**Improving the interpretability of integer linear programming methods for biological subnetwork inference**

by

Deborah A. Chasman

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

University of Wisconsin–Madison

2014

Date of final oral examination: 8/14/2014

The dissertation is approved by the following members of the Final Oral Committee:
Mark Craven, Professor, Biostatistics and Medical Informatics
Paul Ahlquist, Professor, Oncology and Molecular Virology
Michael Ferris, Professor, Computer Sciences
Audrey Gasch, Associate Professor, Genetics
C. David Page, Professor, Biostatistics and Medical Informatics
Xiaojin Zhu, Associate Professor, Computer Sciences

*In memory of my grandparents:*
*Gertrude and Bernard Chasman,*
*Irene and John Kowalak*

# Acknowledgments

Like the host-dependent viruses whose data I consider in this dissertation, a graduate student benefits from the support of many others to complete her life cycle. I am happy to express my sincere gratitude to the people who have influenced and guided my graduate education.

First, I thank my advisor, Mark Craven. Mark's commitment to science, integrity, and to his students is inspiring. Working with him, I have learned not only many technical skills, but also patience, an appreciation for detail, and how to respond to challenges. Having him as an advisor and mentor has been a great privilege.

I also thank the other members of my thesis committee. Michael Ferris guided my study of optimization, and provided computational resources. David Page and Jerry Zhu posed insightful questions, pushing me to look at my research problems from new vantage points. Collaborating with Paul Ahlquist and Audrey Gasch has been a great highlight of graduate school for me, and I am grateful for their knowledge and for the opportunities that they provided me.

I have enjoyed working with and learning from my collaborators in virology and genetics. In addition to Paul and Audrey, I worked closely with Brandi Gancarz, Linhui Hao, Eunju Park, David Berry, and Elisha Ho.

The machine learning and biomedical informatics communities at Wisconsin have contributed to my development as a researcher. Early on, courses from Mark, David, Colin Dewey and Jude Shavlik were very influential. Weekly meetings with the Craven research group guided my learning and helped me to improve my presentation skills. Its members during my time have been Dave Andrzejewski, Andy Chang, Alex Cobian, Emily Kawaler, Kyubin Lee, Deborah Muganda-Rippchen, Irene Ong, Gary Pack, Yue Pan, Anna Rissman, Burr Settles, Adam Smith, Andreas Vlachos, and Matt Ziegler. My fellow graduate students and postdocs in the AI Reading Group, MSC offices, WACM, and the greater CS department have also created a supportive and fun learning community here. Special thanks to Irene, who has been a great mentor to me. Also, to Deborah and to Kendrick Boyd for their camaraderie in the KDD office, undaunted by wildlife and extreme temperatures.

I wish to thank the departments of Computer Sciences and Biostatistics and Medical Informatics at the University of Wisconsin–Madison. I received training and financial support from the Computation and Informatics in Biology and Medicine (CIBM) program (NIH/NLM 5T15-LM007359), and funding from NIH/NHGRI training grant 5T32-HG002760

and NSF grant IIS-1218880. Some of the many people who facilitated my participation in these programs were Angela Thorp, Cathy Whitford, Louise Pape and Karen Nafzger.

My friends and family, in Madison and afar, have been a great source of encouragement. I'm grateful to have shared a roof with several compassionate and curious people, and great conversations with many more. Brian Aydemir has been an invaluable source of tea and mindfulness. My parents, Jon and Janice, and my sister, Merc, have always supported and celebrated my education and happiness. Thank you.

# Contents

# List of Tables

# List of Figures

# Abstract

Biologists often screen an entire genome for involvement in a cellular phenotype of interest. To analyze the resulting data, computational methods have been developed to infer subnetworks that posit how the identified genes interact to produce the phenotype. While these methods have been successfully used to guide new discoveries, there is still a large representational gap between the output they produce and actual biology. This dissertation advances the state of the art of subnetwork inference methods by improving the interpretability of the inferred subnetworks.

Our main contribution is a general integer linear programming method for inferring subnetworks that predict the mechanism by which relevant genes modulate a cellular phenotype. We promote the interpretability of the inferred subnetworks at each stage of the method. In the input data, we integrate heterogeneous data types and give new representations for non-binary biological concepts. During inference, we use biologically motivated constraints and objective functions. Finally, we provide tools to assist in the interpretation of the inferred subnetworks and the generation of testable hypotheses.

We developed our method through application to three biological domains. First, we apply the method to infer directed, host-virus interaction subnetworks in a yeast host, including which host factors are most likely to directly interact with the virus.

Our second extension shows that the method can be scaled to infer relevant subnetworks from larger, sparser input data. Our application is to infer HIV-relevant human gene subnetworks. As part of this study, we present a method for extracting a tailored view of the inferred subnetwork based on a user-supplied set of query genes.

In our third extension, we infer the signaling subnetwork that regulates the yeast transcriptional response to stress. Here, we also present a method for predicting which genes are most directly involved in coordinating complementary cellular responses.

For each extension, we demonstrate the accuracy of our method's predictions through several computational and literature-based evaluations. In addition, our collaborators have experimentally validated several of the predictions made by the inferred yeast salt-responsive subnetwork, and have used the subnetwork to inspire further experimental inquiry.

# 1 Introduction

The philosophy behind systems biology is that a broader understanding of cellular processes can be gained through holistic measurements of cellular activity, with minimal bias of the choice of which measurements to take. The challenge then arises of how to interpret the vast amounts of data that are gathered from these high-throughput experiments, and, furthermore, how to generate hypotheses about physical mechanisms in the cell that can be tested by further experiments. Several recent works have proposed that physical mechanisms can be predicted by integrating high-throughput data with other data representing direct physical and causal interactions between cellular gene products. In these works, the data are represented as a graph in which the nodes represent genes and gene products, and the edges represent interactions from various experimental sources. Overlaid on this graph is the new information from the high-throughput experiment. This thesis proposes a new approach for inferring interpretable subnetworks from this kind of input graph. The inferred subnetworks hypothesize which physical cellular interactions are truly relevant to the condition being studied, and predict missing information such as the directions of edges and the identity of additional relevant nodes. Our approach is based on integer linear programming, and uses biologically motivated constraints and objective functions to infer subnetworks that provide consistent interpretations of the experimental data. We apply our method to three biological problems: inferring host-virus interaction mechanisms in a yeast host and a mammalian host, and inferring the yeast osmotic stress-responsive signaling subnetwork. We also offer contributions toward improving the interpretability of the inferred subnetworks.

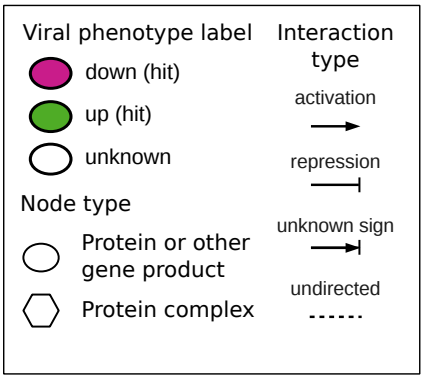
We introduce our general approach with an example task, which we will return to in Chapter 3: the study of host-virus interactions. Viruses require the use of host machinery for nearly every step of their replication cycle. To identify which host factors (genes and gene products) are relevant to the viral life cycle, several recent studies have performed host-genome-wide loss-of-function experiments to measure changes in viral replication product when individual host genes are suppressed. These studies have used either yeast **mutant** libraries (Kushner *et al.*, 2003; Serviene *et al.*, 2005, 2006; Gancarz *et al.*, 2011; Hao *et al.*, 2014) or **RNA interference** (Cherry *et al.*, 2005; Brass *et al.*, 2008; Hao *et al.*, 2008; König *et al.*, 2008) to systematically suppress the production of host gene products. Typically, these genome-wide screens identify a large number of host genes, which we refer to as **hits**, whose loss has a significant effect on the virus. Because they are identified by a gene suppression screen, they may also be referred to as *genetic hits*. An example hit set is provided in Figure 1.1A. We make the assumption that the gene suppression experiment

**A** Hit list

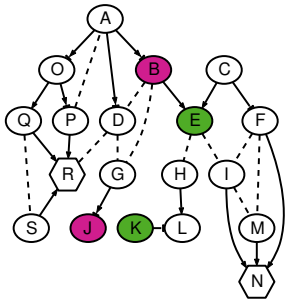
| Suppressed host gene | Viral phenotype label |
|---|---|
| B | down |
| E | up |
| J | down |
| K | up |

**B** Enriched Gene Ontology terms

| Term | Hits | Total genes with term |
|---|---|---|
| core mediator complex | B | 20 |
| lipid biosynthetic process | E, K | 166 |
| response to nutrient | J | 2 |




**C** Background network + viral phenotype labels




**D** Inferred subnetwork

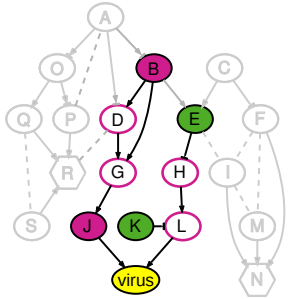



**Figure 1.1:** Overview of host-virus subnetwork inference task.
**A** Hit list of relevant host factors, derived from a genome-wide suppression experiment measuring the level of viral replication in a host cell.
**B** Gene Ontology enrichment analysis results for the hit list.
**C** Integration of the hit list with a host background network.
**D** Inferred subnetwork, including predicted host-virus interfaces, additional genes that are predicted to be relevant, and edge signs and directions. The greyed-out nodes and edges are inferred to be irrelevant.

interrupted a cellular **pathway** that normally results in a product or activity that is involved in either enhancing or inhibiting viral replication. Therefore, we say that the hits are acting **upstream** of some direct interaction between the host cell and the virus. The **sign** of the effect on viral replication is indicated with a **phenotype label**: **down**, indicating that viral replication is inhibited by the suppression of the gene and therefore that the gene is required for normal viral replication, or **up**, indicating the opposite.

While these screens identify host factors that are relevant to viral replication, they do not reveal how the hits are organized into the pathways that modulate the virus, nor do they distinguish between which host genes directly interface with a viral component and which indirectly affect the virus. Furthermore, it has been observed that each screen only identifies a fraction of the truly relevant host genes (Hao *et al.*, 2013). The screening technologies themselves have limitations. Regarding **yeast deletion libraries**, many genes

are essential to cellular survival and cannot be removed. In other cases, the yeast may adapt to a gene deletion by expressing another redundant gene or pathway. Gene suppression technologies such as **dox-repressible mutant**s or RNA interference (RNAi) may also result in incomplete suppression or, in the case of RNAi, off-target effects.

A popular first step toward deducing which cellular mechanisms are represented by an experimentally derived hit list is to test the hit set for enrichment with curated categories of genes. The **Gene Ontology** (Ashburner *et al.*, 2000) is a tool that is often used for this purpose; it represents an effort to annotate all known genes with information about their functions using a hierarchical vocabulary. The output of an enrichment analysis is a list of the ontology terms that are statistically over-represented in the input experimental hit list. An example is shown in Figure 1.1B. While Gene Ontology enrichment analysis is undoubtedly useful, what it provides is an interpretation of the hit list at the level of categories. It does not predict detailed mechanisms, or predict which host factors directly interface with the virus, and which others are relevant further upstream.

As an alternative to providing interpretation through pre-selected gene lists, recently, several computational methods (reviewed in Chapter 2) have been developed to integrate the hit lists (or other high-throughput measurements) with direct protein-protein and gene-regulatory interactions that are available from public databases. The subnetwork inference method presented in this thesis provides advancements in this area. In this approach, the interactions are represented as a graph, in which the nodes are gene products and the edges are interactions that have been observed using various experimental techniques and under various conditions. We call this graph the **background network**, and, in our method, it represents the space of possible interactions that may be relevant to viral replication in the host. Figure 1.1C depicts a background network. Notice that some interactions have directions, represented by arrows or tees. In such cases, a causal relationship was observed by an experiment; for example, a phosphorylation event between a kinase and its substrate. Some interactions also have signs. For positively-signed interactions (arrowheads), the data supports a positive relationship between the activity, expression, or function of the two gene products, such as transcriptional activation. Negatively-signed interactions (tees) represent an inverse relationship between the two genes products, such as transcriptional repression. Most edges represent protein binding and are undirected and unsigned.

