REx – eine Webapplikation zur Visualisierung der Evolution von Ontologien in den Lebenswissenschaften

Victor Christen
mam08bfa@studserv.uni-leipzig.de

Universität Leipzig,
Master Studiengang Informatik

Abstract: Ontologien sind im Bereich der Lebenswissenschaften für die maschinelle Analyse und Auswertung von Daten unerlässlich. Mithilfe von Ontologien lassen sich molekular-biologische Objekte wie Proteine oder Gene konsistent und einheitlich beschreiben (annotieren), so dass ein maschineninterpretierbarer und semantischer Rahmen für ein Themengebiet spezifiziert werden kann. Durch neue Anforderungen sowie geändertes Domänenwissen unterliegen sie einem ständigen Veränderungsprozess (Evolution). Die Webanwendung REx (Region Evolution Explorer) ermöglicht die Visualisierung von Veränderungen innerhalb einer Ontologie. Dabei werden für veröffentlichte Versionen innerhalb eines Zeitraums Regionen mit hoher (geringer) Änderungsintensität ermittelt. Nutzer können sich diese Regionen über verschiedene Workflows visualisieren lassen, um somit die Evolution insbesondere von großen Ontologien kompakter und besser zu verstehen. REx wurde anhand zahlreicher, großer Ontologien in den Lebenswissenschaften evaluiert und ist unter `http://dbs.uni-leipzig.de/rex` verfügbar.

1 Einleitung

Aufgrund des exponentiellen Anstiegs des Datenvolumens im Bereich der Informationsverarbeitung ist eine semantische Strukturierung der Daten unerlässlich, um eine eindeutige Interpretation zu gewährleisten. Mithilfe von Ontologien ist eine flexible semantische Anreicherung der Daten möglich. Eine Ontologie ist nach Gruber [Gru95] eine gemeinsame explizite Spezifikation einer Konzeptualisierung. Durch den Einsatz einer Ontologie ist ein einheitliches Format der annotierten Daten gewährleistet, so dass verschiedene Institutionen oder Anwendungen diese verwenden und interpretieren können. Insbesondere in den Lebenswissenschaften spielen Ontologien eine essentielle Rolle [BS06]. Sie werden primär für die einheitliche Annotation molekular-biologischer Objekte verwendet. So werden bspw. Proteine in der UniProt Wissensbasis [BAW+05] mithilfe von Konzepten aus der Gene Ontology (GO) [Gen08] annotiert, um u.a. deren molekulare Funktionen oder Beteiligung an biologischen Prozessen zu spezifizieren. Solche Wissensbasen können für weitere Analysen verwendet werden, z.B. für die Vorhersage von Proteinstrukturen [Ste03] oder für funktionelle Genexpressionsanalysen [HSL09].

Jede Ontologie repräsentiert eine Domäne, die einen Realweltausschnitt darstellt. Speziell

in den Lebenswissenschaften verändern sich die Ontologien in regelmäßigen Abständen, um neue/geänderte Forschungserkenntnisse in einer Domäne in die entsprechende Ontologie zu integrieren [HKR08]. Des Weiteren müssen Ontologien bspw. verändert werden, um Designfehler aus früheren Versionen zu beseitigen. Infolgedessen werden ständig neue Versionen einer Ontologie veröffentlicht, welche dann durch Endanwender verwendet werden können. Die Veröffentlichung neuer Versionen erfordert die Identifikation der Änderungen der neuen Version bzgl. der Älteren, um festzustellen, ob ein Nutzer aufgrund der durchgeführten Änderungen betroffen ist und ggf. Daten anpassen oder Analysen wiederholen muss.

Da Ontologien in den Lebenswissenschaften oft sehr groß und komplex sind (z.B. umfasst die GO mehr als 30.000 Konzepte), können mögliche Auswirkungen häufig nicht direkt oder nur mit viel manuellem Aufwand ermittelt werden. Derartiges Wissen könnte jedoch gewinnbringend eingesetzt werden. Hat sich bspw. ein spezieller Teil einer Ontologie nicht verändert (d.h. er ist stabil geblieben) wären Nutzer/Anwendungen die diesen Teil verwenden von Änderungen nicht betroffen. Andersherum müssten Datenquellen oder Applikationen, welche einen stark veränderten (instabilen) Ontologieteil verwenden ggf. angepasst oder migriert werden. Gleichermaßen könnten sich Entwickler/Koordinatoren in einem Ontologieprojekt darüber informieren, welche Ontologieteile in den letzten Jahren stark oder eher marginal verändert wurden, um daraus weitere Entwicklungsschritte für künftige Arbeiten abzuleiten und zu planen. Ein entsprechendes Tool für derartige Analysen müsste einerseits die Größe der zu analysierenden Ontologien beachten und sollte andererseits möglichst intuitive und kompakte Darstellungsformen/Visualisierungen zur verständlichen Erfassung der Ontologieevolution anbieten.

