab_{comb}$), seit längerem verfügbar (*middle* oder *old*) und manuell erstellt oder geprüft (*exp*, *auth*, *cur*, *comp*) wurde. Diese drei Kriterien treffen auf 34.179 (36.790) Annotationen, d. h. 76% (14%) aller in Swiss-Prot (Ensembl) verfügbaren Annotationen zu. Die Auswahl der Qualitätskriterien und Grenzwerte (z. B. für die Stabilität) hängen stark von der betrachteten Applikation ab. Nutzer sollten auch an neuen und instabilen Annotationen interessiert sein, da diese dank eines hohen Forschungsinteresses häufig überarbeitet werden. Dieser letzte Aspekt unterstreicht, dass Instabilität nicht notwendigerweise eine negative Eigenschaft ist, sondern auch interessante Objekte oder biologische Erkenntnisse hervorheben kann. Umgekehrt könnte eine hohe Stabilität für Objekte beobachtet werden, die derzeit von eher geringem Interesse sind. Die vorgestellte Evaluierungsmethode erlaubt die Auswahl von stabilen oder instabilen Annotationen und kann somit die Anforderungen unterschiedlicher Anwendungen und Nutzungsszenarien erfüllen.

## 7.5 Zusammenfassung

In diesem Kapitel wurde ein generischer Ansatz zur evolutionsbasierten Bewertung der Qualität von Annotationen vorgestellt. Der Ansatz berücksichtigt verschiedene Annotationsänderungen, u. a. infolge von Instanz- und Ontologieänderungen. Zudem unterstützen verschiedene Bewertungskriterien eine Beurteilung der Qualität von Annotationen hinsichtlich ihrer Glaubwürdigkeit. Um Aussagen über die Herkunft bzw. Erstellungsmethode von Annotationen zu treffen, wurden existierende Informationen zu *Evidence Codes* ausgenutzt. Es wurden zwei Stabilitätsmaße vorgeschlagen, die das zwischenzeitliche Fehlen bereits existierender Annotationen sowie Änderungen der Herkunftsinformationen einbeziehen. In einer vergleichenden Evaluierung wurden funktionale Annotationen zweier großer Datenquellen aus den Lebenswissenschaften (Swiss-Prot und Ensembl) untersucht. Dabei konnte beobachtet werden, dass die meisten Annotationsänderungen aus Instanzänderungen sowie Veränderungen der *Evidence Codes* resultieren. Ontologieänderungen hatten in den betrachteten Quellen einen geringeren Einfluss auf existierende Annotationen. Darüber hinaus zeigte sich, dass die *Evidence Code*-Informationen existierender Annotationen teilweise häufig aktualisiert werden. Insgesamt weisen insbesondere automatisch generierte sowie auf Autorenaussagen basierende Annotationen deutliche Instabilitäten auf. Einige Annotationen werden mit der Zeit experimentell bestätigt und dementsprechend angepasst, so dass ein Teil der Instabilitäten von Annotationen auf eine positive Weiterentwicklung zurückzuführen ist. Der Ansatz kann für verschiedene Anwendungsfälle genutzt werden, beispielsweise um Annotationen als Eingabe von Analysen und anderen Applikationen anhand der verschiedenen Kriterien zu filtern. Zudem können Konsortien die Datenqualität erhöhen, indem Annotationen anhand der Maße gefiltert werden, z. B. bevor diese automatisch aus anderen Datenquellen integriert werden.

# 8

# Einfluss der Ontologieevolution auf funktionale Analysen

## 8.1 Motivation

Ontologien sowie ontologiebasierte Annotationen werden in den Lebenswissenschaften sehr häufig genutzt. Eine weit verbreitete Anwendung sind sogenannte funktionale Analysen (*engl. functional analysis/profiling, term enrichment analysis*) sehr großer Datensätze wie z. B. Genexpressionsdaten. Ziel ist es, signifikant überrepräsentierte Eigenschaften beispielsweise typische Funktionen einer gewissen Gruppe von Genen oder Genprodukten zu bestimmen (siehe Übersichtsartikel [89, 176]). Zur Bestimmung der Eigenschaften werden zumeist die Gene Ontology (GO) sowie Annotationen zwischen der GO und den betrachteten biologischen Objekten genutzt. Wie die meisten biomedizinischen Ontologien bildet die GO mit ihren Konzepten und der *is-a*-Hierarchie einen gerichteten, azyklischen Graphen (*engl. directed acyclic graph*, DAG). Konzepte auf höheren Ebenen der GO repräsentieren abstraktere Beschreibungen und umfassen die Semantik ihrer Kinderkonzepte. GO-Konzepte werden genutzt, um Gene und Genprodukte semantisch zu beschreiben [172]. Dabei wird ein bestimmtes Gen durch seine direkt annotierten Konzepte sowie durch deren Elternkonzepte beschrieben (indirekte Zuordnung aufgrund der *is-a*-Hierarchie). Funktionale Analysen nutzen diese Eigenschaft, um für eine Menge von Genen allgemeinere Kategorien zu identifizieren, die signifikant über- oder unterrepräsentiert sind im Vergleich zu den annotierten Konzepten einer Hintergrundmenge (z.B. alle Gene einer Spezies).

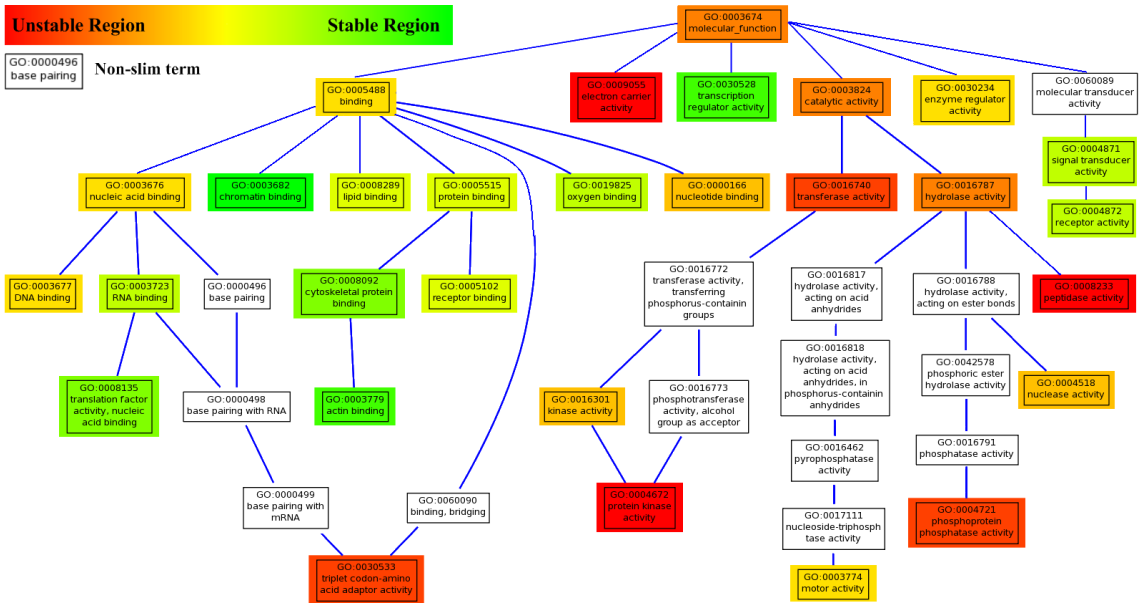



Abbildung 8.1: Evolution einiger Slim-Terme für GO-MF zwischen 2007 und 2010. Die Farben geben die Evolutionsintensitäten der Slim-Terme an. Weiße Konzepte liegen auf dem Weg zur Wurzel, sind jedoch keine Slim-Terme. Tabelle A.2 im Anhang zeigt eine Liste der GO-MF-Slim-Terme mit ihren Evolutionsintensitäten. Die Grafik wurde mit Hilfe des AmiGO [26] Visualisierungswerkzeugs[27]erstellt.

Die GO hat sich seit ihrer Einführung im Jahr 2000 erheblich weiterentwickelt. Dabei änderten sich die drei Subontologien Biologische Prozesse (BP), Molekulare Funktionen (MF) und Zelluläre Komponenten (CC) unterschiedlich. Zwischen 2007 und 2010 ist die BP-Subontologie um circa 70% gewachsen, wohingegen sich CC und MF um jeweils $\approx 40\%$ bzw. $\approx 20\%$ vergrößerten. Tabelle A.1 im Anhang gibt einen Überblick zur Anzahl der Konzepte und Beziehungen für die drei GO-Subontologien zwischen 2007 und 2010. Zudem soll die Anwendung des in [76] vorgestellten Regionenalgorithmus zeigen wie stark sich verschiedene Teile der GO weiterentwickeln. Dabei werden Evolutionsintensitäten innerhalb der Ontologiehierarchie propagiert und über einen gewissen Zeitraum aggregiert. Abbildung 8.1 veranschaulicht anhand der sogenannten GO-Slim-Terme[28], dass sich verschiedene Teilbereiche der GO-MF unterschiedlich stark entwickeln. GO-Slim-Terme bilden eine Submenge der GO-Konzepte und geben einen groben Ontologieüberblick ohne die Detailinformationen sehr fein-granularer Konzepte. Grün dargestellte Regionen sind sehr stabil (der Subgraph eines grünen Slim-Terms unterlag kaum Änderungen), wohingegen rote Konzepte instabile Regionen markieren (der Subgraph eines roten Slim-Terms unterlag zahlreichen Änderungen). Die Analyse der Evolutionsintensitäten verdeutlicht, dass Änderungen sehr ungleich in einer Ontologie verteilt sein können. Im

[27] `http://amigo.geneontology.org/cgi-bin/amigo/visualize?mode=client`
[28] `http://www.geneontology.org/GO.slims.shtml`

vorherigen Kapitel 7 wurde zudem gezeigt, dass GO-basierte Annotationen regelmäßig weiterentwickelt werden, u.a. um neues Wissen einzuarbeiten oder mögliche Inkonsistenzen durch Änderungen der zugrunde liegenden Instanzen und Ontologien zu beheben.

Bisherige Arbeiten zur Evolution von Ontologien (siehe Kapitel 2) und ontologiebasierten Annotationen (Kapitel 7) vernachlässigen den potenziellen Einfluss der Evolution auf Anwendungen, die Ontologien und Annotationen nutzen. So können Änderungen der verwendeten Ontologien und Annotationen Auswirkungen auf die Ergebnisse funktionaler Analysen haben. Es ist relativ naheliegend, dass der hohe Änderungsanteil der GO und GO-Annotationen die Analyseergebnisse beeinflusst. Allerdings ist nicht bekannt, ob die Erkenntnisse früherer Experimente signifikant verändert oder sogar ungültig werden können. Der Einfluss von Ontologie- und Annotationsänderungen auf funktionale Analysen hängt davon ab, wo die Änderungen in der Ontologie lokalisiert sind und welche Arten von Änderungen überwiegen. Beispielsweise könnten Hinzufügungen von Konzepten auf der Blattebene weniger kritisch sein als größere strukturelle Überarbeitungen innerhalb der Ontologie.

In diesem Kapitel soll untersucht werden, zu welchem Grad Änderungen der GO und GO-Annotationen Einfluss auf funktionale Analysen haben und deren Ergebnisse verändern. Es werden verschiedene Methoden vorgestellt, um die Stabilität der Ergebnisse funktionaler Analysen zu ermitteln. Um die Anwendbarkeit und Nützlichkeit des vorgestellten Ansatzes zu zeigen, werden zwei reale experimentelle Datensätze sowie 50 zufällig generierte Datensätze analysiert. Für die experimentellen Datensätze wird eine detaillierte Untersuchung der zugrunde liegenden Änderungen sowie deren Einfluss auf die Analyseergebnisse durchgeführt. Die vorgestellten Methoden und Analyseergebnisse sind insbesondere hilfreich für Ontologieentwickler und -kuratoren sowie für Anwender der funktionalen Analysemethoden.

## 8.2 Methoden

### 8.2.1 Ontologie- und Annotationsmodell

Das verwendete Modell beruht auf dem in Kapitel 3.1 eingeführten Modell für Versionen von Ontologien und Annotationsmappings. Ein Annotationsmapping bzw. die Menge der betrachteten Annotationen wird hier als $A$ bezeichnet. Im Rahmen dieser Studie sollen die GO [55] sowie Annotationen der *Gene Ontology Annotation* (GOA) [10] verwendet werden. Ein Konzept $c$ mit einer eindeutigen *Accession* (z.B. *GO:0007596* für "blood coagulation") kann zu verschiedenen biologischen Objekten assoziiert sein. Algorithmen zur funktionalen Analyse propagieren typischerweise alle Annotationen $A$ aufwärts entlang der *is-a*-Hierarchie im DAG, d.h. die Wurzel hat indirekte Assoziationen zu sämtlichen annotierten Genen. Die Menge

aller Annotationen eines Konzepts ($A(c)$) umfasst somit alle direkten Annotationen zu $c$ sowie alle indirekten Annotationen seiner Kinderkonzepte (Subgraph von $c$). Annotationen sind typischerweise *Evidence Codes* zugeordnet. Diese können genutzt werden, um die Herkunft und wahrscheinliche Qualität einer Annotation zu bewerten (z. B. experimentell verifiziert oder automatisch generiert, siehe Kapitel 7).

Diese Studie erfordert die Betrachtung verschiedener Versionen von Ontologien und Annotationen. Die Notation der Versionen von Annotationsmappings $A^v$ orientiert sich am Veröffentlichungsdatum $v$ der Ontologieversion $O^v$. Dies ist möglich, da GOA immer auf der aktuellen GO-Version basiert. Die Evolution von Instanzen (biologischen Objekten) wird in dieser Studie nicht untersucht, spiegelt sich jedoch indirekt in der Evolution der Annotationen wider (siehe Kapitel 7). Zur Identifizierung von Änderungen (Diff-Bestimmung) zwischen Ontologieversionen wird der COnto-Diff-Algorithmus [75] verwendet. Dabei werden insbesondere *addC*, *toObsolete*, *merge*, *split*, *substitute*, *addR*, *delR*, *move* berücksichtigt (siehe Kapitel 3.2.2).

### 8.2.2 Funktionale Analysen unter Verwendung von FUNC

Funktionale Analysen stellen eine typische Methode zur Untersuchung der semantischen Eigenschaften biologischer Datensätze wie z. B. Gendatensätze dar [89, 176]. Abbildung 8.2 zeigt den typischen Aufbau eines funktionalen Analyseexperiments. Die Eingabe umfasst eine Menge relevanter Gene (*Gene Set*) sowie eine Version einer Ontologie ($O^v$) und eines Annotationsmappings ($A^v$). Die Relevanz von Genen wurde z. B. bezüglich einer bestimmten biologischen Eigenschaft in einem Genexpressionsexperiment ermittelt. Dabei wird jedem Gen ein Wert zugeordnet (z. B. das Expressionslevel), der von einem statistischen Test während der funktionalen Analyse genutzt wird. Die funktionale Analyse produziert ein experimentelles Ergebnis $ER^v$, dass auf $O^v$ und $A^v$ beruht. $ER^v$ beinhaltet eine Menge statistisch signifikanter Konzepte aus $O^v$. Offensichtlich werden Änderungen von $O^v$ und $A^v$ Auswirkungen auf $ER^v$ haben. Die Änderungen können beispielsweise die statistische Signifikanz erhöhen, indem z. B. Konzepte, die zu Genen mit ähnlichen Merkmalen assoziiert sind, zusammengelegt werden. Die zugrunde liegende Ontologiestruktur erleichtert die funktionelle Gruppierung der Geneigenschaften, da Beziehungen zwischen den Konzepten ausgenutzt werden können, um alle Annotationen eines Kinderkonzepts zu dessen Elternkonzepten zu propagieren. Im Rahmen dieser Untersuchung wird das frei zur Verfügung stehende Programm FUNC [150] verwendet, um funktionale

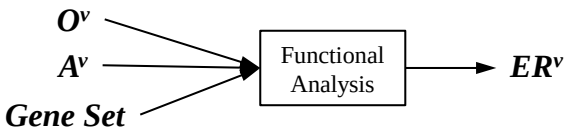



Abbildung 8.2: Generelles experimentelles Design einer funktionalen Analyse.

Analysen durchführen zu können. Dabei werden keine Änderungen an FUNC vorgenommen und stets die gleichen Parameterkonfigurationen eingesetzt. FUNC testet jedes Konzept einer Eingabeontologie auf seine statistische Signifikanz bezüglich der untersuchten Gen- und Annotationsdatensätze. Zur Korrektur führt FUNC multiples Testen auf Basis von Randomisierungen durch. FUNC bietet verschiedene statistische Tests (z. B. Binomialtest, Wilcoxon-Rang-Test) an. Das Ergebnis eines Tests ist eine Liste von Konzepten, denen jeweils ein p-Wert zugeordnet ist. Für diese Untersuchung wird eine Korrektur der p-Werte durch Verwendung der *Family Wise Error*-Rate in FUNC durchgeführt (für Details siehe [150]). Der korrigierte p-Wert wird verwendet, um anhand eines Grenzwerts (z. B. $\alpha = 0,05$) statistisch signifikante Konzepte auszuwählen.

### 8.2.3 Stabilitätsmaße

Um den Einfluss von Ontologie- und Annotationsänderungen auf die experimentellen Ergebnisse funktionaler Analysen bewerten zu können, werden zwei Stabilitätsmaße vorgeschlagen. Dazu werden die Ergebnismengen für eine feste Eingabemenge von Genen zu unterschiedlichen Zeitpunkten, d.h. unter Verwendung verschiedener Ontologie- und Annotationsversionen, berechnet.

Für den Vergleich zweier Ergebnismengen $ER^i$ und $ER^j$ werden folgende Mengenoperationen verwendet. Dabei basieren Ergebnisse in $ER^i$ und $ER^j$ jeweils auf unterschiedlichen Ontologie- und Annotationsversionen.

$|ER^i|$, $|ER^j|$ – *Anzahl der Konzepte in* $ER^i$ *und* $ER^j$
$|ER^i \cap ER^j|$ – *Anzahl überlappender Konzepte zwischen* $ER^i$ *und* $ER^j$
$|ER^i \setminus ER^j|$ – *Anzahl der Konzepte, die nur in* $ER^i$ *aber nicht in* $ER^j$ *auftauchen*
$|ER^j \setminus ER^i|$ – *Anzahl der Konzepte, die nur in* $ER^j$ *aber nicht in* $ER^i$ *auftauchen*

Konzepte gelten als unterschiedlich, wenn sie verschiedene *Accessions* haben. Die Schnittmenge ($ER^i \cap ER^j$) umfasst alle Konzepte mit identischen *Accessions* in $ER^i$ und $ER^j$. Zur Veranschaulichung werden die Stabilitätsmaße anhand des Beispiels in Abbildung 8.3 erläutert. Dazu werden zwei Versionen $i$ und $j$ der Ergebnismenge $ER$ einer funktionalen Analyse bezüglich einer Ontologiestruktur[29] betrachtet. Farbige Ontologiekonzepte bezeichnen signifikante Ergebniskonzepte, wobei gelbe Konzepte in beiden Ergebnisversionen auftreten und rote (grüne) Konzepte nur in der Ergebnismenge $ER^i$ ($ER^j$) signifikant sind.

#### Basisstabilität

Grundsätzlich gilt eine Ergebnismenge als stabil bezüglich einer anderen Ergebnismenge, wenn alle Konzepte der beiden Mengen vollständig überlappen und keine

[29] Aus Gründen der Übersicht bleibt die Ontologieversion im Beispiel stabil.

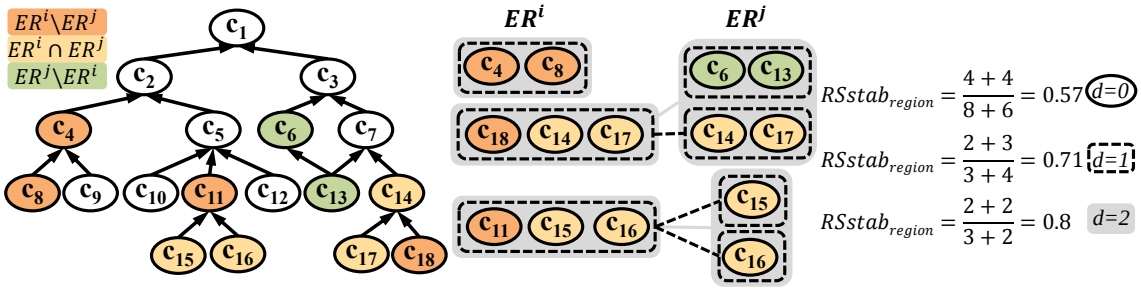



Abbildung 8.3: Beispielergebnisse für zwei $ER$-Versionen und Stabilitätsmaße. Farbige Ergebniskonzepte: gelb - signifikant in $ER^i$ und $ER^j$, rot (grün) - nur in $ER^i$ ($ER^j$) signifikant. Die Stabilität wird für Konzeptregionen ($CR$) der Distanz $d = 0 \ldots 2$ berechnet. $d = 0$: jedes Konzept ist eine Region, gelbe Konzepte überlappen. $d = 1$ ($d = 2$): gestrichelt umrandete (grau hinterlegte) Regionen; durch gestrichelte (graue) Linien verbundene Regionen überlappen.

der beiden Mengen zusätzliche (nicht überlappende) Konzepte enthält. Wenn nur ein Teil der Konzepte überlappt, sinkt der errechnete Stabilitätswert und zeigt somit Instabilität an. Die Basisstabilität nutzt das *Dice*-Maß und wird wie folgt berechnet:

$$RSstab_{basic}(ER^i, ER^j) = \frac{2 \cdot |ER^i \cap ER^j|}{|ER^i| + |ER^j|}$$

Das Maß gibt einen Wert von 0 bis 1 zurück, wobei 0 vollständige Instabilität anzeigt, d. h. zwei Ergebnismengen haben kein Konzept gemeinsam. Eine Stabilität von 1 zeigt hingegen an, dass die zwei betrachteten Ergebnismengen vollständig überlappen. Das Beispiel in Abbildung 8.3 führt zu einer Stabilität von 0,57, da $ER^i$ und $ER^j$ jeweils acht bzw. sechs Konzepte enthalten, wovon vier überlappen.

Die Stabilität zwischen Annotationen verschiedener Versionen ($Astab$) berechnet sich analog. Die Menge der Annotationen umfasst entweder sämtliche Annotationen ($A$) einer Version oder alle Annotationen zu einem bestimmten Konzept $c$ ($A(c)$). Es werden überlappende Annotationen zwischen zwei Versionen bestimmt und mit der Gesamtanzahl der Annotationen normalisiert. Die Berechnung der Stabilität für $A$ (und analog für $A(c)$) ist also identisch zur Stabilität von Ergebnismengen:

$$Astab(A^i, A^j) = \frac{2 \cdot |A^i \cap A^j|}{|A^i| + |A^j|}$$

Für die Ziele dieser Studie eignet sich das *Dice*-Maß, auch wenn durchaus weitere Ansätze zur Bestimmung der Mengenähnlichkeit, wie z. B. *Kosinus* oder *Jaccard*, existieren. Im Kontext der Evolution werden Mengen ähnlicher Größe verglichen und *Dice* entspricht dem harmonischen Mittel. Entsprechend [124] produzieren *Jaccard* und *Kosinus* von *Dice* leicht abweichende Werte, insbesondere für den Fall, dass sich zwei Mengen kaum überschneiden. Lin et al. [118] stellen ein allgemeineres

Maß basierend auf der Informationstheorie vor. Dabei werden Wahrscheinlichkeiten anstelle der Mengenüberlappung genutzt. Ein solcher Ansatz könnte sich ebenfalls gut zur evolutionsbasierten Evaluierung von funktionalen Analysen eignen und sollte in zukünftigen Arbeiten berücksichtigt werden.

### Regionenstabilität

Die Basisstabilität wertet überlappende Konzepte zwischen Ergebnismengen aus, ohne dabei die semantischen Beziehungen zwischen Konzepten zu berücksichtigen. Das bedeutet, alle Konzepte werden unabhängig voneinander betrachtet. Dies könnte zu einer scheinbaren Instabilität führen, auch wenn die unterschiedlichen Konzepte semantisch ähnlich sind und miteinander in Beziehung stehen. Aus diesem Grund wird das Modell erweitert, um zusätzlich strukturelle Ähnlichkeiten zu berücksichtigen. Die Ergebnismengen werden zunächst angereichert, indem „semantisch nahe“ (verwandte) Konzepte in sogenannten *Konzeptregionen* zusammengefasst werden. Die Basisstabilität bleibt als Spezialfall des regionenbasierten Ansatzes (ohne Gruppierung der Konzepte) gültig.

*Semantische Gruppierung der ER-Konzepte:* Die semantische Gruppierung der Konzepte in den Ergebnismengen basiert auf deren Distanz in der Ontologie. Dabei wird der Fakt ausgenutzt, dass verwandte (ähnliche) Konzepte nahe in der Ontologiestruktur beieinander liegen. Das bedeutet, sie sind entweder direkt durch eine Beziehung miteinander verbunden oder liegen nur wenige Kanten voneinander entfernt. Mit dem Parameter $d$ für die Distanz kann die Gruppierung beeinflusst werden. Der Basisfall ohne Gruppierung entspricht $d = 0$. Für $d > 0$ werden rekursiv alle Konzepte gruppiert, die durch $\leq d$ Kanten verbunden sind (siehe Anhang Algorithmus 8). Im Beispiel in Abbildung 8.3 ergeben sich für $d = 1$ drei Regionen in $ER^i$ und vier Regionen in $ER^j$ (gestrichelt umrandete Regionen). Beispielsweise werden $c_6$ und $c_{13}$ in $ER^j$ in einer Region gruppiert. Für $d = 2$ reduziert sich die Anzahl der $ER^j$-Regionen von vier auf zwei (grau hinterlegte Regionen). Zum Beispiel werden $c_{14}$ und $c_{17}$ mit $c_6$ und $c_{13}$ in $ER^j$ zusammengefasst.

Es existieren alternative Methoden zur Gruppierung von Ergebnismengen, wie z. B. verschiedene semantische Ähnlichkeitsmaße (z. B. [147, 180]) oder Klassifikationsalgorithmen (z. B. [142]). Der hier angewendete Ansatz nutzt die semantische Ähnlichkeit von Konzepten basierend auf der Ontologiestruktur und ist einfach anzuwenden. Dies entspricht dem Ansatz vieler funktionaler Analysemethoden (wie z. B. in FUNC), welche die Signifikanz von Konzepten ebenfalls auf Basis der Ontologiestruktur bestimmen und versuchen signifikante Konzepte auf wenige Regionen in der Ontologie zu beschränken.

*Bestimmung der Regionenstabilität*: Die semantische Gruppierung von Konzepten erlaubt die Bestimmung der Stabilität von Ergebnismengen basierend auf Konzeptregionen. Ergebnismengen gelten als stabil, solange die gleichen oder zumindest überlappende Regionen erhalten bleiben. Dazu wird die Anzahl der Regionen in

$ER^i$ bestimmt, die eine Überlappung mit Regionen in $ER^j$ haben und umgekehrt (siehe Anhang Algorithmus 10). Zwei Regionen werden als überlappend betrachtet, wenn sie mindestens ein Konzept teilen. Die überlappenden Regionen werden jeweils für $ER^i$ als $CR^i_o$ und für $ER^j$ als $CR^j_o$ bezeichnet. Um die Regionenstabilität zu berechnen, werden überlappende Regionen wie folgt genutzt:

$$RSstab_{region}(CR^i, CR^j) = \frac{|CR^i_o|+|CR^j_o|}{|CR^i|+|CR^j|}$$

Die Stabilitätswerte liegen wie zuvor im Bereich 0 bis 1. Für $d = 0$ (keine Gruppierung) entsprechen die Stabilitätswerte jenen der Basisstabilität. In Abbildung 8.3, beträgt die berechnete Regionenstabilität 0,57 für $d = 0$, 0,71 für $d = 1$ und sogar 0,8 für $d = 2$. Die Erhöhung des Distanzparameters führt zu weniger, dafür jedoch größeren Regionen. Somit ist es wahrscheinlicher, dass die Ergebnisregionen unterschiedlicher Versionen überlappen. Die Hinzufügung oder Löschung einer Region liegt vor, wenn diese keine Überlappung zu den Regionen der anderen Ergebnisversion aufweist. Hinzufügungen und Löschungen von (ein- oder mehrelementigen) Ontologieregionen stellen somit signifikantere Änderungen dar als die einzelnen, mit der Basisstabilität bestimmten Konzeptänderungen.

### 8.2.4 Datensätze

Für die Evaluierung soll die Analyse zweier realer Datensätze wiederholt werden. Diese wurden zuvor in einer Studie [108] aus dem Jahr 2007 untersucht. Die Autoren führten funktionale Analysen für Gene durch, die Anzeichen positiver Selektion während der Evolution von Primaten und Rodentia (Nagetiere) zeigten. Für diese Untersuchung werden funktionale Analysen mit verschiedenen Ontologie- und Annotationsversionen, unter Verwendung der Datensätze aus [108] durchgeführt. Dazu werden jährliche Annotations- und Ontologieversionen zwischen 2003 und 2010 betrachtet. Dies entspricht acht GOA-Versionen (8, 17, 27, 38, 47, 59, 70, 81) und den zugehörigen GO-Versionen (01-2003, 02-2004, 01-2005, 01-2006, 01-2007, 01-2008, 02-2009, 01-2010). In der Originalpublikation [108] wurden die Versionen $GOA^{47}$ und $GO^{01-2007}$ verwendet. Alle Ergebnisse, basierend auf neueren (2003-2006) und älteren (2008-2010) Versionen ($ER^{comp}$), werden mit den Ergebnissen der Referenzversionen (2007, $ER^{ref}$) verglichen. Für die Untersuchung wurde der Wilcoxon-Rang-Test in FUNC mit den folgenden Parametereinstellungen verwendet. Die Anzahl der Zufallsmengen für die Randomisierung wurde auf 10.000 gesetzt und betrachtete Konzepte müssen mindestens 20 Genen zugeordnet sein. Signifikante Konzepte dürfen einen p-Wert von 0,05 nicht überschreiten.

Experimente auf Basis der realen Datensätze liefern Erkenntnisse zum Einfluss der Evolution auf funktionale Analysen. Um zu testen, ob Beobachtungen verallgemeinert werden können, werden 50 zusätzliche Datensätze simuliert. Für jeden Datensatz wird eine Teilmenge der Konzepte einer Ontologieversion künstlich auf signi-

fikant gesetzt. Dazu wird für diese Konzepte ein höherer Anteil der zugeordneten Gene als „relevant“ markiert, als es für nicht signifikante Konzepte der Fall ist. Die simulierten Datensätze können dann unter Verwendung verschiedener GO-Versionen untersucht werden. Es werden jeweils 50 Datensätze für die GO-Versionen $GO^{01-2007}$ und $GO^{01-2010}$ generiert. Die auf 2007-Versionen basierenden Datensätze werden dann mit GO- und GOA-Versionen aus 2010 (*Task A)* und umgekehrt (*Task B*) getestet und verglichen.

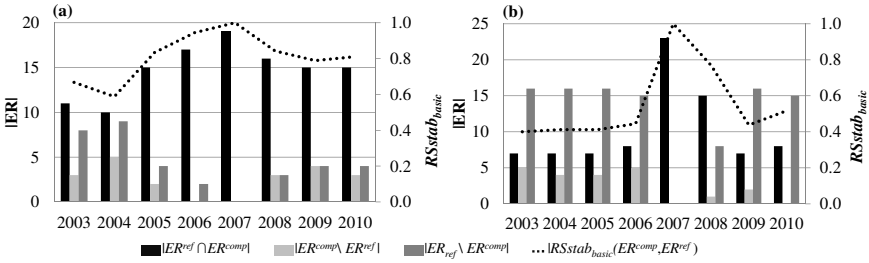
## 8.3 Ergebnisse und Diskussion

Zwischen 2003 und 2010 ist GO um den Faktor 2,4 gewachsen. Gleichermaßen hat sich auch die Anzahl der Eingabeannotationen um den Faktor 2,7 für Rodentia und Primaten erhöht (siehe Tabelle A.3 im Anhang). Dennoch wird von Version zu Version ein Teil der Annotationen entfernt. Der Vergleich der vollständigen Annotationsmappings in 2007 und 2010 zeigt deutliche Instabilitäten ($Astab(A^{2007}, A^{2010}) = 0,7$), d. h. jede dritte Annotation ist von Änderungen betroffen. Um den Einfluss der Ontologie- und Annotationsevolution auf die Ergebnisse funktionaler Analysen zu verstehen, wird die Stabilität zweier realer Datensätze für Primaten und Rodentia zu verschiedenen Zeitpunkten gemessen. Änderungen in den Ergebnismengen der funktionalen Analysen werden mithilfe der Basisstabilität erfasst. Außerdem werden Gründe für Änderungen mithilfe der einfachen Annotationsstabilität *Astab* und dem COnto-Diff-Ergebnis für Ontologieänderungen analysiert. Um schwerwiegendere Änderungen der Ergebnisse zu identifizieren, wird die Regionenstabilität angewendet. Um zu testen, ob die Beobachtungen verallgemeinerbar sind, werden zusätzlich 50 simulierte Datensätze analysiert.

### 8.3.1 Untersuchung der Datensätze für Primaten und Rodentia

#### Basisstabilität

Unter Verwendung von FUNC werden die Ergebnismengen für die jährlichen Versionen zwischen 2003 und 2010 jeweils für den Primaten- (Abbildung 8.4a) sowie für den Rodentia-Datensatz (Abbildung 8.4b) berechnet. Die Ergebnismengen $ER^{comp}$ werden mit der Referenzversion $ER^{ref}$ (2007) durch Identifikation übereinstimmender und abweichender Konzepte verglichen. Daraus resultieren die Menge der überlappenden Konzepte (schwarze Balken) und die Mengen der Konzepte, die in $ER^{comp}$ ($ER^{ref}$) nicht aber in $ER^{ref}$ ($ER^{comp}$) (hellgraue (dunkelgraue) Balken) auftauchen. Unter Verwendung der Referenzversionen aus 2007 umfasst die Ergebnismenge des Primaten-Datensatzes (Abbildung 8.4a) 19 signifikante Konzepte. Allgemein neigen zeitlich nähere Versionen dazu, sich einen größeren Anteil signifikanter Konzepte zu




| Change | RSstab basic (ER 2007, ER 2010) primates | rodents |
|---|---|---|
| $O^v, A^w$ | 0.81 | 0.52 |
| $O^v$ | 0.84 | 0.7 |
| $A^w$ | 0.97 | 0.5 |

Abbildung 8.4: Evolution der experimentellen Ergebnismengen (ERs) zwischen 2003 und 2010 für Primaten (a) und Rodentia (b). (c) Basisstabilität $RSstab_{basic}$ zwischen 2007 und 2010 mit Veränderung von $O^v$ und/oder $A^w$.

teilen. Ab 2008 sind die Ergebnismengen stabiler. Der Stabilitätswert ($RSstab_{basic}$) von 0,81 zeigt, dass ein recht großer Anteil der signifikanten Konzepte aus 2007 auch mit den Daten aus 2010 gefunden wird. Der Rodentia-Datensatz (Abbildung 8.4b) führt 2007 zu 23 signifikanten Konzepten. Die vorherige (2006) und nachfolgende (2008) Version überlappen jeweils mit nur acht bzw. 15 Konzepten. Unter Verwendung der Versionen aus 2010 verbleiben im Vergleich zum Referenzergebnis, nur acht Konzepte und es kommen keine neuen signifikanten Konzepte hinzu. Insgesamt ist die Stabilität der Ergebnismenge mit 0,52 deutlich niedriger als für den Primaten-Datensatz.