The task is to predict the mechanism by which the suppression of each hit gene results in the repression or enhancement of viral replication. Simply overlaying the hit genes on the background network is insufficient for this purpose. Even in this small example network, doing so has perhaps only increased the complexity of the problem. The difficulty increases when we consider actual biological networks, which have thousands, or tens

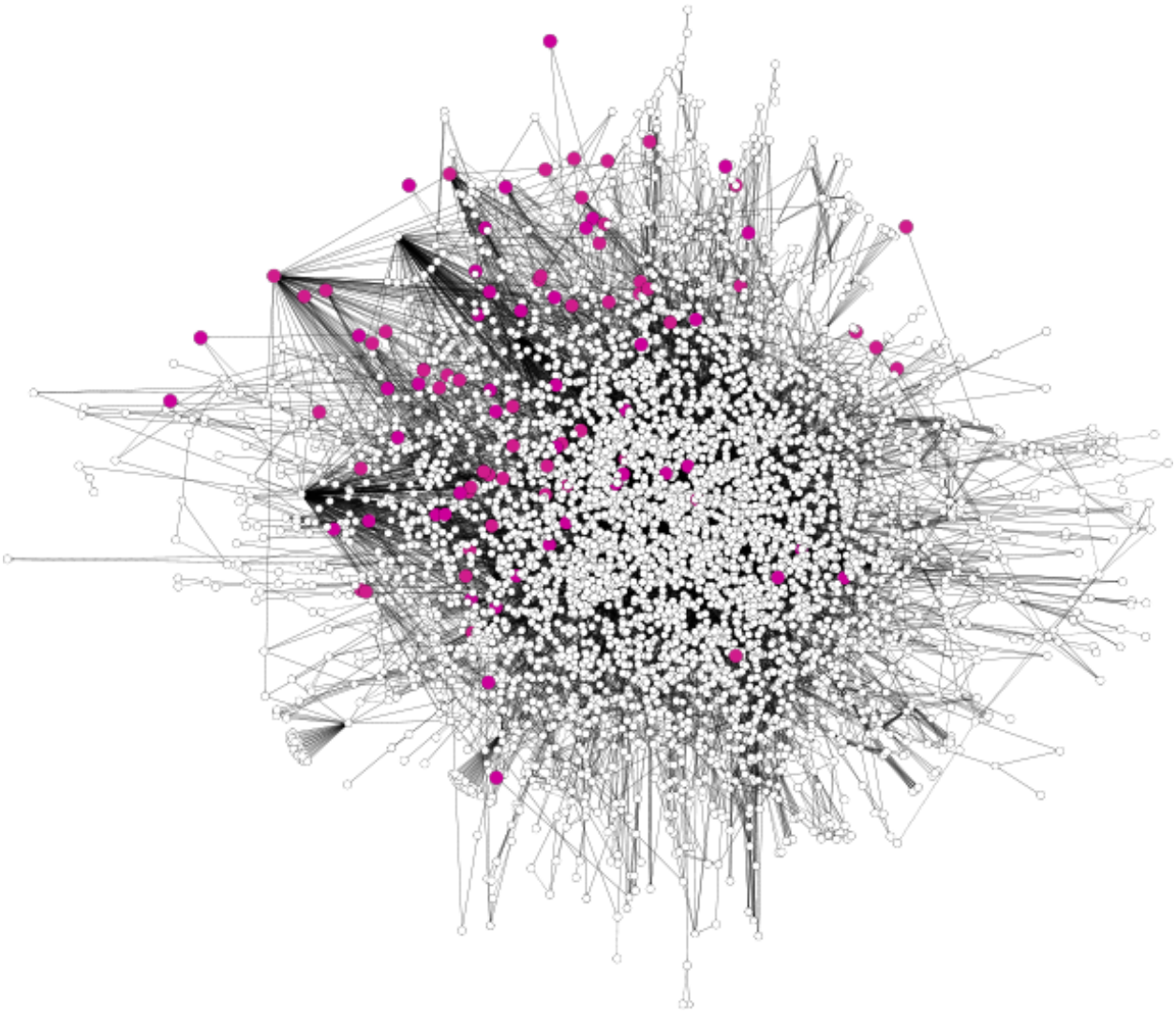


**Figure 1.2:** Example biological network, overlaid with a hit set. Nodes represent genes, and edges represent biological interactions. Dark pink genes are hits identified from a gene suppression assay for host genes involved in viral replication.

of thousands, of nodes. An actual network, with shaded nodes indicating hit genes, is shown in Figure 1.2. It is not at all clear which host factors have a direct interaction with a viral component, nor what the causal relationships are between the hits. A computational method is needed to make sense of this "hairball".

The output of our proposed method is an **inferred subnetwork** providing directed **paths** leading from the hits to the host factors that are predicted to be most proximal to a direct interaction with a viral component. We call these latter host factors **interfaces**. The interfaces are not part of the input data, and must be predicted. Since most of the hits are not directly connected to each other, it is necessary to predict which non-hit nodes are

truly relevant, and to infer the relevant paths through the background network that link hits to interfaces either directly or via predicted hits. Any node or edge not in a relevant path is considered irrelevant. In order to provide a **consistent** interpretation of how the suppression of each hit propagates through the network to affect one or more interfaces, we must infer directions and signs for any predicted relevant edges that lack this information. Edges with inconsistent signs and directions are inferred to be irrelevant to viral replication.

Figure 1.1D shows an example of an inferred subnetwork. Genes J and L are predicted to be host interfaces to the virus, as indicated by the directed edges to the new virus node. Some of the edges and nodes, shown in gray, are predicted to be irrelevant. The predicted relevant edges are assigned directions and signs in cases where these properties are not specified by the background network. The signs and directions for the relevant edges are set such that there is at least one consistent path linking each hit to the virus. An example path is the following: gene E has an **up** phenotype because, under normal expression levels, it represses the function of H, which is to activate the interface L. The subnetwork predicts that several genes, D, G, H, and L, whose phenotypes were unobserved or inconclusive based on the initial screen, are actually key host factors involved in viral replication. Their red outline indicates that they are predicted to have a **down** phenotype.

The approach we present in this thesis is not specific to viral replication, and can be generalized to address other biological questions and other input data types in addition to hit sets. In the following chapters, we discuss three extensions of our general method: two to different settings for inferring host-virus interaction subnetworks, and one for inferring the signaling subnetwork that controls the yeast response to a stressful environment. An overarching goal of this work is to improve the interpretability of subnetworks produced by computational inference methods. We address this by improving the subnetworks' representation of actual biological relationships, attempting to reduce the amount of work required to translate them into meaningful, testable hypotheses.

## 1.1 Thesis statement and contributions

**By integrating multiple sources of biological data, and by carefully specifying biologically motivated constraints and objective functions, we can infer subnetworks that accurately predict the underlying physical mechanisms and can be used to assist biologists in interpreting their data and designing further experiments.**

The major contributions of this thesis are:

**1. A method for inferring interpretable subnetworks from high-throughput biological data** (all chapters). This thesis proposes a general computational method for inferring the relevant subnetworks through which intentionally suppressed genes modulate a cellular phenotype (including, but not limited to, viral replication). These subnetworks can be used to predict which unassayed genes may be involved in the phenotype of interest, interpret the role of each hit in modulating the cellular response, and guide further experimentation that is aimed at uncovering and validating the mechanisms of the response. To account for the incompleteness of the input data, our method infers an ensemble of subnetworks, which, taken together, are used to quantify the method's uncertainty about each prediction. Our approach is based on integer linear programming and flexibly allows for the use of biologically motivated constraints and objective functions.

**2. Applications of the approach to multiple organisms and phenotypes.** We apply our method to three biological problems: inferring host-virus interactions in a yeast host (Chapter 3), inferring host-virus interactions in a mammalian host (Chapter 4), and inferring the signaling subnetwork that orchestrates the yeast response to salt stress (Chapter 5). Because the true structures of these subnetworks are unknown, we evaluated our method by subjecting each inferred subnetwork ensemble to rigorous computational and literature-based evaluations. We also demonstrate the utility of the subnetworks in assisting biologists in designing experiments to uncover novel biological results.

**3. Improvements in the interpretability of the inferred subnetworks.**

- **In the representation of diverse, relevant input data** (all chapters). We incorporate heterogeneous biological data in the background network as well as in the experimental data to be explained (all chapters). The background network includes both binary interactions between proteins and nucleic acids, as well as non-binary relationships like protein complexes and metabolic pathways. With respect to the input experimental data, we investigate hit genes identified by gene suppression assays

(as described earlier in this chapter), proteins identified from phosphoproteomic analysis, host-virus protein interactions, and genetic interactions.

- **In the design of the inference process** (all chapters). In designing the integer programming models, we employ expert knowledge in our desiderata for biologically plausible subnetworks. We design a representation that can make useful, testable predictions, specify how each data type can be incorporated into the inferred subnetwork, and choose which quantities to optimize with the objective functions.

- **In methods for querying the inferred subnetworks.** Even though inferred subnetworks provide structure and context to input experimental data, they may still be difficult for humans to interpret due to their size and scope. We propose two methods for assisting interpretation in the use case of a biologist wishing to query the subnetworks for predictions about a specific biological concept. In Chapter 4, we present a method for using independently derived gene lists to provide context-specific views of the HIV-human inferred subnetwork. In Chapter 5, we present a method for assigning predicted relevant genes to different aspects of the stress response, and, furthermore, predict which regulators in the salt signaling subnetwork are involved in reassigning cellular resources from normal usage to stress-response behavior.

## 1.2 Outline

The thesis is organized as follows:

- In **Chapter 2**, we present a review of related computational methods and place our work in context.

- **Chapter 3** presents our work on inferring host-virus subnetworks in a yeast host, using hit lists identified for two RNA viruses. We evaluate our method via comparison to related approaches, through the use of permutation tests, and by identifying support in the literature for predicted components of the subnetworks. In this chapter, we also propose and evaluate a representation for protein complexes in the background network.

- In **Chapter 4**, we extend our method to infer human-HIV interaction subnetworks. Our human-HIV-specific application includes another proposed representation of protein complexes and metabolic pathways. We also present methods for generating tailored views of the inferred subnetworks, in order to allow the viewer to identify which parts of a subnetwork are most relevant to a cellular process of interest.

- **Chapter 5** presents an extension of our general method to the task of inferring the yeast salt stress signaling subnetwork. In addition to being supported by a multi-faceted computational evaluation, our inferred subnetwork is supported by additional experimental validation. It is also used to guide experiments that have uncovered new insights into stress biology. As part of this analysis, we present a method for associating predicted relevant genes with different aspects of the stress response.
- **Chapter 6** summarizes the contributions of the thesis and proposes future work.
- **Glossary** provides a quick reference to the concepts and terms used in the thesis. The first use of a glossary term is printed in **sans-serif and bold font**.

Each of Chapters 3 to 5 concludes with a Summary section explaining how the chapter supports the main contributions of the thesis.

# 2 Related work

This thesis work extends and relates to several areas of research in computational systems biology, including:

1. Regulatory network reconstruction
2. Subnetwork inference from data from perturbation experiments
3. Subnetwork extraction from one or more hit sets
4. Candidate gene prioritization
5. Protein complex extraction
6. Biological network filtering and integration
7. Gene set enrichment analysis

This chapter gives a tour through the computational literature that is most relevant to this thesis. At the end of each section, we give a direct comparison to our work.

The six panels in Figures 2.1 and 2.2 illustrate one example from each of items 1-6 at a high level. The figure is broken into two pieces for display purposes. Gene set enrichment analysis is illustrated in Figure 1.1A-B.

