Conditional Integration of Biological Pathways

Alexandra Skolozub
Conditional Integration of Biological Pathways

Research Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

Alexandra Skolozub

Submitted to the Senate of the Technion - Israel Institute of Technology

Adar, 5769 Haifa March 2009
The Research Thesis Was Done Under
The Supervision of Assoc. Prof. Ron Y. Pinter
And with Consulting from Assoc. Prof. Ariel Miller
in the Computer Science Department

I would like to thank my supervisor, Assoc. Prof. Ron Y. Pinter for his guidance, the technical and mental support and caring.

Many thanks to my fiancé, Yura, for helping me, encouraging, believing in me and being by my side during all the ups and downs.

Last but not least, I would also like thank my loving parents, Konstantin and Galina Skolozub, without whose guidance, endless support, encouragement and believing in me, I would not be where I am today.

This work I devote to you, my dears.

The Generous Financial Help of the Technion Is Gratefully Acknowledged
# Contents

Abstract ............................................................................................................ 1  
Abbreviations ................................................................................................... 2  
Chapter 1: Introduction .................................................................................. 3  
Chapter 2: Background .................................................................................. 7  
  2.1 Biological pathways ................................................................................ 7  
    2.1.1 Metabolic pathways ..................................................................... 7  
    2.1.2 Signal transduction pathways ..................................................... 11  
    2.1.3 Gene regulatory pathways .......................................................... 15  
  2.2 Pathway representation ......................................................................... 17  
    2.2.1 Modeling of biological pathways ................................................. 17  
    2.2.2 Biological pathway databases ..................................................... 21  
  2.3 Related work ......................................................................................... 24  
Chapter 3: Method ......................................................................................... 31  
  3.1 The Conditional Pathway Algebra ........................................................ 31  
  3.2 Kinds of conditions ............................................................................. 35  
    3.2.1 AND nodes ................................................................................. 35  
    3.2.2 Conditional edges ....................................................................... 36  
    3.2.3 External (boundary) conditions .................................................. 36  
    3.2.4 Putative conditions ..................................................................... 36  
  3.3 Kinds of explanations ........................................................................... 38  
    3.3.1 AND attribute ............................................................................. 38  
    3.3.2 Local ......................................................................................... 38  
    3.3.3 Upstream .................................................................................. 39  
    3.3.4 Upstream – "Putative Conditions" .............................................. 41  
    3.3.5 External (boundary) conditions .................................................. 41  
  3.4 The Conditional Merge Algorithm ......................................................... 43  
    3.4.1 The algorithm ............................................................................. 44  
    3.4.2 Merge Mode ............................................................................... 49  
    3.4.3 Inconsistency between edges ....................................................... 49  
  3.5 Properties of the Conditional Pathway Algebra .................................... 51  
    3.5.1 Closure ..................................................................................... 51
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5.2.</td>
<td>Commutativity</td>
<td>52</td>
</tr>
<tr>
<td>3.5.3.</td>
<td>Associativity</td>
<td>53</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>The Pathway Integration Environment - PIE</td>
<td>55</td>
</tr>
<tr>
<td>4.1</td>
<td>Cytoscape</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>PIE functionality</td>
<td>59</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Biological Validation</td>
<td>61</td>
</tr>
<tr>
<td>5.1</td>
<td>Pathways under study</td>
<td>62</td>
</tr>
<tr>
<td>5.2</td>
<td>Results and Discussion</td>
<td>71</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>Summary and Future work</td>
<td>89</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>91</td>
</tr>
</tbody>
</table>
List of Figures

Fig. 2.1 The citric acid cycle from (A) KEGG; (B) MetaCyc ........................................9
Fig. 2.2 Overview of a signal transduction pathway taken from "Molecular Cell Biology". (Lodish and Harvey, 2003, 5th edition) .........................................................13
Fig. 2.3 Structure of a gene regulatory pathway (taken from the U.S. Department of Energy’s Genome Program website http://genomics.energy.gov)...........................................................................15
Fig. 2.4 Reaction, glutamate + ATP  gammaT glutamyl phosphate + ADP (EC 2.7.2.11, catalyzed by gamma-glutamyl kinase), modeled by different graph types as demonstrated in Deville et al. (2003) .................................................................18
Fig. 2.5 Compound graph taken from Deville et al. (2003). ..............................18
Fig. 2.6 Reaction graph taken from Deville et al. (2003)...............................19
Fig. 2.7 The network describing the relationship between the expression of IME1 and IME2 using conditional edges (blue edges). (Rubinstein et al., 2007) .......................................................................................................................27
Fig. 2.8 Regulation as the target of another regulation. (A). The ability to define a regulation as a target of another regulation is helpful in cases when the effect of regulator A on target B is transmitted to some but not all targets of B. In this schematic example, A specifically inhibits the B-mediated activation of C. (B). A schematic representation of p53 activation by CHK2. (C). A more specific representation: here, the information that the activation of p53 is achieved by CHK2 interfering with the inhibition of p53 by MDM2 is explicitly represented. This interference is achieved by CHK2 phosphorylation of p53, and therefore, the ‘physical target’ attribute of this regulation is p53. (Elkon et al., 2008).........................................................................................................................29
Fig. 3.1 Conditional algebra operations. (A) Graph Union; (B) Positive Conditional Merge; (C) Negative Conditional Merge; and (D) Mixed Conditional Merge..................................................................................................................33
Fig. 3.2 AND attribute examples. (A) Protein modification; (B) Complex formation.........................................................................................................................35
Fig. 3.3 Conditional edges. (A) Upstream condition; (B) External (boundary) condition; (C) Putative condition ..........................................................................................37
Fig. 3.4 AND attribute explanation.................................................................38
Fig. 3.5 Local explanation..............................................................................39
Fig. 3.6 Upstream (positive) explanations....................................................39
Fig. 3.7 Upstream (negative) explanations...................................................41
Fig. 3.8 Upstream – "putative conditions" explanation.................................41
Fig. 3.9 External conditions explanation.....................................................42
Fig. 4.1 Cytoscape website http://www.cytoscape.org/................................55
Fig. 5.1 Filamentous growth pathway from STKE entry...............................63
Fig. 5.2 Pheromone signaling pathway from STKE entry............................67
Fig. 5.3 Network corrections for the pathways under study........................67
Fig. 5.4 High Osmolarity Glycerol (HOG) pathway from STKE entry...........71
Fig. 5.5 Positive Conditional Merge of the pheromone signaling (in green) and filamentous growth (in red) pathways. Nodes in the intersection of the pathways appear in purple.................................................................73
Fig. 5.6 Details of the Positive Conditional Merge of the pheromone signaling and filamentous growth pathways (excerpts from Fig. 5.5)...............................77
Fig. 5.7 Negative Conditional Merge of the pheromone signaling and filamentous growth pathways. The color coding is as in Fig. 5.5. ......................79
Fig. 5.8 Positive Conditional Merge of the pheromone signaling (in red) and HOG (in green) pathways. Nodes in the intersection of the pathways appear in purple, and Putative Condition (PC) nodes are in white..............................81
Fig. 5.9 Details of the Positive Conditional Merge of the pheromone signaling and HOG pathways (excerpts from Fig. 5.8)..........................................................83
Fig. 5.10 Inconsistent edges during the Positive Conditional Merge of the filamentous growth, pheromone signaling, and HOG pathways.........................85
Fig. 5.11 Inconsistent edges during the Positive Conditional Merge of the pheromone signaling, HOG, and filamentous growth pathways.........................87
List of Tables

Table 3.1 Evaluation of conditional edge.................................................................50
Abstract

Many biological pathways that describe complex cellular processes are available in public and commercial databases as well as in the literature. However, each item focuses on a particular cellular function. Moreover, pathways differ in the way they are described in different sources, emphasizing complementary aspects of the biological system under study. Considering related pathways in a unified framework is essential for understanding their behavior and for elucidating and refining open issues involving such systems.

We developed a conditional pathway algebra, in which pathways are enriched with both new node types as well as additional edge types providing significantly more expressive power for the description of existing biological phenomena. During conditional pathway integration, some interactions are made dependent upon a specific predicate (the presence/absence of protein, co-regulation, extracellular factors, etc.). Moreover, such integration enables distinguishing between different data sources and points out problematic interactions in the given pathways. We provide a formal definition of the algebra and prove some of its properties (e.g. conditional merge and union), such as closure, commutativity, and the lack of associativity. Some of these operations are essential when applied to several pathways to form an entire (sub)system.

The algebra was implemented in the Pathway Integration Environment (PIE) as a plugin for Cytoscape (www.cytoscape.org). To demonstrate the utility and effectiveness of our method, we have applied it to three well characterized yeast signaling pathways: the (i) Pheromone signaling, (ii) Filamentous growth, and (iii) High osmolarity glycerol pathways. Most of our computational observations are confirmed in the literature.
Abbreviations

DNA - Deoxyribonucleic acid
RNA - Ribonucleic acid
mRNA - Messenger RNA
cDNA - Complementary DNA
ChIP - Chromatin Immunoprecipitation
KEGG - The Kyoto Encyclopedia of Genes and Genomes
EC - Enzyme Classification
GRN - Gene regulatory network
STKE - Signal Transduction Knowledge Environment
PIE - The Pathway Integration Environment
PC - Putative condition
Chapter 1: Introduction

Many biological networks and pathways that describe complex cellular processes are being reconstructed and are ready for analysis. This information pertains to a variety of functional aspects and different levels of abstraction, e.g. metabolism, signal transduction, and regulation, each providing a partial perspective on the biological issues at hand. Currently, the descriptions of individual pathways is stored in several – both public as well as commercial – databases (such as KEGG (Kanehisa et al., 2008; Kanehisa and Goto, 2000; Kanehisa et al., 2006), MetaCyc (Caspi et al., 2008; Karp et al., 2004), iPath (Letunic et al., 2008), Reactome (Matthews et al., 2007), and STKE (http://stke.sciencemag.org/); see (Vihinen, 2003) for a review) using a variety of representations and presentation mechanisms; a wealth of information can also be found in the literature (in e.g. PubMed, http://www.ncbi.nlm.nih.gov/pubmed/), where typically each paper describes the properties of a specific pathway or subsystem. Consequently, it is near to impossible for a life science researcher to glean a comprehensive and integrative view of several pathways under one framework, which is a clear need for the systematic study of biological processes beyond encapsulated, specific functions. For example, when trying to elucidate drug response, it is critical to capture the impact of a substance (be it constitutive or administered) in its entirety. Thus we propose a method – and a tool that is based on it – for the integration of pathways reflecting inconsistencies among the data sources and providing means to overcome them in subsequent – both static as well as dynamic – analyses.

Examination of several pathways simultaneously can enhance our understanding of the biological system under study. The pertinent pathways can be different, or be separate instances of the same pathway coming from various data sources, each emphasizing distinct aspects of the biological process. By combining these issues we can recognize e.g. regulating factors in pathways and crosstalk between them. Such deep understanding of the biological systems can benefit the investigation of e.g. positive response,
resistance to treatment, and adverse drug reaction in pharmacogenetic studies (Goldstein et al., 2003; Meyer and Ginsburg, 2002; Roses, 2001).

The challenge of pathway integration is compounded by both biological as well as technical issues. It is well known that cells of different types and from distinct tissues under various environmental conditions have different gene expression profiles, which are the results of activating slightly different pathways (Bowcock and Krueger, 2005; Chi et al., 2006; Lindberg and Kappos, 2006; Liu et al., 2006; Staudt and Brown, 2000). Activating factors also play an important role in protein\(^1\) modes and localization within a cell. Data inconsistency between different data sources, i.e. the same reaction is described differently, should not be forgotten either. During pathway integration we should also be aware of the possibility of diverging file formats used for pathway storage (XML-based formats: BioPAX (http://www.biopax.org/); SBML (Hucka et al., 2003); KGML (http://www.genome.jp/kegg/xml/); XGMML (Punin and Krishnamoorthy, 1999) and text formats such as SIF and etc.). In addition, not all databases provide user-friendly data downloading, not to mention pathways that researchers obtain from articles. Moreover, studied pathways are often divided into several sub-pathways as a result of technical limitations of storage and representation. Therefore, inappropriate cutoffs should also be taken into consideration.

