Prize-Collecting Steiner Trees in Directeded Signaling Hypergraphs

A Thesis
Presented to
The Division of Mathematics & Natural Sciences
Reed College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts

Barney Isaksen Potter

May 2016

Approved for the Committee
(Interdisciplinary Committee for Mathematics & Biology)

________________________ ________________________

Dr. Anna Ritz Dr. James Fix

Acknowledgments

I want to thank a few people:

- Mom and Dad
- Mari
- Far Far
- Sam, Will, Rich, and ext. family
- Helen and Melvin
- Michael and Caleb
- Anna and Jim
- Sarah, Jeremy, Leigh, and the Schaack Lab
- My benefactor
- A.T & A.L

Preface

We can only see
a short distance ahead,
but we can see plenty there
that needs to be done.

ALAN TURING

Table of Contents

List of Tables

List of Figures

Abstract

The preface pretty much says it all.

Dedication

You can have a dedication here if you wish.

Introduction

Since the discovery of cells by Robert Hooke in 1665, biologists have worked tirelessly to unlock the mysteries of cell function. Over the last half-century, our knowledge of how cells function has grown tremendously. An important and growing part of this field has been the

0.1 Network Representations

0.2 Graph Theory

0.3 The Steiner Tree Problem

0.3.1 Prize-Collecting Steiner Tree Problem

0.4 Integer Linear Programming

Chapter 1

Cell Signaling Networks

Cell function is governed by countless interactions between proteins, nucleic acids, lipids, carbohydrates, and many other small molecules. The interactions between all of these components form what we call *cell signaling networks*, which are responsible for almost every process within a cell. Cell signaling networks are used to describe how many of the most basic reactions within cells cause propagations of information, and the results of thse signals. Some of the most important types of interactions that we see in cell signaling networks (also refered to as protein-protein interaction or signal transduction networks) include the assembly and destruction of protein complexes, how small molecules such as ATP interact with proteins, the cascade of events that can occur after a membrane-bound protein is bound by a ligand, or where negative feedback loops exist that can have an effect on cell function. We often refer to these cell signaling networks at protein-protein interaction (PPI) networks. Historically, PPI networks have been a useful tool for compiling knowledge about individual interactions that have been studied *in situ*, so that larger scale patterns of interaction can be examined, and both communicated easily and analyzed algorithmically. In recent years, there has been a push to find ways to accurately model these networks so that they can be used to predict potential areas of future research (CITATION), in particular, by using graph-based methods (Aittokallio & Schwikowski (2006)).

Generally speaking, there are a few key steps to a cell signling network. First, a trans-membrane protein will undergo a conformational change in response to a particular ligand. Typically, this signal will be some sort of messenger molecule to which the cell needs to respond. The change in conformation results in a cascade of interaction between proteins and other small molecules within the cell. Some of the changes that can result from this cascade of signaling are protein complex assembly, complex degradation, conformational changes to other proteins, phosphorylation, or dephosphorylation (among many other reactions). The ultimate result of this signaling cascade is a change in one or more transcription factors, proteins that bind to DNA, and regulate the rate of transcription of DNA to mRNA.

Signaling pathways are of particular importance because they are often disregulated in heterogenous diseases such as cancer. Because of this, researchers' ability to obtain accurate information about signaling pathways, and their ability to interpret that information may help them to deduce interactions whose disregulation may be implicated in a particular disease. As these networks grow and become more accurate, we will become increasingly able to develop computational methods to mine these networks for novel areas to research, that may help explain or treat heterogenous diseases. Additionally, as the databases become more complete, cross-pathway interactions may begin to be documented that would not otherwise be apparent upon studying individual pathways. The development of new computational techniques to elucidate regulatory changes in areas where different pathways, once modeled as discrete, crosstalk. If areas like these, could be identified, they could lead to new hypotheses that researchers could investigate *in situ*. Furthermore, while past research has found many individual proteins and pathways that are implicated in particular diseases, our ability to observe or quantify certain elements that could be playing an important role in pathway disregulation is limited by our current sampling methods (RNA-Seq, microarray, etc). The development of more nuanced algorithms could increase our ability to implicate proteins that we cannot currently observe being disregulated because of the ways that we currently quantify protein expression in cells.

Beyond individual signal transduction pathways, it is useful to think of the set of all known interactions, together. We refer to this object as an *interactome*, and our ability to analyze it computationally could lead to discovery (or at least postulation) of completely novel signaling networks. By examining the interactome for a species, and weighting it appropriately, it may be the case that we are able to elucidate sets of interactions that had been observed separately, but actually to be correlated with each other. A simple case of this would be if we find that one signaling pathway were to lead directly into another, that is, the outputs of the first pathway were the inputs to the other. Though interactions between multiple networks as simple as that are likely to have already been observed, it could be the case that an algorithm could find that there is some "chain" of connections between one network and another, or that there were some form of crosstalk between some of the components of two networks. If we could find this type of interactions, it would help researchers understand the extent to which a change in one network may affect another.

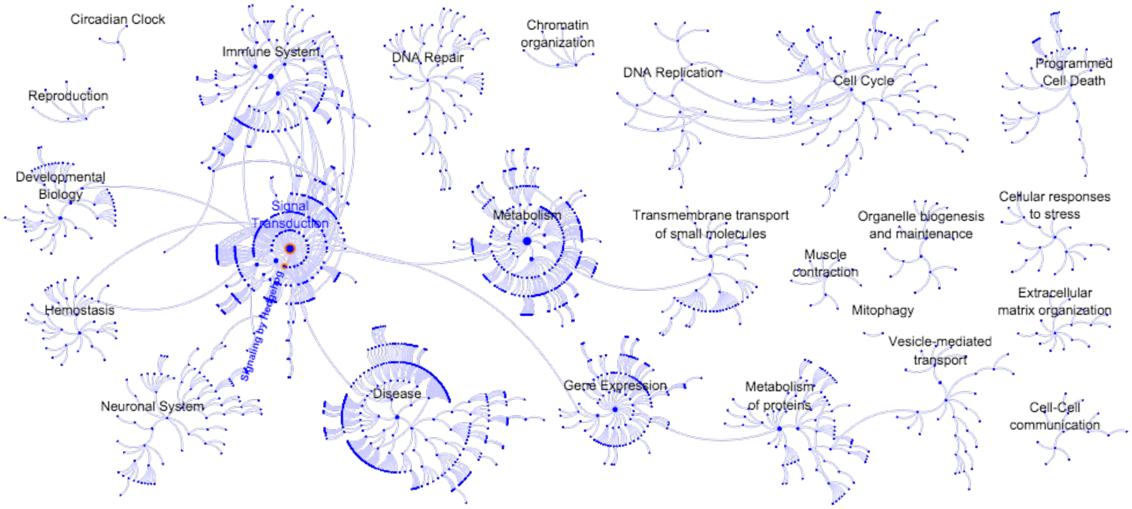


Figure 1.1: The entire human interactome, as displayed by the Reacome interactive pathway browser.

1.1 PPI Databases

In recent years, there has been a push to start curating what is already known about cell signaling, and publishing these networks online (Bauer-Mehren et al. (2009), Cusick et al. (2009)). Many of these databases have been made available to the public, so that the data that they contain can be used collaboratively by anyone. Some of these networks are the Reactome database (`reactome.org`, Matthews et al. (2009)), NetPath (`netpath.org`, Kandasamy et al. (2010)), and SPIKE (`http://spike.cs.tau.ac.il/spike2/` Paz et al. (2011) *Need to fix this URL*).

The purpose of signaling pathway databases is twofold: to create repositories of known interactions, so that they can be easily referred to viewed in a zooomable, searchable manner (Hu et al. (2007)), and so that researchers can take advantage of the computational tools that alerady exist to find novel areas of research from these manually-curated networks (Karlebach & Shamir (2008), Battle et al. (2010)).

1.2 Signaling by *Hedgehog*

I decided to go with Hh, because a. it is a less complicated pathway, so it is easier to discuss without leaving a ton out, and b. because it has more orthologs in A. thaliana. Also, I realized that Hh disregulations are most closely associated with the kind of skin cancer that I had.

One example of a cell signaling pathway that has a variety of biological consequences is the *Hedgehog* (Hh) signal transduction network. Hedgehog is a protein that helps regulate limb formation during early development, cell development and differentiation, and the development of neural tubes (Hui & Angers (2011)). Hedgehog has been implicated in the development of basal cell carcinoma (BCC[1]) when it is overexpressed (**?** *Issue importing bib reference. Need to fix.*). Additionally, it has been shown that Hh has very powerful effects on the proper layout and development of tissues in mammals, and it has been proposed that Hh is responsible for assisting with stem cell assisted tissue regeneration in adult mammals (*CITATION*).