Abbildung 8.4c zeigt einen Vergleich der Basisstabilität $RSstab_{basic}$ für die Ergebnismengen von 2007 und 2010 für beide Datensätze. Die Ergebnisse werden durch unabhängige Änderung der Ontologie- und Annotationsversionen getestet, um die Hauptursachen für die beobachteten Ergebnisänderungen zu identifizieren. Zunächst wird wie zuvor gleichzeitig die Ontologie- und Annotationsversion verändert (*Change* $O^v, A^w$). Anschließend wird jeweils nur die Ontologieversion (*Change* $O^v$) oder nur die Annotationsversion (*Change* $A^w$) verändert, wobei die jeweils die andere Quellversion unverändert bleibt. Durch *Change* $O^v$ werden beide Datensätze beeinflusst (Basisstabilität: Rodentia 0,84, Primaten 0,7). Das Ändern der Annotationsversion ($A^w$) bei gleichbleibender Ontologieversion hat nur geringen Einfluss auf den Primaten-Datensatz (0,97), wohingegen die Basisstabilität des Rodentia-Datensatzes deutlich auf 0,5 reduziert wird. Dies zeigt, dass sowohl Ontologie- als auch Annotationsevolution die Ergebnisse funktionaler Analysen beeinflussen.

Im Folgenden sollen die Gründe für Änderungen der Konzeptsignifikanz näher untersucht werden. Dazu werden Ontologieänderungen, welche Einfluss auf signifikante Konzepte $c$ aus $ER$ hatten, sowie die Annotationsstabilität (*Astab*) der Konzepte betrachtet. Das Diagramm in Abbildung 8.5 zeigt alle Ergebniskonzepte in $ER^{2007}$ und $ER^{2010}$ (x-Achse) für Primaten (a) und Rodentia (b). Eine Liste der Ergebniskonzepte inklusive ihrer *Accessions* findet sich im Anhang (Abbildung A.1). Für jedes Konzept wird die Überlappung der Annotationen zwischen beiden Versionen

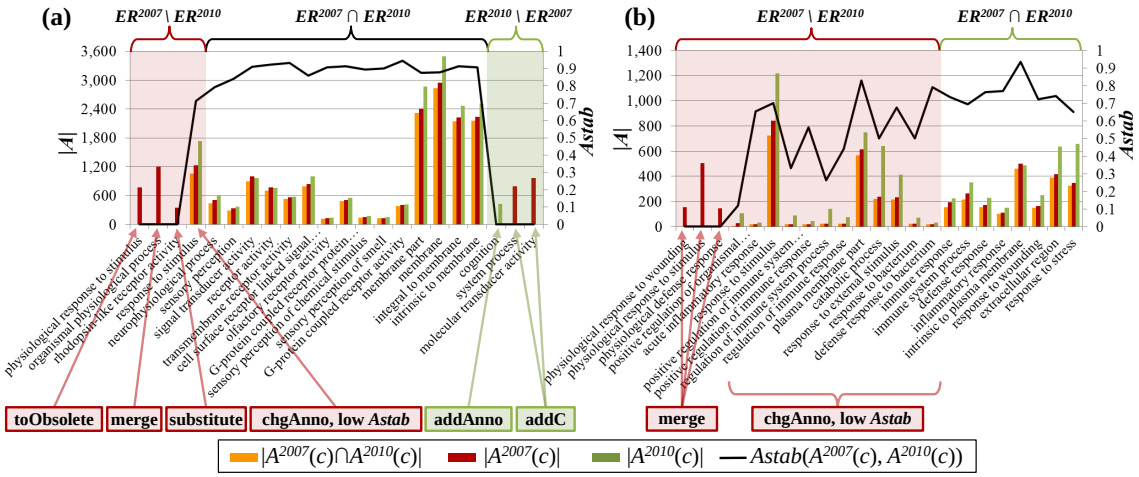



Abbildung 8.5: Annotationsstabilität der Ergebniskonzepte in $ER^{2007}$ und $ER^{2010}$ für (a) Primaten und (b) Rodentia.

($|A^{2007}(c) \cap A^{2010}(c)|$, orange Balken, y-Achse), die Anzahl der Konzeptannotationen in 2007 ($|A^{2007}(c)|$, rote Balken, y-Achse) und 2010 ($|A^{2010}(c)|$, grüne Balken, y-Achse) sowie die resultierende Annotationsstabilität $Astab(A^{2007}(c), A^{2010}(c))$ (schwarze Kurve, z-Achse) dargestellt. Konzepte in den hellrot bzw. hellgrün hinterlegten Diagrammbereichen sind jeweils nur in den Ergebnissen von 2007 bzw. 2010 enthalten. Alle anderen Konzepte sind in beiden $ER$-Versionen signifikant. Zur Identifikation von Ontologieänderungen wurde COnto-Diff verwendet. Die unteren Boxen heben Konzeptänderungen hervor, die direkten Einfluss auf signifikante Konzepte haben (rote Box - informationsreduzierende / -überarbeitende Operation, grüne Box - informationserweiternde Operation).

2010 kommen drei neue signifikante Konzepte für den Primaten-Datensatz hinzu. Die zwei Konzepte „*molecular transducer activity*“ (GO:0060089) und „*system process*“ (GO:0003008) wurden zwischen 2007 und 2010 in GO eingefügt, so dass diese während der funktionalen Analyse zusätzlich detektiert werden. Ein weiteres, bereits seit 2004 existierendes Konzept („*cognition*“, GO:0050890) kommt zur Ergebnismenge hinzu, da sich die Annotationsmenge des Konzepts zwischen 2007 und 2010 vergrößert. Im Gegensatz dazu werden für den Rodentia-Datensatz keine neuen, signifikanten Konzepte ermittelt.

Einige der in $ER^{2007}$ enthaltenen Konzepte bleiben unter Verwendung der Datensätze aus 2010 nicht länger signifikant. Drei der 15 ehemals signifikanten Konzepte im Rodentia-Datensatz werden direkt von einer Ontologieänderung (*merge*-Operation) beeinflusst. Im Gegensatz dazu zeigt ein Großteil der anderen nicht mehr signifikanten Konzepte eine deutliche Reduktion der Annotationsstabilität ($< 0{,}7$). Beispielsweise überlappen für „*regulation of immune system process*“ (GO:0002682) nur 22 der 143 Annotationen in 2010 mit den Konzeptannotationen in 2007 ($Astab < 0{,}3$). Im Primaten-Datensatz sind drei der nicht mehr signifikanten Kon-

zepte von Ontologieänderungen betroffen (*merge*, *substitute*, *toObsolete*) und nur ein Konzeptverlust beruht auf Annotationsänderungen.

Des Weiteren werden starke, strukturelle Veränderungen im direkten semantischen Kontext der signifikanten Konzepte zwischen 2007 und 2010 beobachtet: $|addR| = 31$ (48), $|delR| = 10$ (11), $|move| = 102$ (57) für Primaten (Rodentia) (siehe Anhang Abbildung A.2). Beispielsweise gewann „*catabolic process*“ (GO:0009056) durch *move*-Operationen fünf Kinder hinzu und 11 ehemalige Kinder von „*plasma membrane part*“ (GO:0044459) wurden unter andere Elternkonzepte verschoben. Beide Konzepte haben eine reduzierte Annotationsstabilität und verlieren 2010 ihre Signifikanz im Rodentia-Datensatz. Derartige strukturelle Modifikationen beeinflussen die Propagierung der Annotationen durch die Ontologiehierarchie. Das Hinzufügen oder Löschen von Beziehungen unterhalb eines Konzepts $c$ führt zur Veränderung seiner Annotationsmenge $A(c)$ und somit zu einer geringeren Annotationsstabilität *Astab*. Dies kann schließlich die Signifikanz von $c$ verändern. Insgesamt zeigen die meisten unbeeinflussten, signifikanten Konzepte eine höhere Annotationsstabilität als Konzepte die an Signifikanz zu- oder abnehmen (siehe Abbildung 8.5).

Automatisch generierte Annotationen könnten weniger glaubwürdige Ergebnisse als manuell geprüfte Annotationen produzieren (siehe Kapitel 7). Um diese Hypothese zu prüfen, wird die Analyse für beide Datensätze unter ausschließlicher Verwendung manuell geprüfter Annotationen (ohne IEA) wiederholt. Da 60% der Annotationen automatisch generiert wurden, reichen manuelle Annotationen möglicherweise nicht aus, um signifikante Ergebnisse zu produzieren (siehe Anhang Abbildung A.3). Für Primaten umfasst die Ausgabe für 2007 (2010) ohne IEA-Annotationen lediglich 4 (1) statt 19 (18) signifikante Konzepte. Ähnlich werden für Rodentia nur 9 (2) statt 23 (8) Ergebniskonzepte für 2007 (2010) produziert (siehe Anhang Abbildung A.4). Dementsprechend ist die Menge der Ergebniskonzepte ohne Nutzung automatisch generierter Annotationen zu klein, um Schlussfolgerungen bezüglich der Ergebnisstabilität mit und ohne IEA-Annotationen treffen zu können. Nach derzeitigem Stand ist die Verwendung automatisch generierter Annotationen in funktionalen Analysen notwendig, da sonst kein ausreichend starkes Signal zur Detektion signifikanter Konzepte erreicht wird.

### Regionenstabilität

Die Anwendung der Regionenstabilität (Distanz $d = 1$) fasst einzelne, signifikante Konzepte in größeren Regionen zusammen (Abbildung 8.6). Für die Primaten werden 2007 vier Konzeptregionen identifiziert. Alle vier Regionen überlappen und der Inhalt der Regionen hat sich von 2007 nach 2010 kaum geändert. Durch die Betrachtung der Konzeptregionen anstelle einzelner Konzepte ergibt sich eine perfekte Stabilität ($RSstab_{region} = 1$ statt $RSstab_{basic} = 0,81$). Für Rodentia, ergeben sich jeweils vier signifikante Regionen in 2007 ($R1$,$R2$,$R3$,$R4$) und 2010 ($R2$,$R3$,$R4a$,$R4b$) jedoch überlappen nur drei der vier Regionen aus 2007. Die Region $R1$ ("catabolic process")

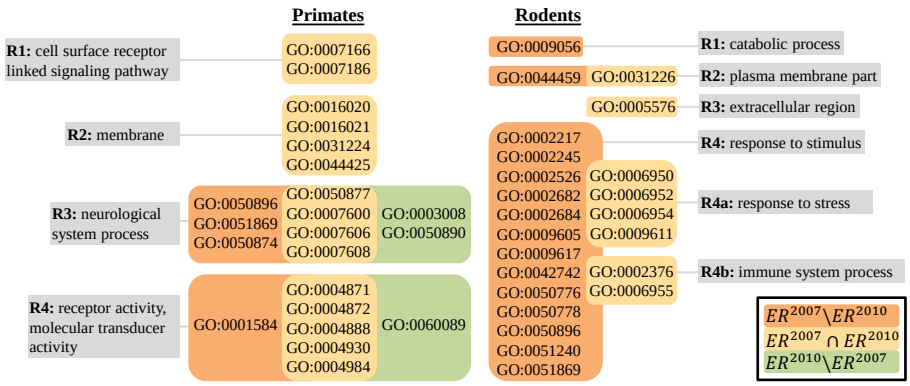



Abbildung 8.6: Konzeptregionen für Primaten und Rodentia, $d = 1$.

hat aufgrund starker Annotationsevolution (siehe Abbildung 8.5) an Signifikanz verloren und ist in 2010 nicht mehr Teil der Ergebnismenge. Weiterhin existiert 2007 die sehr große Region $R4$ ("response to stimulus"), aus welcher 13 Konzepte 2010 nicht mehr signifikant sind. Die sechs verbleibenden Konzepte $R4$ teilen sich in zwei Regionen ($R4a$ „*response to stress*“, $R4b$ „*immune system process*“). Der Rodentia-Datensatz hat eine Regionenstabilität von $RStab_{region} = \frac{3+4}{4+4} = 0,875$ (anstelle von $RStab_{basic} = 0,52$). Der Vergleich von Konzeptregionen bezieht semantische Informationen mit ein und betrachtet größere Regionen anstelle einzelner Konzepte. Dies führt zu höheren Stabilitätswerten. Wenn die Regionenstabilität reduziert ist, können Rückschlüsse auf bedeutsame, inhaltliche Änderungen der Ergebnismenge gezogen werden.

Anhand der hier analysierten Datensätze zeigt sich, dass Ontologie- und Annotationsänderungen Einfluss auf die Ergebnisse funktionaler Analysen haben. Die Auswirkungen auf den Primaten- bzw. Rodentia-Datensatz unterscheiden sich stark. Während die Ergebnisse des Primaten-Datensatzes relativ stabil sind, zeigen sich für den Rodentia-Datensatz bedeutsamere Änderungen. Der Primaten-Datensatz wurde eher durch Ontologieänderungen beeinflusst, wohingegen Annotationsänderungen deutliche Auswirkungen auf den Rodentia-Datensatz hatten. Häufig produzieren funktionale Analysen semantisch verwandte Konzepte, was den Einfluss von Ontologie- und Annotationsänderungen reduziert. Die Mehrheit der Änderungen hat keinen Einfluss auf die semantische Interpretation der funktionalen Analysen, auch wenn wenige Fälle identifiziert werden konnten, für die sich die Interpretation der Daten ändern könnte. Die Ergebnisse funktionaler Analysen sind folglich relativ robust gegenüber der Evolution von Ontologien und Annotationen.

### 8.3.2 Simulierte Datensätze

Für die realen Datensätze wurde beobachtet, dass einige Ergebniskonzepte durch Ontologie- und Annotationsevolution beeinflusst werden und dass diese Änderungen zu Unterschieden in den Ergebnissen der funktionalen Analysen führen. Um zu testen, ob diese Erkenntnisse verallgemeinert werden können, wurden je 50 Datensätze für die zwei GO-Versionen (2007, 2010) simuliert (Task A, Task B), wobei ein Teil der Konzepte der $GO^{01-2007}$ bzw. $GO^{01-2010}$ zufällig auf signifikant gesetzt wurde (siehe Kapitel 8.2.4).

| | ***Task A*** | ***Task B*** |
|---|---|---|
| $avg(\left\|ER^{only2007}\right\|)$ | 129.3 | 62.6 |
| $avg(\left\|CR^{only2007}\right\|)$ | 8.9 | 4.2 |
| $avg(\left\|ER^{only2010}\right\|)$ | 440.7 | 856.8 |
| $avg(\left\|CR^{only2010}\right\|)$ | 18.9 | 21.4 |
| $avg(RSstab_{basic})$ | 0.625 | 0.519 |
| $avg(RSstab_{region})$ | 0.719 | 0.707 |

Tabelle 8.1: Durchschnittliche Anzahl signifikanter Ergebniskonzepte ($|ER|$ entspricht $|CR|, d = 0$) und Konzeptregionen ($|CR|$, $d = 1$), die nur im 2007-Ergebnis (fehlend) bzw. nur im 2010-Ergebnis (neu) enthalten sind. $avg(RSstab_{basic})$ - durchschnittliche Basisstabilität, $avg(RSstab_{region})$ - durchschnittliche Regionenstabilität.

Tabelle 8.1 zeigt Durchschnittswerte der 50 simulierten Datensätze (für Details siehe Tabelle A.4 und A.5). Die simulierten Datensätze sind größer und decken folglich größere Ontologieteile ab, wodurch die Stabilität generell niedriger sein kann als für die beiden realen Datensätze. Die Analyse der Änderungen von 2007 nach 2010 (*Task A*), führt durchschnittlich zu neun Konzeptregionen, die ihre Signifikanz verlieren ($avg(\left|CR^{only2007}\right|)$), wohingegen 19 neue signifikante Regionen identifiziert werden ($avg(\left|CR^{only2010}\right|)$). Änderungen von 2010 nach 2007 (*Task B*) führen zum Verlust von durchschnittlich vier signifikanten Konzeptregionen. Zudem kommen durchschnittlich 21 neue signifikante Regionen hinzu. Der Vergleich einzelner Konzepte führt zu weniger kompakten Ergebnissen. Beispielsweise existieren für *Task A* durchschnittlich 129 signifikante Ergebniskonzepte ($avg(\left|ER^{only2007}\right|)$), die für $d = 1$ zu den circa neun Konzeptregionen zusammengefasst werden. Die Anwendung der Regionenstabilität $avg(RSstab_{region})$ ($d = 1$) liefert höhere Stabilitätswerte (≈0,7) als die Basisstabilität $avg(RSstab_{basic})$ (≈0,5 - ≈0,6), da Regionen bei Änderungen semantisch ähnlicher Konzepte häufig erhalten bleiben und dies nicht als echte Hinzufügung bzw. als echter Verlust betrachtet werden kann. Die Beobachtungen bestätigen die Ergebnisse der realen Datensätze: Ontologie- und Annotationsevolution haben Einfluss auf die Ergebnisse funktionaler Annotationen. Zudem sind signifi-

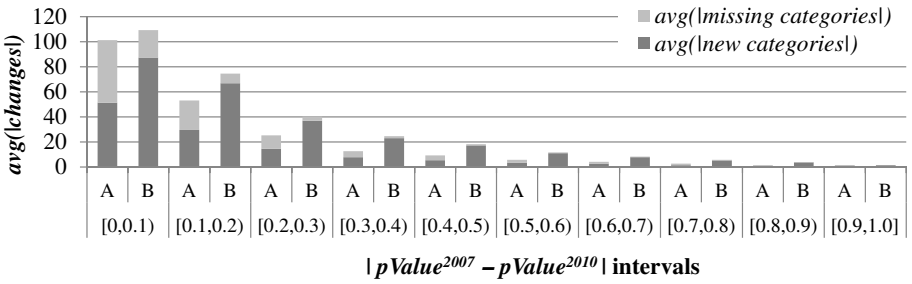



Abbildung 8.7: Durchschnittliche Anzahl geänderter Konzepte, gruppiert nach der Differenz ihrer absoluten p-Werte in 2007 und 2010 ($|pValue^{2007} - pValue^{2010}|$) für *Task A* und *B*.

kante Ergebniskonzepte oft semantisch ähnlich bezüglich der Distanz in der Ontologiehierarchie. Sie können somit zu größeren Regionen zusammengefasst werden, die robuster gegenüber der Evolution sind.

Da Konzepte bezüglich eines festen Grenzwerts für p-Werte ($\alpha = 0,05$) als signifikant bezeichnet werden, ist es möglich, dass einige der Konzepte, die an Signifikanz gewinnen oder verlieren, dies nur aufgrund sehr kleiner p-Wert-Veränderungen in der Nähe des Grenzwerts tun. Um den relativen Einfluss dieses Effekts zu untersuchen, werden die Differenzen der p-Werte zwischen 2007 und 2010 für jedes signifikante Konzept berechnet. Abbildung 8.7 zeigt die Verteilung der p-Wert-Differenzen für fehlende und hinzukommende, signifikante Konzepte. Die meisten Konzepte weisen relative kleine Änderungen des p-Werts auf ($< 0,1$). Dies zeigt, dass viele der Verluste und Hinzufügungen durch kleinere Fluktuationen der Signifikanz verursacht werden. Trotzdem zeigen einige Konzepte substanzielle p-Wert-Unterschiede, was darauf hinweist, dass derartigen Änderungen des p-Werts folgenreiche Ontologie- oder Annotationsänderungen zugrunde liegen.

## 8.4 Zusammenfassung

Funktionale Analysen nutzen Ontologien und Annotationen, um besondere Eigenschaften in Form signifikant über- oder unterrepräsentierter Ontologiekonzepte für biologische Datensätze wie z. B. Genexpressionsdaten zu identifizieren. Es wurde der Einfluss der Ontologie- und Annotationsevolution auf funktionale Analysen untersucht, indem Ergebnisse auf Basis verschiedener Ontologie- und Annotationsversionen verglichen wurden. Zur Bewertung der Stabilität der Ergebnisse funktionaler Analysen wurden verschiedene Maße vorgeschlagen und zur Untersuchung der Auswirkungen der Evolution auf zwei reale und 50 simulierte Datensätze angewendet. Die Betrachtung einzelner Ergebniskonzepte zeigte, dass die Ergebnisse funktionaler Analysen teilweise signifikant durch die Evolution der verwendeten Ontologie und Annotationen beeinflusst werden. Insbesondere kommen einige neue Konzepte zum

Ergebnis hinzu und einige ursprünglich signifikante Konzepte fallen weg. Jedoch verändern diese Änderungen nicht zwingend die Interpretation der Ergebnisse, da signifikante Konzepte häufig semantisch ähnlich sind. Dieser Effekt wird insbesondere durch das Maß der Regionenstabilität erfasst. Dessen Anwendung zeigte, dass funktionale Analysen insgesamt relativ robust gegenüber der Ontologie- und Annotationsevolution sind, da selten sämtliche Konzepte einer Region wegfallen oder völlig (semantisch) neue Konzepte zum Ergebnis hinzukommen.

Durch Anwendung der Maße können Nutzer Konzepte identifizieren, die aufgrund struktureller Ontologieänderungen oder starker Veränderungen der zugeordneten Annotationen dazu neigen, ihr Signifikanzlevel zu verändern. Zudem ist der Einsatz von COnto-Diff hilfreich, um betroffene Konzepte zu identifizieren. Die folgenden Zielgruppen können von den vorgestellten Methoden und Ergebnissen profitieren:

1. *Ontologiekuratoren*: Für diese Nutzer ist es wichtig zu wissen, ob geplante Änderungen in der Ontologie oder den Annotationen zu semantisch bedeutsamen Änderungen führen. In dieser Studie wurde gezeigt, dass strukturelle Ontologieänderungen nicht notwendigerweise eine semantische Veränderung der Ergebnisse implizieren. Die vorgestellten Stabilitätsmaße ermöglichen es, geplante Änderungen in GO sowie deren Auswirkungen auf funktionale Analysen zu testen.

2. *Biologen, die GO für funktionale Analysen verwenden*: Für diese Nutzer ist es interessant zu wissen, dass sich Ergebnisse funktionaler Analysen aufgrund der Evolution von Ontologien und Annotationen über die Zeit verändern können. Die Nutzer sollten dies bei der Interpretation ihrer Ergebnisse berücksichtigen.

# Teil IV

# Matching großer Ontologien

# 9 Kompositionsbasiertes Matching

## 9.1 Motivation

Häufig existieren mehrere Ontologien, welche teilweise überlappende Informationen einer Domäne enthalten. Beispielsweise findet sich anatomisches Wissen im *Foundational Model of Anatomy* (FMA) [158], im *NCI Thesaurus* (NCIT) [162], in der *Adult Mouse Anatomy* (MA) [84] sowie in zahlreichen weiteren Ontologien. Diese Situation erfordert die Bestimmung von Mappings zwischen ähnlichen Ontologien. Die Mappings sind u. a. hilfreich für zahlreiche Datenintegrationsaufgaben ([93]), zur Kombination (Merging) mehrerer Ontologien (z. B. [110, 157, 156]) oder in der translationalen Medizin z. B. bei der Erstellung von Mausmodellen für die humane Krebsforschung[30] (z. B. [18]).

Da die manuelle Erstellung von Mappings insbesondere zwischen großen Ontologien sehr aufwendig oder teilweise nicht realisierbar ist, kommen häufig automatische Verfahren zum Einsatz (siehe Kapitel 2.3.1). Nur wenige Arbeiten beschäftigten sich bisher mit der Wiederverwendung bereits existierender Mappings, um bisher nicht verknüpfte Ontologien abzugleichen (siehe Kapitel 2.3.2 und 2.3.3). Insbesondere kann die Komposition von Mappings zu einer Zwischenontologie genutzt werden, um Ontologiemappings indirekt zu berechnen. Ein Ontologiemapping zwischen MA und NCIT kann (zumindest teilweise) durch Komposition zweier existierender Mappings zur *Uber anatomy ontology* (Uberon) [135] oder zum Metathesaurus des *Unified*

[30] Mouse Models of Human Cancers Consortium:
http://www.nih.gov/science/models/mouse/resources/hcc.html

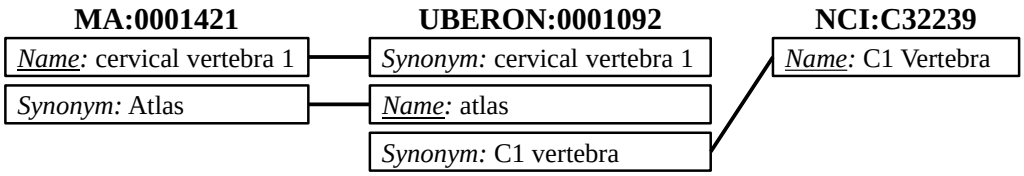



Abbildung 9.1: Komposition von Korrespondenzen mithilfe eines Uberon-Konzepts.

*Medical Language Systems* (UMLS) [16] erstellt werden. Abbildung 9.1 veranschaulicht den Ansatz für zwei ausgewählte Konzepte (MA:0001421 und NCI:C32239), die durch einen Namen und Synonyme näher beschrieben werden. Ein direkter, automatischer Abgleich der beiden Konzepte ist nicht trivial, da sich ihre Namen unterscheiden. Der Einsatz von Uberon als Zwischenontologie erlaubt hingegen eine Wiederverwendung der Korrespondenzen MA:0001421-UBERON:0001092 (exakte Übereinstimmung von '*atlas*') und UBERON:0001092 - NCI:C32239 (exakte Übereinstimmung von '*C1 Vertebra*'). Die Komposition dieser zwei Korrespondenzen resultiert in der neuen Korrespondenz MA:0001421 - NCI:C32239.

Die Komposition existierender Mappings zum automatischen Abgleich von Ontologien ist in mehrfacher Hinsicht vielversprechend. Erstens entstehen im Bereich der Lebenswissenschaften regelmäßig neue Ontologiemappings, die in öffentlichen Repositories wie BioPortal [141] gesammelt und zur Verfügung gestellt werden. Die Wiederverwendung ist besonders hilfreich, wenn die Mappings eine hohe Qualität aufweisen, indem sie beispielsweise durch Domänenexperten validiert werden. Zweitens kann die Komposition von Ontologiemappings sehr schnell ausgeführt werden. Im Gegensatz dazu ist das direkte Matching großer Ontologien sehr zeitaufwendig, insbesondere wenn jedes Konzept der Quellontologie mit jedem Konzept der Zielontologie abgeglichen wird (Berechnung des kartesischen Produkts). Drittens kann eine Zwischenontologie zusätzliches Wissen enthalten, das nützlich ist, um weitere Korrespondenzen zu finden und somit die Mappingqualität zu verbessern. Schließlich kann eine sehr umfassende Zwischenontologie als eine Art Mediator (*engl. hub*) aufgefasst werden, wenn diese eine zentrale Rolle (innerhalb einer bestimmten Domäne) spielt und bereits Mappings zu verschiedenen Ontologien bietet. Eine solche Mediatorontologie hat ein besonders gutes Wiederverwendungspotenzial. Wenn eine neue Ontologie hinzukommt, muss nur ein Mapping zwischen der neuen sowie der Mediatorontologie erstellt werden, so dass durch Komposition Mappings zu allen weiteren verknüpften Ontologien bestimmt werden können.

Aufgrund dieser Vorteile soll das kompositionsbasierte Matching von Ontologien im Bereich der Lebenswissenschaften untersucht werden. Das Kapitel umfasst die folgenden Beiträge:

- Es wird ein kompositionsbasiertes Match-Verfahren auf Basis der Wiederverwendung zuvor bestimmter Mappings mit einer oder mehreren Zwischenontologien vorgestellt.

- Der Ansatz basiert auf generischen Ontologie- und Mappingoperatoren wie `compose`, `match` und `extract`. Zudem wird eine inkrementelle Erweiterung kombinierter Mappings zur Verbesserung der Mappingqualität unterstützt.

- Die Evaluierung erfolgt durch Bestimmung von Ontologiemappings zwischen MA und dem Anatomieteil des NCIT unter Verwendung der Zwischenontologien UMLS, FMA, Uberon und RadLex. Die Ergebnisse zeigen die hohe Effektivität und Effizienz des kompositionsbasierten Ontologie-Matchings.

## 9.2 Mappingkomposition

In diesem Abschnitt wird der kompositionsbasierte Match-Algorithmus zum indirekten Matching von Ontologien auf Basis der Wiederverwendung von Ontologiemappings vorgestellt. Zur direkten Bestimmung von Mappings wird das linguistische Name/Synonym-Match-Verfahren (*NameSyn*) aus GOMMA (Kapitel 3.2.1) verwendet. Alle berechneten Mappings werden durch Anwendung von Selektionskriterien (minimaler Grenzwert, MaxDelta) und anderen Nachbearbeitungsschritten gefiltert, um möglichst präzise Ergebnisse zu erhalten. Für den hier vorgestellten Ansatz wird jeweils nur eine Version jeder betrachteten Ontologie benötigt. Zur Vereinfachung wird daher auf den Index zur Repräsentation von Ontologieversionen verzichtet. Zwischen zwei Ontologien $O1$ und $O2$ wird also ein Mapping $OM_{O1,O2}$ bestimmt. Zunächst wird die generelle Idee zur Verwendung von Zwischenontologien diskutiert (Kapitel 9.2.1). Zudem werden Ontologie- und Mappingoperatoren eingeführt (Kapitel 9.2.2). Diese werden anschließend im kompositionsbasierten Match-Algorithmus eingesetzt (Kapitel 9.2.3).

### 9.2.1 Indirektes Matching mithilfe von Zwischenontologien

Die generelle Idee des Ansatzes ist es, Mappings zu Zwischenontologien für das indirekte Matching von Ontologien zu verwenden. Solche Mappings werden typischerweise durch einen aufwendigen Match-Prozess bestimmt, insbesondere wenn (Teile der) Mappings manuell erstellt oder durch fortgeschrittene Match-Algorithmen berechnet werden. Daher ist es vielversprechend bestehende Ontologiemappings wiederzuverwenden, um sich den hohen Aufwand einer vollständigen Neubestimmung zu sparen oder zumindest zu reduzieren.

Abbildung 9.2(a) zeigt die Grundsituation für zwei Ontologien $O1$ und $O2$ sowie Mappings von $O1/O2$ zu mehreren Zwischenontologien (*Intermediate Ontologies*) $IO_1, \ldots, IO_k$. Zwischenontologien sollten eine signifikante Überlappung mit den Ontologien $O1$ und $O2$ haben, d. h. es sollte Korrespondenzen zu einem großen Teil der Ontologiekonzepte in $O1$ und $O2$ geben. Es ist sinnvoll, das Wissen verschiedener Zwischenontologien auszunutzen, da diese sich gegenseitig ergänzen können. Die

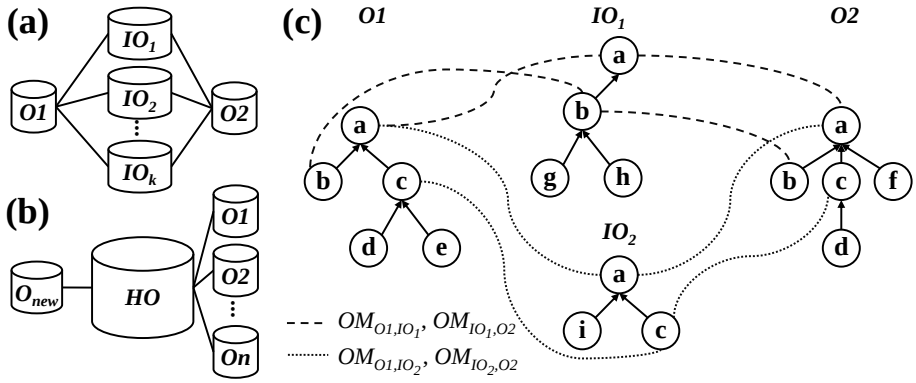



Abbildung 9.2: (a) Mappingkomposition über Zwischenontologien $IO_1, \ldots, IO_k$
(b) Matching einer neuen Ontologie $O_{new}$ zur Mediatorontologie $HO$
(c) Beispiel zum kompositionbasierten Ontologie-Matching.

Komposition von Ontologiemappings ist voraussichtlich sehr effizient, so dass auch für mehrere Zwischenontologien eine schnelle Mappingkomposition möglich ist.

In einigen Fällen existiert eine zentrale, dominierende Mediatorontologie (*Hub Ontology*) $HO$ in einer Domäne (siehe Abbildung 9.2(b)). Typischerweise hat eine solche Ontologie viele Mappings zu anderen Ontologien. Jede neue Ontologie einer Domäne $O_{new}$ kann dann mit jeder anderen Ontologie $O1, \ldots, On$ verknüpft werden, indem zunächst ein Mapping zwischen $O_{new}$ und $HO$ erstellt wird. Anschließend kann das Mapping $OM_{O_{new},HO}$ mit jedem verfügbaren Mapping $OM_{HO,Oi}$ $(i = 1, \ldots, n)$ der Domäne kombiniert werden. Demzufolge können Mappings zwischen $O_{new}$ und jeder anderen Ontologie $Oi$ sehr effizient berechnet werden.

### 9.2.2 Operatoren

Bisherige Forschungsarbeiten im Bereich des Model Management (siehe Kapitel 2.2.1) stellen bereits verschiedene Operatoren zur Verfügung, die sich für die Anwendung zum Matching von Ontologien anpassen lassen. Im Folgenden werden die Ontologie- und Mappingoperatoren `match`, `compose` und `extract` eingeführt. Zudem ermöglicht ein `merge`-Operator[31] die Aggregation mehrerer Mappings.

Der `match`-Operator gleicht die Konzepte einer Ontologie $A$ gegen die Konzepte einer zweiten Ontologie $B$ ab und bestimmt direkt ein Ontologiemapping $OM_{A,B}$, das aus Korrespondenzen mit Ähnlichkeitswerten zwischen 0 und 1 besteht. Im Rahmen dieser Studie wird eine vereinfachte Notation für Mappings ohne den semantischen Typ (*semType*) und den Status (*status*) verwendet. Der `match`-Operator erzeugt Äquivalenzmappings (*semType* '=') mit dem Status *to verify*.