Hence, during pathway integration all of the abovementioned stumbling blocks should be accommodated. Customarily, pathways of all kinds – be it signaling, transcription regulation, or metabolic networks – are modeled as labeled, directed graphs, where nodes represent genes and their products (mRNA, proteins, complexes, etc.) and edges represent interactions between them (Fukuda and Takagi, 2001). This commonly used abstraction of a biological pathway is not expressive enough to reflect minor but potentially critical differences between them. Many of those differences that include tissue, cell type, cell condition etc. are expressed verbally by biologist however cannot be included in current representation. This representation can be enriched in several ways, as has been proposed by several authors (Choi

\(^1\) We refer to proteins, but this holds for other biological entities as well.
et al., 2004), by e.g. associating a variety of attributes with the nodes and the edges and interpreting their proximity in e.g. a conjunctive or disjunctive manner. We hereby focus on those extensions that are instrumental in characterizing the conditions and dependencies under which pathways operate, both externally as well as internally; as noted above, this is crucial when trying to put together several pathways and study them in tandem.

External circumstances can be manifested by adding nodes that represent the conditions under which the pathway was observed, such as cell type, tissue and physiological conditions, and linking them to the affected original nodes. Internal conditions are expressed using node attributes that describe cellular localization, activation state, modifications, etc. Another interesting type of conditional dependency is achieved by defining conditional edges that each start at a node and point to an edge (Rubinstein et al., 2007). These edges are used to model the case in which the activity of the (signaling or regulation) edge that has been pointed at depends on the state of the start node; furthermore, they can reflect either positive or negative dependencies. Note that a conditional edge is a triple (rather than a pair) of nodes, thus – formally – it can be regarded as a hyperedge of size 3. Hence pathways – in this representation – are actually directed hypergraphs (where the maximum size of a hyperedge is 3).

Another extension to the representation of pathways is the addition of nodes that represent assumptions that reflect distinctions between pathways of different origin. These are used to maintain decisions that have been made during an integration process and they help in tracing the source of the information that is being retrieved. In particular, these nodes are used as start nodes of conditional edges and thus allow analysis of an integrated pathway in a systematic fashion.

Thus we developed a conditional pathway algebra, in which a "simple" pathway graph is enriched both with new node types as well as additional edge types accompanied with various attributes. We provide a formal definition of the algebra and prove some of its properties, such as closure and commutativity, and demonstrate its lack of associativity. Of these the most important is closure, i.e. ensuring that the resulting pathway is always contained in the set of pathways to which our operations are applicable.
The algebra is embodied in the Pathway Integration Environment (PIE) as a plugin for Cytoscape, a general-purpose, open-source environment for the large scale integration of molecular interaction network data and its visualization (Shannon et al., 2003). Cytoscape’s core is extensible through a plug-in architecture, allowing usage of already existent properties and the development of additional computational analyses and features. We will also describe in detail features that were used and our extensions. We applied PIE to all possible two-way as well as three-way integrations of three well characterized yeast signaling pathways: the i) Pheromone signaling; ii) Filamentous growth, and iii) High osmolarity glycerol (HOG) pathways; the obtained results are confirmed in the literature.

The rest of this thesis is organized as follows: Chapter 2 presents some background on biological pathways, their representation, and related work. In Chapter 3 we present the Conditional Pathway Algebra, the conditional merge algorithm concluding by proving properties of the algebra. Then, in Chapter 4, the implementation of PIE and its usage are described followed by its application to real biological data along with an evaluation of the results in Chapter 5. This report is concluded in Chapter 6, where a summary and some future work issues are mentioned.
Chapter 2: Background

2.1 Biological pathways

Biological networks are used to represent a collection of interactions or functional relationships among the physical and/or genetic components of a cell. A pathway is a fragment of a biological network that describes a known physiological process or phenotype. Biological (biochemical) pathways can be of several types: metabolic, signal transduction, and gene regulatory pathways.

2.1.1 Metabolic pathways

Metabolic pathways are composed of a series of biochemical reactions catalyzed by enzymes, and connected by their intermediates: substrates of one reaction are the products of previous ones, and so on. In Fig. 2.1 we can see metabolic pathways as retrieved from the KEGG and the MetaCyc databases, representing the homo-sapiens citrate acid, also known as the tricarboxylic acid cycle (TCA cycle) or the Krebs cycle. The biochemical reactions that comprise these pathways usually occur in one direction (although all reactions are chemically reversible, cell conditions are such that it is thermodynamically more favorable for flux to be in one direction). A substrate enters a metabolic pathway depending on the cell’s needs and the availability of a substrate. An increase in the anabolic and catabolic end product concentrations slows the metabolic rate of the particular pathway. Metabolic pathways are often regulated by a feedback inhibition or by a cycle in which one product of the cycle starts the reaction again, such as occurs in the Krebs Cycle. Anabolic and catabolic pathways in eukaryotes are separated by either compartmentation or by their usage of different enzymes and cofactors (minerals, vitamins) that catalyze the reactions that are needed for proper functioning. In addition to being used immediately, products of
metabolic reactions can be stored by the cell, or initiate another metabolic pathway.

Fig. 2.1 The citric acid cycle from (A) KEGG; (B) MetaCyc.
2.1.2 Signal transduction pathways

Signal transduction is the process of information (in the form of signals) being transferred from an extracellular medium to a cell's nucleus. A signal transduction process coordinates a series of reactions and interactions realizing a signal transduction, which is carried out by enzymes and is activated by secondary messengers. During this process a series of proteins undergoes a covalent modification, namely phosphorylation. Such processes are usually rapid, lasting on the order of milliseconds in the case of ion flux, or minutes for an activation of protein and lipid-mediated kinase cascades; some can take hours, and even days (as is the case with gene expression), to complete. The number of proteins and other molecules participating in events involving signal transduction increases as a process flows out from an initial stimulus, resulting in a "signal cascade", beginning with a relatively small stimulus that elicits a large response. This is referred to as signal amplification. A failure during signal transduction may cause many common diseases, such as cancer, diabetes, psoriasis and multiple sclerosis. In Fig. 2.2 we can see a typical signal transduction pathway in molecular cell biology (Lodish and Harvey, 2003), which is activated by different extracellular signals that are transferred to the nucleus by messengers.
Fig. 2.2 Overview of a signal transduction pathway taken from *Molecular Cell Biology* (Lodish and Harvey, 2003, 5th edition).
2.1.3 Gene regulatory pathways
To describe this kind of pathways we quote the U.S. Department of Energy's Genome Programs website http://genomics.energy.gov as follows:

"Gene regulatory networks (GRNs) are the on-off switches and rheostats of a cell operating at the gene level. They dynamically orchestrate the level of expression for each gene in the genome by controlling whether and how vigorously that gene will be transcribed into RNA. Each RNA transcript then functions as the template for synthesis of a specific protein by the process of translation. A simple GRN would consist of one or more input signaling pathways, regulatory proteins that integrate the input signals, several target genes (in bacteria a target operon), and the RNA and proteins produced from those target genes. In addition, such networks often include dynamic feedback loops that provide for further regulation of network architecture and output. As indicated in the schematic below, input signaling pathways transduce intracellular and/or extracellular signals to a group of regulatory proteins called transcription factors. Transcription factors activated by the signals then interact, either directly or indirectly, with DNA sequences belonging to the specific genes they regulate. The factors also interact with each other to form multiprotein complexes bound to the DNA."

Fig. 2.3 illustrates the structure of a typical gene regulatory pathway.

Fig. 2.3 Structure of a gene regulatory pathway (taken from the U.S. Department of Energy’s Genome Program website http://genomics.energy.gov).
2.2 Pathway representation

2.2.1 Modeling of biological pathways

Biological pathways are modeled using a graph theoretic framework. A labeled graph $G(V, E)$ is a mathematical object where $V$ is a set of nodes and $E$ is a set of edges, connecting pairs of nodes. An edge is an ordered pair of nodes for directed graphs and an unordered pair for undirected graphs. An overview of models for the analysis of biochemical pathways can be found in Deville et al. (2003).

Compound graphs

A compound graph is used for modeling chemical reactions (see Fig. 2.4). Nodes are labeled with chemical compounds. Edges between compounds can be defined in two ways. An undirected edge connects two compounds if they occur in the same reaction without distinguishing substrates and products. The more common approach uses directed edges: a compound A is connected to a compound B if A functions as a substrate and B as a product in the same reaction. Compound graphs are commonly used for modeling metabolic reactions and are usually called metabolite graphs. Metabolite graphs illustrate chains of chemical reactions.

Given that compound graphs are very simple, they have a lot of coverage limitations. One particularly significant limitation is the loss of a reaction’s structure. It is impossible to distinguish whether two substrates or two products are involved in the same reaction. Different sets of reactions can lead to the same compound graph (Friedler et al., 1992), thus introducing ambiguity (see Fig. 2.5). In addition, no information about the enzymes catalyzing these reactions can be stored.

In addition, compound graphs are also very limited in how they are able to represent different biological entity types (compounds, genes, proteins and etc.) and types of interactions represented by edges (assembly, transcriptional regulation, protein–protein interaction, and translocation).
Fig. 2.4 Reaction, glutamate + ATP \to gamma-glutamyl phosphate + ADP (EC 2.7.2.11, catalyzed by gamma-glutamyl kinase), modeled by different graph types as demonstrated in Deville et al. (2003).

Fig. 2.5 Compound graph taken from Deville et al. (2003).

**Reaction graphs**

The dual form of a compound graph is a reaction graph. Here, nodes represent reactions and an edge between reactions R1 and R2 is added if a product of R1 is used as a substrate in R2. In fact, the reaction graph is the line graph of the corresponding component graph. The graph can be directed or undirected, depending on whether the reactions are considered reversible or not. In Fig. 2.4, a reaction graph is reduced to a single node, since only one reaction is involved.
Reaction graphs are mostly limited to representing metabolic pathways since it is there that chemical reactions occur. The most commonly used reaction graph is a metabolic enzyme graph in which nodes are the EC (Enzyme Classification) numbers of enzymes that catalyze reactions (NC-IUBMB). This type of reaction graph is basically inappropriate for modeling non-metabolic networks.

Since compound and reaction graphs are dual, the representation limitations are similar. Both graphs do not provide information about reaction components. Therefore, as noted above, different sets of reactions can lead to the same reaction graph (Friedler et al., 1992), thus introducing ambiguity (see Fig. 2.6).

Nonetheless, although compound and reaction graphs offer only partial and sometimes ambiguous views of biochemical networks, such representations turn out to be sufficient and useful for some simple pathway analyses such as when examining topological and statistical properties, or looking for basic patterns (Yeger-Lotem et al., 2004; Alon, 2007). These representations can also be helpful in some specific applications, such as detection of functionally related enzyme clusters, as noted in Ogata et al. (2000).

Fig. 2.6 Reaction graph taken from Deville et al. (2003).
To overcome the limitations of the two abovementioned pathway representations, bipartite graphs and hypergraphs can be used. These graphs are a combination of component and reaction graphs, as explained next.

**Bipartite graphs**
There are two kinds of nodes in a bipartite graph: compound nodes and reaction nodes. Edges cannot connect nodes from the same set; thus, an edge necessarily links a compound node and a reaction node (see Fig. 2.4). Edges can be undirected or directed. A directed edge from a compound node to a reaction node denotes a substrate of a reaction, while an edge from a reaction node to a compound node denotes a product of a reaction. Bipartite graphs can represent reactions without any ambiguity, as illustrated in Figs. 2.5 and 2.6.