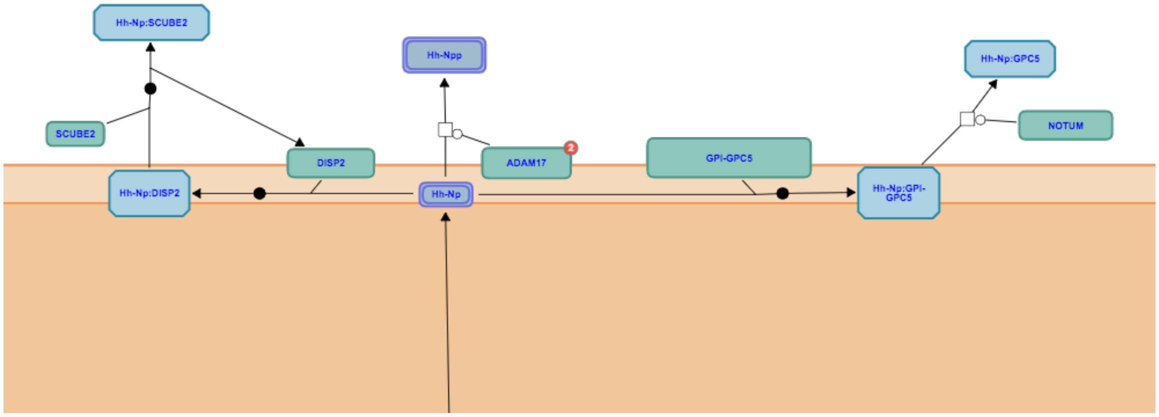


Figure 1.2: Hh-Np secretion, as shown by `reactome.org` in *Homo sapiens*.

The Hedgehog signaling pathway makes a useful example for modeling graph (or hypergraph) algorithms, since the network is small enough to look at manually, but also complex enough that its analysis is nontrivial. Furthermore, because it has been implicated with BCC, along with many other forms of cancer, data for weighting signaling networks are available through public databases such as The Cancer Genome Atlas (`http://cancergenome.nih.gov/`). These factors make it an excellent choice as an example network for testing new hypergraph algorithms.

[1] Basal cell carcinoma is thought to be the most common form of cancer in humans.

1.3 Signaling Networks in *Arabidopsis thaliana*

In addition to humans, there is a wealth of data available for other species that can also be studied using PPI database mining. One such species is *Arabidopsis thaliana* (mouse-ear cress), a plant in the mustard family that is commonly used as a model for plant genetics. Researchers often use *A. thaliana* to examine the effects of genetic manipulation on plants. Because of this, there is a huge quantity of data available that make implementing the same kind of algorithms that are possible in human signaling pathways in *A. thaliana*.

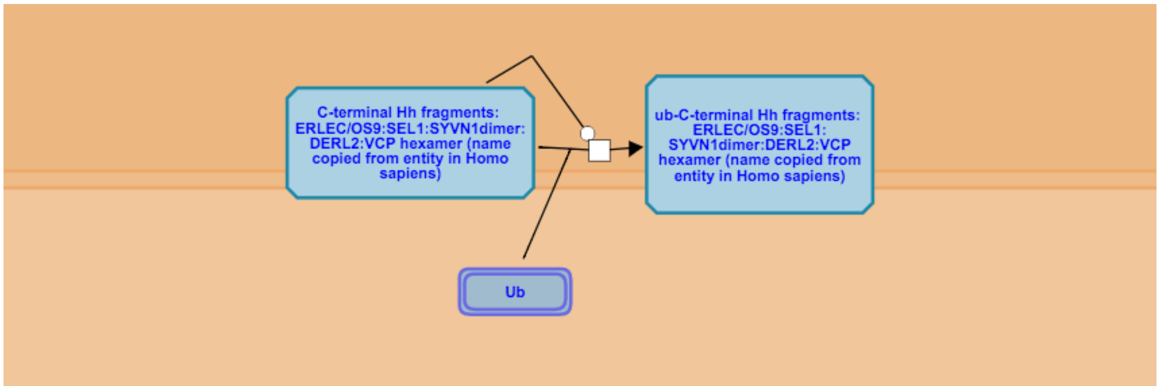

C-terminal Hh fragments: ERLEC/OS9:SEL1:SYVN1dimer: DERL2:VCP hexamer (name copied from entity in Homo sapiens)
ub-C-terminal Hh fragments: ERLEC/OS9:SEL1: SYVN1dimer:DERL2:VCP hexamer (name copied from entity in Homo sapiens)
Ub

Figure 1.3: The Reactome Hedgehog signaling pathway in *A. thaliana*. For this pathway, and many others, the Reacome database is very sparse relative to the orthologous pathways in humans, and is therefore not very useful for implementing network algorithms, at this point.

Unfortunately, for many small-scale pathways (such as the Hedgehog pathway), there is not as much pathway data available in pathway databases, since much more research has been done on humans than on any other organism. Fortunately, however, there is enough pathway data available that the overall interactome can be searched and may yield useful outcomes.

Our ability to look at the entire interactome of species such as *A. thaliana*allows us to investigate some of the global effects of genetic manipulation on signal transduction. This is especially useful in model organisms like *A. thaliana*, since we can actually perform genetic manipulations on them quickly, safely, and ethically, unlike with human beings. Additionally, our ability to breed model organisms in the lab means that we can use specific datasets with large sample sizes. This is very useful compared to human datasets, which are generally based on small sample sizes, since human gene expression data is generally collected from (more or less) random, sparse cases when a person has a degenerate tissue type, such as a tumor, rather than under direct

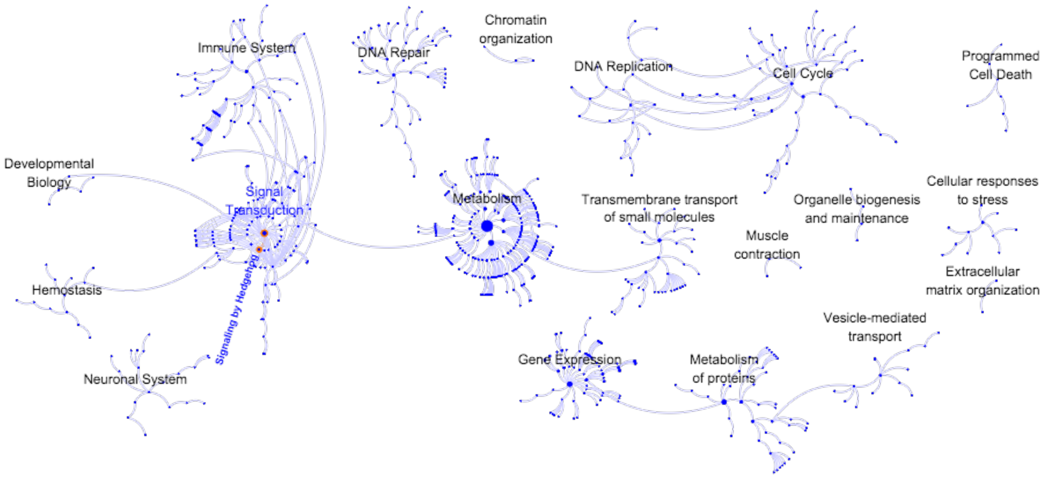


Figure 1.4: The entire *A. thaliana*interactome, as displayed by the Reacome interactive pathway browser. This interactome is comparably dense to the human interactome seen in 1.1, which makes it a better candidate for running network algorithms.

experimental manipulation.

This needs a good transition, so that it isn't so jarring to the reader.

Chapter 2

Hypergraphs & Hyperpaths

While standard[1] graphs are useful for many applications, they are severely limited in their ability to represent cell-signaling interactions. Since they can only show pairwise interactions between nodes, whenever there is an interaction that requires more than two connections, the visualization of the graph becomes confusing, and biologically meaningless. Furthermore, interactions involving multiple molecules require an enormous amount of different edges to represent all of the sub-interactions that take place. Determining whether two edges are part of the same biological event is a non-trivial problem, and requires manual curation to solve, at this time.