Die vorgestellte Webapplikation REx (Region Evolution Explorer – `http://dbs.uni-leipzig.de/rex`) visualisiert die Intensität der Veränderungen einer Ontologie über einen bestimmten Zeitraum mittels Ontologieregionen in struktureller und statistischer Form. Es werden die folgenden Beiträge geleistet:

- Konzeption und Implementierung der Webanwendung auf Basis eines Repository für Ontologieversionen und eines Algorithmus zur Erkennung von stabilen bzw. instabilen Ontologieregionen
- Bereitstellung von zwei Workflows zur (1) graphbasierten sowie (2) statistischen Analyse der Ontologieevolution mittels Ontologieregionen
- Evaluierung und Test der Webanwendung für zahlreiche große Ontologien aus den Lebenswissenschaften

Der Beitrag ist wie folgt gegliedert. Im nächsten Abschnitt werden Grundlagen vermittelt und das in der Webapplikation verwendete Verfahren für die Analyse der Evolution von Ontologien vorgestellt. Der dritte Abschnitt befasst sich mit der Webapplikation REx. Dazu werden die beiden Workflows näher vorgestellt und Informationen zur Architektur präsentiert. Eine Evaluierung von REx findet im vierten Abschnitt statt. Im fünften Abschnitt werden verwandte Arbeiten diskutiert. Eine Zusammenfassung sowie ein Ausblick bilden den Abschluss des Beitrags.

2 Grundlagen – Ontologie, Evolution und Regionen

2.1 Ontologie und Ontologieversionen

Formal wird eine Ontologie O durch eine Menge von Konzepten C, Relationen R und Attributen A beschrieben: $O = (C, R, A)$. Konzepte besitzen einen eindeutigen Identifizierer (z.B. URI oder accession number). Relationen erweitern die Semantik einer Ontologie, indem Konzepte miteinander in Beziehung gesetzt werden können. Die *is_a* Relation, die eine Subklassen Beziehung zwischen zwei Konzepten definiert, ist die essentiellste Relation. Neben dieser Relation existieren andere Relationen wie z.B. *part_of* oder domänenspezifische wie z.B. *regulates* . Attribute wie Name, Synonym oder Definition werden genutzt, um ein Konzept näher zu beschrieben.

Eine Ontologie ist keine statische Spezifikation. In regelmäßigen Abständen werden Änderungen vorgenommen, um neue Erkenntnisse oder Fehler einzubeziehen bzw. zu beseitigen. Für eine Analyse der Veränderungen einer Ontologie ist eine Versionierung unerlässlich. Eine Version v einer Ontologie O ist eine Momentaufnahme zu einem bestimmten Zeitpunkt und wird als O_v bezeichnet. Eine Version ist solange gültig bis eine neue Version veröffentlicht wird. Die Zeitpunkte von Versionen stellen somit eine Ordnungsrelation dar, d.h. jede Version (mit Ausnahme der Ersten bzw. Letzten) besitzt exakt eine Vorgänger- bzw. Nachfolgeversion.

2.2 Regionenalgorithmus

Eine Ontologieregion OR ist ein Teilgraph der Ontologie, der ein Wurzelkonzept und dessen Nachfolgekonzepte beinhaltet, die durch eine *is_a* Relation verknüpft sind. Im Folgenden wird kurz erläutert wie (in)stabile Regionen zwischen zwei Ontologieversionen O_v und O_{v+1} berechnet werden können. Für eine ausführliche Darstellung inkl. Beispiel wird auf [HGKR10] verwiesen.

Zunächst müssen die Änderungen zwischen O_v und O_{v+1} mittels eines Diff-Algorithmus identifiziert werden. Der Diff von zwei Versionen O_v und O_{v+1} beschreibt die Menge von Änderungsoperationen, welche angewandt auf O_v die Version O_{v+1} erzeugen würden. Es wird zwischen einfachen Änderungen (z.B. Einfügen eines Konzepts) und komplexen Änderungen (z.B. Merge mehrerer Konzepte in ein Konzept) unterschieden. Für die Berechnung des Diffs existieren diverse Verfahren wie z.B. PromptDiff [NM02] oder COnto-Diff [HGR12]. Basierend auf dem Diff zwischen beiden Versionen wird die Intensität der Veränderung mithilfe des Regionenalgorithmus ermittelt. Für die Berechnung der Intensität wird ein Kostenmodell verwendet, das jeder Änderungsoperation Kosten zuweist, z.B. Kosten zwei für das Löschen eines Konzepts oder eins für das Hinzufügen einer Relation.

