INTEGRATING BIOLOGICAL PATHWAYS AND GENOMIC PROFILES WITH CHIBE 2

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Merve Çakır

July, 2013

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Uğur Doğrusöz(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Can Alkan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Özlen Konu

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

INTEGRATING BIOLOGICAL PATHWAYS AND GENOMIC PROFILES WITH CHIBE 2

Merve Çakır
M.S. in Computer Engineering
Supervisor: Assoc. Prof. Dr. Uğur Doğrusöz
July, 2013

Biological pathways store information about spatial and temporal organization of interactions taking place in an organism. They hold valuable information that can assist scientific community in understanding the details of a particular mechanism or deciphering the reasons of disruption when the system goes wrong. However, extracting knowledge from these pathways is not trivial as they can be huge and complicated. Additionally, simple visualization of pathways will only reveal limited knowledge, whereas their integration with experimental results can identify distinct and intriguing relationships. Therefore, it is critical to have tools that are specialized in analyzing and understanding biological pathways.

ChiBE is one such tool that can visualize, manipulate and analyze pathway data stored in BioPAX format. While preparing the second version of the tool, there have been improvements regarding pathway searches, high throughput data integration, and database connections. Visual notation has also been updated in order to follow standards in visualizations defined by the SBGN community.

Previously defined pathway query algorithms have been adapted to be compatible with the BioPAX model. New query types have also been designed to offer a wider range of options. With these queries, ChiBE now offers a variety of ways of pathway decomposition and thorough analysis of complex pathway views.

There has also been improvements in integration of high throughput experimental results. To offer easy access to expression microarrays, a gateway to the GEO database has been added. The cBio Cancer Genomics Portal is also now reachable within ChiBE in order to obtain information about genomic status of various cancer cells. After simply asking for an identifier of a particular experiment, ChiBE retrieves the results from databases and then integrates them with

the available pathway view through color codes. Furthermore, a connection to DAVID database is available, in case users want to annotate a list of genes with respect to biological terms associated with them.

With these new features and improvements, ChiBE 2 has become a comprehensive tool that offers a wide range of analysis options with a genomics-oriented workflow to deepen our understanding of biological pathways.

Keywords: Computational biology, bioinformatics, pathway informatics, pathway visualization, genomics.

ÖZET

BİYOLOJİK YOLAKLARIN VE GENOMİK PROFİLLERİN CHIBE 2 YOLUYLA ENTEGRASYONU

Merve Çakır
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Doçent Dr. Uğur Doğrusöz
Temmuz, 2013

Biyolojik yolaklar bizlere bir organizmanın devamlılığını sağlayan etkileşimlerin mekansal ve zamansal organizasyonu hakkında bilgiler sunabilir. Bu yolaklar, bilim dünyasına önemli bir biyolojik mekanizmayı daha detaylı anlamasında veya sistemde gözlenen bir sorunun sebebinin anlaşılmasında önemli katkıda bulunabilirler. Fakat yolaklar büyük ve karmaşık olabilecekleri için, onlardan bilgi elde etmek basit bir işlem değildir. Ayrıca sadece görselleştirme odaklı analizler bizlere sınırlı miktarda bilgi sağlayabilirken, bu görsellerin deneysel sonuçlar ile entegre edilmesi farklı ve ilginç etkileşimleri açığa çıkarabilir. Bu nedenle, biyolojik yolakların analizi ve daha detaylı anlaşılması üzerine yoğunlaşmış yazılım araçlarına sahip olmanın önemi büyüktür.

ChiBE, BioPAX formatında saklanan yolak verilerinin görselleştirilmesi, düzenlenmesi ve analizi üzerine odaklanmış bir yazılım aracıdır. Yazılımın ikinci versiyonu geliştirilirken çizge üzerinde aramalar, yüksek çıktılı deney sonuçlarının entegrasyonu ve veritabanı bağlantılarının sağlanması üzerine yoğunlaşan eklemeler yapıldı. Ayrıca ChiBE'nin görsel sembolleri de SBGN tarafından belirlenmiş standartları takip edecek şekilde güncellendi.

Daha önceden tanımlanmış olan yolak sorgulama algoritmaları BioPAX modelleri üzerinde kullanılabilecek şekilde uyarlandı. Bunlara ek olarak yeni yolak sorgulama türleri de geliştirildi. Bu algoritmaların eklenmesiyle birlikte, ChiBE artık yolakları daha küçük parçalara ayrıştıran ve karmaşık çizgelerin detaylı analizini sağlayan birçok farklı yola sahip oldu.

Yüksek çıktılı deneylerin sonuçlarının çizgeler ile entegrasyonu ile ilgili de bazı iyileştirmeler yapıldı. Mikrodizi deneyleri sonuçlarına kolay erişim sunmak için GEO veritabanına ulaşım noktası eklendi. Ayrıca, kanser hücrelerinin

genomik durumu hakkında bilgi sunmak için ‘cBio Cancer Genomics Portal’ı da ChiBE içerisinden erişilebilir hale getirildi. Kullanıcıdan deney verisini belirleyen bilgiyi aldıktan sonra, ChiBE öncelikle istenilen deneyin sonuçlarını veritabanından elde eder ve ardından bu sonuçları çizge ile renk kodları kullanarak entegre eder. Bunlara ek olarak, DAVID veritabanına da kolay bir erişim sunulmaya başlandı. Böylelikle, kullanıcılar bir gen listesini bilinen biyolojik terimlerle eşleştirme şansına sahip olacaklar.

Bu yeni özellikler ve iyileştirmelerle birlikte ChiBE 2 genomik odaklı iş akışına elverişli, geniş çapta analiz seçenekleri sunan ve biyolojik yolaklar hakkındaki bilgi birikimimizi derinleştirebilecek kapsamlı bir yazılım aracı haline geldi.

Anahtar sözcükler: Hesaplamalı biyoloji, biyoenformatik, yolak görselleştirmesi, genomik.

Acknowledgement

First of all, I would like to thank to my advisor Assoc. Prof. Dr. Uğur Doğrusöz who has been a great mentor to me for many years. I am grateful to him for giving me the chance and courage to pursue this path, then guiding and supporting me through it.

Özgün Babur deserves special thanks as I have learned a lot from him about ChiBE and received significant advice and support from him while working on ChiBE. I want to thank Arman Aksoy for his contribution in parts related with the cBio Cancer Genomics Portal and Emek Demir for his feedbacks and opinions related to ChiBE.

Assist. Prof. Dr. Özlen Konu and Assist. Prof. Dr. Can Alkan have reviewed this thesis, so I would like to thank them for their valuable opinions and for helping me improve my thesis.

I would also like to thank to the Scientific and Technological Research Council of Turkey, TÜBİTAK, for their support.

This journey would have been harder to complete if I did not have the support from my friends. I would like to express my gratitude to all of them, including but not limited to i-Vis Research Group members - Mecit, Çağdaş, Begüm, İstemi, Marzie, Onur - Salim, Muhsin Can, Shatlyk, all my "molecular biologist" friends especially Damla and Derya. Finally, I would like to thank to my family for their support and understanding.

Contents

List of Figures

Chapter 1

Introduction

1.1 Biological Pathways

Cells, the tiny building blocks of living organisms, can be seen as a huge factory filled with metabolites, proteins or nucleic acids that are in constant interaction with each other to keep the factory running. Understanding details of these mechanisms and sets of interactions in its full context is a challenging task. Therefore rather than dealing with the whole set, researchers opted to divide cellular events into smaller yet meaningful pieces. This smaller set of interactions are chosen to be bound together as they contain molecules having causal relationships with each other, in the end to accomplish a biological event. This set of meaningful interactions is called *pathways* [1].

With the advances in molecular biology field, our knowledge in cellular events and pathways has vastly increased. With ongoing experiments, missing pieces in cellular machinery are being completed which leads to a more and more comprehensive state of knowledge in biological pathways. After deciphering the components of a pathway, the next step is more detailed analysis of this pathway in order to fully understand its function, discover its relationship with other pathways or study the possible outcomes when the machinery of the pathway goes wrong. This means that the focus of research is now diverting from gene level

to pathway level, which mandates the development of approaches that will aid scientific community in this quest.

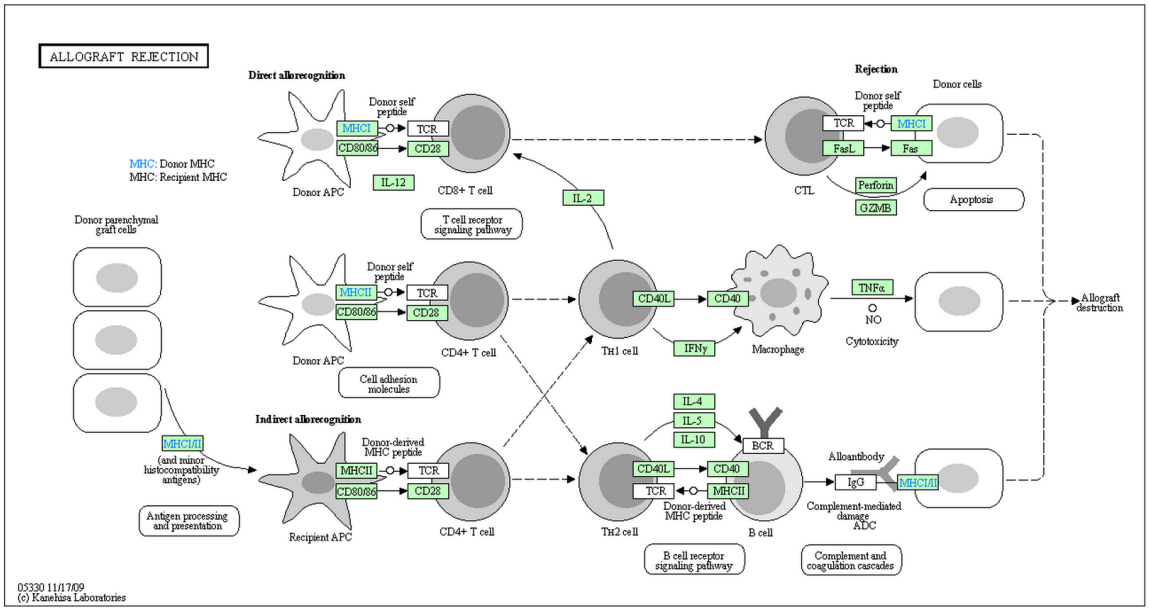
One of the first steps of pathway analysis is the visualization of a pathway. The details of a pathway deciphered through experiments can be transmitted to other researchers through textual descriptions (Figure 1.1(a)). However, long and complicated texts can make it harder to follow the interaction path and extract the critical points of the pathway. Instead, a visual representation can ease this communication. In Figure 1.1(b), we see a visualization obtained from KEGG database [2] of the textual description seen in Figure 1.1(a). With this visual representation, it is much easier to follow the flow of events. Also thanks to spatial organization, it is only trivial to find a subset of interactions, whereas one would have to scan the whole text to find the same information. This is just one instance reflecting the importance of proper choice of representation in easing the pathway analysis steps.

Unfortunately, this simple conversion of text into a diagram is not enough. One of the biggest problems associated with this is the fact that it is a static image. This means that you are limited to knowledge offered by the image and you cannot expand it or focus on a subpathway of the original image. Therefore, the advantages of this static image representation is limited. We need further improvements in pathway visualization approaches in order to be able to perform progress in research through them. There are many different approaches that can be followed to convert these static images into better analysis devices. For instance, being able to crop a pathway to generate a smaller subpathway or hiding unnecessary nodes and edges can enable researchers to focus on the pieces of a large pathway that attracts their attention most [3, 4, 5]. Alternatively, query operations can be performed on the pathway or on pathway databases [6, 7]. With such options, users can start with a set of molecules in hand and search them and their interactions in the pathway in order to generate a subpathway focusing on the molecules of interest and their relationships. Multiple different layout options can be offered, which can help the user arrange the topology of image in hand in order to understand the spatial organization of the pathway better [4, 5].

Allograft rejection - Homo sapiens (human)

Allograft rejection is the consequence of the recipient's alloimmune response to nonself antigens expressed by donor tissues. After transplantation of organ allografts, there are two pathways of antigen presentation. In the direct pathway, recipient T cells react to intact allogeneic MHC molecules expressed on the surface of donor cells. This pathway would activate host CD4 or CD8 T cells. In contrast, donor MHC molecules (and all other proteins) shed from the graft can be taken up by host APCs and presented to recipient T cells in the context of self-MHC molecules - the indirect pathway. Such presentation activates predominantly CD4 T cells. A direct cytotoxic T-cell attack on graft cells can be made only by T cells that recognize the graft MHC molecules directly. Nontheless, T cells with indirect allospecificity can contribute to graft rejection by activating macrophages, which cause tissue injury and fibrosis, and are also likely to be important in the development of an alloantibody response to graft.

(a) Textual description of a biological pathway



(b) Visual representation of the same pathway

Figure 1.1: A sample pathway obtained from KEGG database

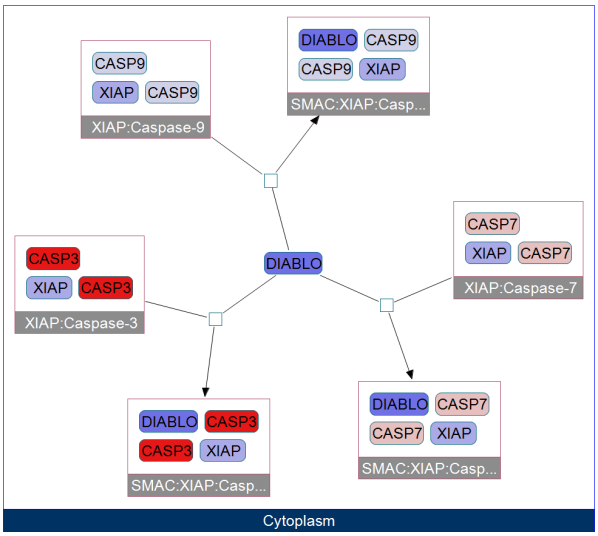


Figure 1.2: A sample pathway image integrated with microarray experiment result. The color codes are based on the experimental values each gene is associated with.

All these exemplify the cases of extension of analysis through changes in pathway structure or topology. Another trail of approach is the integration of experiment results with pathway knowledge [3, 6, 8]. Traditional way of analyzing high throughput experiment results is gene-centric. For instance, to make use of results of microarray experiments, researchers tend to identify genes that are significantly differentially expressed from rest of the gene set. Although this returns a set of genes that can be building blocks of new hypotheses, one can gain a greater amount of knowledge from these experiments if pathway knowledge is also added to the equation. Integrating high throughput experiment results with pathways generates new pathway images that not only reflect the interactions between molecules but also their experimental conditions (Figure 1.2). Researchers will be able to see whether there are clustered overexpressed molecules or if there is a dysregulated transcription factor causing differential expression in molecules in its downstream. With this approach, we can increase the amount of knowledge we can extract from both pathway information and experiment results. The set of experiments that is suitable for this integration is not limited to microarray experiments. Sequencing experiments that reflect the alteration status of genes

or copy number variation experiments are some of other cases that possess the advantage of being analyzed at a pathway level.

Pathway visualization and analysis field is still open to developments that includes but not limited to the ones discussed above. Answering the needs of the field in these respects is an important step in ensuring full knowledge acquisition from pathway information reconstructed from biological experiments.

1.2 Standards for Pathway Data Exchange

The shift of focus from gene-centric approaches to pathway-centric approaches has resulted in a rapid increase in the amount of available pathway data. Lots of different research groups focused on reconstructing pathways from biological experiments and knowledge. One problem associated with this expansion is the lack of compatibility between different curators. Due to lack of a standard format and notation for generation of pathway data, different research groups have come up with different ways of storing and representing pathway data. This builds a barrier in front of the hopes of combining data from different resources in order to build a comprehensive resource. It also makes it harder for users to utilize this data as they will need to adjust to a new way of representation every time they want to use data from a different source.

To overcome these problems, some community efforts have emerged to standardize pathway data storage and representation. One of them is Biological Pathway Exchange (BioPAX), whose objective is to develop a standard format for storing and sharing pathway data [1]. This community effort has been initiated in order to define a standard syntax for representation of pathway data, which will in turn remove the barriers in collecting, storing and sharing them. The developed syntax is based on OWL (Web Ontology Language). The initial stages of the project, called Level 1, only supported metabolic pathways. Then each level expanded the capabilities of the standard. Level 2 added signaling pathways and molecular interactions on top of metabolic pathways. The latest

version, which is Level 3, can additionally represent gene regulatory networks and genetic interactions. With this standardization, one of the biggest communication barrier between pathway researchers has been eliminated.

The followup to a standard in storage format is a central database storing and offering pathway data in that format. Pathway Commons project has started to answer this demand [9]. The main aim of this effort is to build a central hub for reaching pathway data stored in BioPAX format. Pathway data that is already stored in several different databases are collected together and integrated to be stored in one database. Pathway Commons offers different ways of reaching to this data. Users can simply use their web site in order to search for specific pathways. Download site can be used for bulk downloads of integrated set of pathway data and web service is offered to software developers as a connection to the database that can be used in their software. With these offerings, researchers can easily reach to a comprehensive set of already available pathway data in the standard format of BioPAX.

The problem of storing pathway data in a standard format and database has been solved, yet there still are points needing consideration: the visualization of this standard data. BioPAX and Pathway Commons have helped researchers share pathway data with each other without considering compatibility or understandability problems. However there was still no standard in visualizing this data. If everyone was to come up with their own notation for visualization, then it would be problematic for the end user to familiarize with all these different notations. Thus a standard way of representation was required and Systems Biology Graphical Notation (SBGN) project has been born for this reason [10]. The purpose of this effort is to develop standard notations for visualizing biological processes. SBGN community have developed three different languages for representation of biological pathways: process description, entity relationship and activity flow. Figure 1.3 shows an example for process description language, in which the events observed in a neuro-muscular junction are visualized with a focus on interactions taking place [11]. These languages form a comprehensive and unambigous way of representing pathway data.

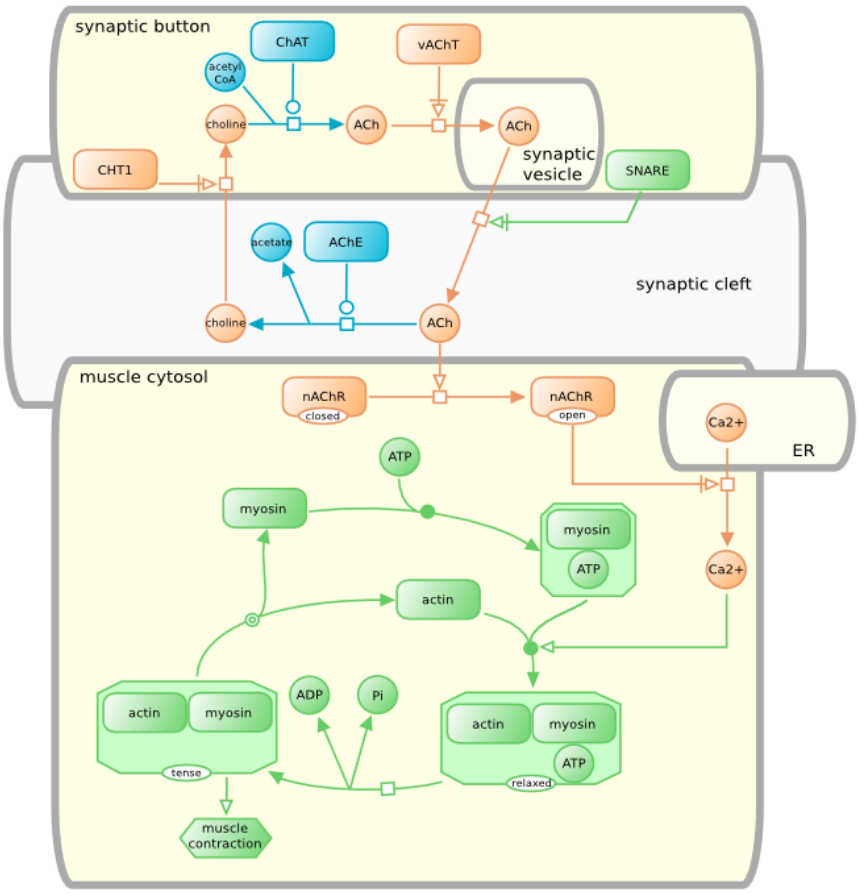


Figure 1.3: A sample SBGN process description map describing the events occurring in a neuro-muscular junction.

All these efforts have helped significantly in building a pathway research community that stores, shares and represents pathway data in a centralized and understandable way. This in turn has helped software developers to stop being concerned about these matters and to start focusing on developing tools that will answer the needs of pathway informatics field that has been discussed above.

1.3 Contribution

ChiBE is an open source tool that specializes in visualization and analysis of pathway data stored in BioPAX format [3]. Its first version is capable of visualizing BioPAX Level 2 pathway data, supporting on-demand layout of loaded pathway views. The tool offers various features that can help researchers in their analysis of BioPAX formatted pathways, including cropping subpathways, highlighting particular elements in pathways or creating simple interaction views. It can access Pathway Commons to reach the neighborhood of a specific molecule and perform a mapping between high throughput experiments and pathway view through user input.

Although it offers such a wide range of features, there is still room for improvement to make ChiBE more useful to scientific community. We wanted to improve the tool in such a way that it will offer a variety of features that can help the researchers to analyze BioPAX formatted pathway data from different perspectives. The remainder of this thesis will focus on new methods and improvements to existing methods and their implementation in ChiBE, which can be summarized as follows:

- Breadth-first search algorithm has been tailored to the specific structures stored in BioPAX format. Using this search approach, previously defined query algorithms have been implemented [12]. (Chapter 3)
- New queries have been designed and implemented to offer users the flexibility of searching a pathway model based on a set of biological entities. (Chapter 3)
- A one-click access to GEO database [13] has been added to ease the integration of expression microarrays with pathway view. (Chapter 4)
- Integration of cancer genomics data with pathway view based on alteration ratios of genes is now offered through the communication with the cBio Cancer Genomics Portal [14]. (Chapter 4)

- Connection to DAVID database [15] is supplied in order to offer a way of annotating and understanding a particular list of genes more thoroughly. (Chapter 5)
- Visual notation of ChiBE is updated in order to follow guidelines specified by SBGN [16]. (Chapter 5)
- Support for visualization of Level 3 BioPAX models is now provided, on top of the existing support for Level 2 models.

After discussing these improvements in detail in the specified chapters, Chapter 6 will focus on examples related to these features in order to highlight their use cases. Chapter 7 will conclude the thesis along with a discussion on future directions.