One area of cell-signaling that becomes particularly problematic in standard graphs is the formation, interaction, and destruction of protein complexes. The only way that complexes can be represented in standard graphs is by creating a complete subgraph of all of the elements of the protein complex. On their own, these complete subgraphs can yield useful information about the make-up of a protein complex, but once they begin interacting with other elements of the graph, the graph becomes much more complex, as all of the proteins in the complex must be represented independently. In the case of interactions between multiple large complexes, it becomes the case that the standard graph representation of this interaction is a large complete subgraph that contains nodes for all of the proteins involved in either complex. It is then computationally impossible to distinguish whether the entity being described by the subgraph is the interaction between multiple protein complexes, or simply one large complex that contains all of the components of both complexes. Furthermore, since the complete subgraphs which represent complexes are undirected, it is extremely difficult to tell what the inputs or outputs of a biochemical reaction may be.

[1] The term “standard” is used to describe traditional graphs, since the terms “regular” and “normal,” which would be natural choices, both refer to specific types of graphs.

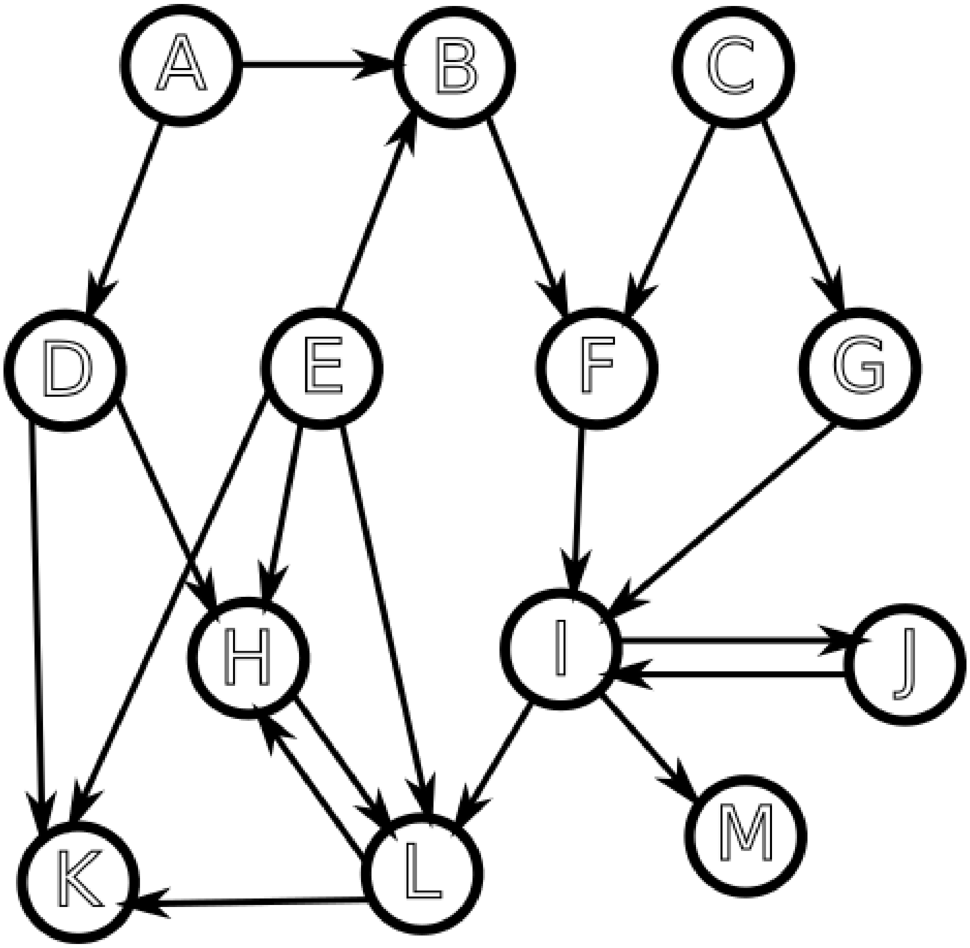


Figure 2.1: An example of a standard graph. All edges represent directed, pairwise interactions between two nodes. Biologically, it is difficult to extract meaningful signaling pathway information from this graph, since protein complex formation and degradation is ambiguous.

Another shortcoming of standard graphs in representing complex biological interactions is that there is no way in which to represent positive or negative regulation of interactions. Since there is a standardized way in which edges interact with nodes, there is no way to differentiate types of interactions between nodes. This poses a challenge when there are regulators or catalysts present that are necessary for a reaction, but are not part of the inputs or outputs of the reaction. If regulators are to be included in a standard graph representation of a cell-signaling network, they become indistinguishable from any other types of interactions that are taking place. This lack of specificity is problematic, as it treats all interactions as equal, and hides potentially useful information from the graph.

To resolve the issues presented by standard graphs, we instead use a generalization called a *hypergraph* that allows for the addition of more detail and specificity within the data structure than standard graphs allow for. In particular, hypergraphs allow for both the representation of protein complexes in the form of *hypernodes*, which we think of simply as a set of one or more nodes, and for the representation of complex, directed interactions that can have multiple inputs and outputs. We represent these interactions with the use of *hyperedges*, which define a set of one or more hypernodes. Since a hyperedge may include more than two hypernodes, we gain the ability to represent both multi-protein interactions, as well as to define the notions of regulation on reactions.

2.1 Prize-Collecting Steiner Hypertrees

Where a directed graph represents directed, pairwise interactions between only two vertices, we can use *directed hypergraph* to represent directed interactions between sets of vertices (nodes). We formally define a directed hypergraph, $\mathcal{H}$, as a pair (V, E), where V is a finite set of vertices and $E \subseteq 2^V \times 2^V$ is a finite set of *directed hyperedges* connecting members of V such that, for every $e = (T(e), H(e)) \in E$, $T(e) \cap H(e) = \emptyset$, and $T(e)$, $H(e) \neq \emptyset$. We refer to $T(e)$ as the *tail* of the hyperedge, and to $H(e)$ as the *head* of the hyperedge.

Furthermore, we can define sets of two or more nodes as a hypernode, which are members of the power set of V. These can be incorporated into a directed hypergraph to form a *complexed directed hypergraph*. We define a *complexed directed hypergraph*, $\mathcal{H}$, as the triple (V, U, E) in which V is a finite set of vertices, $U \subseteq 2^V$ is a finite set of hypernodes, and $E \subseteq 2^U \times 2^U$ is a finite set of hyperedges such that, for every $e = (T(e), H(e)) \in E$, $T(e) \cap H(e) = \emptyset$ and $T(e)$, $H(e) \neq \emptyset$.

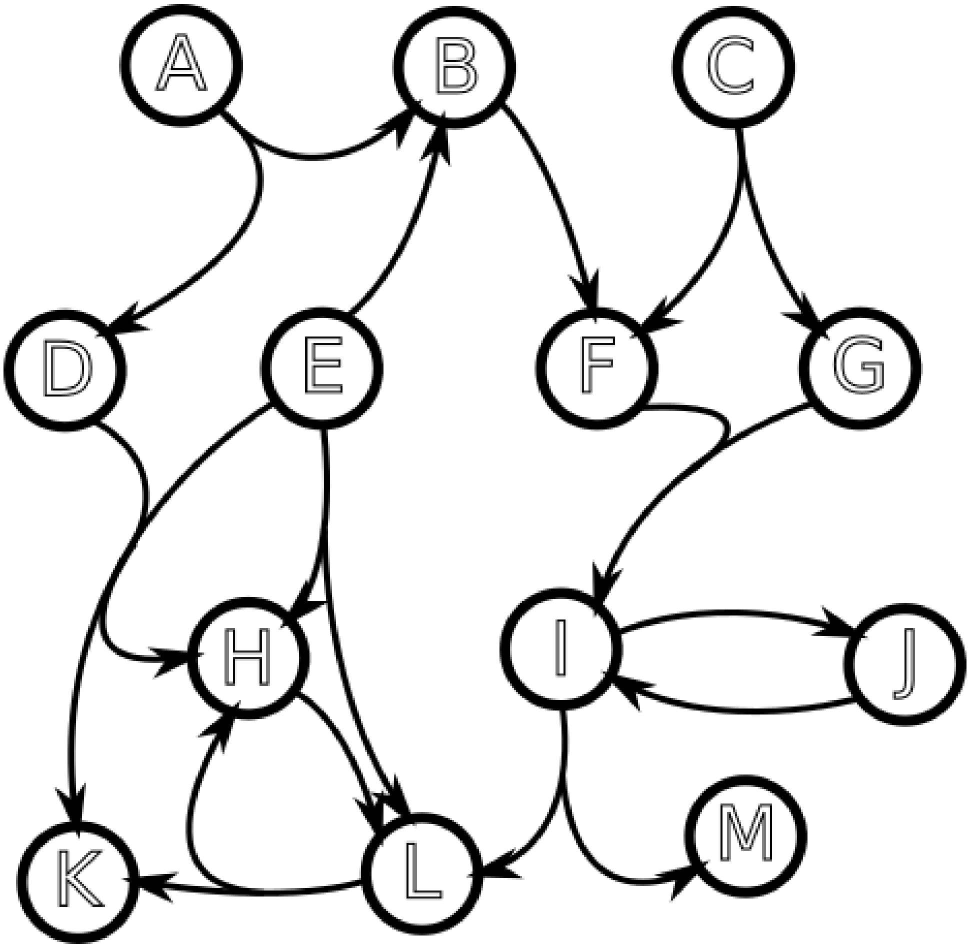


Figure 2.2: An example of the hypergraph that corresponds to the standard graph shown in Figure 2.1. Note that each edge now contains more specific information about the flow of information *information?* through the network.