**Hypergraphs**
Hypergraphs, which connect a set of substrates to a set of products, are generalizations of compound graphs. Directed hypergraphs have been used to model metabolic pathways (Wang et al., 2006). Unlike component and reaction graphs, hypergraphs provide an unambiguous representation of the reactions and compounds in biochemical networks. Consequently, they are used in various biological pathway analyses such as of topological properties, path finding, synthesis and prediction.
2.2.2 Biological pathway databases

Pathway databases are a source of machine readable data describing biological processes that are essential in research. These databases enable researchers to arrive at new insights based on or via existing knowledge. Pathways are stored as images in addition to structured or semi-structured representations, the access to is, unfortunately, not always free. Moreover, pathway databases provide tools for visualization and a variety of analyses that can enhance understanding of the biological systems under study.

Metabolic pathway databases

Metabolic pathway databases generally contain detailed data about biochemical reactions. Today, these databases also include representations of high order cellular processes, which offer users the option of zooming in and seeing the reactions that comprise the processes. These biochemical reactions are usually represented using metabolite (compound) graphs or enzyme (reaction) graphs. Metabolic databases predominantly contain prokaryotic pathways, for which rich datasets have been collected. Researchers, working on well-studied organism pathways, have now been able to map the pathways of other organisms based on functional annotations, mostly Enzyme Classification (EC) numbers and ontology relationships. Unfortunately, these techniques are far from perfect, and therefore, gaps (missing steps in a chain of biochemical reactions) and mistakes can be found (Mano et al., 2009).

The most known publicly available metabolic databases are KEGG (the Kyoto Encyclopedia of Genes and Genomes) (Kanehisa et al., 2008; Kanehisa and Goto, 2000; Kanehisa et al., 2006), BioCyc (Karp et al., 2004), Invitrogen iPath (Letunic et al., 2008), Reactome (Joshi-Tope et al., 2005) and BioCarta (www.biocarta.com).

Signal transduction pathway databases

Signal transduction pathway databases are not as well-established as the metabolic pathway databases. Signaling pathways are studied in multi-cellular organisms, commonly in eukaryotes. This means that in some bacteria these pathways are more complicated than the metabolic pathways and it takes
longer to decode them. Recently, scientists have been paying considerable
attention to signaling pathways since it has been observed that an abnormal
signaling process may be the cause of many diseases, e.g., cancer, diabetes,
psoriasis and multiple sclerosis. The result is a growing number of signal
transduction databases and the amount of data in them.

The most well-known databases for signaling pathways are STKE (http://stke.sciencemag.org/), Invitrogen iPath (Letunic et al., 2008), Reactome (Joshi-Tope et al., 2005), BioCarta (www.biocarta.com), The Signaling Gateway (http://www.signaling-gateway.org/), TRANSPATH (Schacherer et al., 2001) and Spike (Elkon et al., 2008).

**Protein interaction databases**
Protein interaction databases contain information about protein-protein,
protein-DNA etc. interactions. These interactions are products of high-
throughput experiments that provide a lot of data. The commonly used
experiments are: yeast 2-hybrid experiments (protein-protein interactions)
(Young, 1998), band shift assay (Vergnon and Chu, 1999), chromatin
immunoprecipitation followed by cDNA microarray analysis (ChIP) (protein-
DNA interactions) (Wagschal et al., 2007). Protein interaction databases
contain many more interactions than any other type of pathway database.
Unlike metabolic and signaling pathway data, which are generated primarily
by traditional small-scale experimental techniques and have low rates of false
positive, protein interaction data has a high rate of false positive results (von
Mering et al., 2002). Therefore, it is very important for researchers to check
the confidence level and experimental evidence about the interactions they
are interested in further analyzing. Examples of databases that store this kind
of information are BIND (Bader et al., 2003), GRID (Breitkreutz et al., 2003),
IntAct (Hermjakob et al., 2004), MINT (Zanzoni et al., 2002) and others.

**Gene regulatory pathway databases**
Gene regulatory pathway databases share features with both signaling and
protein interaction databases, as they collect protein–DNA interactions and
regulatory (activation and inhibition) events. Some databases store data about
protein-DNA interactions received from high-throughput assays, such as ChIP
(chromatin immunoprecipitation followed by cDNA microarray analysis) (Wagschal et al., 2007) and band shift assays (Vergnon and Chu, 1999). These interactions do not provide any information regarding the functional consequences of protein-DNA binding. Only a few databases store data about gene regulatory pathways and one of these is TRANSFAC (Wingender et al., 2000).

The classification into metabolic, signaling, protein interaction and gene regulation pathways is not logically disjoint. Therefore, some databases, such as BioCarta and Reactome, span multiple categories, which we might classify as both a metabolic and a signaling database.
2.3 Related work

No related work has been performed in the field of cellular processes integration. However, considerable research has been conducted in pathway modeling and representation - the first stage in the pathway integration process. In this section, we will describe shortly a few attempts in defining an ontology, bipartite representation and conditional edges.

Ontologies

An ontology is a structured semantic encoding of knowledge. Gene Ontology (GO) is a structured and precisely defined and controlled vocabulary for describing gene function across several organisms (Ashburner et al., 2000). GO classifies genes into three parallel categories represented as three directed acyclic graphs: biological processes, molecular functions, and cellular components. This categorization helps in defining the function of a gene product at various levels, including its biochemical activities, biological roles and cellular structure. Nodes can often be reached through multiple paths, which facilitate the representation and comparison of genes with multiple functions or once that are involved in more than one process.

Karp (2000) invested a great deal of effort in developing an ontology for cellular processes that is referred to as a functional ontology. He defined different types of molecules with their own class; different molecule states are dealt with as different entities. In addition, his ontology defined reactions as independent entities, and asserted that distinct relations link molecules to the reactions. Each molecule may optionally be tagged with a cellular compartment. This ontology can encode a diverse array of biological processes, including enzymatic reactions involving small-molecule and macro-molecular substrates, signal transduction processes, transport events, and mechanisms of regulation of gene expression.

Demir et al. (2004) also built an ontology for cellular processes, which is suitable for modeling incomplete information and abstractions of varying levels, very similar to what exists for biological pathways. Moreover, Demir et al.’s ontology makes easier to perform extensions and concurrent
modifications to existing data while maintaining its validity and consistency
than Karp's. Demir et al. realized that when a network's biological entities are
in different states, they might have other biological functions and participate in
distinctive processes. A combination of an entity with its state was defined as
a bioentry. Transitions were defined, taking into account the fact that they
occur only when all of the substrates are present and activation conditions are
satisfied, as a function of the presence (or absence) of certain other actors.
Under different conditions, different subsets of transitions may occur, leading
to different cellular responses. Molecular complexes were also mentioned.
Each member of a molecular complex was considered to be in a new state of
its biological entity. Different levels of abstraction are available and this is
useful since current biological data is incomplete and complicated. To model
the abovementioned ontology, the authors used a compound graph from
Fukuda and Takagi (2001). A compound graph in their work is an extension of
a graph in which each node can contain a graph inside itself. This model is
very similar to the UML model (D'Souza and Wills, 1998).

**Bipartite representation**

Signaling pathways were modeled as bipartite directed graphs in Choi et al.
(2004). Recall that a bipartite directed graph has two node types: nodes
representing "molecules" and nodes representing "reactions". Edges
represent the molecules' ability to enter or exit a reaction. The reactions'annotation ("complex formation" or "phosphorylation") can be derived from the
context, i.e., the type of molecules that participate in a given reaction. The
authors noted that there could be a logical connection between incoming (or
exiting) edges of one "reaction" or "molecule" node. An AND logical
connection between edges entering or exiting a "reaction" node indicates that
all molecules are needed to let this reaction happen, or all products have to
appear as its result, respectively. The second mentioned connection is an
XOR connection, a logical connection between several incoming (or several
exiting) edges at a "molecule" node emphasizing that an individual molecule
can be produced or consumed by only one particular reaction. An OR
connection may also appear between edges entering (or exiting) a "reaction"
node, and be used for reactions with several input (or output) alternatives.
Beyond mentioning these logical connections, the authors did not demonstrate their use. They also characterized molecules by a number of attributes: biological species, tissue, cellular or intracellular location, their molecular/functional classification, their post-translational modification status, and their complex status. This information can be stored, but is rather neglected during the integration process in which reactions are concatenated together to receive a pathway network. As in BioCyc, they have a three-layer data model, "the evidence level", which includes published data from wet experiments; "the pathway level", in which reactions are manually annotated by experts, and "the semantic projection level", which stores simplified schemes focusing only on the components along a pathway that are directly involved in processing the signal. To summarize, Choi et al.’s model has more expressive power and does not use defined attributes for building pathways from separate reactions than commonly used ones.

**Conditional edges**

Conditional edges, previously defined in Rubinstein et al. (2007), were named "conjunctive condition (AND)" edges. They were used to model the fact that a regulator's activity depends on the presence or absence of some other elements in a biological system. Formally, the dependency of an edge from node i to node j on a state of node k was denoted by $c_k(i,j)$. Two possibilities for this dependency are allowed: if the effect of i on j requires node k to be active, then $c_k(i,j)=1$ (positive dependence); if this effect requires that node k is inactive, then $c_k(i,j)=-1$ (negative dependence). In addition, a graphical representation of these edges was presented; we use the same graphical representation in our current work (see Fig. 2.7).
The same graphical representation can be seen in Elkon et al. (2008), but there it has a different meaning. There it defines a regulatory interaction between any two biological entities, or between a biological entity and another regulation. This enables researchers to describe regulations that affect only a subset of downstream regulations emanating from an entity, as illustrated in Fig. 2.8.A. Second, it improves the specificity of the description, as demonstrated by the comparison between Figs. 2.8.B and 2.8.C. For regulations that act on other regulations, an additional attribute is added: the physical target, which specifies the physical entity on which the biochemical interaction is exerted (see Fig. 2.8.C).
Fig. 2.8 Regulation as the target of another regulation. (A). The ability to define a regulation as a target of another regulation is helpful in cases when the effect of regulator A on target B is transmitted to some but not all targets of B. In this schematic example, A specifically inhibits the B-mediated activation of C. (B). A schematic representation of p53 activation by CHK2. (C). A more specific representation: here, the information that the activation of p53 is achieved by CHK2 interfering with the inhibition of p53 by MDM2 is explicitly represented. This interference is achieved by CHK2 phosphorylation of p53, and therefore, the 'physical target' attribute of this regulation is p53. (Elkon et al., 2008)
Chapter 3: Method

To perform integration of pathways they must be represented as mathematical objects to which we can apply well-defined operations, thereby forming an algebra. We extend the common abstraction by which a pathway is represented as a labeled directed graph $G(V,E)$, where proteins, genes, and small molecules are represented as labeled nodes, and interactions are represented as edges: we add both node and edge types to reflect conditioning of activation.

3.1 The Conditional Pathway Algebra

We model a biological pathway as a labeled, directed graph $G(V,E)$, where the nodes represent biological entities (proteins, genes, small molecules, enzymes, mRNA, etc.) and the edges represent interactions and other relations between them. The set of nodes $V$ comprises three types: regular biological entities ($V_b$), external (boundary) conditions ($V_c$), and putative conditions ($V_p$), which support the pathway integration process. Each node is labeled with a name, as detailed below. The set of edges $E$ contains both common interactions between pairs of nodes in $V_b$ ($E_{comm} \subseteq (V_b \times V_b)$, as well as conditional edges ($E_{cond}$, see formal definition below) where each starts at a node (of any type) and points to a common edge (Rubinstein et al., 2007). The latter are used to model the case in which the activity of the common edge that is pointed at depends on the state of the start node, as explained below. Both types of edges can reflect either a positive or a negative effect: common edges denote e.g. activation or repression in regulation and signaling pathways, and conditional edges can represent either a positive or a negative dependency.