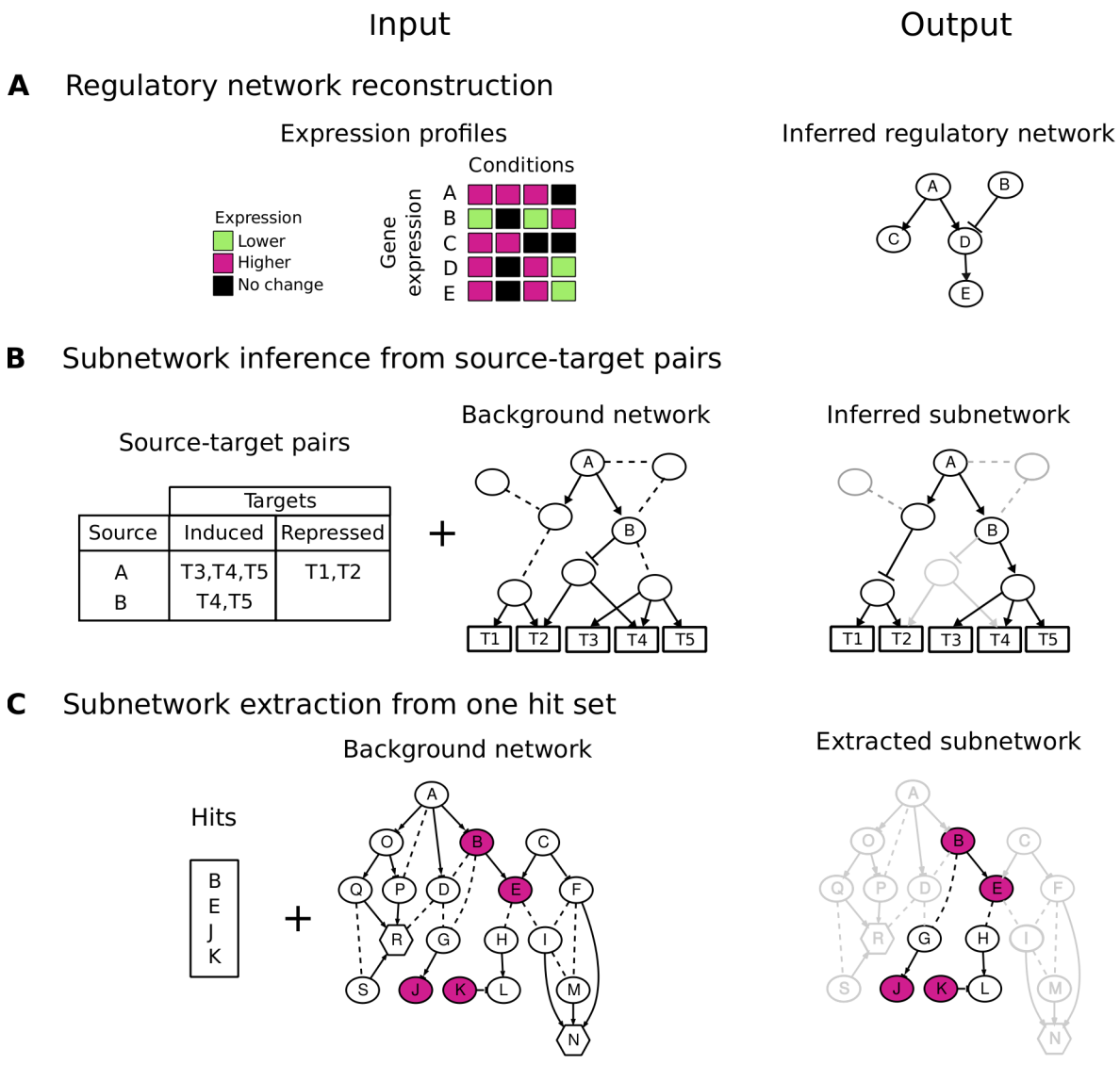



**Figure 2.1:** An illustrated guide to related work, Part 1.

**Caption for Figure 2.1**.
**A** (Section 2.1) The input is a set of gene expression profiles taken at multiple time points or under multiple conditions. In the left side of the panel, the input is shown with expression levels discretized into three values relative to wild type expression. The output is a graph depicting the dependency structure between the genes' expression levels.
**B** (Section 2.2) The input is a set of source-target pairs identified from perturbation experiments and a background network. In the table, 'Induced' targets show higher expression in the source mutant than wild type; the opposite is observed for 'Repressed' targets. The output is an inferred relevant subnetwork that connects the sources to their targets via directed paths. Signs and directions are inferred on the edges. Grey nodes and edges are those deemed irrelevant by the inference method.
**C** (Section 2.3) The input is a set of experimentally derived hits for an experimental condition and a background network. In the background network, hits are shaded in dark pink. The output is an extracted subnetwork that connects the hits and predicts a minimal number of additional relevant hits. Grey nodes and edges are those deemed irrelevant by the extraction method.

**Caption for Figure 2.2**.
**D** (Section 2.4) The input is a set of known relevant hits and a background network (with hits shown in very dark pink). In the output network, the hits' labels are diffused throughout the network, with confidence in hit status decreasing with distance. These confidence values are represented with node shading.The predicted hits are ranked in increasing order of confidence (from dark to light).
**E** (Section 2.5) The input is a background network of protein-protein interactions. The output is a set of predicted protein complexes, which are cliques or near-cliques in the background network.
**F** (Section 2.6) The input is a set of genes that are observed to be expressed in a cell-type of interest and a background network. The genes that are expressed in the cell-type are shown in dark pink. The output is a filtered background network that contains only edges between cell-type-relevant nodes.

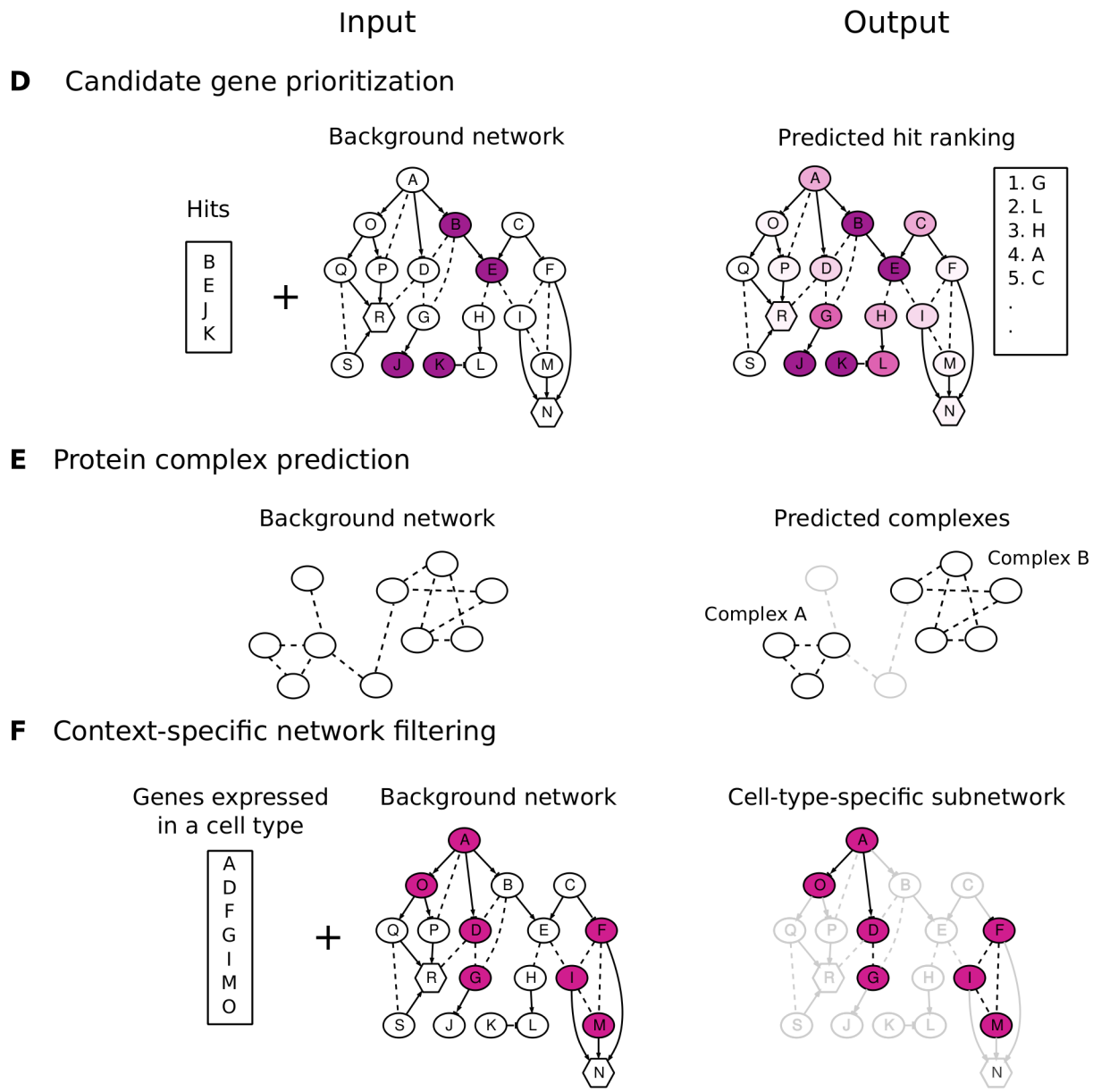



**Figure 2.2:** An illustrated guide to related work, Part 2.

## 2.1 Regulatory network reconstruction from expression profiles (Figure 2.1A)

Messenger RNA (mRNA) expression profiles – quantitative measurements of the abundance of individual gene transcripts, across multiple conditions – are often viewed as a proxy measurement (if imperfect) of functional gene activity. If a group of genes shares a common pattern of expression across multiple conditions, it is interpreted as an indication that the genes share a similar function. Many computational approaches address the task of not only finding these functional modules (*e.g.*, by clustering; Eisen *et al.* (1998)), but also reconstructing the regulatory relationships between genes or modules. The input to this task is a set of expression profiles covering many experimental conditions for many genes. The expression values are often discretized and represent relative expression compared to an experimental control. The output is a directed graph that predicts the regulatory relationships between the genes. An example of the input and output to this task is provided in Figure 2.1A.

Many different representations have been used to approach the network reconstruction problem. Some early approaches discretize the expression levels and learn Boolean networks (Liang *et al.*, 1998; Akutsu *et al.*, 2000), in which each node represents a gene, and each gene's expression value is a Boolean function of its parents'. Friedman *et al.* (2000) cast the problem as Bayesian network structure learning. In the most basic representation, the nodes represent genes and directed edges represent a union of the regulatory relationships present across the conditions. Each gene receives a conditional probability model for each set of parent gene states, which estimates the likelihood of the gene being expressed at any level given the expression levels of its parents. Other graphical modeling approaches extend this basic approach. Some examples include Module Networks (Segal *et al.*, 2003) and Regulatory Modules (Bar-Joseph *et al.*, 2003), which add variables to the graphical model in order to identify co-regulated modules. For further details, see the review by Friedman (2004). Another class of approaches reconstructs regulatory network structure by first calculating the mutual information between pairs of expression profiles, and then choosing a mutual information threshold at which to call the interactions relevant (Butte & Kohane, 2000; Margolin *et al.*, 2006).

As other large-scale biological data sets have become available, network reconstruction methods have been generalized to exploit them. For example, Sachs *et al.* (2005) reconstruct regulatory networks from phospho-proteomic and phospho-lipid measurements, and Gibbs *et al.* (2013) seek to identify functional modules from protein co-abundance

data. Other sources of network structure information have been used to inform network reconstruction approaches in conjunction with expression profiles. Sometimes the complete graph structure is provided as input, and the task is to learn the parameters from the expression data (Gat-Viks *et al.*, 2006; Gat-Viks & Shamir, 2007; Saez-Rodriguez *et al.*, 2009). Other approaches use an input network as a prior (Imoto *et al.*, 2003) or require that inferred regulatory dependencies are supported by short chains of protein-protein or protein-DNA interactions (Novershtern *et al.*, 2011). The use of protein interactions to make hypotheses about biological mechanisms for inferred regulatory structures is a recurring theme in the other approaches discussed in this section.

The optimistic goal of these methods is to accurately describe causal relationships within the cell. Time series gene expression measurements can be argued to assist in the claim of causality. Some approaches that have been applied to time series include dynamic Bayesian networks (Ong *et al.*, 2002), Granger causality (Dahlhaus & Eichler, 2003), ordinary differential equations (Bonneau *et al.*, 2006), inductive logic programming (Ong *et al.*, 2007), and input-output hidden Markov models (Ernst *et al.*, 2007; Schulz *et al.*, 2012). Another benefit of using time series data is that it can allow for inference of feedback loops, which are common in nature, but likely impossible to accurately infer from single measurements at steady-state.

## Comparison

Like the tasks we approach in this thesis, these network reconstruction methods infer signed, directed networks and quantify the certainty of each predicted relationship represented in the network.

The edges in the reconstructed networks are intended to represent causal relationships, but not necessarily direct, physical interactions. With limited exceptions, in our approach, we restrict the possible edges in our subnetworks to known physical interactions from a background network. Our exceptional category of edges is described in Chapters 1 and 3: we allow the method to predict a limited set of edges between the host and virus. By using a physical background network, our methods are able to predict additional relevant genes that are outside of the set of input, assayed genes. This is like the approach of Novershtern *et al.* (2011), but unlike many others in this category. Generally, the network reconstruction methods described in this section can infer new relationships among assayed genes, but not new players.

In contrast to these expression-profile-based methods, which compare available regulatory information between conditions or over time, the methods proposed in this thesis

take a different kind of input data. Our primary data sets are from gene suppression experiments and consist of labels that describe the effect of suppressing one gene on another phenotype of interest. We also use additional data sets that take the form of categorical labels that describe the relationship of a gene product to a phenotype of interest. In the host-virus inference setting of Chapters 3 and 4, we use a set of host genes that modulate viral replication. In Chapter 4, we also have a set of host-virus protein-protein interactions. The data we use for signaling subnetwork inference in Chapter 5 is a bit different. In addition to phenotype labels from gene-suppression assays that measure the fitness of the mutant under stress, we have access to genome-wide expression in a limited number of single-gene knockouts, and another set of labels, indicating differential phosphorylation of proteins under the stress condition being studied.

## 2.2 Subnetwork inference based on results of perturbation experiments (Figure 2.1B)

Another general experimental method for gaining information about gene function or regulatory network structure is to perform genetic perturbation experiments. In these experiments, a gene is knocked out or suppressed (as in deletion and dox-repressible libraries and RNA interference), and a cellular phenotype is measured. Then, a statistical test is applied to assess the deviation from the wild type value. Genes whose perturbations result in a significantly altered phenotype are referred to as *genetic hits*, or, more generally, *hits*, and are inferred to be involved in regulating the studied phenotype.

The cellular phenotype may be a single quantity or a vector of values. Some relevant examples of single-valued cellular phenotypes include the amount of viral replication product in the host cell (Chapters 3 and 4) and the fitness of a cell under a stressful condition (Chapter 5). A commonly measured multiple-valued phenotype is genome-wide gene expression; a group of expression measurements is called an *expression profile*.

This section considers the case in which genome-wide expression profiles are measured in a series of single-gene perturbations. From these data, we can derive a list of what is often referred to as **source-target pairs**. In each pair, the *source* is the gene that was suppressed, and the *target* is one of the genes that is **differentially expressed** in the perturbation condition compared to wild type. From this relationship, the source gene product is inferred to play a role in regulating the target gene's expression. Another way to refer to this kind of relationship is to say that the source is *upstream* of the target. Many researchers have approached the problem of hypothesizing the mechanism of the source-target interactions

by finding paths through a protein-protein and protein-nucleic acid background network. This task is highly similar to the tasks that we address in this thesis, which we illustrated in Chapter 1.