It is important to note that a standard directed graph is a special case of a directed hypergraph. This is the case if every hypernode in the graph contains only one element, and if each edge has exactly one head element, and one tail element. This has two important implications for algorithms that run on directed hypergraphs. First, this means that there is no loss in functionality caused by using a hypergraph representation of a cell network, since anything that could be computed on a standard directed graph can be recreated exactly using the special case of the hypergraph. Secondly, this is important, because it means that anything that can be computed on a standard graph will be at least as computationally difficult to compute when generalized to a hypergraph. In fact, we find that many tasks that are computationally easy on standard graphs become very difficult when generalized to hypergraphs (Ritz et al. (2014b)).

2.2 Hyperpaths & Connectivity

In order to find the shortest route between two vertices, in a directed hypergraph, we define the notion of a *hyperpath*, P, on the directed hypergraph $\mathcal{H}$. We think of $P(s,t)$ as the list of vertices and hyperedges that one must pass through in order to traverse through the directed hypergraph from vertex s to t. The existence of a hyperpath between two nodes encodes the notion of “connectivity” between those nodes. If a hyperpath exists between nodes s and t, we say that they are *connected.* Furthermore, given some root hypernode, u, we refer to the set of all nodes, V in a hypergraph for which there exists some $P(u,v)$ (where $v \in V$) as a *hypertree.* This is a useful definition, because it allows us the notion of a *connected hypergraph*, a hypergraph in which every node is connected by some hyperpath to every other node in the hypergraph.

2.2.1 Directed Hyperpaths

There are many ways in which we can define hyperpaths, but for the purposes of finding Steiner Trees in directed hypergraphs, we must define a simple *directed hyperpath* between nodes s and t. We can think of a directed hyperpath $P(s,t) = \{s, e_2, v_3, ..., e_{n-1}, t\}$ as an ordered list of nodes and edges, beginning with s and ending with t such that for any edge, e_i, in the path, v_{i-1} is in the tail of e_i, and v_{i+1} is in the head of e_i.

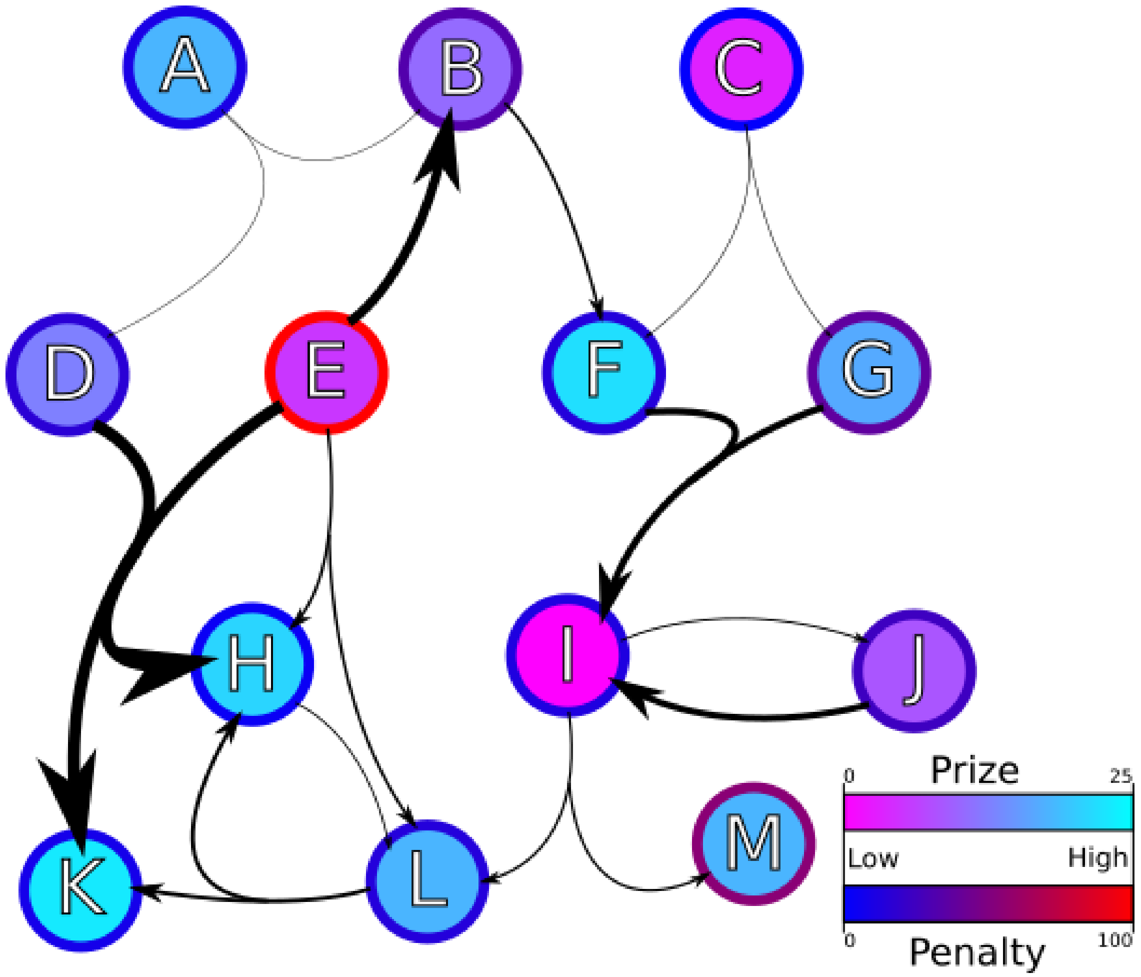


Figure 2.3: The same hypergraph as in 2.2, now weighted with node prizes (inner color) and dangling penalties (outer color). Edge thickness corresponds with edge weight. Intuitively, to find the PCSHT for this hypergraph, we want to find the subnetwork that maximizes the amount of blue, and minimizes the amount of both red nodes and thick lines.

2.3 Definition of PCSHT

We now can define the Prize-Collecting Steiner Hypertree (PCSHT), $\mathcal{S} = (V', E')$, to be the set of hypernodes and hyperedges for which total vertex prizes are maximized, and total edge weights (and dangling penalties) are minimized. The Steiner Hypertree produced should satisfy the following properties:

- There should be no simple cycles between nodes in $\mathcal{S}$.
- The value of all prizes included in $\mathcal{S}$ should be maximal.
- The solution should minimal with respect to edges. *(Should this be defined?)*
- The solution should consist of one or more parallel subnetworks. *(Define this in the HYPERPATHS AND CONNECTIVITY section)*
- If an edge is included in the solution, every hypernode that is incident on that edge should also be included.

In addition to these properties, it is also necessary for us to define two classes of nodes that may be present in a PCSHT: danging nodes and chads (*names subject to change*). We say a node is *dangling* if it is present in a PCSHT and does not have any incoming hyperedge in the PCSHT (that is, it is not in the tail of any edge in the hypergraph). We can define a dangling node, d as any node such $d \in V'$ and $d \notin T(e)$ for all $e \in \mathcal{S}$. Similarly, we say a node is a *chad* if it is present in a PCSHT, and not in the head of any edge in the hypergraph. A chad, c, is defined as any node $c \in V'$ and $c \notin T(e)$ for all $e \in \mathcal{S}$.

PUT IN AN IMAGE SHOWING DANGLERS AND CHADS

Chapter 3

Formulation of ILP

In order for us define and develop a Prize-Collecting Steiner Hyptertree, we begin by formulating the notion of a *Steiner Hypertree.* This allows us to develop an implementation that where some nodes, referred to as our *target set*, are included in the solution, by definition. We begin like this to make the development and debugging of the ILP slightly easier, since it allows us to work from the target set to other nearby nodes, instead of forcing the ILP to discover every node that it needs to include.