[31] Der hier definierte `merge`-Operator fasst mehrere Mappings zu einem Mapping zusammen und entspricht nicht dem im Rahmen des Model Management definierten `merge`-Operator zum Zusammenfassen mehrerer Modelle zu einem Modell.

$$\texttt{match}(A, B)\colon A \times B \rightarrow OM_{A,B}$$
$$OM_{A,B} = \{(c_1, c_2, sim) | c_1 \in A, c_2 \in B, sim \in [0, 1]\}$$

Der `compose`-Operator erlaubt die Komposition von Mappings. Der Operator kombiniert zwei Mappings $OM_{A,B}$ und $OM_{B,C}$, um indirekt ein neues Mapping zu bestimmen. Zwei Korrespondenzen verschiedener Mappings können zu einer neuen Korrespondenz kombiniert werden, wenn das Zielkonzept der ersten mit dem Quellkonzept der zweiten Korrespondenz übereinstimmt. Zur Aggregation der Korrespondenzähnlichkeiten ($aggSim$) können verschiedene Funktionen (z. B. *average*) verwendet werden. Die `compose`-Definition entspricht der zuvor eingeführten Definition im Rahmen der kompositionsbasierten Mappingadaptierung (Kapitel 6.3). In dieser Studie werden Äquivalenzmappings bestimmt ($semType$ '='). Der Korrespondenzstatus ist *to verify* bzw. *handled*, je nachdem ob automatisch bestimmte oder bereits geprüfte Eingabemappings verwendet werden. Für diese Untersuchung wird die `compose`-Definition etwas vereinfacht:

$$\texttt{compose}(OM_{A,B}, OM_{B,C})\colon OM_{A,B} \times OM_{B,C} \rightarrow OM_{A,C}$$
$$OM_{A,C} = \{(c_1, c_2, aggSim(sim_1, sim_2)) | c_1 \in A, b \in B, c_2 \in C :$$
$$\exists (c_1, b, sim_1) \in OM_{A,B} \wedge \exists (b, c_2, sim_2) \in OM_{B,C}\}$$

Der hier definierte `extract`-Operator[32] reduziert eine Ontologie $A$ auf eine Delta-Ontologie $\Delta A$, indem nur diejenigen Konzepte erhalten bleiben, die nicht durch ein Eingabemapping $OM_{A,B}$ zwischen $A$ und einer anderen Ontologie $B$ abgedeckt sind. Der Operator kann somit verwendet werden, um Vergleiche für Korrespondenzen einzusparen, die bereits in dem (partiellen) Mapping $OM_{A,B}$ enthalten sind.

$$\texttt{extract}(A, OM_{A,B})\colon A \times OM_{A,B} \rightarrow \Delta A$$
$$\Delta A = \{c | c \in A, \nexists b \in B : (c, b, sim) \in OM_{A,B}\}$$

Der `merge`-Operator aggregiert mehrere Eingabemappings zwischen zwei Ontologien $A$ und $B$ zu einem Mapping. Dabei werden Korrespondenzen in das aggregierte Mapping übernommen, wenn sie in mindestens $occ$ der $k$ Eingabemappings vorkommen. $occ = 1$ entspricht der klassischen Vereinigung, wohingegen $occ = k$ der Schnittmenge aller Mappings entspricht.

$$\texttt{merge}(OM1_{A,B}, \ldots, OMk_{A,B}, occ))\colon OM1_{A,B} \times \ldots \times OMk_{A,B} \times occ \rightarrow OM_{A,B}$$
$$OM_{A,B} = \{(c_1, c_2, aggSim) | (c_1, c_2, sim)$$
$$\text{occurs in at least } occ \text{ mappings of } OM1_{A,B}, \ldots OMk_{A,B}\}$$

[32] Die Definition orientiert sich an der Idee des `diff`-Operators (sowie dem komplementären `extract`-Operator) im Model Management. In dieser Arbeit wird der Begriff „Diff“ bereits für Evolutionsmappings ($diff$) genutzt. Daher wird der Operator zur Extraktion einer im Mapping nicht abgedeckten Teilontologie als `extract` bezeichnet.

### 9.2.3 Kompositionbasierte Match-Algorithmen

Die eingeführten Operatoren werden in zwei Algorithmen zum kompositionsbasierten Matching verwendet: *composeMatch* (Algorithmus 6) and *extendMatch* (Algorithmus 7). *composeMatch* erhält als Eingabe zwei Ontologien $O1$ und $O2$, eine Liste bestehend aus einer oder mehreren Zwischenontologien $IO_1, \ldots, IO_k$ sowie den Parameter *occ* für die Korrespondenzhäufigkeit im Mapping-`merge`. Der Algorithmus berechnet ein kombiniertes Mapping zwischen $O1$ und $O2$ durch Verwendung existierender Mappings zu einer Zwischenontologie. Zunächst werden für jede Zwischenontologie $IO_i$ die bestehenden Mappings zwischen $O1$ und $IO_i$ sowie $IO_i$ und $O2$ (z. B. aus einem Repository) geladen. Anschließend wird der `compose`-Operator auf die Mappings $OM_{O1,IO_i}$ und $OM_{IO_i,O2}$ angewendet, um ein Mapping zwischen $O1$ und $O2$ zu bestimmen. Dieses kombinierte Mapping wird zur Liste der Mappings *MapList* hinzugefügt. Abschließend werden alle Mappings in *MapList* mithilfe des `merge`-Operators in Abhängigkeit von *occ* zu einem Mapping aggregiert. Das Zusammenführen mehrerer Mappings kann die Mappingqualität, insbesondere den Recall, verbessern. Beispielsweise kann die Vereinigung von Mappings zu komplementären Zwischenontologien helfen, insgesamt mehr korrekte Korrespondenzen zu finden. Falls die Eingabeliste nur eine Zwischenontologie wie z. B. eine zentrale Mediatorontologie enthält, kann der „merge“-Schritt ausgelassen werden.

**Algorithmus 6:** composeMatch($O1, O2, IO_1 \ldots IO_k, occ$)

**Input** : Zwei Ontologien $O_1$ and $O_2$, Liste von Zwischenontologien $IO_1 \ldots IO_k$, Anzahl Korrespondenzvorkommen *occ*

**Output** : Kombiniertes Mapping $CM_{O1,O2}$

```
1 MapList ← empty;
2 foreach IO_i ∈ IO do
3 |   OM_{O1,IO_i} ← getMapping(O1, IO_i);
4 |   OM_{IO_i,O2} ← getMapping(IO_i, O2);
5 |   MapList.add(compose(OM_{O1,IO_i}, OM_{IO_i,O2}));
6 return merge(MapList, occ);
```

Es ist möglich bzw. wahrscheinlich, dass ein kombiniertes Mapping $CM_{O1,O2}$ nicht alle überlappenden Bereiche der Ontologien $O1$ und $O2$ abdeckt. Daher kann der Algorithmus *extendMatch* (Algorithmus 7) optional angewendet werden, um den Recall und somit die Mappingqualität weiter zu verbessern. Die Eingabe umfasst zwei Ontologien sowie das kombinierte Mapping. Um zusätzliche Korrespondenzen zwischen noch nicht verknüpften Ontologiebereichen zu finden, werden mithilfe des `extract`-Operators Subontologien von $O1$ und $O2$ bestimmt, die noch nicht durch $CM_{O1,O2}$ abgedeckt sind. Die resultierenden Delta-Ontologien $\Delta O1$ und $\Delta O2$ werden anschließend direkt abgeglichen. Das Matching der Delta-Ontologien genügt, falls 1:1-Korrespondenzen erwartet werden, da lediglich weitere Korrespondenzen für noch nicht verknüpfte Konzepte gesucht werden. Falls zwischen den betrachteten

Ontologien ein n:m - Mapping besteht, müssen $\Delta O1$ und $\Delta O2$ jeweils mit der gesamten anderen Ontologie ($O2$ und $O1$) abgeglichen werden. Zum Abgleich der Delta-Ontologien kommt ein spezifisches Match-Verfahren (z. B. *NameSyn*) zum Einsatz. Das ermittelte direkte Mapping $DM_{\Delta O1,\Delta O2}$ wird mit dem kombinierten Mapping $CM_{O1,O2}$ vereinigt (`merge` mit $occ = 1$).

**Algorithmus 7:** extendMatch($O1, O2, CM_{O1,O2}$)

**Input** : Zwei Ontologien $O1$ and $O2$, kombiniertes Mapping $CM_{O1,O2}$
**Output** : Erweitertes Mapping $EM_{O1,O2}$

1 $\Delta O1 \leftarrow$`extract(`$O1, CM_{O1,O2}$`)`;
2 $\Delta O2 \leftarrow$`extract(`$O2$,`inverse(`$CM_{O1,O2}$`))`;
3 $DM_{\Delta O1,\Delta O2} \rightarrow$`match(`$\Delta O1, \Delta O2$`)`;
4 $EM_{O1,O2} \leftarrow$`merge(`$\{CM_{O1,O2}, DM_{\Delta O1,\Delta O2}\}, 1$`)`;
5 **return** $EM_{O1,O2}$;

In Kapitel 6.3 wurde der `compose`-Operator eingesetzt, um ein veraltetes Mapping auf aktuelle Ontologieversionen zu migrieren. Zusätzlich wurden neue Konzepte ($Add_{O1}$, $Add_{O2}$) für je zwei Versionen von $O1$ und $O2$ ermittelt. *extendMatch* identifiziert hingegen nicht verknüpfte Ontologiebereiche bezüglich eines bestehenden Mappings. Da für hinzugefügte Konzepte infolge der Weiterentwicklung einer Ontologie nicht klar ist, zu welchen existierenden Konzepten sie verknüpft werden können, wurden während der Mappingadaptierung alle neuen Konzepte mit der gesamten anderen Ontologie abgeglichen.

Abbildung 9.2(c) veranschaulicht beispielhaft die Anwendung von *composeMatch* (Algorithmus 6) zum Abgleich der Ontologien $O1$ und $O2$ über zwei Zwischenontologien $IO_1$ und $IO_2$. Gepunktete Linien repräsentieren Korrespondenzen von $O1$ und $O2$ zu den Zwischenontologien. Die Mappingkomposition (Zeile 2-5) gibt die folgende *MapList* aus: $\{\{(a,a),(b,b)\},\{(c,c),(a,a)\}\}$. Die Aggregation der *MapList* mittels `merge` (Zeile 6) resultiert für $occ = 1$ in dem Mapping $\{(a,a),(b,b),(c,c)\}$, wohingegen für die Anwendung von $occ = 2$ nur eine Korrespondenz im Mapping erhalten bleibt ($\{(a,a)\}$). Die Ähnlichkeitswerte sind im Beispiel nicht abgebildet und müssen natürlich entsprechend aggregiert werden. Die Anwendung von *extendMatch* bestimmt zunächst die zwei Delta-Ontologien: $\Delta O1 = \{d,e\}$, $\Delta O2 = \{d,f\}$ (Zeile 1-2), die anschließend durch Anwendung von `match` abgeglichen werden (Zeile 3). Das ermittelte direkte Mapping ($DM_{\Delta O1,\Delta O2} = \{(d,d)\}$) wird mit dem kombinierten Eingabemapping ($CM_{O1,O2}$) vereinigt, so dass $\{(a,a),(b,b),(c,c),(d,d)\}$ als finales Mapping ausgegeben wird.

## 9.3 Evaluierung

### 9.3.1 Datensätze und Konfigurationen

Für die Evaluierung wird die *Adult Mouse Anatomy*-Ontologie (MA) mit dem anatomischen Teil des NCI Thesaurus (NCITa) abgeglichen. Diese Match-Aufgabe ist Teil des OAEI-Wettbewerbs, so dass ein Referenzmapping frei zur Verfügung steht und für die Evaluierung der Mappingqualität (Precision, Recall, F-Measure) verwendet werden kann. Diese Untersuchung basiert auf dem 2010 veröffentlichten OAEI-Datensatz. Seitdem wurden kleinere Modifikationen des perfekten Mappings vorgenommen, so dass sich (in diesem Kapitel) Verweise auf bisherige OAEI-Ergebnisse auf das Jahr 2010 oder früher beziehen. Die Mappingkomposition wird mithilfe der großen Zwischenontologien FMA, Uberon, Radiology Lexicon (RadLex) [113], sowie UMLS [16] unter Verwendung der Ende 2010 zur Verfügung stehenden Versionen durchgeführt.

Abbildung 9.3 fasst die statistischen Eigenschaften der verwendeten Ontologien und Mappings zusammen. Die Ontologien (Abbildung 9.3(a)) unterscheiden sich signifikant bezüglich der Konzeptanzahl ($|C|$) sowie der Anzahl der verfügbaren Namen und Synonyme pro Konzept ($\varnothing NameSyn$). Alle Zwischenontologien sind deutlich größer als MA und NCITa. Die für den Algorithmus *composeMatch* verwendeten Ontologiemappings wurden einmalig durch Bestimmung der linguistischen Ähnlichkeit auf Basis der Konzeptnamen und -synonyme (*NameSyn*: Trigram, $sim \geq 0,8$) bestimmt. Ein Vorverarbeitungsschritt umfasste zudem typische Normalisierungsschritte wie die Entfernung von Satz- und Trennzeichen, eine Transformation zur Kleinschreibung sowie die Eliminierung von Stoppwörtern. Folglich werden in dieser Evaluierung automatisch bestimmte, anstelle manuell verifizierter Mappings kombiniert, was die Bestimmung qualitativ hochwertiger Mappings erschwert.

Abbildung 9.3(b) zeigt signifikante Unterschiede bezüglich der Mappingabdeckung ($Cov$) und -größen ($|Map|$). Mappings zu UMLS und Uberon decken bis zu 80% von MA und NCITa ab. RadLex erreicht hingegen nur ungefähr 40%, so dass diese Zwischenontologie für zahlreiche Konzepte keine Korrespondenzen zur Verfügung stellen kann. FMA-Mappings zeigen eine mittelmäßige Abdeckung, was durch die vergleichsweise geringe, durchschnittliche Anzahl von Namen und Synonymen pro Konzept bedingt sein kann (Abbildung 9.3(a)) und die Qualität des linguistischen Match-Verfahrens reduziert. Im Gegensatz dazu ist Uberon, aufgrund seines hohen $\varnothing NameSyn$-Werts, eine vielversprechende Mediatorontologie. Generell wird erwartet, dass Ontologien mit zahlreichen Synonymen geeignete Mediatorontologien bezüglich linguistischer Match-Techniken darstellen.

Die Ausführung des direkten Matchings (`match`) im *extendMatch*-Algorithmus besteht wie zuvor aus einer Vorverarbeitung sowie einer Ähnlichkeitsberechnung auf Basis des linguistischen *NameSyn*-Match-Verfahrens. Darüber hinaus werden alle

(a)

| | \|C\| | \|ØNameSyn\| |
|---|---|---|
| **MA** | 2,738 | 1.1 |
| **NCITa** | 3,298 | 2.5 |
| **Uberon** | 4,958 | 4.9 |
| **UMLS** | 87,913 | 3.1 |
| **RadLex** | 30,773 | 1.6 |
| **FMA** | 81,059 | 1.8 |

(b)

| Mapping | $Cov_{Source}$ | $Cov_{Target}$ | \|Map\| |
|---|---|---|---|
| **MA-Uberon** | 80% | 45% | 2300 |
| **Uberon-NCITa** | 33% | 48% | 1703 |
| **MA-UMLS** | 69% | 3% | 2975 |
| **UMLS-NCITa** | 5% | 87% | 4214 |
| **MA-RadLex** | 39% | 3% | 1082 |
| **RadLex-NCITa** | 4% | 40% | 1347 |
| **MA-FMA** | 57% | 2% | 1601 |
| **FMA-NCITa** | 3% | 67% | 2337 |

Abbildung 9.3: (a) Statistiken für Ontologien und (b) Mappings.

finalen Mappings einer Nachbearbeitung unterzogen. Diese umfasst eine MaxDelta-Selektion sowie die Eliminerung widersprüchlicher Korrespondenzen, welche die CrissCross-Bedingung nicht erfüllen (siehe Kapitel 2.3.1).

### 9.3.2 Ergebnisse des kompositionsbasierten Matchings

Zunächst sollen die Qualität der mithilfe des *composeMatch*-Algorithmus bestimmten Mappings und der Einfluss von *extendMatch* evaluiert werden. Abbildung 9.4(a) zeigt die Qualität der indirekt bestimmten Mappings bezüglich F-Measure und vergleicht diese mit der Qualität des direkten Matchings von MA und NCITa (als „no IO“ bezeichnet). Das direkte Matching erreicht ein F-Measure von 86%, was mit den besten Ergebnisse der vorherigen OAEI-Wettbewerbe vergleichbar ist (87,7%)[33]. Die Qualität der kombinierten Mappings hängt stark von der verwendeten Mediatorontologie und den assoziierten Mappings ab. Die besten F-Measure-Werte werden durch Komposition über UMLS (86,2%) und Uberon (84,7%) erreicht. Die Qualität des UMLS-basierten Mappings übersteigt somit die Qualität des direkten Matchings. Die auf FMA und RadLex basierenden Ontologiemappings weisen eine wesentlich niedrigere Qualität von 77% bzw. 59% auf. Dies begründet sich durch die niedrige Mappingabdeckung bezüglich MA und NCITa. RadLex fokussiert nicht primär auf die Repräsentation von Anatomiewissen. Hingegen stellt Uberon eine speziesübergreifende Anatomieontologie dar und ein großer Teil von UMLS deckt anatomisches Wissen ab. Uberon und UMLS eignen sich also sehr gut für das indirekte Matching von Anatomieontologien. Die dunkelgrauen Balken in Abbildung 9.4(a) zeigen die erreichte Mappingqualität, wenn zusätzlich der *extendMatch*-Algorithmus angewendet wird. Dieser zusätzliche Schritt führt für jede Zwischenontologie zu einer Verbesserung der Qualität. Interessanterweise erreicht nun Uberon die beste Qualität (88,2%) und übersteigt damit UMLS (87,0%) sowie die bis dahin besten OAEI-Ergebnisse. Dies zeigt, dass die Komposition über Uberon nicht-triviale Korrespondenzen findet,

[33] 87,7% ist das bis zur OAEI2010 beste, erreichte Ergebnis im *Anatomy Track*.

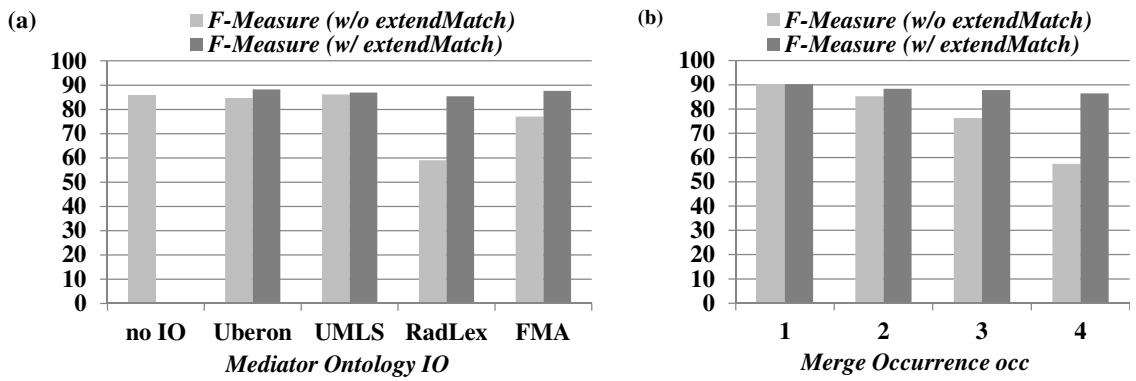



Abbildung 9.4: Mappingqualität bezüglich F-Measure (a) Komposition über verschiedene Zwischenontologien (b) Zusammenfassung mehrerer kombinierter Mappings.

die durch ein direktes Matching nicht identifiziert werden. Der zusätzliche Aufwand für *extendMatch* verbessert die Mappingqualität insbesondere für Zwischenontologien, die vergleichsweise schlechte Kompositionsergebnisse erreichen (z. B. RadLex und FMA).

Nun soll untersucht werden, inwieweit die Kombination mehrerer Mappings zu verschiedenen Zwischenontologien die Qualität verbessern kann. Abbildung 9.4(b) zeigt die F-Measure-Werte für die Zusammenfassung (`merge`) kombinierter Mappings für verschiedene *occ*-Werte. *occ* gibt an, wie häufig eine Korrespondenz in den einzelnen Mappings vorkommen muss. Die Vereinigung ($occ = 1$) mehrerer kombinierter Mappings verbessert die Mappingqualität auf bis zu 90,2% F-Measure (Recall 87,7%, Precision 92,7%). Dies übertrifft die Qualität des direkten Matchings sowie die bis dahin besten OAEI-Ergebnisse. Die Schnittmenge der Mappings ($occ$=4) erweist sich infolge einer signifikanten Reduktion des Recalls als weniger effektiv (F-Measure 57,4%). Dabei wird der Vorteil, dass sich Korrespondenzen durch Verwendung unterschiedlicher Zwischenontologien gegenseitig ergänzen, nicht mehr genutzt. Die zusätzliche Anwendung von *extendMatch* verändert die Ergebnisse für $occ = 1$ kaum (F-Measure 90,3%). Für höhere *occ*-Werte kann die Qualität hingegen signifikant verbessert werden. Die Vereinigung kombinierter Mappings kann also ohne zusätzliches Matching sehr gute Ergebnisse produzieren. Keiner der bisherigen Teilnehmer[34] des OAEI *Anatomy Track* konnte ein F-Measure von ≈87% übersteigen, so dass die Erhöhung auf über 90% eine signifikante Verbesserung darstellt.

Die Ausführungszeiten des Matchings[35] konnten signifikant reduziert werden. Die Komposition der Mappings zu allen vier Zwischenontologien sowie der anschließende `merge`-Schritt benötigten nur 2,8 s. Die zusätzliche Ausführung von *extendMatch* benötigte im Durchschnitt nur 36 s. Im Gegensatz dazu umfasste die Ausführungs-

[34] 2010 oder früher.
[35] Ohne Parsen der Ontologien und Mappings.

zeit des direkten Matchings der vollständigen Ontologien (MA und NCITa) 116 s. Folglich konnte die Laufzeit bei gleichbleibender oder verbesserter Mappingqualität um bis zu Faktor 41 reduziert werden.

## 9.4 Zusammenfassung

Dieses Kapitel stellte einen kompositionsbasierten Ansatz zum indirekten Ontologie-Matching über eine oder mehrere Zwischenontologien vor. Ziel war es, existierende Ontologiemappings wiederzuverwenden, um eine verbesserte Mappingqualität sowie reduzierte Ausführungszeiten zu erreichen. Der Ansatz basiert auf den Ontologie- und Mappingoperatoren `compose`, `match`, `extract` sowie `merge` und erlaubt eine flexible Aggregation mehrerer kombinierter Mappings. Um zusätzliche Korrespondenzen für bis dahin nicht verknüpfte Konzepte zu finden, können Mappings inkrementell durch Anwendung eines direkten Matchings erweitert werden.

Die Evaluierung für das Matching der Anatomieontologien MA und NCITa betrachtete die vier Zwischenontologien UMLS, FMA, Uberon und RadLex. Insgesamt konnte eine sehr gute Mappingqualität von über 90% F-Measure erreicht werden. Die Anwendung eines zusätzlichen direkten Matchings mit *extendMatch* ist generell hilfreich zur Verbesserung der Mappingqualität. Im Vergleich zu einem vollständigen, direkten Matching übertrifft die alleinige Ausführung der Komposition die Mappingqualität und reduziert die Ausführungszeiten deutlich. Dies gilt insbesondere für die Zusammenfassung mehrerer kombinierter Mappings zu verschiedenen Zwischenontologien. Innerhalb der Anatomiedomäne eignen sich insbesondere Uberon und UMLS als Mediatorontologie.

Das System GOMMA wurde im Rahmen dieser Arbeit um den hier präsentierten kompositionsbasierten Ansatz zum indirekten Ontologie-Matching erweitert. Der Ansatz wurde während der Evaluierung von GOMMA im Rahmen der OAEI 2012 genutzt (siehe Kapitel 11). Seit 2010 wurde das Referenzmapping des *Anatomy Tracks* modifiziert, so dass GOMMA mit dem kompositionsbasierten Ansatz ein F-Measure von 92,8% bezüglich des *Anatomy* - Referenzmappings der OAEI 2012 erreicht.

# 10

# Paralleles Ontologie-Matching

## 10.1 Motivation

Typischerweise erfordert die automatische Berechnung von qualitativ hochwertigen Ontologiemappings eine kombinierte Ausführung mehrerer metadaten- und/oder instanzbasierter Match-Verfahren, um Ähnlichkeitswerte zwischen Ontologiekonzepten zu bestimmen (siehe Kapitel 2.3.1). Derartige Matchworkflows sind insbesondere für sehr große Ontologien zeitaufwendig und speicherintensiv. Üblicherweise werten Match-Verfahren das kartesische Produkt aller Konzeptpaare zweier Ontologien aus, was zu einer quadratischen Komplexität bezüglich der Ontologiegröße führt. Die Performanzanforderungen vervielfachen sich zusätzlich durch die Anzahl der ausgeführten Match-Verfahren und der zu lösenden Match-Aufgaben (z. B. Matching mehrerer Ontologieversionen). Zudem ist Ontologie-Matching für sehr große Ontologien speicherintensiv, da die Berechnung typischerweise auf Graphstrukturen der Ontologien im Hauptspeicher ausgeführt wird und mehrere Ähnlichkeitswerte für jedes Konzeptpaar des kartesischen Produkts bestimmt werden müssen. Die Ergebnisse früherer OAEI-Wettbewerbe zum Matching von Anatomieontologien[36] zeigen, dass die Systeme teilweise mehrere Stunden zur Ausführung benötigen. Dies ist der Fall, obwohl die betrachteten Ontologien MA und NCITa nur eine mittlere Größe von circa 3.000 Konzepten aufweisen. Das kartesische Produkt umfasst folglich $\approx 9 \cdot 10^6$ Konzeptpaare. Der Abgleich der GO-Subontologien Molekulare Funktionen (MF) und Biologische Prozesse (BP) mit je ≈10.000 und ≈20.000 Konzepten resultiert hingegen in $\approx 2 \cdot 10^8$ Vergleichen, was dem 22-fachen des Anatomie-Match-

[36] z. B. OAEI 2008: `http://oaei.ontologymatching.org/2008/results/anatomy`

Problems entspricht. Die Speicheranforderungen für die Ähnlichkeitsberechnungen betragen daher bereits mehrere Gigabyte. Viele der an der OAEI teilnehmenden Systeme haben sich in den letzten Jahren deutlich bezüglich der erreichten Laufzeit verbessert [48]. Dennoch haben weiterhin viele Systeme Probleme, insbesondere sehr große Match-Aufgaben wie den *Large BioMed Track*[37] zu lösen. Beispielsweise war zur OAEI 2012 nur circa ein Drittel der Systeme in der Lage, die sehr großen Ontologien FMA, NCIT und SNOMED CT (80.000-300.000 Konzepte) abzugleichen. Aufgrund der sehr hohen Ressourcenanforderungen skalieren viele Systeme nicht für das Matching derartig großer Ontologien.

Das effiziente Matching von Ontologien ist z. B. für aufwendige Tests der Match-Konfiguration oder interaktive Anwendungen wertvoll, so dass Nutzer keine langen Wartezeiten in Kauf nehmen müssen. Die Verbesserung der Performanz wurde in verschiedenen Arbeiten im Bereich des Schema- und Ontologie-Matchings untersucht (siehe Kapitel 2.3.2). Die parallele Ausführung von Ontologie-Matching auf mehreren Rechenknoten wurde jedoch erstmals im Rahmen dieser Arbeit realisiert.

Die hohe Verfügbarkeit von Mehrkernprozessoren sowie die gleichzeitige Nutzbarkeit mehrerer Rechner motivieren das parallele Matching von Ontologien zur Reduktion der Ausführungszeiten. Die Partitionierung einer großen Match-Aufgabe in kleinere, parallel ausführbare Teilaufgaben reduziert zudem die Speicheranforderungen pro Teilaufgabe. Aus diesen Gründen sollen Strategien zur parallelen Ausführung des Ontologie-Matchings untersucht werden. Das Kapitel umfasst die folgenden Beiträge:

- Es werden verschiedene Strategien zum parallelen Ontologie-Matching, insbesondere die *Inter-* und *Intra-Matcher-Parallelisierung* vorgestellt. Während der erste Ansatz unabhängige Matcher parallel ausführt, realisiert der zweite eine interne Parallelisierung der Matcher basierend auf der Partitionierung der Eingabeontologien. Zur weiteren Verbesserung der Performanz können beide Strategien kombiniert werden.
- Die Parallelisierung wird für verschiedene Arten von Match-Verfahren insbesondere element-, struktur- und instanzbasierte Matcher gezeigt.
- Die Implementierung einer verteilten Infrastruktur zum parallelen Ontologie-Matching realisiert die vorgestellten Ansätze, die für große Ontologien aus dem Bereich der Lebenswissenschaften evaluiert werden. Die Ergebnisse zeigen die Effizienz und Skalierbarkeit für einzelne Match-Verfahren sowie kombinierte Match-Strategien.

[37] OAEI *Large BioMed Track*: `http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2012/`

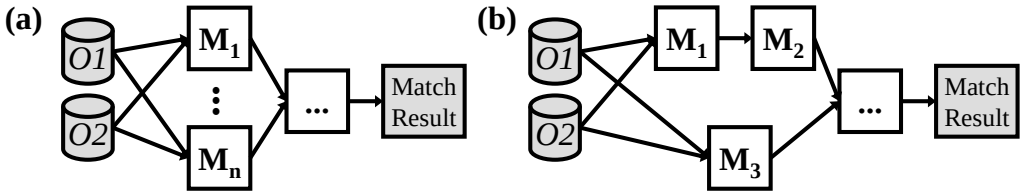



Abbildung 10.1: (a) Inter-Matcher-Parallelisierung (b) Kombination der Inter-Matcher-Parallelisierung mit einer sequentiellen Matcher-Ausführung.

## 10.2 Parallelisierungsstrategien

In diesem Abschnitt werden Möglichkeiten zur Parallelisierung von Match Workflows diskutiert, die aus mehreren sequenziell oder unabhängig ausgeführten Matchern (Match-Verfahren) bestehen. Dies geschieht unter der Annahme, dass für das Ontologie-Matching eine Umgebung aus mehreren Rechenknoten mit Mehrkernprozessoren zur Verfügung steht.

Ein naheliegender Ansatz ist die Nutzung der *Inter-Matcher-Parallelität* zum Matching von Ontologien (Kapitel 10.2.1). Dabei sollen unabhängig ausführbare Matcher parallel auf verschiedenen Rechenknoten oder Kernen (*engl. cores*) abgearbeitet werden. Zudem soll *Intra-Matcher-Parallelität*, d. h. die interne Parallelisierung individueller Matcher, unterstützt werden. Beide Arten von Parallelität können außerdem miteinander kombiniert werden. Die Intra-Matcher-Parallelität (Kapitel 10.2.2) fokussiert auf die parallele Ähnlichkeitsberechnung zwischen Konzeptpaaren des kartesischen Produkts auf Basis der Partitionierung der Eingabeontologien. Insbesondere soll die Parallelisierung von element-, struktur- und instanzbasierten Match-Verfahren beschrieben werden. Im Folgenden werden die Parallelisierungsstrategien detaillierter diskutiert.

### 10.2.1 Inter-Matcher-Parallelisierung

Inter-Matcher-Parallelisierung ermöglicht die parallele Ausführung unabhängiger Match-Verfahren, so dass mehrere Prozessoren für eine schnellere Verarbeitung des Match-Prozesses genutzt werden können. Der beispielhafte Workflow in Abbildung 10.1(a) verwendet Inter-Matcher-Parallelisierung für $n$ Matcher $(M_1, \ldots, M_n)$. Um das finale Ergebnis zu erhalten, können die einzelnen Match-Ergebnisse durch verschiedene Aggregations- und Selektionsstrategien kombiniert werden. Die Inter-Matcher-Parallelisierung verbessert die Ausführungszeit um Faktor $n$, falls die Matcher idealerweise eine ähnliche Komplexität aufweisen. Diese Art von Parallelisierung kann relativ leicht unterstützt werden, indem mehrere Kerne eines einzelnen Rechenknotens oder mehrere Rechenknoten verwendet werden. Allerdings ist die Inter-Matcher-Parallelisierung durch die Anzahl unabhängig ausführbarer Mat-

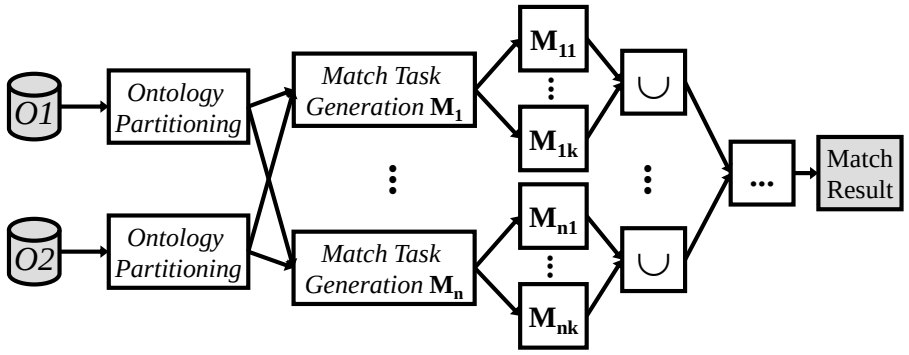



Abbildung 10.2: Intra-Matcher-Parallelisierung.

cher begrenzt. Weiterhin können Matcher unterschiedlicher Komplexität sehr unterschiedliche Ausführungszeiten haben. Dies limitiert den erreichbaren Speedup, da der langsamste Matcher die Gesamtausführungszeit bestimmt. Da die Match-Verfahren vollständige Ontologien auswerten, reduzieren sich die Speicheranforderungen des Matchings nicht.

Der Grad der Parallelität ist eingeschränkt, wenn Match-Verfahren sequenziell ausgeführt werden müssen (z. B. wenn ein strukturbasierter Matcher von einem zuvor ausgeführten elementbasierten Matcher abhängt) oder wenn die Anzahl der verfügbaren Prozessoren kleiner als die Anzahl der unabhängig ausführbaren Matcher ist. In solchen Fällen kann die Inter-Matcher-Parallelität für eine Teilmenge der Matcher angewendet werden. Das Beispiel in Abbildung 10.1(b) nimmt an, dass nur zwei Kerne verwendet werden können und dass der komplexeste Matcher $M_3$ einem Kern zugewiesen wird, wohingegen $M_1$ und $M_2$ sequentiell auf dem anderen Kern ausgeführt werden.

### 10.2.2 Intra-Matcher-Parallelisierung

Intra-Matcher-Parallelisierung behandelt die interne Zerlegung individueller Matcher oder Teile der Matcher (z. B. die Tokenisierung von Konzeptnamen) in mehrere Teilaufgaben, die dann in einzelnen *Match Tasks* parallel verarbeitet werden können. Diese Studie fokussiert auf einen generellen Ansatz zur Unterstützung von Intra-Matcher-Parallelität basierend auf der Partitionierung der Eingabedaten (Ontologien). Eine solche Partitionierung ist sehr flexibel und skalierbar und kann zur Generierung zahlreicher Match Tasks genutzt werden. Die einzelnen Match Tasks sind dann weniger komplex als die vollständige Evaluierung des kartesischen Produkts. Weiterhin kann Intra-Matcher-Parallelisierung für sequentielle sowie parallel ausführbare Matcher angewendet werden, d. h. sie kann auch mit der Inter-Matcher-Parallelisierung kombiniert werden.