Zu Beginn werden sowohl in O_v als auch in O_{v+1} den gelöschten Konzepten bzw. hinzugefügten Konzepten die entsprechenden Kosten zugewiesen. Die Kosten für veränderte Relationen werden den beteiligten Konzepten zugewiesen. Diese Kosten entsprechen einer direkten Veränderung eines Konzepts und werden als lokale Kosten *lc* bezeichnet. Des

Weiteren wird ein Konzept durch die Konzepte innerhalb seiner Region beeinflusst. Diese Art des Einflusses wird durch die aggregierten Kosten ac repräsentiert. Die aggregierten Kosten *ac* eines Konzepts *c* berechnen sich aus den aggregierten Kosten seiner Kinderkonzepte sowie den eigenen lokalen Kosten lc. Die aggregierten Kosten eines Kinderkonzepts c' werden anteilig an dessen Elternkonzepte ($parents(c')$) propagiert:

$$ac(c) = \sum_{c' \in children(c)} \frac{ac(c')}{|parents(c')|} + lc(c)$$

Die Berechnung der aggregierten Kosten aller Konzepte in den beiden Ontologieversionen O_v und O_{v+1} beginnt jeweils bei den Blattkonzepten und wird entlang der is_a Struktur bis zu den Wurzelkonzepten fortgesetzt. Um den Einfluss der Löschung von Elementen ebenfalls in der Version O_{v+1} nachvollziehen zu können, müssen die Kosten aus O_v nach O_{v+1} transferiert werden. Für jedes Konzept, welches sowohl in O_v als auch in O_{v+1} existiert, werden die aggregierten Kosten aus beiden Versionen zusammengefasst.

Ein signifikantes Maß für die Veränderung (Stabilität) einer Region OR sind die durchschnittlichen Kosten *avg_costs*. Diese Kosten relativieren die aggregierten Kosten der Wurzel ($root$) von OR bezüglich der Größe der Region: $avg\_costs = \frac{ac(root)}{|OR|}$. Regionen mit hohen *avg_costs* weisen somit starke Veränderungen auf und können als instabil klassifiziert werden. Für *avg_costs*=0 liegen keine Änderungen in einer Region vor.

Die Analyse für einen Zeitraum, der mehrere Versionen beinhaltet, basiert auf der iterativen Ausführung der beschriebenen Methode. In jeder Iteration mit Ausnahme der ersten wird die neue Version der vorherigen Iteration als die alte angesehen und mit der darauf folgenden Version die Berechnung durchgeführt. Der Algorithmus endet, wenn die Berechnungen für die letzte Version im Zeitraum beendet sind. Somit werden sukzessive die aggregierten Kosten aus allen Versionen aufgesammelt und in der letzten Version kann eine Regionenerkennung stattfinden.

3 REx – Region Evolution Explorer

REx ist eine Webapplikation, die mithilfe des Regionenalgorithmus die Intensität der Evolution von Ontologien visualisiert. Die Visualisierung gliedert sich in die strukturelle Repräsentation der Ontologie in Form eines Graphen und in die statistische Repräsentation. Aufgrund der immensen Größe des resultierenden Graphen ist eine ausschließliche statische Repräsentation ungeeignet, da der Anwender nicht in der Lage ist innerhalb der Ontologie zu navigieren. Das prinzipielle Designprinzip ist das *„Information Seeking Mantra“* [Kei02], das den Anwender in den Explorationsprozess integriert. REx ist mithilfe des GWT (Google Web Toolkit) und den erweiternden Frameworks SmartGWT[1] und ExtGWT[2] sowie den Graphbibliotheken Flare[3] und InfoVis[4] implementiert. Im Folgenden

[1] http://code.google.com/p/smartgwt/
[2] http://www.sencha.com/products/extgwt
[3] http://flare.prefuse.org/
[4] http://thejit.org/

werden die zwei Hauptworkflows näher erläutert und die Architektur beschrieben.

3.1 Funktionalitäten

Generell gliedert sich jede Funktionalität in die Spezifikation der notwendigen Parameter, die Berechnung der spezifischen Größen des Regionenalgorithmus und die graphische Präsentation.

3.1.1 Graphrepräsentation

Der Anwender ist in der Lage sich einen Überblick über die gesamte Struktur inklusive der Analyseergebnisse zu verschaffen und Subregionen eines Konzepts in einem weiteren Graph näher zu analysieren. Neben der graphischen Repräsentation können die Analyseergebnisse in tabellarischer Form betrachtet werden. Der prinzipielle Ablauf der Applikation für die Repräsentation ist in Abb. 1 dargestellt.

Zu Beginn der Analyse werden die Eingabeparameter im *Input Panel* spezifiziert, die aus dem Namen und dem Typ der Ontologie sowie dem zu analysierenden Zeitraum bestehen. Nach Beendigung der Berechnungen wird die gesamte Ontologie als Graph im *Overview Panel* dargestellt. Basierend auf den berechneten Durchschnittskosten *avg_costs* werden

