

**Seminar “Visualisierung in der
Bioinformatik”–
Klukas und Schreiber:
*Dynamic exploration and editing of kegg
pathway diagrams***

Dennis Ruether*

Uni Dortmund:

FB-Informatik: Lehrstuhl für Algorithm Engineering

12. Juli 2007

*dennis_ruether@web.de

Inhaltsverzeichnis

1	Einleitung	3
2	Darstellungen in der Biologie	4
2.1	Traditionelle Darstellung und semistatische Darstellung	4
2.2	Dynamische Darstellung	5
3	KEGG, KGML und KGML Editor	7
3.1	KEGG	7
3.2	KGML	8
3.3	KGML-ED	8
4	Dynamische Entwicklung	10
4.1	Ausdehnung der Übersicht	10
4.2	Sequentielle Erweiterung von Signalpfaden	11
4.3	Signalpfade arrangieren	14
4.4	Signalpfade kollabieren	14
5	Zusammenfassung	15
6	Quellen	16

1 Einleitung

Die vorliegende Arbeit beschäftigt sich mit dem Paper “*Dynamic exploration and editing of kegg pathway diagrams*” von Klukas und Schreiber. Es ist die Ausarbeitung zu dem gleichnamigen Vortrag und fasst die von Klukas und Schreiber entwickelten dynamischen Methoden zur Erstellung und zur Navigation von *KEGG-Signalwegdiagrammen*, zusammen. Die Begriffe, wie *KEGG* (Koyoto Encyclopedia of Genes and Genomes), *KGML* (KEGG Markup Language) und dem KGML-Editor *KGML-ED* werden analog zum Paper, als Grundlage zum besseren Verständnis, erläutert.

Christian Klukas and Falk Schreiber haben im Dezember 2006 das Paper “*Dynamic exploration and editing of kegg pathway diagrams*” veröffentlicht. In diesem Paper stellen sie Methoden vor, mit denen *KEGG-Signalwegdiagramme* auf besonderem Niveau erstellt und editiert werden können. Diese Methoden sind in einem von ihnen entwickelten Editor, dem *KGML-ED*, implementiert. Wie die Begriffe *KEGG*, *KGML* und *KGML-Editor* bereits suggerieren, sind die Graphen an das von der biologischen Datenbank *KEGG* geprägte Format *KGML* gebunden. Die Ausarbeitung erläutert grundlegend im ersten Teil (Kapitel 2) allgemeine Sichtweisen zur Darstellung in der Biologie und erörtert den Übergang von der traditionellen bis hin zur dynamischen Darstellung. Das Kapitel 3 schafft die Voraussetzungen zum Verständnis der dynamischen Methoden, in dem es die *KEGG-Datenbank*, die Formatierungssprache *KGML* und den *KGML Editor (KGML-ED)* vorstellt. Zum Ende (Kapitel 4) werden dann die in dem Paper präsentierten Methoden an grafischen Beispielen erklärt. Vier Methoden werden in dem Paper von Klukas und Schreiber präsentiert:

Ausdehnung der Übersicht Eine Art semantischer Zoom, der die Informationstiefe in Teilgraphen erhöht.

Sequentielle Erweiterung Das Aneinanderfügen von Graphen bzw. das Einfügen von nicht vorhandenen Graphen an den aktuellen Graphen.

Signalpfade arrangieren Das allgemeine Arrangieren von Graphen durch Verknüpfung und bekannte Layoutmethoden.

Signalpfade kollabieren Das Gegenstück zur Ausdehnung der Übersicht, das semantische “Herauszoomen”, das die Informationstiefe in Teilgraphen zurücknimmt.

2 Darstellungen in der Biologie

Die Möglichkeiten, Zusammenhänge in der Naturwissenschaft darzustellen, haben sich durch die Computertechnologie stark verändert. Auch die Visualisierung in der Biologie erfährt Veränderungen und zu alten Visualisierungsmethoden kommen neue dazu. Wie in der Einleitung beschrieben, werden dynamische Methoden zur Erstellung und Entwicklung eines Signalweggraphen das zentrale Thema dieser Arbeit sein. Aus diesem Grund soll in diesem Abschnitt genau abgegrenzt werden, was dynamische Darstellung von der traditionellen und auch der semistatischen Darstellung unterscheidet.