Names of regular nodes are taken from a common namespace of genes, proteins, etc.; we assume that the same name is being used to denote a
certain biological entity in all pathways under study (or that they are normalized by a pre-processing step) so as to allow their identification during the integration process. Additional attributes can be assigned both to whole pathways as well as to individual nodes: the former include cell type, tissue and physiological conditions, and the latter are e.g. cellular localization and activation state.

Finally, we are ready to define conditional edges. For a given pathway $P$, let $\text{cond}_{w}(e)$ be a conditional edge between node $w$ ($w \in V$) and an edge $e = (u_b, v_b)$, denoted $w \rightarrow e$ or $w \rightarrow (u_b, v_b)$. $\text{cond}_{w}(e)$ expresses the fact that the activity of the pointed-at edge $e$ is conditioned upon the status of the node $w$, e.g. whether it is active or not.

There are 3 options for $w$:

- It is a common node [$w \in V_b$]: Note that $u_b$ in addition to being the source node of the edge $e$ can also function as $w$, i.e. $w = u_b$.
- It is an external conditions node [$w \in V_e$]: $w \rightarrow (u_b, v_b)$
- It is a putative conditions (PC) node [$w \in V_p$]: $w \rightarrow (u_b, v_b)$

Notice, that the biological meaning of each type of conditional edges is different. Specifically, the last kind expresses a possible explanation for an edge $e$ which is beyond a pathway’s scope. One interpretation of a source PC node may be the presence of an additional protein or mRNA missing in the original pathway that has a role in regulating an edge it points to. By specifying pathways in which an edge $e$ is present, the PC node can help in emphasizing that this edge is missing in other pathways.

Furthermore, note that a conditional edge is a triple (rather than a pair) of nodes, i.e. $E_{\text{cond}} \subseteq ((V_b \cup V_e \cup V_p) \times V_b \times V_b)$; thus – formally – it can be regarded as a hyperedge of size 3. Hence pathways in our representation are in fact directed hypergraphs where the maximum size of a hyperedge is 3. Still, for sake of presentation, unless it is necessary we refer to conditional hyperedges as edges; similarly, we disregard the signs of edges (positive or negative). To summarize, a pathway is represented as $P(V, E)$, where $V = V_b \cup V_e \cup V_p$ and $E = E_{\text{comm}} \cup E_{\text{cond}}$ with labels associated with nodes, and signs – with edges.
Now four operations are defined between the two pathways $P_1$ and $P_2$:

1. **Graph Union** (denoted $P_1 + P_2$), performed as the set union of both nodes and edges and is based on their name equality (Fig. 3.1.A).

2. **Positive Conditional Merge** ($P_1 \oplus P_2$, Fig. 3.1.B). The result is a conditional pathway which is a refinement of the results obtained by graph union (Operation 1 above) where conditional edges are added that point to differences between $P_1$ and $P_2$ thereby giving possible explanations to these discrepancies. The refinement is performed using conditional edges of the positive type that point to "edges that must be explained" (as the result of the Conditional Merge Algorithm, defined in Section 3.4 below).

3. **Negative Conditional Merge** ($P_1 \ominus P_2$, Fig. 3.1.C). As in Operation 2, but the conditional edges that are added are of the negative type.

4. **Mixed Conditional Merge** ($P_1 \oplus \ominus P_2$, Fig. 3.1.D). The result is a conditional pathway composed of the result of Operation 1 and both positive and negative conditional edges pointing to "edges that must be explained".

**Fig. 3.1** Conditional algebra operations. (A) Graph Union; (B) Positive Conditional Merge; (C) Negative Conditional Merge; and (D) Mixed Conditional Merge.
3.2 Kinds of conditions

The representation of biological network can be enriched by the ability to express conditions on the edges. All of the conditions can be expressed by one of the four following graphical and logical representations.

3.2.1 AND nodes

When several common edges go into a single node their joint effect is open to interpretation and there are various dynamic models that can be subsequently used. This issue is beyond the scope of this thesis, but we found it useful – for purposes of the integration process – to qualify a node as an AND node if all its incoming edges must be active in the system for this node to be activated. Look, for example, at the protein modification process in Fig. 3.2.A; here, both a protein that undergoes modification and its corresponding enzyme have to be present in the system for the activation to occur. If one edge is missing, the protein will not undergo modification. Another example can be a complex formation (Fig. 3.2.B). Let X be a complex; then all its components must be present in the system for its formation. The absence of one component will prevent this process from occurring. An alternative prevalent model, used for both signaling and regulation pathways, is the sigmoid model (Li et al., 2004).

Fig. 3.2 AND attribute examples. (A) Protein modification; (B) Complex formation.
3.2.2 Conditional edges

A conditional edge description (Fig. 3.3.A) was previously depicted in (Rubinstein et al., 2007). For a given pathway $A$, let $\text{cond}_w(e)$ be a conditional edge between a node $w$ ($w \in V_A$) and an activating edge $e = (u, v)$ ($e \in E_A$). The activation of $v$ by $u$ is conditioned upon the activation of the node $w$ in the system. Such a graphical description of the network can also be used during e.g. a process of protein modification ($u$ is an enzyme and $v$ is a modified protein) and its inhibition by some inhibitor $w$. In this case, the conditional edge is negative, which emphasizes that the inhibitor $w$ prevents $v$'s modification by binding to enzyme $u$ and causing its structural configuration to change. In its current form the enzyme cannot modify the target protein $v$.

3.2.3 External (boundary) conditions

It is well known that distinct cell and tissue types under various environmental conditions have different gene expression profiles, which are the results of activating slightly different pathways (Bowcock and Krueger, 2005; Chi et al., 2006; Lindberg and Kappos, 2006; Liu et al., 2006; Staudt and Brown, 2000). We refer to these (cell and tissue type, cell condition such as starvation, humidity, heat shock etc.) as external (boundary) conditions (Fig. 3.3.B). These factors can affect the presence or absence of some edges in the pathways. Fig. 3.3.B illustrates the case in which an edge $u \rightarrow v$ appears in a pathway that occurs under the external conditions that characterize pathway $A$.

3.2.4 Putative conditions

A "putative conditions" (PC) node (Fig. 3.3.C) represents the reasons for the presence of an edge that are beyond the pathway's scope. One interpretation of a source PC node may be the presence of a protein or mRNA in the original pathway that has a role in regulating the edge to which it points. Such additional proteins or mRNA are missing in pathway descriptions due to: (i) an inappropriate pathway cut-off, (ii) lack of knowledge about this protein and its function in the pathway.
Another possibility for PC interpretations is an edge confidence level. A mediator node may be indicating that the edge in another pathway is missing and that the node label mentions the pathways in which it appears. The user can check whether a pointed regulation is not present in another pathway or if it is missing due to incomplete pathway description.

**Fig. 3.3** Conditional edges. (A) Upstream condition; (B) External (boundary) condition; (C) Putative condition
3.3 Kinds of explanations

During conditional pathway integration, as was mentioned in Section 3.1, a refinement of the results obtained by graph union is performed. Consider the simplest case in which we have a node that is present in both pathways, but some (or all) of its exiting edges are present in only one of the pathways. There are five possible explanations for this difference, denoted: AND attribute, local, upstream, external (boundary) conditions, and putative conditions; all of these are explained in this section.

3.3.1 AND attribute

For two given pathways A and B (Fig. 3.4.A and 3.4.B), an AND attribute explains why an edge from P1 to P3 appears only in pathway A. Based on the AND node definition given above, if one of the proteins under an AND attribute is missing, a target protein will not be produced. This case is presented in Fig. 3.4, where the absence of P2 in pathway B prevents the formation of P3.

![Fig. 3.4 AND attribute explanation.](image)

3.3.2 Local

For two given pathways A and B (Fig. 3.5.A and 3.5.B), let the green and purple colors of P1 represent differences in the node attributes (without loss of generality, let the node attribute be "localization within a cell": green and purple colors represent different cell parts, e.g. cytosol and nucleus). Since the same protein may have various functions in distinctive cell parts, the difference in node attributes can be a possible explanation for the difference in exiting edges.
3.3.3 Upstream

For two given pathways A and B (Fig. 3.6.A/B and 3.7.A/B), there are two possible upstream explanations: positive (Fig. 3.6.C) and negative (Fig. 3.7.C) dependencies. Suppose we are looking for an upstream positive regulation explanation for an edge $e$ ($e = (u, v)$, where $e \in E_A$ and $e \notin E_B$). What we need is to find some node $w$ that activates this edge. Therefore, only when $w$ is present in a pathway, the edge $e$ should appear. Since the edge $e$ is located in pathway A, we need to find a node $w$ (or a set of nodes) in it, located upstream to a node $u$ and missing in pathway B.

In the case where we are interested in an upstream negative regulation explanation, the explanation and search process are very similar to the search for a positive regulation explanation. Here we are looking for a node $w$ (or set of nodes) that does not appear in a pathway A, but does appear in pathway B and is located upstream to a node $u$. 

Fig. 3.6 Upstream (positive) explanations.
3.3.4 Upstream – "Putative Conditions"

The upstream – "putative conditions" explanation (Fig. 3.8) can be considered a sub-type of the "upstream" explanation (Section 3.3.3). Let us say that we need to explain an edge $e$ ($e = (u, v)$, where $e \in E_d$ and $e \notin E_b$)). The PC explanation of Section 3.2.4 is one potential explanation here. However, when we are looking for a positive (or negative) upstream explanation, we cannot always find the node $w$ as described in Section 3.3.3. When this happens, a new node, "PC:PathwayA", is added (in case it has not yet been created) and a positive conditional edge is added from it to the edge $e$.

3.3.5 External (boundary) conditions

The last possible explanation for an edge $e$ ($e = (u, v)$, where $e \in E_d$ and $e \notin E_b$)) is by external (boundary) conditions. As was described in Section 3.2.3, differences in external conditions may cause an activation of various regulations. If such differences exist, a new node, "Ext_cond:PathwayA", is
added (if it has not been created yet) and a positive conditional edge is added
from it to the edge $e$ (Fig. 3.9).

**Fig. 3.9** External conditions explanation.
3.4 The Conditional Merge Algorithm

The operations defined in Section 3.1 are realized by the Conditional Merge algorithm. We first perform the union of both graphs that will serve as the basis for further refinement. The union process detects the nodes that are present in both pathways, but some (or all) of whose exiting edges are present in only one of the pathways; we call these exiting edges "edges that must be explained" (denoted diffEdgeList). Then, according to the parameter mergeType, the algorithm finds appropriate explanations for these differences. There are five possible explanations, denoted: AND attribute, local, upstream, external (boundary) conditions, and putative conditions (see 3.3). The order in which we search for an explanation is defined by the user (see 3.4.2) using the parameter mergeMode. In case an explanation for a specific edge was found, the search for an explanation for this edge is stopped.