Chapter 2

Background Information

2.1 Graphs

A graph G can formally be defined as a pair of sets (V, E), where V stands for the non-empty set of nodes and E stands for the set of edges. Edge set E contains elements that connects nodes found in set V. $e = \{a, b\}$ represents an edge joining node a and node b. Nodes a and b are *incident to* edge e. Additionally, the two nodes are *neighbors* of each other.

A non-empty subgraph $P = (V', E')$ will be called as a *path*, if V' and E' follow the upcoming criteria. The node set V' needs to be composed of $\{v_0, v_1, v_2, ..., v_{n-1}, v_n\}$, with v_0 and v_n being the path's *endpoints*. The edge set E' will be composed of edges connecting the nodes in V', which are represented as $\{v_0v_1, v_1v_2, ..., v_{n-1}v_n\}$. The length of path P is equal to the size of edge set E'. For a path to be *directed*, all of the ordered edges have to have the same direction. A *cycle* will be formed if a path's endpoints are the same node n. Relations between nodes based on their location on the path can also be defined. Let node n_0 be the starting point of a directed path and node n_k be its ending point. Then this means that node n_k is in *downstream* of node n_0 and node n_0 is in *upstream* of n_k. [12]

Paths can also be found between sets of nodes. Consider two node sets V_1 and V_2. If only the endpoints of a path is found in V_1 and V_2 and no other intermediate nodes belong to these two node sets, then the path can be named as $V_1 - V_2$ path.

Compound graphs refer to the cases where a node in the graph may contain other nodes and edges within it. On top of the sets of nodes V and edges E, a compound graph is also defined by a set of inclusion edges F. Inclusion set F is used to represent nesting relationships between a compound node and the node found within it, which can be called as *child node*.

2.2 Pathways

The realization of these concepts in pathway visualization context can be seen in Figure 2.1, which represents a sample pathway image from ChiBE. As it can be seen, a pathway view consists of nodes and the interactions between them.

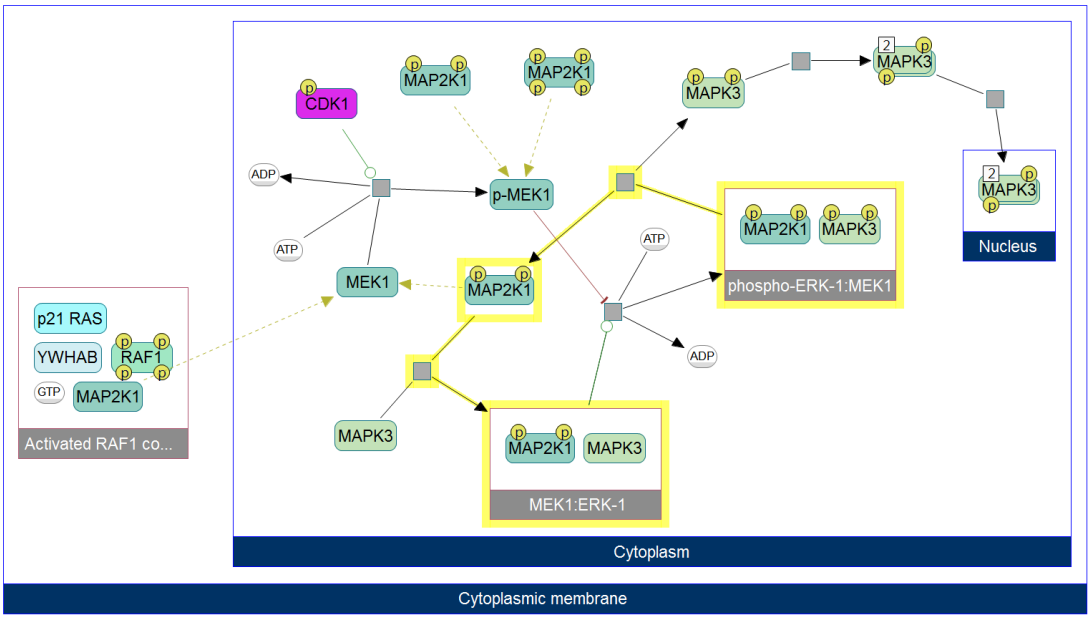


Figure 2.1: A sample graph from ChiBE

ChiBE displays given pathway data with compound graphs and compound

nodes are used for two different purposes. They can either represent complexes composed of multiple entities, such as "MEK1:ERK-1" complex, or compartments corresponding to cellular localizations, such as *cytoplasm*. In the cases of complexes, child nodes of a compound node can only be other physical entity nodes. However, children of compartment compound nodes can be any of them, including other compound nodes. The remaining elements of node set are visualized with simple nodes and they can either be a distinct *state* of a physical entity, drawn with rounded rectangles or transitions between them, drawn with gray squares. *Transitions* signify the cellular events involving these biological entities, including complex formations, transportations or reactions such as phosphorylation.

The elements of edge set connect transitions and states interacting with them based on the type of relationship between these nodes. There are different types of edges used for different purposes, such as *substrate*, *product*, and *effector* edges. Black edges are used to represent substrate and product relationships, whereas green edges signify activating effectors and red ones are used for inhibitory effector edges. There are cases where more than one effector molecule is present or where an effector edge targets another effector edge. To represent these cases, *control* nodes drawn with gray diamonds are used. The complete set of node and edge types supported by ChiBE can be seen in Figure 2.2.

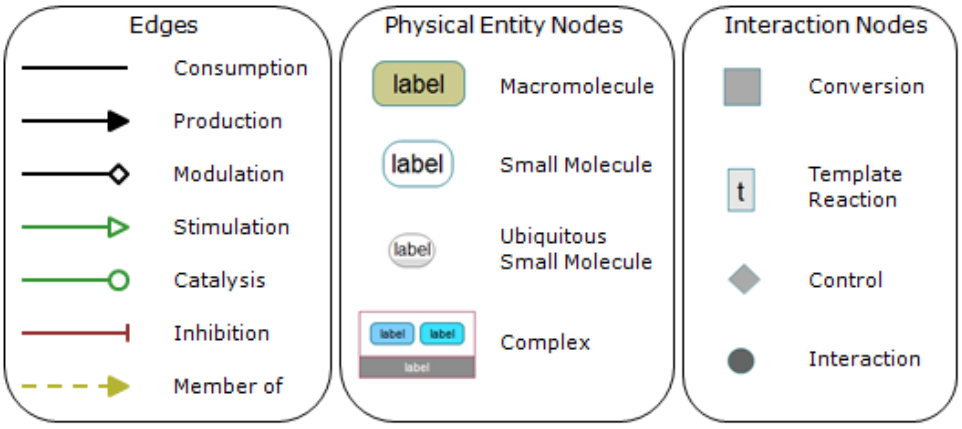


Figure 2.2: Different glyphs supported by ChiBE to represent different biological entities and relationships [35]

An example of a *path* can also be seen in Figure 2.1, which is highlighted in yellow. The start point of this path is the complex phospho-ERK-1:MEK1 and after going through two reaction nodes and a physical entity node MAP2K1, the path ends at the complex MEK1:ERK-1. In this graph, p-MEK1 is in the downstream of CDK1 and MEK1 is in the upstream of p-MEK1.

2.3 Network Traversal and Querying

2.3.1 Breadth-First Search

Breadth-first search [17] is a technique that has been developed to perform searches on graphs with respect to a set of nodes. Its main purpose is to identify the nodes in the graph that can be reached from the nodes found in a distinct source set. The basic logic of *breadth-first* is traversing every reachable node at distance k before moving onto the nodes found at distance $k+1$. A sample breadth-first search algorithm is given in Algorithm 1.

Algorithm 1 BFS(S, T, dir, $limit$)

```
1: for all node n ∈ S do
2:    n.color ← GRAY
3:    n.label(dir) ← 0
4: Q ← ∅
5: R ← ∅
6: Q.enqueue(S)
7: while Q ≠ ∅ do
8:    current ← Q.dequeue()
9:    if current.label(dir) < limit and current ∉ T then
10:      N ← current.neighbor(dir)
11:      for all node neigh ∈ N do
12:         if neigh.color = WHITE then
13:            neigh.color ← GRAY
14:            neigh.label(dir) ← current.label(dir) + 1
15:            Q.enqueue(neigh)
16:      current.color ← BLACK
17:   R ← R ∪ {current}
18: return R
```

The algorithm requires a source set of nodes (S) and a direction *dir* determining whether the search should traverse the graph moving forwards from the source nodes or backwards. It also asks for a *limit*, which is an upper bound for a reachable node's distance from the source set. A target set of nodes (T) can also be defined and when the traversal reaches to an element from this set, it does not go further in that direction. To prevent passing over a particular node multiple times, a coloring scheme is used in breadth-first search implementations. At the beginning all nodes are WHITE, except for the ones in source set. They have the color GRAY as it represents the nodes that are currently in queue. Once a node is reached for the first time, its color is converted from WHITE to GRAY and it is placed in the queue. BLACK is reserved for nodes that have already been processed. The complexity of this algorithm is $O(|V|+|E|)$, linear in the number of nodes and edges.

This search algorithm will become useful when we need to move through a graph in order to perform queries focusing on a set of nodes found in the graph.

2.3.2 Neighborhood Query

Finding the nodes surrounding a particular node is one of the first aspects that comes to mind while trying to extract knowledge from that node using the given graph. Therefore one of the basic uses of graph traversal algorithms is querying the neighborhood of a particular node [12].

Details of a proposed neighborhood query algorithm can be seen in Algorithm 2. The algorithm requires a set of source nodes (S), a traversal direction (*dir*), and a stop distance (*limit*). The search can be done in either downstream or upstream direction of source nodes or alternatively a combined search can be performed to get the complete neighborhood. If that is the case, two different breadth-first searches are performed, one in forwards and one in backwards direction. The target set of these searches are empty because there is no destination; the search is simply bounded by the distance from the source nodes. The algorithm runs in $O(|V| + |E|)$ time.

Algorithm 2 NEIGHBORHOOD(S, dir, $limit$)

1: $R \leftarrow \emptyset$
2: **if** $dir =$ FWD **or** $dir =$ BOTH **then**
3: $\quad R \leftarrow R \cup \text{BFS}(S, \emptyset, \text{FWD}, limit)$
4: **if** $dir =$ REV **or** $dir =$ BOTH **then**
5: $\quad R \leftarrow R \cup \text{BFS}(S, \emptyset, \text{REV}, limit)$
6: **return** R

When implemented, this algorithm can easily be used to identify the set of nodes that are found within a certain distance from the set of source nodes. This way we can get more information about the characteristics of surroundings of these source nodes.

2.3.3 Paths Of Interest

When you have two different sets of nodes of interest, the directed paths between them whose endpoints are these two sets respectively can hold valuable information about the flow of interaction between these sets. Similar to neighborhood query, a breadth-first traversal can be used for identification of these paths. The traversal needs to start from one of the sets and continue until it finds all the paths arriving to the nodes of the second set [12].

Algorithm 3 PATHSOFINTEREST(S, T, $limit$)

1: $C \leftarrow \text{BFS}(S, \emptyset, \text{FWD}, limit)$
2: RESETCOLOR(C)
3: $C \leftarrow C \cup \text{BFS}(T, \emptyset, \text{REV}, limit)$
4: $R \leftarrow \emptyset$
5: **for all** node $n \in C$ **do**
6: $\quad$ **if** $n.label(\text{FWD}) + n.label(\text{REV}) \leq limit$ **then**
7: $\quad\quad R \leftarrow R \cup \{n\}$
8: **return** R

Algorithm 3 reflects the basics of a sample algorithm that follows the proposed reasoning to solve this problem. The algorithm requires a source set S from which the paths will start, a target set T to which the paths will converge and a distance

limit specifying the distance between source and target nodes. At first, a breadth-first search is performed with source set of nodes in forwards direction. This will be followed by a backwards directed search from target set of nodes. The resulting set of nodes will be composed of the ones whose sum of distance labels from these two queries does not exceed the search limit. This algorithm's time complexity is $O(|V| + |E|)$. With this approach, we will be able to recover paths originating from a set of nodes and ending at a different set of nodes whose length is smaller than a predefined threshold.

2.3.4 Graph Of Interest

A special version of the described Paths of Interest query approach can be used to solve a more general problem. Instead of identifying paths following a particular direction, we can identify all of the paths found between the elements of a set of nodes [12]. Rather than dividing the set into start point nodes and end point nodes, in this case we look for a comprehensive set of paths found within all elements of a single set.

Algorithm 4 simply shows how Paths of Interest query can be used for this query type. Rather than specifying a source set and target set, a single set of nodes is given to the algorithm as both source and target sets. As a result, we will be searching for the paths that start at one of the source nodes and ends at another source node. A search limit is also required to bound the length of the recovered paths. The complexity of this algorithm is clearly the same as *Paths Of Interest*, which is $O(|V| + |E|)$.

Algorithm 4 GraphOfInterest(S, $limit$)

1: **return** PathsOfInterest($S, S, limit$)

When *Paths Of Interest* query is run on the pathway in this way, we will be able to recover the ways to traverse the graph between the given set of nodes, irrespective of the direction.

2.3.5 Common Stream Query

Each node's neighborhood holds special information for that particular node, however there may be even more valuable information hidden in the shared neighborhood of a set of nodes [12]. Searching for the elements that are located in the common downstream of a set of nodes will signify the point that the paths of these different nodes merge with each other. Nodes in the upstream that are commonly shared can be used to identify the point of divergence of different paths.

Algorithm 5 CommonStream(S, dir, $limit$)

```
1: C ← ∅
2: R ← ∅
3: for all node n ∈ S do
4:    BFSRun ← BFS({n}, ∅, dir, limit)
5:    for all node res ∈ BFSRun do
6:       res.reached ← res.reached + 1
7:    C ← C ∪ BFSRun
8:    ResetLabel(BFSRun, dir)
9:    ResetColor(BFSRun)
10: for all node c ∈ C do
11:    if c.reached = S.size then
12:       R ← R ∪ {c}
13: return R
```

Algorithm 5 explains the details of the algorithm that has been designed to find the upstream or downstream elements that are shared by a set of nodes. For every node in the source set S, first a breadth-first search is performed in the direction specified (dir) and bounded by the search limit. The difference of this algorithm from the previously discussed ones will be the way of determining final result set. For each node, a count called *reached* is kept. This reflects the number of times a particular node is found to be in the result set of breadth-first searches. So after each search run, the reached count of every node in the result set is incremented. In the end, only the nodes whose reached count is equal to the size of source set are placed in the result set. This is because we only want to recover the nodes that can be reached through every single element of the source set. So, in the end obtaining nodes fulfilling this criteria will return us the nodes that are common to the traversal paths of source nodes in the specified direction.

The run time complexity of this algorithm is $O(|S| \times (|V| + |E|))$.

2.4 Related Work

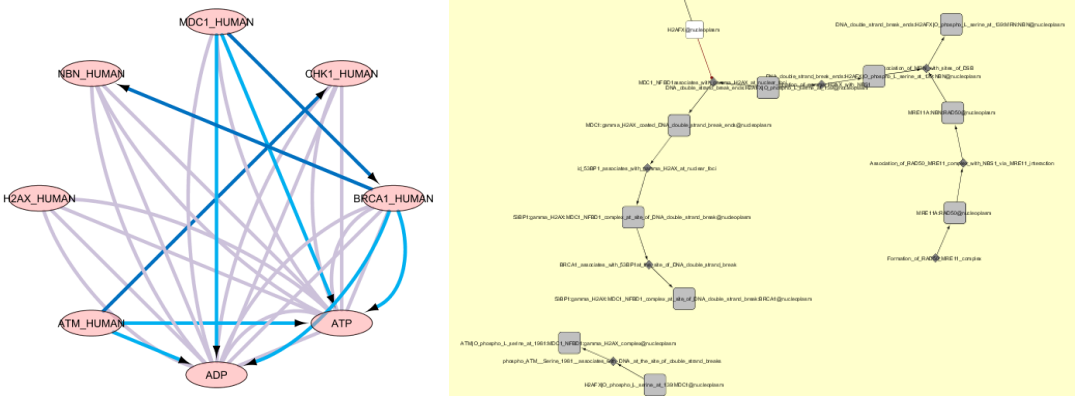
This section will focus on the tools that have been developed for enhancing the visualization and analysis of pathway data in order to give an insight about the current status of the field.

2.4.1 Cytoscape

Cytoscape is a generic visualization tool that can be used to draw, visualize and analyze networks [18]. The tool is useful for purposes of wide range of fields, ranging from computational biology to social networks. There are numerous different plugins that can extend and specialize functions of the tool.

For computational biology purposes, core Cytoscape offers importing of networks in lots of different file formats including .owl for BioPAX, .sbml and .sbgn for SBML and .xml files. Additionally networks can be obtained from online databases such as Pathway Commons. For the visualization of these networks, Cytoscape offers many different layout algorithms, such as circular, grid, spring embedded or sugiyama layouts. Figure 2.3(a) displays a network obtained from Pathway Commons database, laid out with degree sorted circle layout.

One of the Cytoscape plugins called BiNoM [19] can be used to analyze and manipulate biological networks stored in standard systems biology file formats. The plugin can be used to import and export files in BioPAX Level 3 and SBML format. It also offers topological analysis, such as centrality calculations. A sample BioPAX model's representation with BiNoM can be seen in Figure 2.3(b). The difference in visualization preferences of Cytoscape and BiNoM is prominent. Whereas core Cytoscape only offers a simple interaction network displaying the relationships between molecules, BiNoM offers a hierarchical pathway structure



(a) A sample view from Cytoscape (b) Sample pathway view obtained with Cytoscape plugin BiNoM

Figure 2.3: Sample pathway views obtained with Cytoscape

along with reactions connecting different molecules, which reflects more information about the events occuring in the pathway.

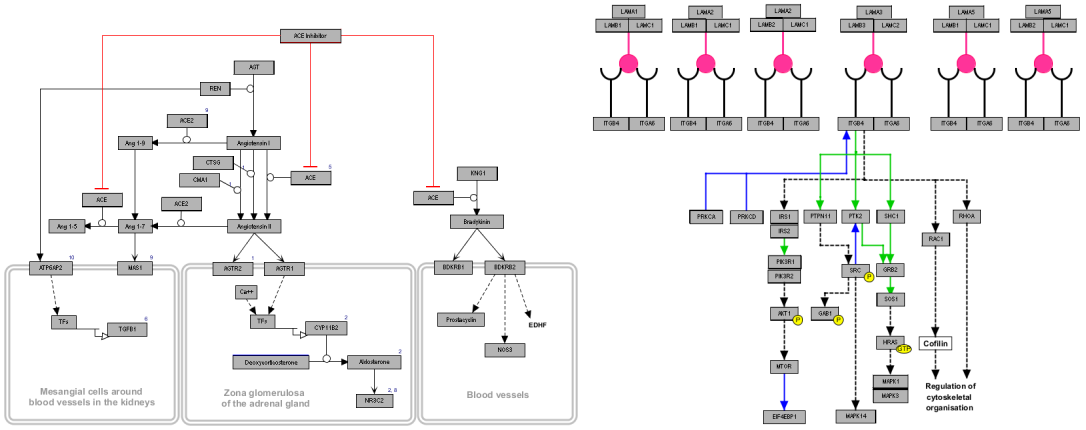
Examples of other plugins include MetScape [20], which can be used to analyze high throughput gene expression and metabolomics data. KeyPathwayMiner [21] also has a similar purpose; it is specialized in identifying significant subnetworks within a biological network by making use of gene expression data. EnrichmentMap [22] can be integrated with Cytoscape if a user wants to see the results of gene set enrichment analysis as a network. These examples show that users can install different plugins in order to customize Cytoscape and make it compatible with their needs.

2.4.2 PathVisio

PathVisio is an open source tool for visualization and editing of biological pathways [23]. The tool can import and export pathways stored in XML files and

.mapp files - which is a format specifically related to GenMAPP [24]. Additionally, pathway images can be exported in .pdf, .png, .svg and .tiff formats. Users are not only dependent on the already available pathway data; generation of pathways from scratch is also possible through drawing options offered by the tool.

The visual notation of PathVisio is quite unique, as seen in Figure 2.4. The tool uses specialized structures to reflect extra information easily - such as the compartmentalization of different environments through extra rectangular nodes seen in Figure 2.4(a). Another sample visualization can be found in Figure 2.4(b), in this example the use of significantly different edges to connect interacting nodes draws the attention of the user.



(a) (b)

Figure 2.4: Sample pathways obtained with PathVisio

Apart from visualizing the biological pathways, PathVisio can also integrate expression data with the pathway image. User is required to perform the mapping between expression data file and the structure of integration mechanism. After that, the data is integrated with the view through color coding.

PathVisio's functionality can be expanded through available plugins. Currently, there are plugins that can be installed to enable data import and export in BioPAX Level 3 or to enable drawing pathway views with SBGN notation.

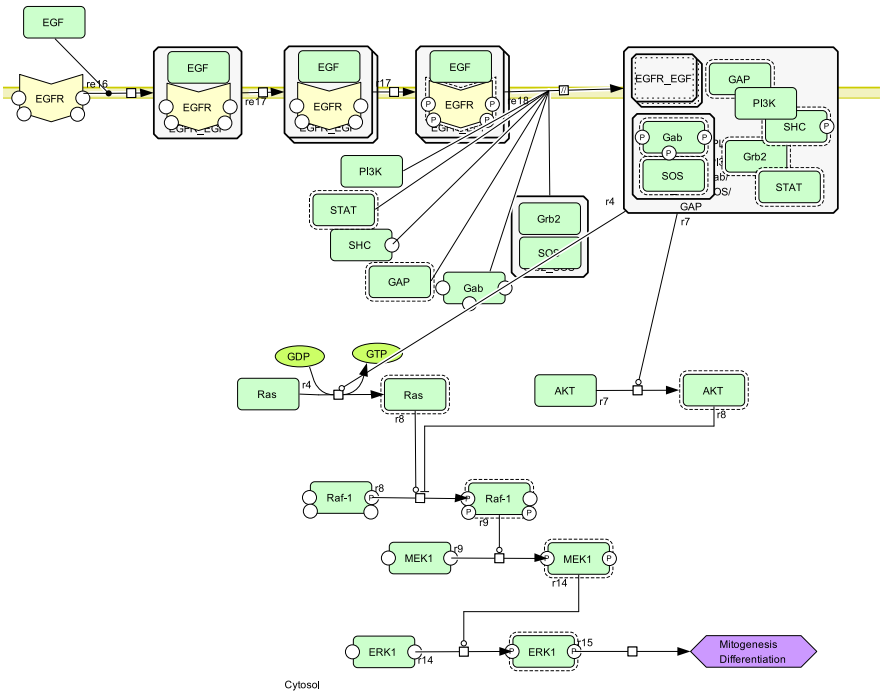
2.4.3 CellDesigner

CellDesigner is another modeling and visualization tool for biological networks [25]. SBML files are the format that has been preferred by CellDesigner, it can import and export pathway data stored in .sbml and .xml files. Public databases such as BioModels and PANTHER can also be used to import pathway data. Similar to PathVisio, it is possible to create your own pathway models through drawing capabilities offered by CellDesigner.