2.1 Traditionelle Darstellung und semistatische Darstellung

Die traditionelle Visualisierung in der Biologie ist geprägt von “gemalten” Bildern (siehe die Abbildung 2.1). Diese mögen als Grafiken abgespeichert oder sogar mit einem Grafikprogramm erstellt worden sein, trotzdem existieren sie als monolithisches Objekt. Die Information in diesen Bildern obliegt nur der menschlichen Interpretation, formale Mathematik und informationstechnische Methoden schaffen keine zusätzliche Information. Sie sind statisch, können nicht verändert werden und ihre Informationen können nicht ohne weiteres von Computern interpretiert werden. Weiterhin reduzieren diese Darstellungen im Allgemeinen die ihnen zugrunde liegende Datentiefe und Komplexität, da korrekte Abbildungen schlichtweg zu groß werden würden. Die Abbildung 2.1 gibt eine sehr vereinfachte Darstellung der Glykolyse wieder, da sie die Einflüsse von Betriebsstoffen, Enzymen und die Unterschiede verschiedener untersuchter Spezies idealisieren. Allerdings bringt die traditionelle Darstellung einen Vorteil mit, der durch moderne Darstellungsmethoden verloren zu gehen droht, nämlich eine für den Menschen intuitiv und kognitiv verstehbare Struktur.

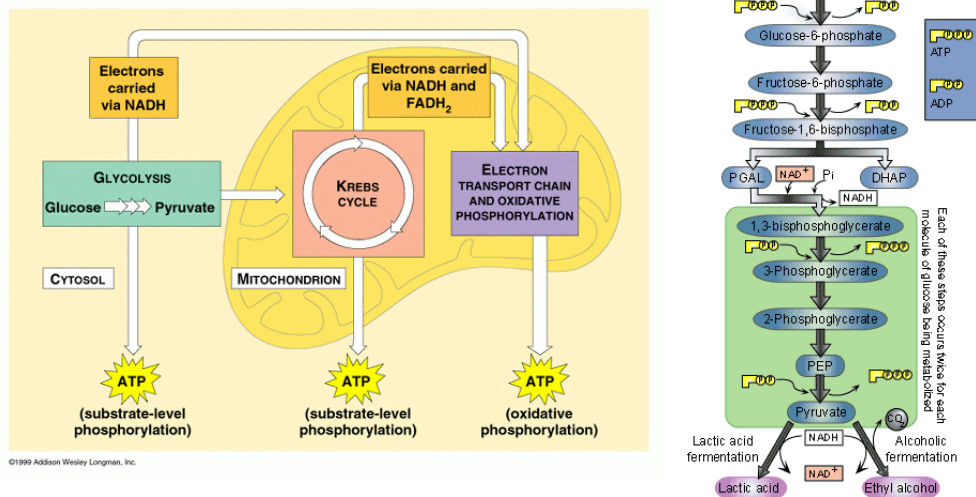


Abbildung 2.1: Traditionelle Grafiken [1][2]

Schaut man genau auf die Abbildung 2.2 wird deutlich, dass sich die Vorgänge in Signalpfaden in Graphenform ausdrücken lassen. Benutzt man Datenbanken, um die Elemente dieser biologischen Vorgänge zu speichern und eine Formatierungssprache, wie zum Beispiel XML, um die Elemente semantisch zu erfassen und dann zu formatieren, erhält man semistatische Darstellungen von Signalwegen. Diese zeichnen sich durch die Verknüpfung von Daten und Grafik aus. Dadurch besteht die Möglichkeit, die Grafik automatisch und dynamisch aus den aktuellen Daten zu erzeugen. Außerdem können solcherlei Grafiken Navigation durch Verlinkung der grafisch dargestellten Elemente unterstützen. In der Abbildung 2.2 ist eine solche Navigation skizziert. Der Anwender klickt auf eine Komponente (in diesem Fall Glycerone Phosphate) und es wird die chemische Abbildung des Moleküls geöffnet. An dieser Stelle tritt das entscheidende Unterscheidungskriterium zur dynamischen Darstellung von Signalpfaden auf, nämlich dass die Navigation nicht innerhalb der Darstellung bleibt, sondern von Grafik zur Grafik geht.

2.2 Dynamische Darstellung

Die Merkmale dynamischer Darstellungen sind eher ergänzend und komplementär zu den Merkmalen semistatischer Darstellung als gegensätzlich. Dynamische Darstellung beinhaltet die Merkmale von semistatischer Darstellung, wie der dynamischen Erzeugung der Grafiken aus Daten, erweitert diese aber, um eine verbesserte Navigation und vor allen Dingen einer damit neu entwickelten Verknüpfung verschiedener in Relation stehender Graphen und Subgraphen. Die Abbildung 4.4 verdeutlicht das Problem:

Die meisten Signalwegdiagramme in KEGG bestehen aus vielen Komponenten, wie Enzymen, chemischen Reaktanten und weiteren Signalpfaden. Nimmt man an, dass der Knoten v_1 ein Referenzknoten eines kompletten Signalpfades G_1 ist, wäre eine Option