To make our Steiner Hypertree, $\mathcal{S}$, we begin with a "parent" hypergraph, $\mathcal{H} = (V, E)$, and a set of target nodes, T, such that $T \subseteq V$. We refer to each hypernode as some x such that $x \in V$[1], and each edge is some e such that $e \in E$. When building $\mathcal{H}$, we initially seed the hypergraph. Each hyperedge is assigned a weight, that is, a cost associated with including that edge in the solution hypergraph, $\mathcal{S}$. We then assign each hypernode in $\mathcal{H}$ two values, a prize for being included in $\mathcal{S}$, as well as a penalty associated with being a *dangling* hypernode.

3.1 Steiner Hypertree ILP

To solve for solutions to the Prize Collecting Steiner Tree

Given an input hypergraph $\mathcal{H} = (V, E)$, and a set of target nodes, T, we construct a Steiner Hypertree, (DEFINE) $\mathcal{S} = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$. We build $\mathcal{S}$ using an ILP which encodes the definition of a Steiner Hypertree. In order to accomplish this, we define three indicator variables, α_v, α_e, and δ_v, where v and e are hypernodes or hyperedges in $\mathcal{H}$. If hypernode x is in the solution of the ILP, α_v will have a value of 1, otherwise it will be equal to 0. Similarly, if edge e is present in the

[1] We use "x" as an element of V, rather than lowercase "v" to avoid the visual confusion between lowercase v and uppercase V (REMOVE THIS IF CHANGING TO V'S LOOKS OKAY)

solution, α_e will take a value of 1. The value of δ_v will be determined by whether a hypernode is dangling.

We find the Steiner Hypergraph $\mathcal{S}$ by optimizing the function:

$$\operatorname*{argmax}_{\alpha,\delta} \sum_{v \in V} g_v \alpha_v - \sum_{e \in E} c_e \alpha_e - \sum_{v \in V} h_v \delta_v \tag{3.1}$$

Subject to the set of linear constraints (*Would it be a good idea to change these so that they all use $\leq$, rather than a mix of $\leq$ and $\geq$?*):

$$\alpha_v \geq 1 \qquad \forall\, v \in T \tag{3.2}$$

$$\sum_{v \in H(e)} \alpha_v \geq |H(e)|\alpha_e \qquad \forall\, e \in E \tag{3.3}$$

$$\sum_{v \in T(e)} \alpha_v \geq |T(e)|\alpha_e \qquad \forall\, e \in E \tag{3.4}$$

$$\delta_v \leq \alpha_v \qquad \forall\, v \in V \tag{3.5}$$

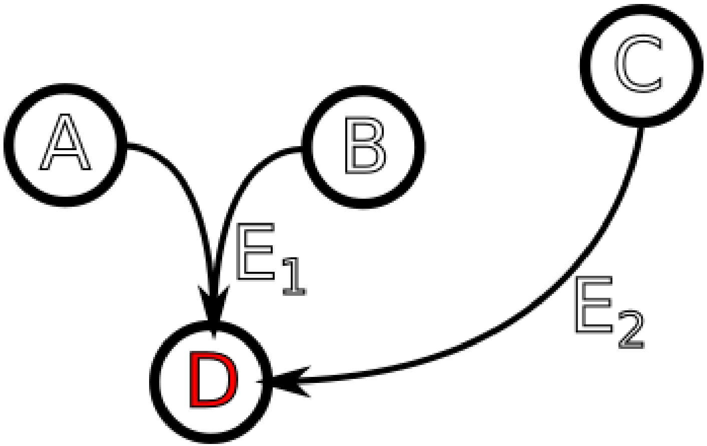
$$\delta_v \geq \alpha_v - \sum_{e : v \in H(e)} \alpha_e \qquad \forall\, v \in V \tag{3.6}$$

$$0 \leq \delta_v \leq 1 \qquad \forall\, v \in V \tag{3.7}$$

Here, the objective function (3.1) tries combinations of nodes (α_v), edges (α_e), and dangling nodes (δ_v) that maximize the sums of node prizes (g_v), minimize the sums of edge costs (c_e), and minimize the sums of dangling penalties (h_v). Constraint (3.2) encodes that every target node in T is in the solution, $\mathcal{S}$. Constraints (3.3) and (3.4) ensure that if an edge is in $\mathcal{S}$, any nodes incident on that edge (i.e. in the head or tail of that edge) will also be in $\mathcal{S}$. Finally, constraints (3.5), (3.6), and (3.7) encode the ability for nodes to dangle if they do not have an incoming edge.

Should I start a new subsection here?

Consider the hypergraph shown in Figure 3.1. For this hypergraph, the PCSHT that comes from this graph should consist of hypernodes A, B, and D, as well as hyperedge E_1. This is obvious, since nodes A and B, which will be dangling in the solution, both have low dangling penalties and high prizes, while node C has a very large dangling penalty relative to its prize. Similarly, E_1 has a relatively low cost, whereas E_2 has a very high cost. Finally, D is considered a target, in this case, and is therefore automatically included in the solution. Knowing what the solution should be, we can use this hypergraph to demonstrate the effect of each linear constrain in our ILP.



Nodes

Node	g_v	h_v
A	10	1
B	10	1
C	1	10
D	10	10

Edges

Edge	c_e
E_1	1
E_2	10

Figure 3.1: A simple hypergraph, and its associated node and edge weights. *This obviously needs to be re-formatted to not look terrible.*

We now know that, for the optimal PCSHT, α_A, α_D, α_D, and α_{E_1} are all equal to 1 (that is, they are included in the solution), and α_C and α_{E_2} are equal 0. Additionally, we know that in this solution A and B are dangling nodes, therefore δ_A and δ_B are both 1, and δ_C and δ_D are both 0. Finally, we are given that D is a target node, hence $T = \{D\}$.

First let us look at the objective function, Equation (*not actually an equation. Does this matter?*) (3.1). We can call each of the three sums included the equation which calculate the total node prizes, edge costs, and dangling penalties Ξ, Φ, and Ψ, respectively. Given the values of α_v, α_e, and δ_v that we know should yield the PCSHT, we get find:

$$\begin{aligned} \Xi &= 30 \\ \Phi &= 1 \\ \Psi &= 2 \\ \implies \Xi - \Phi - \Psi &= 27 \end{aligned}$$

If C or E_2 were to be added, or any node or edge were removed from the PCSHT, the total value of the objective function would decrease, therefore yielding a suboptimal solution to the ILP.

Now, we can begin to look at how each linear constraint governs the behavior of the ILP. Let's begin with constraint (3.2). For our hypergraph, T only has one element, D, constraint (3.2) only needs to be checked for one node. We know that $\alpha_D = 1$, therefore constraint (3.2) simplifies to:

$$\begin{aligned} \alpha_D &\geq 1 \\ 1 &\geq 1 \end{aligned}$$

We see that constraint (3.2) holds for all v in T, therefore we know that all members of the target set are included in the PCSHT.

Next, we can look at constraints (3.3) and (3.4). Since these constraints are the same, except for whether they are concerned with the head or tail of a hyperedge, we can evaluate the effect of just constraint (3.3), and assume that constraint (3.4) works in the same way (*Is this an okay assumption? Should I change it to look at the tails since there is a little more going on with those?*). To assess this constraint, we must see how if the inequality holds for all edges in the hypergraph. We begin by looking at E_1. We know that $H(E_1) = D$, and that $|H(e)| = 1$, therefore we can check:

$$\begin{aligned} \alpha_D &\geq 1 \times \alpha_{E_1} \\ 1 &\geq 1 \times 1 \\ 1 &\geq 1 \end{aligned}$$

So, we see that constraint (3.3) holds for E_1. Now, we can check for E_2 (whose head is also only D):

$$\begin{aligned} \alpha_D &\geq 1 \times \alpha E_2 \\ 1 &\geq 1 \times 0 \\ 1 &\geq 0 \end{aligned}$$

We see that this inequality also holds. This means that, since the inequality holds for all hyperedges, that for every edge in the hypergraph, its head is also included in the hypergraph. We can easily extend this to see that (3.4) enforces the same for the tails of every hyperedge.