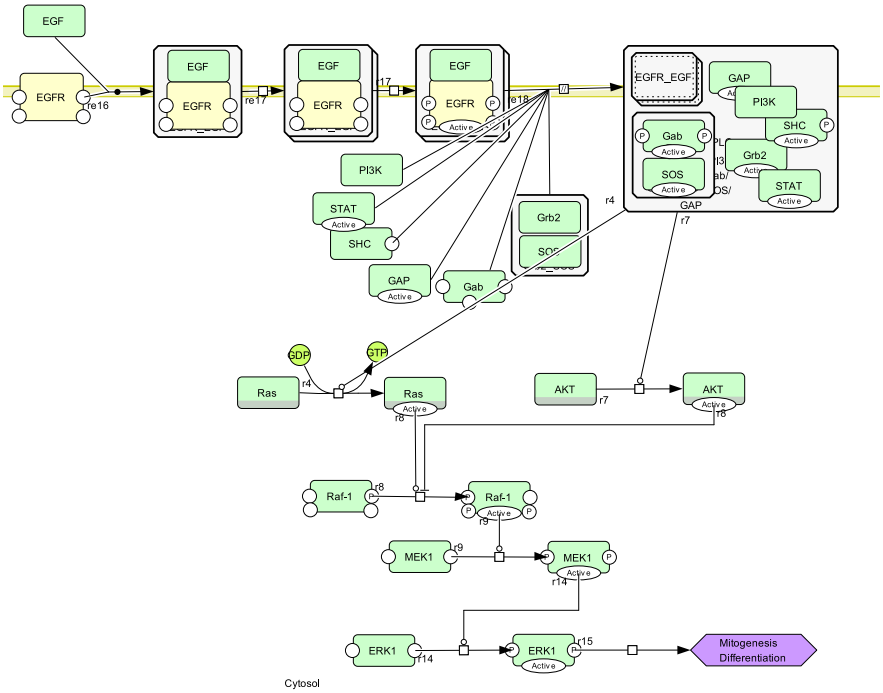
CellDesigner uses a customized visual notation that is quite similar to SBGN specifications (Figure 2.5(a)). However, it is also possible to convert the view into a version that completely follows SBGN guidelines (Figure 2.5(b)). The tool offers multiple different automatic layout options, including but not limited to hierarchic, circular, tree and organic layouts. Users can choose any of them to optimize the pathway view's organization according to their needs.

The connections offered to external databases are not limited to pathway data import. Upon selection from the view, one can choose to connect to many different databases such as PubMed, Entrez Gene, UniProt or MetaCyc to get more information about the selected node.

CellDesigner is also extensible through plugins. For instance, BioPAX Export plugin can be used to export pathway models in BioPAX Level 3 format. Another plugin is available for integrating microarray and proteomics data with the pathway.



(a) Default visualization notation of CellDesigner



(b) The same view, this time with SBGN notations

Figure 2.5: Sample pathways obtained with CellDesigner

Chapter 3

Graph Queries

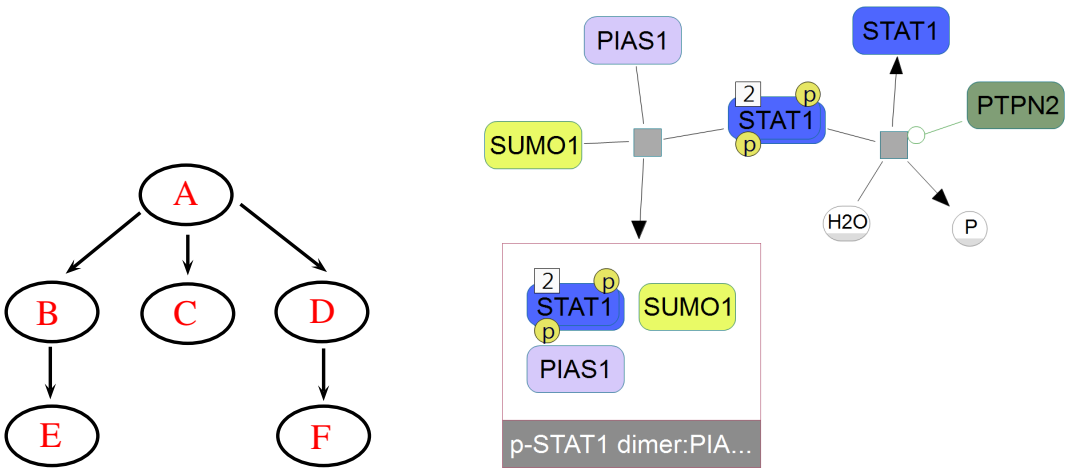
With the advances in biological research, the scientific community's knowledge about biological relationships and pathways have shown an increase at a significant pace. This in turn led to a significant increase in the knowledge stored in pathway structures and databases. The scope and comprehensiveness of empirical pathway data are increasing each and every day. However, there are some drawbacks of this obvious achievement. Increasing pathway size negatively impacts the ease of analysis. Smaller pathway models can be extensively analyzed after simply visualizing it. As the size of pathway gets bigger, it becomes more and more complex to analyze a graph by simply looking at it due to the crowdedness of the resulting picture. This problem has motivated us to offer users a way of easing analysis of complex graphs. For this purpose, we have implemented graph query algorithms that will help the user to decompose the bigger picture into smaller and meaningful parts.

In this chapter, these query algorithms will be discussed in detail. As explained in Section 2.3, graph querying algorithms require graph search approaches to traverse the graph. The given breadth-first search algorithm however is not directly applicable to pathway models visualized by ChiBE due to the nature of BioPAX format. So initially, the way breadth-first search algorithm has been adapted to overcome these problems will be discussed. Then, new graph querying approaches depending on this algorithm that are targeted for biological pathways

will be explained, which will be followed by the application of query algorithms discussed in Section 2.3 to ChiBE through use of adapted breadth-first search approach.

3.1 Adapting Breadth-First Search

While performing queries on pathway models, we can try to use the breadth-first search given in Algorithm 1 in order to move through the pathway view and find distances between physical entities. However, this is not as straightforward as it seems. The regular BFS is a perfect match for graphs that contain one type of node connected with one type of edge (Figure 3.1(a)). However, the graphs displayed with ChiBE are not that simple (Figure 3.1(b)). There are multiple types of nodes and edges. For instance, there are physical entity nodes represented with rounded rectangles and reaction nodes represented with gray squares. During our search, we only want to focus on the distances between physical entity nodes, which means that there will be additional nodes between the nodes that we are interested in, causing problems in the calculation of distances.

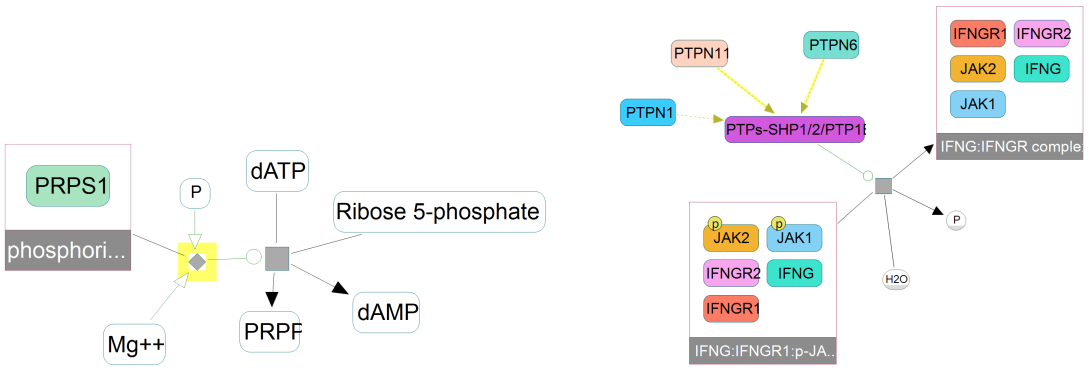


(a) A sample graph

(b) A sample BioPAX graph

Figure 3.1: Sample graphs to highlight the differences observed due to BioPAX structure

In the example seen in Figure 3.1(b), we see an alternating pattern between physical entity nodes and reaction nodes. A physical entity node is followed by a reaction node before connecting to the next physical entity node. BFS is still applicable to this graph with a simple change, which is multiplying the given distance limit with two. The kth physical entity from the source set will actually be at $2k$ distance from the source due to the presence of reaction nodes and with the multiplication, we will be searching within $2k$ distance boundaries and be able to identify physical entities at k distance as desired. Unfortunately, this pattern is not observed in all BioPAX models. There are cases where a reaction node is not followed by a physical entity node, as highlighted in Figure 3.2(a). Additionally, there is a special type of edge that represents generic relationships between nodes, which refers to the cases where a set of physical entities can be represented with a single *generic* version of them. In these cases, a physical entity node is directly connected to another physical entity node omitting a reaction node in between (Figure 3.2(b)). As a result of these special cases, we cannot apply the regular breadth-first search algorithm to calculate the distance between physical entity nodes.

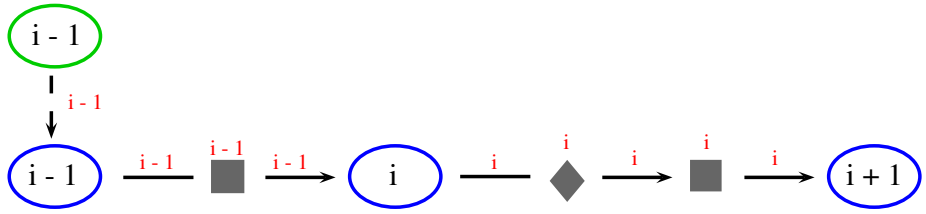


(a) Control node (b) Generic edge

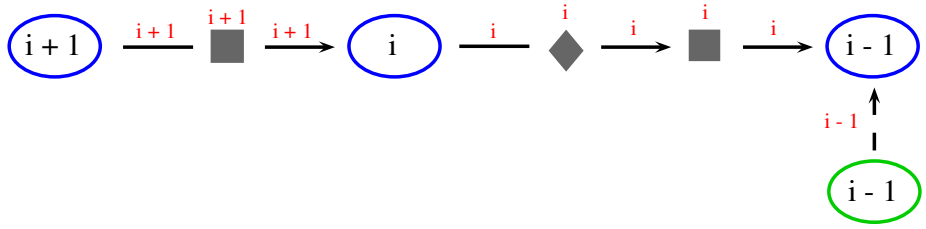
Figure 3.2: Structures of BioPAX that causes problems during application of BFS

To overcome this problem, we have come up with a new way of calculating the distances while traversing the graph (Figure 3.3). As we want to see the distance between physical entities, we will be updating the distance labels only when we interact with a physical entity node - which from now on will be called as breadth

node. If we are traversing the graph in forwards direction, then the distance label will be updated only if the search enters a breadth node (Figure 3.3(a)). While traversing in backwards direction, distance will be increased while leaving a breadth node (Figure 3.3(b)). Exceptions to this will occur in case a generic relationship represented with an equivalence edge is encountered (dashed edges in Figure 3.3); if two breadth nodes are connected with an equivalence edge, then they will have the same distance label. One node is the generic version of the other, so we count them as equal in biological context and therefore give them the same distance label to make them equal in graph context.



(a) Labeling behavior while going forwards



(b) Labeling behavior while going backwards

Figure 3.3: An alternative distance labeling approach designed to overcome the problems

The updated breadth-first search algorithm using the described labeling approach is explained in detail in Algorithm 6. This BFS algorithm requires the following four inputs to be specified: a source set of nodes (S), a target set of nodes (S), direction of traversal (dir), and stop distance ($limit$).

The algorithm starts with usual task of source set handling. The nodes in the source set are added to the queue after updating their color to gray. First important step after that is deciding the distance of the edge that comes out

Algorithm 6 UPDATED-BFS(S, T, dir, $limit$)

```
1: for all node n ∈ S do
2:    n.color ← GRAY
3:    n.label(dir) ← 0
4: Q ← ∅
5: R ← S
6: Q.enqueue(S)
7: while Q ≠ ∅ do
8:    current ← Q.dequeue()
9:    for all edge e of current going dir do
10:      R ← R ∪ {e}
11:      if (dir = FWD) or (current is not BREADTH-NODE)
         or (e is EQUIVALENCEEDGE) then
12:         e.label(dir) ← current.label(dir)
13:      else
14:         e.label(dir) ← current.label(dir) + 1
15:      neigh ← e.otherEnd(dir)
16:      if neigh.color = WHITE then
17:         if (neigh is not BREADTH-NODE) or (dir ≠ FWD)
            or (e is EQUIVALENCEEDGE) then
18:            neigh.label(dir) ← e.label(dir)
19:         else
20:            neigh.label(dir) ← e.label(dir) + 1
21:         R ← R ∪ {neigh}
22:         if neigh ∉ T and (neigh.label(dir) < limit
            or neigh is not BREADTH-NODE) then
23:            neigh.color ← GRAY
24:            if neigh is BREADTH-NODE then
25:               Q.enqueue(neigh)
26:            else
27:               Q.addFirst(neigh)
28:         else
29:            neigh.color ← BLACK
30:   current.color ← BLACK
31: return R
```

of dequeued node in the direction of traversal. First change to regular BFS is observed at this point (Line 11) in order to follow the new distance labeling approach. The required checks are performed in order to decide whether to update the distance label. A similar check is again performed at Line 17, this time in order to decide the distance label of neighboring node. Then this neighboring node is placed into the queue, if it meets the required criteria. Placement of nodes into the queue also requires extra attention. As seen in the if clause starting at Line 24, breadth nodes are placed to the end of queue as expected, whereas non-breadth nodes are added to the beginning of the queue. As a result, they are processed immediately and breadth nodes connected to them are then added to the end of queue as the next breadth of distance. After this process is performed for each node in the queue, resulting set is returned which contains all the nodes and edges that are within the boundaries of stop distance in the direction of traversal.

With this traversal algorithm, we will be able to easily navigate through the biological pathways, in time linear in the number of nodes and edges, and find the distance of any physical entity node with respect to another physical entity node. This will allow us to realize the query algorithms that will search specific paths within the graph, which are discussed in detail in following sections.

3.2 Compartment Query

The division of a cell into specific organelles means that certain reactions are confined to these subspaces, however there are also cases where a directed path spans through several different organelles. For instance, some signaling pathways start at the extracellular space after recognition of ligand through the receptor and end at the nucleus after passing through cytoplasm. In ChiBE, organelles are represented with compound nodes containing all the molecules and reactions found within them, which means that we can use this compartmentalization to search for paths that start at a specific organelle and reach to end at another organelle.

Compartment query can be seen as a special application of *Paths Of Interest* query, described in Algorithm 3, and this is reflected in its design (Algorithm 7). As a result, its time complexity is the same as *Paths Of Interest*, which is $O(|V| + |E|)$. The query requires source compartments, target compartments, and a distance limit. The breadth nodes in source compartments are extracted to form the source set and the ones in target compartments form the target set. Then simply a *Paths Of Interest* query is performed using these sets in order to obtain the paths that originate in source compartments and terminate at target compartments, that are not longer than the specified distance limit.

Algorithm 7 COMPARTMENT($Compartment\_S$, $Compartment\_T$, $limit$)

1: $S \leftarrow$ GETBREADTHNODES($Compartment\_S$)
2: $T \leftarrow$ GETBREADTHNODES($Compartment\_T$)
3: **return** PATHSOFINTEREST($S, T, limit$)

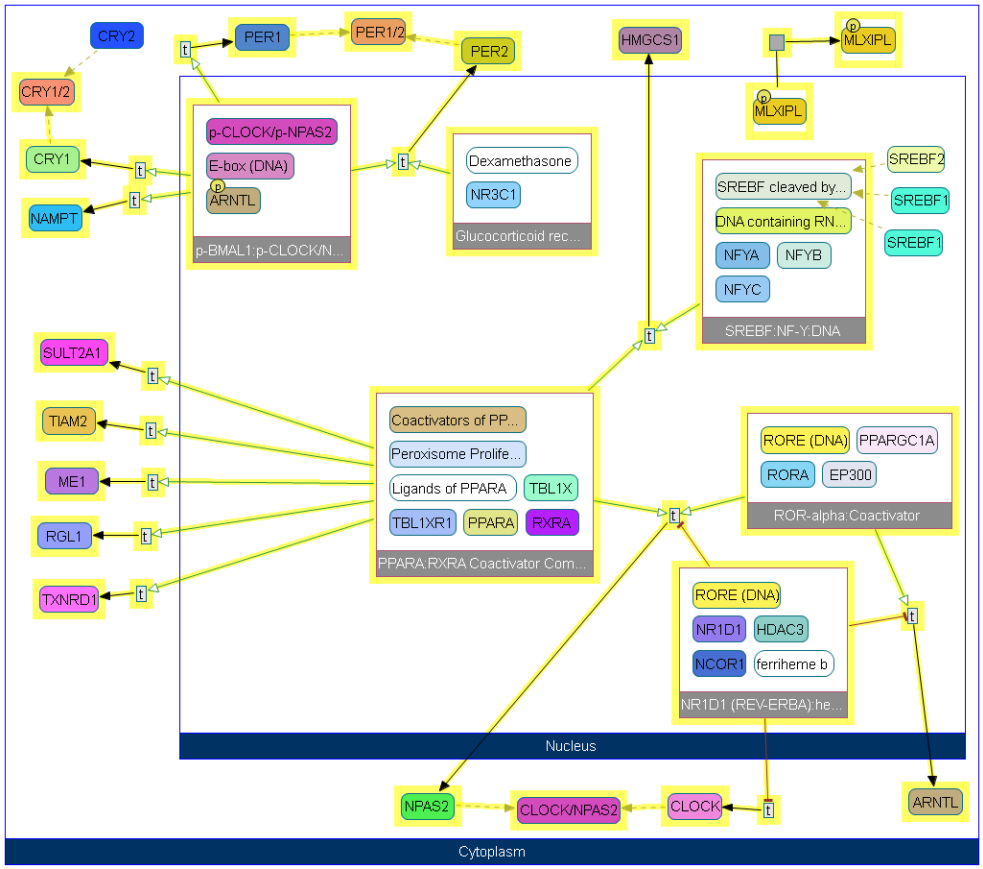


Figure 3.4: An example result for Compartment query

As an example to compartment query, Figure 3.4 represents all the paths starting from *nucleus* and ending at *cytoplasm*, found in "Signaling by EGFR" model. The paths generally represent transcription regulation events taking place in nucleus, probably the end result of the cascade initiated by EGFR.

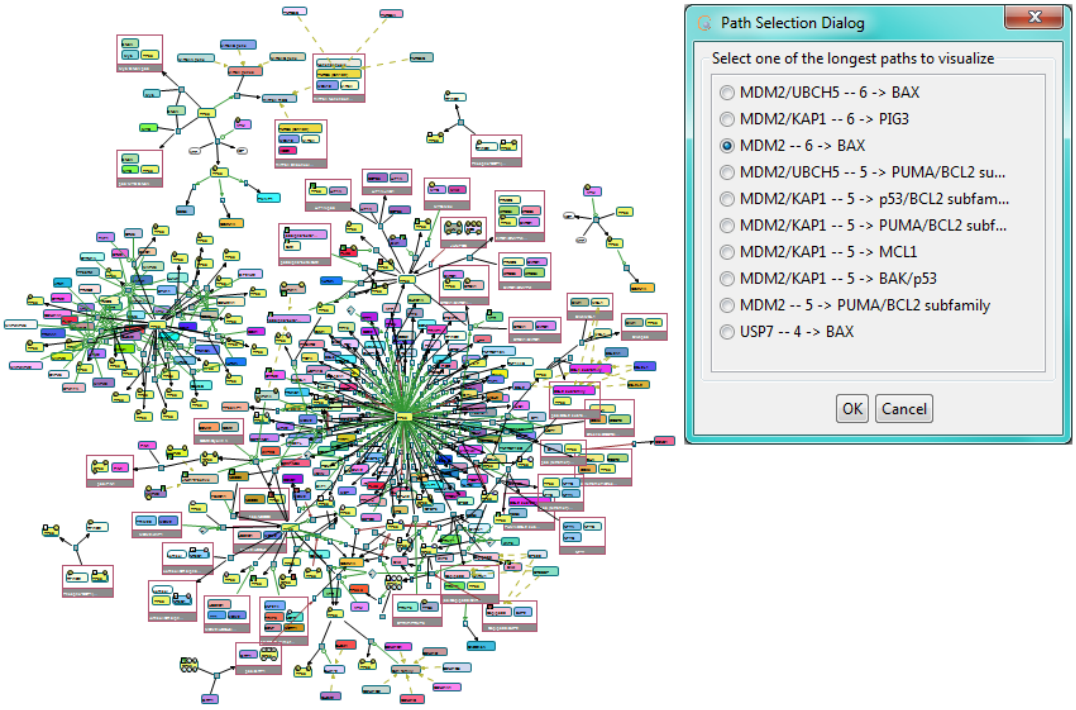
3.3 Path Iteration Query

There may be cases where a researcher has to deal with a huge and complex pathway view and may not have a specific set of proteins that can be used as a start point for decomposing such a huge graph. In that case, the previously described query algorithms will not be applicable as they all require an initial set of seed nodes. Considering such cases, another query has been designed, whose basic aim is to extract meaningful paths in a complicated graph.