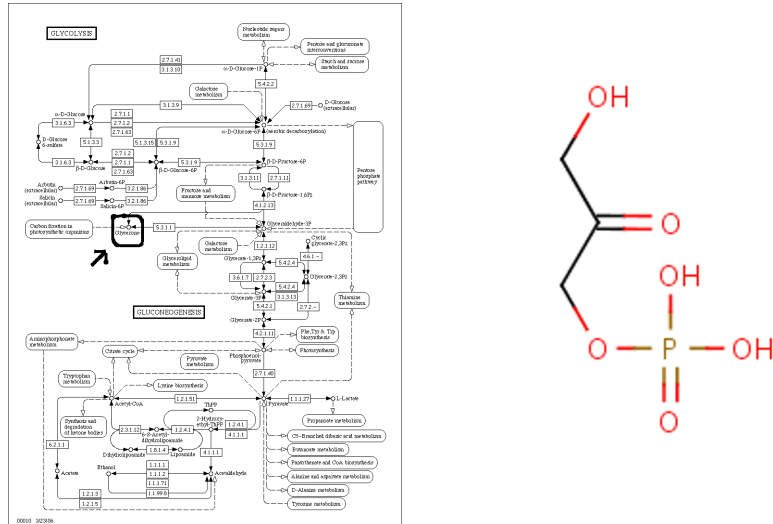


Abbildung 2.2: Dynamik in der Darstellung [3][4]

den kompletten Graphen zu erweitern, indem man v_1 durch G_1 ersetzt, dieser Vorgang wird als semantischer Zoom bezeichnet. Dabei müssen korrespondierende Knoten verbunden werden. Der Anwender soll die Möglichkeit haben, den Graphen durch semantischen Zoom, geometrischen Zoom und Layout zu gestalten. Der so entstandene Graph lässt weiterhin automatische Erstellung und Editierbarkeit des Layouts und der Repräsentation zu, bietet dem Anwender weiterhin die Möglichkeit den Graphen nach seinem Informationsbedarf zu modellieren. Dem in Abschnitt 2.1 angesprochenen drohenden Verlust von für den Menschen intuitiv, kognitiv verstehbarer Struktur, wird damit entgegengewirkt.

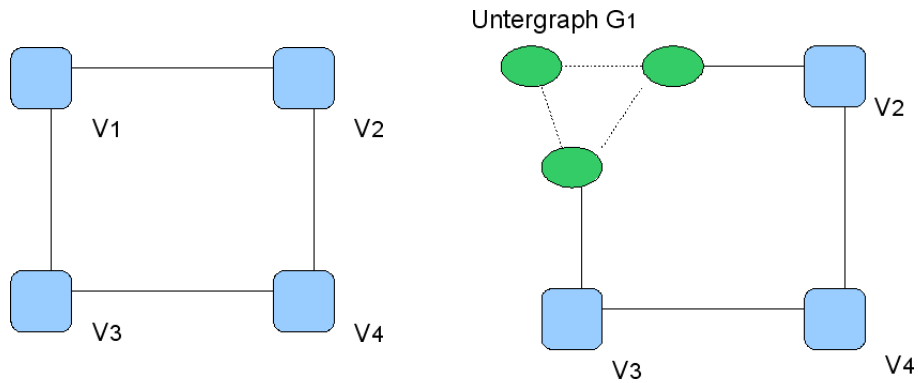


Abbildung 2.3: Dynamische Entwicklung des Signalpfads

3 KEGG, KGML und KGML Editor

3.1 KEGG

Die Kyoto Encyclopedia of Genes and Genomes (KEGG) ist ein Forschungsprojekt der Kanehisa Laboratorien (Universität Kyoto) und des Human Genome Centers (Universität Tokyo). Das Projekt wurde 1995 ins Leben gerufen und ist seitdem betrieben worden. Ziel von KEGG ist eine umfassende computerbasierte Darstellung der Zellen, Organismen und der Biosphäre, um Voraussagen über das Verhalten zellulärer Prozesse auf komplexer Ebene zu machen. So soll das aktuelle Wissen über Interaktionsnetzwerke in biologischen Prozessen, Gene und Proteine und chemische Verbindungen und Reaktionen integriert werden. KEGG ist eine biologische und bioinformatische Datenbank. Die gespeicherten Daten liegen in stark vernetzten Datenbanken und können grafisch durch eine Formatierungssprache, KGML (siehe dazu Abschnitt 3.3), semidynamisch bzw. automatisch erzeugt werden. Neben der dichten Vernetzung der Daten ist die Editierbarkeit und Navigationsfähigkeit der erzeugten Grafiken ein starkes Feature von KEGG. Elemente in den Grafiken sind verlinkt und öffnen Detailübersichten, weitere Signalwege oder weitere Datenbankeneinträge. Die Grafiken bilden die hohe Komplexität der Daten ab, aber die Navigation ist dadurch eingeschränkt, dass sie semistatisch von Bild zu Bild navigiert (und nicht in einem Frame bleibt). Die Rezeption dieser Grafiken im didaktisch, kognitiven Sinn ist durch die Komplexität und Unübersichtlichkeit stark eingeschränkt. Im Kapitel 4 werden Methoden aufgezeigt, die diese beiden Nachteile zum Teil aufheben.