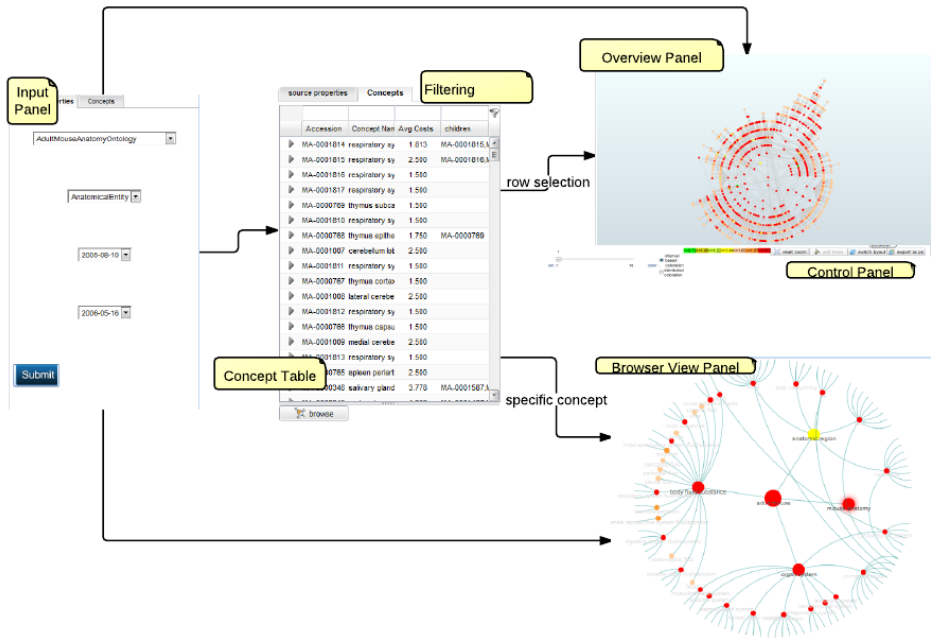


Abbildung 1: Graph-Workflow

die Konzepte mittels einer Grün-Gelb-Rot-Farbskala entsprechend eingefärbt. Für die visuelle Exploration ist der Anwender in der Lage mit dem geometrischen Zoom bzw. mit dem Fisheye-Zoom Teilgebiete detaillierter zu betrachten. Im *Control Panel* ist der geometrische Zoom einstellbar. Durch Auswahl eines Konzepts wird der Graph, von diesem Knoten aus, nach dem Fisheye-Zoom Prinzip [SB94] verzerrt.

Bei einer hohen strukturellen Komplexität einer Ontologie ist es möglich, dass der Graph nicht exakt visuell zu erfassen ist, deshalb ist im *Browser View Panel* eine zweite Ansicht der Ontologie verfügbar, dessen Komplexität durch einen Tiefenfilter der Größe 3 minimiert ist. Die Farbgebung der Konzepte, entspricht der selbigen wie im *Overview Panel*. Dieser Graph bietet eine dynamische Navigation durch die Ontologie, indem bei einem gewählten Konzept ein neuer Graph mit Tiefe 3 und diesem Konzept als Wurzel animiert erstellt wird. Die Elternkonzepte werden ebenfalls angezeigt, um innerhalb der Ontologie navigieren zu können.

Mithilfe der *Concept Table* können Informationen, wie z.B. der Name, die Kosten oder die Kinder eines Konzepts der Ontologie tabellarisch betrachtet werden. Die Tabelle ist sortierbar und darzustellende Informationen können über eine Filteroption eingeschränkt werden. Die Selektion eines Konzepts in der Tabelle wird im Graph durch Hervorhebung des Knotens gekennzeichnet, sodass die textuelle Information intuitiv mit dem Graph der Ontologie verknüpft werden kann. Des Weiteren kann ein selektiertes Konzept als Wurzelkonzept des Graphen im *Browser View Panel* verwendet werden.

3.1.2 Statistikrepräsentation

Die Evolution von Ontologien kann ebenfalls mithilfe von Diagrammen analysiert werden. REx bietet dafür quantitative sowie Sliding-Window Statistiken an. Die Grundlage der quantitativen Statistiken ist die Anzahl der Änderungsoperationen (Diff) zwischen aufeinanderfolgenden Versionen einer Ontologie. Des Weiteren besteht die Möglichkeit in der Sliding-Window Statistik die Evolution anhand des Verlaufs der durchschnittlichen Kosten für eine Region über einen gewählten Zeitraum mithilfe des Sliding-Window Verfahren zu veranschaulichen. Der prinzipielle Ablauf ist in Abb. 2 dargestellt.

Der Anwender hat bei der Visualisierung einer quantitativen Statistik die Wahl zwischen dem Vergleich von zwei Ontologien über einen Zeitraum und dem Vergleich von einer Ontologie über zwei Zeiträume. Je nach gewählter Option müssen die Namen der Ontologien bzw. Zeiträume im vorgesehenen Eingabepanel spezifiziert werden. Das resultierende Diagramm beinhaltet zwei Graphen, je nach gewählter Option entsprechen diese zwei Ontologien bzw. zwei Zeiträumen. Die Koordinaten eines Graphen sind durch die Zeit und die Anzahl der Änderungsoperationen gegeben. Mithilfe der generierten Diagramme können sich Nutzer zunächst einen ersten Überblick über die Evolution verschaffen, um anschließend auf Basis dessen ausgewählte Zeiträume/Regionen näher zu inspizieren.