The pseudo-code for this algorithm is as follows; some of the support sub-functions are described briefly below.
3.4.1 The algorithm

**Definitions** (used throughout):
- \( \text{out}(v) \) – edges exiting from \( v \) in graph \( G \)
- \( \text{nodeIntersectionList} := V_A \cap V_B \) //a list of nodes that appear in both of the graphs

**edges for explanation:**
- \( \text{diffEdgeListA} := \text{out}(v_A) / \text{out}(v_B) \) //exiting edges from a node \( v \) that are in \( A \) but not in \( B \)
- \( \text{diffEdgeListB} := \text{out}(v_B) / \text{out}(v_A) \) //exiting edges from a node \( v \) that are in \( B \) but not in \( A \)

**isExplain** – a Boolean variable that indicates whether an explanation for edges was found

**currentScanLayer** – the tree layer where we are currently looking for an explanation

**nextScanLayer** – a list of common nodes that are located in the next upstream tree layer relatively to the layer we are scanning now

**visitNodes** – nodes from a currentScanLayer that were have already scanned

---

**Procedure** ConditionalMerge (A, B, mergeType, mergeMode)

---

**Input:**
- Graphs A \((V_A, E_A)\), B \((V_B, E_B)\)
- mergeType: one of {
  - UNION, // a union graph based on nodes/edges names equality
  - Positive Conditional Merge,
  - Negative Conditional Merge,
  - Mixed Conditional Merge
}
- mergeMode: one of {
  - "Upstream",
  - "Local, Upstream, External Conditions",
  - "External Conditions, Local, Upstream"
}

**Output:** Graph G(V,E) // a merge result of A and B

---

G: \( A + B \);

**if** mergeType!= UNION **then**

**for** each \( v \) in \( V_A \cap V_B \)

- let \( v_A \): the node \( v \) in A, \( v_B \): the node \( v \) in B;
- **denote** the exiting edges from node \( v \) in graph \( G \) by \( \text{out}(v_G) \);
- // common edges leaving \( v \) in A but not in B
- \( \text{diffEdgeListA} := \text{out}(v_A) / \text{out}(v_B) \);
- // common edges leaving \( v \) in B but not in A
- \( \text{diffEdgeListB} := \text{out}(v_B) / \text{out}(v_A) \);
- \( \text{removeExplainedEdges}( A, B, \text{diffEdgeListA} ) \);
- \( \text{removeExplainedEdges}( B, A, \text{diffEdgeListB} ) \);
- **if** \( \text{diffEdgeListA} \neq \emptyset \)
  - **then** findExplanation(\( \text{diffEdgeListA} \), \( v_A \), mergeMode);
- **if** \( \text{diffEdgeListB} \neq \emptyset \)
  - **then** findExplanation(\( \text{diffEdgeListB} \), \( v_B \), mergeMode);

**return**(G);
Procedure `findExplanation` (diffEdgeList, currentScanLayer, mergeMode)

Input:  
diffEdgeList – a list of edges that need to be explained  
currentScanLayer – a tree layer where we are currently looking for an explanation

Output:  
graph G with an addition of conditional edges added to edges in diffEdgeList

Description:  
The procedure finds – according to its 3rd parameter – appropriate explanations for the edges that need to be explained and are passed to it as its 1st parameter. There are five possible explanations, denoted: AND attribute, local, upstream, external (boundary) conditions, and putative conditions (see Section 3.3). The order in which we search for an explanation is defined by the user (see Section 3.4.2) using the parameter mergeMode. In case an explanation for a specific edge was found, the search for an explanation for this edge is stopped.

```
for each v in currentScanLayer
  switch (mergeMode)
    case "Upstream":
      isExplain := upstreamExplain (v, diffEdgeList, nextScanLayer);

    case "Local, Upstream, External Conditions":
      isExplain := localExplain (v, diffEdgeList);
      if isExplain == false
        isExplain := upstreamExplain (v, diffEdgeList, nextScanLayer);
      if isExplain == false
        isExplain := externalConditionsExplain (v, diffEdgeList);

    case "External Conditions, Local, Upstream":
      isExplain := externalConditionsExplain (v, diffEdgeList);
      if isExplain == false
        isExplain := localExplain (v, diffEdgeList);
      if isExplain == false
        isExplain := upstreamExplain (v, diffEdgeList, nextScanLayer);

  visitNodes := visitNodes ∪ \{v\};
  currentScanLayer := currentScanLayer \ \{v\};
  if currentScanLayer == Ø && nextScanLayer ≠ Ø
    nextScanLayer := nextScanLayer \ visitNodes;
    currentScanLayer := currentScanLayer ∪ nextScanLayer;
    nextScanLayer := Ø;
  if isExplain == false
    pathwayExplain (diffEdgeList);
```
Procedure `removeExplainedEdges` (A, B, diffEdgeList)

**Input:**
- Graph A(V_A, E_A)
- Graph B(V_B, E_B)
- `diffEdgeList` – a list of edges for whose explanation we are looking in A

**Output:**
- `diffEdgeList` – a list of edges for whose explanation we are looking in A that still do not have an explanation

**Description:**
The procedure removes edges from `diffEdgeList` that have an explanation. Assume we are dealing with an edge \( e = (v,u) \). We observe two kinds of possible explanations for it:
- **AND attribute:** \( u \) is produced if an edge \( e \) is present and some edge \( e' = (v',u) \), i.e. proteins \( v \) and \( v' \) should be present in the system in order to create/activate \( u \). If \( v' \) is missing in B, it explains why the edge \( e \) does not appear in B.
- **Existing conditional edge:** if \( e \) is already explained in A and this explanation suffices, then we leave the explanation the way it is and move to the next edge. We say that a conditional edge \( \text{cond}_w(e) \) (where \( w \) is a mediator node and \( e \) is a target edge) is suitable if:
  - \( \text{cond}_w(e) \) is an activating edge and \( w \) is absent in B
  - \( \text{cond}_w(e) \) is a repressing edge and \( w \) is present in B
In case the conditional edge \( \text{cond}_w(e) \) is not suitable, the algorithm adds it to a list of inconsistent edges which is shown to the user at the end of the algorithm's execution (see Section 3.4.3).
**Procedure** *localExplain* *(v, diffEdgeList)*

**Input:**  
v – a node whose attribute values are compared in both networks  
diffEdgeList – a list of edges for whose explanation we are looking

**Output:**  
true/false – whether there is a difference in a node's attributes. In case  
the return value is true, corresponding conditional edges are added

**Description:**  
The procedure is checking whether there is a difference in node attributes  
(protein state, localization within the cell). In case the difference is present,  
activating conditional edges are added from the node itself to edges in  
diffEdgeList and the search for an explanation is stopped.

**Procedure** *upstreamExplain* *(v, diffEdgeList, nextScanLayer)*

**Input:**  
v – a node for which an attribute value is compared in both networks  
diffEdgeList – a list of edges for whose explanation we are looking  
extScanLayer – a list of nodes that are located in the next upstream  
tree layer

**Output:**  
true/false – whether there is a difference in node attributes. In case the  
return value is true, corresponding edges are added

**Description:**  
The procedure searches for an upstream difference. Suppose that edges in  
diffEdgeList are present in pathway A and not in B. Scanning layer by layer,  
we are looking for sets of nodes *(X, Y)* in currentScanLayer such that nodes in  
X are present in pathway A but not in B and the opposite holds for set Y.  
Activating conditional edges are added from nodes in X to edges in  
diffEdgeList and repressing ones from nodes in Y to those edges. Based on  
mergeType, activating/repressing or both type of edges are added. The  
moment the explanation is found in a closer layer, the explanation search is  
stopped.
Procedure externalConditionsExplain \((v, \text{diffEdgeList})\)

**Input:** \(v\) – a node whose attribute value is compared in both networks  
\(\text{diffEdgeList}\) – a list of edges for whose explanation we are looking

**Output:** true/false – whether there is a difference in a node's attributes. In case the return value is true, corresponding edges are added

**Description:**  
The procedure checks whether there is a difference in the node's external conditions attributes (tissue, cell type, full match in external conditions). In case a difference is present, activating conditional edges are added from a newly created node (\(\text{ExtCond:A,B}\)) to edges in \(\text{diffEdgeList}\) and the explanation search is stopped.

Procedure pathwayExplain \((\text{diffEdgeList})\)

**Input:** \(\text{diffEdgeList}\) – a list of edges for whose explanation we are looking

**Output:** graph \(G\) with an addition of a new node and conditional edges from it to edges in \(\text{diffEdgeList}\)

**Description:**  
This procedure is executed only if an explanation to edges in \(\text{diffEdgeList}\) was not found in the abovementioned procedures. In this procedure a new node (putative condition), "PC:pathwayName", is created and activating conditional edges are added from it to edges in \(\text{diffEdgeList}\). The idea is to point to one of two possible options: 1. some "player" in pathway "pathwayName" is missing and it is actually responsible for the difference in the exiting edge list, or 2. this edge is missing in another pathway.
3.4.2 Merge Mode

For a given edge list, diffEdgeList, that contains edges to be explained, the order in which potential explanations are checked is defined by the merge mode we are in. This order, that is provided as a parameter (that can assume one of several possibilities) to the algorithm, can lead to significantly different merge results. Each of the possible modes reflects a slightly different biological approach: The "Upstream" mode, in which we look only for upstream differences in the graphs, is preferable in cases where we want to suppress system condition effects and are interested in investigating the effect of the pathway's structure. When we are interested in the detection of the differences starting from the node state and localization, i.e. the differences that are the closest to the node we are dealing with, the "Local, Upstream, External Conditions" mode should be chosen. External condition attributes are the last to look at since those signal effects come from outside the cell. The last mode, "External Conditions, Local, Upstream", is pretty similar to the previous one besides the assumption that the external condition attributes should be examined first. The intuition is that it is meaningless to merge pathways that occur in different external conditions.

3.4.3 Inconsistency between edges

For a given pathway $A$, let $\text{cond}_w(e)$ be a conditional edge between a node $w \in V_A$ and an edge $e=(u,v)$, (where $e \in E_A$). During a conditional merge, we evaluate whether a conditional edge is consistent with the given pathways.
This evaluation is based on the following truth-table:

<table>
<thead>
<tr>
<th>Case</th>
<th>X: Regulation type in A</th>
<th>Y: ( e \in V_B )</th>
<th>Z: ( w \in V_B )</th>
<th>XOR(X,Y,Z)</th>
<th>whether consistent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 3.1 Evaluation of conditional edge.

Legend: In column X (regulation type) "+" means positive, "-" means negative; in columns Y, Z "+" means that the membership predicate holds and "-" means that it does not hold.

Intuitively, when the regulation type is positive (denoted by +) the meaning is that \( w \) stimulates a reaction from \( u \) to \( v \), i.e. in case \( w \) is inactive the edge from \( u \) to \( v \) will have no effect. The opposite is valid for a negative (devoted by -) type of regulation. Therefore, in Case 1, since we have positive regulation and \( w \) is present in Pathway B, it is reasonable that the edge \( e \) will be present as well. However, in Case 5, \( w \) that represses \( e \) is present, therefore \( e \) should not be present, contrary to what we have in the pathway and therefore it is a case of inconsistency. All the other cases can be analyzed in a similar fashion. Moreover, it is evident from the table that this logic can be expressed mathematically by the XOR operator.
3.5 Properties of the Conditional Pathway Algebra

3.5.1 Closure

A set is closed under an operation if when applied to any members of the set the operation returns a value that is a member of the same set. Similarly, a set is closed under a collection of operations if it is closed under each of the operations individually.

In our case, the system under consideration comprises graphs and the operations defined above. Given \( G(V,E) \), where \( V = V_b \cup V_e \cup V_p \) and \( E = E_{\text{comm}} \cup E_{\text{cond}} \), we want to show that after performing each of the following operations the result is contained in the same set of graphs.

First we consider Graph Union, namely \( G = A + B = (V_A \cup V_B, E_A \cup E_B) \). As can be seen, we start with a set of nodes and edges within the defined set, and remain with the same nodes and edges; therefore the union graph is also in the same set.