Die Datenbanken bieten zwei grundsätzliche Informationstypen an. Informationen über molekulare Grundbausteine (*Gene, Ligand*) und Informationen über komplexe Systeme (*Pathway, Brite*).

KEGG Gene ist eine Gendatenbank komplett sequenzierter Spezies über:

- Sequenzierte Organismen
- Genomkarten und Informationen über Organismen
- Sequenzähnlichkeiten und KEGG Orthologiegruppen

KEGG Ligand ist eine kombinierte Datenbank über chemische Substanzen bzw. Aktivatoren:

- Strukturen chemischer Verbindungen, insbesondere von Kohlenhydraten
- Chemische Reaktion und Reaktionspartner
- Enzymatische Reaktion und Komponenten

KEGG Pathway stellt die Proteininteraktionsnetzwerke in zellulären Prozessen dar, sie steht im Mittelpunkt dieser Arbeit, da die für den KGML-Editor entwickelten Methoden diese Darstellungen von Signalwegen erweitern:

- Stoffwechselwege, wie z.B. Glykolyse oder Zitratzyklus
- Signalwege, wie z.B. Blutgerinnungskaskade
- Zelluläre Prozesse, wie z.B. Chemotaxis

KEGG Brite sind Hierarchien und Relationen der KEGG-Objekte z.B.:

- Gene \Leftrightarrow Proteine
- Komponenten \Leftrightarrow Reaktionen
- Medikamente \Leftrightarrow Krankheiten
- Zellen \Leftrightarrow Organismen

Der Zugriff auf KEGG ist unter der URL <http://www.genome.ad.jp/kegg/> möglich.

3.2 KGML

KEGG Markup Language (KGML) ist ein Austauschformat für die Graphenobjekte für KEGG, insbesondere für die früher manuell erstellten Signalwegdiagramme. KGML wurde von den Betreibern von KEGG zur semantischen Erfassung der Daten und zur Formatierung der dadurch erstellten Grafiken entwickelt. Dabei bietet KGML zwei starke Vorteile gegenüber herkömmlicher Darstellung. Zum einen lassen sich die Grafiken zu den Signalpfaden aus der Datenbank, zusammen mit dem KGML-Format, automatisch erstellen, und zudem können Netzwerke computergestützt analysiert und modelliert werden. Automatische und dynamische Erzeugung von Grafik meint, dass die Grafiken nicht als Objekte physisch erstellt und dann abgespeichert werden, sondern dass aus aktuellen Daten der Datenbank ein "Bild" generiert wird. An dieser Stelle sei nochmals darauf hingewiesen, dass die Navigation in der KEGG-Grafik, also das Anklicken eines Untergraphen bzw. eines weiteren Signalpfades, eine komplett neue Grafik öffnet, und somit KEGG-Darstellungen nur bedingt dynamisch sind.

3.3 KGML-ED

Der KGML Editor (KGML-ED) ist ein von Klukas und Schreiber entwickeltes Editier- und Visualisierungstool. Die Applikation wurde auf den Grapheneditor *Gravisto* aufgesetzt, editiert und visualisiert Signalwege auf KGML-Basis. Weiterhin unterstützt es den Im- und Export von KGML-Dateien.

KGML-ED bietet viele Möglichkeiten um KEGG-Signalweggraphen komfortabel zu erstellen. Dabei funktioniert er prinzipiell, wie ein normaler Grapheneditor. Über ein Menü werden Knoten und Kanten auf einer Fläche arrangiert. Die Knoten und Kanten können dann online mit Daten aus der KEGG-Datenbank gefüllt werden. So kann man einem