Finally, we can look at the three constraints that allow hypernodes to be dangling: constraints (3.5), (3.6), and (3.7). First, we look at constraint (3.5) *I need to figure out the proper way to align these.*:

$$\begin{array}{cccc} \delta_A \leq \alpha_A & \delta_B \leq \alpha_B & \delta_C \leq \alpha_C & \delta_D \leq \alpha_D \\ 1 \leq 1 & 1 \quad \leq 1 & 0 \quad \leq 0 & 0 \quad \leq 1 \end{array}$$

Now, we see that constraint (3.5) holds for all nodes in the PCSHT. *I will get the other two constraints up soon.*

3.2 Danglers and Chads Problem

When we implement the ILP in a more complex hypergraph, such as the hypergraph shown in Figure 2.3, we find that it creates a hypergraph that consists of multiple smaller hypergraphs (Figure 3.2), rather than a continuous subnetwork. Our original implementatioin ends up returning a group of disconnected hypergraphs that represent regions with high prize to edge/dangling cost ratios, separated by gaps where they parent hypergraph had regions of low prize to edge/dangling cost ratios. While this result is intersting, in itself, it will not necessarily create a soltion that is biologically interesting, once we begin looking for PCSHTs in pathway signaling data.

To fix this problem, we must incorporate the notion of hyperpaths and connectedness to our implementation of the PCSHT.

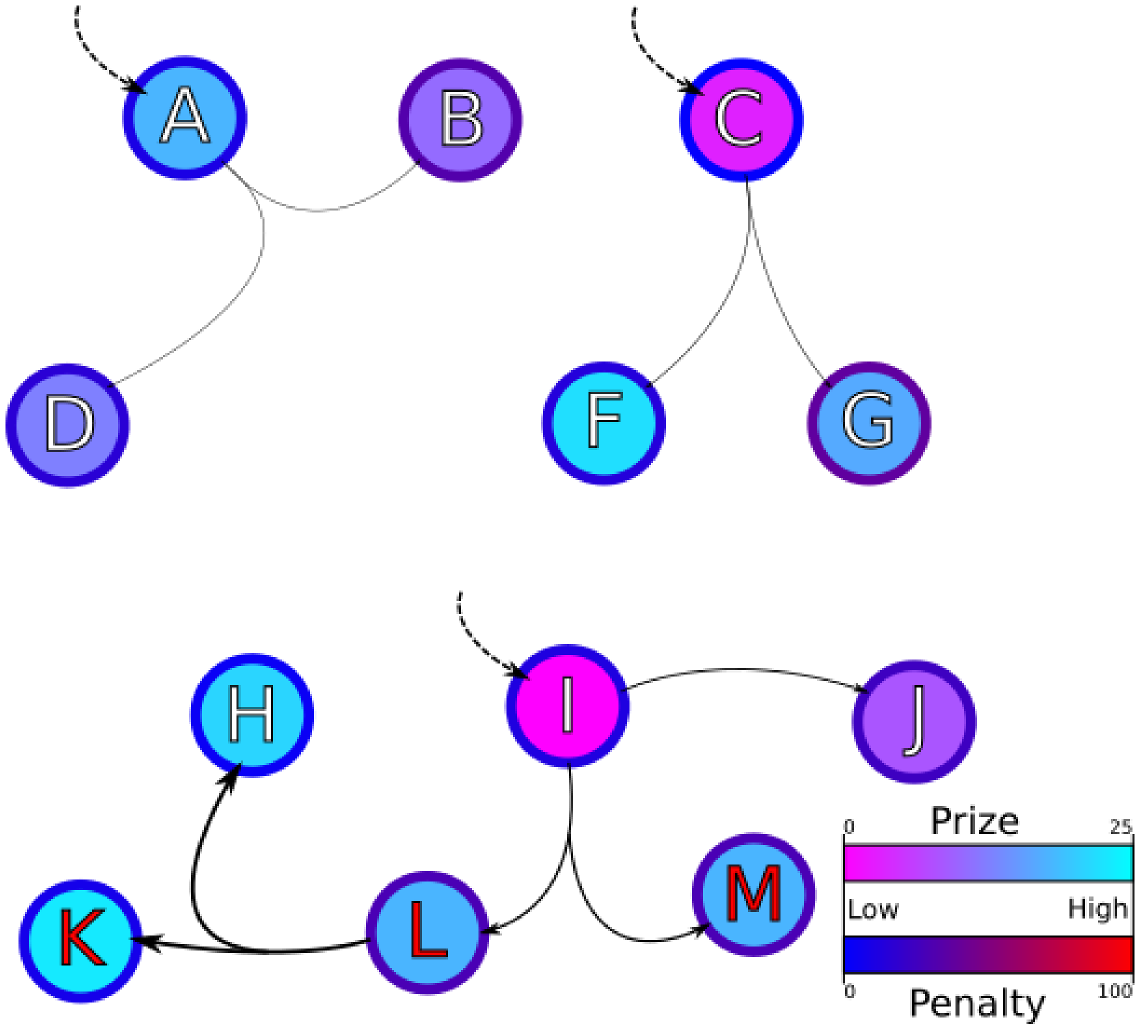


Figure 3.2: The output of our ILP, when it is run on the hypergraph shown in Figure 2.3. The dashed arrows indicate a dangling hypernode.

Chapter 4

Implementation of PCSHT in PPI Data

4.1 Network Construction ***HOW DO I REFERENCE HALP***

In order to construct directed signaling hypergraphs of biological pathways, we use pathway data from preexisting, curated protein-protein interaction networks, available publically. For this, pathway signaling information was downloaded from the curated Reactome pathway database (Croft et al. (2014), Milacic et al. (2012)), and parsed into two flat files containing node and edge information necessary to construct the hypergraph (See Appendix B).

The networks generated from the Reactome database specify each hypernode as a combination of one or more proteins or small molecules. For simplicity, each hypernode was "converted" into a regular node for the purpose of algorithm implementation, and the components' information was saved so that they could be reconstructed *post hoc*. Additionally, each hyperedge contained information about whether the reaction that it represented was regulated either positively or negatively by one or more nodes. Positive regulators were added to the tail of their respective hypernodes, since they are assumed to be necessary reactants for the edge, and should therefore be considered part of the signaling network. Negative regulators, on the other hand, were excluded from the flat files, since the way that they act on reactions poses poses an interesting problem in modeling, since the current definition of PCSHT assumes that all node prizes and edge weights are positive (see Future Directions).

4.2 Network Weighting

For the implementation PCSHT to create meaninful subnetworks, it is vital that the parent hypergraph have a meaningful, biologically informed weighting scheme applied.

4.2.1 Efficiency

Chapter 5

Efficacy (& Proof??)

To assess the efficacy of this ILP in generating the correct Prize Collecting Steiner Hypertree, it is first necessary to perform a series of benchmarking tests on known datasets and hypergraphs. This will allow us to determine if the algorithm is performing as expected, and generating results that can be corroborated manually.

5.1 Simple Example Hypergraphs

5.2 Special Cases

5.3 PCSHT in Real Data

5.3.1 Human Cancer Data

5.3.2 *Arabidopsis thaliana* WGD Data

Chapter 6

Future Directions

- Heat mapping to weight hypergraphs (Random walks).
- Computation of individual or multiple subnetworks.

Conclusion

Here's a conclusion, demonstrating the use of all that manual incrementing and table of contents adding that has to happen if you use the starred form of the chapter command. The deal is, the chapter command in LaTeX does a lot of things: it increments the chapter counter, it resets the section counter to zero, it puts the name of the chapter into the table of contents and the running headers, and probably some other stuff.

So, if you remove all that stuff because you don't like it to say "Chapter 4: Conclusion", then you have to manually add all the things LaTeX would normally do for you. Maybe someday we'll write a new chapter macro that doesn't add "Chapter X" to the beginning of every chapter title.

4.1 More info

And here's some other random info: the first paragraph after a chapter title or section head *shouldn't be* indented, because indents are to tell the reader that you're starting a new paragraph. Since that's obvious after a chapter or section title, proper typesetting doesn't add an indent there.

Appendix A

Algorithms & Programs

Appendix B

Benchmarking Graphs & Supplemental Data

The constructor `build_lp.py` builds a `.lp` file that can be optimized by the ILP solver CPLEX. The constructor takes the following files as arguments, as well as a column delimeter (default "`;`") and a node delimeter (default "`,`").

Edge file (`ex-edges.txt`):

```
tail;head;cost
A;B,D;4
B;F;14
C;F,G;5
D,E;H,K;57
E;H,L;14
F,G;I;33
I;J;9
J;I;31
H;L;8
I;L,M;12
L;H,K;19
E;B;45
```

Node file (`ex-nodes.txt`):

```
name;prize;penalty
A;7;10
B;14;32
C;21;1
D;12;16
E;19;98
F;3;14
G;8;37
H;4;4
I;24;12
J;16;24
K;2;inf
L;7;inf
M;7;inf
```

Appendix C

Biological Data

References

Agrawal, A., Klein, P., & Ravi, R. (1995). When Trees Collide: An Approximation Algorithm for the Generalized Steiner Problem on Networks. *SIAM Journal on Computing*, *24*(3), 440–456.