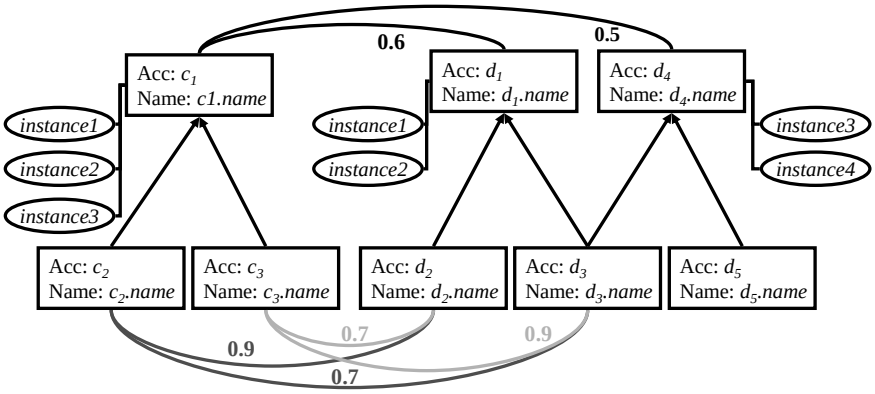
Abbildung 10.2 veranschaulicht die Intra-Matcher-Parallelisierung für $n$ Matcher (d. h. in Kombination mit Inter-Matcher-Parallelität). Für jeden Matcher erfolgt zunächst die Partitionierung der Eingabeontologien $O1$ und $O2$. Anschließend werden mehrere Match Tasks $M_{i1}, \ldots, M_{ik}$ $(i = 1, \ldots, n)$ generiert und parallel ausgeführt, wobei $k$ der Anzahl der generierten Tasks pro Matcher entspricht. Die Vereinigung aller Teilergebnisse ergibt das vollständige Match-Ergebnis. Dabei gleichen die einzelnen Match Tasks nur Partitionen von $O1$ und $O2$ ab und haben folglich geringere Speicheranforderungen und einen niedrigeren Berechnungsaufwand im Vergleich zum Matching der vollständigen Ontologien. Die Intra-Matcher-Parallelisierung eignet sich also besonders gut zum Matching sehr großer Ontologien.

Bevor die Parallelisierung von element-, struktur- und instanzbasierten Match-Verfahren diskutiert wird, soll zunächst der Ansatz zur *Ontologiepartitionierung* vorgestellt werden. Diese initiale Studie zum parallelen Ontologie-Matching konzentriert sich auf einen einfachen aber flexiblen, größenbasierten Ansatz, welcher das parallele Matching des kartesischen Produkts der $O1$- und $O2$-Konzepte ermöglicht. Zur Generierung ähnlich komplexer Match Tasks werden beide Eingabeontologien in Partitionen gleicher Größe (Anzahl der Konzepte) zerlegt. Der Parameter zur Festlegung der Partitionsgröße kann, abhängig von der Größe der Eingabeontologien und der Komplexität der verwendeten Matcher, konfiguriert werden. Jeder Match Task gleicht eine $O1$-Partition mit einer $O2$-Partition ab, so dass $p_1 \cdot p_2$ Match Tasks für $p_1$ ($p_2$) gleich große Partitionen aus $O1$ ($O2$) generiert werden. Wenn beispielsweise zwei Ontologien mit je 10.000 Konzepten in jeweils 10 Partitionen zerlegt werden, entstehen $10 \cdot 10{=}100$ Match Tasks. Diese werden in einer Warteschlange verwaltet und für die parallele Ausführung zugeteilt (siehe Infrastruktur in Kapitel 10.3).

Die größenbasierte Partitionierung hat einige signifikante Vorteile: (1) sie ist skalierbar für große Ontologien, indem überschaubare Partitionsgrößen gewählt werden können, so dass die Verarbeitung folglich unproblematisch ist und die Speicheranforderungen pro Match Task reduziert sind; (2) wird aufgrund gleich großer Partitionen und Match Tasks eine gute Lastbalancierung unterstützt; (3) wird die Performanz optimiert, ohne Verluste bezüglich der Qualität der Ergebnismappings akzeptieren zu müssen, da insgesamt das vollständige kartesische Produkt ausgewertet wird; und (4) kann die größenbasierte Partitionierung für element-, struktur- und instanzbasierte Matcher verwendet werden. Dies soll im Folgenden diskutiert werden.

### Parallelisierung von elementbasierten Match-Verfahren

Die Parallelisierung von elementbasierten Match-Verfahren ist relativ einfach und basiert auf der zuvor eingeführten, größenbasierten Partitionierung. Elementbasierte Matcher vergleichen Ontologiekonzepte unter Verwendung der Metadaten der Konzepte selbst (z. B. Attributwerte wie Name und Synonyme). Durch Partitionierung der Ontologien in Konzeptteilmengen bleiben somit alle benötigten Konzept-




| corr(c,d) | $sim_{Name}(c,d)$ |
|---|---|
| $c_1 - d_1$ | 0.6 |
| $c_1 - d_4$ | 0.5 |
| $c_2 - d_2$ | 0.9 |
| $c_2 - d_3$ | 0.7 |
| $c_3 - d_2$ | 0.7 |
| $c_3 - d_3$ | 0.9 |

Abbildung 10.3: Elementbasiertes Matching auf dem Namensattribut.

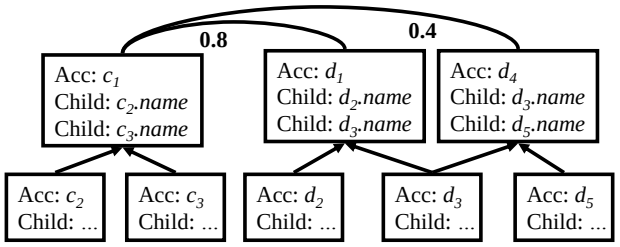
informationen erhalten. Elementbasierte Match-Verfahren lassen sich folglich relativ einfach auf Ontologiepartitionen anwenden.

Abbildung 10.3 zeigt ein Beispiel zum Matching zweier Ontologieteile $c_1, \ldots, c_3 \in O1$ und $d_1, \ldots, d_5 \in O2$. Konzept $c_1$ hat zwei Kinder $c_2$ und $c_3$. Konzept $d_3$ aus $O2$ hat zwei Elternkonzepte $d_1$, $d_4$ (Mehrfachvererbung). Einige Konzepte haben Assoziationen zu Instanzen, die später für das instanzbasierte Matching betrachtet werden. Die Konzepte sollen nun mit einem String-basierten Match-Verfahren verglichen werden. Der *Name*-Matcher evaluiert die String-Ähnlichkeit (z. B. Trigram) für alle $(3 \cdot 5 = 15)$ Konzeptpaare. Die Ergebnismenge (Abbildung 10.3 rechts) enthält sechs Korrespondenzen mit Ähnlichkeitswerten zwischen 0,5 und 0,9. Alle weiteren möglichen Konzeptpaare haben einen Ähnlichkeitswert von 0, d. h. sie bilden keine Korrespondenz.

### Parallelisierung von strukturbasierten Match-Verfahren

Die Parallelisierung strukturbasierter Match-Verfahren gestaltet sich komplexer, da die Verfahren, im Gegensatz zu elementbasierten Techniken, Informationen aus dem strukturellen Kontext der Konzepte (z. B. Kinder, Eltern, Geschwister) oder sogar die gesamte Ontologie benötigen. Eine Ontologiepartition besteht aus einer Teilmenge der Konzepte und enthält somit nicht alle für das Struktur-Matching erforderlichen Informationen. Noch schwieriger ist die Parallelisierung für iterative Struktur-Matcher wie Similarity Flooding [132], das mit initialen elementbasierten Ähnlichkeitswerten beginnt und diese iterativ, entlang der Konzeptbeziehungen, durch die gesamte Ontologiestruktur propagiert. Für derartige Match-Verfahren ist Parallelisierung grundsätzlich schwierig und muss vermutlich auf das initiale, elementbasierte Matching beschränkt werden.

Daher konzentriert sich die folgende Diskussion auf Struktur-Matcher, die Informationen aus einem begrenzten Umfeld, d. h. einem lokalen Kontext, nutzen. Um




| $corr(c,d)$ | $sim_{Children}(c,d)$ |
|---|---|
| $c_1 - d_1$ | (0.9 + 0.7 + 0.7 + 0.9) / (2·2) = 0.8 |
| $c_1 - d_4$ | (0.7 + 0 + 0 + 0.9) / (2·2) = 0.4 |

Abbildung 10.4: Attributbasierter *Children*-Matcher.

Ressourcen- und Speicheranforderungen zu minimieren, sollen die Match Tasks nicht auf vollständigen Ontologien arbeiten, weshalb die Größe der Partitionen ähnlich wie für das parallele Element-Matching beschränkt sein soll. Dazu werden konzeptassoziierte Informationen durch Verwendung spezieller mehrwertiger Attribute um den lokalen *Kontext* erweitert, der für das strukturbasierte Matching erforderlich ist. Die jeweils für die Kontextattribute benötigten Werte, z. B. Namen und Synonyme von Kindern/Eltern oder Namenspfade (*NamePath*), werden in einem Vorverarbeitungsschritt durch einmaliges Traversieren der Eingabeontologien (linearer Aufwand) aufgesammelt. Konzepte mit ihren zusätzlichen Kontextattributen können dann wie für das elementbasierte Matching partitioniert werden. Jeder Match Task führt Struktur-Matching für ein Paar von Partitionen aus, wobei die Informationen der Kontextattribute verwendet werden.

Abbildung 10.4 veranschaulicht den Ansatz für Kontextattribute anhand eines Match-Verfahrens, das Konzepte auf Basis der Ähnlichkeit ihrer Kinder abgleicht (*Children*-Matcher). Dazu wird das Beispiel aus Abbildung 10.3 genutzt. Der verwendete *Children*-Matcher bestimmt die Ähnlichkeit zweier Konzepte, in dem die durchschnittliche elementbasierte Ähnlichkeit ihrer Kinder berechnet wird. Um dieses Match-Verfahren parallel ausführen zu können, wird ein mehrwertiges *Child*-Kontextattribut für jedes Konzept, das kein Blatt ist, verwendet. Die Kontextattribute werden im Vorverarbeitungsschritt mit Werten befüllt, z. B. mit den Namen der jeweiligen Kinder. Anschließend gleicht ein Match Task jedes Konzept $c$ einer $O1$-Partition mit jedem Konzept $d$ einer $O2$-Partition ab, indem alle *Child*-Attribute von $c$ mit allen *Child*-Attributen von $d$ bezüglich ihrer String-/Namensähnlichkeit verglichen und durch die Anzahl der möglichen Kinderpaare geteilt werden:

$$sim_{Children}(c,d) = \frac{\sum_{i,j} sim_{Name}(c.child_i, d.child_j)}{(|c.child| \cdot |d.child|)}$$

Dies ist nur eine von verschiedenen Möglichkeiten, die Ähnlichkeit zweier Konzepte unter Verwendung der Ähnlichkeit der Kinderkonzepte zu berechnen. Im Beispiel in Abbildung 10.4 ist $c_1$ ähnlicher zu $d_1$ als zu $d_4$, da $c_1$ und $d_1$ ähnlichere Kinder haben (unter Verwendung der Ähnlichkeitswerte in Abbildung 10.3).

Der Ansatz zur Verwendung von Kontextattributen kann in ähnlicher Weise für andere Arten lokalen Kontexts wie z. B. Eltern, Geschwister oder Namenspfade ein-

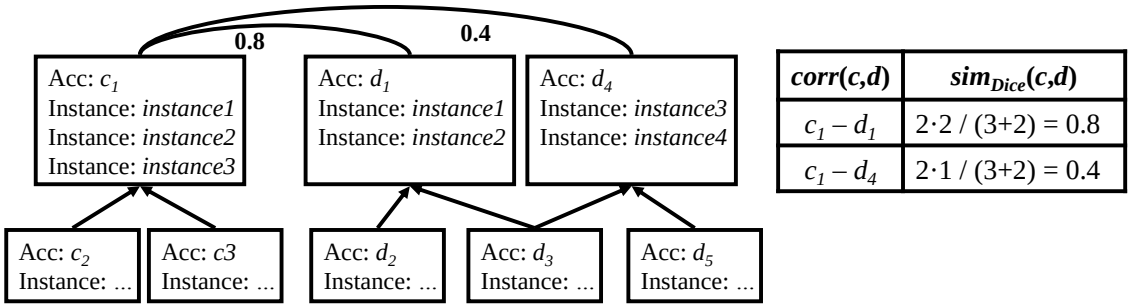



Abbildung 10.5: Attributbasiertes instanzbasiertes Matching.

gesetzt werden. Um beispielsweise den *NamePath*-Matcher zu realisieren, werden in der Vorverarbeitung die Namen der Vorgängerkonzepte auf dem Weg zur Wurzel bestimmt und als konkatenierter *NamePath* in einem mehrwertigen Attribut abgelegt. Das Matching ähnelt dann dem elementbasierten Matching, allerdings werden *NamePath*-Attribute und strukturelle Informationen zu den Namen der Vorgängerkonzepte verwendet. Frühere Evaluierungen [42] zeigten, dass *NamePath* eines der effektivsten individuellen Match-Verfahren ist, so dass es sehr nützlich ist, eine parallele *NamePath*-Implementierung zu haben.

### Parallelisierung von instanzbasierten Match-Verfahren

Abschließend wird die Parallelisierung von instanzbasierten Match-Verfahren diskutiert. Dabei werden Instanzen in der Vorverarbeitung auf mehrwertige *Instance*-Attribute abgebildet. Ein üblicher Ansatz wertet die zu Konzepten assoziierten Instanzen aus und betrachtet zwei Konzepte als ähnlich, wenn sie möglichst viele, gemeinsame bzw. ähnliche Instanzen aufweisen. Da Instanzen direkt zu Konzepten assoziiert sind, kann die Ähnlichkeit zwischen Konzepten anhand konzeptspezifischer Informationen berechnet werden. Dadurch kann eine ähnliche Parallelisierungsstrategie wie für elementbasiertes Matching und das Matching des lokalen Kontexts angewendet werden.

Abbildung 10.5 zeigt, wie Instanzen auf mehrwertige *Instance*-Attribute abgebildet werden. *Instance* kann beispielsweise die *Accessions* biologischer Objekte enthalten, die mit Ontologiekonzepten annotiert wurden. Die größenbasierte Partitionierung wird auf die Eingabeontologien sowie die assoziierten Instanzen angewendet. Anschließend kann z. B. das *Dice*-Maß eingesetzt werden. Dazu kann die Anzahl der gemeinsamen *Instance*-Attributwerte $N_{cd}$ zweier Konzepte $c \in O1$ und $d \in O2$ ermittelt werden, um dann die Ähnlichkeit $sim_{Dice}(c,d) = 2 \cdot N_{cd}/(N_c + N_d)$ zu berechnen, wobei $N_c$ ($N_d$) der Anzahl der Instanzen des Konzepts $c$ ($d$) entspricht. Im Beispiel (Abbildung 10.5) enthält das Ergebnis zwei Korrespondenzen. Dabei hat die Korrespondenz $c_1$-$d_1$ mehr gemeinsame Instanzen als $c_1$-$d_4$, wodurch sich ein höherer Ähnlichkeitswert ergibt.

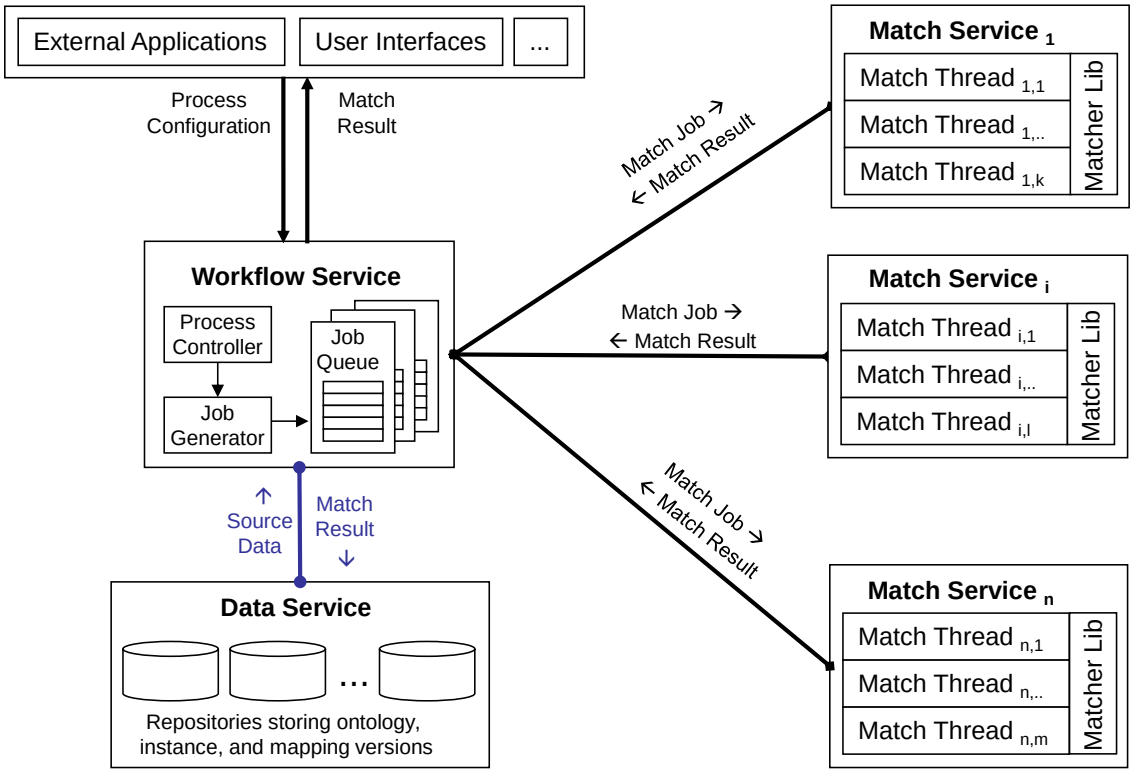



Abbildung 10.6: Verteilte Infrastruktur zum parallelen Ontologie-Matching.

## 10.3 Verteilte Infrastruktur zum parallelen Ontologie-Matching

Um komplexere Match Workflows parallel ausführen zu können, wurde eine verteilte, servicebasierte Infrastruktur implementiert (siehe Abbildung 10.6). Diese umfasst verschiedene Dienste: einen Workflow-Dienst (*Workflow Service*), einen Datendienst (*Data Service*) und mehrere Match-Dienste (*Match Services*). Die Dienste laufen auf verschiedenen, lose gekoppelten Servern und Arbeitsplatzrechnern. Der *Workflow Service* koordiniert die Ausführung des gesamten Match Workflows. *Match Services* berechnen Ontologiemappings für zwei Ontologien oder Ontologiepartitionen. Der *Data Service* implementiert das Repository-Schema zur effizienten Verwaltung von Ontologie- und Mappingversionen aus [103] (siehe GOMMA Kapitel 3.2). Er verwaltet die Ontologie- und Instanzdaten, welche die Eingabe eines Match Workflows darstellen, und speichert das finale Ontologiemapping als Ergebnis ab. Die gesamte Infrastruktur wurde in Java implementiert.

Match-Systeme wie GOMMA [102] nutzen den *Workflow Service*, um zentral auf die Infrastruktur zugreifen zu können. Es wird angenommen, dass derartige Anwendungen einen konkreten Match Workflow konfigurieren, d. h. sie spezifizieren die

Ontologien und Instanzdaten (oder deren Versionen) als Eingabedaten, die verwendeten Match-Verfahren und die Schritte zur Vorverarbeitung der Eingabedaten (z. B. Ontologiepartitionierung) sowie Nachbearbeitung der Ergebnisse (z. B. Vereinigung von Teilergebnissen, Selektion). Diese Spezifikation definiert folglich alle notwendigen Schritte des Matchings, insbesondere welche Matcher parallel oder sequentiell ausgeführt werden. Der *Workflow Service* nimmt diese Konfiguration als Eingabe und arbeitet den vorgegebenen Match Workflow ab.

Der *Workflow Service* realisiert die Vorverarbeitung der Ontologien. Beispielsweise müssen für das strukturbasierte Matching die Werte der Kontextattribute (siehe Kapitel 10.2.2) bestimmt werden. Der *Workflow Service* führt die Matcher des Workflows parallel oder sequentiell in der definierten Reihenfolge aus. Zu diesem Zweck wird eine Job-Warteschlange für jeden Matcher verwaltet. Für die Intra-Matcher-Parallelisierung generiert der *Workflow Service* alle Match Tasks und speichert sie in einer Matcher-spezifischen Job-Warteschlange. Wenn keine Intra-Matcher-Parallelisierung realisiert wird, besteht die Warteschlange aus genau einem Job pro Matcher. Solange sich unbearbeitete Jobs in der Warteschlange befinden, sendet der *Workflow Service* die anstehenden Jobs an verfügbare bzw. frei werdende *Match Services*. Die *Match Services* führen die Jobs aus und senden ihre Ergebnisse zurück zum *Workflow Service*, der die einzelnen Ergebnisse vereinigt. Aus Gründen der Effizienz werden die Match Jobs durch einen Ähnlichkeitsgrenzwert beschränkt, so dass nur Korrespondenzen zurückgegeben werden, die eine minimale Ähnlichkeit überschreiten.

Die *Match Services* laufen auf dedizierten Knoten, um deren Rechenleistung voll ausschöpfen zu können. Jeder *Match Service* enthält mehrere parallel arbeitende *Match Threads*, welche die Match Jobs (jeweils ein Thread pro Job) ausführen. Die Anzahl der Match Threads pro Service ($k$, $l$, $m$ in Abbildung 10.6) kann entsprechend der Anzahl verfügbarer Kerne variieren. Folglich kann die Infrastruktur mit heterogen konfigurierten Rechnersystemen umgehen, d. h. es können Server sowie übliche Arbeitsplatzrechner mit unterschiedlicher Rechenleistung und Kernanzahl eingesetzt werden. Die Match Threads erhalten ihre Eingabedaten (Ontologiepartitionen) vom Match Job und führen die festgelegte Matcher-Implementierung aus einer *Matcher Library* aus.

## 10.4 Evaluierung

Zur Evaluierung der vorgestellten Parallelisierungsstrategien wird die verteilte Infrastruktur für das Matching von Ontologien eingesetzt. Kapitel 10.4.1 beschreibt die betrachteten Ontologien und Match-Verfahren. Kapitel 10.4.2 zeigt Ergebnisse des parallelen Matchings auf einem Rechenknoten mit Mehrkernprozessor. Anschließend wird die Skalierbarkeit des parallelen Ontologie-Matchings unter Verwendung mehreren Rechenknoten untersucht (Kapitel 10.4.3).

### 10.4.1 Datensätze und Konfigurationen

Die *Match Services* laufen auf bis zu vier Knoten mit je vier Kernen, d. h. es kommen bis zu 16 Kerne zum Einsatz. Jeder Knoten hat eine Intel(R) Xeon(R) W3520 4x2,66 GHz CPU mit 4GB Hauptspeicher sowie ein 64-Bit Debian GNU/Linux Betriebssystem mit einer 64-Bit JVM. Es werden 3 GB Hauptspeicher (*Heap*-Größe) pro Knoten verwendet. Der *Workflow* und *Data Service* laufen auf anderen zusätzlichen Knoten.

Für die Experimente werden ein Match-Problem mittlerer Größe und ein großes Match-Problem betrachtet. Als mittelgroßes Match-Problem dient der OAEI *Anatomy Task* mit den Ontologien MA (2.737 Konzepte) und NCITa (3.289 Konzepte). Das große Match-Problem berechnet ein Ontologiemapping zwischen GO-MF und GO-BP, die jeweils 9.395 und 17.104 Konzepte (Versionen vom Juni 2009) umfassen. Für die beiden Match-Probleme kommen unterschiedliche Partitionsgrößen für die Intra-Matcher-Parallelisierung zum Einsatz. Für das mittelgroße Problem wird die Partitionsgröße auf maximal 500 Konzepte gesetzt. Dies führt zu 6 (7) Partitionen für MA (NCITa) und folglich zu 42 Match Tasks. Für das große Match-Problem wird die Partitionsgröße auf 1.500 gesetzt, so dass MF (BP) in 7 (12) Partitionen unterteilt wird, was in 84 Match Tasks resultiert.

Diese erste Evaluierung zum parallelen Ontologie-Matching konzentriert sich auf element- und strukturbasierte Match-Verfahren. Es werden drei verschiedene einzelne Matcher betrachtet: *NameSyn*, *Children* und *NamePath*. *NameSyn* bestimmt wie zuvor die maximale Trigram-Ähnlichkeit zwischen den Namen und mehrwertigen Synonymattributen zweier Konzepte. *Children* und *NamePath* nutzen die Trigram-Ähnlichkeit jeweils auf den Kontextattributen *Child* und *NamePath* (siehe Kapitel 10.2.2). *NamePath* wird auf drei Ebenen bezüglich der Vorfahren eines Konzepts in der *is-a/part-of*-Hierarchie beschränkt. Diese Studie fokussiert auf die Evaluierung der Effizienz, d. h. der Ausführungszeiten, und nicht der Effektivität (z. B. Precision, Recall). Die Parallelisierung zielt nur auf die Verbesserung der Ausführungszeiten hin und beeinflusst nicht die Mappingqualität, da durch die größenbasierte Partitionierung immer das kartesische Produkt ausgewertet wird (siehe Kapitel 10.2.2). Eine Evaluierung der Qualität von Mappings, die automatisch mit GOMMA generiert wurden, findet sich in Kapitel 11.

### 10.4.2 Individuelle Matcher-Parallelisierung auf einem Knoten mit Mehrkernprozessor

Zunächst soll die Intra-Matcher-Parallelisierung individueller Matcher (*NameSyn*, *Children*, *NamePath*) auf einem einzelnen Rechenknoten mit Mehrkernprozessor analysiert werden. Abbildung 10.7 zeigt die Ausführungszeiten sowie die Speedup-Ergebnisse für die Parallelisierung der drei Match-Verfahren für bis zu acht parallele

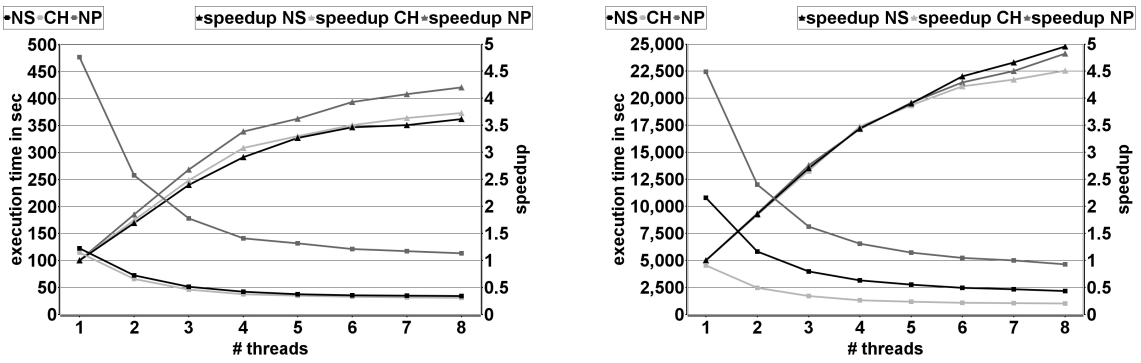



Abbildung 10.7: Intra-Matcher-Parallelisierung auf einem Knoten für *NameSyn* (NS), *Children* (CH) und *NamePath* (NP).
Links: mittelgroßes Match-Problem, rechts: großes Match-Problem.

Match Threads jeweils für das mittelgroße (links) und das große Match-Problem (rechts). Die Ausführungszeiten können für alle Match-Verfahren und beide Match-Probleme signifikant reduziert werden, indem der Grad der Paralleliliät erhöht wird. Der *NamePath*-Matcher mit seinen langen konkatenierten Strings ist mit Abstand das aufwendigste Match-Verfahren (circa vier mal längere Ausführungszeiten als *Children*). Für das große Match-Problem benötigt der Matcher ohne Parallelisierung mehr als sechs Stunden. Für das mittelgroße Match-Problem benötigen *NameSyn* und *Children* ungefähr genauso viel Zeit, wohingegen *NameSyn* für das große Match-Problem wesentlich länger braucht. Das liegt daran, dass GO sehr viele Synonyme pro Konzept enthält, so dass für jedes Konzeptpaar circa elf Vergleiche (anstatt drei für das mittelgroße Problem) ausgewertet werden müssen. Alle Matcher erreichen sehr gute Speedup-Werte von 3,6 - 4,2 für das mittelgroße Problem und sogar 4,5 - 5 für das große Match-Problem. Für bis zu vier Threads (= Anzahl der verfügbaren Kerne auf einem Knoten) kann ein annähernd linearer Speedup von 3,5 erreicht werden. Obwohl der verwendete Rechner nur vier Kerne hat, bringt eine Erhöhung der Thread-Anzahl weitere Verbesserungen (insbesondere für das große Match-Problem), allerdings auf einem reduzierten Level. Zusätzliche Threads können freie Kerne nutzen, während andere Match Threads auf einen neuen, zu bearbeitenden Task warten.

### 10.4.3 Paralleles Ontologie-Matching auf mehreren Knoten

Nun sollen die Parallelisierungsstrategien unter Verwendung von bis zu vier Rechenknoten (16 Kerne) mit bis zu vier Threads pro Knoten evaluiert werden. In diesem Experiment werden die drei individuellen Match-Verfahren *NameSyn*, *Children* und *NamePath* entsprechend der folgenden Parallelisierungsstrategien kombiniert: keine Parallelisierung (*NoPar*), Inter-Matcher-Parallelisierung (*Inter*),

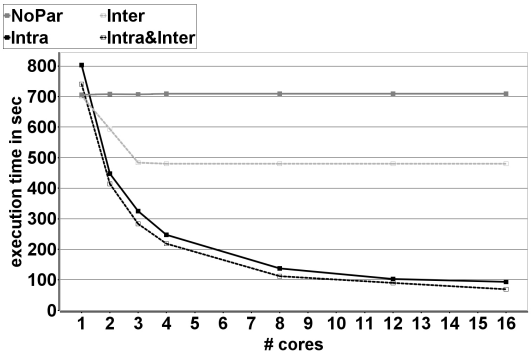



Abbildung 10.8: Parallelisierungsstrategien für das mittlere Match-Problem.

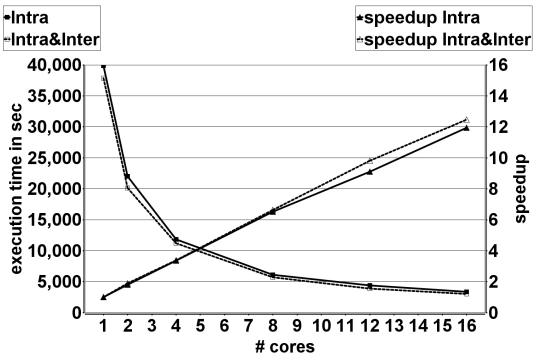



Abbildung 10.9: Intra-Matcher Parallelisierungsstrategien für das große Match-Problem.

Intra-Matcher-Parallelisierung (*Intra*) sowie die Kombination von Inter- und Intra-Matcher-Parallelisierung (*Intra&Inter*).

Abbildung 10.8 zeigt die Ausführungszeiten dieser Strategien für das mittelgroße Match-Problem. *NoPar* ist der Basisfall und profitiert nicht von der Verwendung multipler Threads oder Kerne. Alle anderen Parallelisierungsstrategien führen für die Verwendung von mehr als einem Kern zu einer Performanzverbesserung, allerdings unterscheiden sich die einzelnen Strategien. *Inter* profitiert nur bedingt, da von der verwendeten Match-Strategie maximal drei Kerne genutzt werden können. Die Gesamtlaufzeit wird durch den langsamsten Matcher (*NamePath*) limitiert. Im Gegensatz dazu sind *Intra* und *Intra&Inter* sehr effektiv und erreichen Ausführungszeiten unter 100 s. Die kombinierte *Intra&Inter*-Parallelisierung erreicht einen Speedup von 10,6 und ist somit etwas besser im Vergleich zur alleinigen Verwendung von *Intra* (Speedup 8,6). Im Gegensatz zu dem kombinierten Ansatz *Intra&Inter* führt *Intra* die drei Matcher sequentiell aus, so dass es zu Verzögerungen und Zeitverlust zwischen den einzelnen Match-Verfahren kommt.

Abbildung 10.9 zeigt die Ausführungszeiten und Speedup-Ergebnisse für die zwei Parallelisierungsstrategien *Intra* und *Intra&Inter* für das große Match-Problem. Aufgrund der enormen Ontologiegröße werden die Strategien ohne Intra-Matcher-Parallelität und Partitionierung (*NoPar*, *Inter*) ausgelassen. Die sequentielle Ausführungszeit der drei Matcher beträgt 11 Stunden. Unter Verwendung von 16 Kernen reduzieren *Intra* und *Intra&Inter* die Gesamtausführungszeit auf 50 bzw. 55 Minuten und erreichen jeweils einen Speedup von 11,9 und 12,5. Der Speedup konnte folglich, ähnlich wie für die Parallelisierung auf einem einzelnen Knoten, im Vergleich zum mittelgroßen Match-Problem verbessert werden. Somit stellen *Intra* und *Intra&Inter* besonders wertvolle Strategien für das parallele Matching sehr großer Ontologien dar.

Im Rahmen dieser Arbeit wurde GOMMA um die hier vorgestellten Strategien zum parallelen Matching von Ontologien erweitert. Ein weiterer Ansatz zur verteilten

Ausführung von Ontologie-Matching wurde 2012 unter Verwendung von MapReduce realisiert [185]. Der Ansatz erreicht insgesamt einen schlechteren Speedup als die Parallelisierungsstrategien von GOMMA. Insbesondere benötigte eine aufwendige Vorverarbeitung von RDF-Triplen zur Erstellung virtueller Dokumente mehr als 60% der Gesamtlaufzeit, wohingegen GOMMA's Vorverarbeitung nur einen marginalen Anteil der Gesamtlaufzeit ausmacht. Dementsprechend erreichte der MapReduce-basierte Ansatz einen Speedup von 2 unter Verwendung von 10 Rechenknoten, wohingegen das hier vorgestellte parallele Matching einen nahezu linearen Speedup erreicht.

## 10.5 Zusammenfassung

Das Kapitel stellte generelle Strategien für das parallele Ontologie-Matching unter Verwendung mehrerer Rechenknoten, insbesondere *Inter-* und *Intra-Matcher-Parallelisierung* sowie deren Kombination, vor. Diese Strategien erlauben die parallele Ausführung vollständiger, unabhängiger Matcher sowie die interne Parallelisierung der Matcher durch Partitionierung der Daten. Für Intra-Matcher-Parallelität wurde eine größenbasierte Partitionierung vorgestellt, die eine gute Lastbalancierung und Skalierbarkeit sowie reduzierte Speicheranforderungen aufweist, ohne die Qualität der Ergebnisse einzuschränken. Zudem wurde die Parallelisierung von element-, struktur- und instanzbasierten Match-Verfahren diskutiert. Für strukturbasierte Verfahren kommen mehrwertige *Kontextattribute* zum Einsatz. Im Rahmen der Studie wurde eine verteilte Infrastruktur implementiert, die das parallele Matching von Ontologien erlaubt. Der Ansatz wurde für das Matching großer Ontologien aus dem Bereich der Lebenswissenschaften evaluiert. Die Ergebnisse zeigen die Effizienz und Skalierbarkeit für einzelne Match-Verfahren sowie für kombinierte Match-Strategien. Dabei konnten insbesondere für sehr große Match-Probleme und die Kombination aus Inter- und Intra-Matcher-Parallelität sehr gute Ergebnisse erzielt werden.