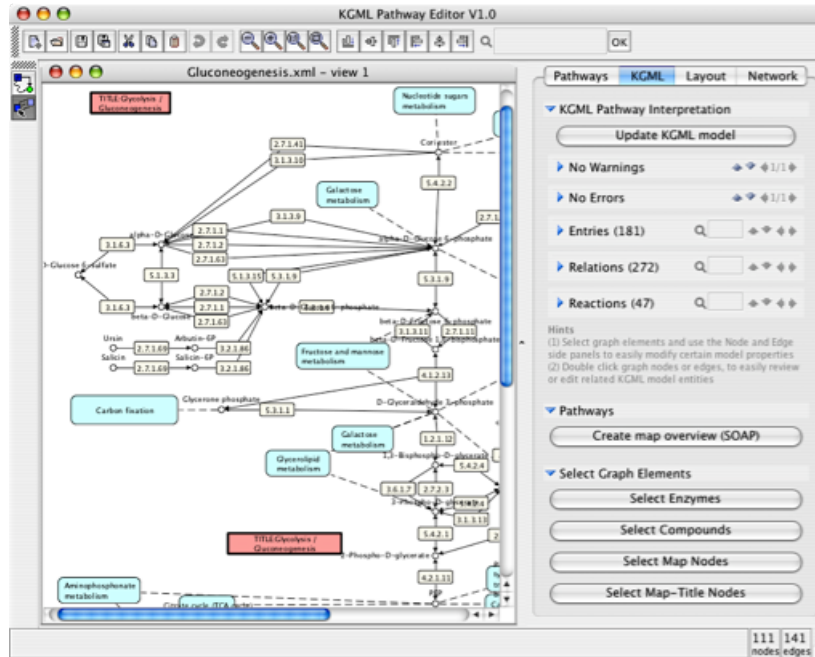


Abbildung 3.1: Screenshot: KGML-ED [5]

Knoten einen bestimmten Glykolyse-Signalpfad zuweisen und dementsprechend mit seinen Kanten verfahren. Die Fähigkeit online Objekte aus den Datenbanken Pathway, Ligand, Gene und Brite abzurufen und diese Information in den Graphen zu arrangieren, schafft die Möglichkeit schnell umfangreiche, komplexe Signalpfade zu erstellen. Als Editor bietet KGML-ED die gängigen Methoden zum Editieren seiner Graphen an. So können Größe, Farbe und Platzierung von Knoten eingestellt werden. Zudem bietet KGML-ED eine Vielzahl von Layoutmöglichkeiten an (Gridlayout, Circlelayout, Forcedirectedlayout usw.). Die erstellten Graphen können in das KGML-Format exportiert werden, so dass eine manuelle Programmierung obsolet ist. Wie bereits erwähnt, ist die dynamische Darstellung und Entwicklung von KEGG-Signalpfadgraphen der Schwerpunkt der von Klukas und Schreiber entwickelten Software. Die Erweiterung, das Arrangieren und das Kollabieren von Graphen werden in dem Kapitel 4 genau beschrieben, stellen aber grundsätzliche, notwendige Methoden zu Verfügung, um Graphen, solcher Informationstiefe, dynamisch zu gestalten.

4 Dynamische Entwicklung

In diesem Abschnitt werden vier Methoden vorgestellt, um Signalwegdiagramme dynamisch zu erstellen. Dazu werden anfangs einige mathematische Grundlagen erklärt. Grundsätzlich sollte zum besseren Verständnis der Methoden, die innere Struktur von KEGG-Diagrammen erläutert werden. Elemente der Diagramme sind nicht nur Proteine, Enzyme, Reaktionen und ähnliche biologische Bausteine, sondern auch Signalpfade selber. Das führt dazu, dass man Diagramme betrachtet, die selber Signalpfade darstellen und zudem Signalpfade enthalten. Die so angeordneten Signalpfade enthalten zudem redundante Bausteine (so kann z.B. ein Enzym in verschiedenen Signalpfaden mehrfach auftreten), was zu redundanten Knoten in den Graphen führt. Weiterhin besitzen alle Signalpfadgraphen, die man in irgendeiner Form arrangieren kann, korrespondierende Knoten (so z.B.: Enzym x in Signalpfad G_i interagiert mit Betriebsstoff y in Signalpfad G_j). Das heißt, wenn zwei Signalpfade G_i und G_j in Relation stehen, muss in G_i ein Knoten v_j und in G_j ein Knoten v_i existieren, die für interagierende biologische Bausteine stehen. Sämtliche der vorgestellten Methoden sind im KGML-ED implementiert und stehen zur Verfügung.

- $G_0 = (V_0, E_0)$ ist der KEGG – Übersichtsgraph
 - $v \in V_0$ ein KEGG-Signalweg
 - $e \in E_0$ eine Beziehung zwischen zwei Signalwegen
- in G_0 repräsentiert jedes $v_i \in V_0$ ein Signalwegdiagramm $G_i = (V_i, E_i)$
 - $v \in V_i$ kann Enzym, Protein oder anderes Objekt sein
 - $e \in E_i$ ist Relation oder Reaktion

4.1 Ausdehnung der Übersicht

Wie bereits erwähnt, bestehen KEGG-Signalweggraphen aus vielen Elementen. Signalwege in Signalwegen können dementsprechend als Knoten oder ganzer Signalweg angezeigt werden. Durch ein semantisches Zoomen kann somit an der gewünschten Stelle im Diagramm die Datentiefe verändert werden, wenn ein Signalweg nur als Referenzknoten angezeigt wird. Technisch ausgedrückt, soll im Übersichtsgraph G_0 ein Knoten v_i durch seinen (Signalweg-) Graphen G_i ersetzt werden.