Next we show that the **positive conditional merge** operation \( (\odot) \) preserves the closure property. This property holds similarly for the other two conditional merge operations, with a proof that goes along similar lines. In order to prove closure of the merge operation, we will follow the algorithm’s steps that modify the graph and show that its operations preserve closure:

- Union of two graphs preserves closure.
- Calculating the node set intersection \( V_A \cap V_B \) and for each node in it performing \( \text{out}(v_A)/\text{out}(v_B) \) and \( \text{out}(v_B)/\text{out}(v_A) \) (\( \text{diffEdgeListA} \) and \( \text{diffEdgeListB} \), resp.): both operations do not change either of the graphs so the resulting graphs are still in the original graph set.
- For each edge \( e = (u_b,v_b) \) in \( \text{diffEdgeListA} \) and \( \text{diffEdgeListB} \), we are looking for an explanation at each of the four possible levels and add corresponding conditional edges. Note that added conditional edges point only to common edges. We now show that all possible conditional edges that can be added are within our set. During this operation, activating conditional edges are the only edge type we add.
- Local: conditional edges from the node itself to an edge coming out of it, \( i.e. \) a conditional edge \( \text{cond}_u(e) \) where \( e=(u,v) \) and \( \text{cond}_u(e) \in E \).

- Upstream: conditional edges \( \text{cond}_w(e) \) in which the source node \( w \in V_b \) can be added and it is also within the set \( E \).

- External conditions: in order to add an explanation of this type, first we create a source node of type external conditions \( (w_\theta) \) and then draw a conditional edge from it to the edge \( e \) \( (w_\theta \rightarrow (u_b,v_b)) \). Also in this case we end with members in sets \( V \) and \( E \).

- Putative conditions: this type of explanation is added in case we did not find an appropriate one among all the other possible explanations. Here we create a new node \( (w_p) \) of \( V_p \) and add an edge \( w_p \rightarrow (u_b,v_b) \). Both the node and the edge are within the set of nodes and edges we defined.

We have shown that nodes which are added as well as conditional edges are within our domain. Therefore, since the graph that is returned is in \( G \), merge operations preserve closure. Since the closure of all operations holds the set we defined is closed under all the operations defined above.

### 3.5.2. Commutativity

For any two graphs \( G_1, G_2 \) that are members of the set of conditional graphs we want to show that commutative property holds. Again, we first consider \textit{Graph Union}, \( G_1 + G_2 = G_2 + G_1 \). During this operation we create a new graph that is composed of nodes and edges from two graphs. The order of the graphs determines the order in which nodes and edges are scanned and added to the new graph. At the end of the scanning process, we receive the same sets in both of scan orders. Therefore, the commutative property of \textit{Graph Union} holds.

Once again a similar proof outline will be valid for the three conditional merge operations. After performing the commutative \textit{Graph Union} operation, we calculate the node intersection set, and for each node in it the exiting edges difference set. Those operations are performed symmetrically on both
of the graphs. Afterwards, the procedure that looks for an explanation is called for both of the graphs; hence the commutative property is preserved.

3.5.3. Associativity
For any three given graphs $G_1$, $G_2$, and $G_3$ within our set, the associative property would guarantee that: $(G_1 \otimes G_2) \otimes G_3 = G_1 \otimes (G_2 \otimes G_3)$, where $\otimes$ is one of our four defined operations.

Unfortunately, the associative property holds only for the Graph Union operation, not for the rest of the operations. When our merge algorithm encounters inconsistency in the explanations, it allows for the user's intervention or leaves the previous explanation. Therefore, the result of the second operation depends on condition edges that were previously added as well as user intervention.
Chapter 4: The Pathway Integration Environment - PIE

PIE is a software tool that realizes our conditional pathway algebra. It was implemented in JAVA as a plugin for Cytoscape, a free bioinformatics software platform for visualizing molecular interaction pathways and integrating these interactions with gene expression profiles and other static data.

4.1 Cytoscape

Cytoscape is a general purpose, open-source software environment for visualizing and working on biological networks. The central abstraction of a biological network is a labeled graph, with biological entities represented as nodes and interactions as edges between nodes. Cytoscape's core software provides basic functionalities for assigning attributes to graphs' nodes and edges, a visual representation of the graph and its attributes, and selection and filtering tools. Moreover, Cytoscape's core is extensible through plug-in architecture, allowing development of additional computational analyses and features.

Fig. 4.1 Cytoscape website http://www.cytoscape.org.
Cytoscape’s features:

- **Network import / export**
  Cytoscape provides a very easy-to-use method for network importing. It is well known that there is no one standard data format for storing biological networks. Usually, each pathway database stores data in its own format, commonly, an XML-based format. Cytoscape supports several of the most useful data formats: BioPAX (http://www.biopax.org/), SBML (System Biology Makeup Language) (Hucka et al., 2003), GML (Graph Modeling Language) and XGMML (eXtensible Graph Markup and Modeling Language) (Punin and Krishnamoorthy, 1999) http://www.cs.rpi.edu/~puninj/XGMML). Cytoscape also supports two table formats: a simple interaction format (SIF) (a text file with the .sif extension), and Excel (.xls). In these formats a user has to specify the source interaction (start node), interaction type and target interaction (end node).

  In addition, a user can retrieve pathways directly from web services. This option appeared in the last version of Cytoscape (2.6.0), but is still far from perfect.

  Although export options are much more limited that import ones, Cytoscape does support exporting in several very commonly used formats: BioPAX, XGMML, GML and SIF. A user also can export networks as pictures.

- **Network drawing and editing**
  Not less important in fluent work with biological networks is the ability to add and/or edit information that was stored in files describing specific pathways and import it to Cytoscape. Cytoscape enables both drawing new networks from scratch and editing an open network.

- **Network layout**
  One of the most basic tools for understanding biological networks is their visualization as a two-dimensional network. Cytoscape supports a variety of automated network layout algorithms, including spring-
embedded layout (the mostly widely used method for arranging general
two-dimensional graphs), hierarchical layout, and circular layout.

- **Network selection and filtering**
  To focus our attention on a specific sub-network, it must be selectively
displayed. Nodes and edges may be selected according to a wide
variety of criteria, including selection by name, by list of names, or on
the basis of some attribute. More complex network selection queries
are supported by a filtering toolbox that includes built-in filters and an
option to build new ones. Several selections can be joined together
using AND, OR and other Boolean functions.

- **Attribute-to-visual mapping: VizMapper**
  Whereas the layout determines the nodes' and edges' location on a
screen, an attribute-to-visual mapping allows data attributes to control
the appearance of their associated nodes and edges. Cytoscape
supports a wide variety of visual properties, such as node color, shape,
and size; edge color, thickness, and style.

To summarize, PIE benefited from Cytoscape by taking its functionalities,
interface and underlying software, expanding them and adding new
functionalities to be detailed later. By using Cytoscape a user can pipeline
results of several plugins and everything is done in the same working
environment. Moreover, the large Cytoscape community and efficient forum,
which is critical for such kind of an environment, offers users possible
solutions for problems that might appear while working in/with Cytoscape.
4.2 PIE functionality

We can divide PIE’s functionality into two, as follows:

1. *Performing the Union and Merge operations*: these operations were defined earlier in Section 3.1 and are implemented in PIE.

2. *Enriching pathways with known data*:

   - **Attributes’ data entry**: Different conditions are stored in pathways, nodes, and edges as attributes. Whereas cell type, tissue, and physiological conditions (such as starvation and heat shock) that can be additionally assigned to a specified sub-pathway are more appropriate to be referred to as pathway attributes, protein localization and activation state (phosphorylation, ubiquitination, glycozilation, etc.) are naturally represented as node attributes. Since PIE is implemented as a plug-in to Cytoscape, we could use an existing attributes platform. However, in the current version of Cytoscape an attribute value is assigned to all the proteins with the same id in all opened pathways. Sometimes we are interested in assigning different attribute values to the same protein in different pathways. Since it is critical, we implement this feature differently allowing a shorter and more intuitive way to assign attributes to nodes.

   - **Adding conditional edges**: Known dependencies that are expressed by conditional edges can be manually entered using the Cytoscape editor (which was enhanced by the HyperEdgeEditor plug-in to support this feature). Those conditional edges can utter conditional regulation on protein state, and localization within the cell by adding an activation or repressing edge from the node itself to one of its exiting edges. Similarly, by adding edges from another node to an edge that represents regulation we want to impose a condition on it. In addition, there is a possibility to add additional activating or repressing mediators to an already conditioned edge. Note that a user can add another conditional edge in which a source node points to an existing conditional edge. Our algorithm will not
generate such a situation; it can, however, work with such a pathway as an input.

- **Adding an AND attribute to a node:** Not less important is the ability to emphasize that for the creation of some proteins, complexes, mRNAs, etc. there is a need in simultaneous regulation of several proteins, enzymes, etc. This property is implemented by PIE as an AND attribute of the node that is conditioned on others.

To summarize, PIE enables the representation of conditioning regulation by one of the following: a) external conditions which are edited from the plug-in and appropriate conditioning is received after performing one of the conditional merge operations; b) conditioning protein production or activation by the presence of other proteins in the system (the AND node attribute); c) missing information which is derived from the result of one of the conditional merge operations; and d) conditional edges in which activating or repressing a mediator can be a node itself (e.g. its state, localization) or some other upstream proteins.
Chapter 5: Biological Validation

To validate our method we applied it to three well characterized *Saccharomyces cerevisiae* (yeast) pathways: the Filamentous Growth, Pheromone Signaling, and High Osmolarity Glycerol (HOG) pathways as they are described in the Science magazine cell signaling database STKE (http://stke.sciencemag.org/cgi/collection/pw_algae_and_fungi). Yeast is an important model system for eukaryotic organisms, and the selected pathways represent well-studied and highly curated biological functions, allowing us to evaluate our *in silico* predictions: the pathways (as described in the rest of this section) each comprise a small and simple subsystem, but they are strongly interrelated (as reflected in the descriptions as well), so it is possible to check the predictions that were obtained when integrating them. Still, each STKE pathway entry had to be corrected to reflect later updates in the literature and some curation errors; these changes are specified herein for each of the cases under network correction.
5.1 Pathways under study

- Filamentous Growth Pathway
  http://stke.sciencemag.org/cgi/cm/stkecm;CMP_14554 – Fig. 5.1

In response to nitrogen starvation and other signals, diploid a/α yeast cells undergo a developmental change and switch to a filamentous form of growth called pseudohyphal development. This transition includes cell elongation, a switch to a unipolar budding pattern, maintenance of the attachment between mother and daughter cells, and the consequent ability to invade semisolid media. This morphological change is likely to cause a foraging response that allows cells to scavenge for nutrients. In haploid cells this switch is assumed to occur in response to glucose starvation and is termed haploid invasive growth (Gustin et al., 1998; Stanhill et al., 1999).

*Network correction:* The database entry includes the Fus3 repressing Tec1 edge which is not functionally related to this network; it exists only when the pheromone pathway is active as one way to represent cross-talk between the filamentous growth and pheromone signaling pathways (Bao et al., 2004; Chou et al., 2004). The authors probably added this edge to the network to show a broader view of regulation for this pathway. We, however, thought that moving this edge to the pheromone signaling pathway (Fig. 5.3.A) will provide a more correct view of the networks.

Another correction was made with the Ste11 and Ste50 nodes. All three pathways under study include these nodes; in two of the networks (this one and the pheromone signaling pathway, described below) the edge between these nodes is described as activation, whereas in the HOG pathway (also described below) it appears as neutral. In the literature Ste50p and Ste11p are described to interact constitutively via their N-terminal regions, which include putative SAM domains (Grimshaw et al., 2004; Jansen et al., 2001; Posas et al., 1998; Ramezani-Rad, 2003). This description also appears in this network and in the HOG pathway. Considering all the information above, we decided to accept the neutral form of the edge (Fig. 5.3.B).

Overall this network included 32 nodes and 40 edges.
Fig. 5.1 Filamentous growth pathway from STKE entry.
• **Pheromone Signaling Pathway**

http://stke.sciencemag.org/cgi/cm/stkecm;CMP_13999 – Fig. 5.2

Yeast cells can exist as either haploid or diploid cells. Haploid cells of the opposite mating type (a or α) can mate, i.e., fuse and form a diploid. This process is stimulated by the release of small peptide mating pheromones, α-factors from MATa cells and α-factors from MATα cells, which act on cells of the opposite mating type to prepare these cells for mating. Cellular responses to mating pheromones include: arrest in the G1 phase of the cell-cycle, oriented growth towards the mating partner, and – ultimately – fusion of the plasma membranes of the mating partners, followed shortly thereafter by the fusion of their nuclei (Bardwell, 2005; Gustin et al., 1998).