Für die zweite Statistik wird ein Sliding-Window Verfahren verwendet. Der Anwender spezifiziert die Ontologie, den Zeitraum und die Konzepte, deren Analyseergebnisse visualisiert werden sollen. Darüber hinaus müssen für das Verfahren spezifische Parameter wie Fenstergröße und Schrittweite angegeben werden. Das Sliding-Window Verfahren be-

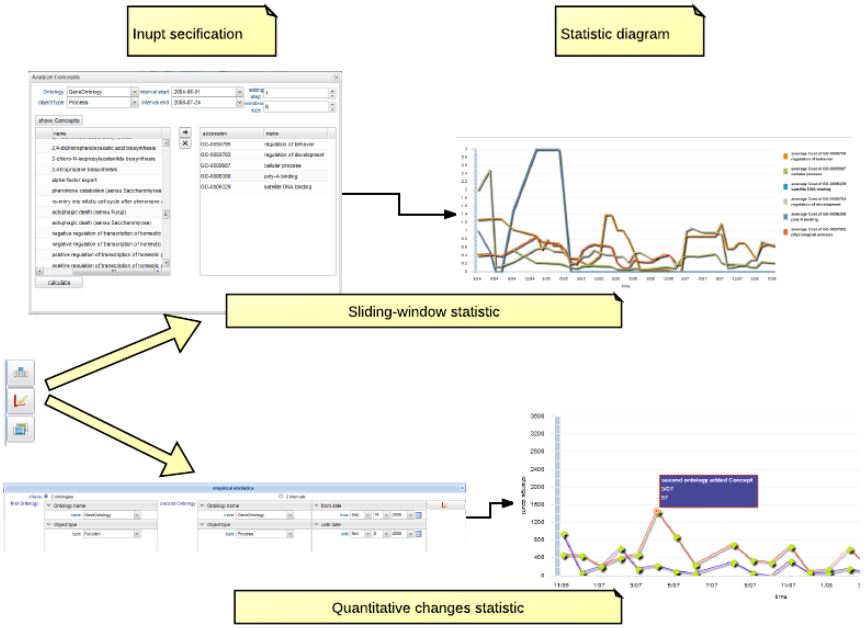


Abbildung 2: Statistik-Workflow

rechnet schrittweise die durchschnittlichen Kosten innerhalb des Fensters für den gesamten Zeitraum. Das Fensterende wird zu Beginn mit dem spezifizierten Startzeitpunkt des gewählten Gesamtzeitraums gleichgesetzt. Nun werden mithilfe des Regionenalgorithmus die durchschnittlichen Kosten für jedes Konzept für den fensterspezifischen Zeitraum ermittelt. Die ermittelten Kosten werden mit dem Endzeitpunkt des Fensters gespeichert. Danach wird das Fenster um die spezifizierte Schrittweite in die Zukunft geschoben und das Verfahren berechnet erneut die Kosten für die neue Position des Fensters. Das Verfahren endet, wenn das Fenster das Ende des Analysezeitraums erreicht hat. Die Koordinaten des resultierenden Graphen repräsentieren die entstandenen Kosten in Abhängigkeit vom Zeitpunkt.

3.2 Architektur

Aufgrund des hohen Grades der Interaktion, die durch die Menge an Kontroll- und Anzeigekomponenten geboten wird, zählt die Applikation zu den Rich Internet Applications. Prinzipiell wurden die Pakete bzw. Klassen sowohl auf Clientseite als auch auf Serverseite nach den einzelnen Workflows modular realisiert. Der Aufbau der Applikation gliedert sich in drei große Subkomponenten. Die Client-Komponente beinhaltet alle Klassen, die die Clientfunktionalitäten realisieren, d.h. die Implementierung der Weboberfläche und das Event-Handling für die Interaktion. Generell ist die Oberfläche mit allen Kontrollelemen-

ten mithilfe des GWT, ExtGWT und SmartGWT implementiert. Die einzelnen Graphen wurden in ActionScript basierend auf der Flare Graphbibliothek sowie in JavaScript mithilfe der InfoVis Bibliothek realisiert und in die GWT-Umgebung eingebunden. Die Flare Bibliothek basiert auf dem IVR Modell (Information Visualization Reference Model), das eine Dekomposition des Visualisierungsprozesses definiert. Die Teilprozesse beinhalten die Datenspeicherung, das Mappen der Rohdaten auf Objekte visueller Form und die Speicherung der Informationen für das Aussehen der Objekte sowie das Rendering. Diese Bibliothek bietet eine hohe Diversität an Möglichkeiten für die Gestaltung der Visualisierung, sodass sich der Fisheye-Zoom oder der geometrische Zoom des Gesamtgraphen realisieren ließen. Das Layout des Übersichtsgraphen ist ein Radial Hierarchical Layout [BK01]. Dieses Layout ordnet die Knoten radial an, wobei mit zunehmender Tiefe bzgl. des Graphen die Entfernung der Knoten vom Zentrum zu nimmt. Die Präsentation der Ontologie im *Browser View Panel* ist durch einen Hypergraph realisiert [HL99]. Mithilfe dieser Darstellung wird der Fokus auf das Konzept im Zentrum konzentriert,welches der Anwender spezifiziert hat. Die Kommunikation der eingebetteten Komponenten und der GWT-Umgebung basiert auf JSNI (JavaScript Native Interface) Methoden, die es erlauben direkt JavaScript zu implementieren, um somit auf die einzelnen Komponenten im DOM-Baum zu zugreifen.