Prozedur:

- Löschen von v_1 und der redundanten Knoten in G_1 , also v_1 und v_2 .

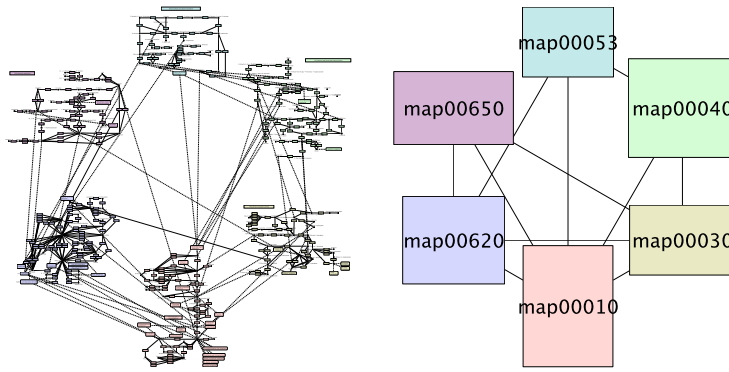


Abbildung 4.1: Signalpfadgraphen und ihr Referenzgraph [6][7]

- Verbinden der offenen Kanten mit den korrespondierenden Knoten.

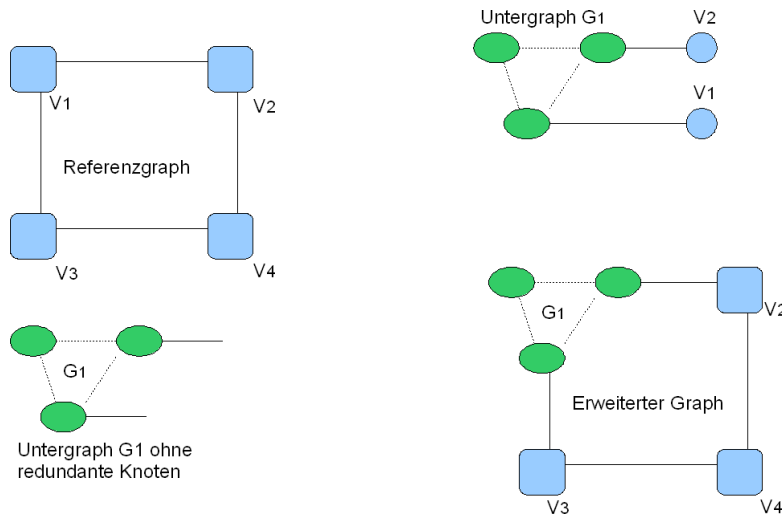


Abbildung 4.2: Erweitern der Übersicht

4.2 Sequentielle Erweiterung von Signalpfaden

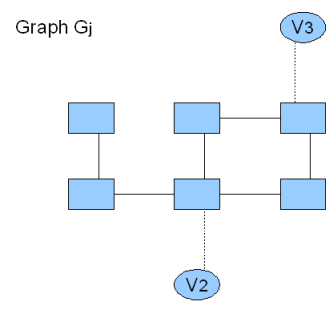
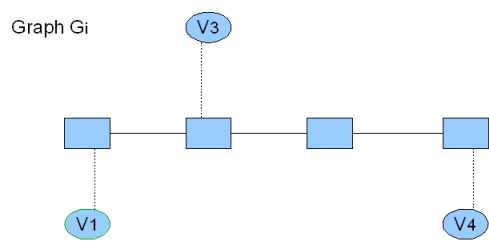
Ein gegebenes Signalwegdiagramm G_i soll sukzessiv um weitere Signalwegdiagramme G_j zu einem Gesamtdiagramm erweitert werden. Die Anforderung mehrere Signalwegdiagrammen zu verbinden wird sequentiell gelöst, indem immer ein Pfad einzeln an den Gesamtpfad oder den Ursprungspfad angebracht wird. In Abbildung 4.3 sieht man oben die beiden Diagramme, die verschmolzen werden sollen. Dabei sind redundante und interagierende Knoten zu beachten.