Aittokallio, T., & Schwikowski, B. (2006). Graph-based methods for analysing networks in cell biology. *Briefings in Bioinformatics*, *7*(3), 243–255.

Aldridge, B. B., Burke, J. M., Lauffenburger, D. A., & Sorger, P. K. (2006). Physicochemical modelling of cell signalling pathways. *Nature cell biology*, *8*(11), 1195–1203. `/Users/yurikoharigaya/Documents/ReadCubeMedia/ncb1497.pdf$\delimiter"026E30F$nhttp://dx.doi.org/10.1038/ncb1497`

Archer, A., Bateni, M. H., Hajiaghayi, M. T., & Karloff, H. (2009). Improved approximation algorithms for Prize-Collecting Steiner Tree and TSP. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, (pp. 427–436).

Ausiello, G., Franciosa, P. G., & Frigioni, D. (2001). Directed Hypergraphs: Problems, Algorithmic Results, and a Novel Decremental Approach. *Theoretical Computer Science, Vol. 2202 of Lecture Notes in Computer Science*, *2202*, 312–328. `http://link.springer.com/10.1007/3-540-45446-2`

Ausiello, G., Giaccio, R., Italiano, G. F., & Nanni, U. (1992). Optimal traversal of directed hypergraphs. *Operations Research*, (20244), 1–24. `http://scholar.google.com/scholar?hl=en{&}btnG=Search{&}q=intitle:Optimal+Traversal+of+Directed+Hypergraphs{#}0$\delimiter"026E30F$nhttp://scholar.google.com/scholar?hl=en{&}btnG=Search{&}q=intitle:optimal+traversal+of+directed+hypergraphs{#}0`

Bateni, M., Chekuri, C., Ene, A., Hajiaghayi, M. T., Korula, N., & Marx, D. (2011). Prize-collecting Steiner problems on planar graphs. *SIAM*, (pp. 1028–1049). `http://dl.acm.org/citation.cfm?id=2133036.2133115`

Bateni, M., & Hajiaghayi, M. (2011). Euclidean Prize-Collecting Steiner Forest. *Algorithmica (New York)*, (pp. 1–24).

Bateni, M., Hajiaghayi, M., & Marx, D. (2010). Prize-collecting Network Design on Planar Graphs. *Computer*, (pp. 1–27). `http://arxiv.org/abs/1006.4339`

Bateni Mohammadhossein; Hajiaghayi, M. M. D. (2011). Approximation Schemes for Steiner Forest on Planar Graphs and Graphs of Bounded Treewidth. *Journal of the ACM*, *58*(5), 1–37.

Battle, A., Jonikas, M. C., Walter, P., Weissman, J. S., & Koller, D. (2010). Automated identification of pathways from quantitative genetic interaction data. *Molecular systems biology*, *6*(379), 1–13. `http://www.ncbi.nlm.nih.gov/pubmed/20531408`

Bauer-Mehren, A., Furlong, L. I., & Sanz, F. (2009). Pathway databases and tools for their exploitation: benefits, current limitations and challenges. *Molecular systems biology*, *5*(290), 1–13. `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2724977{&}tool=pmcentrez{&}rendertype=abstract`

Cambini, R., Gallo, G., & Scutellà, M. G. (1997). Flows on hypergraphs. *Mathematical Programming*, *78*(2), 195–217. `http://link.springer.com/10.1007/BF02614371`

Chekuri, C., Ene, A., & Korula, N. (2010). Prize-Collecting Steiner Tree and Forest in Planar Graphs. *Data Structures and Algorithms*, (pp. 1–24).

Chen, X., Tukachinsky, H., Huang, C. H., Jao, C., Chu, Y. R., Tang, H. Y., Mueller, B., Schulman, S., Rapoport, T. A., & Salic, A. (2011). Processing and turnover of the Hedgehog protein in the endoplasmic reticulum. *Journal of Cell Biology*, *192*(5), 825–838.

Conant, J. (2008). Boolean formulae, hypergraphs and combinatorial topology. `http://arxiv.org/abs/0808.0739`

Croft, D., Mundo, A. F., Haw, R., Milacic, M., Weiser, J., Wu, G., Caudy, M., Garapati, P., Gillespie, M., Kamdar, M. R., Jassal, B., Jupe, S., Matthews, L., May, B., Palatnik, S., Rothfels, K., Shamovsky, V., Song, H., Williams, M., Birney, E., Hermjakob, H., Stein, L., & D'Eustachio, P. (2014). The Reactome pathway knowledgebase. *Nucleic Acids Research*, *42*(Database), 472–477.

Cusick, M. E., Yu, H., Smolyar, A., Venkatesan, K., Carvunis, R., Simonis, N., Rual, J.-f., Borick, H., Braun, P., Dreze, M., Galli, M., Yazaki, J., Hill, D. E., Ecker, J. R., Roth, F. P., & Vidal, M. (2009). Literature-curated protein interaction datasets. *Nature Methods*, *6*(1), 39–46.

Demir, E., Babur, Ö., Rodchenkov, I., Aksoy, B. A., Fukuda, K. I., Gross, B., Sümer, O. S., Bader, G. D., & Sander, C. (2013). Using Biological Pathway Data with Paxtools. *PLoS Computational Biology*, *9*(9), e1003194. `http://dx.plos.org/10.1371/journal.pcbi.1003194`

Demir, E., Cary, M. P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., Wu, G., D'Eustachio, P., Schaefer, C., Luciano, J., Schacherer, F., Martinez-Flores, I., Hu, Z., Jimenez-Jacinto, V., Joshi-Tope, G., Kandasamy, K., Lopez-Fuentes, A. C., Mi, H., Pichler, E., Rodchenkov, I., Splendiani, A., Tkachev, S., Zucker, J., Gopinath, G., Rajasimha, H., Ramakrishnan, R., Shah, I., Syed, M., Anwar, N., Babur, Ö., Blinov, M., Brauner, E., Corwin, D., Donaldson, S., Gibbons, F., Goldberg, R., Hornbeck, P., Luna, A., Murray-Rust, P., Neumann, E., Reubenacker, O., Samwald, M., van Iersel, M., Wimalaratne, S., Allen, K., Braun, B., Whirl-Carrillo, M., Cheung, K.-H., Dahlquist, K., Finney, A., Gillespie, M., Glass, E., Gong, L., Haw, R., Honig, M., Hubaut, O., Kane, D., Krupa, S., Kutmon, M., Leonard, J., Marks, D., Merberg, D., Petri, V., Pico, A., Ravenscroft, D., Ren, L., Shah, N., Sunshine, M., Tang, R., Whaley, R., Letovksy, S., Buetow, K. H., Rzhetsky, A., Schachter, V., Sobral, B. S., Dogrusoz, U., McWeeney, S., Aladjem, M., Birney, E., Collado-Vides, J., Goto, S., Hucka, M., Novère, N. L., Maltsev, N., Pandey, A., Thomas, P., Wingender, E., Karp, P. D., Sander, C., & Bader, G. D. (2010). The BioPAX community standard for pathway data sharing. *Nature Biotechnology*, *28*(9), 935–942. `http://www.nature.com/doifinder/10.1038/nbt.1666`

Estrada, E., & Rodríguez-Velázquez, J. A. (2006). Subgraph centrality and clustering in complex hyper-networks. *Physica A: Statistical Mechanics and its Applications*, *364*, 581–594.

Fukuda, K., & Takagi, T. (2001). Knowledge representation of signal transduction pathways. *Bioinformatics (Oxford, England)*, *17*(9), 829–837. `http://www.ncbi.nlm.nih.gov/pubmed/11590099`

Gallagher, S. R., & Goldberg, D. S. (2013). Clustering Coefficients in Protein Interaction Hypernetworks Categories and Subject Descriptors. *BCB-ACM*, (pp. 552–560).

Gallet, A. (2011). Hedgehog morphogen: From secretion to reception. *Trends in Cell Biology*, *21*(4), 238–246. http://dx.doi.org/10.1016/j.tcb.2010.12.005

Gallo, G., Longo, G., Pallottino, S., & Nguyen, S. (1993). Directed hypergraphs and applications. *Discrete Applied Mathematics*, *42*, 177–201. http://www.sciencedirect.com/science/article/pii/0166218X9390045P

Garey, M. R., & Johnson, D. S. (2016). The Rectilinear Steiner Tree Problem is NP-Complete. *SIAM Journal on Applied Mathematics*, *32*(4), 826–834.