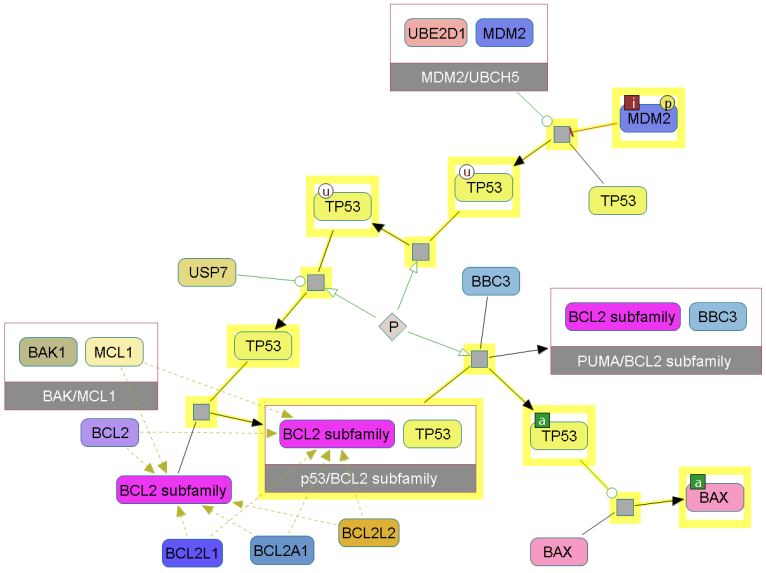
Algorithm 8 PathIteration

```
1: R ← ∅
2: C ← ∅
3: for all breadth-node n do
4:     C ← Updated-BFS({n}, ∅, FWD, ∞)
5:     for all node c ∈ C do
6:         R ← R ∪ {n, c, c.label(FWD)}
7:     ResetLabel(C, FWD)
8:     ResetColor(C)
9: sort R in order of decreasing labels
10: return R
```

The logic of this approach can be seen in Algorithm 8. In Line 4, we see that the algorithm performs a downwards breadth-first search starting from every single breadth node found in the pathway without a distance limit or target set. Then, a result set is formed, which contains the start node, the node that has been reached at the end of the search, and the distance between them. This means that the query finds the pairwise paths between the breadth nodes found in the graph. After this pair building step, resulting pairs are sorted in descending order of distances. The reason behind this sorting step is that the longer the path, the more information it will contain. The path obtained in this way will be long



(a) A complex graph and the dialog listing the paths found within this pathway



(b) The selected option is visualized with the path highlighted in yellow

Figure 3.5: An example result for Path Iteration query

enough to display enough information for a random starting point of analysis and short enough to yield a simpler network than the original one. The complexity of this algorithm is clearly $O(|V| \times (|V| + |E|))$.

The actual result of the query is left to be decided by the user. A subset of sorted list of pairs are displayed to the user (Figure 3.5(a)). The dialog first lists all the paths with the longest distance. If their number is smaller than 10, then the paths are added in decreasing order of distance until the number reaches to 10. Then only the path selected by the user is visualized (Figure 3.5(b)).

3.4 Adapting Query Algorithms

The graph querying algorithms explained in Section 2.3 can be useful tools in analyzing biological pathways more deeply. Therefore, we have decided to implement them in ChiBE in order to offer a wider range of queries. The important point to note here is that all queries will be making use of UPDATED-BFS given in Algorithm 6, instead of the regular breadth-first search detailed in Algorithm 1. With this simple adaptation, these query algorithms will become applicable to the biological pathways. Upcoming sections will explain why one might need to use these queries in context of a biological pathway and then give example results to demonstrate their implementations.

3.4.1 Neighborhood Query

In biological pathways, neighborhood of a protein contains valuable information related to that particular protein. From looking at the neighborhood, one can learn the molecules regulating that protein, the remaining substrates of the reaction that it is involved with or the proteins that are targeted by it. So we have decided to implement Algorithm 2 in order to identify the neighborhood of molecules of interest found in a pathway model.

An example result of this neighborhood query can be seen in Figure 3.6. In

this query, 1-level neighborhood of HEY1 entity has been searched. The resulting path is highlighted in yellow.

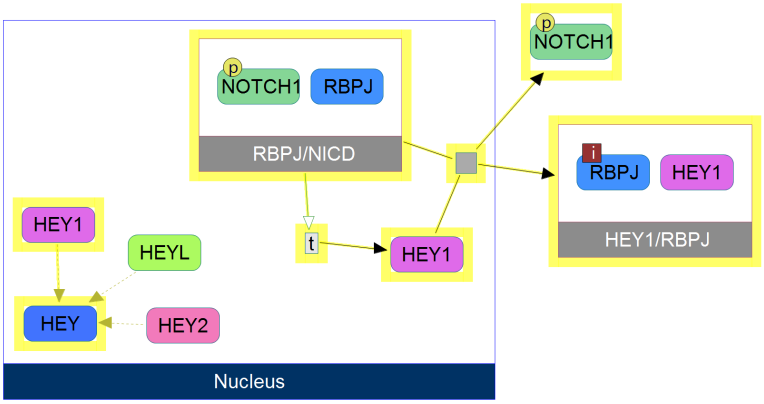


Figure 3.6: An example result for Neighborhood query

3.4.1.1 Entity vs State

At a given instance, a cell contains many different versions of a given molecule. In nucleus, it can be found as its gene as part of DNA. It can be translated into a protein and that protein may undergo different modifications, such as phosphorylation or cleavage. Although they have very different properties from each other, they still can be collected to form one type of biological molecule.

In BioPAX ontology [1], a distinction between these cases is applied. Each different version is called a *state* of that molecule. *Entity* is used to describe the collection of all states of a particular molecule. For instance, BRCA1 can be used to describe an entity without specifying a particular modification or cellular localization whereas phosphorylated BRCA1 is a state of BRCA1 entity.

While searching a given pathway model, breadth-first search can use either a specific state as part of the given set of molecules or can collectively use all of the states if an entity based search is preferred. Consider the query displayed in Figure 3.6. In that example an entity based search is requested, therefore all of the different versions of HEY1 entity is included in source set and the paths that

are reached from any of them is included in the result. However for performing the query seen in Figure 3.7, a specifically selected state of HEY1 is given as the source. As a result, the path that can only be reached through the particular state has been discovered at the end of the query.

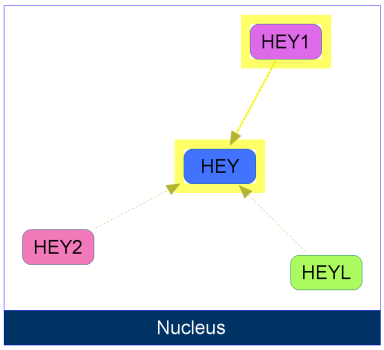


Figure 3.7: An example result for state-based Neighborhood query

The choice between an entity-based and a state-based query can be made at the starting point of the query. When the menu item offered in top menu bar is used, user is required to specify entities for the query. There is also a pop-up menu that is reachable through each node in the pathway view and it can be used to specify a state-based query. Particular states of an entity that the user is interested in need to be selected on the pathway view, then a query that only uses these specified states is performed. With this distinction, users will be enabled to focus on specific version of a given molecule without obtaining results for unrelated states of an entity.

In the remaining of the query discussions, it will be assumed that entity-based version is used unless specified otherwise.

3.4.2 Paths From To Query

Biological interactions tend to be targeted pathways, starting at a specific molecule and ending at an effector molecule through intermediate pieces. Cascade of proteins can transfer a signal throughout the cell in order to complete a

cellular event. At certain situations, understanding the flow of these interactions may become critical or helpful. Therefore we have made use of *Paths Of Interest* query approach (Algorithm 3) to develop a query that will answer this need. To make the premise of query more clear to the user, we have called the query as "Paths From To".

Figure 3.8 shows an example result. The source molecule is CASP9 and the target molecule is CASP3, length limit is 2. With these parameters, ChiBE returns the requested path in a new view with the result being highlighted. One point to note here is that Paths From To can only be performed in an entity-based manner. This is because one cannot distinguish between source nodes and target nodes when a simple selection on graph is used to specify the nodes of interest. As the distinction between source and target nodes is not available, we cannot offer a state-based query for this type.

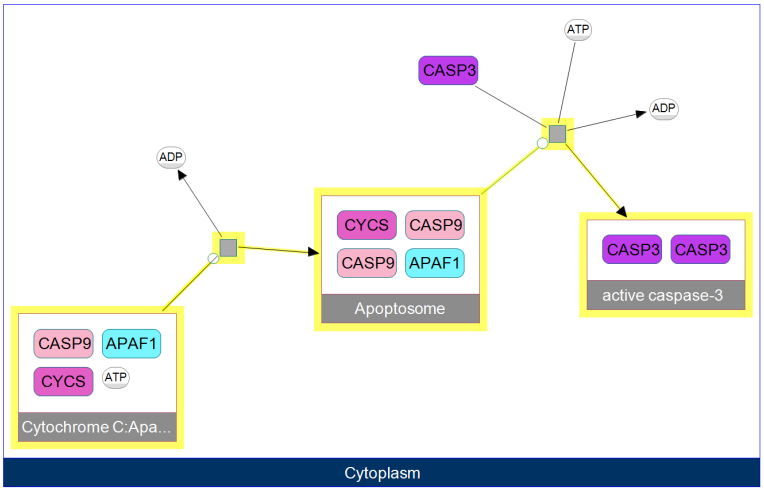


Figure 3.8: An example result for Paths From To query

3.4.2.1 Alternative Types

In Algorithm 3, we have described a way of implementing *Paths Of Interest* query. However, there are also other versions of this query type offered in ChiBE, with slight modifications.

One of the changes concerns the determination of search distance limit. In the original implementation, user is expected to define a positive integer as the limit. However, there may be cases where the user does not have an idea about the size of the path. For these situations, we offer a shortest-path based distance limit. ChiBE first calculates the length of shortest path between the source and target nodes. Then as the distance limit, the summation of shortest path's length and a user defined parameter *limit* is used. The changes that has been done can be seen in Algorithm 9. The length of shortest path is calculated by summing the *forward* and *reverse* distance labels of nodes in the candidate set (Line 5) as this will return the length of the path passing through a node. The minimum of these summations will give the minimum length of the path that can pass through any node in the network. With this new distance limit calculation approach, users can easily recover paths based on the shortest distance between the two sets. Like *Paths Of Interest* algorithm, this algorithm's time complexity is $O(|V| + |E|)$.

Algorithm 9 PathsOfInterest-Shortest(S, T, *limit*)

1: $C \leftarrow \text{BFS}(S, \emptyset, \text{FWD}, \infty)$
2: ResetColor(C)
3: $C \leftarrow C \cup \text{BFS}(T, \emptyset, \text{REV}, \infty)$
4: $R \leftarrow \emptyset$
5: $shortest \leftarrow minimum(v.label(\text{FWD}) + v.label(\text{REV}))$ where $v \in C$
6: **for all** node $n \in C$ **do**
7: **if** $n.label(\text{FWD}) + n.label(\text{REV}) \leq shortest + limit$ **then**
8: $R \leftarrow R \cup \{n\}$
9: **return** R

The second modification is about handling the relationships within source set and target set. Algorithm 10 shows its difference from Algoritm 3. Instead of calling breadth-first searches with empty target sets, this modified type defines a search with target sets. This means that the search stops at the first time a node from the target set is reached during traversal and as a result, the paths that start from a source (target) node and end at a source (target) node are not added to the result set. So this query strictly finds the paths from source nodes to target nodes and ignore relationships observed within sets.

Algorithm 10 PathsOfInterest-Strict(S, T, $limit$)

1: $C \leftarrow \text{BFS}(S, T, \text{FWD}, limit)$
2: ResetColor(C)
3: $C \leftarrow C \cup \text{BFS}(T, S, \text{REV}, limit)$
4: $R \leftarrow \emptyset$
5: **for all** node $n \in C$ **do**
6: **if** $n.label(\text{FWD}) + n.label(\text{REV}) \leq limit$ **then**
7: $R \leftarrow R \cup \{n\}$
8: **return** R

3.4.3 Paths Between Query

Paths From To query focuses on paths that follows a particular direction, however there may be cases where a user is interested in all paths between a set of molecules rather than a subset of them in a specific direction. The cases where a researcher is interested in a number of proteins but does not know the true nature of the relationship between these molecules are suitable candidates for making use of this query. To provide a solution, we require a query type that finds the paths between the molecules in a single set, irrespective of path direction and *Graph of Interest* query explained in Algorithm 4 is the exact answer to this problem. In context of ChiBE, this query is called "Paths Between" again to be more clear about the nature of the query.

The sample pathway view seen in Figure 3.9 is the result of a Paths Between query, with NBN and MRE11A entities specified as the source set. The length limit in this case is 2.

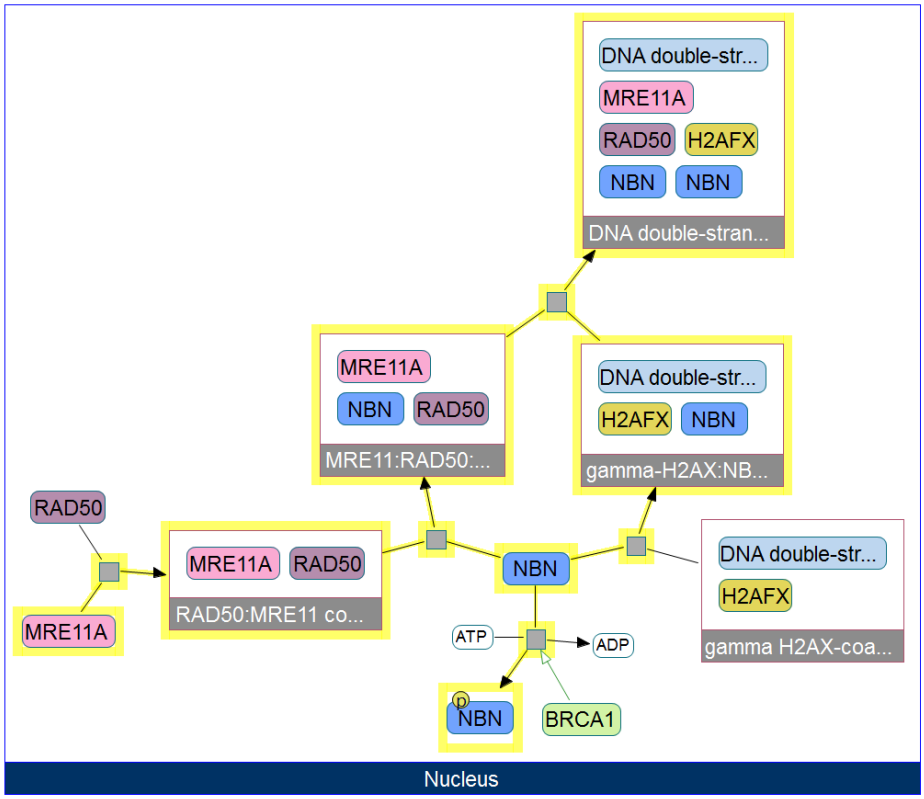


Figure 3.9: An example result for Paths Between query

3.4.4 Common Stream Query

A single transcription factor can regulate the expression of multiple different proteins. Conversely, a single protein might be regulated through the action of multiple molecules. The general trend in these cases is that there may be common molecules in the upstream or downstream of a particular set of molecules. Discovering these common regulators or targets is an important step in understanding the causative relationships between these molecules. To be able to identify these paths, we have made use of Commons Stream query described in Algorithm 5.

Example result for this query type can be seen in Figure 3.10, in which the common downstream of TP53 and ATF2 with distance limit of 3 has been searched.

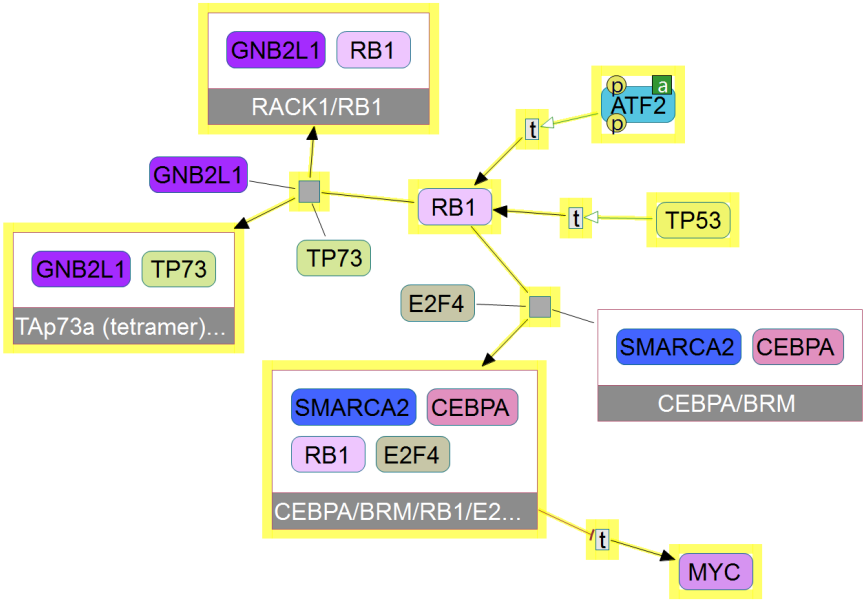


Figure 3.10: An example result for Common Stream query

3.5 Pathway Commons Queries

All of these queries described up to this point are performed on currently loaded model. This means that the search space is limited to the pathway model that has been loaded. ChiBE offers another set of queries with a much wider search space: Pathway Commons Database [9]. Instead of being limited to the model in hand, users can perform queries to this remote database. Another advantage is that users can use these queries to obtain a model to start working with when they do not have a model in hand.

Neighborhood, Paths Between, Paths From To, and Common Stream queries detailed in Section 2.3 are offered in this set. In local queries, input set of molecules is selected from the entities found in the model, whereas to perform Pathway Commons queries, HGNC symbols [26] of molecules need to be supplied. The remaining set of inputs for queries, such as direction and distance limit, are similar to their counterparts in local queries. Based on these given inputs and query type, ChiBE performs searches on the Pathway Commons database using

these query algorithms. The result is then presented to the user in a new view.

There are two other query types that are performed on Pathway Commons database, but these are not available for local models. The first one is called *Pathways With Keyword*, which as its name suggests performs the query based on a given keyword. The keyword is searched in the database in order to extract the pathway models containing the given keyword. The list of pathways obtained through this search is presented to the user in a dialog. For instance when "EGFR signaling" is given as the keyword, the list of pathways seen in Figure 3.11 is obtained from Pathway Commons database. Then user simply needs to choose one of them for it to be retrieved from the database for visualization. This query is especially useful for the cases where user is interested in a topic but does not possess a BioPAX model related to it.

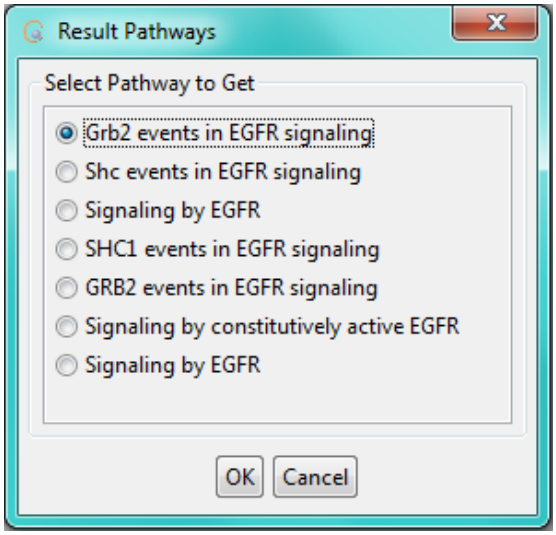


Figure 3.11: A sample result of Pathways With Keyword query

The second query specific to Pathway Commons is called *Object With Database ID*. This query asks for the RDF ID of an object of interest and then retrieves it from Pathway Commons database.

All of these different queries offers users an easy to use connection to Pathway Commons database. Normally one needs to search a database, download the

model from the database and then open that model in a tool. However thanks to these queries, users can perform all of these steps within ChiBE in one click which eases the process of reaching to a pathway view.

Chapter 4

High Throughput Data Integration

Integrating high throughput experiment results, such as microarray or copy number variation experiments, in an easy and understandable way is an important step in the analysis of these data in pathway context. This chapter will focus on the efforts related with the solutions offered by ChiBE to this problem. In the first section, microarray data integration through GEO database will be discussed. Then the next section focuses on genomic data integration and the cBio Cancer Genomics Portal.

4.1 Fetch From GEO

ChiBE 1.0 offers users a flexible and comprehensive feature of integrating experiment results with pathway view. Expression, mass spectrometry, copy number variation and mutation data can be loaded onto the network after the user has manually performed a mapping between experiment results and entities in the pathway view. This manual step of mapping can be overwhelming and may put off users from utilizing this feature. Therefore, we decided to offer simpler alternative with a smaller scope in order to eliminate the need for manual user input

for successful data integration.

Certain assumptions have to be made to automate this process. One of them is limiting the type of experiment to expression data. GEO [13] is one of the most popular databases that has been used to reach to expression microarrays. It is home to thousands of microarray experiments ranging from cancer vs. tumor comparisons to expression characteristics of embryonic cells. Because of this, we chose GEO database for the source of expression data that will be available to users.

To overlay a particular microarray data hosted in GEO database onto a particular pathway of choice, the user only needs to supply GEO series ID of the experiment (Figure 4.1). After that, ChiBE automatically downloads series matrix and platform files from the database. Then, it combines the experimental values found in series matrix file with the external database reference values found in platform file into a single XML based file. This file is then used to associate physical entity nodes of the pathway model with the corresponding experiment data, based on the mapping between database reference values. One point to note here is that pathway model needs to be obtained from Pathway Commons database in order to perform this step successfully. Details of this mechanism will be explained in the following subsection.

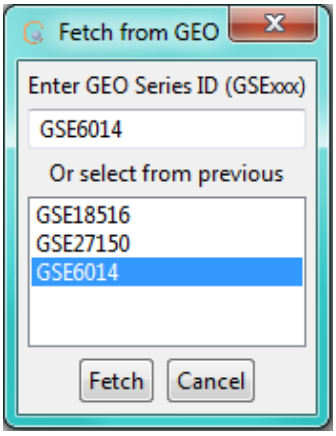


Figure 4.1: Sample "Fetch From GEO" dialog

As seen in Figure 4.1, the dialog also displays a list of GEO series IDs. These are the microarray experiments that have already been downloaded from GEO database and converted into native format of ChiBE. Considering the fact that a user might be interested in seeing same microarray data in context of multiple pathways, the generated files are stored in a local directory to ease their reuse. The downloaded series matrix and platform files are also stored in this directory.

4.1.1 Implementation of Fetch From GEO Feature

The main steps performed in "Fetch from GEO" feature offered by ChiBE is summarized in the Algorithm 11.

Algorithm 11 FetchFromGEO($seriesID$)

Require: $seriesID$ is a string starting with "GSE"

```
1: cedFile ← null
2: if seriesID.ced exists then
3:     cedFile ← seriesID.ced
4: else
5:     seriesMatrix ← GetSeriesMatrix
6:     platform ← GetPlatform
7:     ReadPlatformFile(platform)
8:     ReadSeriesFile(seriesMatrix)
9:     cedFile ← Finish
10: ColorWithExperiment(cedFile)
```

$seriesID$ is the only input required from the user, it stores the GEO series ID of the experiment of interest. The method first checks the local directory for an already existing .ced file belonging to this series ID. If found, this .ced file is directly used to integrate pathway view with the experiment data through color codes. If ChiBE has not already prepared an experiment file for the particular experiment, the remaining required steps are performed.