We call this the *subnetwork inference* task. The input is a set of source-target pairs, and a background network. The output is a subnetwork in which the edges hypothesize the physical mechanism of the measured phenotype. This means that included edges are considered relevant, while non-included edges are considered irrelevant. Additional information may be predicted as well, including signs or directions on edges, and signs on relevant nodes that predict what the effect will be on the phenotype when the node's gene expression is suppressed.

In the following sub-sections, we give an overview of the work that is most related to this thesis. A review of other approaches to this problem of "understanding the cell by breaking it" is provided by Markowetz (2010).

### 2.2.1 Inferring mechanistic subnetworks to explain source-target pairs

Some successful methods that have been used to analyze this type of data are Boolean networks (Liang *et al.*, 1998; Akutsu *et al.*, 1999; Ideker *et al.*, 2000), logical models (Reiser *et al.*, 2001; Tamaddoni-Nezhad *et al.*, 2006), and graphical models (Markowetz *et al.*, 2005). Here we will discuss in detail a few of the most related approaches, which use factor graphs (Yeang *et al.*, 2004) or integer linear programming (Ourfali *et al.*, 2007) and a protein-interaction background network.

In the basic version of this problem, the input data are the set of source-target pairs and a *background network* (depicted in Figure 2.1B, Input panel). Often, genes and their protein products are represented as separate nodes in this background network. The source labels are generally assigned only to the protein version of the node, whereas the target labels are assigned to gene nodes. The output (Figure 2.1B, Output panel) is a subnetwork that predicts the direct interactions by which the source perturbation results in differential expression of the target. The subnetwork consists of a set of linear paths, each of which is an acyclic chain of edges that begins with the source and ends with one of the source's targets. Generally, the final interaction in each path must represent the binding of a transcription factor to the target gene. The paths that are chosen to connect sources and targets are deemed *relevant*, and consequently so are the nodes and edges in those paths. All other nodes and edges are considered irrelevant to the biological condition being studied.

Each approach specifies additional desiderata for the inferred subnetwork. Often, signs and directions are required to be inferred for relevant edges when this information is missing from the background network:

- The directions of edges must be inferred so that one can follow all relevant paths forward from the sources to the targets. Paths for which this is impossible will be deemed irrelevant.

- Signs are generally inferred when the input experimental data provides a *sign* to each source-target pair. Similar to the edge signs discussed in Chapter 1, a positive sign is given to a pair for which the source perturbation causes a significant **induction** of the target's expression, and a negative sign is assigned in the case of a significant **repression** of target expression. It has been proposed that the signs of regulatory interactions among the *source* proteins can be inferred using the signs of the pairs. Toward this end, several methods enforce *sign consistency* along the paths. This entails inferring a sign on each predicted relevant edge (or node) so that the paths consistently predict the change in expression of the target gene in the source knockout and any other sources that appear along the same path.

In the integer linear programming formulation of this problem, these requirements (or preferences) are encoded as constraints; in a factor graph formulation, they are encoded as potential functions.

There are generally many configurations of relevant paths that could be used to connect the source-target pairs. Consequently, an objective function is specified to find the most optimal configuration. (Often, the optimal configuration is still not unique.) The methods that approach this problem generally differ based on their objective functions. For example, some approaches prefer inferred subnetworks that provide multiple paths between each source-target pair (Yeang *et al.*, 2004; Ourfali *et al.*, 2007), whereas others prefer a parsimonious subnetwork (Maeyer *et al.*, 2013).

Other efforts to analyze source-target pair data with sign consistency have been proposed in the absence of protein interaction data (Peleg *et al.*, 2010) or with the option of using protein interaction data as a prior rather than a hard constraint (Vaske *et al.*, 2009). One interesing contribution of the method of Vaske *et al.* (2009), Factor Graph Nested Effects Models, and its predecessor, Nested Effects Models (Markowetz *et al.*, 2005), is the way in which they use patterns of target overlap to infer both ordered and signed relationships between the sources. The pattern of the overlap is used to constrain the structure among the sources.

### 2.2.2 Global network orientation

A similar problem is one of *global network orientation* to allow directed connections between upstream sources and downstream targets. While the methods discussed in the previous section enforce sign consistency and infer which individual paths are truly relevant, the focus of network orientation approaches is to assign directions to all edges in order to allow a large number of source-target pairs to be connected via directed paths. Paths may later be ranked, but are generally not inferred to be relevant or irrelevant. Most of these approaches also assign weights to edges (and therefore paths) and prioritize directing high-confidence paths over low-confidence ones. Path weights are generally assigned such that short paths receive higher weights.

These approaches generally maximize the number or combined weight of directed paths between source-target pairs (Medvedovsky *et al.*, 2008; Gitter *et al.*, 2011; Blokh *et al.*, 2013). Other contributions in this area include efficient integer linear programming formulations (Silverbush *et al.*, 2011) and other algorithms for finding approximately optimal orientations (Medvedovsky *et al.*, 2008).

### 2.2.3 Global network orientation using separate source and target sets

Network orientation methods have also been extended to use separate sets of source proteins and target genes. To do so, paths between any source and any target are considered, rather than only those with a relationship identified by a perturbation experiment. Gitter *et al.* (2013) infer the signaling network in response to stress by combining network orientation (Gitter *et al.*, 2011) with time series-based inference of regulatory networks (Ernst *et al.*, 2010). The regulatory network inference component identifies which transcription factors can be used by the network orientation component to connect the upstream hits to the downstream targets. In another application of this method, Gitter & Bar-Joseph (2013) combine HIV genetic hits from RNAi screens with a set of genes that are differentially expressed under HIV infection. Specific differences between this method and our approach to the host-virus subnetwork inference problem are described in more detail in Chapters 3.1.1 and 4.1.1.

### 2.2.4 eQTL prioritization

The methods above attempt to connect all source-target pairs. A different problem that has been approached with similar methods is that of identifying the causal gene within an expression quantitative trait locus (eQTL). eQTLs are genomic regions in which sequence variation is observed to correlate with the expression level of one or more target genes.

Typically, a locus contains multiple genes. The task is to rank the candidate sources for association with the targets. An early example of assaying the entire yeast genome for eQTLs is Brem & Kruglyak (2005), who provide measurements of near-genome-wide expression and genotype data for nearly 3,000 markers spaced at regular intervals across the yeast genome. Several groups have approach the eQTL problem as one of identifying causal structure using a probabilistic model representation (Schadt *et al.*, 2005; Lee *et al.*, 2006; Kulp & Jagalur, 2006; Pérez-Enciso *et al.*, 2007).

As in the case of the regulatory network inference task, it has proven useful to use a protein-protein interaction background network to postulate mechanistic explanations of the connection between the candidate causal genes and their associated targets. Methods that use this bias rank candidates according to some measure of their proximity to the targets. Proximity has been measured in various ways, including by computing the expected length of random walks through the background network (Tu *et al.*, 2006), by modeling the background network as an electrical circuit and simulating the flow of current (Suthram *et al.*, 2008; Kim *et al.*, 2011), by finding the $k$-shortest paths between each source and all of the targets (Shih & Parthasarathy, 2012), and by using kernels on graphs that represent different kinds of random walks (Verbeke *et al.*, 2013).

### 2.2.5 Reconstructing physical subnetworks from genetic interactions

Having measurements of a single-valued cellular phenotype, such as fitness, from both single and double gene perturbations (or even more) may also shed light on the order of gene products in the subnetwork. When a double-gene perturbation results in a cellular phenotype that does not match an expected combination of the two single perturbation phenotypes, it is called a *genetic interaction* or *epistatic interaction*. When the single perturbations both affect the phenotype, it is proposed that the result of the double-perturbation may indicate whether the two genes function in the same linear pathway or in parallel, compensatory pathways (Avery & Wasserman, 1992). Several algorithmic approaches infer subnetworks from this type of data (Kaufman *et al.*, 2005; Yosef *et al.*, 2006), including some that also use protein-protein interactions to predict relevant connections (Kelley & Ideker, 2005; Carter *et al.*, 2007).

### 2.2.6 Comparison

In approaching our problems, we draw inspiration from many of the approaches discussed in this section (Sections 2.2.1 and 2.2.2 in particular). We use a similar computational approach: we employ integer linear programming to infer subnetworks that have some

similar characteristics to the subnetworks inferred by these methods. These related methods infer mechanistic paths, and in doing so predict edge signs, directions, or both. They can predict the relevance of additional unassayed gene products. Most of the methods provide a way to quantify the certainty about predicted subnetwork elements, or at least rank them.

However, we consider different types of input data, and have therefore designed different constraints and objective functions. We also use a more expressive background network representation, as we include protein complexes and metabolic pathways in addition to protein-protein interactions.

In Chapter 3, instead of having source-target pairs as input, we have a set of hit proteins known to be relevant to viral replication. Our method infers paths to connect these hits to a node representing the virus. It is almost as if we have a set of sources with a common target. However, the points of direct interaction between the host and virus are unknown and must be inferred. Like the methods of Yeang *et al.* (2004) and Ourfali *et al.* (2007), we infer edge signs along the paths and maximize the size of the subnetwork (subject to constraints). Like the methods that approach edge orientation, we also infer edge directions.

In Chapter 4, as in Chapter 3, our input is a set of genetic hits involved in viral replication. We also have a separate set of host proteins that have been observed to bind with viral proteins. We consider the task of inferring paths to connect the hits to the viral proteins.

In Chapter 5, one of our input data sets consists of source-target pairs identified under salt stress conditions. In this way, our approach is highly relevant to the methods discussed in this section. However, other aspects of our input data are different, as we integrate two additional hit sets that do not have corresponding targets. Like many of the methods described in this section, we orient the edges in the inferred subnetwork, but do not enforce sign consistency along the paths. Our objective function balances the inclusion of these known relevant proteins with a preference for a parsimonious subnetwork.

## 2.3 Subnetwork extraction to connect one or more input hit sets (Figure 2.1C)

In this section, we discuss methods for finding relevant subnetworks to connect two different types of input data:

- *One hit set.* A set of hit genes that have been identified as relevant to a condition of interest, as from a genome-wide suppression experiment. (This setting is discussed at a high level in Chapter 1 and in detail in Chapter 3.)

- *One hit set and one target set.* Both (i) a set of hits from a genome-wide suppression experiment and (ii) an independently derived set of genes that are differentially expressed in the wild type in that same condition. While the assumption is often made that the target genes are downstream of the source proteins, there is no relationship between specific pairs as there is in the previous setting.

We categorize nearly all of the methods in this section as *subnetwork extraction* methods because they primarily predict relevant nodes and assume that all edges in the background network are relevant. The extracted subnetworks typically only include enough edges to connect the input hits. In contrast, *inference* methods go a step further to also predict which edges are relevant, and to infer other missing information such as edge sign and direction. An illustration of subnetwork extraction from one hit set is provided in Figure 2.1C.

### 2.3.1 Subnetwork extraction to connect a set of hits

Under the assumption that protein interactions provide information about cellular mechanisms, several methods have been developed to extract connecting structures from a background network to explain a set of hits. Some of these methods use weights on the edges, such as values that indicate the reliability of the experiments that identified the edges.

Some approaches are based on local objectives: for example, finding paths in order to connect one hit gene to another with the shortest or most highly scoring path. Scott *et al.* (2006) provide biologically motivated extensions and applications of a general high-scoring path-finding algorithm of Alon *et al.* (1995). Other methods integrate protein-protein interactions and mRNA expression profiles, scoring edges according to correlation of the expression profiles of the interacting nodes (Steffen *et al.*, 2002). Faust *et al.* (2010) provide a review and benchmarked comparison of several similar methods for extracting subnetworks that connect hits. The methods that they cover define subnetworks based on shortest-paths (or $k$-shortest paths), random walks, and Steiner trees.

Other approaches globally optimize a quantitative measurement of the subnetwork. A popular objective function to use is a variation on the Steiner Tree problem: to connect as many hits as possible into a tree structure, using a minimal number of connecting edges, and allowing additional intermediate nodes. Nodes and edges can be weighted; the weighted version of the problem is referred to as the Prize-Collecting Steiner Tree (PCST) problem. To our knowledge, the original application of the Steiner tree algorithm to biological data is due to Scott *et al.* (2005). Dittrich *et al.* (2008) also approach the PCST problem and contribute a scoring function for converting $p$-values (indicating the significance of experimental results)

to node scores. Yosef *et al.* (2009) provide another variation: inferring a directed Steiner tree using the algorithm of Charikar *et al.* (1999). Their method also orients the edges such that, in the subnetwork, each hit lies upstream of a pre-specified node.

### 2.3.2 Extracting subnetworks to connect upstream hits to independently derived downstream targets

As mentioned in the introduction to this section, another setting that has been approached is one in which two experimental gene or protein sets are available: one set of hit proteins identified by gene perturbation assays, and one set of gene targets, whose change in expression is assumed to be regulated by the hits. The following methods integrate these data with a protein-protein or protein-DNA background network and seek to extract the relevant paths that connect the two sets.

In their ResponseNet method, Yeger-Lotem *et al.* (2009) cast the problem as maximizing network flow. In the process, they add an additional source node (connected to all of the hits) and an additional sink node (connected to all of the targets). Viewing the problem as the Prize-Collecting Steiner Tree has also been a popular approach. Huang & Fraenkel (2009) introduce the idea of incorporating multiple hit sets and target sets, each derived from a different experimental data source. Some recent extensions to the PCST formulation include the Prize-Collecting Steiner Forest (PCSF) (Tuncbag *et al.*, 2013), which allows the extraction of multiple, disconnected trees, and the multi-sample PCSF (Gitter *et al.*, 2014), which can be used to gather a collection of patient-specific PCSFs.