Die serverseitige Implementierung realisiert die Datenanfragebehandlung für die einzelnen Workflows und umfasst für die Berechnung der Regionen den Regionenalgorithmus. Diese Berechnungen werden sowohl für die Sliding-Window Statistik als auch für die zu visualisierenden Daten benötigt. Um dem Anwender eine große Flexibilität bzgl. der Parameterspezifikation zu gewähren, erfolgt die Berechnung der Regionen ad hoc. Eine Materialisierung der Analyseergebnisse von potenziellen Konfigurationen hätte den Vorteil performante Anfragen zu gewährleisten, würde aber entweder einen immensen Speicheraufwand beanspruchen oder eine Einschränkung der Eingabeparameter bedeuten. Des Weiteren werden für den Regionenalgorithmus, die quantitativen Statistiken und die anzuzeigenden Informationen SQL-Queries bereit gestellt, so dass die Ontologieversionen oder andere relevante Daten von der Datenbank extrahiert werden können. Eine Datenbank zur Verwaltung von Ontologieversionen stellt das Backend der Applikation dar.

4 Evaluation

Um die Anwendbarkeit der Applikation zu testen, erfolgte eine Evaluation anhand unterschiedlich großer Ontologien aus OnEX [HKGR09]. Beispielhaft werden nachfolgend Ergebnisse für die Teilontologie Molekulare Funktionen der GO von 2007 bis 2009 präsentiert, welche 9214 Konzepte und 10604 Relationen enthält.

Das Ergebnis der strukturellen Repräsentation ist in Abb. 3 ersichtlich. Aufgrund der Rotverteilung(schwarz) ist intuitiv die Instabilität einiger Regionen der Ontologie erkennbar. Konzepte, die rot eingefärbt sind, weisen überdurchschnittliche *avg_costs* über 1,5 auf. Im Folgenden erfolgt eine nähere Analyse der Kinderkonzepte des Wurzelkonzepts. Auffällig ist der grün(hellgrau) gefärbte Sektor im rechten oberen Teil des Graphen. Ein Großteil dieses Sektors wird durch die Region *binding* gebildet. Die *avg_costs* betragen ledig-

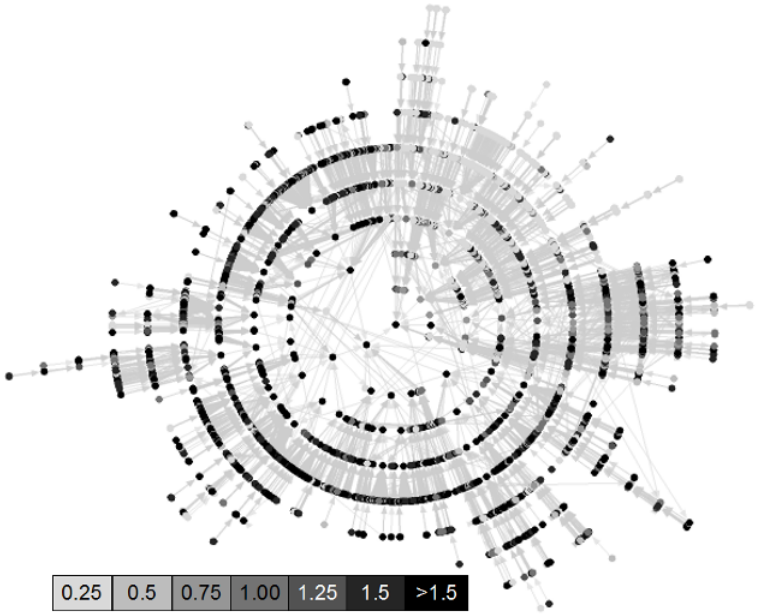
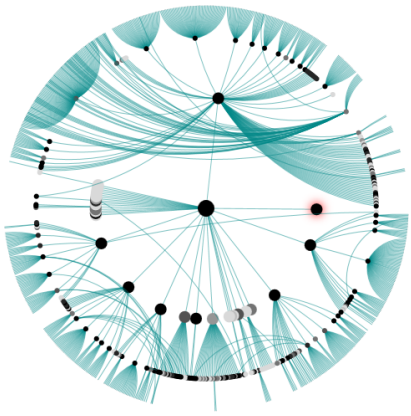


Abbildung 3: Änderungsintensitäten in GO Molekulare Funktionen zwischen 2007 und 2009