First of all, ChiBE requires series matrix and platform files belonging to the specified experiment. Lines 5 and 6 refers to these steps. These files are downloaded from GEO database, only if they are not already found in the local directory. The elapsed time for download will depend on the size of these files, so

storing them in a local directory will eliminate spending extra time if the user wants to reuse the same experiment for more than once. When we ensure that both series matrix and platform files are ready for use, method's first action will be obtaining header lines from both platform and series matrix files. As it can be seen in the line highlighted in red in Figure 4.2, each platform file contains a particular header line which lists the reference types stored in the platform file. In this example, we can see "Gene Symbol" or "ENTREZ_GENE_ID" of the particular gene, with the identifier of microarray probe stored in "ID" column. ReadPlatformFile method extracts these reference names from the platform file. Additionally, this method is also responsible for automatically finding the matches between the reference list found in platform file and the known reference set stored by ChiBE. ChiBE stores a mapping between known database names and their synonyms commonly encountered in platform files to describe these databases (Figure 4.3). So ReadPlatformFile method finds the reference names that have counterparts in the knownn reference set in order to identify columns of platform file that will be used in the upcoming steps.

^PLATFORM = GPL96
#ID = Affymetrix Probe Set ID LINK_PRE:"https://www.affymetrix.com/LinkServlet?array=U133&probeset="
#GB_ACC = GenBank Accession Number LINK_PRE:"http://www.ncbi.nlm.nih.gov/nuccore/?term="
#Sequence Source = The database from which the sequence used to design this probe set was taken.
#Target Description =
#Gene Title = Title of Gene represented by the probe set.
#Gene Symbol = A gene symbol, when one is available (from UniGene).
#ENTREZ_GENE_ID = Entrez Gene Database UID LINK_PRE:"http://www.ncbi.nlm.nih.gov/gene/" DELIMIT:" /// "
#RefSeq Transcript ID = References to multiple sequences in RefSeq. The field contains the ID and Description for each entry
!platform_table_begin

ID	GB_ACC	Sequence Source	Target Description	Gene Title	Gene Symbol	ENTREZ_GENE_ID	RefSeq Transcript ID
1007_s_at	U48705	Affymetrix Proprie	U48705 /FEATURE=i	discoidin don	DDR1	780	NM_001954
1053_at	M87338	GenBank	M87338 /FEATURE=	replication fa	RFC2	5982	NM_002914
117_at	X51757	Affymetrix Proprie	X51757 /FEATURE=c	heat shock 7(	HSPA6	3310	NM_002155
121_at	X69699	GenBank	X69699 /FEATURE=	paired box 8	PAX8	7849	NM_003466

Figure 4.2: A portion of a platform file GPL96

Similar to platform files, series matrix files also contain informative lines related to contents of the file before the actual experimental values begin (Figure 4.4). In this example, first column contains the probe identifier and the upcoming columns hold GEO accession values for each different experiment found

// Do not change this part. First columns represent the generic term for the remaining reference names in the same row.				
GB_ACCESSION	GenBank Accession	GB_ACC	GenBank	GB_LIST
GENE_SYMBOL	Gene Symbol	Gene symbol	GENE_SYMBOL	Symbol
GENE_ID	Locuslink	Entrez Gene	Locuslink ID	Locuslink_ID
UNIGENE	UNIGENE	Unigene	Unigene_ID	Database DB:unigene
OMIM	OMIM	MIM		
REF_SEQ_PROT_ID	Refseq prot	REF_SEQ	refseq	RefSeq
REF_SEQ_TRANSCRIPT_ID	Refseq trans	RefSeq Transcript ID		
ENSEMBLE_COLUMN	Ensembl Gene ID	Ensemble	Database DB:ensembl	ENSEMBL
SWISSPROT	Swissprot	SWISSPROT	SwissProt	swissprot
CPATH	CPATH			
HGNC	HGNC			
Key to data file(s)	ID	Key	KEY	Key to data file(s)
// You can add your own reference matchings below. Please follow the notation above.				

Figure 4.3: A portion of known reference set that is stored in ChiBE listing the generic name and synonyms for popular references

!Series_title	A Mitochondria-K+ Channel Axis Is Suppressed in Cancer & Its Normalization Promotes Apoptosis and Inhibits Cancer Growth			
!Series_geo_accession	GSE6014			
!Series_pubmed_id	17222789			
!Series_type	Expression profiling by array			
!Series_platform_id	GPL96			
!Sample_title	A549: contro	M059K: cont	M059KDCA:	A549DCA: DCA treated
!Sample_source_name_ch1	cultured epit	malignant br	malignant br	DCA treated Cultured Lu
!Sample_treatment_protocol_ch1	No treatmen	no treatment	48 h with Dic	48 h with dichloroaceta
!Sample_description	disease : Ca	disease : m	disease : m	disease : Carcinoma
!Sample_description	Age: 58 yea	Age: 33 yea	Age: 33 yea	Age: 58 years old
!series_matrix_table_begin				
ID_REF	GSM139658	GSM139662	GSM139665	GSM139670
1007_s_at	3144.7	5544.2	3448.6	5349
1053_at	40.7	468.7	362.6	79.5
117_at	79.5	415.1	524.5	371.7
121_at	3310.5	3168.5	3113.1	5409

Figure 4.4: A portion of a series matrix file GSE6014

within GEO series. ReadSeriesFile method extracts this header line to identify the column that contains the probe identifier values and the columns that contain actual experimental values. One point to highlight is the matching between probe identifier columns of platform and series matrix files. Thanks to the use of same identifiers for same probes, we can link the experiment values found within series matrix files to external references stored in platform files.

After these preliminary steps, the critical work is performed in Finish method, whose basic steps are highlighted in Algorithm 12.

Algorithm 12 Finish

```
1: experiment ← CreateExperimentData
2: FillInReferences(experiment)
3: FillInDataValues(experiment)
4: ClearUselessRows(experiment)
5: if experiment is not empty then
6:    cedFile ← WriteExperimentData(experiment)
7: return cedFile
```

ChisioExperimentData(.ced) file generated at the end of this process is an XML based file (Figure 4.5). So as the first step, an XML object *experiment* is created to be filled in during the upcoming steps. Each *row* in it corresponds to a single microarray probe and the methods found in Finish are responsible for filling up *ref* and *value* tuples of every row. We have already obtained the matching of reference names between platform file and known reference set with ReadPlatformFile (Line 6 in Algorithm 11). FillInReferences method uses these reference columns in order to fill reference values for each row of resulting .ced file. For each probe, it consults the platform file to get the values of selected references and then stores them as *ref* values, as seen in Figure 4.5. After filling reference values for each microarray entry, Finish then moves on to actual experiment values with FillInDataValues method. As explained previously, platform and series matrix files contain equivalent probe identifiers and these will be used at this step to associate experimental values with rows already possessing reference values. The number of *value* tuples found in each row is specific to the number of experiments found in the original series matrix; each experiment is integrated separately. At the end of these steps, two separate files that we had