### 2.3.3 Comparison

Like the methods proposed in this thesis, these subnetwork extraction methods use a background network to identify a relevant subnetwork that connects experimentally identified input hits, and thereby predict the relevance of additional hits. Many of these approaches also offer a way to rank the predicted hits. However, with the exception of the directed Steiner tree method (Yosef *et al.*, 2009), they do not directly infer the irrelevance, direction, or sign of edges.

Like our work, many of these methods integrate multiple experimental data types and employ an objective function that balances the inclusion of known relevant nodes against the inclusion of additional predicted relevant nodes. However, our objective functions are more generous, and try to pick up multiple parallel paths in the inferred subnetworks. These methods, by contrast, extract edge-sparse subnetworks, taking the philosophy that

background network edges indicate functional relatedness, and that the edges selected for the extracted subnetwork are not necessarily mechanistic.

A unique feature of our approach in Chapter 3 is that we predict which host factors are host-virus interfaces. This is facilitated by the addition of a new node to the background network, which represents the virus and is initially connected to all host factors. The inference method predicts a subset of relevant host-virus interactions and orients the subnetwork so that the virus node is located at the 'bottom'. None of the subnetwork extraction methods predict host-virus interactions, but some of them do add a new node to the background network in order to loosely orient the extracted subnetwork (Yosef *et al.*, 2009; Tuncbag *et al.*, 2013; Yeger-Lotem *et al.*, 2009). However, with the exception of the directed Steiner tree method (Yosef *et al.*, 2009), these related methods do not actually infer edge directions.

## 2.4 Candidate gene prioritization using a protein background network (Figure 2.2D)

Genetic hit sets identified by gene-suppression experiments are unlikely to be complete due to confounding biological factors such as functional redundancy and off-target RNAi effects. In other cases, a small set of known relevant genes has been identified through expensive low-throughput experiments or other means. A computational task that arises is the following: given a known set of hits, predict and rank which other genes are likely to be relevant to the cellular response being studied. One popular name for this task is *candidate gene prioritization*. A common application area is in the identification of genes that are relevant to human disease. Many methods have been developed to approach this problem (see reviews by Navlakha & Kingsford (2010) and Börnigen *et al.* (2012)). Candidates may be restricted to genes within a locus identified by linkage analysis, or include the entire genome.

Our work is most closely related to those methods that use a protein interaction background network as a way to identify candidates with high similarity to the known hits. Similarity can be measured using graph methods, including betweenness, diffusion kernels, random walks (Köhler *et al.*, 2008) and iterative propagation (Vanunu *et al.*, 2010). In addition to the input hit set, some methods incorporate other data, such as information about other diseases for which network genes have been identified as hits (Chen *et al.*, 2011). One particularly relevant method is that of Murali *et al.* (2011), which prioritizes human genes for relevance to HIV using the same RNAi screens that we study in Chapter 4.

We illustrate a diffusion kernel method for gene prioritization in Figure 2.2D. The input is a hit set and a background network, with hits shaded in dark pink. (We use a darker pink here than in other figures in order to allow better visualization of the output.) In the output network, the hits' labels are diffused throughout the network, decreasing with distance. The intensity of the color indicates the closeness of the new predicted hits to the original hit set. The predicted hits are ranked in order of confidence.

### Comparison

The greatest difference between our task and gene prioritization is the type of output. Gene prioritization methods generally only rank candidate genes and use the protein interaction network as one type of input data. Our goal, however, is to infer subnetworks that posit mechanistic explanations for the hit set as well as prioritize additional candidates.

## 2.5 Protein complex prediction (Figure 2.2E)

Most of the high-throughput experimental methods that are used to identify protein-protein interactions test only pairs of proteins at a time, and can therefore only identify binary interactions. However, in the cell, the relevant functional unit is just as often (or more often) a protein complex. Much algorithmic work has gone into predicting functional complexes from protein interaction networks using topological information and optionally a set of seed proteins. We refer to this task as *complex prediction*.

Often, these approaches are a variation on identifying densely connected sets of proteins. A challenging aspect of this problem is that biological networks generally have a power-law-like distribution of node degree, and cannot easily be split into connected components. Each complex prediction method must provide some way to cut the background network to determine which proteins are in each complex. Some determine complexes based on the density of connectivity among nodes, and provide tunable local density thresholds (Bader & Hogue, 2003). Others leverage additional sources of evidence, such as gene expression data (Maraziotis *et al.*, 2007), to posit which proteins are likely to function together.

An illustration of the task is given in Figure 2.2D. The input is a background network; the output is a set of predicted protein complexes.

### Comparison

Compared to the approaches proposed by this thesis, the goal for complex-prediction methods is to identify relatively small, individual functional units that persist across many

cellular conditions. In our setting, we wish to identify which interactions are truly relevant to our condition, and infer subnetworks that provide mechanistic explanations for our input data. The subnetworks may connect multiple cellular functions.

## 2.6 Network filtering and integration (Figure 2.2F)

A consideration that we have made in this thesis work is how to integrate the different types of network data (including protein-protein interactions, protein-DNA interactions, and metabolic pathways). The following chapters in this thesis discuss our decisions in detail. Two related problems include network *filtering* to identify a context-specific background network, and network *integration* to predict new relevant nodes and edges by combining multiple network data sets.

### 2.6.1 Context-specific network filtering

In Chapter 4, we propose a method for providing tailored views of an inferred subnetwork based on a biologist's queries. In our setting, we apply this filtering step to the inferred subnetwork. Other scientists have developed methods to identify context-specific protein interaction networks using expression data gathered from specific tissues or conditions, Gene Ontology annotations, or other indications of context-specificity. We illustrate network filtering in Figure 2.2F. The input is a set of genes that are observed to be expressed in a particular cell type, and the output is a filtered background network that contains only cell-type-relevant genes.

Simple network filtering approaches have been applied to assist biologists in viewing and interpreting data, and several interaction network browsers have made this feature available. When a tissue-specific gene or protein set is available, one simple filter for interaction data is to accept only edges in which both gene products are in the tissue-specific list. This kind of filter has been made available in Cytoscape plugins (Yang *et al.*, 2008) and web tools (Schaefer *et al.*, 2013). In their POINeT software package, Lee *et al.* (2009) implement a tissue-specificity ranking scheme from Hsu & Taksa (2005) by measuring the significance of the difference in a node's degree in the tissue-filtered subnetwork compared to the global interaction network. Schaefer *et al.* (2013) also report an increase in precision in a candidate gene identification task when a tissue-specific filter is applied. Lopes *et al.* (2011) use tissue-specific filtering to perform a comparative analysis of several popular interaction databases and to make recommendations for parts of the interaction network that may benefit from additional experimental attention. Manually curated data sources

have also been used in lieu of tissue-specific experimentally derived gene lists. Lan *et al.* (2011) adapt a source-target path-inference method to score edges by the interacting nodes' overlap in Gene Ontology annotations.

Another related task is to identify subnetworks that are highly responsive to the condition of interest compared to baseline measurements of gene expression. A goal is to identify subnetworks that are signatures of a specific disease state and can be used to either provide interpretation of the disease or assist in diagnosis. Approaches to this problem include methods that focus on identifying densely connected, highly differentially expressed nodes (Ideker *et al.*, 2002), edges between strongly correlated nodes (Guo *et al.*, 2007), or a combination of the two (Ma *et al.*, 2011). Other methods explicitly cast the problem as a classification task, using network features to discriminate between disease and non-disease expression profiles (Nibbe *et al.*, 2010; Su *et al.*, 2010; Dutkowski & Ideker, 2011).

### 2.6.2 Network integration

Several previous efforts have considered how to combine multiple biological network data sets in order to perform classification or collaborative recommendation tasks. A popular example application is to predict Gene Ontology annotations. Some relevant approaches employed for this purpose include kernel matrix completion (Kato *et al.*, 2005), special kernels or combinations of kernels that are used in support vector machines (Pavlidis *et al.*, 2002; Tsuda *et al.*, 2005; Lippert *et al.*, 2010), and matrix factorization (Zitnik & Zupan, 2014). Because the trained models represent graphs, they predict not only labels for genes, but also new edges that connect the genes to the input networks.

### 2.6.3 Comparison

Network filtering and integration methods have more in common with subnetwork extraction methods than with subnetwork inference methods like our own, in that they can be used to identify network structures that may be relevant to a condition of interest, but do not infer mechanistic subnetworks. They do not infer paths to explain experimental data, or infer signs or directions on edges.

One of the sub-tasks we consider in Chapter 4 is related to network filtering: the task of identifying a specific 'view' into the subnetwork to show connections to a biological process or concept of interest. However, our approach forgoes context-filtering of the *input* background network because most of the HIV-relevant human genetic hits did not even register on lists of genes and proteins expressed in HIV-relevant tissues. We perform

function-specific filtering *after* inference so as not to bias the inference process toward the gene set that we use as a filter.

In contrast to the network integration task, the background network in our setting only contains interactions that represent direct, physical interactions. We use other data to determine which of those interactions are relevant, but not to predict new edges. Also, in the collaborative filtering setting, there are many target classes. In contrast, we only try to predict which genes are relevant to one condition.

## 2.7 Gene set enrichment analysis (Figure 1.1A-B)

As mentioned in Chapter 1, gene set enrichment techniques are widely used to interpret hit sets identified by high-throughput experiments. These methods identify which predefined biological components, functions, and processes, such as Gene Ontology (GO) annotations or KEGG pathways (Ashburner *et al.*, 2000; Kanehisa *et al.*, 2012), are represented in a set of genes (Huang *et al.*, 2009). Others have expanded upon these methods by incorporating network data (signed or unsigned) to improve the specificity of the enriched gene sets (Liu *et al.*, 2007; Ulitsky & Shamir, 2007; Geistlinger *et al.*, 2011). Recently, model-based enrichment analysis methods have been proposed as an alternative to the traditional hypothesis-testing approach (Lu *et al.*, 2008; Bauer *et al.*, 2010; Wang *et al.*, 2013). These methods aim to find a small set of GO annotations that, together, cover the input gene set.

### Comparison

These methods identify pre-defined gene sets that are represented by an input gene set, rather than inferred subnetworks that offer mechanistic explanations for the input gene set. In contrast to enrichment-based methods, our approach does not rely on predefined gene sets when predicting which interactions are relevant to host-virus interactions or stress signaling. Enrichment methods also suggest interpretations of a hit set at the level of gene sets. Instead, our method predicts mechanisms at a finer granularity, by predicting which physical interactions are relevant. We do use gene set enrichment analyses, however, to evaluate the accuracy of our inferred subnetworks.

# 3 Inferring host-virus interaction subnetworks in a yeast host

---

This chapter presents our initial method for inferring interpretable, host-virus interaction subnetworks. The method combines a diffusion kernel method (for candidate gene prioritization) with an integer linear program for inferring signed, directed subnetworks. This represents a real-life version of the basic task that we introduced in Chapter 1, in which the input is a set of hits identified from a genome-wide suppression experiment, plus a background network. We apply our method to data from experiments screening a yeast host genome for genes that modulate the replication of two RNA viruses.

Because a complete, gold-standard subnetwork is unavailable, we evaluate our method using both computational and literature-based assessments. The inferred subnetworks are highly accurate, and enriched with a number of protein complexes and Gene Ontology terms. Our comparison to a random control suggests that these results are not simply due to topological properties of the input data. Importantly, several of the predicted relevant host factors and interfaces are supported by the literature.

**The work in this chapter is published with the following citation:**
Deborah Chasman, Brandi Gancarz, Linhui Hao, Michael Ferris, Paul Ahlquist, Mark Craven (2014) Inferring host gene subnetworks involved in viral replication.
*PLoS Computational Biology* **10**(5):e1003626. doi: 10.1371/journal.pcbi.1003626

**Supporting website:**
`http://www.biostat.wisc.edu/~craven/chasman_host_virus/`

## 3.1 Introduction

A virus requires host cellular machinery to complete its life cycle. Understanding the interactions that occur between viruses and their hosts can contribute to the development of preventative and therapeutic methods to control their effects on human health.

In Chapter 1, we introduced the idea of systematically suppressing host gene products in order to identify host genes that modulate the virus life cycle in a host cell. Gene suppression is carried out using yeast mutant libraries or RNA interference, among other technologies. For each host gene that is manipulated, the effect on the virus is assessed by measuring the replicative yield of viral genetic material or viral proteins relative to a control. Typically, these genome-wide screens identify a large number of host genes, which we refer to as *hits*, whose loss has a significant effect on the virus. However, the screens themselves do not reveal how the gene products of these hits are organized into the pathways that modulate the virus, nor do they indicate which host gene products directly interface with a viral component. We consider the computational task of inferring directed subnetworks that hypothesize the pathways through which each hit modulates viral replication. The value of these inferred subnetworks is that they can be used to (i) predict which unassayed genes may be involved in viral replication, (ii) interpret the role of each hit in modulating the virus, and (iii) guide further experimentation that is aimed at uncovering and validating the mechanisms of host-virus interaction.