# 11

# Evaluierung von GOMMA im Rahmen der OAEI 2012

## 11.1 OAEI 2012

Die *Ontology Alignment Evaluation Initiative* (OAEI)[38] führt jährlich eine vergleichende Evaluierung von Ontologie- und Schema-Matching-Systemen durch. Dabei werden insbesondere die Mappingqualität aber auch andere Kriterien wie Ausführungszeiten oder die Konsistenz von Mappings evaluiert. Die OAEI bietet mehrere *Tracks* an, die Match-Aufgaben aus unterschiedlichen Domänen umfassen. Die Evaluierung wird über die sogenannte SEALS (*Semantic Evaluation at Large Scale*)-Plattform[39] durchgeführt. Teilnehmende Match-Systeme müssen einheitliche Schnittstellen für die Eingabe zweier Ontologien und die Ausgabe eines Mappings implementieren. Somit können die Organisatoren eine vergleichende Evaluierung der Systeme durchführen. Die OAEI-Kampagne 2012 [3] umfasste folgende Aufgaben:

- *Anatomy Track*: Matching der *Adult Mouse Anatomy Ontology* (MA) zum Anatomieteil des NCI Thesaurus (NCITa).
- *Large BioMed Track*: Matching der sehr großen biomedizinischen Ontologien NCIT, SNOMED CT (SCT) und Foundational Model of Anatomy (FMA).

[38] `http://oaei.ontologymatching.org/`
[39] `http://www.seals-project.eu/`

- *Library Track*: Matching der in Bibliotheken genutzten, geistes- und wirtschaftswissenschaftlichen Ontologien *TheSoz Thesaurus* und *STW Thesaurus for Economics*.
- *Benchmark Test Library*: Matching einer Menge systematisch veränderter Ontologien zur jeweiligen Referenzontologie aus verschiedenen Domänen (z. B. Bibliographie, Konferenz, Systembiologie).
- *Conference Track*: Matching von Ontologien aus dem Bereich der Konferenzorganisation.
- *MultiFarm Track*: Matching multilingualer Ontologien basierend auf Übersetzungen der *Conference Track*-Ontologien.
- (*Instance Track*: Matching von Instanzdaten, u. a. personenbezogene und geographische Daten. Der *Instance Track* ist optional und wird nicht einheitlich auf der SEALS-Plattform evaluiert.)

GOMMA hat zur OAEI 2012 (sowie zur Zwischenevaluierung 2011.5[40]) an sämtlichen über die SEALS-Plattform evaluierten Tracks teilgenommen. Im Folgenden wird der von GOMMA für die OAEI 2012 verwendete Match Workflow näher vorgestellt, bevor in Kapitel 11.3 die Evaluierungsergebnisse präsentiert werden.

## 11.2 Präsentation des GOMMA-Systems

Die umfassende Infrastruktur GOMMA beinhaltet eine generische Komponente für das Matching von Ontologien (siehe Kapitel 3.2.1) und ist insbesondere in der Lage sehr große Ontologien abzugleichen. Dazu stellt GOMMA die folgenden skalierbaren Match-Techniken zur Verfügung:

1. das parallele Matching von Ontologien unter Ausnutzung mehrerer Rechenknoten mit Mehrkernprozessoren (siehe Kapitel 10),
2. die indirekte Berechnung von Ontologiemappings auf Basis der Wiederverwendung und Komposition bereits existierender Ontologiemappings (siehe Kapitel 9), sowie
3. einen Blocking-Ansatz zur Reduktion des Suchraums durch Einschränkung des Matchings auf überlappende Ontologieteile.

Diese Techniken unterstützen die Effizienz des Matchings insbesondere zur Reduktion der Ausführungszeiten. Die beiden letzteren Ansätze können zudem die Qualität der berechneten Mappings verbessern. GOMMA's Match-Komponente ist generisch, auch wenn das System ursprünglich auf den Bereich der Lebenswissenschaften fokussierte. Dementsprechend konnte GOMMA erfolgreich an allen im Rahmen der OAEI 2012 gestellten Aufgaben zum Ontologie-Matching teilnehmen.

[40] `http://oaei.ontologymatching.org/2011.5/`

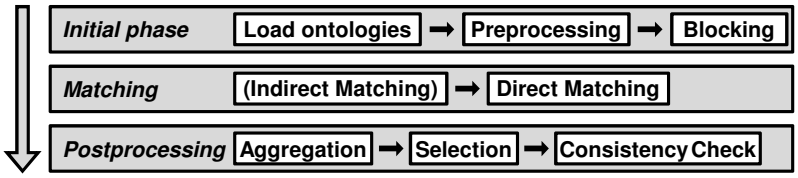



Abbildung 11.1: GOMMA Match Workflow für die OAEI 2012-Teilnahme.

Abbildung 11.1 zeigt den für die OAEI 2012 verwendeten GOMMA Match Workflow. Dieser umfasst eine initiale Phase (inklusive einer neuen Blocking-Strategie), die Match-Phase sowie verschiedene Schritte zur Nachbearbeitung der generierten Mappings. GOMMA basiert auf den in Kapitel 3.1 eingeführten Modellen für Ontologien und Mappings. Die Eingabe des Match Workflows sind zwei Ontologien[41] $O1$ und $O2$ sowie optional eine Menge bereits existierender Mappings. Die Ausgabe umfasst das Mapping $OM_{O1,O2}$.

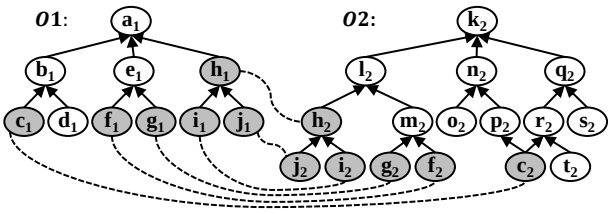
### 11.2.1 Initiale Phase und Blocking

In der initialen Phase findet das Einlesen (Parsen) und Laden der Ontologien aus dem OWL-Format statt (*Load ontologies*). In dieser Phase werden sämtliche für das Matching relevanten Informationen den Konzepten zugeordnet. Dazu zählen insbesondere Namen, Synonyme, Kommentare und Instanzen. Die Zuordnung von Instanzen entspricht jener des attributbasierten Ansatzes aus Kapitel 10.2.2. Die Ergebnismappings sollen neben Korrespondenzen zwischen OWL-Klassen auch Korrespondenzen zwischen OWL-*Properties* (Eigenschaften) enthalten. Daher werden in GOMMA sowohl `owl:class` als auch verschiedene Eigenschaften wie `owl:ObjectProperty` oder `owl:DatatypeProperty` als Konzepte verwaltet. Der Typ *class* oder *property* sowie die einer Eigenschaft (*property*) zugeordneten *Domain*- und *Range*-Klassen werden ebenfalls als Konzeptattribute hinterlegt. Sämtliche für das Matching benötigten Informationen sind folglich als Textattribute gespeichert und können für String-basierte Ähnlichkeitsberechnungen verwendet werden.

Während der Vorverarbeitung (*Preprocessing*) wird zunächst die Sprache der Attributwerte geprüft und vermerkt (Verwendung von `xml:lang` in `rdfs:label`). Falls die verwendete Sprache nicht Englisch ist, wird der betreffende Attributwert ins Englische übersetzt und dem zugehörigen Konzept als neues Synonym hinzugefügt. Für die automatische Übersetzung der Terme wird die freie Übersetzungsschnittstelle *MyMemory*[42] verwendet. Zur Erfassung der Synonyme wird iterativ ein Wörterbuch aufgebaut, wobei neue Terme einmalig unter Verwendung der API nachgeschlagen und im Wörterbuch hinterlegt werden. Alle Konzeptattributwerte werden

[41] Auf die Verwendung der Versionsindices wird verzichtet, da im Rahmen dieser Evaluierung jeweils nur eine feste Version einer Ontologie verwendet wird.

[42] `http://mymemory.translated.net/`




$$corrFrac(root) = \frac{|OM_{initial}(subgraph(root))|}{|M_{initial}|}$$

$$corrFrac(b_1) = \frac{1}{6} = 0.16 \quad corrFrac(l_1) = \frac{5}{6} = 0.83$$

$$corrFrac(e_1) = \frac{2}{6} = 0.33 \quad corrFrac(n_2) = \frac{0.5}{6} = 0.083$$

$$corrFrac(h_1) = \frac{3}{6} = 0.5 \quad corrFrac(q_2) = \frac{0.5}{6} = 0.083$$

Abbildung 11.2: Blocking von Ontologiesubgraphen.

zudem normalisiert, indem Trennzeichen und Stoppwörter entfernt und alle Strings in Kleinschreibung überführt werden.

Darüber hinaus kommt in der initialen Phase eine *Blocking*-Strategie für den Abgleich sehr großer Ontologien zum Einsatz. Ziel ist es, den Suchraum bzw. die Anzahl der notwendigen Vergleiche zu reduzieren. Es existieren bereits verschiedene Partitionierungs- und Blocking-Ansätze im Bereich des Ontologie-Matchings (siehe Kapitel 2.3.2). Der hier verwendete Ansatz fokussiert auf asymmetrische Match-Probleme, wenn eine spezifische Ontologie mit einer umfassenderen (nur teilweise relevanten) Ontologie verglichen wird. Ein Beispiel für ein asymmetrisches Match-Problem stellt das Matching einer reinen Anatomieontologie wie FMA zu einer umfassenden biomedizinischen Ontologie wie NCIT dar. Ein anderes Szenario aus dem Bereich *Linked Data* ist das Matching einer domänenspezifischen (z. B. geographischen) Ontologie zu der eher allgemeinen DBpedia-Ontologie.

Ziel der Blocking-Strategie ist es den relevanten Teil der allgemeineren bzw. umfassenderen Ontologie automatisch zu identifizieren, so dass nur dieser Teil mit der anderen, spezifischeren (und häufig kleineren) Ontologie verglichen werden muss. In anwendbaren Fällen kann sich sowohl die Effizienz des Matchings als auch die Qualität der Mappings verbessern. Insbesondere erhöht sich die Precision, wenn weniger falsch positive Korrespondenzen generiert werden. Die Blocking-Strategie basiert auf einem initialen Mapping und funktioniert folgendermaßen:

1. Bestimme ein initiales Mapping $OM_{initial}$ unter Verwendung einer sehr effizienten Matchmethode (z. B. Namensgleichheit auf Basis eines Indexes[43] oder Komposition existierender Mappings).

2. Identifiziere eine Menge von Subgraph-Wurzeln (*Roots*) unterhalb der höchsten Wurzel. Bestimme die Anzahl der Korrespondenzen aus $OM_{initial}$ pro $root \in Roots$ ($|OM_{initial}(subgraph(root))|$), indem die Anzahl der Korrespondenzen von der Blattebene nach oben propagiert wird. Falls Mehrfachvererbung vorliegt, wird die Korrespondenzanzahl anteilig (partiell) den jeweiligen Elternkonzepten zugeordnet (für das Beispiel in Abbildung 11.2 geschieht dies

[43] Diese Methode wurde für die OAEI 2012 zur Bestimmung des initialen Mappings verwendet.

für das $O2$-Konzept $c_2$). Die Summe aller Korrespondenzen pro $root \in Roots$ entspricht somit der Gesamtmenge aller Korrespondenzen in $OM_{initial}$.

3. Berechne für jede *root* den Anteil der zugeordneten Korrespondenzen ihres Subgraphen $|OM_{initial}(subgraph(root))|$ im Verhältnis zur Anzahl aller initialer Korrespondenzen $|OM_{initial}|$ (siehe $corrFrac(root)$ in Abbildung 11.2).

4. Wähle eine oder mehrere Wurzel(n) mit einer *corrFrac* oberhalb eines bestimmten Grenzwerts. Nur die Konzepte im Subgraphen dieser Wurzel(n) werden im Matching verwendet. Wenn keine Wurzel oberhalb des Grenzwerts liegt, wird kein Blocking angewendet. In diesem Fall muss die gesamte Ontologie abgeglichen werden, da keine „dominierende“ Teilontologie identifiziert wurde.

Abbildung 11.2 veranschaulicht den Ansatz für zwei Ontologien und eine Menge vorbestimmter Korrespondenzen. Um einen geeigneten Subgraphen für das Matching zu identifizieren, werden die Wurzeln auf der zweiten Ontologieebene betrachtet ($b_1$, $e_1$, $h_1$ für $O1$ und $l_2$, $n_2$, $q_2$ für $O2$). Die Anwendung eines $corrFrac$-Grenzwerts von 0,7 bedeutet, dass ein Subgraph mindestens 70% aller initialen Korrespondenzen abdecken muss. Dies ist nur für Wurzel $l_2$ aus $O2$ der Fall, d. h. im Beispiel kann nur $O2$ partitioniert werden. Dementsprechend wird die gesamte $O1$-Ontologie mit dem $l_2$-Subgraphen von $O2$ abgeglichen.

### 11.2.2 Matching

GOMMA's Match-Komponente ermöglicht das direkte sowie indirekte Matching von Ontologien. Direkte Match-Verfahren beziehen internes Ontologiewissen wie z. B. konzeptassoziierte oder strukturelle Informationen ein. Indirekte Match-Verfahren basieren hingegen auf der Komposition existierender Mappings zu Mediatorontologien. Um ein effizientes Matching insbesondere großer Ontologien zu ermöglichen, werden direkte Match-Verfahren parallel ausgeführt. Für die OAEI 2012 wurden die folgenden Match-Verfahren verwendet.

Der direkte Abgleich zweier Ontologien kombiniert bis zu drei Match-Verfahren. In jedem Fall wird der *NameSyn*-Matcher eingesetzt, der unter Verwendung von TriGram die maximale Stringähnlichkeit der Namen und Synonyme pro Konzeptpaar bestimmt. Zusätzlich berechnen ein Kommentar- und ein Instanzmatcher die Stringähnlichkeit zwischen Instanzen bzw. Kommentaren der Konzepte sofern diese verfügbar sind. GOMMA bietet auch strukturbasierte Match-Verfahren [102]. Allerdings erwiesen sich diese in ersten eigenen Untersuchungen im Bereich der Lebenswissenschaften als weniger effektiv, weshalb GOMMA's OAEI-Match-Strategie keine strukturbasierten Verfahren verwendet.

Um Ontologien effizient abgleichen zu können, wird die Intra-Matcher-Parallelisierung (Kapitel 10) angewendet. Dazu werden die Eingabeontologien in kleinere, gleich

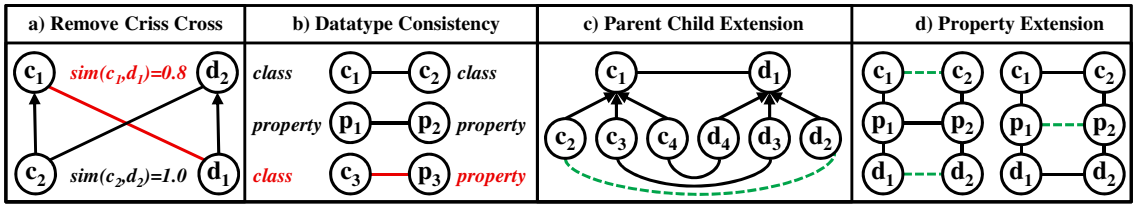



Abbildung 11.3: Konsistenzbedingungen. Rote, durchgezogene (grüne, gestrichelte) Linie = Löschung (Hinzufügung) einer Korrespondenz.

große Fragmente zerlegt, so dass einzelne Match Tasks für diese Fragmente parallel ausgeführt werden können. Diese Art der Parallelisierung kann für die angewendeten Match-Verfahren leicht realisiert werden, da alle für das Matching benötigten Informationen direkt zu den Konzepten assoziiert sind. Im Rahmen der OAEI-Evaluierung wird ein einzelner Rechner für die Ausführung des Matchings genutzt. Daher wurde das parallele Matching dahingehend modifiziert, dass *Threading* zur verteilten Ausführung mehrerer Match-Jobs auf den verfügbaren Kernen einer Maschine (anstelle mehrerer Rechenknoten) angewendet wird.

Zur Verbesserung der Mappingqualität wird zudem der indirekte, kompositionsbasierte Match-Ansatz verwendet (Kapitel 9). Dieser erlaubt die Wiederverwendung bereits existierender, qualitativ hochwertiger Mappings und ermöglicht das effiziente Matching zweier bisher nicht verknüpfter Ontologien. Beispielsweise können zwei Ontologien $O1$ und $O2$ abgeglichen werden, indem zwei Ontologiemappings von $O1$ und $O2$ zu einer Mediatorontologie $HO$ (z. B. $OM_{O1,HO}$ und $OM_{HO,O2}$) kombiniert werden. Eine geeignete Mediatorontologie (z. B. UMLS) enthält typischerweise wertvolles Hintergrundwissen wie zahlreiche Synonyme. Das indirekt erzeugte Mapping kann unvollständig sein und wird entsprechend erweitert. Dazu werden bisher nicht verknüpfte Bereiche in $O1$ ($O2$) identifiziert und durch Verwendung der direkten Match-Strategie mit der gesamten anderen Ontologie abgeglichen.

### 11.2.3 Nachbearbeitung

Die Hauptaufgabe dieser Phase ist die Kombination und Aggregation der direkt und indirekt bestimmten Mappings sowie die Selektion der wahrscheinlichsten Korrespondenzen (siehe Kapitel 2.3.1). Zunächst werden Korrespondenzen mit einer Ähnlichkeit unterhalb eines bestimmten Grenzwerts gefiltert. Alle erzeugten Mappings werden vereinigt, wobei für jede Korrespondenz der durchschnittliche Ähnlichkeitswert gebildet wird. Um nur die besten Korrespondenzen für jedes Quell- und Zielkonzept zu behalten, wird jeweils die *MaxDelta*-Selektion angewendet.

Darüber hinaus dienen einige Techniken der Verbesserung der Mappingkonsistenz, indem vermutlich falsche (fehlende) Korrespondenzen entfernt (hinzugefügt) werden. Bisher verwendet GOMMA vier einfache Bedingungen. Abbildung 11.3 zeigt

einfache Beispielszenarien für jede Konsistenzbedingung. Ähnlich zur semantischen Verifikation in ASMOV [94] überprüfen die ersten beiden Bedingungen Situationen, die zur Entfernung von Korrespondenzen (d. h. zur Verbesserung der Precision) führen können. Die Erfüllung der anderen beiden Bedingungen resultiert in der Hinzufügung von Korrespondenzen (zur Verbesserung des Recalls).

Zunächst werden Korrespondenzen bezüglich der sogenannten *CrissCross*-Bedingung überprüft (a). Dabei werden Konflikte identifiziert, falls zwei Korrespondenzen $(c_1, d_1)$ und $(c_2, d_2)$ existieren, wobei $c_2$ ein Kind von $c_1$, $d_1$ jedoch ein Kind von $d_2$ ist (oder umgekehrt). Zur Auflösung des Konflikts kann z. B. die Korrespondenz mit dem kleineren Ähnlichkeitswert entfernt werden. Zudem wird die Konsistenz der Konzepttypen überprüft (b). Insbesondere werden Korrespondenzen zwischen Klassen und Properties entfernt, d. h. nur (*class-class*)- und (*property-property*)-Korrespondenzen sind erlaubt. Die dritte Regel testet, ob zwischen zwei Konzepten eine Korrespondenz besteht, jedoch gleichzeitig nur ein Teil ihrer Kinderkonzepte verknüpft sind (c). In diesem Fall wird eine Korrespondenz für die ähnlichsten, nicht verknüpften Kinder hinzugefügt. Falls eine Korrespondenz zwischen zwei Properties existiert, werden fehlende Korrespondenzen für die zugehörigen *Domain* / *Range*-Klassen ergänzt (d). Ebenso wird eine Korrespondenz zwischen zwei Properties hinzugefügt, wenn sowohl die *Domain*- als auch die *Range*-Klasse durch eine Korrespondenz verknüpft sind.

## 11.3 Evaluierungsergebnisse

In diesem Abschnitt werden die Evaluierungsergebnisse von GOMMA zur OAEI 2012-Kampagne vorgestellt und mit den Ergebnissen anderer Systeme verglichen. GOMMA hat an sechs *Tracks* teilgenommen: *Anatomy*, *Large BioMed*, *Benchmark*, *Library*, *Conference* und *MultiFarm*. Weitere Ergebnisse und detaillierte Beschreibungen der einzelnen Tracks und der jeweils eingesetzten Maschinen finden sich auf den Ergebnisseiten der OAEI 2012[44]. Zur OAEI 2012 haben 23 Systeme bzw. Systemkonfigurationen[45] an insgesamt 103 Tests der sechs SEALS-Tracks teilgenommen. In den Abbildungen der folgenden Abschnitte sind jeweils nur Systeme aufgeführt, die die jeweiligen Aufgaben absolvieren konnten. In einigen Tracks wenden die Organisatoren zur Orientierung eine sehr einfache Matchmethode (z. B. einen exakten Namensvergleich) an. Diese sind in den Abbildungen jeweils als *„Basic reference matcher“* gekennzeichnet. Entsprechend der OAEI-Richtlinien nutzt GOMMA keine problemspezifischen Parameterkonfigurationen (z. B. für Grenzwerte), sondern einen

[44] OAEI 2012-Ergebnisse: http://oaei.ontologymatching.org/2012/results/

[45] AROMA, ASE, AUTOMSv2, CODI, GOMMA, GOMMA-bk, Hertuda, HotMatch, LogMap, LogMapLt, LogMap-noe, MaasMatch, MapSSS, MEDLEY, OMR , OntoK , Optima, ServOMap, ServOMapLt, TOAST, WeSeE, Wikimatch, YAM++

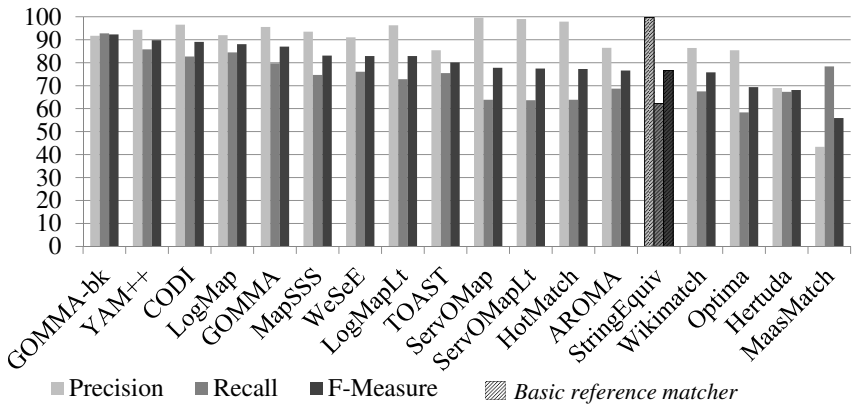



Abbildung 11.4: Mappingqualität - *Anatomy Track*.

einheitlichen Match Workflow für alle sechs SEALS-Tracks[46] (siehe Kapitel 11.2). Zur Verwendung des kompositionsbasierten Matchings kann ein Parameter ein- und ausgeschaltet werden (GOMMA und GOMMA-bk)[47]. Die von GOMMA-bk verwendeten Mappings zu Mediatorontologien einer Domäne wurden mit dem *NameSyn*-Matcher generiert.

Da GOMMA insbesondere für das Matching großer Ontologien im Bereich der Lebenswissenschaften geeignet ist, werden zunächst der *Anatomy* und *Large BioMed Track* detaillierter diskutiert, bevor im Anschluss die Ergebnisse für *Library* und *Benchmark* sowie *Conference* und *MultiFarm* vorgestellt werden.

### 11.3.1 Anatomy Track

Abbildung 11.4 zeigt die Ergebnisse der einzelnen Systeme bezüglich der erreichten Mappingqualität (Precision, Recall, F-Measure) im *Anatomy Track* der OAEI 2012. Die indirekte Match-Strategie GOMMA-bk erreicht das bis dahin beste F-Measure (92,3%) für das Matching von MA und NCITa. Dabei werden Mappings zu drei Mediatorontologien (UMLS, Uberon und FMA) verwendet. Im Vergleich zur direkten Match-Strategie von GOMMA (F-Measure 87%) konnte insbesondere der Recall um ≈13% auf den insgesamt besten Wert von 92,8% verbessert werden. Einige Ansätze erreichen auf Kosten eines um circa 30% reduzierten Recalls eine nahezu perfekte Precision (z. B. ServOMap [9] und das Basisverfahren *StringEquiv*). GOMMA-bk erreicht mit 15 Sekunden die zweitbeste Ausführungszeit nach der „Lite"-Konfiguration von LogMap (LogMapLt, 6 Sekunden). Insgesamt liegt die Ausführungszeit von sieben Systemen unterhalb einer Minute (siehe Anhang Tabelle B.1), wohingegen fünf Systeme ungefähr 1-5 Stunden für das Matching benötigen. Dies verdeutlicht die

[46] GOMMA-Parameterkonfigurationen zur OAEI 2012: $sim \geq 0{,}8$; $\Delta = 0{,}0$; $corrFrac \geq 0{,}6$.

[47] Falls die Ergebnisse von GOMMA und GOMMA-bk gleich sind (da z. B. keine kombinierbaren Mappings verfügbar sind), wird GOMMA-bk nicht separat aufgeführt.

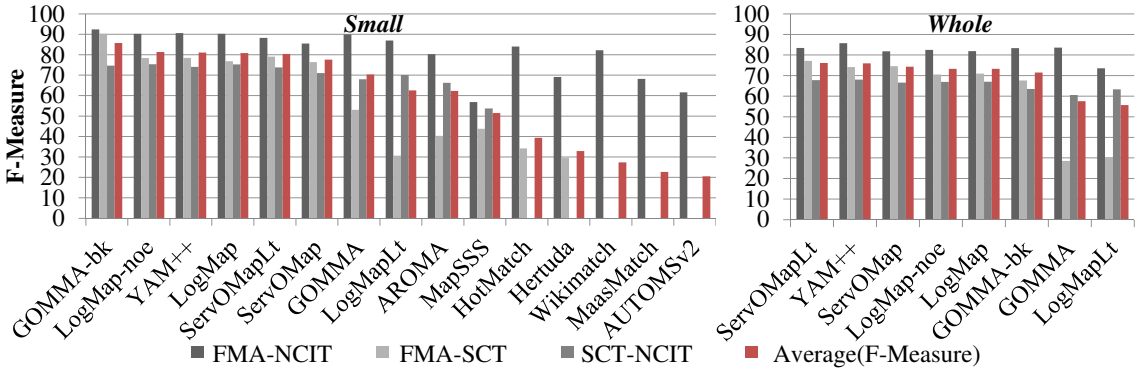



Abbildung 11.5: Mappingqualität - *Large BioMed Track* für *Small* und *Whole*.

Probleme bezüglich der Skalierbarkeit einiger Match-Systeme für mittelgroße Match-Probleme wie das Matching von MA (2.700 Konzepte) und NCITa (3.300 Konzepte). Im Folgenden werden die Ergebnisse für das Matching der weitaus größeren Ontologien des *Large BioMed Tracks* diskutiert.

### 11.3.2 Large BioMed Track

Der *Large BioMed Track* umfasst drei Match-Probleme für die besonders großen, biomedizinischen Ontologien FMA, NCIT und SCT (FMA–NCIT, FMA–SCT, SCT–NCIT). Die Organisatoren haben für jedes Match-Problem jeweils drei Teilaufgaben erstellt. Neben den vollständigen Ontologien (*Whole*) sollen die Systeme kleine (*Small*) sowie mittelgroße (*Large*) Ontologiefragmente abgleichen (siehe Anhang Tabelle B.2). Die vollständigen NCIT- und FMA-Ontologien umfassen jeweils circa 79.000 und 67.000 Konzepte. Für SCT wird ein Teilfragment von ≈120.000 Konzepten für *Large* und *Whole* verwendet. Die Referenzmappings[48] wurden zunächst aus UMLS extrahiert und gelten als „Silberstandard“. Zudem wurden die Referenzmappings durch Anwendung der Reparaturverfahren von LogMap [97] und Alcomo [128] verfeinert. Die im Folgenden verwendeten Precision/Recall/F-Measure-Werte mitteln die Ergebnisse bezüglich der drei Referenzmappings (UMLS, UMLS-LogMap, UMLS-Alcomo).

Die GOMMA-Ansätze für das kompositionsbasierte und parallele Matching sowie das Blocking-Verfahren erweisen sich im *Large BioMed Track* als besonders wertvoll, um Mappings guter Qualität in niedrigen Ausführungszeiten zu generieren. Die Blocking-Strategie kommt für sehr große Ontologien mit über 5.000 Konzepten zum Einsatz. Dies führt zur Auswahl von Subgraphen für NCIT (im FMA–NCIT Task) und SCT (FMA–SCT, SCT–NCIT), wodurch der Suchraum um Faktor 2–6 reduziert werden kann. Für den kompositionsbasierten Ansatz (GOMMA-bk) stehen Mappings zu zwei Mediatorontologien (UMLS und Uberon) zur Verfügung.

[48] `http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2012/oaei2012_umls_reference`

| | Average Quality | | | Sum Time | | Incoherence |
|---|---|---|---|---|---|---|
| | Precision | Recall | F-Measure | in s | in h | |
| **YAM++** | 87.6 | 71.0 | 78.2 | 67817 | 18.8 | 45.30% |
| **ServOMapLt** | 89.0 | 69.9 | 78.0 | 2405 | 0.7 | 51.46% |
| **LogMap-noe** | 86.9 | 69.5 | 77.0 | 3964 | 1.1 | 0.00% |
| **GOMMA-bk** | 76.7 | 79.1 | 76.8 | 5821 | 1.6 | 45.32% |
| **LogMap** | 86.9 | 68.4 | 76.2 | 3077 | 0.9 | 0.01% |
| **ServOMap** | 90.3 | 65.7 | 75.8 | 2310 | 0.6 | 55.36% |
| **GOMMA** | 74.6 | 55.3 | 62.5 | 5341 | 1.5 | 24.01% |
| **LogMapLt** | 83.1 | 51.5 | 58.6 | 711 | 0.2 | 33.17% |

Tabelle 11.1: Gesamtergebnisse des *Large BioMed Tracks* bezüglich durchschnittlicher Mappingqualität, Gesamtausführungszeiten und Inkohärenz-Grad.

Abbildung 11.5 zeigt die Ergebnisse zur Mappingqualität (F-Measure) für alle drei Match-Probleme jeweils für die *Small* (links)- und *Whole* (rechts)-Teilaufgabe. Die vollständigen Ergebnisse für alle neun Tasks bezüglich Qualität und Laufzeit zeigt Tabelle B.3 im Anhang. Die roten Balken geben das durchschnittliche F-Measure eines Systems für alle *Small*- bzw. *Whole*-Tasks an. Elf Systeme sind in der Lage alle drei *Small*-Aufgaben zu absolvieren und weitere fünf Systeme können immerhin das kleinste Match-Problem FMA-NCIT ausführen. GOMMA-bk erreicht das beste, durchschnittliche F-Measure für den *Small*-Task (≈86%). Der SCT–NCIT-Task erscheint für GOMMA-bk wesentlich schwieriger (≈75% F-Measure) als FMA–NCIT und FMA–SCT (≥90% F-Measure). Hingegen erreicht GOMMA ein niedriges F-Measure für FMA–SCT (≈53%), welches durch das kompositionsbasierte Matching um ≈37% verbessert werden kann. Nur acht von insgesamt 23 teilnehmenden Systemen konnten den *Whole*-Task für alle drei Match-Probleme absolvieren. ServOMapLt [9] liegt mit durchschnittlich 76,2% knapp vor YAM++ [137] (76%). Auch für den *Whole*-Task führt die Verwendung des kompositionsbasierten Ansatzes (GOMMA-bk) zu einer deutlichen Verbesserung im Vergleich zum direkten Abgleich der Ontologien (durchschnittlich ≈72% statt ≈58% F-Measure).

Tabelle 11.1 zeigt die Ergebnisse der acht besten Systeme für den *Large BioMed Track* 2012. Weitere 15 evaluierte Systeme konnten entweder keine oder nur wenige Teilaufgaben realisieren. GOMMA-bk erreicht den besten Recall, allerdings ist die Precision im Vergleich zu den anderen Systemen reduziert. Insgesamt erreicht YAM++ die beste Mappingqualität, jedoch benötigt das System fast 19 Stunden für die Berechnung. Die Laufzeit-optimierte „Lite"-Konfiguration von LogMap (LogMapLt) absolviert hingegen sämtliche Teilaufgaben in knapp 12 Minuten. Der kompositionsbasierte Ansatz GOMMA-bk benötigt insgesamt etwas länger als das alleinige direkte Matching von GOMMA (1,6 h vs. 1,5 h). Ein Grund dafür ist, dass die Referenzmappings des *Large Bio Tracks* nur 2-5% der Ontologien abdecken, so dass GOMMA-bk im Anschluss an die Komposition beinahe das gesamte kartesische Produkt evaluiert. LogMap und LogMap-noe erreichen sehr gute Werte bezüglich der Inkohärenz der Mappings, d. h. beim Reasoning der Ontologien und Mappings werden

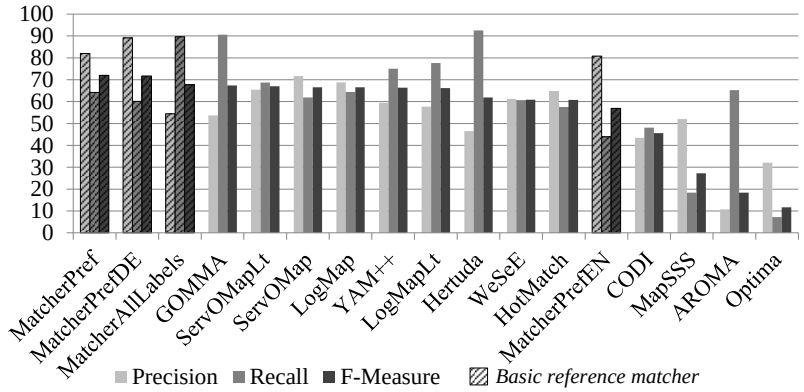



Abbildung 11.6: Mappingqualität - *Library Track*.

kaum Inkonsistenzen in Form unerfüllbarer Klassen detektiert. Die mit GOMMA erzeugten Mappings weisen einen niedrigeren Inkohärenzanteil als GOMMA-bk-Mappings auf. Vermutlich erzeugt der kompositionsbasierte Ansatz, zugunsten eines hohen Recalls, einen höheren Anteil widersprüchlicher Korrespondenzen. GOMMA und GOMMA-bk nutzen zwar bereits einige Nachbearbeitungsschritte zur Beseitigung von Inkonsistenzen, jedoch sollten in Zukunft weitere, fortgeschrittene Techniken zum Einsatz kommen.

### 11.3.3 Library und Benchmark Tracks

Der *Library Track* wurde 2012 neu eingeführt und beinhaltet das Matching der zwei leichtgewichtigen Vokabulare STW und TheSoz Thesaurus, die jeweils aus 6.500 und 8.500 Konzepten bestehen. Im Gegensatz zu den anderen Tracks konnten Teilnehmer ihre Systeme vor der Evaluierung zwar testen, erhielten jedoch keine Aussage zur erreichten Mappingqualität. GOMMA erreichte in dieser „blinden“ Evaluierung mit

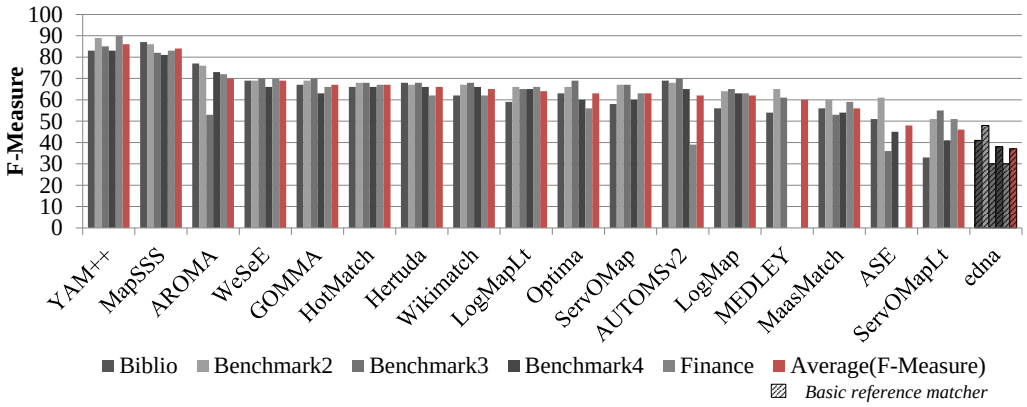



Abbildung 11.7: Mappingqualität - *Benchmark Track*.