at the beginning gets merged into a single structure that is easy to use in the upcoming processes. Then ClearUselessRows is called to get rid of the rows in *experiment* object that lacks either reference or experimental values. Final step of this method is writing of this XML object into a .ced file, provided that *experiment* object actually contains reference and experimental values.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rootExperimentData xmlns="http://mada.patika.org/dataXML">
    <experimentType>Expression Data</experimentType>
    <experimentSetInfo>!Series_title    "A Mitochondria-K+ Channel Axis Is Suppressed in Cancer
    <experiment>
    <experiment>
    <experiment>
    <experiment>
    <row>
        <ref>
            <db>GENE_SYMBOL</db>
            <value>DDR1</value>
        </ref>
        <ref>
            <db>REF_SEQ_TRANSCRIPT_ID</db>
            <value>NM_001954</value>
        </ref>
        <ref>
        <ref>
        <ref>
        <ref>
        <value>
            <no>0</no>
            <value>3144.7</value>
        </value>
        <value>
            <no>1</no>
            <value>5544.2</value>
        </value>
        <value>
            <no>2</no>
            <value>3448.6</value>
        </value>
        <value>
            <no>3</no>
            <value>5349.0</value>
        </value>
    </row>
```

Figure 4.5: A portion of .ced file highlighting its components

With this step finished, microarray experiment result stored in GEO database has been automatically converted into an XML based .ced file, without causing any practical concern to user apart from supplying a series ID. One point to note here is that the generated .ced file is independent of currently loaded pathway model onto which data will be overlayed. There is no a dependency to information

stored in pathway model at any step of this process. Thanks to this, once a .ced file is created for a particular microarray experiment, it can be integrated with any pathway model ensuring the reusability of the files.

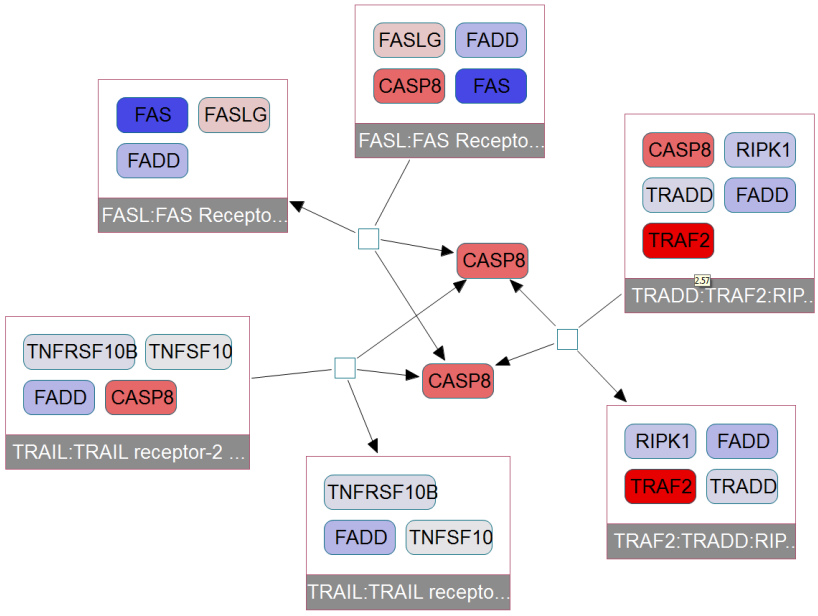


Figure 4.6: Example pathway view with microarray data (GSE6014) mapped onto the pathway objects

The final act to complete data integration is to match these experiment values in .ced file to the nodes in the pathway and to update color of each node based on the experiment results. This step is handled by ColorWithExperimentAction method, which maps the experiment values to the nodes in the graph based on reference values of each node. Each node possesses entries of certain external databases and the overlap between these set of reference databases with the ones in the platform file is used to identify the node that a microarray probe corresponds to. An example result can be seen in Figure 4.6, in which pathway named "Activation of Pro-Caspase 8" is integrated with GSE6014. Tooltip of a node is updated to contain the experimental value and color of the node is updated based on a predefined color code. Shades of red are used for positive values and shades of blue are used for negative values, darkness of which corresponds to the magnitude of the experimental value. The specific subset of experiments

that are used to calculate these experimental values is user-dependent. From all the available experiments found within a series matrix files, the user can choose a subset of them or all of them to visualize. If that is the case, their average is calculated as the final experimental value. Users can also perform comparisons, if they choose two different sets of experiments. To reflect the comparison, log-2 ratios of the averages of the experiments found in each set is calculated.

As it can be seen, this new feature integrates expression microarray data with the pathway view seamlessly and without being a burden to the user. This ease of use will help to remove the barriers in analysis of expression data along with pathway knowledge.

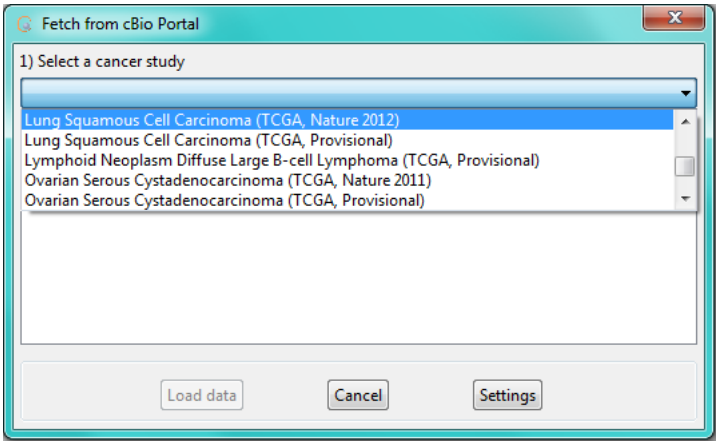
4.2 Fetch From cBioPortal

The cBio Cancer Genomics Portal, shortly the cBioPortal, [14] is a web based source that provides in depth analysis for cancer genomics data. With the advences in genomics technologies, TCGA and similar community efforts have been trying to unravel the mysteries of cancer cells in genome level by sequencing their genome and searching for alterations at nucleotide and expression level. These efforts in turn have led to the requirement for tools to analyze the vast amount of data that has been generated through the experiments. The cBioPortal has been developed to answer these needs. The portal currently hosts data for more than 30 cancer genome datasets, with the number increasing based on the availability of new experiments. For these cancer samples, it offers a variety of data including DNA copy number, mutations, mRNA expression, microRNA expression, phosphoprotein level (RPPA) and DNA methylation data. ChiBE offers a gateway for the cBio Cancer Genomics Portal, in order to integrate it with powerful pathway knowledge. With the cBioPortal, users can see the alterations at genomic and expression level for each gene. By connecting ChiBE to the portal, we are offering the possibility of an analysis at pathway level. This way, users will be able to see the roles of highly altered genes in critical biological pathways or search for driver mutations in known cancer pathways.

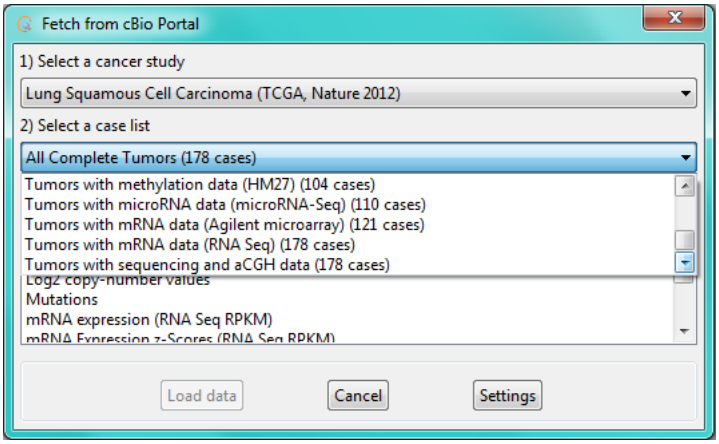
ChiBE uses web service interface offered by the cBioPortal in order to retrieve required data from portal. The user is required to determine three parameters in order to use this feature (Figure 4.7). The first one is the *cancer study* (Figure 4.7(a)). All available cancer studies are listed in a drop-down menu and the user is expected to choose one of them. Based on this selection, the choices for second parameter - which is *case list* - are updated. This option allows the user to decide subset of samples to focus on (Figure 4.7(b)). For instance, one may choose to see the experimental results only for tumors with sequencing and aCGH data or all of the available tumor samples. After this, the only remaining input is the *genomic profile* (Figure 4.7(c)). At this step, the user is required to select type of experimental results that s/he wants to retrieve from the portal, for instance copy number alterations or mRNA expression or mutations. Selection of multiple genomic profiles is possible at this step, in that case the combined results for all selected profiles are calculated.

After these selections are completed, ChiBE requests the specified data from cBio Cancer Genomics Portal through its web service interface. The specific experiment results for the specified subset of cancer dataset are then used to calculate alteration percentage of each gene found in the pathway model. For this purpose, certain thresholds are determined in order to determine a gene's status as *altered* based on the experimental values. Figure 4.8 reflects the predefined values for these thresholds. For instance for a gene to be labeled as altered, its expression value has to be greater than 2.0 or smaller than -2.0. These thresholds can be updated by the user through "Settings" dialog available in cBio Portal parameter selection dialog (Figure 4.7).

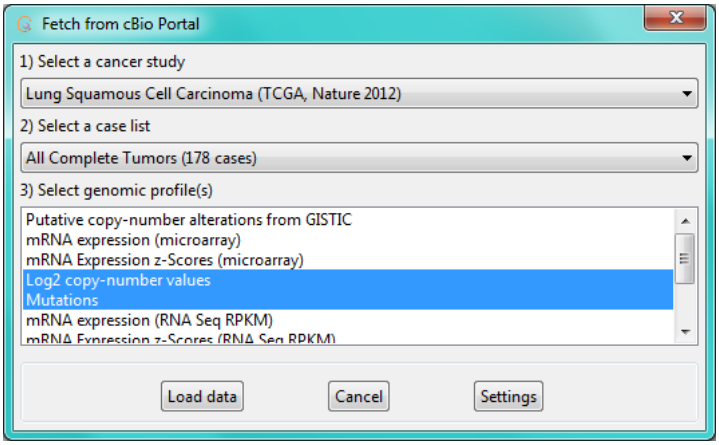
Similar to the integration of GEO data, ChiBE requires an external reference to map data obtained from cBio Portal to the nodes found in the pathway model. However, in this case this reference matching solely depends on HGNC gene symbols [26]. HGNC IDs of genes found in the pathway are first obtained and then these are used to retrieve experimental results from the portal. For each gene, the values are gathered for each sample of the specified case list. Then based on the thresholds (Figure 4.8), a binary decision is made. If the gene is found to be altered in the particular sample, its value becomes 1.0. If it is not



(a) Cancer study selection



(b) Case list selection



(c) Genomic profile selection

Figure 4.7: cBio Portal parameter selection dialog

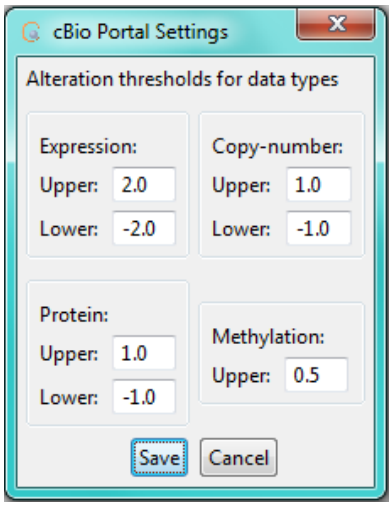


Figure 4.8: cBio Portal Settings dialog where you can adjust thresholds for different data types

altered, then the given value is 0. After this conversion is done for every sample in the case list, the average is calculated to determine the final alteration value for that particular gene. After every gene's alteration percentage is calculated, new coloring is performed on the pathway view based on the color codes reflecting this alteration values (Figure 4.9). Again, tooltip of each physical entity node displays this final value.

Additional insight into this alteration value is provided through the cBioPortal data details dialog that is offered in pop-up menu of physical entity nodes. As it can be seen in Figure 4.9, this additional dialog gives detailed information about alteration status of a particular gene. When multiple genomic profiles are loaded, it separately provides information for each of them and for their cumulative result. In addition, the percentage of alterations whose outcome is known - activating or inhibiting - are separately given to enhance analysis.

With this new feature, users can integrate genomic status of a specific cancer with any pathway data they choose in one click, and then analyze how these alterations can cause disturbances in biological interactions.

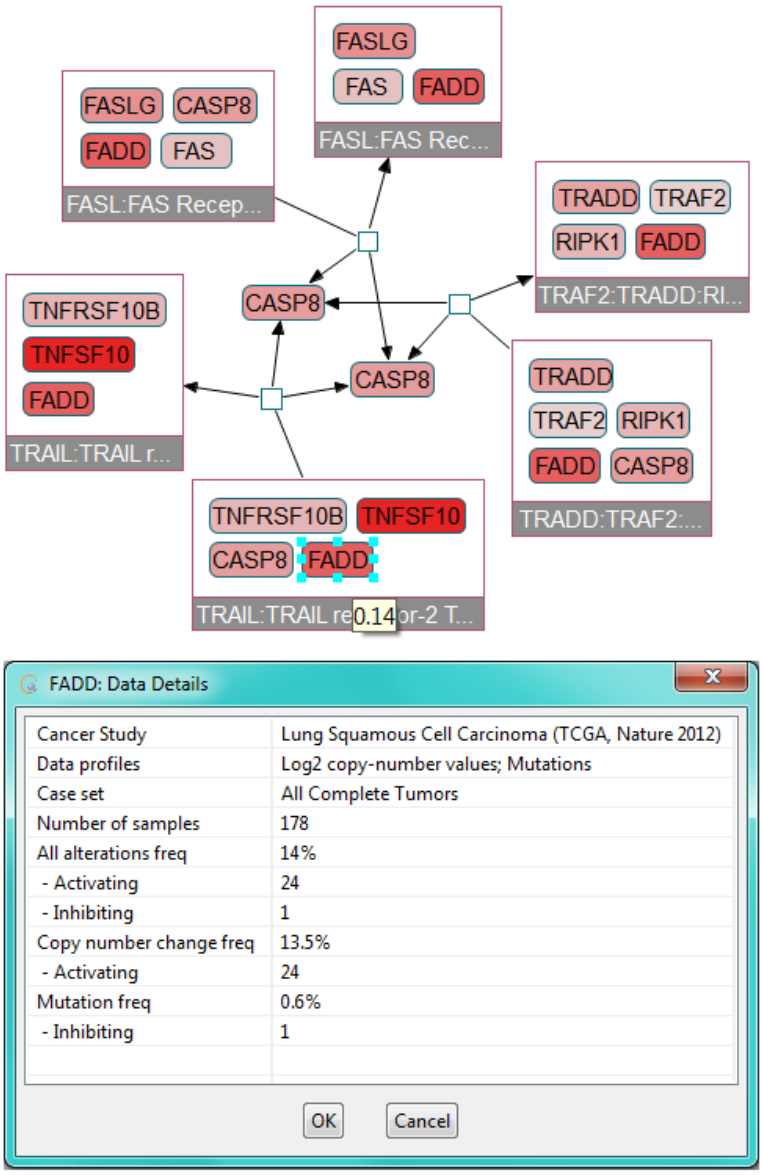


Figure 4.9: A sample pathway overlayed with data from the cBioPortal, also exemplifying the cBioPortal Data Details Dialog of a selected node.

Chapter 5

Further Improvements

This chapter discusses miscellaneous additional changes done in ChiBE with version 2. First, a new connection offered to DAVID database will be described. Then, the updates in visual notation of ChiBE that has been done to increase compatibility with SBGN notation will be presented.

5.1 Connecting to DAVID

The Database for Annotation, Visualization and Integrated Discovery (DAVID) [15] is a popular web portal that has been designed to offer biological research community ways of annotating and understanding functionalities and properties of sets of genes and proteins. The target of the tool is gene or protein lists that has been identified by researchers to be related with each other. Then, this list is supplied to DAVID, in order to perform analysis that will provide more insight into the elements of the list and give clues about their relationships. One of the main features is identification of Gene Ontology [27] terms that are commonly observed within the list. Based on these Gene Ontology terms, DAVID can also perform clustering of this list into highly related groups.

While performing their analysis in ChiBE, users may in the end reach to

a set of molecules that they are interested in. They would like to know more about this set of molecules in order to understand why their analysis has revealed this particular set of molecules. For this purpose, DAVID might be a useful tool. However, to be able to use DAVID to annotate the result set, users need to first identify the IDs of these molecules as DAVID requires unique IDs of genes/proteins from a particular type as its input. Manually identifying the specific ID of each molecule and then combining them to supply to DAVID may be a long and demanding task, especially when the set of molecules is large. To overcome this obstacle, we offer an easy-to-use connection to DAVID to open new ways of analysis to the user.

This new feature called "Connect To DAVID" will take the set of molecules from the user and then automatically direct them to DAVID to see the result of analysis. User first needs to select nodes of interest from the pathway view. Then, clicking on "Connect To DAVID" item under "Data" menu of the top menubar will open a dialog that will list the available analysis options offered by DAVID (Figure 5.1). There are four different options related to the features offered by DAVID. "Functional Annotation" option will perform annotation enrichment analysis, in order to identify Gene Ontology terms that are found to be significantly enriched in this gene set. "Gene Functional Classification" divides the list into clusters based on the functions and annotations of the elements within. To get extensive information about each gene, "Gene Report" can be used. "Gene Name Batch Viewer" lists each gene along with its identification ID, species and genes related to it.

Based on the selection performed, a URL following DAVID's API service [28] syntax will be automatically prepared. For this URL, ID of each selected node is required and it is automatically extracted by ChiBE from the information associated with each node. No user input is required to find IDs of selected nodes. Then, the user will be directed to this URL to reach to the result of the analysis. Figure 5.2 gives an example result when "Gene Name Batch Viewer" option is selected with the nodes selected in Figure 5.1.

With this feature, a quick connection to DAVID database is offered in order

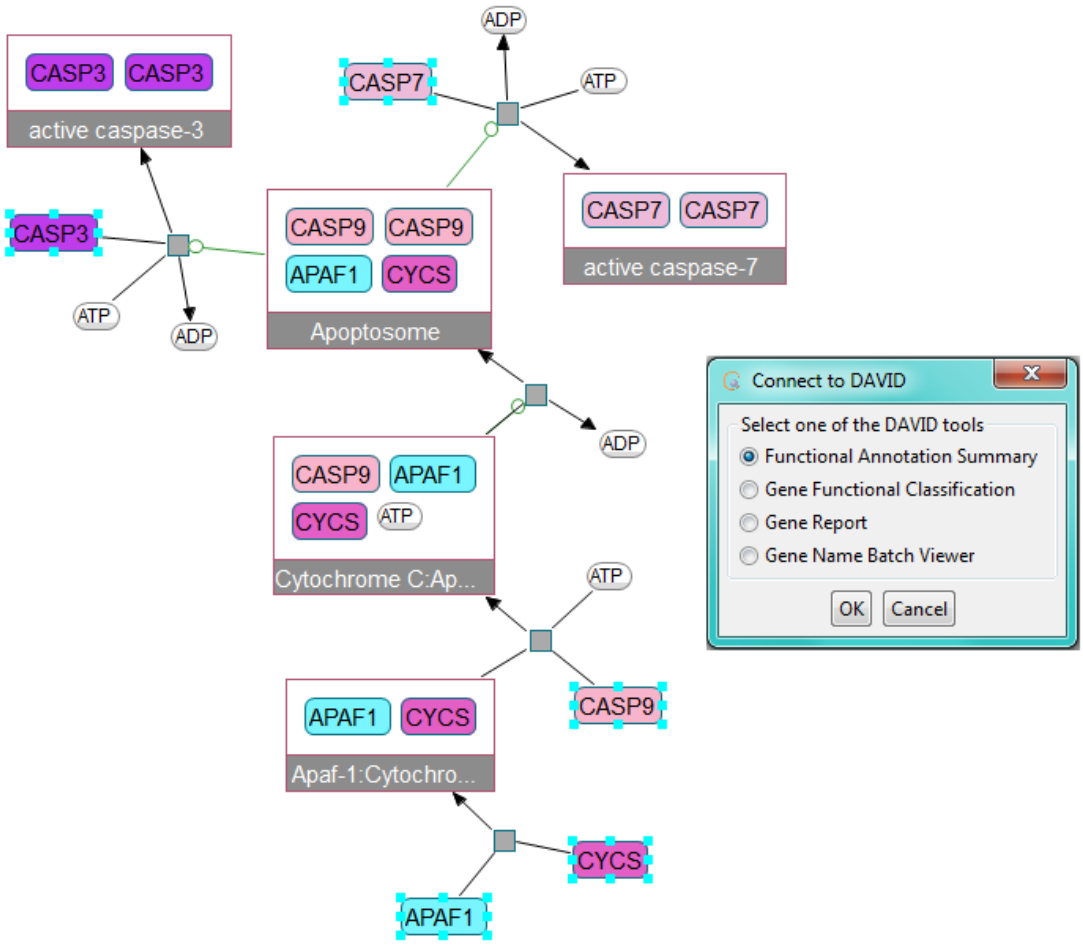
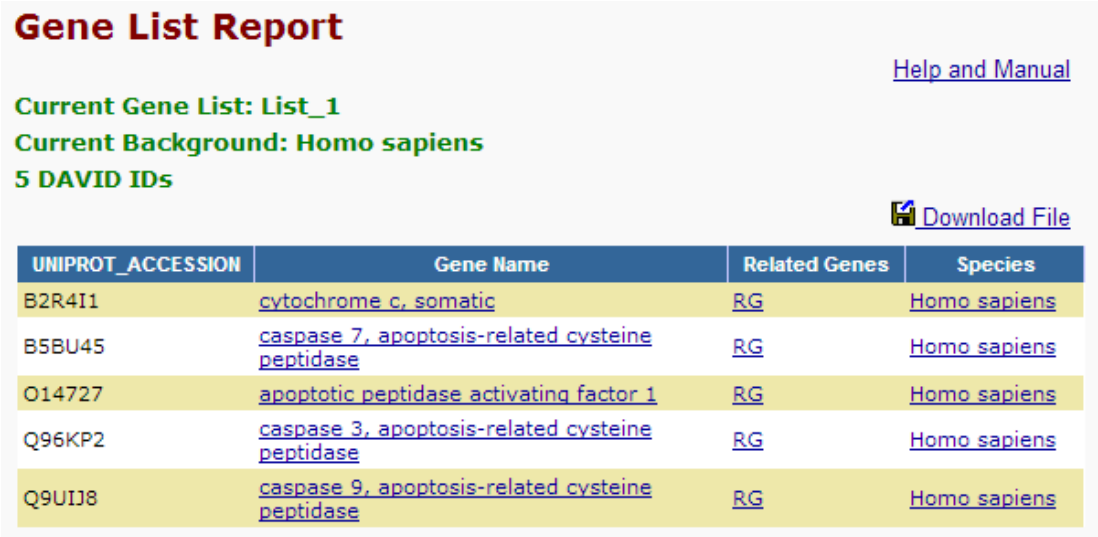


Figure 5.1: A sample pathway image used to connecting to DAVID, along with the dialog that will be used for selection of analysis option.



Gene List Report

Help and Manual

Current Gene List: List_1
Current Background: Homo sapiens
5 DAVID IDs

Download File

UNIPROT_ACCESSION	Gene Name	Related Genes	Species
B2R4I1	cytochrome c, somatic	RG	Homo sapiens
B5BU45	caspase 7, apoptosis-related cysteine peptidase	RG	Homo sapiens
O14727	apoptotic peptidase activating factor 1	RG	Homo sapiens
Q96KP2	caspase 3, apoptosis-related cysteine peptidase	RG	Homo sapiens
Q9UIJ8	caspase 9, apoptosis-related cysteine peptidase	RG	Homo sapiens

Figure 5.2: The result of "Gene Name Batch Viewer" analysis

to ease the annotation of genes or proteins of interest that has been obtained through the use of other features of ChiBE.

5.2 SBGN-PD Notation

SBGN community has developed three different notation specifications for visualization of biological pathways and one of them - *process description* [16] - is relevant to our purposes. Process description language has been designed to visualize biological reactions and interactions occurring in a cell with spatial information and temporal flow of events. This specification is compatible with the type of information that ChiBE is designed to present. Although following a similar reasoning to process description diagrams, previously ChiBE had its own visual notation not following the SBGN standard. With version 2, ChiBE now mostly complies with SBGN process description (SBGN-PD) notation (Figure 5.3). This will decrease the adaptation period of new users, who are already familiar with SBGN notation, to ChiBE.

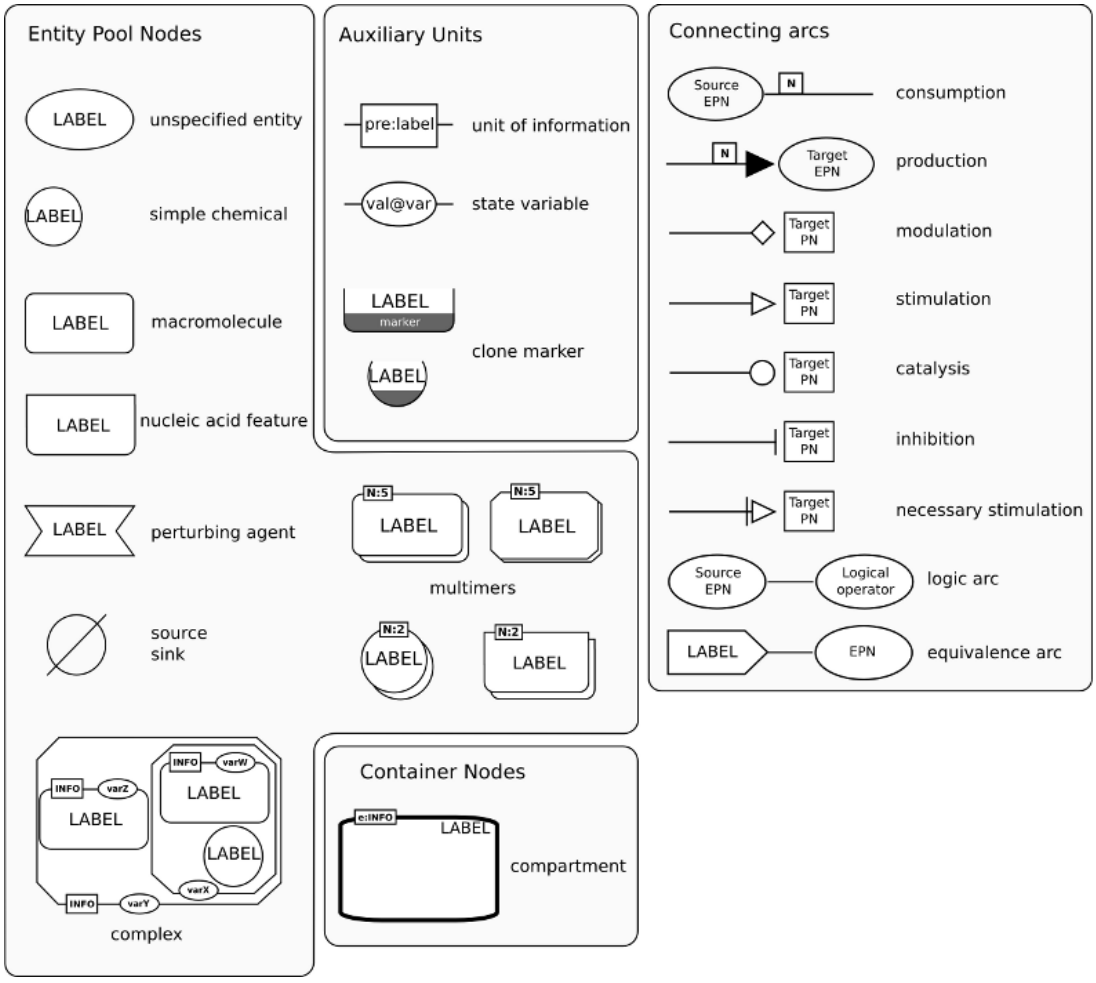


Figure 5.3: SBGN-PD's visual notation (modified from [16])

The updated set of nodes and edges used by ChiBE can be found in Figure 2.2. Just like SBGN-PD, ChiBE uses rounded rectangles to signify macromolecules. However, there is a difference in visualization of small molecules (simple chemicals in SBGN-PD). SBGN suggests use of circles for their display, but label placement in a circle is not as effective as label placement in a rounded rectangle. Therefore, ChiBE uses rounded rectangles also for visualization of small molecules. The differentiation with macromolecules is ensured by coloring. Small molecules are always white, whereas each macromolecule is given a different color during visualization. Clone markers are recently added in order to emphasize the cases where a small molecule is found ubiquitously in a pathway. Multimer visualization has

also been updated to follow SBGN-PD guidelines.

Changes in the visualization of edges have also been done. Previously, ChiBE was using color code to differentiate the roles of edges: green for activating relationships, red for inhibiting relationships and black for the remaining. We have merged this color code with the suggestions of SBGN-PD notation. The red inhibition edge's arrow type has been updated. Green activating relationships has been divided into two as simulation and catalysis, adding different edge types for these two. As SBGN suggests, a modulation edge has also been added. Different from SBGN notation, ChiBE has one more edge type called "member of", as this relationship does not have a counterpart in SBGN standards. It is used to represent generic relationships between physical entities and visualized with a specific colored dashed edge.