We present an approach that uses an integer linear program (IP, for brevity) to infer the pathways that are involved in the lifecycle of a virus in a host cell. The inputs to our approach are the list of phenotypes measured in a genome-wide loss-of-function assay, including a list of those host genes that are hits, and a partially-directed *background network* characterizing known physical interactions among host cellular components. Using these data, our approach predicts the identity of a small number of host-virus *interfaces* (host factors that are closest to a direct interaction with the virus), and infers a subnetwork of directed interactions that provides at least one path from every hit to a predicted interface. By providing these paths, we say that the subnetwork plausibly *explains* or *accounts for* the viral phenotype observed when each hit is suppressed. Because the background network and experimental observations are incomplete, many different subnetworks may be inferred for the same set of hits. To account for this, our method infers an ensemble of subnetworks, each of which provides paths for all of the hits. We use the ensemble to assess our confidence in various aspects of the predicted subnetworks.

Figure 3.1 provides an illustration of the input and output of our computational approach. The example is similar to what we presented in Figure 1.1, but more detailed.

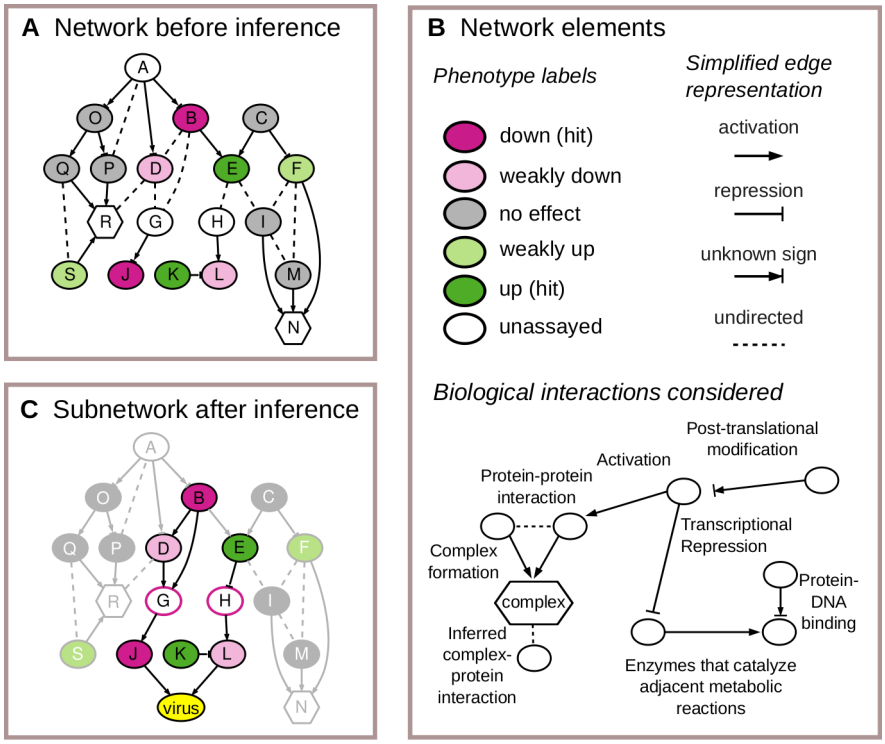



**Figure 3.1:** Input and output for our subnetwork inference approach.
**A** The inputs to our subnetwork inference approach are phenotypes measured in a loss-of-function assay and a background network characterizing known interactions.
**B** The network elements represented in panels **A**, **C**, and other figures.
**C** An inferred subnetwork for the given inputs. The subnetwork includes a directed, consistent path linking each hit (gene with an **up** or **down** phenotype) to the virus. The red borders on the unassayed nodes G and H indicate that they are inferred to have the **down** phenotype. Edges shown in gray are not included in the subnetwork.

Figure 3.1A shows what is provided as input to the approach using a graph representation. Nodes in the graph represent host genes, proteins, and protein complexes. Both the gene and its encoded protein are represented using the same node. The connecting edges in the graph provide a simplified representation of known interactions among the nodes.

Figure 3.1B presents a graphical guide to the network elements used by our method. The color of a gene node specifies the observed phenotype when expression of the gene's product is suppressed. Using the loss-of-function assay data, we derive discrete viral phenotype labels that describe the sign and magnitude of the measured effect of each host gene on viral replication: down and **weakly-up** for genes whose loss of function reduces viral replication, up and **weakly-up** for genes whose loss increases viral replication, and **no-effect** for genes with no consistent, measurable effect on viral replication. (Note that we have expanded the possible phenotype labels from those used in the example

from Chapter 1, shown in Figure 1.1.) The figure also shows the types of interactions in the background network and how they are distilled into a simplified representation. Each interaction is represented by an edge indicating the *direction* and *sign* (activation or inhibition) of the interaction, when these properties are known.

Figure 3.1C shows the result of the inference process, which is a directed subnetwork that accounts for the loss-of-function phenotype of each hit (B,E,J,K) by providing potential mechanistic paths leading to a direct interaction with the virus. In the subnetwork shown, host gene products J and L are predicted to be interfaces between the host and virus, as indicated by the directed edges to the virus node. Some of the edges and nodes, shown in gray, are deemed to be not relevant to viral replication, and hence not useful for explaining the measured hits; these include all genes with **no-effect** (gray) viral phenotypes. The dark edges, which are considered part of the inferred subnetwork, are assigned directions and signs in cases where these properties are not specified by the background network. The directions for the relevant edges are set so that for each hit, there is at least one path that proceeds forward from it to the virus. The signs for the relevant edges are set so that each one gives a biologically plausible interpretation of how the interaction is relevant to viral replication. For example, protein E has an **up** phenotype and modulates the virus by inhibiting the expression or function of protein H, which activates the function or expression of the interface protein L. Additionally, the subnetwork predicts that two genes whose phenotypes are unknown (G, H), and two genes whose phenotypes are weak (D, L), are actually key host factors involved in viral replication.

The integer linear program used in our approach consists of an objective function and a set of constraints characterizing subnetworks that are deemed biologically interpretable. Due to functional redundancy in the host genome and the inability to assay some host-gene suppressions, many true hits are not identified by individual loss-of-function experiments. Therefore, to predict additional hits and to identify multiple paths between hits and interfaces, our objective function maximizes the inclusion of unassayed genes and genes with weak viral phenotypes, subject to other constraints on the subnetwork. These genes are prioritized using a diffusion kernel (DK) scoring method, which assigns scores to genes based on their network proximity and connectivity to the hits. As a counterpoint to the objective function, which is generous in including genes in the subnetwork, the IP's constraints provide restrictions on which paths may be inferred to be part of the subnetwork. All of the inferred paths must be *directed*, meaning that each interaction in a path is directed forward from the hit to the virus, and directions are inferred for undirected interactions. The paths must also be *consistent*, meaning that the sign (activating or inhibitory) of each interaction between host factors agrees with the viral phenotypes of the interactors. For a

pair of host factors that both inhibit or both facilitate viral replication when suppressed, an activating interaction is consistent. For a pair of host factors that affect the virus in opposite ways, an inhibitory interaction is consistent. Using these rules, our method infers the signs of unsigned interactions and the viral phenotypes for unassayed host factors.

We assess the inferred subnetworks using both computational experiments and an analysis of the relevant literature. First, we conduct a cross-validation experiment to evaluate the accuracy of our inferred subnetworks in predicting host factors involved in viral replication. We compare the accuracy of our approach to several baselines including a diffusion kernel method which is used as an input to our approach. Our results demonstrate that (i) the high-confidence predictions of our IP approach achieve a high level of accuracy, (ii) the predictions made by our method are more accurate than those made by several baselines, and (iii) the accuracy of our method for this task is comparable to the diffusion kernel method which does not infer detailed causal pathways like our IP approach. Second, we use our approach to predict a set of host-virus interfaces and a set of unassayed host genes that are likely to be modulators of viral replication. We discuss independent biological evidence that supports a number of these predictions. Finally, we perform a suite of additional computational experiments to assess our method's predictions in other ways. These include (i) a comparative analysis to IP components inspired by related work, (ii) a Gene Ontology analysis to evaluate the ability of our inferred subnetworks to better identify relevant functional categories than an analysis of the experimental data alone, and (iii) a Monte Carlo analysis to assess whether the protein complexes that our method predicts to be relevant are well supported by the experimental data and subnetwork-inference process.

### 3.1.1 Related Work

As discussed in Chapter 2, our work is related to methods that address several different categories of problems. Here, we provide a more detailed comparison to a few specific categories and approaches.

Our work bears some similarities to integer-linear-programming-based methods for *subgraph inference* from source-target pairs (Chapter 2.2). However, our method differs in some key respects. In our setting, the common target of all hits – the virus – is external to the background network, and the identity of the host factors that interact with it directly must be predicted. Additionally, our background network encompasses a greater variety of biological interactions than the background networks used by these other approaches. Unlike the methods that use mRNA expression profiles as the basis for determining direct

or indirect relationships between genes, ours uses only phenotypes derived from a genome-wide mutant assay.

Recently, Gitter & Bar-Joseph (2013) presented an application of their source-target pair-based method to inferring human signaling pathways involved in influenza A viral infection. In their approach, sources are human proteins that are known to directly interact with a viral component, analogous to the interfaces in our conceptual model. Targets are human genes whose expression is measured over several time points during viral infection. The method orients paths through a protein-protein interaction network from the sources to the targets, preferring paths that contain influenza-relevant genes identified by RNAi experiments. Conceptually, this method infers the signaling pathways that control the host's transcriptional response to viral infection. In this paper, we look at host-virus interactions from the opposite direction and infer the mechanistic pathways by which suppressed host genes inhibit or enable the normal viral replication cycle.

We also draw comparison to *subgraph extraction* methods (Chapter 2.3). Unlike our method, these extraction approaches do not distinguish (or infer) phenotype signs and edge signs, nor do they apply global constraints to the extracted subnetwork other than a global edge minimization. In contrast, we employ global constraints such as an upper bound on the number of interfaces. We do not believe that for our task it is appropriate to assume the entire network will be minimal, which is an assumption made by the methods that extract a Steiner tree or set of shortest paths between hits.

An additional unique feature of our approach is that we predict which host factors are host-virus interfaces by adding a new node to the background network. This node represents the virus, and is initially connected to all host factors. The inference method predicts a subset of relevant host-virus interactions and orients the subnetwork so that the virus node is located at the 'bottom'. None of the subnetwork extraction methods predict host-virus interactions, but some of them do add a new node to the background network in order to orient the extracted subnetwork (Yosef *et al.*, 2009; Tuncbag *et al.*, 2013; Yeger-Lotem *et al.*, 2009). This node is connected to a subset of selected genes that are expected to be at either the 'top' or 'bottom' of the extracted subnetwork. For example, 'top' nodes may be receptor proteins, and 'bottom' nodes may be transcription factors or proteins related to a particular cellular process. The new anchoring nodes are added for two chief reasons: a) they create a loose sense of orientation in the extracted subnetwork, though individual edge directions are not actually inferred except in the case of directed Steiner trees, and b) they allow extracted subnetworks to be fragmented, rather than completely connected. While none of these approaches use the additional node to predict host-virus interactions,

there is a resemblance in that that they use the node to provide a 'top' or 'bottom' to the subnetwork.

Like *candidate gene prioritization* (Chapter 2.4) approaches, our method uses a gene ranking method (diffusion kernel) to prioritize genes for inclusion in the inferred subnetwork. We note that Murali *et al.* (2011) apply a gene prioritization method to predict which genes modulate HIV replication in human cell lines. However, their method does not infer consistent, directed pathways, nor does it predict which host factors directly interact with the virus. Our approach combines a gene prioritization method with a directed, signed, subnetwork inference method.

In contrast to *gene set enrichment analysis* (Section 2.7), our method does not restrict our pool of candidate genes and interactions to predefined gene sets. Additionally, gene set enrichment-based methods are typically better suited when the task is to identify common annotations within a gene set, rather than to predict a set of high-precision additional hits or relevant mechanistic interactions among known hits.

## 3.2 Materials and methods

### 3.2.1 Data

The input to our approach consists of a set of viral phenotypes observed in a loss-of-function experiment and a background network of intracellular interactions. When available, we can also take advantage of confirmed relevant interactions curated from the literature.

#### Experimental observations

We analyze data from experiments screening the yeast genome for genes that modulate the replication of two RNA viruses: Brome Mosaic Virus (BMV) (Gancarz *et al.*, 2011; Kushner *et al.*, 2003) and Flock House Virus (FHV) (Hao *et al.*, 2014). The experiments measure the replication of the virus in a yeast host when the expression of one gene is partially or completely depleted. Yeast mutant strains allow the majority of cell genes (of about 5,800 total genes in yeast) to be screened in parallel. For nonessential genes, the experiment was performed using the yeast deletion library (Winzeler *et al.*, 1999). Essential genes were screened using a collection of yeast strains, each with a single essential gene promoter replaced by a doxycycline-repressible promoter, allowing repression of gene expression by adding doxycycline to the growth medium (Mnaimneh *et al.*, 2004). Each data set includes at least two replicate assays for each mutant strain.