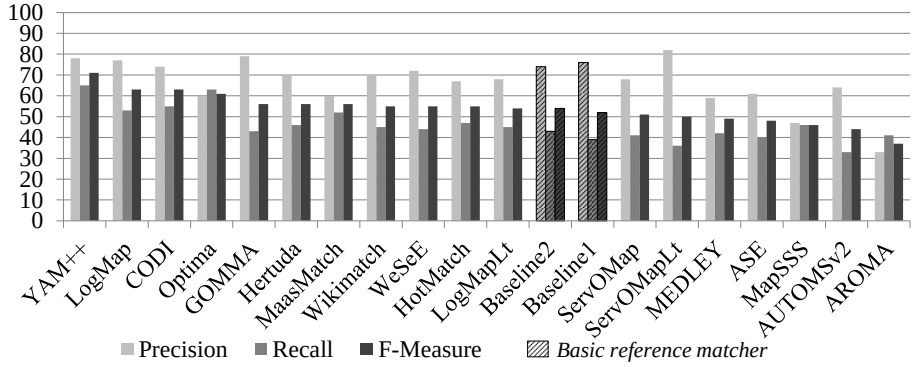



Abbildung 11.8: Mappingqualität - *Conference Track*.

minimalem Vorsprung das beste F-Measure der teilnehmenden Systeme (≈67,4%), insbesondere aufgrund eines hohen Recalls von ≈91% (siehe Abbildung 11.6). Allerdings ist die Precision mit ≈54% vergleichsweise niedrig. Auffällig ist, dass drei durch die Organisatoren eingesetzte Basis-Match-Verfahren bessere Ergebnisse erreichen als alle teilnehmenden Systeme. Zudem liegt das F-Measure der sechs besten Systeme sehr eng beieinander (≈66-67%). Die Vokabulare beinhalten sehr viele Namen und Synonyme (circa fünf pro Konzept), wodurch sich zum einen GOMMA's hoher Recall, aber auch die niedrige Precision begründen lässt, da möglicherweise nicht alle Synonyme exakt sind und somit ungenaue oder falsche Korrespondenzen entstehen können.

Abbildung 11.7 zeigt die F-Measure-Ergebnisse des *Benchmark Tracks*, welcher aus fünf Teilaufgaben (Biblio, Finance und Benchmark 2–4) besteht. Jede Teilaufgabe umfasst circa 50 kleine Match-Aufgaben, wobei jeweils eine systematisch modifizierte Ontologie mit der Ausgangsontologie abgeglichen werden muss. Insgesamt erreichte GOMMA gute F-Measure-Werte zwischen 60–70% (Durchschnitt 67%), wobei die Precision meist besser als der Recall ist[49].

### 11.3.4 Conference und Multifarm Tracks

Der *Conference Track* umfasst 16 kleine Ontologien aus der Domäne der Konferenzorganisation. Dabei muss jede Ontologie mit jeder der verbleibenden Ontologien abgeglichen werden. Abbildung 11.8 zeigt die Ergebnisse bezüglich der durchschnittlichen Mappingqualität (Precision, Recall, F-Measure). Alle teilnehmenden Systeme bevorzugen eine gute Precision gegenüber einem hohen Recall. GOMMA erreichte zusammen mit Hertuda und MaasMatch das fünft-beste Ergebnis von 56% und ist somit etwas besser als die zwei Basisverfahren *Baseline* 1 und 2 (52-54% F-Measure).

[49]Siehe `oaei.ontologymatching.org/2012/results/benchmarks`

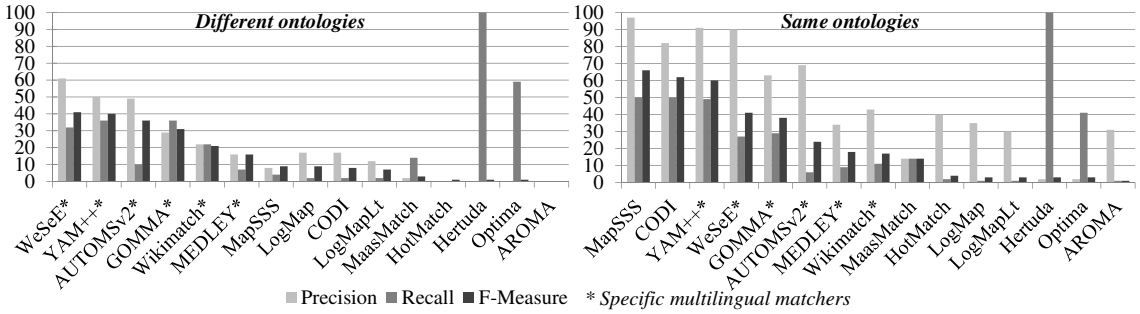


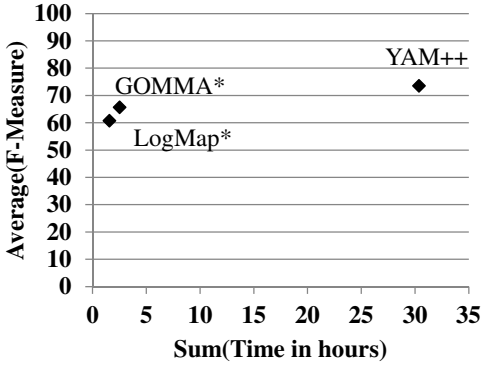

Abbildung 11.9: Mappingqualität - *MultiFarm Track*.

Der *MultiFarm Track* basiert auf den 16 Ontologien des *Conference Tracks*. Dazu wurden die englischsprachigen Ausgangsontologien in acht weitere Sprachen (z. B. Russisch, Chinesisch, Spanisch) übersetzt, so dass der Track 36 Sprachpaare (z. B. Chinesisch-Spanisch, Englisch-Russisch) umfasst. Für alle Sprachpaare werden zwei Teilaufgaben unterschieden, da das multilinguale Matching der gleichen Ontologien (*Same ontologies*) sowie unterschiedlicher Ontologien (*Different ontologies*) durchgeführt werden muss. Die Ergebnisse in Abbildung 11.9 zeigen die durchschnittliche Mappingqualität (Precision, Recall, F-Measure) für alle betrachteten Ontologie- und Sprachpaare. Einige Systeme nutzen spezifische multilinguale Match-Verfahren (mit $*$ gekennzeichnet). GOMMA führt eine automatische Übersetzung (Kapitel 11.2.1) von nicht-englischen Namen und Synonymen ins Englische durch, so dass im Anschluss übersetzte Ontologien miteinander verglichen werden können.

Insgesamt gestaltet sich das Matching von Ontologien in unterschiedlichen Sprachen schwierig, was durch das insgesamt relativ niedrige F-Measure Niveau der teilnehmenden Systeme deutlich wird. Die meisten Systeme erreichen für das Matching der gleichen Ontologien in verschiedenen Sprachen bessere Ergebnisse als für das multilinguale Matching unterschiedlicher Ontologien. Für das Matching der gleichen Ontologien schneiden auch einige Systeme ohne spezifische multilinguale Match-Technik sehr gut ab (MapSSS und CODI). GOMMA erreicht mit durchschnittlich jeweils 31% und 38% F-Measure gute Ergebnisse in beiden Teilaufgaben. Dabei profitiert GOMMA insbesondere von der Übersetzungsstrategie, so dass das System verhältnismäßig viele korrekte Korrespondenzen identifiziert (36% Recall für *Different ontologies*). Die besten Ergebnisse ergaben sich für Sprachpaare zwischen Englisch und anderen Sprachen sowie Paare nah verwandter Sprachen (z. B. Spanisch und Portugiesisch).

## 11.4 Zusammenfassung

Insgesamt konnten nur die drei Systeme YAM++, GOMMA und LogMap erfolgreich sämtliche Teilaufgaben zum Ontologie-Matching während der OAEI 2012 ab-




| | Sum(Time in h) | Average(F-Measure) |
|---|---|---|
| **YAM++** | 30.4 | 73.6 |
| **GOMMA*** | 2.5 | 65.7 |
| **LogMap*** | 1.6 | 60.8 |

GOMMA* = Best F-Measure(GOMMA, GOMMA-bk)
LogMap* = Best F-Measure(LogMap, LogMap-noe, LogMapLt)

Abbildung 11.10: Aggregierte Ergebnisse der drei besten Systeme der OAEI 2012.

solvieren. Abbildung 11.10 zeigt aggregierte Gesamtergebnisse, wobei die Qualität (F-Measure) über alle Tasks gemittelt und die Gesamtlaufzeit aufsummiert wurde. YAM++ erreicht durchschnittlich die beste Mappingqualität ≈74% und benötigt dafür ungefähr 30 Stunden. Im Gegensatz dazu benötigen GOMMA und LogMap wesentlich geringere Ausführungszeiten. Dabei zeichnet sich GOMMA durch ein etwas besseres durchschnittliches F-Measure von ≈66% (in 2,5 h) und LogMap durch die niedrigere Gesamtlaufzeit (1,6 h, ≈61% F-Measure) aus.

GOMMA's erfolgreiche Teilnahme an sechs Tracks der OAEI 2012 zeigte, dass das System in der Lage ist unterschiedlich große Ontologien aus sehr verschiedenen Domänen abzugleichen, und bestätigt GOMMA's Stärken. Das System ermöglicht skalierbares Matching, insbesondere für sehr große Ontologien, durch Ausführung von Blocking, parallelem Matching und Mappingkomposition. Insbesondere für das Matching von Ontologien aus dem Bereich der Lebenswissenschaften (*Anatomy* und *Large Bio Track*) erreicht GOMMA sehr gute Ergebnisse bezüglich Effektivität und Effizienz. Eine deutliche Verbesserung der Mappingqualität wird durch das Einbeziehen von Domänenwissen in der Match-Strategie erreicht. Dabei erwiesen sich die Komposition über domänenspezifische Mediatorontologien sowie die Verwendung eines multilingualen Übersetzungsdienstes zur Generierung verbesserter Synonymmengen als besonders wertvoll.

# Teil V

# Zusammenfassung und Ausblick

# 12 Zusammenfassung und Ausblick

## 12.1 Zusammenfassung

Die vorliegende Arbeit beschäftigte sich mit der Evolution ontologiebasierter Mappings aus dem Bereich der Lebenswissenschaften. Dabei wurde die generische Infrastruktur GOMMA zur Verwaltung und Analyse der Evolution von Ontologien und Mappings genutzt und erweitert. Nach einer Einführung wurden im zweiten Teil die Evolution von Ontologiemappings untersucht und Ansätze zu deren Adaptierung präsentiert. Der dritte Teil betrachtete die Evolution ontologiebasierter Annotationsmappings sowie den Einfluss der Änderungen auf funktionale Analysen. Der vierte Teil stellte Ansätze zum effizienten Matching besonders großer Ontologien sowie eine umfassende Evaluierung zum Ontologie-Matching mit GOMMA vor.

### 12.1.1 Evolution von Ontologiemappings

Bisherige Arbeiten untersuchten kaum die Evolution und Adaptierung von Ontologiemappings. Daher wurde zunächst eine vergleichende Analyse zur Evolution von Ontologiemappings für drei verschiedene Domänen der Lebenswissenschaften (Anatomie, Molekularbiologie und Chemie) durchgeführt. Die untersuchten Mappings wurden zwischen Ontologieversionen im Zeitraum 2006 – 2010 unter Verwendung typischer metadatenbasierter Match-Verfahren generiert. Basierend auf verschiedenen Maßen zum Vergleich der Evolutionsintensitäten zeigte die Analyse einen deutlichen Einfluss der Ontologieänderungen auf automatisch generierte Mappings. Während

Ontologien sowie Mappings in der Anatomiedomäne relativ stabil waren, änderten sie sich im Molekularbiologie- und Chemie-Szenario sehr stark. Ein komplexes, kontextbasiertes Match-Verfahren produzierte besonders instabile Ontologiemappings. Insgesamt führten Ontologieerweiterungen häufig zu Korrespondenzhinzufügungen, wohingegen reduzierende Änderungen eher Korrespondenzlöschungen auslösten.

Eine weitere Studie nutzte die Evolution von Ontologiemappings, um anhand der Historie der Ähnlichkeitswerte einzelner Korrespondenzen, Stabilitätswerte zu errechnen. Diese können als zusätzliches Qualitätskriterium zur Bewertung automatisch generierter Ontologiemappings dienen. Die Evaluierung für ein instanzbasiertes Ontologie-Matching-Szenario in den Lebenswissenschaften zeigte, dass Korrespondenzen mit gleichen Ähnlichkeitswerten durchaus unterschiedliche Stabilitätswerte aufweisen. Instanzbasierte Match-Verfahren hängen zusätzlich von der Evolution der verwendeten Instanzen und Annotationen ab und produzieren daher tendenziell instabilere Ergebnisse als metadatenbasierte Match-Verfahren. Anhand der vorgestellten Maße wurden automatisch generierte Korrespondenzen bewertet und in verschiedene Kategorien eingeteilt, so dass Experten bei deren manueller Verifikation unterstützt werden.

Die beiden Studien zur Mappingevolution zeigten, dass Ontologieänderungen Auswirkungen auf abhängige Ontologiemappings haben. Dementsprechend können bereits bestehende Ontologiemappings infolge der Weiterentwicklung von Ontologien ungültig werden, so dass ihre Migration auf aktuelle Ontologieversionen notwendig wird. In dieser Arbeit wurden zwei generische Ansätze zur (semi-)automatischen Adaptierung von Ontologiemappings infolge von Ontologieevolution vorgestellt. Beide Ansätze ermöglichen die Wiederverwendung unbeeinflusster, bereits bestätigter Korrespondenzen und passen nur diejenigen Teile eines Mappings an, welche Änderungen unterlagen. Der erste Adaptierungsalgorithmus basiert auf der Komposition des veralteten Ontologiemappings mit einem Mapping zwischen Ontologieversionen. Das migrierte Mapping kann um neue Korrespondenzen infolge von Konzepthinzufügungen erweitert werden. Alternativ ermöglicht ein neuartiger, Diff-basierter Adaptierungsalgorithmus die individuelle Behandlung von Änderungen. Für die verschiedenen, teilweise komplexen Änderungsarten eines Diff-Evolutionsmappings (z. B. Aufspaltung, Löschung, Hinzufügung eines Konzepts) wurden individuelle Adaptierungsstrategien vorgeschlagen. Beide Adaptierungverfahren berücksichtigen zudem die Semantik der migrierten Korrespondenzen und unterstützen eine Verifikation des Mappings durch Experten. Eine Evaluierung für sehr große, biomedizinische Ontologien und Mappings zeigte die hohe Effektivität beider Verfahren. Insbesondere das Matching hinzugefügter Konzepte ermöglichte die Erzeugung möglichst vollständiger Mappings. Während der kompositionsbasierte Ansatz konzeptionell einfacher ist, erlaubt der Diff-basierte Ansatz eine flexible, änderungsspezifische Mappingadaptierung und erreichte die bessere Mappingqualität, insbesondere für stark von Änderungen betroffene Mappings.

### 12.1.2 Evolution von Annotationsmappings

Neben Ontologiemappings werden auch ontologiebasierte Annotationsmappings durch Ontologieänderungen beeinflusst. Die Evolution von Annotationen sowie deren Auswirkungen wurde durch bisherige Forschungsarbeiten nicht untersucht. Der hier vorgestellte, generische Ansatz dient der Bewertung der Qualität von Annotationsmappings auf Basis der Annotationshistorie. Dazu wurden verschiedene Qualitätsmaße bezüglich der Stabilität und der Herkunft bzw. Erstellungsmethode von Annotationen vorgeschlagen. Eine umfassende Evaluierung zeigte Instabilitäten für zahlreiche Annotationen, insbesondere aufgrund von Instanzänderungen, temporären Annotationslöschungen sowie häufigen Aktualisierungen der Herkunftsinformationen. Die vorgestellten Maße unterstützen die Identifikation glaubwürdiger Annotationen für die Verwendung in weiterführenden Analysen.

Eine weitere Studie untersuchte erstmals die Auswirkungen der teilweise starken Evolution von Ontologien und Annotationen auf sogenannte funktionale Analysen. Diese identifizieren signifikante Ontologiekonzepte bzw. Eigenschaften für eine Menge biologischer Objekte. Ziel war es, herauszufinden, inwieweit die Evolution der Eingabedaten zur Veränderung der Analyseergebnisse oder sogar zu einer veränderten biologischen Interpretation führt. Dazu wurden die Ergebnisse der funktionalen Analysen für verschiedene Ontologie- und Annotationsversionen generiert und bezüglich der eingeführten Basis- und Regionenstabilität bewertet. Einige Ergebnisse veränderten sich stark, d. h. neue Konzepte kamen hinzu und einige zuvor signifikante Konzepte gingen durch Verwendung neuerer Versionen verloren. Da signifikante Konzepte jedoch häufig in der gleichen Ontologieregion lokalisiert und dementsprechend semantisch ähnlich sind, veränderte sich die Interpretation der Ergebnisse kaum. Insgesamt zeigte die Evaluierung für zwei reale sowie 50 simulierte Datensätze, dass funktionale Analysen relativ robust gegenüber Ontologie- und Annotationsevolution sind. Nutzer sollten sich dennoch bewusst sein, dass Änderungen in Ontologien und Annotationen Einfluss auf abhängige Anwendungen wie funktionale Analysen haben und dies bei der Interpretation ihrer Ergebnisse berücksichtigen.

### 12.1.3 Matching großer Ontologien

Weiterhin wurden effiziente Verfahren für das Matching sehr großer Ontologien entwickelt. Diese werden u. a. für den Abgleich neuer Konzepte während der Adaptierung von Ontologiemappings benötigt. Allgemein sind qualitativ hochwertige Ontologiemappings essentiell für zahlreiche Datenintegrationsaufgaben wie z. B. das Merging von Ontologien. Der Abgleich besonders großer Ontologien, wie sie im Bereich der Lebenswissenschaften auftreten, kann kaum manuell realisiert werden und selbst viele der existierenden Match-Systeme skalieren nicht für derartig große Match-Aufgaben.

Zunächst wurde ein kompositionsbasierter Ansatz zum indirekten Matching von Ontologien vorgestellt, wobei existierende Mappings zu Zwischenontologien wiederverwendet und miteinander kombiniert wurden. Der Ansatz basiert auf generischen Ontologie- und Mappingoperatoren wie `compose`, `match` und `extract`. Die Evaluierung für das Matching von Anatomieontologien betrachtete vielversprechende Mediatorontologien der Domäne (z. B. UMLS und Uberon). Insgesamt konnte das indirekte Matching eine sehr gute Mappingqualität von über 90% F-Measure erreichen. Diese konnte durch die zusätzliche Anwendung eines direkten Matchings bisher nicht verknüpfter Ontologiebereiche verbessert werden. Im Vergleich zu einem vollständigen direkten Abgleich konnte die alleinige Ausführung der Komposition die Mappingqualität verbessern und die Ausführungszeiten deutlich reduzieren.

Darüber hinaus wurden generelle Strategien für das parallele Ontologie-Matching unter Verwendung mehrerer Rechenknoten vorgestellt. Die Strategien zur *Inter-* und *Intra-Matcher-Parallelisierung* erlauben die parallele Ausführung vollständiger, unabhängiger Matcher sowie deren interne Parallelisierung. Durch eine größenbasierte Partitionierung der Eingabeontologien konnten eine gute Lastbalancierung und Skalierbarkeit erreicht werden, ohne die Qualität der generierten Mappings zu beeinflussen. Neben elementbasierten Match-Verfahren, konnten auch komplexere, struktur- und instanzbasierte Verfahren durch das Verwenden spezieller Kontextattribute parallel ausgeführt werden. Im Rahmen der Studie wurde eine verteilte Infrastruktur innerhalb des Systems GOMMA implementiert und für das Matching großer Ontologien aus dem Bereich der Lebenswissenschaften evaluiert. Die Ergebnisse zeigten die hohe Effizienz und Skalierbarkeit des parallelen Ontologie-Matchings, insbesondere für sehr große Match-Probleme.

Neben einer guten Effizienz und Skalierbarkeit für sehr große Match-Probleme ist insbesondere eine hohe Qualität automatisch generierter Mappings von großer Bedeutung. Die generische Match-Komponente des GOMMA-Systems wurde im Rahmen der OAEI 2012 vergleichend mit den Ergebnissen anderer Systeme evaluiert. GOMMA konnte erfolgreich sämtliche Aufgaben zum Ontologie-Matching in verschiedenen Domänen absolvieren. Dabei profitierte das System von der Anwendung des parallelen und kompositionsbasierten Matchings sowie einer Blocking-Strategie für große Ontologien. GOMMA erreichte insbesondere für das Matching von Ontologien aus dem Bereich der Lebenswissenschaften (*Anatomy* und *Large Bio Track*) sehr gute Ergebnisse bezüglich der Effektivität und Effizienz. Das Einbeziehen von Domänenwissen erwies sich als besonders wertvoll. So konnten die Verwendung domänenspezifischer Mediatorontologien sowie die automatische Übersetzung multilingualer Ontologien deutlich zur Verbesserung der Mappingqualität beitragen.

## 12.2 Ausblick

Die in dieser Arbeit vorgestellten Methoden und Algorithmen dienen als Grundlage zukünftiger Forschungsarbeiten zur Evolution ontologiebasierter Mappings. Die Verfahren wurden in verschiedenen Subdomänen der Lebenswissenschaften angewendet und evaluiert. In dieser Domäne entwickeln sich Erkenntnisse aufgrund des hohen Forschungsinteresses stets rasant weiter, so dass die hier untersuchten Methoden von besonderem Nutzen sind. Die vorgestellten Ansätze sind generisch konzipiert und können somit in weiteren Domänen wie beispielsweise den Geisteswissenschaften eingesetzt werden. Aufbauend auf den Methoden und Ergebnissen dieser Arbeit diskutieren die folgenden Abschnitte zukünftige Forschungsrichtungen.

**Bestimmung und Adaptierung semantischer Mappings:** Die in dieser Arbeit vorgestellten Algorithmen ermöglichen die Migration veralteter Ontologiemappings auf aktuellere Ontologieversionen. Die Ansätze berücksichtigen bereits, dass Korrespondenzen zwischen Ontologiekonzepten nicht zwingend eine Äquivalenzsemantik besitzen. Jedoch enthalten Ontologiemappings häufig keine komplexeren, semantischen Korrespondenzen. Um die Ausdrucksstärke und Korrektheit der Mappings zu verbessern, ist es sinnvoll, aktuelle Verfahren zur semantischen Anreicherung von Äquivalenzmappings [6] anzuwenden und weiterzuentwickeln. Ebenso beziehen Algorithmen zur Bestimmung eines Diff-Evolutionsmappings (z. B. COnto-Diff [75], PromptDiff [140]) nicht explizit die Beziehungstypen zwischen Konzepten unterschiedlicher Versionen ein. Dabei spiegeln gerade Operationen wie das Aufspalten (*split*) oder Zusammenführen (*merge*) von Konzepten eine komplexe Semantik wider. Zukünftig bedarf es erweiterter Verfahren, die verschiedene Beziehungstypen wie *is-a* und *part-of* in Änderungsoperationen berücksichtigen. Während der Mappingadaptierung ist es notwendig, die semantischen Korrespondenzen miteinander zu kombinieren. In dieser Arbeit wurden bereits erste Regeln zur Verknüpfung einiger Beziehungstypen vorgeschlagen. Diese sollten zukünftig erweitert und anhand realer semantischer Mappings evaluiert werden. Zudem sind Modifikationen der Algorithmen sinnvoll, um auch die Adaptierung anderer ontologiebasierter Mappings wie z. B. Annotationen biologischer Objekte zu ermöglichen. Nutzer sollten durch ein Werkzeug unterstützt werden, das ihnen die Aktualisierung veralteter, ontologiebasierter Mappings ermöglicht. Ein solches Werkzeug erlaubt auch eine interaktive Qualitätsanalyse und Verifikation der vorgeschlagenen Korrespondenzen durch Experten.

**Qualität von Annotationen:** Die Ergebnisse dieser Arbeit zeigten signifikante Instabilitäten funktionaler Annotationen auf. In diesem Zusammenhang stellt sich die Frage wie glaubwürdig und qualitativ hochwertig Annotationen sind. Aktuelle Studien bestätigen die Ergebnisse aus Kapitel 7 und 8, wonach Annotationsdatensätze teilweise sehr instabil sind [57] und Änderungen in Annota-

tionen Einfluss auf Ergebnisse funktionaler Analysen haben [30]. Beispielsweise bestimmten die Autoren in [57] die Ähnlichkeit von Genen anhand überlappender Annotationen und zeigten auf, dass 20% der untersuchten Gene nach zwei Jahren nicht mehr auf sich selbst abgebildet werden. Eine weitere aktuelle Arbeit [163] zeigte, dass automatisch generierte GO-Annotationen (*Evidence Code* IEA), die von manuell verifizierten *Swiss-Prot Keywords* abgeleitet wurden, eine wesentlich höhere Glaubwürdigkeit aufweisen als jene, die auf Sequenzähnlichkeiten in *Ensembl Compara* basieren. Viele Analysen und Algorithmen nutzen fragwürdige, automatisch generierte Annotationen, um wiederum eine Vorhersage z. B. für neue funktionale Annotationen zu treffen. Dadurch verlieren die generierten Ergebnisse zunehmend an Qualität. Demgegenüber steht das Problem, dass viele Applikationen auf automatisch bestimmte Annotationen angewiesen sind (z. B. funktionale Analysen, siehe Kapitel 8), da sie sonst aufgrund reduzierter Eingabedaten keine oder kaum Ergebnisse produzieren.

Die Erkenntnisse dieser Arbeit und aktueller Studien unterstreichen den Bedarf einer präziseren Bewertung der Annotationsqualität, als es bisher z. B. durch *Evidence Codes* ermöglicht wird. Es ist sinnvoll, Annotationsdatensätze mit Qualitätskennzahlen (z. B. Stabilität) zu versehen, welche dann in Anwendungen wie funktionalen Analysen oder Algorithmen zur Vorhersage neuer Annotationen verwendet werden. Beispielsweise kann die Vorhersage von Annotationen auf Basis eines Graphalgorithmus [173] Qualitätswerte der Eingabeannotationen als Kantengewichte nutzen, um so die Glaubwürdigkeit der neu generierten Annotationen zu verbessern. Die in dieser Arbeit vorgestellten, evolutionsbasierten Maße zur Bewertung der Annotationsqualität dienen als Basis weiterer Verfahren und sollten Nutzern zugänglich gemacht werden. Beispielsweise können existierende Werkzeuge zur Analyse der Ontologieevolution (z. B. CODEX [78], REx [29]) ebenso Informationen zur Evolution und Qualität von Annotationen zur Verfügung stellen.

**Optimierte Verfahren des Ontologie-Matchings:** Die in dieser Arbeit vorgestellten Match-Verfahren ermöglichen insbesondere den Abgleich sehr großer Ontologien. Die Effizienz konnte u. a. durch Partitionierung der Eingabeontologien und die parallele Ausführung kleinerer Teilaufgaben auf mehreren Prozessoren (CPUs) verbessert werden. Heutzutage werden moderne Technologien wie *Graphical Processing Units* (GPUs) eingesetzt, um rechenintensive Algorithmen wie z. B. zur Duplikaterkennung [54] zu realisieren. Es ist vielversprechend auch linguistische Match-Verfahren, wie sie in GOMMA [102] zum Einsatz kommen, durch den Einsatz von GPUs zu beschleunigen. Eine erste eigene Arbeit [82] erweitert die hier vorgestellte Methode zum parallelen Ontologie-Matching und ermöglicht eine GPU-basierte Ausführung der n-Gram-Ähnlichkeitsberechnung in GOMMA. Der erfolgreiche Einsatz von GPUs erfordert adäquate Methoden, um einigen inhärenten Limitierungen zu begegnen, beispielsweise der ausschließlichen Verwendung primitiver Daten-

typen, einem teuren Datentransfer zur GPU sowie der vorherigen Festlegung des benötigten Speichers. Aufbauend auf ersten Erkenntnissen für n-Gram sollen zukünftig weitere Ähnlichkeitsmetriken auf GPUs realisiert werden.

Darüber hinaus können insbesondere algorithmische Optimierungen wie fortgeschrittene Blocking- und Partitionierungsstrategien oder die Verwendung indexierter Tokens beim String-Vergleich helfen, die Performanz des Ontologie-Matchings zu verbessern. Das vorgeschlagene Blocking-Verfahren kann wie folgt erweitert werden. Neben einem effizienten, exakten Matching soll das kompositionsbasierte Matching über vielversprechende Mediatorontologien durchgeführt werden, so dass möglichst vollständige Mappings erzeugt werden. Anstatt sämtliche, bisher nicht verknüpfte Konzepte einer relevanten Teilontologie abzugleichen, werden anschließend nur Konzepte im strukturellen Kontext der initial detektierten Korrespondenzen für das weitere Matching betrachtet. Dies verbessert die Genauigkeit der Mappings, da ähnlich zu existierenden Partitionierungsstrategien nur potenziell überlappende Bereiche der Ontologien verglichen werden. Um Inkonsistenzen zu vermeiden, soll GOMMA zudem um neue, effiziente Techniken der Mappingreparatur erweitert werden.

Die Evaluierung des kompositionsbasierten Matchings über verschiedene Mediatorontologien (GOMMA-bk) zeigte, dass spezifisches Domänenwissen zu zusätzlichen korrekten Korrespondenzen führen kann. In ähnlicher Weise nutzte AgreementMakerLight (AML) zur OAEI 2013 [51] Hintergrundwissen in Form manuell verifizierter *Cross-Reference*-Mappings aus Uberon und UMLS und erzielte das beste durchschnittliche F-Measure von 96% im *Large BioMed Track*. Aktuelle, eigene Arbeiten [79, 77] zeigen, dass neben der Komposition zweier Mappings die Verwendung längerer Mappingketten über mehrere Zwischenontologien vielversprechend ist. Zudem kann die Güte der Zwischenontologien für eine Domäne bewertet werden. So ist UMLS für biomedizinische Match-Probleme, jedoch nicht für das Matching geographischer Datenquellen wertvoll. Neue Methoden könnten aus einem globalen Wissenskorpus geeignete Mediatorontologien einer Domäne automatisch erkennen und nutzen.

Die Diskussion möglicher, zukünftiger Arbeiten zeigt, dass in den untersuchten Bereichen noch offene Probleme existieren. Die vorgestellten Methoden und Algorithmen liefern einen Beitrag zur Erforschung der Evolution ontologiebasierter Mappings im Bereich der Lebenswissenschaften. Die generischen Verfahren und Algorithmen dienen als Grundlage für weitere Entwicklungen und tragen zudem zur Lösung ähnlicher Probleme in anderen Domänen bei.

# Teil VI

# Anhang

# A

# Einfluss der Ontologieevolution auf funktionale Analysen

## A.1 Evolution von GO

| | **2007** | **2008** | **2009** | **2010** | **growth** |
|---|---|---|---|---|---|
| **BP** | 12,905\|21,991 | 14,833\|26,181 | 16,286\|32,257 | 18,515\|37,914 | 1.4\|1.7 |
| **MF** | 8,050\|9,301 | 8,779\|10,144 | 9,214\|10,604 | 9,458\|10,937 | 1.2\|1.2 |
| **CC** | 1,977\|3,708 | 2,162\|4,022 | 2,370\|4,449 | 2,746\|5,084 | 1.4\|1.4 |

Tabelle A.1: Evolution der drei GO-Subontologien zwischen 2007 und 2010. Die Tabelle zeigt für jedes Jahr und jede Subontologie (Biologische Prozesse (BP), Molekulare Funktionen (MF), Zelluläre Komponenten (CC)) die Anzahl der Konzepte $|C|$ und Beziehungen $|R|$ ($|C|||R|$). Die letzte Spalte zeigt das Wachstum von der ersten (2007) zur letzten (2010) betrachteten Version für $C$ und $R$.

| accession | concept name | avg_costs |
|---|---|---|
| GO:0008233 | peptidase activity | 1.70 |
| GO:0009055 | electron carrier activity | 1.03 |
| GO:0004672 | protein kinase activity | 0.82 |
| GO:0030533 | triplet codon-amino acid adaptor activity | 0.65 |
| GO:0032791 | lead ion binding | 0.55 |
| GO:0004721 | phosphoprotein phosphatase activity | 0.53 |
| GO:0016740 | transferase activity | 0.48 |
| GO:0016787 | hydrolase activity | 0.42 |
| GO:0003824 | catalytic activity | 0.42 |
| GO:0003674 | molecular_function | 0.38 |
| GO:0005215 | transporter activity | 0.38 |
| GO:0000166 | nucleotide binding | 0.35 |
| GO:0005509 | calcium ion binding | 0.31 |
| GO:0016301 | kinase activity | 0.29 |
| GO:0004518 | nuclease activity | 0.28 |
| GO:0003677 | DNA binding | 0.27 |
| GO:0030234 | enzyme regulator activity | 0.27 |
| GO:0003676 | nucleic acid binding | 0.23 |
| GO:0005488 | binding | 0.23 |
| GO:0003774 | motor activity | 0.23 |
| GO:0016209 | antioxidant activity | 0.22 |
| GO:0005102 | receptor binding | 0.21 |
| GO:0005515 | protein binding | 0.20 |
| GO:0008289 | lipid binding | 0.20 |
| GO:0030246 | carbohydrate binding | 0.20 |
| GO:0004872 | receptor activity | 0.19 |
| GO:0005216 | ion channel activity | 0.18 |
| GO:0004871 | signal transducer activity | 0.18 |
| GO:0003723 | RNA binding | 0.17 |
| GO:0005198 | structural molecule activity | 0.15 |
| GO:0019825 | oxygen binding | 0.15 |
| GO:0045735 | nutrient reservoir activity | 0.15 |
| GO:0005326 | neurotransmitter transporter activity | 0.15 |
| GO:0045182 | translation regulator activity | 0.15 |
| GO:0008135 | translation factor activity, nucleic acid binding | 0.12 |
| GO:0008092 | cytoskeletal protein binding | 0.11 |
| GO:0003779 | actin binding | 0.08 |
| GO:0030528 | transcription regulator activity | 0.07 |
| GO:0003682 | chromatin binding | 0.05 |
| GO:0003700 | sequence-specific DNA binding transcription factor activity | 0.00 |
| GO:0031386 | protein tag | 0.00 |

Tabelle A.2: Evolution der GO-MF-Slim-Terme. Die Evolutionsintensitäten bzw. durchschnittlichen Kosten ($avg\_costs$) einer Region wurden anhand der in [76] vorgestellten Methode berechnet. Dazu wurden monatliche Versionen der gesamten MF-Subontologie zwischen 2007 und 2010 betrachtet, jedoch sind zur Übersicht nur MF-Slim-Terme dargestellt. Die Region im Subgraph eines Slim-Terms mit niedrigen/hohen $avg\_costs$ ist stabil/instabil (grün/rot).