Der Algorithmus wird von Klukas und Schreiber in vier Schritten angegeben. Es sollte beachtet werden, dass diese Schritte in der Regel gängige Layoutmethoden nach sich

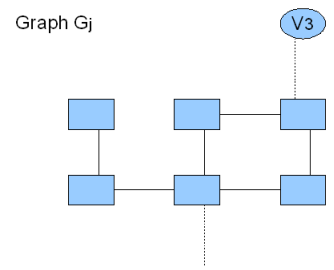
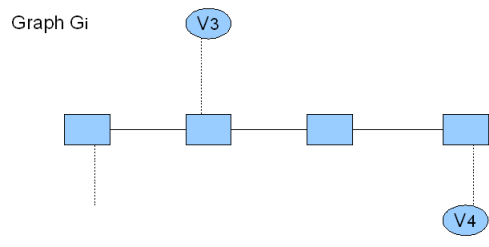
ziehen:

1. Knoten v_j (hier v_1) wird vom Graphen G_i entfernt und Knoten v_i (hier v_2) von G_j .
2. Stattdessen werden die Knoten die mit v_i und v_j verbunden sind, mit neuen Kanten verbunden.
3. Der Signalweggraph G_j enthält unter Umständen Signalwege, die auch in G_i präsent sind (hier v_3).
4. Diese Knoten werden von G_j (hier v_3) entfernt und mit korrespondierenden Knoten in G_i verknüpft.

Die mittig platzierten Graphen in Abbildung 4.3 zeigen die noch nicht verbundenen Graphen, deren interagierenden Knoten bereits gelöscht sind. Der jetzt verbundene Gesamtgraph wird nach redundanten Knoten abgesucht. Diese werden aus G_j entfernt und die losen Kanten mit der noch existierenden Instanz verbunden, siehe rechte Seite der Abbildung 4.3 ganz unten.



Löschen der interagierenden Knoten



Löschen der redundanten Knoten

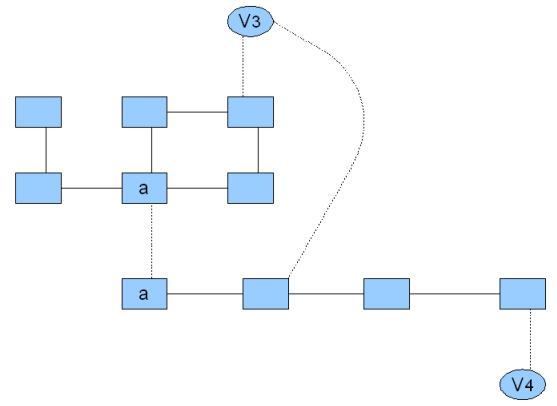
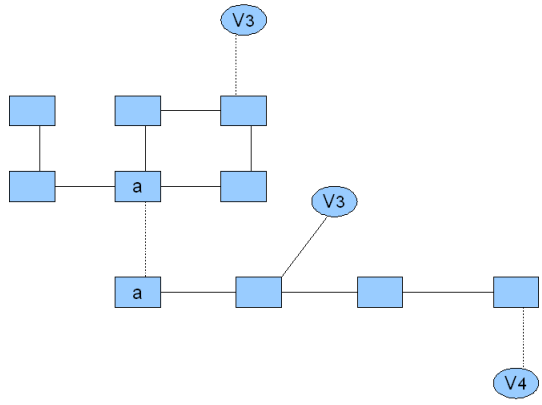


Abbildung 4.3: Ablauf des Algorithmus

4.3 Signalpfade arrangieren

Signalwegdiagramme arrangieren beinhaltet das Erstellen und Layouten eines neuen Signalpfades. Dabei muss die richtige Verknüpfung von Elementen und die Güte des Layout beachtet werden. Layout in seiner Qualität zu bemessen ist dabei nicht nur subjektiv. Die Vermeidung von Knickkanten, Überlappung von Knoten und eine möglichst kleine Anzahl von gekreuzten Kanten sind für einen guten Layoutalgorithmus notwendig. Das Arrangieren von Signalwegdiagrammen ist weiterhin notwendig, da die Diagramme bei fortschreitendem Arbeitsprozess tendenziell größer werden, und somit ständige Anpassung des Layouts stattfinden sollten. Zwar ist ein automatisches Layout notwendig, aber von hohem Rechenaufwand.

Für eine gegebene Menge an Signalpfaden kann das Layout so erstellt werden:

1. Erstellen des Referenzgraphen G_0
 - Jeder Signalpfad stellt einen Knoten v_i da
 - Verbinden der Knoten, genau dann wenn die Signalpfade in Relation stehen
 - Die Größe des Knoten v_i korrespondiert mit der Größe des Signalpfadgraphen G_i
2. Layout durch Force-Layout, Grid-Layout oder Circle-Layout
3. Falls es immer noch Überlappungen gibt, den “Layout-Schritt” wiederholen

4.4 Signalpfade kollabieren

Der entgegengesetzte Schritt zur “Ausdehnung der Übersicht” ist das Kollabieren von Signalpfaden. Der Anwender möchte die Detailtiefe in einem Teil des Signalpfades reduzieren. So würde er im KGML-ED einen detailliert angezeigten Signalpfad markieren und diesen über semantisches “Herauszoomen” als einzelnen Knoten darstellen lassen. Allgemein ausgedrückt soll Untergraph G_i durch seinen Referenzknoten v_i ersetzt werden. Der Knoten v_i wird eingefügt und alle losen Kanten auf ihn projiziert.