_Network correction:_ As explained above, we decided to move the Tec1 node and the Fus3 repressing Tec1 edge from the filamentous growth pathway to the pheromone pathway. We also changed the edge between the nodes Ste11 and Ste50 to be neutral as in the filamentous growth pathway (Fig. 5.3.B, text above). Furthermore, in both pathways, filamentous growth and pheromone signaling, Dig1 and Dig2 inhibit Ste12 (Gustin et al., 1998). In the filamentous growth network Dig1/2 is repressed by Kss1, while in the pheromone signaling pathway Dig1/2 is activated by Fus3. Nevertheless, based on the explanation given by the authors of both networks in STKE and also according to literature (Bardwell, 2005; Gustin et al., 1998; Stanhill et al., 1999), the regulation of Dig1/2 in the two pathways is the same, but still the pheromone pathway authors interpreted it as activating whereas the filamentous growth authors interpreted it as repression. After looking at the literature per above we decided to use repression in both cases (Fig. 5.3.C). Overall this network included 32 nodes and 46 edges.
Fig. 5.2 Pheromone signaling pathway from STKE entry.

Fig. 5.3 Network corrections for the pathways under study.
High Osmolarity Glycerol (HOG) Pathway

http://stke.sciencemag.org/cgi/cm/stkecm;CMP_14620 – Fig. 5.4

The internal osmolarity of a growing yeast cell is maintained to be higher than the external osmolarity. Increasing external osmolarity is a commonly encountered stress for a yeast cell in various natural environments such as a split-open grape drying under the sun, a Petri dish left open in the incubator, or the start of fermentation when sugar is added. The high osmolarity glycerol (HOG) MAPK pathway is activated by an increased environmental osmolarity and results in a rise of the cellular glycerol concentration so as to adapt the intracellular osmotic pressure (Dihazi et al., 2004; Gustin et al., 1998).

Network correction: No changes were necessary for this network. Overall this network included 29 nodes 31 edges.

Fig. 5.4 High Osmolarity Glycerol (HOG) pathway from STKE entry.
5.2 Results and Discussion

We defined a conditional pathway algebra that safely extends the traditional graph theoretical-based pathway description model to include e.g. protein localization and external conditions in which pathways are activated, and take these factors into consideration during pathway integration. Not only is it the case that no information is lost during the integration process, but rather new information regarding either of the pathways and about crosstalk between them – as well as possible effects of some proteins on specific regulations – can be generated.

PIE is a bioinformatic tool that implements the formal algebra and the integration algorithm. It was applied successfully to several cases, as described in this section. Moreover, it proved to be an excellent research tool: each of the conditional edges that were generated by the conditional merge algorithm contains important information about the relationships between the involved pathways, the reliability of the edges in their original description, or lack of pertinent information in it. Each conditional edge can then be validated by performing wet experiments or a literature search. In the integration experiments that we performed and report herein, no contradictions were detected between the literature and conditional edges that were added to the graphs. Moreover, in one of the cases, when a contradiction seemed to appear, we found a more recent paper that corrected it.
• **Pheromone Signaling pathway** ☰ **Filamentous Growth pathway**

The resulting network (see Fig. 5.5) includes 62 nodes, 9 of which are common. Out of 105 edges, only 9 edges have to be explained and this is done by adding merely 12 conditional edges. Still, each and every one of these edges reflects either some biological observation or lack of important data in the original description, as explained next.

![Fig. 5.5 Positive Conditional Merge of the pheromone signaling (in green) and filamentous growth (in red) pathways. Nodes in the intersection of the pathways appear in purple.](image)

The filamentous growth network is composed of at least three signaling modalities that control the switch from budding to filamentous growth in *S. cerevisiae* (Truckses et al., 2004). The core of the main pathway is a three-tiered mitogen-activated protein kinase (MAPK) cascade (Truckses et al., 2004). This cascade shares multiple components with the pheromone signaling pathway that uses almost the same MAPK cascade, while the other two modalities do not show any overlap with the pheromone signaling pathway ((Gustin et al., 1998; Sabbagh et al., 2001) and Fig. 5.5). In fact, these two pathways are an excellent example for the case in which two pathways that are quite different in their original composition converge to one
unit centered around an almost identical subcircuit that in itself is highly conserved in evolution. The main difference between the two pathways is that they use two different MAPKs as the target of the phosphorylation cascade: Kss1 in the filamentous growth pathway and Fus3 in the pheromone signaling one (Schwartz and Madhani, 2004; Truckses et al., 2004). In the pheromone signaling pathway, Fus3 – together with the MAPKK Ste7 and the MAPKKK Ste11 – is bound to Ste5, a scaffold protein that is pheromone signaling pathway specific (Schwartz and Madhani, 2004; Truckses et al., 2004). In addition, Fus3 – in response to a mating signal – is specifically down regulating Tec1, a filamentous growth pathway specific transcription factor (Chou et al., 2004). In the end of the MAPK filamentous growth pathway, Tec1 forms a heterodimer with another transcription factor, Ste12, to mediate various gene expression responses (Gustin et al., 1998; Stanhill et al., 1999). In the pheromone signaling pathway, Tec1 is down regulated and two Ste12 molecules join together to form a homodimer that induces or represses genes that are required for successful mating (Bardwell, 2005; Gustin et al., 1998).

The conditional merge algorithm succeeded in pinpointing the right players that are specific to one of the pathways, meaning: Ste5 and Fus3 in the pheromone signaling pathway and Kss1 and Tec1 in the filamentous growth pathway. We also expected that the algorithm would identify other pathway specific components that are more upstream to the MAPK core of the pathways, and indeed the Ste20 to Bmh1/2 edges are examples for such components (Fig. 5.6.A): Bmh1 and Bmh2 are two genes in yeast that show strong similarity to the 14-3-3 proteins (acidic dimeric molecules that likely play a role in signal transduction). Bmh1 and Bmh2 – when associated with Ste20 – are required for filamentous growth but not for the pheromone signaling (Roberts et al., 1997). Ras2 and Msb2 are the two Ste20 upstream components of the filamentous growth pathway that are specific to this pathway ((Truckses et al., 2004) and Fig. 5.5). The algorithm correctly identified this relationship.

The conditional merge algorithm also identified the proteins that are specific to only one of the pathways and that will be used as explanations for the edges that must be explained. The edge from Ste7 to Kss1 (Fig. 5.6.B) is a good example for this: as explained previously, one of the main differences
between the filamentous growth and the pheromone signaling pathways is the use of two different MAPKs, Kss1 and Fus3. As described in Section 5.1, Kss1 is used only in the filamentous growth pathway; furthermore, Ras2 and Msb2 are the inputs for the filamentous growth pathway (Truckses et al., 2004) and Fig. 5.5). The algorithm recognized that Kss1 is unique to the filamentous growth pathway and inferred that Ras2 and Msb2 are responsible for this edge. Moreover, the algorithm correctly connected between the filamentous growth pathway Kss1 MAPK and this pathway's unique output Tec1 and Flo11 (the edges from Ste12 to Tec1 and from Ste12 to Flo11; see Fig. 5.6.C).

Fig. 5.6 Details of the Positive Conditional Merge of the pheromone signaling and filamentous growth pathways (excerpt from Fig. 5.5).

An additional feature of our algorithm is to identify edges with low confidence or incomplete data. The Cdc42 and Ste50 nodes appear in all of the three pathways, but this edge exists only in the filamentous growth pathway (Fig. 5.6.B). The algorithm found this conflict and marked it for further questioning. Although this edge exists in the filamentous growth pathway, it appears there with low confidence. The three networks we used for our analysis were last update in 2005. Interestingly, a more recent paper reports
that this edge exists in the HOG pathway as well (Tatebayashi et al., 2006).
Apparently the data for this edge is not complete; the algorithm recognized
this fact and used the available data to call Ras2 and Msb2 as the best
explanations for this edge using information that is captured in those
pathways.

Finally, an important property of our algorithm is the ability to use different
merge operations (activation, repression, or both repression and activation).
The usage of these operations may provide varying perspectives of the
networks and highlight interesting nodes. When choosing conditional
activation/repression, we can obtain different graphs (Fig. 5.7). In case of the
filamentous growth and pheromone signaling pathways, on the edge Ste7 to
MAPK (Fig. 5.6.D, 5.6.E) we get two different MAPKs and explanations, which
are both reasonable. The algorithm explains the edge by the nearest
explanation. After examination of the results of different merge operations for
the networks we worked with, we concluded that the most appropriate
operation was indeed "conditional merge activation".

**Fig. 5.7** Negative Conditional Merge of the pheromone signaling and filamentous growth
pathways. The color coding is as in Fig. 5.5.
• **Pheromone Signaling pathway + HOG pathway**

The pheromone signaling pathway and the HOG pathway share some components and regulation, but this commonality is less than what we found between the pheromone signaling and filamentous growth pathways (Dihazi et al., 2004; Dohlman and Thorner, 2001; Gustin et al., 1998) and Fig. 5.8). Thus, in their integration we can see more examples of "Putative Conditions" *(e.g. PC:Pheromone Signaling and PC:HOG).*

The obtained network includes 63 nodes, 6 of which are common. Out of 105 edges, only 6 edges have to be explained and it is done by 7 conditional edges.

![Fig. 5.8 Positive Conditional Merge of the pheromone signaling (in red) and HOG (in green) pathways. Nodes in the intersection of the pathways appear in purple, and Putative Condition (PC) nodes are in white.](image)

Notably, here the algorithm identified pathway specific components that are more upstream, for example in the case of the Ste11 to Pbs2 edge (Fig. 5.9.A) the algorithm detected that the Ste11 node is present in both pathways but the Pbs2 node is present only in the HOG pathway (Fig. 5.8). As
explained in Section 3.3, the algorithm looks upstream to find an appropriate explanation to this edge, and it found that the unique elements of the HOG pathway that can be responsible for this edge are Msb2 and Sho1. When we looked in the literature we found that indeed these proteins were associated with Pbs2 activation through Ste11 (Cullen et al., 2004; Posas and Saito, 1997).

A similar situation (one node present in both pathways and another that is unique to only one pathway) is manifested in two edges: Ptp2/3 to Fus3 and Ptp2/3 to Hog1. There is, however, one important difference compared to the previous situation: whereas then the algorithm had upstream components to resolve the conflict with, now there are no upstream nodes to explain the conflict (Fig. 5.8). In this situation the algorithm adds a PC node and an edge from it to the pathway's original edge (Fig. 5.9.B); this unknown node is a sign to the user that he or she needs to find some other explanation to the conflict.

![Diagram](image)

**Fig. 5.9** Details of the *Positive Conditional Merge* of the pheromone signaling and HOG pathways (excerpts from Fig. 5.8).
• **Filamentous Growth pathway** + **Pheromone Signaling pathway** + **HOG pathway**

When merging three pathways the situation is more complicated since inconsistencies between some of the edges can appear. Furthermore, the outcome may depend on the order of the merging, due to the lack of associativity. On the other hand, we may obtain a more accurate explanation after adding the 3rd pathway for an edge for which we had a poor explanation when merging only two of the three pathways. For example, if we merge the three pathways in this order: (Filamentous Growth + Pheromone Signaling) + HOG, we get the conflict shown in Fig. 5.10.A. If, however, we perform the merging in another order: Filamentous Growth + (Pheromone Signaling + HOG), we get a new edge with one explanation (Fig. 5.10.B). This, incidentally, supports Ras2 as being the better explanation.