## A.2 Pseudocode zur Berechnung der Konzeptregionen und Regionenstabilität

**Algorithmus 8:** groupConcepts

**Input** : Ergebnismenge $ER$ mit signifikanten Konzepten, Distanz $d$
**Output** : Menge von Konzeptregionen $CR$

```
1  Ungrouped ← ER;
2  CR ← ∅;
3  repeat    /* Check for all ungrouped concepts to which region they belong */
4  |  Region ← {Ungrouped.next()};
5  |  Ungrouped ← Ungrouped \ Region;
6  |  repeat  /* Successive search for neighbors and increase of the region */
7  |  |  Neighbors ← getNextNeighbors(Region, Ungrouped, d);
8  |  |  Region ← Region ∪ Neighbors;
9  |  |  Ungrouped ← Ungrouped \ Neighbors;
10 |  until Neighbors = ∅;
11 |  CR ← CR ∪ Region;
12 until Ungrouped = ∅;
13 return CR;
```

**Algorithmus 9:** getNextNeighbors

**Input** : Aktuelle Region $Region$, nicht-gruppierte signifikante Konzepte $Ungrouped$, Distanz $d$
**Output** : Konzepte, die zur Eingaberegion gehören $Neighbors$

```
1  Neighbors ← ∅;
2  foreach c1 ∈ Region do
3  |  foreach c2 ∈ Ungrouped do
4  |  |  if getDistance(c1, c2) ≤ d then        /* Add ungrouped c2 as neighbor if
   |  |  within d */
5  |  |  |  Neighbors ← Neighbors ∪ {c2};
6  return Neighbors;
```

**Algorithmus 10:** computeRegionStability

**Input** : Menge von Regionen der ersten Ergebnismenge $CR^i$, Menge von Regionen der zweiten Ergebnismenge $CR^j$

**Output** : Regionenstabilität $RSstab_{region}$

```
1 CR_o^i ← ∅;
2 CR_o^j ← ∅;
3 foreach R1 ∈ CR^i do /* Compare regions of the two sets among each another */
4     foreach R2 ∈ CR^j do
5         if R1 ∩ R2 ≠ ∅ then /* Check if R1-R2 overlap; if so, add to CR_o's */
6             CR_o^i ← CR_o^i ∪ {R1};
7             CR_o^j ← CR_o^j ∪ {R2};
8 RSstab_region ← (|CR_o^i|+|CR_o^j|) / (|CR^i|+|CR^j|);
9 return RSstab_region;
```

## A.3 Evolutionsanalysen der realen Datensätze

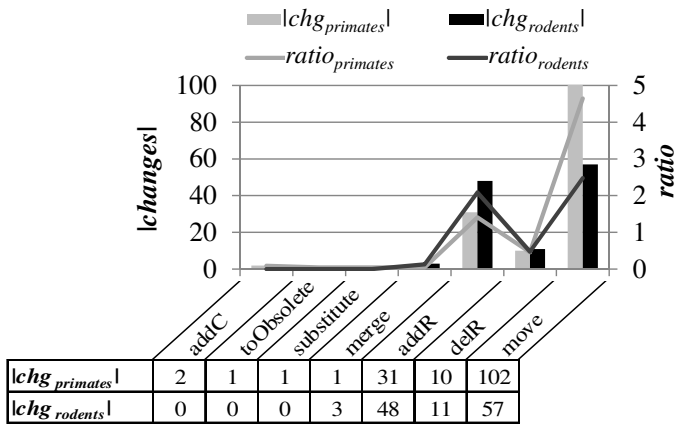
| | **GO concepts** | **annotated genes** | **GO annotations** |
|---|---|---|---|
| **2003** | 13,000 | 5,700 (4,000) | 28,000 (20,000) |
| **2010** | 31,000 | 8,200 (5,800) | 76,000 (55,000) |
| **growth factor** | 2.4 | 1.4 (1.4) | 2.7 (2.7) |

Tabelle A.3: Anzahl GO-Konzepte, Gene und Annotationen in 2003 und 2010 für Primaten (Rodentia). Es werden die gleichen GO-Eingabedaten für Primaten und Rodentia genutzt. Die Anzahl der betrachteten Gene und zugehörigen Annotationen unterscheidet sich entsprechend der relevanten Gengruppen für Primaten (Rodentia) aus der Orginialpublikation [108].

| PRIMATES | | | |
|---|---|---|---|
| 2007 | | 2010 | |
| Accession | Category Name | Accession | Category Name |
| GO:0001584 | rhodopsin-like receptor activity | GO:0050877 | neurological system process |
| GO:0051869 | physiological response to stimulus | GO:0007600 | sensory perception |
| GO:0050874 | organismal physiological process | GO:0004871 | signal transducer activity |
| GO:0050896 | response to stimulus | GO:0004872 | receptor activity |
| GO:0050877 | neurophysiological process | GO:0004888 | transmembrane receptor activity |
| GO:0007600 | sensory perception | GO:0007166 | cell surface receptor linked signal transduction |
| GO:0004871 | signal transducer activity | GO:0004984 | olfactory receptor activity |
| GO:0004872 | receptor activity | GO:0007186 | G-protein coupled receptor protein signaling pathway |
| GO:0004888 | transmembrane receptor activity | GO:0007606 | sensory perception of chemical stimulus |
| GO:0007166 | cell surface receptor linked signal transduction | GO:0007608 | sensory perception of smell |
| GO:0004984 | olfactory receptor activity | GO:0004930 | G-protein coupled receptor activity |
| GO:0007186 | G-protein coupled receptor protein signaling pathway | GO:0044425 | membrane part |
| GO:0007606 | sensory perception of chemical stimulus | GO:0016020 | membrane |
| GO:0007608 | sensory perception of smell | GO:0016021 | integral to membrane |
| GO:0004930 | G-protein coupled receptor activity | GO:0031224 | intrinsic to membrane |
| GO:0044425 | membrane part | GO:0050890 | cognition |
| GO:0016020 | membrane | GO:0003008 | organ system process |
| GO:0016021 | integral to membrane | GO:0060089 | molecular transducer activity |
| GO:0031224 | intrinsic to membrane | | |

| RODENTS | | | |
|---|---|---|---|
| 2007 | | 2010 | |
| Accession | Category Name | Accession | Category Name |
| GO:0002245 | physiological response to wounding | GO:0006955 | immune response |
| GO:0051869 | physiological response to stimulus | GO:0002376 | immune system process |
| GO:0002217 | physiological defense response | GO:0006952 | defense response |
| GO:0051240 | positive regulation of organismal physiological process | GO:0006954 | inflammatory response |
| GO:0002526 | acute inflammatory response | GO:0031226 | intrinsic to plasma membrane |
| GO:0050896 | response to stimulus | GO:0009611 | response to wounding |
| GO:0002684 | positive regulation of immune system process | GO:0005576 | extracellular region |
| GO:0050778 | positive regulation of immune response | GO:0006950 | response to stress |
| GO:0002682 | regulation of immune system process | | |
| GO:0050776 | regulation of immune response | | |
| GO:0044459 | plasma membrane part | | |
| GO:0009056 | catabolic process | | |
| GO:0009605 | response to external stimulus | | |
| GO:0009617 | response to bacterium | | |
| GO:0042742 | defense response to bacterium | | |
| GO:0006955 | immune response | | |
| GO:0002376 | immune system process | | |
| GO:0006952 | defense response | | |
| GO:0006954 | inflammatory response | | |
| GO:0031226 | intrinsic to plasma membrane | | |
| GO:0009611 | response to wounding | | |
| GO:0005576 | extracellular region | | |
| GO:0006950 | response to stress | | |

Abbildung A.1: Signifikante Konzepte für Primaten und Rodentia 2007 und 2010. Rot/grün hebt fehlende/neue, signifikante Konzepte hervor (bezüglich Evolution 2007 $\rightarrow$ 2010); weiße Konzepte überlappen für beide Versionen.




| | addC | toObsolete | substitute | merge | addR | delR | move |
|---|---|---|---|---|---|---|---|
| $\|chg_{primates}\|$ | 2 | 1 | 1 | 1 | 31 | 10 | 102 |
| $\|chg_{rodents}\|$ | 0 | 0 | 0 | 3 | 48 | 11 | 57 |

Abbildung A.2: Ontologieänderungen für Primaten- und Rodentia-Datensätze. Verwendung von COnto-Diff zwischen $GO^{01-2007}$ und $GO^{01-2010}$. Das COnto-Diff-Ergebnis wurde auf Änderungsoperationen, welche signifikante Konzepte (der alten oder neuen Version) enthalten reduziert. Attributänderungen sind nicht dargestellt. Die Rate (ratio) normalisiert die Anzahl der durch COnto-Diff bestimmten Änderungsoperationen mit der Anzahl der betrachteten, signifikanten Konzepte (22 für Primaten, 23 für Rodentia).

### A.3.1 Analyse automatisch generierter Annotationen

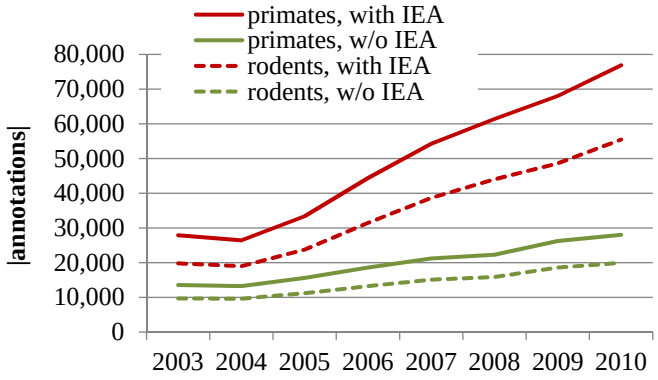



Abbildung A.3: Anzahl Annotationen mit und ohne IEA-*Evidence* (*Evidence Code* IEA = *inferred from electronic annotation*) zwischen 2003 und 2010 bezüglich der gegebenen Datensätze für Primaten und Rodentia.

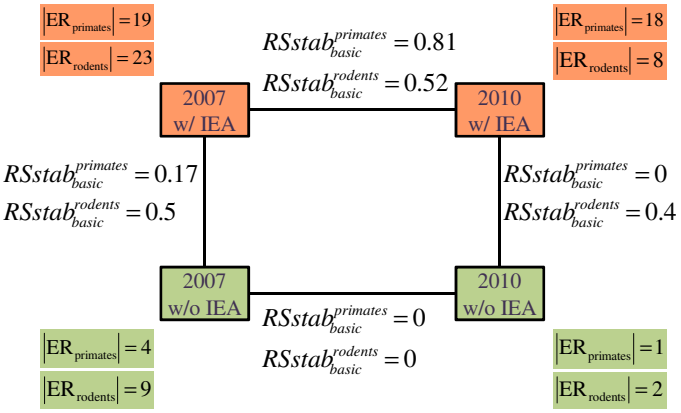



Abbildung A.4: Ergebnisstabilität mit (w/) oder ohne (w/o) automatisch generierte Annotationen (*Evidence Code* IEA) in 2007 und 2010 für Primaten und Rodentia. Jede Ecke zeigt die Ergebnisgröße ($|ER|$) jeweils für das betrachtete Jahr, die Spezies und Annotationsmenge (w/, w/o IEA). Die Kanten geben die Stabilitätswerte an, die aus dem Vergleich der jeweiligen Ergebnismengen (der Ecken) resultieren.

## A.4 Simulierte Datensätze

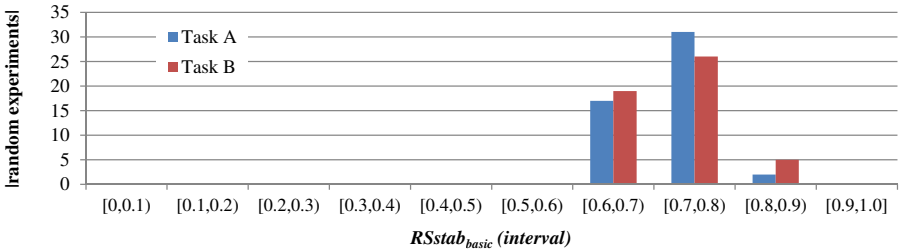



Abbildung A.5: Verteilung der Stabilität $RSstab_{basic}$ der Ergebnismengen für 50 simulierte Datensätze für Task A und B.

| random set no. | $\|cluster_{old}\|$ | $\|cluster_{new}\|$ | \|disappearing clusters\| | \|arising clusters\| | $RSstab_{basic}$ | $RSstab_{region}$ |
|---|---|---|---|---|---|---|
| **1** | 47 | 59 | 13 | 22 | 0.68 | 0.67 |
| **2** | 48 | 42 | 7 | 13 | 0.61 | 0.78 |
| **3** | 59 | 63 | 8 | 20 | 0.55 | 0.77 |
| **4** | 41 | 57 | 9 | 21 | 0.59 | 0.69 |
| **5** | 43 | 50 | 9 | 19 | 0.63 | 0.70 |
| **6** | 60 | 67 | 11 | 23 | 0.66 | 0.73 |
| **7** | 50 | 52 | 12 | 20 | 0.64 | 0.69 |
| **8** | 42 | 46 | 7 | 15 | 0.62 | 0.75 |
| **9** | 43 | 48 | 15 | 11 | 0.63 | 0.71 |
| **10** | 34 | 37 | 5 | 12 | 0.64 | 0.76 |
| **11** | 47 | 57 | 7 | 14 | 0.61 | 0.80 |
| **12** | 37 | 44 | 5 | 19 | 0.61 | 0.70 |
| **13** | 48 | 56 | 7 | 23 | 0.62 | 0.71 |
| **14** | 42 | 49 | 8 | 23 | 0.60 | 0.66 |
| **15** | 54 | 62 | 10 | 14 | 0.60 | 0.79 |
| **16** | 40 | 59 | 8 | 27 | 0.63 | 0.65 |
| **17** | 47 | 33 | 11 | 10 | 0.61 | 0.74 |
| **18** | 46 | 47 | 10 | 15 | 0.61 | 0.73 |
| **19** | 47 | 58 | 8 | 19 | 0.62 | 0.74 |
| **20** | 44 | 59 | 4 | 20 | 0.62 | 0.77 |
| **21** | 49 | 54 | 12 | 16 | 0.61 | 0.73 |
| **22** | 56 | 70 | 10 | 29 | 0.62 | 0.69 |
| **23** | 32 | 53 | 5 | 26 | 0.62 | 0.64 |
| **24** | 48 | 60 | 11 | 22 | 0.58 | 0.69 |
| **25** | 42 | 36 | 9 | 13 | 0.60 | 0.72 |
| **26** | 36 | 43 | 7 | 13 | 0.70 | 0.75 |
| **27** | 46 | 66 | 7 | 26 | 0.66 | 0.71 |
| **28** | 42 | 68 | 7 | 30 | 0.64 | 0.66 |
| **29** | 46 | 52 | 6 | 12 | 0.67 | 0.82 |
| **30** | 50 | 46 | 11 | 16 | 0.62 | 0.72 |
| **31** | 43 | 55 | 8 | 19 | 0.63 | 0.72 |
| **32** | 49 | 52 | 8 | 20 | 0.63 | 0.72 |
| **33** | 32 | 37 | 5 | 11 | 0.64 | 0.77 |
| **34** | 58 | 62 | 11 | 18 | 0.67 | 0.76 |
| **35** | 40 | 51 | 10 | 19 | 0.58 | 0.68 |
| **36** | 40 | 55 | 8 | 22 | 0.64 | 0.68 |
| **37** | 41 | 64 | 6 | 19 | 0.61 | 0.76 |
| **38** | 26 | 39 | 8 | 18 | 0.58 | 0.60 |
| **39** | 45 | 47 | 11 | 18 | 0.63 | 0.68 |
| **40** | 48 | 56 | 12 | 21 | 0.64 | 0.68 |
| **41** | 52 | 60 | 10 | 23 | 0.62 | 0.71 |
| **42** | 49 | 65 | 12 | 20 | 0.66 | 0.72 |
| **43** | 36 | 48 | 5 | 21 | 0.63 | 0.69 |
| **44** | 43 | 52 | 12 | 19 | 0.64 | 0.67 |
| **45** | 43 | 47 | 5 | 15 | 0.62 | 0.78 |
| **46** | 42 | 53 | 5 | 18 | 0.64 | 0.76 |
| **47** | 55 | 64 | 14 | 15 | 0.59 | 0.76 |
| **48** | 47 | 61 | 10 | 25 | 0.62 | 0.68 |
| **49** | 63 | 56 | 13 | 21 | 0.63 | 0.71 |
| **50** | 63 | 60 | 13 | 19 | 0.65 | 0.74 |
| **average** | **45.6** | **53.5** | **8.9** | **18.9** | **0.625** | **0.719** |

Tabelle A.4: Detailergebnisse für 50 simulierte Datensätze - Task A.

| random set no. | $\|cluster_{old}\|$ | $\|cluster_{new}\|$ | \|disappearing clusters\| | \|arising clusters\| | $RSstab_{basic}$ | $RSstab_{region}$ |
|---|---|---|---|---|---|---|
| **1** | 54 | 46 | 4 | 14 | 0.52 | 0.82 |
| **2** | 35 | 42 | 3 | 19 | 0.55 | 0.71 |
| **3** | 40 | 49 | 3 | 24 | 0.54 | 0.70 |
| **4** | 35 | 38 | 2 | 20 | 0.51 | 0.70 |
| **5** | 38 | 39 | 5 | 13 | 0.54 | 0.77 |
| **6** | 51 | 46 | 8 | 22 | 0.54 | 0.69 |
| **7** | 46 | 52 | 4 | 28 | 0.47 | 0.67 |
| **8** | 33 | 33 | 5 | 21 | 0.55 | 0.61 |
| **9** | 44 | 48 | 2 | 26 | 0.49 | 0.70 |
| **10** | 44 | 70 | 3 | 39 | 0.53 | 0.63 |
| **11** | 55 | 52 | 7 | 29 | 0.55 | 0.66 |
| **12** | 44 | 51 | 4 | 30 | 0.44 | 0.64 |
| **13** | 42 | 49 | 4 | 29 | 0.50 | 0.64 |
| **14** | 49 | 52 | 7 | 18 | 0.54 | 0.75 |
| **15** | 37 | 55 | 2 | 33 | 0.51 | 0.62 |
| **16** | 43 | 48 | 9 | 24 | 0.51 | 0.64 |
| **17** | 40 | 53 | 2 | 27 | 0.51 | 0.69 |
| **18** | 41 | 46 | 5 | 20 | 0.50 | 0.71 |
| **19** | 43 | 49 | 1 | 25 | 0.50 | 0.72 |
| **20** | 53 | 41 | 6 | 16 | 0.50 | 0.77 |
| **21** | 37 | 46 | 2 | 27 | 0.54 | 0.65 |
| **22** | 35 | 44 | 3 | 19 | 0.49 | 0.72 |
| **23** | 42 | 42 | 5 | 19 | 0.53 | 0.71 |
| **24** | 39 | 42 | 5 | 17 | 0.56 | 0.73 |
| **25** | 33 | 32 | 5 | 16 | 0.53 | 0.68 |
| **26** | 30 | 40 | 4 | 21 | 0.54 | 0.64 |
| **27** | 41 | 48 | 3 | 25 | 0.56 | 0.69 |
| **28** | 33 | 38 | 3 | 18 | 0.53 | 0.70 |
| **29** | 44 | 33 | 3 | 18 | 0.49 | 0.73 |
| **30** | 40 | 56 | 5 | 27 | 0.58 | 0.67 |
| **31** | 46 | 43 | 2 | 15 | 0.57 | 0.81 |
| **32** | 46 | 48 | 3 | 21 | 0.51 | 0.74 |
| **33** | 55 | 41 | 4 | 16 | 0.54 | 0.79 |
| **34** | 44 | 35 | 3 | 13 | 0.52 | 0.80 |
| **35** | 45 | 50 | 2 | 25 | 0.52 | 0.72 |
| **36** | 40 | 38 | 5 | 15 | 0.54 | 0.74 |
| **37** | 35 | 43 | 3 | 19 | 0.48 | 0.72 |
| **38** | 53 | 62 | 7 | 30 | 0.52 | 0.68 |
| **39** | 55 | 39 | 5 | 20 | 0.51 | 0.73 |
| **40** | 41 | 30 | 6 | 14 | 0.54 | 0.72 |
| **41** | 33 | 40 | 5 | 22 | 0.47 | 0.63 |
| **42** | 45 | 48 | 4 | 24 | 0.54 | 0.70 |
| **43** | 42 | 37 | 7 | 15 | 0.50 | 0.72 |
| **44** | 58 | 59 | 6 | 37 | 0.50 | 0.63 |
| **45** | 35 | 37 | 3 | 17 | 0.54 | 0.72 |
| **46** | 44 | 47 | 4 | 20 | 0.53 | 0.74 |
| **47** | 45 | 31 | 5 | 16 | 0.46 | 0.72 |
| **48** | 47 | 37 | 8 | 21 | 0.50 | 0.65 |
| **49** | 49 | 41 | 3 | 11 | 0.52 | 0.84 |
| **50** | 43 | 35 | 2 | 13 | 0.50 | 0.81 |
| **average** | **42.8** | **44.4** | **4.2** | **21.4** | **0.519** | **0.707** |

Tabelle A.5: Detailergebnisse für 50 simulierte Datensätze - Task B.

# B

# Evaluierung im Rahmen der OAEI 2012

| | Precision | Recall | F-Measure | Runtime in s |
|---|---|---|---|---|
| **GOMMA-bk** | 91.7 | 92.8 | 92.3 | 15 |
| **YAM++** | 94.3 | 85.8 | 89.8 | 69 |
| **CODI** | 96.6 | 82.7 | 89.1 | 880 |
| **LogMap** | 92.0 | 84.5 | 88.1 | 20 |
| **GOMMA** | 95.6 | 79.7 | 87.0 | 17 |
| **MapSSS** | 93.5 | 74.7 | 83.1 | 453 |
| **WeSeE** | 91.1 | 76.1 | 82.9 | 15833 |
| **LogMapLt** | 96.3 | 72.8 | 82.9 | 6 |
| **TOAST** | 85.4 | 75.5 | 80.1 | 3464 |
| **ServOMap** | 99.6 | 63.9 | 77.8 | 34 |
| **ServOMapLt** | 99.0 | 63.7 | 77.5 | 23 |
| **HotMatch** | 97.9 | 63.9 | 77.3 | 672 |
| **AROMA** | 86.5 | 68.7 | 76.6 | 29 |
| ***StringEquiv*** | *99.7* | *62.2* | *76.6* | - |
| **Wikimatch** | 86.4 | 67.5 | 75.8 | 17130 |
| **Optima** | 85.4 | 58.4 | 69.4 | 6460 |
| **Hertuda** | 69.0 | 67.3 | 68.1 | 317 |
| **MaasMatch** | 43.4 | 78.4 | 55.9 | 28890 |

Tabelle B.1: Ergebnisse *Anatomy Track*.

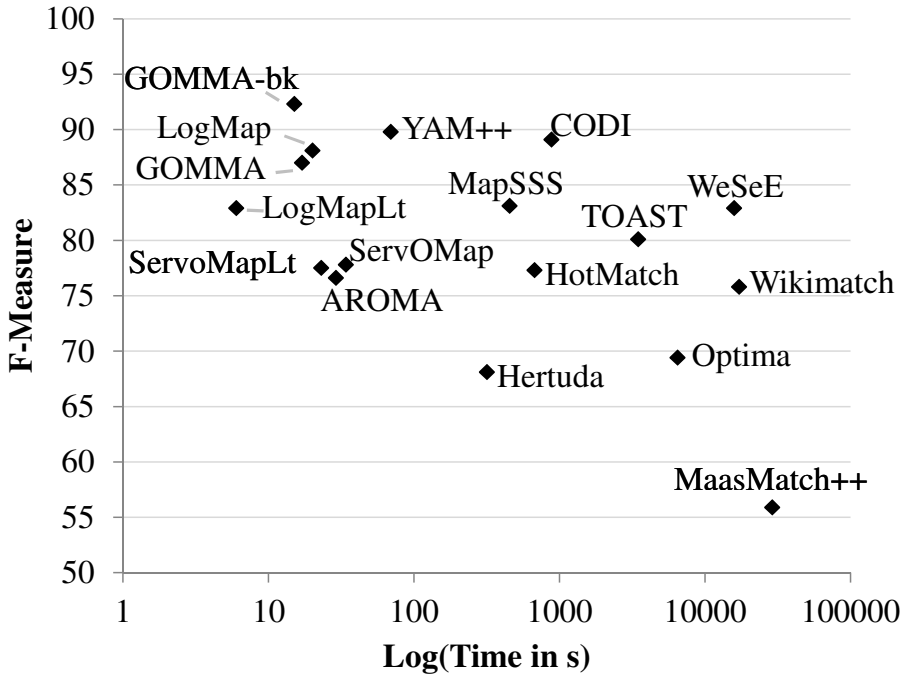



Abbildung B.1: *Anatomy Track*: F-Measure vs. Ausführungszeit.

| | small | | large | | whole | |
|---|---|---|---|---|---|---|
| | **source \|C\|** | **target \|C\|** | **source \|C\|** | **target \|C\|** | **source \|C\|** | **target \|C\|** |
| **FMA-NCIT** | 3696 | 6488 | 28861 | 25591 | 78989 | 66724 |
| **FMA-SCT*** | 10157 | 13412 | 50523 | 122464 | 78989 | 122464 |
| **SCT-NCIT*** | 51128 | 23958 | 122464 | 49795 | 122464 | 66724 |

*** The whole task takes SCT large fragments (40% of SCT).**

Tabelle B.2: Größen der Ontologiefragmente bzw. vollständigen Ontologien im *Large BioMed Track*, $|C|$ - Konzeptanzahl.

| | *Small* | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **FMA-NCIT** | | | | **FMA-SCT** | | | | **SCT-NCIT** | | | |
| | **Time** | **P** | **R** | **F** | **Time** | **P** | **R** | **F** | **Time** | **P** | **R** | **F** |
| **GOMMA-bk** | 26 | 93.6 | 91.3 | **92.4** | 148 | 89.3 | 91.3 | **90.3** | 226 | 93.9 | 62.1 | **74.7** |
| **YAM++** | 78 | 95.8 | 85.9 | **90.6** | 326 | 90.7 | 69.2 | **78.5** | 1901 | 94.6 | 60.9 | 74.1 |
| **LogMap-noe** | 18 | 93.2 | 87.6 | **90.3** | 63 | 91.0 | 68.8 | 78.4 | 211 | 89.5 | 65.2 | **75.4** |
| **LogMap** | 18 | 93.2 | 87.6 | **90.3** | 65 | 91.0 | 66.7 | 76.9 | 221 | 89.7 | 64.9 | **75.3** |
| **GOMMA** | 26 | 94.9 | 85.5 | 90.0 | 54 | 87.5 | 38.1 | 53.1 | 197 | 93.9 | 53.3 | 68.0 |
| **ServOMapLt** | 20 | 96.2 | 81.5 | 88.3 | 39 | 92.0 | 69.4 | **79.1** | 147 | 95.4 | 60.2 | 73.8 |
| **LogMapLt** | 8 | 94.5 | 80.6 | 87.0 | 14 | 93.8 | 18.3 | 30.7 | 54 | 94.5 | 55.7 | 70.1 |
| **ServOMap** | 25 | 96.9 | 76.5 | 85.5 | 46 | 91.8 | 65.5 | 76.4 | 153 | 96.5 | 56.3 | 71.1 |
| **HotMatch** | 4271 | 95.7 | 74.9 | 84.0 | 31718 | 84.3 | 21.4 | 34.2 | - | - | - | - |
| **Wikimatch** | 65399 | 78.8 | 86.0 | 82.2 | - | - | - | - | - | - | - | - |
| **AROMA** | 63 | 85.6 | 75.6 | 80.3 | 51191 | 52.7 | 32.7 | 40.4 | 15624 | 85.4 | 54.2 | 66.3 |
| **Hertuda** | 3327 | 58.0 | 85.8 | 69.2 | 17625 | 55.5 | 20.1 | 29.6 | - | - | - | - |
| **MaasMatch** | 27157 | 60.8 | 77.7 | 68.2 | - | - | - | - | - | - | - | - |
| **AUTOMSv2** | 62407 | 80.4 | 50.0 | 61.6 | - | - | - | - | - | - | - | - |
| **MapSSS** | 561 | 84.3 | 42.9 | 56.9 | 3129 | 75.1 | 30.9 | 43.8 | 27381 | 78.9 | 40.8 | 53.8 |

| | *Large* | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **FMA-NCIT** | | | | **FMA-SCT** | | | | **SCT-NCIT** | | | |
| | **Time** | **P** | **R** | **F** | **Time** | **P** | **R** | **F** | **Time** | **P** | **R** | **F** |
| **YAM++** | 245 | 90.2 | 83.2 | **86.6** | 3780 | 82.1 | 68.3 | **74.6** | 6127 | 85.9 | 60.5 | **71.0** |
| **ServOMapLt** | 95 | 89.1 | 80.7 | **84.7** | 234 | 88.3 | 68.9 | **77.4** | 363 | 86.4 | 59.4 | **70.4** |
| **GOMMA** | 69 | 85.7 | 82.7 | **84.2** | 437 | 38.9 | 25.9 | 31.1 | 527 | 79.5 | 52.7 | 63.4 |
| **GOMMA-bk** | 83 | 81.4 | 87.1 | 84.1 | 636 | 57.1 | 85.5 | 68.4 | 638 | 72.4 | 61.0 | 66.2 |
| **LogMap-noe** | 74 | 87.2 | 79.8 | 83.3 | 521 | 83.7 | 64.2 | 72.7 | 575 | 87.9 | 62.4 | **73.0** |
| **LogMap** | 77 | 87.0 | 79.3 | 83.0 | 484 | 83.3 | 62.3 | 71.2 | 514 | 87.4 | 57.1 | 69.1 |
| **ServOMap** | 98 | 91.3 | 75.7 | 82.8 | 315 | 87.7 | 65.4 | **74.9** | 282 | 89.1 | 55.8 | 68.6 |
| **LogMapLt** | 29 | 72.9 | 80.6 | 76.6 | 96 | 84.8 | 18.3 | 30.2 | 104 | 81.2 | 55.7 | 66.1 |
| **AROMA** | 7538 | 52.7 | 69.8 | 60.0 | 62801 | 66.1 | 19.6 | 30.3 | - | - | - | - |
| **MapSSS** | 30575 | 38.4 | 34.1 | 36.1 | - | - | - | - | - | - | - | - |
| **AUTOMSv2** | - | - | - | - | - | - | - | - | - | - | - | - |
| **Hertuda** | - | - | - | - | - | - | - | - | - | - | - | - |
| **HotMatch** | - | - | - | - | - | - | - | - | - | - | - | - |
| **MaasMatch** | - | - | - | - | - | - | - | - | - | - | - | - |
| **Wikimatch** | - | - | - | - | - | - | - | - | - | - | - | - |

| | *Whole* | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **FMA-NCIT** | | | | **FMA-SCT** | | | | **SCT-NCIT** | | | |
| | **Time** | **P** | **R** | **F** | **Time** | **P** | **R** | **F** | **Time** | **P** | **R** | **F** |
| **YAM++** | 1304 | 88.5 | 83.2 | **85.8** | 23900 | 81.4 | 68.1 | **74.2** | 30155 | 79.0 | 59.9 | **68.1** |
| **GOMMA** | 217 | 84.7 | 82.6 | **83.6** | 1994 | 35.0 | 24.2 | 28.6 | 1820 | 71.4 | 52.6 | 60.6 |
| **ServOMapLt** | 251 | 86.8 | 80.5 | **83.5** | 517 | 87.7 | 68.8 | **77.2** | 738 | 79.1 | 59.4 | **67.8** |
| **GOMMA-bk** | 231 | 80.1 | 87.0 | 83.4 | 1893 | 56.1 | 85.5 | 67.7 | 1940 | 66.3 | 60.8 | 63.5 |
| **LogMap-noe** | 206 | 86.6 | 78.7 | 82.5 | 791 | 81.6 | 62.1 | 70.6 | 1505 | 81.1 | 57.0 | 67.0 |
| **LogMap** | 131 | 86.0 | 78.3 | 81.9 | 612 | 82.8 | 62.1 | 71.0 | 955 | 81.4 | 57.0 | **67.1** |
| **ServOMap** | 204 | 89.2 | 75.5 | 81.8 | 532 | 87.0 | 65.3 | **74.6** | 654 | 82.9 | 55.6 | 66.6 |
| **LogMapLt** | 55 | 67.7 | 80.6 | 73.6 | 171 | 84.6 | 18.3 | 30.1 | 178 | 73.7 | 55.7 | 63.4 |
| **AROMA** | - | - | - | - | - | - | - | - | - | - | - | - |
| **AUTOMSv2** | - | - | - | - | - | - | - | - | - | - | - | - |
| **Hertuda** | - | - | - | - | - | - | - | - | - | - | - | - |
| **HotMatch** | - | - | - | - | - | - | - | - | - | - | - | - |
| **MaasMatch** | - | - | - | - | - | - | - | - | - | - | - | - |
| **MapSSS** | - | - | - | - | - | - | - | - | - | - | - | - |
| **Wikimatch** | - | - | - | - | - | - | - | - | - | - | - | - |

**Marked green - System among top 3 systems in track**
**Time in seconds; P = Precision, R = Recall, F = F-Measure**

Tabelle B.3: Ergebnisse *Large BioMed Track*.

# Literaturverzeichnis


[1] Agrawal, R., Borgida, A., Jagadish, H. V. Efficient management of transitive relationships in large data and knowledge bases. In *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data* (1989), pp. 253–262.

[2] Agrawal, R., Psaila, G. Active Data Mining. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD)* (1995), AAAI Press, pp. 3–8.

[3] Aguirre, J.-L., Eckert, K., Euzenat, J., Ferrara, A., van Hage, W. R., Hollink, L., Meilicke, C., Nikolov, A., Ritze, D., Scharffe, F., et al. Results of the Ontology Alignment Evaluation Initiative 2012. In *Proceedings of the 7th International Workshop on Ontology Matching (OM)* (2012), CEUR-WS.org, pp. 73–115.

[4] Aleksovski, Z., Klein, M., Ten Kate, W., Van Harmelen, F. Matching unstructured vocabularies using a background ontology. *Managing Knowledge in a World of Networks* (2006), 182–197.

[5] Algergawy, A., Schallehn, E., Saake, G. Improving xml schema matching performance using prüfer sequences. *Data & Knowledge Engineering 68*, 8 (2009), 728–747.

[6] Arnold, P., Rahm, E. Semantic Enrichment of Ontology Mappings: A Linguistic-based Approach. In *Proceedings of the 17th East-European Conference on Advances in Databases and Information Systems (ADBIS)* (2013), Springer, pp. 42–55.

[7] 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., et al. Gene Ontology: tool for the unification of biology. *Nature Genetics 25*, 1 (2000), 25–29.

[8] Aumueller, D., Do, H. H., Massmann, S., Rahm, E. Schema and ontology matching with COMA++. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (2005), pp. 906–908.