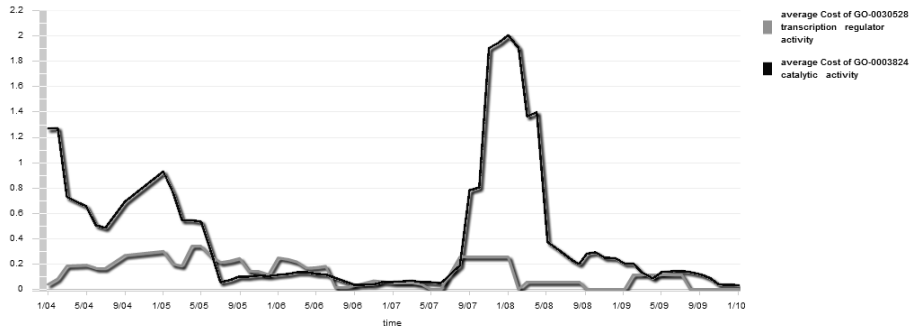
lich 0,62. Diese geringen Kosten resultieren aus den geringen *avg_costs* der Konzepte die in dieser Region liegen (siehe Abb. 4(a)). Die *avg_costs* dieser Konzepte betragen weniger als 0,25. Aufgrund der niedrigen Kosten ist die Region des Konzepts *binding* als stabil zu erachten. Ebenfalls eine stabile Region ist *transcription regulator activity* mit *avg_costs*<0,5. Im Gegensatz zu den stabilen Regionen existieren für diesen Zeitraum auch instabile Regionen, z.B. *catalytic activity* (siehe Abb. 4(b)). Insbesondere in den Teilgebieten *DNA polymerase activity, peptidase activity* sowie *electron carrier activity* fanden mit *avg_costs*>11 immense Änderungen statt, welche zur Instabilität der gesamten Region beitragen.

Um festzustellen, ob die Regionen *transcription regulator activity* und *catalytic activity* einer ständigen Veränderung unterlagen oder teilweise stabil waren, soll im Folgenden eine Sliding-Window Analyse über einen längeren Zeitraum durchgeführt werden. Dazu wurden die folgenden Parameter verwendet: der Zeitraum umfasst 2004–2010 bei einer Fenstergröße von sechs Versionen und einer Schrittweite von einer Version. Die Ergebnisse sind in Abb. 4(c) dargestellt. Die Region *catalytic activity* (dunkle Linie) weist häufig Instabilitäten auf, z.B. erfolgten von Juni 2007 bis Januar 2008 zahlreiche Änderungen was sich in einem Anstieg der *avg_costs* von 0,05 auf 2 widerspiegelt. Seither sind innerhalb der Region weniger Modifikationen, d.h. eine ansteigende Stabilität, zu beobachten (Abfall der *avg_costs*). Die Region *transcription regulator activity* weist aufgrund der geringen *avg_costs* von höchstens 0,3 über den gesamten Zeitraum eine eher geringe Änderungsintensität auf. In den letzten Jahren stiegen die *avg_costs* nie über 0,1, was vermuten lässt, dass die Ontologieentwicklung in diesem Bereich nahezu abgeschlossen ist.

(a) Subregion des Konzepts *binding*



(b) Subregion des Konzepts *catalytic activity*



(c) Sliding-Window Analyse zwischen 2004–2010 für *transcription regulator activity* und *catalytic activity*

Abbildung 4: Detailanalysen für GO Molekulare Funktionen

5 Verwandte Arbeiten

Verwandte Arbeiten gliedern sich in Arbeiten aus dem Bereich der Ontologieevolution (siehe [FMK+08] für Übersichtsartikel) und web-basierten Systemen, welche einen Zugang zu Ontologien erlauben und Inhalte entsprechend visualisieren können.

In den Lebenswissenschaften haben sich mit BioPortal [NSW+09] sowie der OBO Foundry [SAR+07] zwei Plattformen entwickelt, welche die Verwaltung zahlreicher biomedizinischer Ontologien verfolgen. Es werden primär Ontologien sowie zugehörige Versionen zum Download angeboten, so dass Anbieter ihre Ontologien (inkl. Versionen) über diese Plattformen zentral veröffentlichen können. Für einige Ontologien existieren ebenfalls spezielle Webapplikationen. AmiGO[5] ist ein Beispiel zur Visualisierung der GO. Durch einen Tree-Browser ist es möglich die Abhängigkeiten bzw. Zusammengehörigkeit von

[5] http://amigo.geneontology.org/cgi-bin/amigo/go.cgi

Konzepten durch das interaktive Selektieren zu identifizieren. Des Weiteren lassen sich durch Selektion eines Konzepts weitere Eigenschaften, wie z.B. Name und Beschreibung anzeigen. Der Tree-Browser ist als Single-Rooted-Tree realisiert, so dass die Problematik bezüglich der Darstellung der Polymorphie existiert. In einer Graphansicht können Graphen in Form von Bildern aus Konzepten und Relationen im Tree-Browser generiert werden. Diese Visualisierungsform bietet dadurch keine weiteren Möglichkeiten der Interaktion, so dass eine weitere Exploration nur über den Tree-Browser möglich ist. Die Visualisierung basiert immer auf der neuesten GO Version.