Chapter 6

Example Results

In this chapter, we will be exemplifying the features of ChiBE discussed in previous chapters as use cases in the analysis of biological pathways.

6.1 From Pathway View to Genes of Interest

We will start this example with a relatively complicated pathway view and try to reach to a set of genes that may be significant through integration of experimental results. The focus of the analysis will be liver cancer. Some cancer cases are associated with disruptions in particular pathways that are critical to survival of cells. These disruptions include mutations or copy number variations that affect a gene at the critical point of the pathway, so when it is not functioning, remaining of the pathway collapses. Alternatively, changes in expression of genes causing under or over-abundance of proteins may cause significant dysregulations in pathways. Identifying these dysregulated genes can help scientific community to better understand the mechanics of the disease or to develop treatments and with this example, the aim will be using ChiBE to discover problematic genes hidden in a cellular pathway that is known to be associated with disease. For instance, Fibroblast Growth Factor (FGF) signaling pathway has been identified to be associated with liver cancer [29] and so this pathway will be the focus of

analysis to see whether there are genes that are misbehaving in the context of liver cancer.

To start our analysis, we first perform *Pathways With Keyword* query on Pathway Commons database with "FGFR" as the keyword. The pathway named "Signaling by FGFR" seems to be a good fit for our purpose, so it is selected for download from Pathway Commons database.

The first line of analysis can be integration of this pathway view with expression data. Liao *et al.* have performed microarray analysis in order to decipher the gene expression profiles of normal and tumorigenic liver tissues [30]. This dataset can be a useful experiment to understand the expression differences between cancerous and normal tissues in the context of FGFR signaling pathway. To integrate the result of this experiment with the pathway model obtained from Pathway Commons, we can use "Fetch from GEO" feature, with the GEO series ID of GSE6222. Then, we specifically assign normal tissue samples to one set and cancer samples to another set in order to make a comparison between them through calculation of log2-ratios. The resulting view, in which experimental values are reflected with coloring of the nodes, can be seen in Figure 6.1. Shades of red correspond to cases where the average expression in tumor samples are higher than the average expression in normal samples and converse of it is valid for shades of blue.

In analysis of expression values, generally, outlier samples have importance compared to the average samples. Therefore, we would like to see which entities in the view have the highest and lowest values when tumor and normal samples are compared. To identify these entities in the network, we can use "Highlight With Data Values" feature of ChiBE. By using the dialog seen in Figure 6.1, the threshold for highlighting is first set to the maximum value possible, which in this case is equal to 62.34. As a result, IL17RD has been identified as the entity with highest tumor-to-normal expression ratio (Figure 6.2). This large number means that this gene is significantly overexpressed in tumor samples compared to normal tissues. Then, the threshold is set to the minimum value possible of -6.03. MAP2K1 is found to be the entity possessing this value (Figure 6.2). In

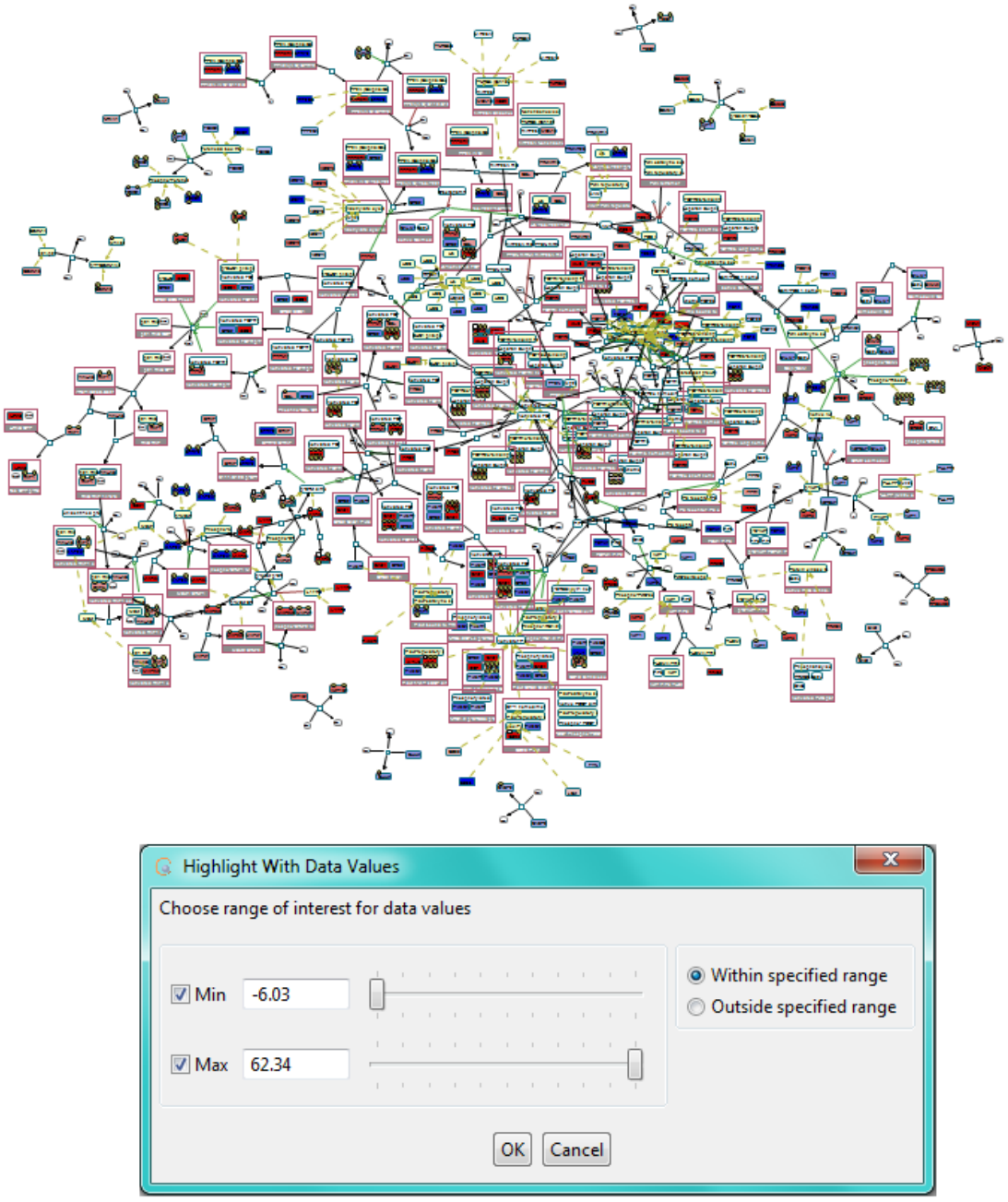


Figure 6.1: *Signaling by FGFR* pathway is integrated with expression data. "Highlight With Data Values" dialog can also be seen, reflecting the smallest and largest expression values found in the experiment.

contrast to IL17RD, MAP2K1 is significantly downregulated in tumor samples as its expression value is higher in normal samples.

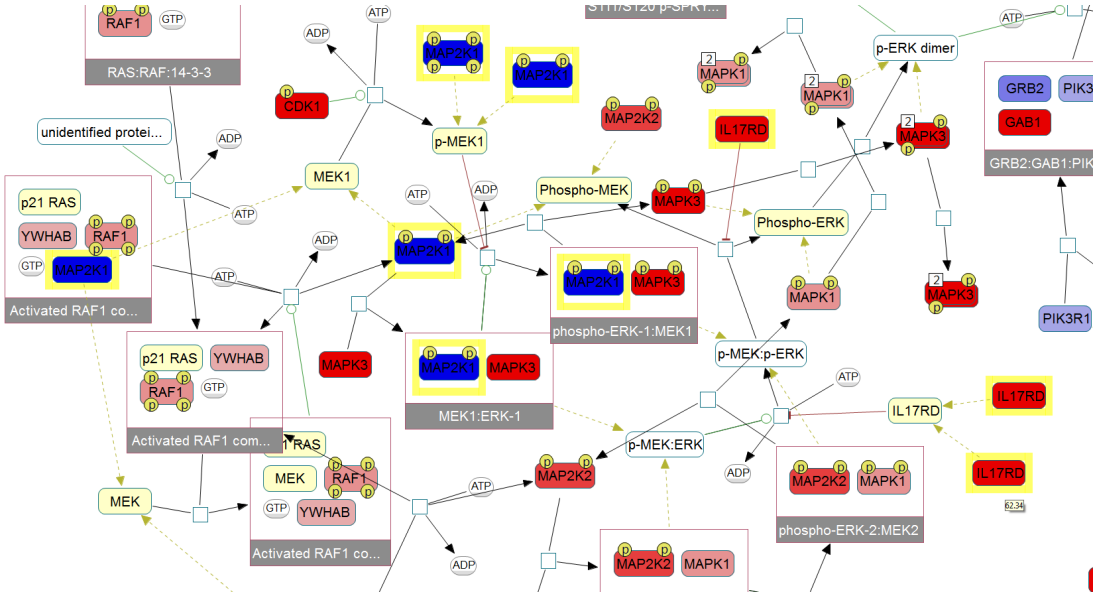


Figure 6.2: IL17RD and MAP2K1 entities, which are highlighted in yellow, are found to be the outliers of this expression dataset.

Genomic status of this network can also possess interesting information. To perform this analysis, "Fetch From cBio Portal" feature will be used. From the cBioPortal, we would like to obtain genomics data of a liver cancer related experiment. Cancer Cell Line Encyclopedia cancer study set [31] has a case list specific for liver cancer and *putative copy-number alterations* and *mutation* genomic profiles from this case list are chosen for integration with pathway view. The result of this integration can be seen in Figure 6.3. Shades of red are used in order to represent the percentage of alteration observed in each gene.

Similar to the previous case, we can use "Highlight With Data Values" feature in order to find out the entity which has been altered in most of the liver cancer samples. As it can be seen in the dialog in Figure 6.3, this value is 0.36. The entity possessing this alteration percentage is EGF. As seen in data details dialog found in Figure 6.4, the majority of alterations that this gene undergoes have inhibiting effect meaning that in 32% of these cancer samples, EGF is not functioning

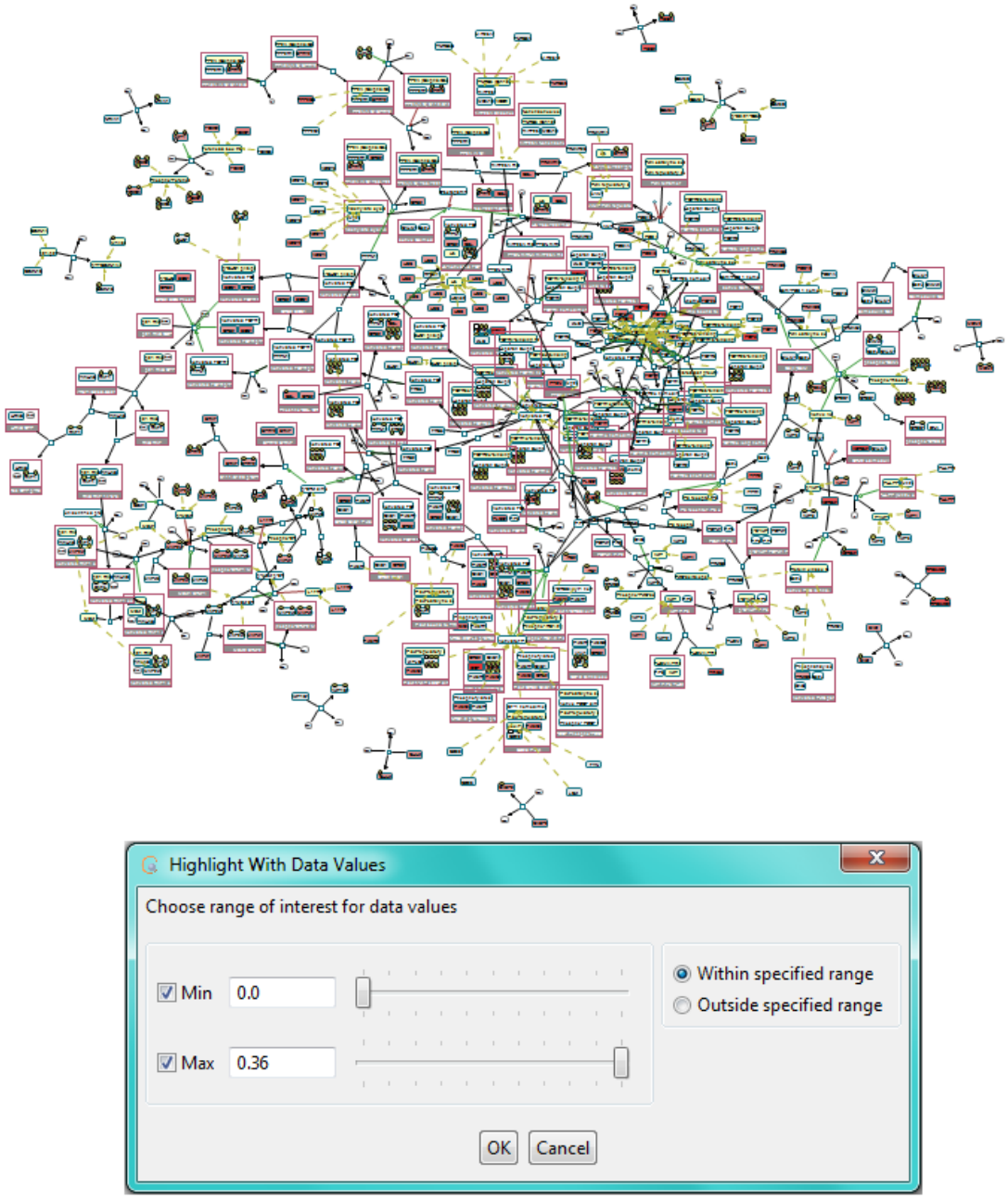


Figure 6.3: *Signaling by FGFR* pathway is integrated with alteration data. "Highlight With Data Values" dialog can also be seen, reflecting the smallest and largest alteration percentages found in the experiment.

properly due to inhibiting genomic events.

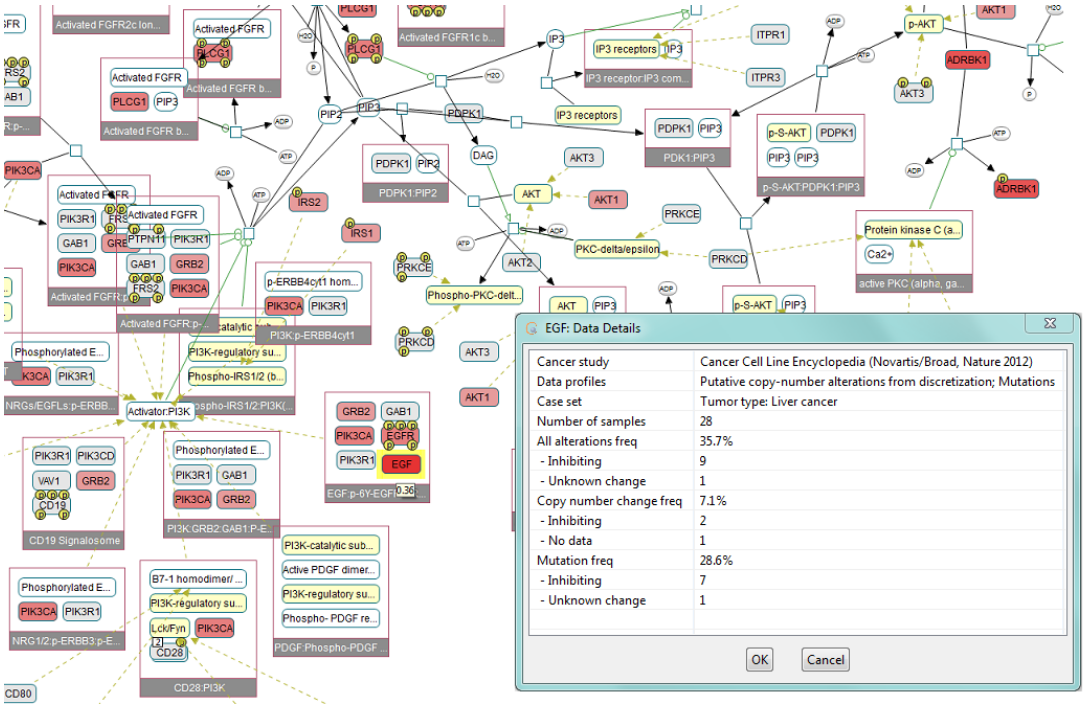


Figure 6.4: EGF, which is highlighted in yellow, is found to be the entity with highest alteration percentage. The details of genomic changes observed in EGF can be seen in "EGF: Data Details" dialog.

At the end of these high throughput experiment result integration steps, we have identified couple of genes that possess significant results within all different genes found in "Signaling by FGFR" pathway. From this point on, one can perform a "Paths Between" query using these three entities to see whether or not there are paths that connect them in a meaningful way, explaining why they are outliers in these experimental results. Alternatively, we can perform a functional annotation analysis in order to get a better understanding of the roles of these proteins and to identify the annotations that are common to them. For this purpose, we can use "Connect To DAVID" feature of ChiBE. Upon selection of the three entities on the view, we choose "Functional Annotaion Summary" option offered by "Connect To DAVID" feature. A subset of the result of this analysis can be seen in Figure 6.5. The results are sorted in ascending order based on Benjamini corrected p-values, placing the most significantly enriched

annotation at top.

Sublist	Category	Term	RT	Genes	Count	%	P-Value	Benjamini
	KEGG_PATHWAY	Prostate cancer	RT		2	66,7	1,8E-2	8,2E-2
	KEGG_PATHWAY	Gap junction	RT		2	66,7	1,8E-2	8,2E-2
	KEGG_PATHWAY	Pancreatic cancer	RT		2	66,7	1,4E-2	8,9E-2
	KEGG_PATHWAY	ErbB signaling pathway	RT		2	66,7	1,7E-2	9,2E-2
	KEGG_PATHWAY	Melanoma	RT		2	66,7	1,4E-2	1,0E-1
	KEGG_PATHWAY	Glioma	RT		2	66,7	1,2E-2	1,1E-1
	KEGG_PATHWAY	Non-small cell lung cancer	RT		2	66,7	1,1E-2	1,3E-1
	KEGG_PATHWAY	Regulation of actin cytoskeleton	RT		2	66,7	4,2E-2	1,6E-1
	KEGG_PATHWAY	Focal adhesion	RT		2	66,7	4,0E-2	1,6E-1
	KEGG_PATHWAY	MAPK signaling pathway	RT		2	66,7	5,3E-2	1,7E-1
	KEGG_PATHWAY	Endometrial cancer	RT		2	66,7	1,0E-2	1,8E-1
	KEGG_PATHWAY	Pathways in cancer	RT		2	66,7	6,5E-2	1,9E-1
	BIOCARTA	Sprouty regulation of tyrosine kinase signals	RT		2	66,7	1,0E-2	2,3E-1
	BIOCARTA	EGF Signaling Pathway	RT		2	66,7	1,8E-2	2,6E-1
	KEGG_PATHWAY	Bladder cancer	RT		2	66,7	8,3E-3	2,8E-1
	BIOCARTA	Keratinocyte Differentiation	RT		2	66,7	2,6E-2	2,8E-1
	BIOCARTA	Map Kinase Inactivation of SMRT Corepressor	RT		2	66,7	7,7E-3	3,2E-1
	GOTERM_BP_FAT	regulation of transferase activity	RT		2	66,7	2,7E-2	4,0E-1
	GOTERM_BP_FAT	positive regulation of transferase activity	RT		2	66,7	1,8E-2	4,2E-1
	GOTERM_BP_FAT	positive regulation of catalytic activity	RT		2	66,7	3,8E-2	4,3E-1
	GOTERM_BP_FAT	regulation of phosphorus metabolic process	RT		2	66,7	3,6E-2	4,3E-1
	GOTERM_BP_FAT	regulation of phosphate metabolic process	RT		2	66,7	3,6E-2	4,3E-1
	GOTERM_BP_FAT	protein kinase cascade	RT		2	66,7	2,7E-2	4,3E-1
	GOTERM_BP_FAT	positive regulation of molecular function	RT		2	66,7	4,3E-2	4,4E-1
	GOTERM_BP_FAT	regulation of phosphorylation	RT		2	66,7	3,4E-2	4,4E-1

Figure 6.5: A portion of the result of functional annotation analysis performed with DAVID using IL17RD, MAP2K1 and EGF

This example demonstrates a way of using different querying and data integration options of ChiBE in order to get a better understanding of a complex biological network in the context of a specific disease. The features described here can be combined in any way that is suitable to the user's needs, in the end to perform a user friendly and thorough analysis of the network of interest.

6.2 Comparison of Experimental Results

ChiBE can also be used to reproduce results obtained through experimental procedures in order to be able to place them into a pathway context, to discover relationships that might have been missed or to compare results of different experiments.

In numerous articles, it has been emphasized that the signaling path from CDKN2A to RB1 proteins is affected by mutations that disrupt its functionality and contribute to development of cancer [32, 33, 34]. Glioblastoma [32] and ovarian serous cystadenocarcinoma [33] are two of the cancer types affected by this dysregulation. Alterations in a common pathway does not necessarily mean that the specific changes in particular elements are the same. So, in this example, we will compare the same path in the context of these two different studies, in order to see whether there are significant differences between the disruption of the same path in different cancers.

For this purpose, we first start by obtaining the path from CDKN2A to RB1. Performing *Paths From To* query on Pathway Commons Database is a proper choice to obtain such a path. Length limit is set to four. Then alteration data for these two cancers are integrated with the pathway view, through *Fetch From cBio Portal* feature. For glioblastoma, "Glioblastoma(TCGA, Nature 2008)" cancer study is selected. As case list, "Tumors with sequencing and aCGH data" is chosen. *Putative copy-number alterations* and *mutations* are selected as genomic profiles. "Ovarian Serous Cystadenocarcinoma (TCGA, Nature 2011)" study is also loaded with the same case list and genomic profiles.

Figure 6.6 reflects the alteration status of glioblastoma samples. CDKN2A has a high rate of alteration (%46.2), which is mostly dominated by inhibiting copy number changes. RB1 has a lower percentage of alteration (%11) and in this case, inhibiting mutations are in the majority. The remaining of the pathway has either none or low alteration, except for CDKN2B. The values for ovarian serous cystadenocarcinoma study can be seen in Figure 6.7. The level of alteration in RB1 is similar to glioblastoma samples but there is a significant reduction in the case of CDKN2A. In only %2.5 of ovarian serous cystadenocarcinoma samples, there is an alteration in CDKN2A. Again, like glioblastoma samples, inhibiting alterations are dominant. From this, one could infer that the dysregulation observed in this path is related to the proteins losing their ability to function as opposed to being over-active.

Compared to glioblastoma samples, the amount of participants of the path

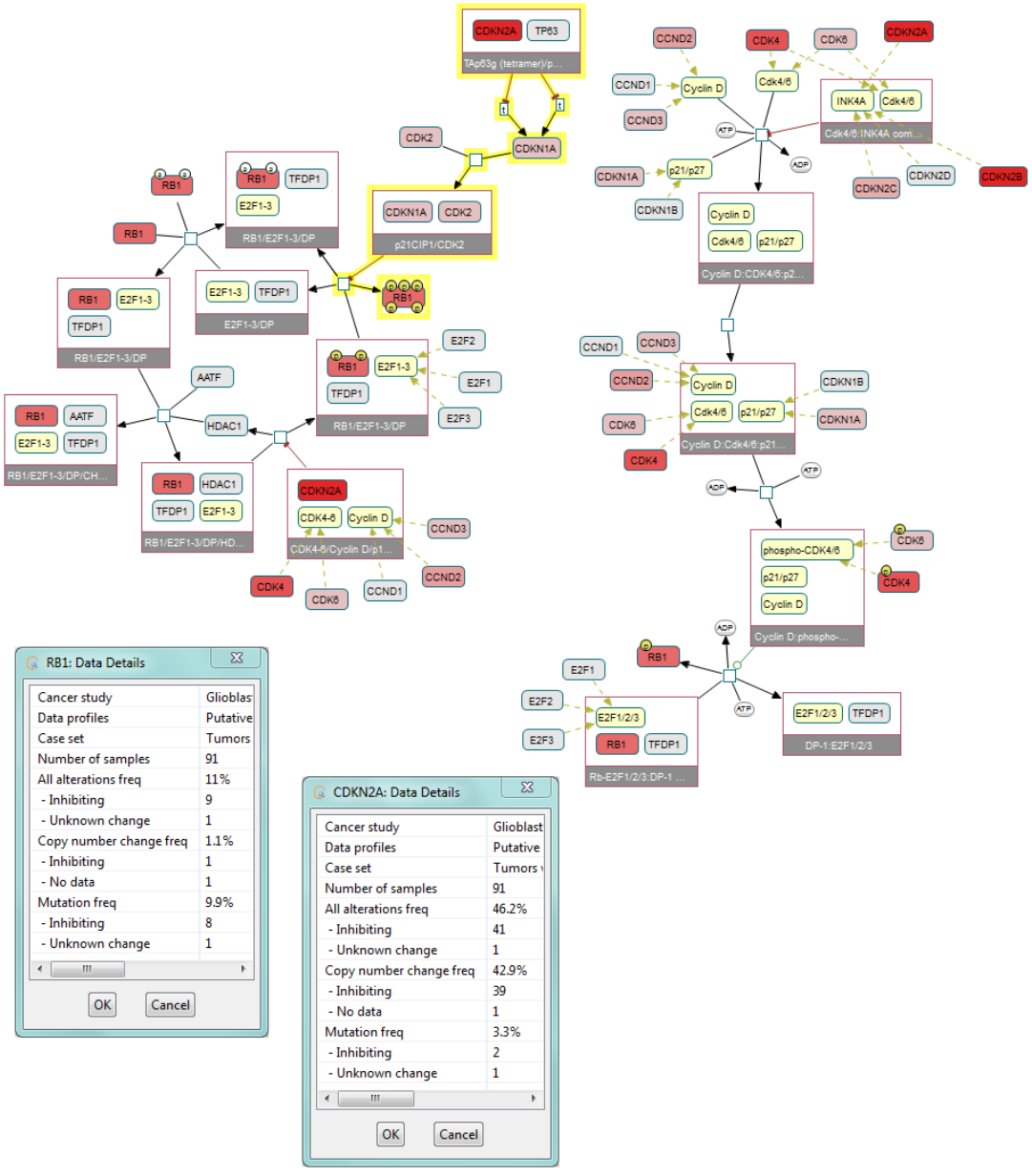


Figure 6.6: Path from CDKN2A to RB1 is integrated with alteration values of glioblastoma samples.

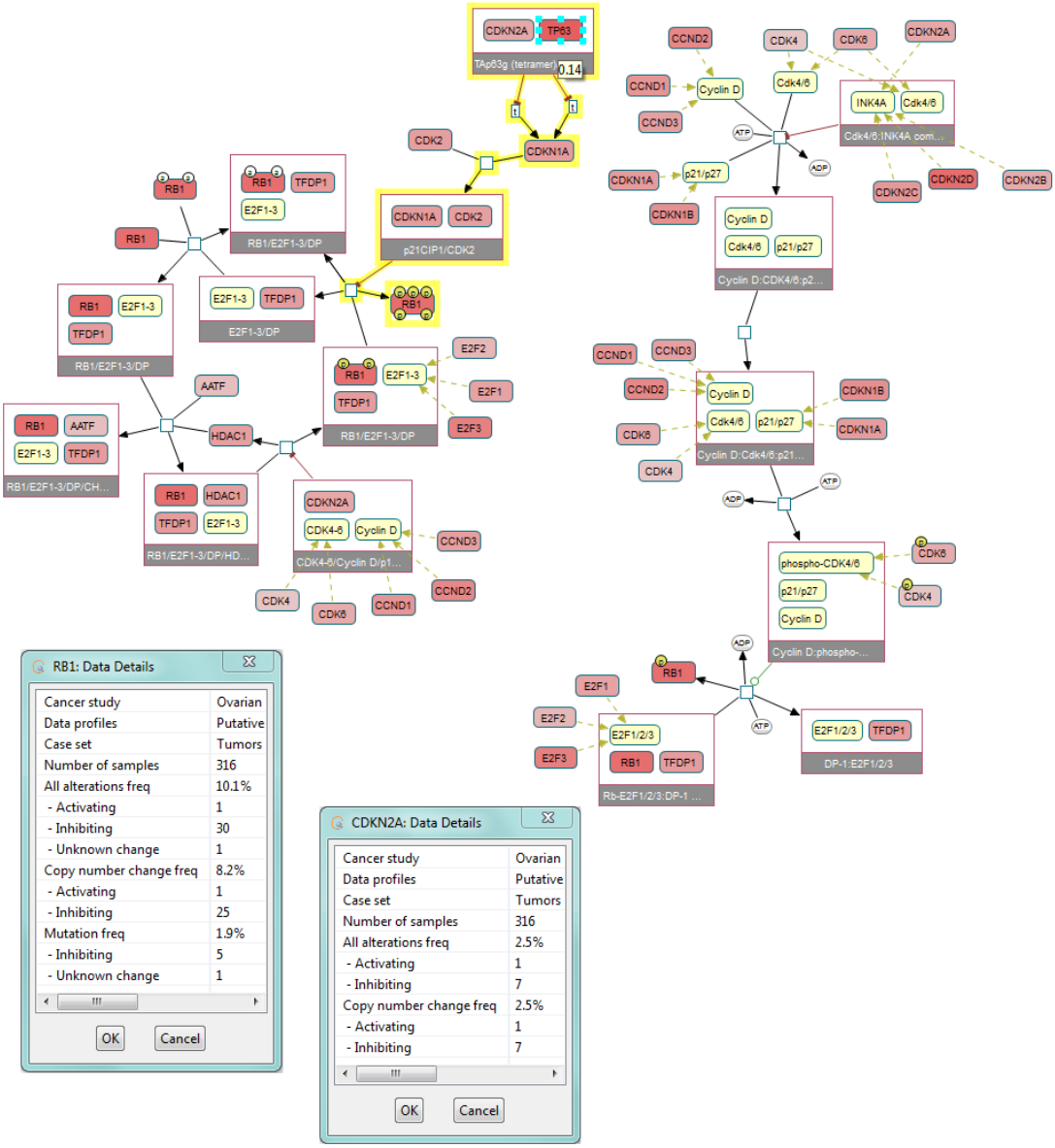


Figure 6.7: Path from CDKN2A to RB1 is integrated with alteration values of ovarian serous cystadenocarcinoma samples.

that are altered is higher in ovarian serous cystadenocarcinoma samples. One point to highlight in these differences is TP63. In glioblastoma samples, no alteration is observed in this gene. However, in ovarian serous cystadenocarcinoma samples, it is altered in %14 of the samples. More importantly, TP63 forms a complex with CDKN2A which has RB1 in its downstream (highlighted in yellow in Figure 6.7). This means that although CDKN2A is altered only in a limited set of samples, the same path can still be disturbed in a higher amount of samples due to the alteration observed in its partner TP63. In the original articles, authors have focused on cyclins and cyclin dependent kinases as the intermediate molecules between CDKN2A and RB1 [32, 33]. However with this pathway analysis, a potential role for TP63 gene is identified. Thanks to ChiBE, we have not only compared two different cancer samples in the context of the same pathway but also identified molecules that can be further investigated while studying these cancer sets.

6.3 Expansion of Experimental Results

In this example, a case where analysis performed in ChiBE can help us identify cases that have been missed during experimental procedures is found [35]. We focus on a research on endometrial carcinoma performed by The Cancer Genome Atlas Research Network [36].

Through genomic, transcriptomic and proteomic profiling of tumor samples, researchers identified a path of interest that has been associated with alterations that might have significance in the cancer development (Figure 6.8). We first start our analysis by reaching to a pathway view that represents its first step, from ERBB2 to KRAS. For this purpose, *Paths From To* Pathway Commons query is used, with length limit 1.

The result of this query can be seen in Figure 6.9. From these paths, one of them specifically gains our interest. The path highlighted in yellow is interesting because it focuses on transfer of a GTP molecule to a RAS family protein and

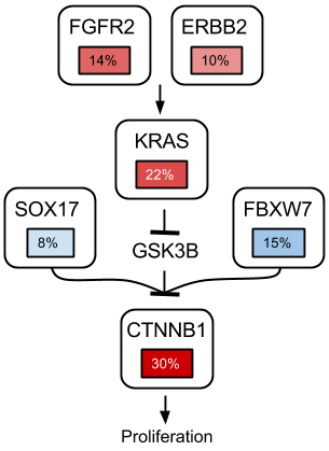


Figure 6.8: A simple network displaying the identified path along with alteration percentages [35]

this reaction is catalyzed by a complex containing ERBB2. To see whether this reaction is really significant and holds an interesting result for the analysis, we load genomic alteration data associated with endometrial cancer. Again, "Fetch From cBio Portal" feature is used to retrieve the genomic status of endometrial cancer cells. Uterine Corpus Endometrioid Carcinoma (TCGA, Nature 2013) is the name of selected cancer study and "tumors with sequencing and aCGH data" is the case list. "Mutations" and "putative copy-number alterations" are selected as genomic profiles as we want to focus on the changes that has occurred at genome level. The results are reflected in the color codes (Figure 6.9).

As it is apparent from the coloring, all members of the complex containing ERBB2 undergo genomic events in case of endometrial cancer. Although only ERBB2 is mentioned in the original study [36], ERBB3's alteration percentage is equal to ERBB2's which is %10. When genomic status of all components of the complex is considered, it can be seen that at least one member of this "ErbB2/ErbB3/neuregulin 1 beta/SHC/GRB2/SOS1" complex is altered in %35 of cancer samples. Confirming their findings, KRAS is found to have a high rate of alteration. In fact, with %22 alteration, it has the largest value in this network.

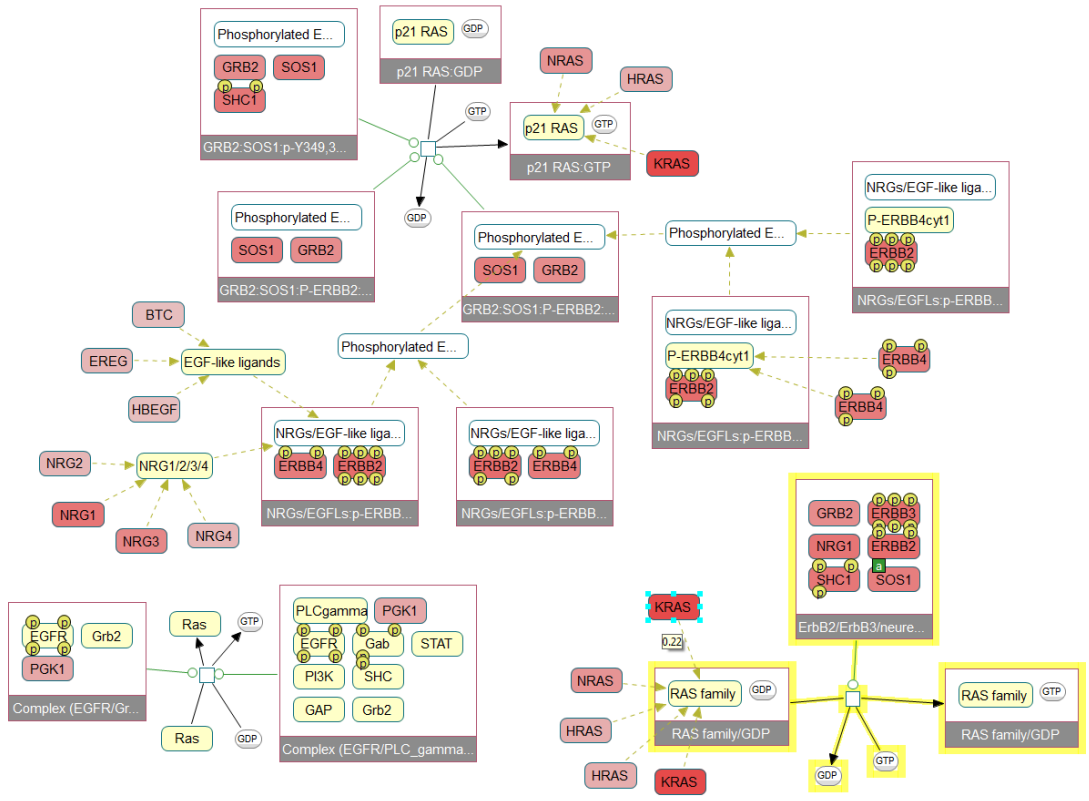


Figure 6.9: The paths from ERBB2 to KRAS with length 1 are integrated with alteration data of endometrial cancer.

With this approach of analysis, we have been able to reproduce the results identified through experiments *in silico*. We have found a path starting from ERBB2 and ending at KRAS with both of the molecules having significant alteration percentages. On top of this, we have identified that 5 additional genes, sharing the same complex structure with ERBB2, also are altered in case of endometrial tumors. This finding have expanded the findings of the original paper as it has identified molecules that are in a similar situation as ERBB2. This can in turn open ways to new hypotheses in which focus of research shifts from ERBB2 to the other genes if ERBB2 is not suitable for handling the problem in hand.

Chapter 7

Conclusion