![Fig. 5.10](image)

**Fig. 5.10** Inconsistent edges during the *Positive Conditional Merge* of the filamentous growth, pheromone signaling, and HOG pathways

Let us consider the results of the first ordering, namely (Filamentous Growth + Pheromone Signaling) + HOG. All new regulations that were added to the (Filamentous Growth + Pheromone Signaling) pathway after merging it with the HOG pathway were the same as those observed and explained for the (Pheromone Signaling + HOG) merger (described above). This result is not surprising since the filamentous growth and the HOG pathways hardly share any proteins. No additional regulations were received also in the case of
the other merging order, namely Filamentous Growth (Pheromone Signaling ⊕ HOG), for the same reason as above.

Fig. 5.11 Inconsistent edges during the **Positive Conditional Merge** of the pheromone signaling, HOG, and filamentous growth pathways.

Note, however, that as the number of merged pathways increases (more than two), the results could be more cluttered and the user may get new inconsistent explanations (as exemplified in Figs. 5.10 and 5.11). In this situation, the user must use the algorithm with care and to decide – based on other sources of information – what the best explanation would be.
Chapter 6: Summary and Future work

PIE is a powerful tool for combining data from different sources that describe pathways based on experiments in a variety of conditions. It is based on the conditional pathway algebra that we have defined, enabling its users to enrich biological pathway representation with knowledge that comes from the experimental conditions that were used and from previous studies. It also allows systems biologists to add to the pathways' representations important information that has up to now been described informally (in words), such as "this regulation occurs only if a certain protein is phosphorylated and is located in the cytoplasm or is conditioned on the presence/absence of some other protein in the system". In other cases, we are also able now to express the notion that for activating some protein X, there is a need in simultaneous co-regulation of three other proteins, X1, X2, and X3. Moreover, PIE can point out interactions that are conditioned by specific regulators (presence/absence of proteins, co-regulation, extracellular factor etc.); using simple graph union we would miss these issues, with no way for reconstruction. Finally, PIE can be used not only for safe information integration; it can also be leveraged as a research tool. During pathway integration, the user – using different modes of merging – can see both "core" differences between pathways as well as edges with low confidence or without enough support information. Focusing on and further investigation of these differences can enhance our understanding of the biological systems under study.

A natural next step would be the integration of pathways of different types, e.g. signaling, regulation, and metabolic pathways that all pertain to the same biological function, into one framework. This would lead us to deeper understanding of biological systems as a whole.

Finally there is the technical challenge of dealing with pathways that are represented in different, divergent file formats that are being used for pathway retrieval. These are both XML-based representations, such as BioPAX (http://www.biopax.org/), SBML (Hucka et al., 2003), KGML
(http://www.genome.jp/kegg/xml/), and XGMML (Punin and Krishnamoorthy, 1999), as well as text formats, such as SIF. Platforms like Cytoscape are making progress towards the convergence of this issue.
Bibliography


Sabbagh, W., Jr., Flatauer, L.J., Bardwell, A.J. and Bardwell, L. (2001) Specificity of MAP kinase signaling in yeast differentiation involves transient versus sustained MAPK activation, Mol Cell, 8, 683-691.


Web Site References

STKE (Signal Transduction Knowledge Environment): http://stke.sciencemag.org/

BioCarta: http://www.biocarta.com/

The Signaling Gateway: http://www.signaling-gateway.org/


BioPAX (Biological Pathways Exchange): http://www.biopax.org/


STKE (Signal Transduction Knowledge Environment): http://stke.sciencemag.org/

Filamentous growth pathways:
http://stke.sciencemag.org/cgi/cm/stkecm;CMP_14554

Pheromone signaling pathway:
http://stke.sciencemag.org/cgi/cm/stkecm;CMP_13999

High osmolarity glycerol pathway:
http://stke.sciencemag.org/cgi/cm/stkecm;CMP_14620
הפעולה אוות על שלושה מוסלים לתועבת סיגנל בשר רוב ש_signals מודל

של אוקריוטיס. המוסלים נבחניםUNKならない, משייט פשטים ושילמי חלקיים מושתפים

בינים. LAN האור בעדック שלו המוסלים על סמי מתרקים שישעם LAN📦 לחזון בתמון, הפעולה

את הכל, שלום עליה. כל התחדויות שנוספו ולאומת בסופר.
אלגוריתם מנסה למצוא הסבר לעובדה ה-בשלב הבא. הקיימות רק באחד מהגרפים הייצוגיים לקשתות שצריך להסביר – מסלול בודד באחת מהקשתות קיימת רק שכל הסברים אפשריים (אופרטור: AND ו- OR) תנאים, "של צומת מאפיינים" תנאים פנימיים (כפי שיוגדר בהמשך). תנאים משוערים (מאפיינים של המערכת: אנימהו Ning ו- מהלך), תכניות ושעת חלבון בין צומת המסלולים שלグラף

** yakınימי פסימיס**: מאפייני החלבון: חסר החלבון ממתGametes בחר בלט בנצוגים מתאימים

ובמקרה כל נספח מתאם, אינטראקציה המבוססת אופרטור. מבקר ו-4 חסר את המאפים

יהיו שואלי וברש"ץ. מנן, נספח לא הני 오ירטקדציה החלבון שיוו:/power של החלבון V

לresponseData לשחזר, "קשתות שבין צומת מתאימים לשחזר קוסמולוגיות לא מספקים" לא מתייחסות.

**.closest קוסמולוגיות:** קוסמולוגיות ונמצאות במערכת במעלה עבורה

**nearest הני קוסמולוגיות: נמצאות במערכת ו- כחיוניים של הרשת. חסר הסבר הדבורה ולייה花纹ו

שشؤף מבקר ו- הערכה של צומת המחשה ומקוה של צומת המחשה בשתי מחלקות אופרטורים: הבננאות

ובמקורה, זה ויור מצו צומת מסוב שמחושב וול יסודות את התאמה האהל וקשת

התנאים ייקוט مليون המחלקה מ-not קוסמולוגיות

**"הלא עמוד" במקורה לא מציא חסרavern המשטחים של החלבון ומקוה, מסוב זה מפר הערכה: putative

ואופרטור החזקה לא ניתן להמצא את התאמה האהל אופרטור

**conditiונים (ぁוחה האופרטור הכי משה)-** במקורה לא מציא חסרavern המשטחים של החלבון ומקוה, מסוב זה מפר הערכה: putative

**עטרת בעיקר על החלבון מסובים בתוכנות:ologi.**

**Cytoscape,** תשתית תוכנה להצגת אינטראקציות

**Pathway Integration (PIE).**
There is integration throughout, hence...

...hence, the considerations of pruned trees, which are received must be taken into account. Points must be considered from the model, pathways for signal or metabolic pathways are to be included in, usually... the results are shown in the... The results also show differences... The difference in the pathways is not as... Other biologists are considered as... The connections between the regular model are represented by... Comm (Rubinstein et al., 2007) is used to show the connections... The operations that we defined maintain closure and commutativity allows for theoretical properties... A central algorithm in the field of biology is to focus on a specific vertex... We search for connected vertices in the pathways of the input... And focus only on the edges of each graph...

The operations are defined as follows: (1) Pathway composition (or Merged paths) (2) Pathway union (3) Positive regulation (4) Negative regulation... The operations that we defined maintain closure and commutativity allows for theoretical properties... A central algorithm in the field of biology is to focus on a specific vertex... We search for connected vertices in the pathways of the input... And focus only on the edges of each graph...
תקציר

מספר רב של רישוט ומחלולים בנוילוגים התיאורים ההליכים מורכבים בתוכם חתך מאוחדים במוכנים ואנליזים. מספר הד_mpiוד הಯיצ掣ים פונקציונליים והisNewו רמותuggage אבסטרקציה (למשל: מטרולוזה, הינברג סינגל וורוניצי). חסר כח 앞 של המסלול ההקליקי על התהליך הבינוגי את התקנים, תיאורים של מחלולים אחרים מאוחדים במוספרים מראים, גורם ביחסים ופריטים, 컬ון

לסקירה. הנון, הרבח מאמנים מספר מציאותו הצהיר, והתאצורה של סקירה-מספר ספליצי. און מסלול יוני החידות ובמספרים מקרות

במידה,.createSequentialGroup, אלה מחרץ את התאצורה, כאשר (перед עט עם ורגולציה, למשל מטבוליזם) אבסטרקציה, מאגרי מידע במספר אלה מאוחזרים. תיאורים של מסלולים. אותו חוקרים בוancialי התהליך על, פרטיים הןו ציבוריים הן

(וVihinen, 2003 (למאמר מתאר ככאשר, מידע נמצא בספרות הרבה מאוד, בנוסף, לסקירה מקורותבמספראותו מסלול יכול להימצא. ת ספציפי, מסלול pastors philanthropic המאגד את זהל מספר מסלולים המתארים תהליכים דומים או משלימים עכולי

Ծ, חוקרים במדעי, לתרופהלהבין תגובה על מנת, כך למשל, לחוקריםתבלתי אפשרי

הינו  הראשון האופן, אופנים ניבש גהמיזוהליך תאת לבצע.Win הם היו  והביולוגיה מולקולרית. לא אחד מהם 느לחת על (באמצעות ל, ודומים הבדלים בין מסלוליםאת הלראות לה помогה ויכהייתה מסוג זה אינטגרציה ( או תת מסוימתנטראקציהאי תנאים היה ניתן להסיק באילו( מסלוליםהניסוי של הסמך תנאי

אינומקובל היום לתיאור מאגלו ביולוגיים הגרפי התיאור שהאקדים ואנה בצ ימי מופעל מסלול

היאנו, ולצלו את מסלולים תוך שמירה על מידע שלל אינטגרציה.אנו מ מציעים שיטה בעבודה זו. הבולים ביניהםבטא ל

ידוע . סלוליםממספר אינטגרציה של המהלך ביולוגיים וטכניים באתגרים לא מעט קיימים

ונים הם שתאים שונים מרקמות שונות בתנאי סביבה שונים מראים פרופיל גנטיים ש

אקטיבציה המצב לע והןיש השפעה של פקטור מד multipart צמחים. שוניםמעט  מסלולים תתוצאה של הפעל,שוניםהעקביות בין מאגרי המידע החוסר את לשכוח אין. מיקומו בתוך התא לע והןשל חלבון כתוצאה ( לעתים קרובות והעובדה ש, במקורות שוניםאותו מסלול מתואר אחרתהתקרה בו למשל כתוצאה . בצורה אחרתמסלולים(לתתי מחולקיםמסלולים ( ות טכניות של אכסון או ייצוגגבלה

לאבל הבולטים ביניהם.
ברצוני להודות ללהדך שליל, פרופ"י ורד פינטר על הנחייתו, וערדה טכנית מנכילה, על אכפתיה של.

הרבת תודעות בלבלוולי, ויר, על עזרתי המדענית, על עדדות, על כל שמתמי
האמני ביתוח זכרי
אנני רצה להודות לאחרונים אובל אל פיתוח חשבים, להורי האוהבים, קונסטנטין
וללה סקולוב, שביל הדרכת, תמיות עגון-סופית, עדדות אמונתה, לשניא
היהתי מינעה לאויה שאות נצאת חים.

העבידה את מוקדשת לכל, אהוביה.

אני מודה למט حسين העם הטכניים חסיפת הנדרה בהשתלמות.
מיזוג מותנת-slot מסלולים ביולוגיים

חיבור על מחקר

לשם מיסוי חלקי של הדירוגוסל הקבוע בתוכן
מ الناس Wrocław למידת במדעי החומרים

אלכסנדרה סקולוזוב

הנהלת הטכניון - מכון טכנולוגי לישראל

מרץ 2009
ミיזוג מותנת של מסלולים
ביולוגיים

אלכסנדרה סקולוזוב