As yeast is not the natural host for either virus, an artificial experimental system was used to initiate viral replication. Each mutant yeast strain was grown and transformed with two DNA plasmids expressing viral components. The plasmid expressing viral RNA also contained a luciferase reporter gene, allowing the accumulation of viral RNA to be measured by the intensity of the light produced from luciferase gene expression. The output of the assay is the *fold-change* in accumulation of viral RNA between each mutant strain and the control. Let $m$ be the virus expression level in the mutant strain, and $c$ be the expression level in the control strain. Fold-change is computed as $-\frac{c}{m}$ if $m < c$, or $\frac{m}{c}$ if $m > c$.

We derive a discrete phenotype label for each assayed gene based on the sign, magnitude, and reproducibility of the fold-change across replicate assays. If a mutant reproducibly yields a decrease in viral replication, the interpretation is that the missing gene product directly or indirectly facilitates virus replication. We label such mutants with a down or weakly-up phenotype, depending on the magnitude of the fold-change. Conversely, the interpretation for a mutant that reproducibly results in an increase in viral replication is that, when expressed, the missing gene product directly or indirectly inhibits the replication of the virus. We label such mutants up or weakly-up. The mutants with high-magnitude phenotypes, **down** and **up**, are considered *hits*. While we include mutants with weak phenotypes in our analysis, we are primarily interested in explaining the hits.

The threshold used to divide the hit and weak phenotypes was determined separately for each screen and, for BMV, is described in greater detail in the original publications. In the BMV data set, different thresholds were used for essential and nonessential genes. To be considered a hit for BMV, a nonessential gene mutant resulted in at least a 2.5-fold change in two replicates and at least an average 3-fold change. A more stringent threshold was used for essential gene mutants, which cause expression knockdown rather than complete knockout. Essential gene hits conferred at least a 6-fold change in BMV expression in two replicates. As for FHV, the data set consists of only nonessential yeast gene mutants. FHV hits conferred at least a 2-fold change in viral replication in two replicates, and additionally passed a secondary validation by northern blot.

We assign a third category of phenotype, no-effect, to genes for which the sign of the fold-change is different across replicates. Finally, genes that were either not screened, or for which the yeast colony did not grow, are labeled **unobserved**. Table 3.1 presents the distribution of phenotypes considered here for the BMV and FHV assays. While all available gene mutants were assayed in the experiments, we limit our analysis to only those genes that are represented in the background network.

**Table 3.1:** Distribution of phenotype labels for genes in the background network. The labels were derived from genome-wide assays of Brome Mosaic Virus and Flock House Virus replication in yeast.

| Phenotype | BMV | FHV |
|---|---|---|
| **up** (hit) | 49 | 48 |
| **weak-up** | 623 | 826 |
| **weak-down** | 1,067 | 668 |
| **down** (hit) | 55 | 7 |
| **no-effect** | 1,074 | 991 |

**Table 3.2:** Types of host factors represented by nodes in the background network.

| Node type | Count |
|---|---|
| Yeast ORFs | 4,167 |
| Protein complexes | 472 |
| Small RNAs | 15 |
| Mitochondrial ORFs | 8 |

**Background network**

The interactions in the subnetworks inferred by our method are drawn from a background network that we have assembled from various publicly available data sets. The entities represent gene products and protein complexes. The interactions describe protein-protein and protein-DNA interactions, post-translational modifications of proteins, protein complex membership, transcriptional regulatory interactions, metabolic pathways, and inferred physical interactions between complexes and proteins. In concordance with our goal of inferring mechanistic subnetworks, nearly all interaction types represent direct physical interactions. The exception is the metabolic pathway interactions, which are edges between enzymes that catalyze adjacent metabolic reactions.

High-confidence interactions were selected from each database using stringent filters; for example, protein-protein interactions were selected from BioGRID (Stark *et al.*, 2006) only if the interaction was observed using at least two different types of experimental methods. In total, the background network consists of 4,667 entities and 14,447 interactions.

**Table 3.3:** Intracellular interactions in the background network.

| Interaction | Source | Directed | Signed | Count |
|---|---|---|---|---|
| Protein-protein | Stark *et al.* (2006) | N | N | 4,132 |
| Inferred complex-complex interactions | Heavner *et al.* (2012); Pu *et al.* (2009); Stark *et al.* (2006) | N | N | 22 |
| Inferred complex-protein interactions | Heavner *et al.* (2012); Pu *et al.* (2009); Stark *et al.* (2006) | N | N | 1,128 |
| Between metabolic enzymes | Heavner *et al.* (2012) | N | N | 713 |
| | Heavner *et al.* (2012) | Y | N | 440 |
| Post-translational modifications | Stark *et al.* (2006) | Y | N | 514 |
| Protein-DNA, unsigned | MacIsaac *et al.* (2006) | Y | N | 4,067 |
| Protein-DNA, signed | Guelzim *et al.* (2002); Everett *et al.* (2009) | Y | Y | 1,248 |
| Complex membership | Heavner *et al.* (2012); Pu *et al.* (2009) | Y | Y | 2,183 |

Node and edge counts and citations for the intracellular interaction network are described in Tables 3.2 and 3.3.

Since we are focused on inferring the direction and consistency of paths, we do not need to represent all of the distinctions among the various types of interactions in our background network. Instead, we use a simple, general representation. In this representation, both genes and their gene products are represented using the same node; in this text, we identify nodes using the protein name. Each edge may have a direction and a sign. The direction determines which interactor is the source, and which is the target. For example, for a protein-DNA interaction, a transcription factor is the source, and the regulated gene is the target. The sign describes the effect, positive or negative, of the source on the synthesis, stability, or specific activity of the target. A positive sign is called *activation*, whereas a

negative sign is called *inhibition*. Many edges in the background network are not provided with a sign or direction. For example, transcription factor-gene binding interactions and post-transcriptional modifications are directed but unsigned, and most protein-protein interactions are undirected and unsigned.

Most of the interaction data sets we use are already encoded as binary interactions. However, we extract binary edges from two additional data sets that were not originally in that format: metabolic pathway data and protein complex membership data. To extract binary interactions from the metabolic pathway data (Heavner *et al.*, 2012), we draw an edge between enzymes that catalyze adjacent reactions. This edge is directed unless both reactions were annotated as reversible.

We also represent protein complexes in the background network. Pu *et al.* (2009) and Heavner *et al.* (2012) provide manually-curated protein complex information in the form of sets of genes that are each labeled with the name of a protein complex. To represent the protein complexes, we first add a node that represents the complex, and next add activating, directed edges from each constituent gene to the complex node. Protein complex nodes are treated the same as any other node. One implication of our representation is that only the components of a complex that share the same phenotype label will be drawn into predicted relevant paths that involve the complex.

We also infer a set of undirected complex-complex and protein-complex interactions by combining the protein complex membership information (Heavner *et al.*, 2012; Pu *et al.*, 2009) with the protein-protein interactions (Stark *et al.*, 2006). For a pair of complexes with disjoint protein membership, we draw an undirected edge between them if at least 50% of the possible interactions between one protein from each complex are present in the protein-protein interaction data set. Similarly, for a complex and a single protein, we draw an undirected edge between them if at least 50% of proteins in the complex have a protein-protein interaction with the protein.

**Relevant interactions curated from literature**

The mechanisms for some yeast hits for BMV have been studied in detail (Lee & Ahlquist, 2003; Noueiry *et al.*, 2003; Tomita *et al.*, 2003; Beckham *et al.*, 2007; Diaz *et al.*, 2010; Wang *et al.*, 2011). To leverage this information in our approach, we encode domain knowledge from the literature in the same format as our background network. We have encoded 28 binary interactions among 24 host factors and the external virus node. This set includes the addition of three nodes representing protein complexes, and four interactions between a host component and the virus. Only four of the intracellular interactions were present

**Table 3.4:** Domain knowledge about interactions between yeast and BMV encoded as binary interactions.

| Cellular process | Source | Nodes | Edges |
|---|---|---|---|
| Ubiquitin-proteasome pathway and lipid production | Lee & Ahlquist (2003); Wang *et al.* (2011) | 8 | 11 |
| Membrane conformational stability | Diaz *et al.* (2010) | 4 | 4 |
| Translation | Noueiry *et al.* (2003); Beckham *et al.* (2007) | 12 | 12 |
| Chaperone proteins | Tomita *et al.* (2003) | 1 | 1 |

in the original background network. Table 3.4 summarizes the interactions derived from literature. Visualizations are available at the supplementary website.

### 3.2.2 Computational Methods

We have developed an integer-linear-programming-based approach to infer a directed subnetwork of interactions that are relevant to virus replication in a host cell. The approach infers subnetworks that have the following properties:

- The subnetwork maximizes the nodes included, subject to constraints.
- A small number of interfaces are predicted; these interfaces are the most downstream nodes in the subnetwork.
- The subnetwork accounts for each hit by providing at least one directed path from the hit to an interface.
- Each relevant edge is assigned a single direction.
- The sign of each relevant edge in the subnetwork is consistent with the phenotypes of its interacting host factors.
- The subnetwork is acyclic.

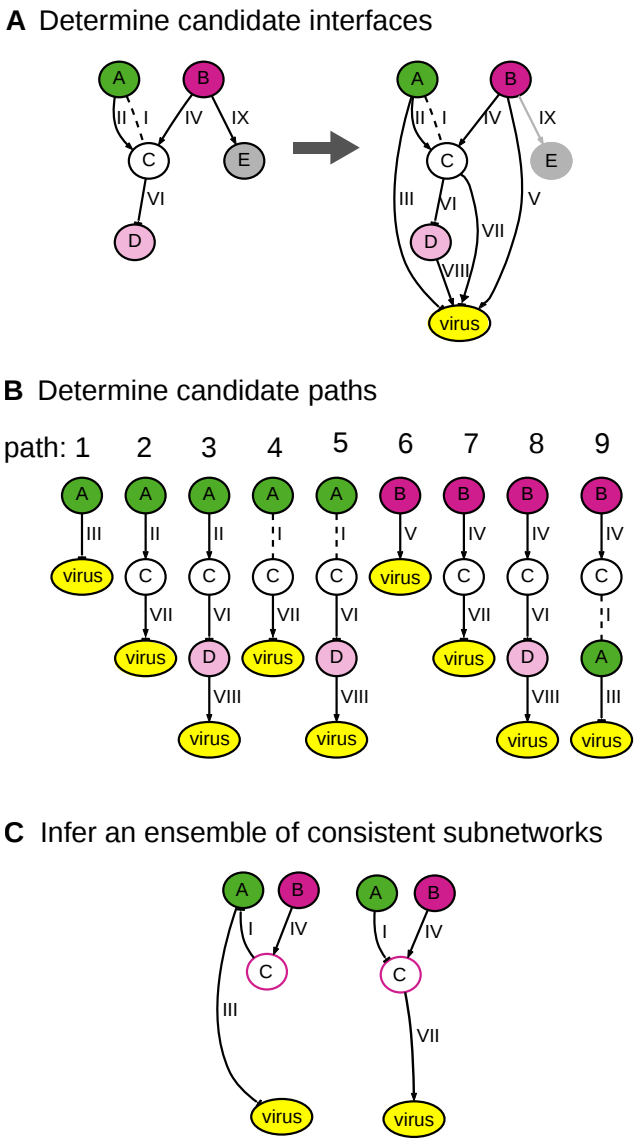



**Figure 3.2:** The steps of our subnetwork inference approach. Each edge is shown with a numeric identifier for cross-reference.
**A** Add a new node to the background network, representing the virus. Add connections between all nodes except **no-effects** to the new virus node, representing the possibility of any host factor having a direct interaction with a viral component.
**B** For each hit identified by the genome-wide suppression assay, enumerate candidate paths through the background network that could explain it by providing a linear path to the virus node.
**C** Infer an ensemble of consistent subnetworks. Each subnetwork is a union of paths that accounts for all of the hits and is consistent with virus phenotype data.

**Overview of approach**

In this section we present an overview of the three steps of our approach. Figure 3.2 illustrates each step applied to a small example background network and set of phenotype labels.

**Step 1: Determine candidate interfaces.** One aspect of the inference procedure is to predict which host factors most directly interact with a viral component. We refer to these as *interfaces*. Before running inference, we identify as *candidate interfaces* all of the host factors that do not have a **no-effect** phenotype label. To represent the possibility of any of these host components interacting with a viral component, we add a special "virus" node to the background network, and add a directed edge from each candidate interface to the virus node. We refer to the edges between host factors and the virus node as *external* edges, and edges between pairs of host factors as *internal* edges. Figure 3.2A depicts the addition of the external edges to the five-node background network shown. No edge is added for node E, which has a **no-effect** phenotype. The set of external edges could be constrained if additional knowledge were available (*e.g.*, experimental evidence for specific interactions between host and viral proteins).

**Step 2: Determine candidate paths.** An inferred subnetwork must account for each hit's viral phenotype by either predicting the hit gene to be an interface itself, or by providing a directed, acyclic path to a predicted interface. We enumerate all possible candidate paths of a specified depth leading from each hit to the virus through a candidate interface (as defined in Step 1). Nodes with a **no-effect** phenotype are not included in candidate paths. Nodes with a weak viral phenotype may appear in paths, but are not used as starting points. Figure 3.2B shows the nine candidate paths for the given network.