ChiBE is an open-source tool that is specialized in visualization, manipulation and analysis of BioPAX formatted pathway data [3]. This thesis has focused on the expansion of its capabilities by enriching its querying facilities and improving integration with various genomics-oriented databases.

The large set of available querying options is a valuable addition to the tool as it will help users to enhance their analysis by focusing on specific sets of molecules. The huge and complicated pathway views often confronted during the analysis of critical cellular events pose a problem when a thorough analysis of the view is required. The query options will come in handy in these cases as it will help users to extract knowledge that has been buried in the complex pathway model. Different types of queries ranging from neighborhood to compartment query will be at their service based on their specific needs.

Moving the analysis of high-throughput experiments from gene-specific level to a pathway level can open up new insights into analysis of these experiments. So offering a way of pathway-level analysis is a significant contribution to the range of analysis options available in ChiBE. From wide-range of available experiment types and result sources, we have focused on two popular ones: GEO and the cBioPortal. From GEO, users can easily obtain expression microarray

data and then integrate the results with the pathway view. A similar, easy-to-use connection is also offered to the cBio Cancer Genomics Portal, this time in order to retrieve genomic status of cancer cells. This knowledge is reflected on the pathway view through the percentages of cases, where the particular gene is found to be altered. Both for GEO and the cBioPortal data integration, color codes are used, which makes it easy to distinguish the significant results through their darker coloring. Overlaying these experiment results through color codes enables users to understand spatial organization of significant results and help them to identify causative relationships that arise due to disruptions that has been identified through experiments.

Convenient access to another external database has also been added. DAVID database works on a list of proteins and genes, whose main offering is the annotation of the elements of this set with biological themes, such as Gene Ontology terms. ChiBE now has seamless connection to DAVID with the set of nodes selected on the view. This way, users can identify the biological themes that are associated with their specific set of entities or see which terms are significantly enriched within their set.

Finally, ChiBE's visual notation has been revised to mostly comply with the community standard of SBGN process description notation. As a result, new users who are already familiar with SBGN notations should have less trouble adjusting to ChiBE's visual notation. To stay up-to-date with the developments in standards, support for Level 3 models of BioPAX format [1] is also added. Users now are able to visualize both Level 2 and Level 3 BioPAX models seamlessly.

Collectively, all these improvements have expanded the functionality and scope of the tool. ChiBE is now a unique software that can access popular external databases for pathway data models or experiment results and offer a wide range of analysis tools for pathways stored in standard BioPAX format.

7.1 Comparing ChiBE with Other Tools

At its current state, ChiBE has earned a unique place in pathway visualization and analysis fields, which becomes apparent when its functionalities are compared with similar tools.

CellDesigner [25] can display rich pathway information, similar to ChiBE. However, it cannot work with BioPAX formatted pathway data. It is designed for SBML files. Another disadvantage of CellDesigner is that it does not offer graph-based searches. PathVisio [23] can perform integration of experiment data with pathway view. However, it is limited to expression data and the process heavily depends on user input for data mapping, which can be complicated for most users. PathCase [7] is comparable to ChiBE when its search options are considered, they offer various query options to users. The narrow focus on metabolic pathways, however, limits the search space of these queries. Additionally, PathCase does not support integration with experimental results. Cytoscape [18] visualizes networks with simple interaction views, unless specific plugins are installed. When compared to ChiBE's visual notation representing rich pathway information, this visualization option can offer limited knowledge. Its functionality can be extended with specific plugins, but these plugins only offer portions of ChiBE's wide range of features.

These examples mean that although different tools can offer certain aspects of ChiBE's functionality, none of them can give users the chance of a comprehensive and integrated workflow like the one offered by ChiBE. With this integrated workflow, ChiBE can help researchers to generate hypotheses based on their *in silico* analyses, which can then be tested experimentally for verification.

7.2 Future Work

Although at this point ChiBE offers a fairly large and an advanced set of features, new approaches of analysis can be considered for addition. One point can be

focusing on the properties of these visualized networks. We can look at these networks from a purely topological point of view, to obtain knowledge that is inherent to network structure. For instance, global topological properties of the networks, such as *degree distribution* or *clustering coefficient*, can be calculated to get a grasp of the general organization of the network. They are also useful for comparing different networks. To get a more detailed look at the local structure of the network, centrality measures can be calculated. They can offer knowledge about the critical nodes hidden in the network structure, as these measures can be used to understand importance of nodes within a network relative to each other [37]. Addition of these measures that look into a pathway from a graph theoretical window can bring new insights into the analysis of these biological networks.

Expansion of the available experimental data that can be integrated with pathways could be another trail of future additions. On top of GEO and the cBioPortal, we can provide connections to other data sources. For instance, we can add ways of integrating high-throughput sequencing experiments, such as exome sequencing. Additionally, we can refine the current approach we are using to represent the mutation data with more information. Each mutation will have different effects on the genes and there already are tools that can predict the scale of the damage a mutation can cause on a gene. While integrating mutation data, we can give weights in correlation with the severity of mutations.

As a way of reducing the complexity of the pathway view, we can add filtering features. For instance, after integrating experimental results with the view, users can be enabled to eliminate nodes that are not within the boundaries of a threshold.

Another future direction to follow could be transferring the capabilities of ChiBE to a web-based framework. This will eliminate the requirement for the user to download and install the tool, which will make the tool available to a wider audience.

Bibliography

[1] E. Demir, M. P. Cary, S. Paley, K. Fukuda, C. Lemer, I. Vastrik, G. Wu, P. D'Eustachio, C. Schaefer, J. Luciano, F. Schacherer, I. Martinez-Flores, Z. Hu, V. Jimenez-Jacinto, G. Joshi-Tope, K. Kandasamy, A. C. Lopez-Fuentes, H. Mi, E. Pichler, I. Rodchenkov, A. Splendiani, S. Tkachev, J. Zucker, G. Gopinath, H. Rajasimha, R. Ramakrishnan, I. Shah, M. Syed, N. Anwar, O. Babur, M. Blinov, E. Brauner, D. Corwin, S. Donaldson, F. Gibbons, R. Goldberg, P. Hornbeck, A. Luna, P. Murray-Rust, E. Neumann, O. Ruebenacker, O. Reubenacker, M. Samwald, M. van Iersel, S. Wimalaratne, K. Allen, B. Braun, M. Whirl-Carrillo, K.-H. Cheung, K. Dahlquist, A. Finney, M. Gillespie, E. Glass, L. Gong, R. Haw, M. Honig, O. Hubaut, D. Kane, S. Krupa, M. Kutmon, J. Leonard, D. Marks, D. Merberg, V. Petri, A. Pico, D. Ravenscroft, L. Ren, N. Shah, M. Sunshine, R. Tang, R. Whaley, S. Letovksy, K. H. Buetow, A. Rzhetsky, V. Schachter, B. S. Sobral, U. Dogrusoz, S. McWeeney, M. Aladjem, E. Birney, J. Collado-Vides, S. Goto, M. Hucka, N. Le Novère, N. Maltsev, A. Pandey, P. Thomas, E. Wingender, P. D. Karp, C. Sander, and G. D. Bader, "The BioPAX community standard for pathway data sharing.," *Nature biotechnology*, vol. 28, pp. 935–42, Sept. 2010.

[2] M. Kanehisa, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucleic Acids Research*, vol. 28, pp. 27–30, Jan. 2000.

[3] O. Babur, U. Dogrusoz, E. Demir, and C. Sander, "ChiBE: interactive visualization and manipulation of BioPAX pathway models.," *Bioinformatics (Oxford, England)*, vol. 26, pp. 429–31, Feb. 2010.

[4] S. D. Hooper and P. Bork, "Medusa: a simple tool for interaction graph analysis.," *Bioinformatics (Oxford, England)*, vol. 21, pp. 4432–3, Dec. 2005.

[5] Z. Hu, Y.-C. Chang, Y. Wang, C.-L. Huang, Y. Liu, F. Tian, B. Granger, and C. DeLisi, "VisANT 4.0: Integrative network platform to connect genes, drugs, diseases and therapies," *Nucleic Acids Research*, pp. gkt401–, May 2013.

[6] S. Kozhenkov, Y. Dubinina, M. Sedova, A. Gupta, J. Ponomarenko, and M. Baitaluk, "BiologicalNetworks 2.0–an integrative view of genome biology data.," *BMC bioinformatics*, vol. 11, p. 610, Jan. 2010.

[7] B. Elliott, M. Kirac, A. Cakmak, G. Yavas, S. Mayes, E. Cheng, Y. Wang, C. Gupta, G. Ozsoyoglu, and Z. Meral Ozsoyoglu, "PathCase: pathways database system.," *Bioinformatics (Oxford, England)*, vol. 24, pp. 2526–33, Nov. 2008.

[8] A. Theocharidis, S. van Dongen, A. J. Enright, and T. C. Freeman, "Network visualization and analysis of gene expression data using BioLayout Express(3D).," *Nature protocols*, vol. 4, pp. 1535–50, Jan. 2009.

[9] E. G. Cerami, B. E. Gross, E. Demir, I. Rodchenkov, O. Babur, N. Anwar, N. Schultz, G. D. Bader, and C. Sander, "Pathway Commons, a web resource for biological pathway data.," *Nucleic acids research*, vol. 39, pp. D685–90, Jan. 2011.

[10] N. Le Novère, M. Hucka, H. Mi, S. Moodie, F. Schreiber, A. Sorokin, E. Demir, K. Wegner, M. I. Aladjem, S. M. Wimalaratne, F. T. Bergman, R. Gauges, P. Ghazal, H. Kawaji, L. Li, Y. Matsuoka, A. Villéger, S. E. Boyd, L. Calzone, M. Courtot, U. Dogrusoz, T. C. Freeman, A. Funahashi, S. Ghosh, A. Jouraku, S. Kim, F. Kolpakov, A. Luna, S. Sahle, E. Schmidt, S. Watterson, G. Wu, I. Goryanin, D. B. Kell, C. Sander, H. Sauro, J. L. Snoep, K. Kohn, and H. Kitano, "The Systems Biology Graphical Notation.," *Nature biotechnology*, vol. 27, pp. 735–41, Aug. 2009.

[11] "SBGN PD L1 Examples." `http://www.sbgn.org/Documents/PD_L1_Examples`, Accessed in July 2013.

[12] U. Dogrusoz, A. Cetintas, E. Demir, and O. Babur, "Algorithms for effective querying of compound graph-based pathway databases.," *BMC bioinformatics*, vol. 10, p. 376, Jan. 2009.

[13] R. Edgar, "Gene Expression Omnibus: NCBI gene expression and hybridization array data repository," *Nucleic Acids Research*, vol. 30, pp. 207–210, Jan. 2002.

[14] E. Cerami, J. Gao, U. Dogrusoz, B. E. Gross, S. O. Sumer, B. A. Aksoy, A. Jacobsen, C. J. Byrne, M. L. Heuer, E. Larsson, Y. Antipin, B. Reva, A. P. Goldberg, C. Sander, and N. Schultz, "The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data.," *Cancer discovery*, vol. 2, pp. 401–4, May 2012.

[15] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources.," *Nature protocols*, vol. 4, pp. 44–57, Jan. 2009.

[16] S. Moodie, N. Le Novere, E. Demir, H. Mi, and A. Villeger, "Systems Biology Graphical Notation: Process Description language Level 1," *Nature Precedings*, Feb. 2011.

[17] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction To Algorithms.* The MIT Press, 3rd ed., Nov. 2009.

[18] M. E. Smoot, K. Ono, J. Ruscheinski, P.-L. Wang, and T. Ideker, "Cytoscape 2.8: new features for data integration and network visualization.," *Bioinformatics (Oxford, England)*, vol. 27, pp. 431–2, Feb. 2011.

[19] E. Bonnet, L. Calzone, D. Rovera, G. Stoll, E. Barillot, and A. Zinovyev, "BiNoM 2.0, a Cytoscape plugin for accessing and analyzing pathways using standard systems biology formats.," *BMC systems biology*, vol. 7, p. 18, Jan. 2013.

[20] A. Karnovsky, T. Weymouth, T. Hull, V. G. Tarcea, G. Scardoni, C. Laudanna, M. A. Sartor, K. A. Stringer, H. V. Jagadish, C. Burant, B. Athey, and G. S. Omenn, "Metscape 2 bioinformatics tool for the analysis and

visualization of metabolomics and gene expression data.," *Bioinformatics (Oxford, England)*, vol. 28, pp. 373–80, Feb. 2012.

[21] N. Alcaraz, H. Kücük, J. Weile, A. Wipat, and J. Baumbach, "KeyPathwayMiner: Detecting Case-Specific Biological Pathways Using Expression Data," *Internet Mathematics*, vol. 7, pp. 299–313, Nov. 2011.

[22] D. Merico, R. Isserlin, O. Stueker, A. Emili, and G. D. Bader, "Enrichment map: a network-based method for gene-set enrichment visualization and interpretation.," *PloS one*, vol. 5, p. e13984, Jan. 2010.

[23] M. P. van Iersel, T. Kelder, A. R. Pico, K. Hanspers, S. Coort, B. R. Conklin, and C. Evelo, "Presenting and exploring biological pathways with PathVisio.," *BMC bioinformatics*, vol. 9, p. 399, Jan. 2008.

[24] N. Salomonis, K. Hanspers, A. C. Zambon, K. Vranizan, S. C. Lawlor, K. D. Dahlquist, S. W. Doniger, J. Stuart, B. R. Conklin, and A. R. Pico, "GenMAPP 2: new features and resources for pathway analysis.," *BMC bioinformatics*, vol. 8, p. 217, Jan. 2007.

[25] A. Funahashi, M. Morohashi, H. Kitano, and N. Tanimura, "CellDesigner: a process diagram editor for gene-regulatory and biochemical networks," *BIOSILICO*, vol. 1, pp. 159–162, Nov. 2003.

[26] K. A. Gray, L. C. Daugherty, S. M. Gordon, R. L. Seal, M. W. Wright, and E. A. Bruford, "Genenames.org: the HGNC resources in 2013.," *Nucleic acids research*, vol. 41, pp. D545–52, Jan. 2013.

[27] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.," *Nature genetics*, vol. 25, pp. 25–9, May 2000.

[28] "DAVID API Service." `http://david.abcc.ncifcrf.gov/content.jsp?file=DAVID_API.html`, Accessed in July 2013.

[29] X. Wu and Y. Li, "Signaling Pathways in Liver Cancer," *intechopen.com*, 2011.

[30] Y.-L. Liao, Y.-M. Sun, G.-Y. Chau, Y.-P. Chau, T.-C. Lai, J.-L. Wang, J.-T. Horng, M. Hsiao, and A.-P. Tsou, "Identification of SOX4 target genes using phylogenetic footprinting-based prediction from expression microarrays suggests that overexpression of SOX4 potentiates metastasis in hepatocellular carcinoma.," *Oncogene*, vol. 27, pp. 5578–89, Sept. 2008.

[31] J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, D. Sonkin, A. Reddy, M. Liu, L. Murray, M. F. Berger, J. E. Monahan, P. Morais, J. Meltzer, A. Korejwa, J. Jané-Valbuena, F. A. Mapa, J. Thibault, E. Bric-Furlong, P. Raman, A. Shipway, I. H. Engels, J. Cheng, G. K. Yu, J. Yu, P. Aspesi, M. de Silva, K. Jagtap, M. D. Jones, L. Wang, C. Hatton, E. Palescandolo, S. Gupta, S. Mahan, C. Sougnez, R. C. Onofrio, T. Liefeld, L. MacConaill, W. Winckler, M. Reich, N. Li, J. P. Mesirov, S. B. Gabriel, G. Getz, K. Ardlie, V. Chan, V. E. Myer, B. L. Weber, J. Porter, M. Warmuth, P. Finan, J. L. Harris, M. Meyerson, T. R. Golub, M. P. Morrissey, W. R. Sellers, R. Schlegel, and L. A. Garraway, "The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity.," *Nature*, vol. 483, pp. 603–7, Mar. 2012.

[32] The Cancer Genome Atlas Research Network, "Comprehensive genomic characterization defines human glioblastoma genes and core pathways.," *Nature*, vol. 455, pp. 1061–8, Oct. 2008.

[33] The Cancer Genome Atlas Research Network, "Integrated genomic analyses of ovarian carcinoma.," *Nature*, vol. 474, pp. 609–15, June 2011.

[34] The Cancer Genome Atlas Research Network, "Comprehensive genomic characterization of squamous cell lung cancers.," *Nature*, vol. 489, pp. 519–25, Sept. 2012.

[35] O. Babur, U. Dogrusoz, M. Cakir, A. Aksoy, N. Schultz, C. Sander, and E. Demir, "Integrating biological pathways and genomic profiles with ChiBE 2.," *under preparation*.

[36] C. Kandoth, N. Schultz, A. D. Cherniack, R. Akbani, Y. Liu, H. Shen, A. G. Robertson, I. Pashtan, R. Shen, C. C. Benz, C. Yau, P. W. Laird, L. Ding, W. Zhang, G. B. Mills, R. Kucherlapati, E. R. Mardis, and D. A. Levine, "Integrated genomic characterization of endometrial carcinoma.," *Nature*, vol. 497, pp. 67–73, May 2013.

[37] T. Aittokallio and B. Schwikowski, "Graph-based methods for analysing networks in cell biology.," *Briefings in bioinformatics*, vol. 7, pp. 243–55, Sept. 2006.