[9] Ba, M., Diallo, G. Large-scale biomedical ontology matching with ServO-

Map. *IRBM 34*, 1 (2013), 56–59.

[10] Barrell, D., Dimmer, E., Huntley, R., Binns, D., O'Donovan, C., Apweiler, R. The GOA database in 2009–an integrated Gene Ontology Annotation resource. *Nucleic Acids Research 37*, suppl 1 (2009), D396–D403.

[11] Baxter, R., Christen, P., Churches, T. A comparison of fast blocking methods for record linkage. In *Proceedings of the Ninth ACM SIGKDD Workshop on Data Cleaning, Record Linkage and Object Consolidation* (2003), vol. 3, Citeseer, pp. 25–27.

[12] Bellahsene, Z., Bonifati, A., Duchateau, F., Velegrakis, Y. On Evaluating Schema Matching and Mapping. In *Schema Matching and Mapping*. Springer, 2011, ch. 9, pp. 253–291.

[13] Bernstein, P., Melnik, S. Model Management 2.0: Manipulating Richer Mappings. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (2007), pp. 1–12.

[14] Bernstein, P. A. Applying Model Management to Classical Meta Data Problems. In *Proceedings of Conference on Innovative Database Research (CIDR)* (2003), pp. 209–220.

[15] Berriz, G., King, O., Bryant, B., Sander, C., Roth, F. Characterizing gene sets with FuncAssociate. *Bioinformatics 19*, 18 (2003), 2502–2504.

[16] Bodenreider, O. The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Research 32*, suppl 1 (2004), D267–D270.

[17] Bodenreider, O., Burgun, A. Linking the Gene Ontology to other biological ontologies. In *Proceedings of the ISMB 2005 SIG meeting on Bio-ontologies* (2005), pp. 17–18.

[18] Bodenreider, O., Hayamizu, T. F., Ringwald, M., De Coronado, S., Zhang, S. Of Mice and Men: Aligning Mouse and Human Anatomies. In *AMIA Annual Symposium Proceedings* (2005), pp. 61–65.

[19] Bodenreider, O., Stevens, R. Bio-ontologies: current trends and future directions. *Briefings in Bioinformatics 7*, 3 (2006), 256–274.

[20] Bose, R., Frew, J. Lineage retrieval for scientific data processing: A survey. *ACM Computing Surveys (CSUR) 37*, 1 (2005), 1–28.

[21] Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M., Bairoch, A. UniProtKB/Swiss-Prot. *Plant Bioinformatics 406* (2007), 89–112.

[22] Boyle, E., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J., Sherlock, G. GO::TermFinder–open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics 20*, 18 (2004), 3710–3715.

[23] Brown, E. G., Wood, L., Wood, S. The medical dictionary for regulatory activities (MedDRA). *Drug Safety 20*, 2 (1999), 109–117.

[24] Buneman, P., Chapman, A., Cheney, J. Provenance Management in Curated Databases. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of data* (2006), pp. 539–550.

[25] Buza, T., McCarthy, F., Wang, N., Bridges, S., Burgess, S. Gene Ontology annotation quality analysis in model eukaryotes. *Nucleic Acids Research 36*, 2:e12 (2008).

[26] Carbon, S., Ireland, A., Mungall, C. J., Shu, S., Marshall, B., Lewis, S., the AmiGO Hub, the Web Presence Working Group. AmiGO: online access to ontology and annotation data. *Bioinformatics 25*, 2 (2009), 288–289.

[27] Castano, S., De Antonellis, V., Fugini, M. G., Pernici, B. Conceptual schema analysis: techniques and applications. *ACM Transactions on Database Systems (TODS) 23*, 3 (1998), 286–333.

[28] Chalmel, F., Lardenois, A., Thompson, J. D., Muller, J., Sahel, J.-A., Léveillard, T., Poch, O. GOAnno: GO annotation based on multiple alignment. *Bioinformatics 21*, 9 (2005), 2095–2096.

[29] Christen, V. REx–eine Webapplikation zur Visualisierung der Evolution von Ontologien in den Lebenswissenschaften. In *Studentenkonferenz Informatik Leipzig (SKIL)* (2012), pp. 13–24.

[30] Clarke, E. L., Loguercio, S., Good, B. M., Su, A. I. A task-based approach for Gene Ontology evaluation. *Journal of Biomedical Semantics 4*, Suppl 1:S4 (2013).

[31] Cross, V., Silwal, P., Morell, D. Using a reference ontology with semantic similarity in ontology alignment. In *Proceedings of the 3rd International Conference on Biomedical Ontology (ICBO)* (2012), CEUR-WS.org.

[32] Cruz, I. F., Antonelli, F. P., Stroe, C. AgreementMaker: efficient matching for large real-world schemas and ontologies. *Proceedings of the VLDB Endowment 2*, 2 (2009), 1586–1589.

[33] Cruz, I. F., Stroe, C., Caimi, F., Fabiani, A., Pesquita, C., Couto, F. M., Palmonari, M. Using AgreementMaker to Align Ontologies for OAEI 2011. In *Proceedings of the 6th International Workshop on Ontology Matching (OM)* (2011), CEUR-WS.org, pp. 114–121.

[34] Curino, C. A., Moon, H. J., Zaniolo, C. Graceful database schema evolution: the PRISM workbench. *Proceedings of the VLDB Endowment 1*, 1 (2008), 761–772.

[35] Dahlquist, K. D., Salomonis, N., Vranizan, K., Lawlor, S. C., Conklin, B. R. GenMAPP, a new tool for viewing and analyzing microarray data

on biological pathways. *Nature Genetics 31*, 1 (2002), 19–20.

[36] Daraselia, N., Yuryev, A., Egorov, S., Mazo, I., Ispolatov, I. Automatic extraction of gene ontology annotation and its correlation with clusters in protein networks. *BMC Bioinformatics 8*, 1:243 (2007).

[37] Day-Richter, J., Harris, M. A., Haendel, M., The Gene Ontology OBO-Edit Working Group, Lewis, S. OBO-Edit–an ontology editor for biologists. *Bioinformatics 23*, 16 (2007), 2198–2200.

[38] Dean, J., Ghemawat, S. MapReduce: simplified data processing on large clusters. *Communications of the ACM 51*, 1 (2008), 107–113.

[39] Degtyarenko, K., de Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcántara, R., Darsow, M., Guedj, M., Ashburner, M. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Research 36*, suppl 1 (2008), D344–D350.

[40] Do, H.-H. Schema matching and mapping-based data integration. *Dissertation, Leipzig, Universität Leipzig* (2006).

[41] Do, H.-H., Rahm, E. COMA–A System for Flexible Combination of Schema Matching Approaches. In *Proceedings of 28th International Conference on Very Large Data Bases (VLDB)* (2002), Morgan Kaufmann, pp. 610–621.

[42] Do, H.-H., Rahm, E. Matching large schemas: Approaches and evaluation. *Information Systems 32*, 6 (2007), 857–885.

[43] Donnelly, K. SNOMED-CT: The Advanced Terminology and Coding System for eHealth. *Studies in Health Technology and Informatics–Medical and Care Compunetics 3 121* (2006), 279–290.

[44] Dos Reis, J., Pruski, C., Da Silveira, M., Reynaud, C. Analyzing and Supporting the Mapping Maintenance Problem in Biomedical Knowledge Organization Systems. In *Proceedings of Workshop on Semantic Interoperability in Medical Informatics (SIMI)* (2012).

[45] Dos Reis, J. C. Maintaining mappings valid between dynamic kos. In *The Semantic Web: Semantics and Big Data* (2013), Springer, pp. 650–655.

[46] Duan, S., Fokoue, A., Srinivas, K., Byrne, B. A clustering-based approach to ontology alignment. In *The Semantic Web–ISWC 2011–10th International Semantic Web Conference* (2011), Springer, pp. 146–161.

[47] Ehrig, M., Staab, S. QOM–quick ontology mapping. In *The Semantic Web–ISWC 2004–3rd International Semantic Web Conference* (2004), Springer, pp. 683–697.

[48] Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C. Ontology alignment evaluation initiative: Six years of experience. *Journal on Data Semantics XV* (2011), 158–192.

[49] Euzenat, J., Shvaiko, P. *Ontology Matching*. Springer, 2007.

[50] Fagin, R., Kolaitis, P. G., Popa, L., Tan, W.-C. Schema mapping evolution through composition and inversion. In *Schema Matching and Mapping*. Springer, 2011, ch. 7, pp. 191–222.

[51] Faria, D., Pesquita, C., Santos, E., Cruz, I. F., Couto, F. M. AgreementMakerLight Results for OAEI 2013. In *Proceedings of the 8th International Workshop on Ontology Matching (OM)* (2013).

[52] Fernández-Suárez, X. M., Galperin, M. Y. The 2013 Nucleic Acids Research Database Issue and the online molecular biology database collection. *Nucleic Acids Research 41*, D1 (2013), D1–D7.

[53] Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G. Ontology change: Classification and survey. *The Knowledge Engineering Review 23*, 2 (2008).

[54] Forchhammer, B., Papenbrock, T., Stening, T., Viehmeier, S., Draisbach, U., Naumann, F. Duplicate detection on gpus. In *Proceedings of 15. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW)* (2013), pp. 165–184.

[55] Gene Ontology Consortium. The Gene Ontology project in 2008. *Nucleic Acids Res. 36*, Database Issue (2008).

[56] Ghazvinian, A., Noy, N., Musen, M. Creating mappings for ontologies in biomedicine: Simple methods work. In *Proceedings of AMIA Annual Symposium* (2009).

[57] Gillis, J., Pavlidis, P. Assessing identity, redundancy and confounds in gene ontology annotations over time. *Bioinformatics 29*, 4 (2013), 476–482.

[58] Giunchiglia, F., Shvaiko, P., Yatskevich, M. S-Match: an algorithm and an implementation of semantic matching. *The Semantic Web: Research and Applications, First European Semantic Web Symposium (ESWS)* (2004), 61–75.

[59] Giunchiglia, F., Yatskevich, M., Shvaiko, P. Semantic matching: Algorithms and implementation. *Journal on Data Semantics IX 9* (2007), 1–38.

[60] Gross, A., Dos Reis, J. C., Hartung, M., Pruski, C., Rahm, E. Semi-automatic adaptation of mappings between life science ontologies. In *Proceedings of the 9th International Conference on Data Integration in the Life Sciences (DILS)* (2013), Springer, pp. 90–104.

[61] Gross, A., Hartung, M., Kirsten, T., Rahm, E. Estimating the Quality of Ontology-Based Annotations by Considering Evolutionary Changes. In *Proceedings of the 6th International Workshop on Data Integration in the Life Sciences (DILS)* (2009), Springer, pp. 71–87.

[62] Gross, A., Hartung, M., Kirsten, T., Rahm, E. On Matching Large Life Science Ontologies in Parallel. In *Proceedings of the 7th International Conference on Data Integration in the Life Sciences (DILS)* (2010), Springer, pp. 35–49.

[63] Gross, A., Hartung, M., Kirsten, T., Rahm, E. Mapping Composition for Matching Large Life Science Ontologies. In *Proceedings of the 2nd International Conference on Biomedical Ontology (ICBO)* (2011), CEUR-WS.org, pp. 109–116.

[64] Gross, A., Hartung, M., Kirsten, T., Rahm, E. GOMMA results for OAEI 2012. In *Proceedings of the 7th International Workshop on Ontology Matching (OM)* (2012), vol. 11, CEUR-WS.org.

[65] Gross, A., Hartung, M., Prüfer, K., Kelso, J., Rahm, E. Impact of Ontology Evolution on Functional Analyses. *Bioinformatics 28*, 20 (2012), 2671–2677.

[66] Gross, A., Hartung, M., Thor, A., Rahm, E. How do computed ontology mappings evolve?-A case study for life science ontologies. *Proceedings of the 2nd Joint Workshop on Knowledge Evolution and Ontology Dynamics (EvoDyn)* (2012).

[67] Groth, P., Weiss, B., Pohlenz, H.-D., Leser, U. Mining phenotypes for gene function prediction. *BMC Bioinformatics 9*, 1 (2008).

[68] Gruber, T. R. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition 5*, 2 (1993), 199–221.

[69] Grumbling, G., Strelets, V., Consortium, T. F. FlyBase: anatomical data, images and queries. *Nucleic Acids Research 34*, suppl 1 (2006), D484–D488.

[70] Hall, P. A., Dowling, G. R. Approximate String Matching. *ACM Computing Surveys (CSUR) 12*, 4 (1980), 381–402.

[71] Hamdi, F., Safar, B., Reynaud, C., Zargayouna, H. Alignment-based partitioning of large-scale ontologies. In *Advances in Knowledge Discovery and Management [Best of EGC 2009]* (2009), Springer, pp. 251–269.

[72] Hamdi, F., Zargayouna, H., Safar, B., Reynaud, C. TaxoMap in the OAEI 2008 Alignment Contest. In *Proceedings of the 3rd International Workshop on Ontology Matching (OM)* (2008), CEUR-WS.org, pp. 206–213.

[73] Hamosh, A., Scott, A. F., Amberger, J. S., Bocchini, C. A., McKusick, V. A. Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research 33*, suppl 1 (2005), D514–D517.

[74] Hartung, M. Evolution von Ontologien in den Lebenswissenschaften. *Dissertation, Leipzig, Universität Leipzig* (2011).

[75] Hartung, M., Gross, A., Rahm, E. COnto-Diff: generation of complex evolution mappings for life science ontologies. *Journal of Biomedical Informatics 46*, 1 (2013), 15–32.

[76] Hartung, M., Gross, A., Kirsten, T., Rahm, E. Discovering Evolving Regions in Life Science Ontologies. In *Proceedings of the 7th International Conference on Data Integration in the Life Sciences (DILS)* (2010), Springer, pp. 19–34.

[77] Hartung, M., Gross, A., Kirsten, T., Rahm, E. Effective mapping composition for biomedical ontologies. In *Proceedings of Workshop on Semantic Interoperability in Medical Informatics (SIMI)* (2012).

[78] Hartung, M., Gross, A., Rahm, E. CODEX: Exploration of semantic changes between ontology versions. *Bioinformatics 28*, 6 (2012), 895–896.

[79] Hartung, M., Gross, A., Rahm, E. Composition methods for link discovery. In *Proceedings of 15. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW)* (2013), pp. 261–277.

[80] Hartung, M., Kirsten, T., Gross, A., Rahm, E. OnEX: Exploring changes in life science ontologies. *BMC Bioinformatics 10*, 1:250 (2009).

[81] Hartung, M., Kirsten, T., Rahm, E. Analyzing the Evolution of Life Science Ontologies and Mappings. In *Proceedings of the 5th International Workshop on Data Integration in the Life Sciences (DILS)* (2008), Springer, pp. 11–27.

[82] Hartung, M., Kolb, L., Gross, A., Rahm, E. Optimizing Similarity Computations for Ontology Matching–Experiences from GOMMA. In *Proceedings of the 9th International Conference on Data Integration in the Life Sciences (DILS)* (2013), Springer, pp. 81–89.

[83] Hartung, M., Terwilliger, J., Rahm, E. Recent advances in schema and ontology evolution. In *Schema Matching and Mapping*. Springer, 2011, ch. 6, pp. 149–190.

[84] Hayamizu, T. F., Mangan, M., Corradi, J. P., Kadin, J. A., Ringwald, M. The Adult Mouse Anatomical Dictionary: a tool for annotating and integrating data. *Genome Biology 6*, 3:R29 (2005).

[85] He, B., Chang, K. C.-C. Automatic complex schema matching across web query interfaces: A correlation mining approach. *ACM Transactions on Database Systems (TODS) 31*, 1 (2006), 346–395.

[86] Hennig, S., Groth, D., Lehrach, H. Automated Gene Ontology annotation for anonymous sequence data. *Nucleic Acids Research 31*, 13 (2003), 3712–3715.

[87] Hu, W., Qu, Y. Falcon-AO: A practical ontology matching system. *Web Semantics: Science, Services and Agents on the World Wide Web 6*, 3 (2008).

[88] Hu, W., Qu, Y., Cheng, G. Matching large ontologies: A divide-and-conquer approach. *Data & Knowledge Engineering 67*, 1 (2008), 140–160.

[89] Huang, D., Sherman, B., Lempicki, R. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research 37*, 1 (2009), 1.

[90] Huang, Y. Stabilitätsanalyse von Ontologie-Mappings in GOMMA. *Diplomarbeit, Leipzig, Universität Leipzig* (2011).

[91] Hubbard, T. J., Aken, B. L., Ayling, S., Ballester, B., Beal, K., Bragin, E., Brent, S., Chen, Y., Clapham, P., Clarke, L., et al. Ensembl 2009. *Nucleic Acids Research 37*, suppl 1 (2009), D690–D697.

[92] Jaccard, P. Distribution de la flore alpine: dans le bassin des dranses et dans quelques régions voisines. *Bulletin de la société vaudoise des sciences naturelles 37* (1901), 241–272.

[93] Jakoniene, V., Lambrix, P. Ontology-based integration for bioinformatics. In *Proceedings of the VLDB Workshop on Ontologies-based techniques for DataBases and InformationSystems (ODBIS)* (2005), pp. 55–58.

[94] Jean-Mary, Y., Shironoshita, E., Kabuka, M. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web 7*, 3 (2009).

[95] Jensen, L. J., Gupta, R., Staerfeldt, H.-H., Brunak, S. Prediction of human protein function according to gene ontology categories. *Bioinformatics 19*, 5 (2003), 635–642.

[96] Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R. Logic-based assessment of the compatibility of UMLS ontology sources. *Journal of Biomedical Semantics 2* (2011).

[97] Jiménez-Ruiz, E., Grau, B. C. LogMap: Logic-Based and Scalable Ontology Matching. In *The Semantic Web–ISWC 2011–10th International Semantic Web Conference* (2011), Springer, pp. 273–288.

[98] Jiménez-Ruiz, E., Grau, B. C., Horrocks, I., Berlanga, R. Ontology integration using mappings: Towards getting the right logical consequences. In *The Semantic Web: Research and Applications, 6th European Semantic Web Conference (ESWC)* (2009), Springer, pp. 173–187.

[99] Jones, C. E., Brown, A. L., Baumann, U. Estimating the annotation error rate of curated go database sequence annotations. *BMC Bioinformatics 8*, 1:170 (2007).

[100] Khattak, A., Pervez, Z., Latif, K., Lee, S. Time efficient reconciliation of mappings in dynamic web ontologies. *Knowledge-Based Systems 35* (2012), 369–374.

[101] Khattak, A. M., Latif, K., Khan, S., Ahmed, N. Managing change history in web ontologies. In *Fourth International Conference on Semantics, Knowledge and Grid (SKG)* (2008), IEEE, pp. 347–350.

[102] Kirsten, T., Gross, A., Hartung, M., Rahm, E. GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. *Journal of Biomedical Semantics 2* (2011).

[103] Kirsten, T., Hartung, M., Gross, A., Rahm, E. Efficient Management of Biomedical Ontology Versions. In *On the Move to Meaningful Internet Systems: OTM Workshops* (2009), Springer, pp. 574–583.

[104] Kirsten, T., Thor, A., Rahm, E. Instance-Based Matching of Large Life Science Ontologies. In *Proceedings of the 4th International Workshop on Data Integration in the Life Sciences (DILS)* (2007), Springer, pp. 172–187.

[105] Klein, M., Fensel, D., Kiryakov, A., Ognyanov, D. Ontology versioning and change detection on the web. *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web* (2002), 197–212.

[106] Klein, M. C. A. Change Management for Distributed Ontologies. *Dissertation, Amsterdam, Vrije Universiteit* (2004).

[107] Kondylakis, H., Plexousakis, D. Ontology Evolution: Assisting Query Migration. In *Proceedings of the 31st International Conference on Conceptual Modeling (ER)* (2012), Springer, pp. 331–344.

[108] Kosiol, C., Vinar, T., Da Fonseca, R., Hubisz, M., Bustamante, C., Nielsen, R., Siepel, A. Patterns of positive selection in six mammalian genomes. *PLOS Genetics 4*, 8:e1000144 (2008).

[109] Koudas, N., Marathe, A., Srivastava, D. Flexible string matching against large databases in practice. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB)* (2004), Morgan Kaufmann, pp. 1078–1086.

[110] Lambrix, P., Edberg, A. Evaluation of ontology merging tools in bioinformatics. In *Pacific Symposium on Biocomputing* (2003), vol. 8, pp. 589–600.

[111] Lambrix, P., Tan, H. SAMBO–A system for aligning and merging biomedical ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web 4*, 3 (2006).

[112] Lambrix, P., Tan, H., Jakoniene, V., Strömbäck, L. Biological ontologies. *Semantic Web* (2007), 85–99.

[113] Langlotz, C. P. Radlex: A new method for indexing online educational materials. *Radiographics 26*, 6 (2006), 1595–1597.

[114] Lassila, O., McGuinness, D. The role of frame-based representation on the semantic web. *Linköping Electronic Articles in Computer and Information*

*Science 6*, 5 (2001).

[115] Leonelli, S., Diehl, A., Christie, K., Harris, M., Lomax, J. How the gene ontology evolves. *BMC Bioinformatics 12*, 1:325 (2011).

[116] Leser, U., Naumann, F. *Informationsintegration–Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen.* dpunkt.verlag, 2006.

[117] Levenshtein, V. I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady 10*, 8 (1966), 707–710.

[118] Lin, D. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning* (1998), pp. 296–304.

[119] Lin, F., Sandkuhl, K. A survey of exploiting wordnet in ontology matching. In *Artificial Intelligence in Theory and Practice II* (2008), Springer, pp. 341–350.

[120] Lipscomb, Carolyn E. Medical subject headings (MeSH). *Bulletin of the Medical Library Association 88*, 3 (2000), 265.

[121] Liu, B., Hsu, W., Ma, Y. Discovering the set of fundamental rule changes. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge discovery and data mining* (2001), ACM, pp. 335–340.

[122] Maarek, Y. S., Berry, D. M., Kaiser, G. E. An information retrieval approach for automatically constructing software libraries. *IEEE Transactions on Software Engineering 17*, 8 (1991), 800–813.

[123] Madhavan, J., Bernstein, P. A., Doan, A., Halevy, A. Corpus-based schema matching. In *Proceedings. 21st International Conference on Data Engineering (ICDE)* (2005), IEEE, pp. 57–68.

[124] Manning, C. D., Schütze, H. *Foundations of Statistical Natural Language Processing.* MIT Press, 2001.

[125] Martins, H., Silva, N. A User-Driven and a Semantic-Based Ontology Mapping Evolution Approach. In *Proceedings of the 11th International Conference on Enterprise Information Systems (ICEIS)* (2009), pp. 214–221.

[126] Mascardi, V., Locoro, A., Rosso, P. Automatic ontology matching via upper ontologies: A systematic evaluation. *Knowledge and Data Engineering, IEEE Transactions on 22*, 5 (2010), 609–623.

[127] McGuinness, D. L., Van Harmelen, F. OWL Web Ontology Language overview. *W3C recommendation 10* (2004).

[128] Meilicke, C. Alignment incoherence in ontology matching. *Dissertation, Mannheim, Universität Mannheim* (2011).

[129] Meilicke, C., Stuckenschmidt, H. Incoherence as a basis for measuring the quality of ontology mappings. In *Proceedings of the 3rd International*

*Workshop on Ontology Matching (OM)* (2008), CEUR-WS.org, pp. 1–12.

[130] Meilicke, C., Stuckenschmidt, H., Tamilin, A. Reasoning support for mapping revision. *Journal of Logic and Computation 19*, 5 (2009), 807–829.

[131] Melnik, S., Bernstein, P. A., Halevy, A., Rahm, E. Supporting Executable Mappings in Model Management. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of data* (2005), pp. 167–178.

[132] Melnik, S., Garcia-Molina, H., Rahm, E. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. In *Proceedings of the 18th International Conference on Data Engineering (ICDE)* (2002), IEEE, pp. 117–128.

[133] Melnik, S., Rahm, E., Bernstein, P. A. Rondo: A Programming Platform for Generic Model Management. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data* (2003), pp. 193–204.

[134] Mork, P., Bernstein, P. A. Adapting a generic match algorithm to align ontologies of human anatomy. In *Proceedings of 20th International Conference on Data Engineering, Boston, USA* (2004), IEEE, pp. 787–790.

[135] Mungall, C. J., Torniai, C., Gkoutos, G. V., Lewis, S. E., Haendel, M. A. Uberon, an integrative multi-species anatomy ontology. *Genome Biology 13*, 1 (2012).

[136] Naumann, F., Leser, U., Freytag, J. C. Quality-driven Integration of Heterogenous Information Systems. In *Proceedings of 25th International Conference on Very Large Data Bases (VLDB)* (1999), Morgan Kaufmann, pp. 447–458.

[137] Ngo, D., Bellahsene, Z. YAM++: a multi-strategy based approach for ontology matching task. In *Knowledge Engineering and Knowledge Management* (2012), Springer, pp. 421–425.

[138] Noy, N. F., Chugh, A., Liu, W., Musen, M. A. A framework for ontology evolution in collaborative environments. In *The Semantic Web–ISWC 2006–5th International Semantic Web Conference* (2006), Springer, pp. 544–558.

[139] Noy, N. F., Klein, M. Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems 6*, 4 (2004), 428–440.

[140] Noy, N. F., Musen, M. A. PromptDiff: A Fixed-Point Algorithm for Comparing Ontology Versions. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI)* (2002), pp. 744–750.

[141] Noy, N. F., Shah, N. H., Whetzel, P. L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D. L., Storey, M.-A., Chute, C. G., et al. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research 37*, suppl 2 (2009), W170–W173.

[142] Pandey, G., Myers, C., Kumar, V. Incorporating functional interrelationships into protein function prediction algorithms. *BMC Bioinformatics 10*, 1:142 (2009).

[143] Papavassiliou, V., Flouris, G., Fundulaki, I., Kotzinos, D., Christophides, V. On Detecting High-Level Changes in RDF/S KBs. In *The Semantic Web–ISWC 2009–8th International Semantic Web Conference*. Springer, 2009, pp. 473–488.

[144] Park, J., Kim, T., Park, J. Monitoring the evolutionary aspect of the Gene Ontology to enhance predictability and usability. *BMC Bioinformatics 9*, Suppl 3:S7 (2008).

[145] Pavel, S., Euzenat, J. Ontology matching: State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering 25*, 1 (2013), 158–176.

[146] Pesquita, C., Couto, F. M. Where GO is Going and What it Means for Ontology Extension. In *Proceedings of the 2nd International Conference on Biomedical Ontology (ICBO)* (2011), CEUR-WS.org, pp. 3–9.

[147] Pesquita, C., Faria, D., Falcão, A., Lord, P., Couto, F. Semantic similarity in biomedical ontologies. *PLoS Computational Biology 5*, 7:e1000443 (2009).

[148] Pesquita, C., Faria, D., Stroe, C., Santos, E., Cruz, I. F., Couto, F. M. What's in a 'nym'? Synonyms in Biomedical Ontology Matching. In *The Semantic Web–ISWC 2013–12th International Semantic Web Conference* (2013), pp. 526–541.

[149] Peukert, E., Berthold, H., Rahm, E. Rewrite Techniques for Performance Optimization of Schema Matching Processes. In *Proceedings of the 13th International Conference on Extending Database Technology (EDBT)* (2010), ACM, pp. 453–464.

[150] Prüfer, K., Muetzel, B., Do, H., Weiss, G., Khaitovich, P., Rahm, E., Pääbo, S., Lachmann, M., Enard, W. FUNC: a package for detecting significant associations between gene sets and ontological annotations. *BMC Bioinformatics 8*, 1:41 (2007).

[151] Rahm, E. *Mehrrechner-Datenbanksysteme–Grundlagen der verteilten und parallelen Datenbankverarbeitung*. Addison-Wesley, 1994.

[152] Rahm, E. Towards Large-Scale Schema and Ontology Matching. In *Schema Matching and Mapping*. Springer, 2011, ch. 1, pp. 3–27.

[153] Rahm, E., Bernstein, P. A. A Survey of Approaches to Automatic Schema Matching. *The VLDB Journal 10*, 4 (2001), 334–350.

[154] Rahm, E., Do, H. H. Data Cleaning: Problems and Current Approaches. *IEEE Data Engineering Bulletin 23*, 4 (2000), 3–13.

[155] Rance, B., Gibrat, J.-F., Froidevaux, C. An Adaptive Combination of Matchers: Application to the Mapping of Biological Ontologies for Genome Annotation. In *Proceedings of the 6th International Workshop on Data Integration in the Life Sciences (DILS)* (2009), Springer, pp. 113–126.

[156] Raunich, S., Rahm, E. Target-driven merging of Taxonomies. *CoRR abs/1012.4855* (2010).

[157] Raunich, S., Rahm, E. ATOM: Automatic target-driven ontology merging. In *Proceedings of the 27th International Conference on Data Engineering (ICDE)* (2011), IEEE, pp. 1276–1279.

[158] Rosse, C., Mejino Jr, J. L. A reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of Biomedical Informatics 36*, 6 (2003), 478–500.

[159] Seddiqui, M. H., Aono, M. An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Web Semantics: Science, Services and Agents on the World Wide Web 7*, 4 (2009), 344–356.

[160] Shvaiko, P., Euzenat, J. A Survey of Schema-Based Matching Approaches. *Journal on Data Semantics IV* (2005), 146–171.

[161] Shvaiko, P., Euzenat, J. Ten challenges for ontology matching. In *On the Move to Meaningful Internet Systems (OTM)* (2008), Springer, pp. 1164–1182.

[162] Sioutos, N., Coronado, S. d., Haber, M. W., Hartel, F. W., Shaiu, W.-L., Wright, L. W. NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics 40*, 1 (2007), 30–43.

[163] Škunca, N., Altenhoff, A., Dessimoz, C. Quality of computationally inferred gene ontology annotations. *PLoS Computational Biology 8*, 5:e1002533 (2012).

[164] Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology 25*, 11 (2007), 1251–1255.

[165] Spiliopoulos, V., Vouros, G. A., Karkaletsis, V. On the discovery of subsumption relations for the alignment of ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web 8*, 1 (2010), 69–88.

[166] Stojanovic, L. Methods and tools for ontology evolution. *Dissertation, Karlsruhe, Universität Karlsruhe* (2004).

[167] Stojanovic, L., Maedche, A., Motik, B., Stojanovic, N. User-driven ontology evolution management. *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW): Ontologies and the Semantic Web* (2002), 285–300.

[168] Su, W., Wang, J., Lochovsky, F. Holistic schema matching for web query interfaces. In *Proceedings of the 10th International Conference on Extending Database Technology (EDBT)* (2006), Springer, pp. 77–94.

[169] Takai-Igarashi, T., Takagi, T. Signal-ontology: Ontology for cell signaling. *Genome Informatics Series* (2000), 440–441.

[170] Tan, H., Jakonienė, V., Lambrix, P., Aberg, J., Shahmehri, N. Alignment of biomedical ontologies using life science literature. In *Knowledge Discovery in Life Science Literature* (2006), Springer, pp. 1–17.

[171] Tenschert, A., Assel, M., Cheptsov, A., Gallizo, G., Della Valle, E., Celino, I. Parallelization and Distribution Techniques for Ontology Matching in Urban Computing Environments. In *Proceedings of the 4th International Workshop on Ontology Matching (OM)* (2009), CEUR-WS.org, pp. 248–249.

[172] Thomas, P. D., Mi, H., Lewis, S. Ontology annotation: mapping genomic regions to biological function. *Current Opinion in Chemical Biology 11*, 1 (2007), 4–11.

[173] Thor, A., Anderson, P., Raschid, L., Navlakha, S., Saha, B., Khuller, S., Zhang, X.-N. Link prediction for annotation graphs using graph summarization. In *The Semantic Web–ISWC 2011–10th International Semantic Web Conference* (2011), Springer, pp. 714–729.

[174] Thor, A., Hartung, M., Gross, A., Kirsten, T., Rahm, E. An Evolution-based Approach for Assessing Ontology Mappings–A Case Study in the Life Sciences. In *Proceedings of 13. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW)* (2009), pp. 277–286.

[175] Thor, A., Kirsten, T., Rahm, E. Instance-based matching of hierarchical ontologies. In *Proceedings of 12. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW)* (2007), pp. 436–448.

[176] Tilford, C., Siemers, N. Gene set enrichment analysis. *Methods in Molecular Biology 563* (2009), 99–121.

[177] Tordai, A., Ghazvinian, A., van Ossenbruggen, J., Musen, M. A., Noy, N. F. Lost in Translation? Empirical Analysis of Mapping Compositions for Large Ontologies. In *Proceedings of the 5th International Workshop on Ontology Matching (OM)* (2010), CEUR-WS.org, pp. 13–24.

[178] Velegrakis, Y., Miller, R. J., Popa, L. Mapping Adaptation under Evolving Schemas. In *Proceedings of 29th International Conference on Very Large Data Bases (VLDB)* (2003), pp. 584–595.

[179] Volz, R., Oberle, D., Staab, S., Motik, B. KAON SERVER–A Semantic Web Management System. In *Proceedings of the Twelfth International World Wide Web Conference (WWW)–Alternate Paper Tracks* (2003).

[180] Wang, J., Du, Z., Payattakool, R., Philip, S., Chen, C. A new method to measure the semantic similarity of GO terms. *Bioinformatics 23*, 10 (2007), 1274–1281.

[181] Wang, P., Xu, B. Debugging ontology mappings: a static approach. *Computing and Informatics 27*, 1 (2012), 21–36.

[182] Whetzel, P. L., Noy, N. F., Shah, N. H., Alexander, P. R., Nyulas, C., Tudorache, T., Musen, M. A. BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic Acids Research 39*, suppl 2 (2011), W541–W545.

[183] Yang, Z., Zhang, D., Ye, C. Ontology Analysis on Complexity and Evolution Based on Conceptual Model. In *Proceedings of the 3rd International Workshop on Data Integration in the Life Sciences (DILS)* (2006), Springer, pp. 216–223.

[184] Yu, C., Popa, L. Semantic Adaptation of Schema Mappings when Schemas Evolve. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)* (2005), ACM, pp. 1006–1017.

[185] Zhang, H., Hu, W., Qu, Y. Constructing virtual documents for ontology matching using mapreduce. In *Proceedings of the Semantic Web–Joint International Semantic Technology Conference*. Springer, 2012, pp. 48–63.

[186] Zhang, S., Bodenreider, O. Aligning Representations of Anatomy using Lexical and Structural Methods. In *AMIA Annual Symposium Proceedings* (2003), American Medical Informatics Association, pp. 753–757.

[187] Zhang, S., Bodenreider, O. Alignment of multiple ontologies of anatomy: Deriving indirect mappings from direct mappings to a reference. In *AMIA Annual Symposium Proceedings* (2005), American Medical Informatics Association, pp. 864–868.

[188] Zhang, S., Bodenreider, O. Experience in Aligning Anatomical Ontologies. *International Journal On Semantic Web and Information Systems 3*, 2 (2007), 1–26.

[189] Zhang, S., Mork, P., Bodenreider, O., Bernstein, P. A. Comparing two approaches for aligning representations of anatomy. *Artificial Intelligence in Medicine 39*, 3 (2007), 227–236.

[190] Zhong, Q., Li, H., Li, J., Xie, G., Tang, J., Zhou, L., Pan, Y. A Gauss Function Based Approach for Unbalanced Ontology Matching. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data* (2009), pp. 669–680.