Hajiaghayi, M., Khandekar, R., Kortsarz, G., & Nutov, Z. (2010). Prize-collecting Steiner network problems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *LNCS 6080*, 71–84.

Haus, U.-U., Niermann, K., Truemper, K., & Weismantel, R. (2009). Logic integer programming models for signaling networks. *Journal of computational biology: a journal of computational molecular cell biology*, *16*(5), 725–743.

Hu, Z., Mellor, J., Wu, J., Kanehisa, M., Stuart, J. M., & DeLisi, C. (2007). Towards zoomable multidimensional maps of the cell. *Nature Biotechnology*, *25*(5), 547–554.

Hui, C.-c., & Angers, S. (2011). Gli Proteins in Development and Disease. *Annual Review of Cell and Developmental Biology*, *27*(1), 513–537.

Johnson, D. S., Minkoff, M., & Phillips, S. (2000). The prize collecting Steiner tree problem: theory and practice. *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, (pp. 760–769). http://dl.acm.org/citation.cfm?id=338219.338637

Kandasamy, K., Mohan, S. S., Raju, R., Keerthikumar, S., Kumar, G. S. S., Venugopal, A. K., Telikicherla, D., Navarro, J. D., Mathivanan, S., Pecquet, C., Gollapudi, S. K., Tattikota, S. G., Mohan, S., Padhukasahasram, H., Subbannayya, Y., Goel, R., Jacob, H. K., Zhong, J., Sekhar, R., Nanjappa, V., Balakrishnan, L., Subbaiah, R., Ramachandra, Y., Rahiman, B. A., Prasad, T. K., Lin, J.-X., Houtman, J. C., Desiderio, S., Renauld, J.-C., Constantinescu, S. N., Ohara, O., Hirano, T., Kubo, M., Singh, S., Khatri, P., Draghici, S., Bader, G. D., Sander, C., Leonard, W. J., & Pandey, A. (2010). NetPath: a public resource of curated signal transduction pathways. *Genome Biology*, *11*(1), R3. http://www.ncbi.

nlm.nih.gov/pmc/articles/PMC2847715/$\delimiter"026E30F$nhttp://www.ncbi.nlm.nih.gov/pmc/articles/PMC2847715/pdf/gb-2010-11-1-r3.pdf

Karlebach, G., & Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature reviews. Molecular cell biology*, *9*(10), 770–780.

Khatri, P., Sirota, M., & Butte, A. J. (2012). Ten years of pathway analysis: Current approaches and outstanding challenges. *PLoS Computational Biology*, *8*(2), e1002375.

Kirouac, D. C., Saez-Rodriguez, J., Swantek, J., Burke, J. M., Lauffenburger, D. A., & Sorger, P. K. (2012). Creating and analyzing pathway and protein interaction compendia for modelling signal transduction networks. *BMC Systems Biology*, *6*(29), 1–18. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3436686{&}tool=pmcentrez{&}rendertype=abstract

Klamt, S., Haus, U.-U., & Theis, F. (2009). Hypergraphs and Cellular Networks. *PLoS Computational Biology*, *5*(5), e1000385. http://dx.plos.org/10.1371/journal.pcbi.1000385

Klamt, S., Saez-Rodriguez, J., Lindquist, J. A., Simeoni, L., & Gilles, E. D. (2006). A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC bioinformatics*, *7*(56), 1–26. http://www.biomedcentral.com/1471-2105/7/56

Knuth, D. E., Larrabee, T., & Roberts, P. M. (1987). Mathematical Writing.

Matthews, L., Gopinath, G., Gillespie, M., Caudy, M., Croft, D., de Bono, B., Garapati, P., Hemish, J., Hermjakob, H., Jassal, B., Kanapin, A., Lewis, S., Mahajan, S., May, B., Schmidt, E., Vastrik, I., Wu, G., Birney, E., Stein, L., & D'eustachio, P. (2009). Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Research*, *37*(Database), 619–622.

Meysman, P., Saeys, Y., Sabaghian, E., Bittremieux, W., Ban de Peer, Y., Goethals, B., & Laukens, K. (2015). Discovery of Significantly Enriched Subgraphs Associated with Selected Vertices in a Single Discovery of Significantly Enriched Subgraphs Associated with Selected Vertices in a Single Graph. *ACM*, (Augugust).

Milacic, M., Haw, R., Rothfels, K., Wu, G., Croft, D., Hermjakob, H., D'Eustachio, P., & Stein, L. (2012). Annotating cancer variants and anti-cancer therapeutics in Reactome. *Cancers*, *4*(4), 1180–1211.

Navlakha, S., Gitter, A., & Bar-Joseph, Z. (2012). A Network-based Approach for Predicting Missing Pathway Interactions. *PLoS Computational Biology*, *8*(8).

Newman, M. E. J. (2003). The structure and function of complex networks. *Society for Industrial and Applied Mathemathics Review*, *45*(2), 167–256.

Nguyen, S., Pretolani, D., & Markenzon, L. (1998). On some path problems on oriented hypergraphs. *Informatique théorique et applications(Imprimé)*, *32*(1-3), 1–20. `http://cat.inist.fr/?aModele=afficheN{&}amp;cpsidt=2388950`

Pan, A., Chang, L., Nguyen, A., & James, A. W. (2013). A review of hedgehog signaling in cranial bone development. *Frontiers in Physiology*, *4*(61), 1–14.

Paz, A., Brownstein, Z., Ber, Y., Bialik, S., David, E., Sagir, D., Ulitsky, I., Elkon, R., Kimchi, A., Avraham, K. B., Shiloh, Y., & Shamir, R. (2011). SPIKE: A database of highly curated human signaling pathways. *Nucleic Acids Research*, *39*(Database), 793–799.

Rahman, A., Poirel, C. L., Badger, D. J., Estep, C., & Murali, T. (2013). Reverse Engineering Molecular Hypergraphs. *IEEE/ACM Trans Comput Biol Bioinform*, *10*(5), 1113–1124.

Ramadan, E., Perincheri, S., & Tuck, D. (2010). A hyper-graph approach for analyzing transcriptional networks in breast cancer. *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, (pp. 556–562). `http://portal.acm.org/citation.cfm?doid=1854776.1854882`

Ritz, A., Avent, B., & Murali, T. M. (2014a). Pathway Analysis with Signaling Hypergraphs. *IEEE Transactions on Computational Biology and Bioinformatics*, (pp. 249–258).

Ritz, A., & Murali, T. M. (2014). Pathway Analysis with Signaling Hypergraphs. *BCB-ACM*, (pp. 249–258).

Ritz, A., Tegge, A. N., Kim, H., Poirel, C. L., & Murali, T. (2014b). Signaling hypergraphs. *Trends in Biotechnology*, *32*(7), 356–362. `http://linkinghub.elsevier.com/retrieve/pii/S0167779914000717`

Samaga, R., & Klamt, S. (2013). Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks. *Cell Communication and Signaling*, *11*(43), 1–19.

Schaefer, C. F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., & Buetow, K. H. (2009). PID: the Pathway Interaction Database. *Nucleic Acids Research*, *37*(Database), D674–D679. `http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gkn653`

Sharan, R., Ulitsky, I., & Shamir, R. (2007). Network-based prediction of protein function. *Molecular Systems Biology*, *3*(88), 1–13. `<GotoISI>://WOS:000245381300003`

Thakur, M., & Tripathi, R. (2009). Linear connectivity problems in directed hypergraphs. *Theoretical Computer Science*, *410*, 2592–2618. `http://linkinghub.elsevier.com/retrieve/pii/S0304397509002011`

Tuncbag, N., Braunstein, A., Pagnani, A., Huang, S.-S. C., Chayes, J., Borgs, C., Zecchina, R., & Fraenkel, E. (2013). Simultaneous Reconstruction of Multiple Signaling Pathways via the Prize-Collecting Steiner Forest Problem. *Journal of Computational Biology*, *20*(2), 124–136. `http://online.liebertpub.com/doi/abs/10.1089/cmb.2012.0092`

Vyas, N., Goswami, D., Manonmani, A., Sharma, P., Ranganath, H. A., VijayRaghavan, K., Shashidhara, L. S., Sowdhamini, R., & Mayor, S. (2008). Nanoscale Organization of Hedgehog Is Essential for Long-Range Signaling. *Cell*, *133*(7), 1214–1227.