OnEX [HKGR09] ist eine Webapplikation, die auf die Analyse der Evolution von Ontologien in den Lebenswissenschaften spezialisiert ist. Die Präsentation gliedert sich in die Wiedergabe von Evolutionstrends mittels quantitativer Statistiken und Tabellen sowie der Evolution von Konzepten innerhalb einer Ontologie. Die quantitativen Analysen beruhen auf der Anzahl von Änderungsoperationen, die zwischen Versionen einer Ontologie aufgetreten sind. Es besteht die Möglichkeit eines Browsing durch die Historie der Versionen. Dadurch ist es möglich sich sowohl die Veränderung einer Ontologie im Ganzen als auch die Veränderung einzelner Konzepte zu betrachten. Eine baumartige Navigation durch die Inhalte einer Ontologie wird nicht angeboten.

Im Gegensatz zu vorherigen Arbeiten fokussiert diese Arbeit auf die visuelle Präsentation der Evolution von großen Ontologien. Mithilfe eines speziellen Regionenalgorithmus kann die Intensität der Veränderungen für jede Region einer Ontologie berechnet und dynamisch visualisiert werden. So können Nutzer intuitiv stark bzw. schwach veränderte Teile innerhalb einer Ontologie ausmachen und ggf. auch im Detail inspizieren.

6 Zusammenfassung und Ausblick

Diese Arbeit befasste sich mit der Thematik der Datenvisualisierung in Bezug auf die Evolution von Ontologien der Lebenswissenschaften. Die Untersuchung der Evolution von Ontologien ist notwendig, um Forschungstrends zu identifizieren, die Intensität der Veränderungen von Ontologien zu messen und Entwicklungen von Ontologien zu planen bzw. zu überwachen. Aufgrund der hohen Datenmenge ist eine textuelle Repräsentation ungeeignet. Diesbezüglich wurde die Webanwendung REx entwickelt, welche auf Basis eines Regionenalgorithmus Veränderungen innerhalb einer Ontologie aggregiert und visualisiert. REx bietet Anwendern diverse Workflows, um Ergebnisse des Regionenalgorithmus als Graph oder Statistik zu präsentieren.

Derzeit arbeitet REx auf Basis von Ontologieversionen aus dem OnEX Repository. Künftig wäre es wünschenswert beliebige Ontologien mit REx analysieren zu können. Ein Anwender sollte z.B. in der Lage sein eigene Ontologieversionen in REx zu laden, um anschließend für diese Ontologie eine Regionenanalyse durchführen zu können.

Danksagung. Die vorliegende Arbeit entstand auf der Basis meiner Bachelorarbeit. Mein Dank gilt meinen Betreuern Anika Groß und Dr. Michael Hartung, die mir stets helfende Ratschläge und Verbesserungsvorschläge geboten haben.

Literatur

[BAW+05] A. Bairoch, R. Apweiler, C.H. Wu et al. The universal protein resource (UniProt). *Nucleic acids research*, 33(suppl 1), 2005.

[BK01] G. Book und N. Keshary. Radial Tree Graph Drawing Algorithm for Representing Large Hierarchies. *University of Connecticut*, 2001.

[BS06] O. Bodenreider und R. Stevens. Bio-ontologies: current trends and future directions. *Briefings in Bioinformatics*, 7(3), 2006.

[FMK+08] G. Flouris, D. Manakanatas, H. Kondylakis et al. Ontology change: classification and survey. *The Knowledge Engineering Review*, 23(2), 2008.

[Gen08] Gene Ontology Consortium. The Gene Ontology project in 2008. *Nucleic Acids Research*, 36(Database Issue), 2008.

[Gru95] T.R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5), 1995.

[HGKR10] M. Hartung, A. Groß, T. Kirsten und E. Rahm. Discovering Evolving Regions in Life Science Ontologies. In *Data Integration in the Life Sciences (DILS)*, 2010.

[HGR12] M. Hartung, A. Groß und E. Rahm. COnto–Diff: Generation of Complex Evolution Mappings for Life Science Ontologies. *Journal of Biomedical Informatics*, 2012.

[HKGR09] M. Hartung, T. Kirsten, A. Groß und E. Rahm. OnEX: Exploring changes in life science ontologies. *BMC Bioinformatics*, 10(1), 2009.

[HKR08] M. Hartung, T. Kirsten und E. Rahm. Analyzing the evolution of life science ontologies and mappings. In *Data Integration in the Life Sciences (DILS)*, 2008.

[HL99] R. Haenni und N. Lehmann. Efficient hypertree construction. Bericht, Institute of Informatics, University of Fribourg, 1999.

[HSL09] D.W. Huang, B.T. Sherman und R.A. Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research*, 37(1), 2009.

[Kei02] D.A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1), 2002.

[NM02] N.F. Noy und M.A. Musen. PROMPTDIFF: A Fixed-Point Algorithm for Comparing Ontology Versions. In *Proc. AAAI/IAAI*, 2002.

[NSW+09] N.F. Noy, N.H. Shah, P.L. Whetzel et al. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic acids research*, 37(suppl 2), 2009.

[SAR+07] B. Smith, M. Ashburner, C. Rosse et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11), 2007.

[SB94] M. Sarkar und M.H. Brown. Graphical fisheye views. *Communications of the ACM*, 37(12), 1994.

[Ste03] G. Steger. *Bioinformatik: Methoden zur Vorhersage von RNA-und Proteinstrukturen*. Birkhauser, 2003.