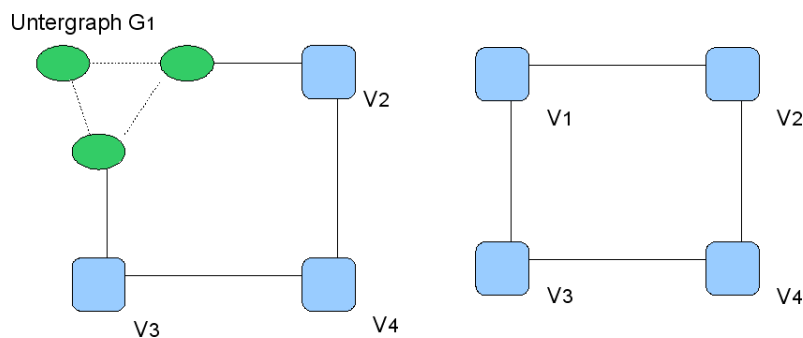


Abbildung 4.4: Semantisches Herauszoomen

5 Zusammenfassung

In der Darstellung von biologischen Kontexten hat es einen Übergang von der traditionellen zur dynamischen Darstellung gegeben. Die Darstellungen benötigen zunehmend mehr Komplexität und damit wächst der Bedarf an technischen Möglichkeiten dieses zu erfüllen. Die Verknüpfung von Daten und Grafik durch Datenbanken, wie z.B. KEGG, und Formatierungssprachen, wie z.B. KGML, bringt in die Darstellung von Signalwegen semistatisches Niveau. Klukas und Schreiber haben, in diesem von KEGG festgelegtem Umfeld, angesetzt und durch den KGML-ED – und die darin implementierten Methoden – die Dynamik der Darstellung von KEGG-Signalwegdiagrammen nochmals erhöht. Gerade die Möglichkeit, Detailtiefen in Bereichen einer Darstellung zu variieren in Kombination mit automatischem Layout ohne die Grafik zu verlassen, bietet Navigation auf dynamischen Niveau. Während die Algorithmen zu den vier vorgestellten dynamischen Methoden im mathematischen Sinn nicht überbewertet werden sollte, ist der KGML-ED als Tool für dynamische Erstellung und Editierung von KEGG-Signalwegdiagrammen ein großer Fortschritt, analog zu den ersten WYSIWYG-Applikationen. Die Software ist dabei in ihre Funktionalität dem durch das Paper gestellten Anforderungen gerecht geworden. Sie bietet sowohl die von Klukas und Schreiber entwickelten dynamischen Methoden an, als auch einen umfangreichen Grundstock an Edititions- und Entwicklungsmethoden. Der Nachteil ist eine kaum vorhandene Dokumentation bzw. eine nur rudimentär ausgeführte Dokumentation. So können viele Features der Software nicht systematisch erschlossen werden. Auch die weniger starke Verbreitung im Internet lässt wenig Schlüsse auf die tatsächliche Akzeptanz des Produkts zu. Zwar wurde KGML-ED schon im Dezember 2006 veröffentlicht, es bleibt aber offen in wieweit KGML-ED im Zusammenhang mit KEGG als Hilfsmittel in der Biologie frequentiert werden wird.

6 Quellen

Referenzen:

Christian Klukas and Falk Schreiber (2007): *Dynamic Exploration and Editing of KEGG Pathway Diagrams*, *Bioinformatics*, 23 (3), 344-350.

Koyoto Encyclopedia of Genes and Genomes - <http://www.genome.ad.jp/kegg/>

KEGG Markup Language - <http://www.genome.jp/kegg/docs/xml/>

KGML-ED - <http://kgml-ed.ipk-gatersleben.de/Installation.html>

Grafiken:

1 Neil Campbell (1998): *Biologie*; Spektrum Akademischer Verlag Heidelberg, Germany.

2 Neil Campbell (1998): *Biologie*; Spektrum Akademischer Verlag Heidelberg, Germany.

3 <http://www.genome.jp/kegg/pathway/map/map00010.html>

4 <http://www.genome.jp/fig/compound/C00111.gif>

5 <http://kgml-ed.ipk-gatersleben.de/Introduction.html>

6 http://kgml-ed.ipk-gatersleben.de/supp_material

7 http://kgml-ed.ipk-gatersleben.de/supp_material