**Step 3: Infer an ensemble of consistent, directed subnetworks.** An inferred subnetwork comprises a union of directed candidate paths that predicts which host factors are interfaces and provides consistent and directed paths for each hit. We refer to a candidate path that has been chosen to be part of the inferred subnetwork as a *relevant* path. Similarly, we refer to an edge (node) in a relevant path as a *relevant edge (node)*. If an external edge (edge between a host factor and the virus) is predicted to be relevant, the host factor is predicted to be an interface. Inference is a matter of determining the optimal combinations of relevant paths, node phenotypes, interfaces, and edge signs and directions, and is carried out by the IP method.

During inference, the method infers binary viral phenotype labels for all unassayed relevant nodes, and the signs of all relevant edges in cases where they are not specified in the given data. (We do not infer these attributes for nodes and edges that are deemed irrelevant.) While the input data differentiates between weak and strong (hit) viral phenotypes, we

predict only the labels **up** or **down** for the unassayed genes that we infer to be relevant. For an edge to be considered relevant, its sign must be consistent with the phenotypes of the interacting nodes. We refer to Figure 3.2A to illustrate this notion of consistency. In the background network, notice that we have evidence that both nodes A and B can activate node C. If edge II is relevant, node C would have the phenotype **up**, to match A's phenotype. However, if edge IV is relevant, node C would have the phenotype **down**, to match B. Since a relevant node can have only one phenotype label, we cannot predict that both edges II and IV are relevant. In addition to using the consistency concept to rule out inconsistently signed edges, we can also use it to infer missing edge signs. If both edges IV and I are relevant, then the inferred phenotype for node C is **down**, and we infer that edge I's sign is inhibition.

The inference process also assigns a direction to all relevant, undirected edges. In the inferred subnetwork, each hit must be able to reach an interface by a directed path. Since a relevant edge can only take one direction, paths 4 and 9 in the example cannot both be predicted to be relevant because they require opposite directions for edge I.

Because of the incompleteness of the background network and experimental data, the space of possible subnetworks that meet all of our requirements is very large. To represent this space, we find an ensemble of subnetworks, where each one corresponds to a different optimal solution to the IP. We initially solve the IP to optimality using a branch-and-cut method (Danna *et al.*, 2007), and collect multiple solutions by returning to untaken branches. With the ensemble of subnetworks, we thereby assess the confidence in the relevance of a path (node, edge) as the fraction of subnetworks in the ensemble containing that path (node, edge). We measure confidence in the same way for the other inferred quantities: phenotypes, edge signs, and edge directions. Figure 3.2C shows an ensemble of two inferred subnetworks that each account for both hits A and B using one interface.

**Integer program (IP) variables and notation**

Subnetwork inference is performed by solving an integer program (IP), which consists of a set of linear constraints and an objective function, all of which are defined over a set of integer variables that characterize possible subnetworks. The values of some of the variables are determined by the input to the inference process (the phenotypes and background network), whereas others are inferred by the IP. In our implementation, some variables need not be explicitly declared as integer variables because they are constrained such that they can only feasibly take integer values. The implemented program is therefore more precisely a mixed integer linear program.

**Table 3.5:** Integer program variables. Binary variables represent the status of nodes, edges, and paths in the network.

| Network elements | Variable | Interpretation | Values |
|---|---|---|---|
| Paths $p$ | $\sigma_p$ | Relevant | no=0, yes=1 |
| Edges $e$ | $x_e$ | Relevant | no=0, yes=1 |
| | $a_e$ | Relevant, activating | no=0, yes=1 |
| | $h_e$ | Relevant, inhibiting | no=0, yes=1 |
| | $d_e$ | Direction | back=0, forward=1 |
| Nodes $n$ | $y_n$ | Relevant | no=0, yes=1 |
| | $v_n$ | Phenotype | down=0, up=1 |

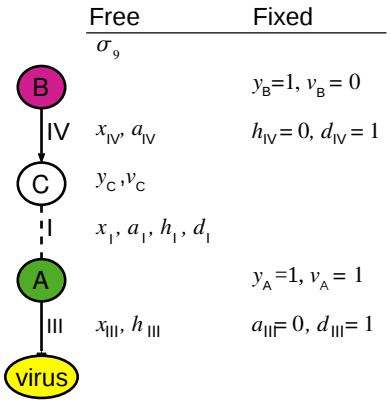



**Figure 3.3:** Variables for pathway 9 from Figure 3.2. The values of some variables are fixed by the data. The values of free variables are determined by the IP.

First, we describe the variables and notation that we use to define the IP. The background network is represented as a graph of nodes $\mathcal{N}$, edges $\mathcal{E}$, and candidate paths $\mathcal{P}$. $\mathcal{E}(p)$ and $\mathcal{N}(p)$ refer to the edges and nodes in a particular candidate path $p$, $\mathcal{N}(e)$ refers to the nodes in a particular edge $e$, and $\mathcal{E}(n)$ refers to the edges that touch a particular node $n$. We denote an edge between nodes $n_i$ and $n_j$ as $(n_i, n_j)$.

These sets are further divided into subsets based on experimental data. $\mathcal{N}^H \subseteq \mathcal{N}$ is the set of hit nodes. $\mathcal{E}^U \subseteq \mathcal{E}$ is the set of undirected edges. The complete set of edges $\mathcal{E}$ can also be divided into external edges $\mathcal{E}^X$, which are added during the execution of our method to provide connections to the virus node, and internal edges $\mathcal{E}^I$, which represent the original background network.

Each node $n$ has two variables: $y_n$, representing whether or not the node is present in any relevant paths, and $v_n$, representing its observed or inferred phenotype sign. For hits, we fix $y_n = 1$ to require that they are present in the inferred subnework. For **down** and **weak-down** genes, we fix $v_n = 0$; for **up** and **weak-up** genes, we fix $v_n = 1$. As many as four variables describe each edge. The predicted relevance of an edge $e$ is represented with the variable $x_e$, which takes the value 1 if the edge is in at least one relevant path. The sign of an edge is represented by two mutually exclusive variables $a_e$ and $h_e$. If $a_e = 1$ ($h_e = 1$), the edge is predicted to be relevant, and inferred to describe an activating (inhibitory) interaction. If an edge is not predicted to be relevant, $x_e = a_e = h_e = 0$. For activating edges given in the background network, $h_e$ is fixed at 0; similarly, for inhibitory edges, $a_e$ is fixed at 0. Undirected edges also have an associated variable $d_e$, representing the inferred direction of the edge (relative to an arbitrarily chosen canonical direction). If the inferred direction is the same as the canonical direction of the edge, or "forward", then $d_e = 1$; otherwise, $d_e = 0$. The predicted relevance of a path $p$ is represented with the variable $\sigma_p$, which takes the value 1 if the path is included in the inferred subnetwork, and 0 if it is not.

The variables are summarized in Table 3.5. Figure 3.3 shows the variables used to characterize one specific example path.

**Diffusion kernel (DK) for node prioritization**

To represent the ways in which a hit may modulate the virus through many paths, our inferred subnetworks will generously include consistent nodes and edges. Inspired by the use of graph diffusion kernels to prioritize candidate genes, we use a diffusion kernel method to prioritize non-hit nodes (those with unobserved or weak phenotypes) for inclusion in the subnetwork. (All hits are already required to be included.) The intuition behind this method is that each hit carries some amount of weight that is partially diffused out via its neighbors in the background network. Each node in the network thereby receives a weight according to its proximity and connectivity to the set of hits. This score is used in the objective function of our integer program method.

To calculate the DK scores, we first calculate a regularized Laplacian kernel matrix $K$ (Smola & Kondor, 2003), in which the value in each cell represents the proximity and connectivity between two nodes in the graph. The first step is to use the background network to calculate an $|\mathcal{N}| \times |\mathcal{N}|$ symmetric adjacency matrix $A$. In this matrix, $A_{ij} = 1$ if there is an edge (regardless of direction) between nodes $n_i$ and $n_j$ in the background network (internal edges only), and 0 otherwise. Second, we calculate $D$, a diagonal degree matrix derived from $A$, where $D_{ii} = \sum_j^{|\mathcal{N}|} A_{ij}$. From these, we calculate a normalized Laplacian matrix $L = 1 - D^{-1/2} A D^{-1/2}$. Finally, the kernel matrix is $K(\lambda) = [I + \lambda L]^{-1}$.

Next, we use the kernel matrix to calculate how close and connected each node is to the set of hits. We define $q$ as a binary vector of length $|\mathcal{N}|$ where $q_i = 1$ if $n_i \in \mathcal{N}^H$ (is a hit) and $q_i = 0$ otherwise. Finally, for each node $n_i$, the DK score score$(n_i)$ is calculated as $\sum_{j=1}^{|\mathcal{N}|} K_{ij}(\lambda)\ q_j$.

**Global objective function and constraints in the IP**

The following objective function and two constraints control global properties of the inferred subnetwork.

*Maximize the inclusion of nodes that are proximal and connected to hits.* In order to capture multiple pathways between the hits and the virus, we want to include in the inferred subnetwork the nodes that are most proximal and connected to the hits. Which nodes can be included is limited by the IP's constraints, and so we prioritize nodes using their diffusion kernel score. The objective function of our integer program maximizes the combined score of relevant nodes that are not hits ($\mathcal{N} - \mathcal{N}^H$).

$$\max \left( \sum_{n \in \mathcal{N} - \mathcal{N}^H} \text{score}(n)\ y_n \right)$$

*A small number of interfaces are inferred.* The true number of interfaces is unknown. As a heuristic, we limit the number of interfaces in the inferred subnetwork to a specified integer $\gamma$. In the inferred subnetwork, we can count the number of interfaces by counting the number of relevant external edges $\mathcal{E}^X$, which connect yeast gene nodes to the virus node.

$$\left( \sum_{e \in \mathcal{E}^X} x_e \right) \leq \gamma$$

While the objective function tends to maximize the number of nodes in the inferred subnetwork, we can control the size of the subnetwork by restricting the number of interfaces. Depending on the prediction task that the inferred subnetwork will be used for, we may use a more constrained or more generous number of interfaces. If constrained to use only a small number of interfaces, the inference process will identify those interfaces that can explain the most hits. This setting would be appropriate to use when the goal is to predict a high-confidence set of interfaces. On the other hand, allowing more interfaces expands the network and allows for more parallel paths and alternative explanations for hits.

*The proportion of edge signs is constrained.* In the BMV and FHV screens, a hit's phenotype sign (**up** or **down**) is highly correlated with those of its neighbors in the background

network. Therefore, we require that the proportion of activating edges in the inferred network is close to a proportion estimated from data. Considering all pairs of hits that interact (under any interaction) in the background network, we record the proportion of pairs with the same phenotype sign. For the BMV data set, this is about 95%; for FHV, it is 100%. The following constraint gives a lower bound $\alpha$ on the proportion of activating internal edges (edges that do not involve the virus node).

$$\sum_{e \in \mathcal{E}^I} a_e \geq \alpha \sum_{e \in \mathcal{E}^I} x_e$$

By default, we set this $\alpha$ =0.9 to allow a small deviation from the proportion estimated from the data set. If we did not allow a small number of inhibitory edges into the inferred subnetwork, our inferred subnetworks would not be able to represent connections between two differently-signed hits and the same downstream interface.

**Local constraints in the IP**

Our other subnetwork desiderata are represented as constraints that are used to select which edges and paths are deemed relevant.

*Every hit is included in the inferred subnetwork.* By fixing $y_n = 1$ for each hit node $n \in \mathcal{N}^H$, we force the solver to infer a relevant path to account for the hit.

$$\forall n \in \mathcal{N}^H \qquad\qquad y_n = 1$$

*All edges in a relevant path are relevant.* A relevant edge $e$ (where $x_e = 1$) must be in at least one relevant path (for which $\sigma_p = 1$); we refer to all paths $p$ for an edge $e$ as $\mathcal{P}(e)$. For a relevant path $p$, all of its edges $\mathcal{E}(p)$ must be relevant.

$$\forall e \in \mathcal{E} \qquad\qquad x_e \leq \sum_{p \in \mathcal{P}(e)} \sigma_p$$

$$\forall p \in \mathcal{P}, e \in \mathcal{E}(p) \qquad\qquad \sigma_p \leq x_e$$

*All nodes in a relevant edge are relevant.* A node $n$ is relevant (that is, $y_n = 1$) if one if its edges $e \in \mathcal{E}(n)$ is relevant ($x_e = 1$). For a relevant edge $e$, both of its nodes $\mathcal{N}(e)$ are also relevant.

$$\forall n \in \mathcal{N} \qquad\qquad y_n \leq \sum_{e \in \mathcal{E}(n)} x_e$$

$$\forall e \in \mathcal{E}, n \in \mathcal{N}(e) \qquad\qquad x_e \leq y_n$$

*All relevant edges must be either activating or inhibitory.* The following constraints require that at most one sign variable ($a_e$ or $h_e$) can be equal to 1 for any edge, and that for a relevant edge $e$ (where $x_e = 1$), exactly one sign variable must be equal to 1.

$$\forall e \in \mathcal{E} \qquad a_e + h_e \leq 1$$
$$a_e + h_e = x_e$$

*The sign of a relevant edge is consistent with the phenotypes of the nodes that it connects.* The following set of constraints guide the inference of phenotypes and edge signs for relevant nodes and edges. If a relevant internal edge $e = (n_i, n_j)$ represents activation ($a_e = 1$), the interacting nodes must have the same phenotype ($v_{n_i} = v_{n_j}$).

$$\forall e = (n_i, n_j) \in \mathcal{E}^I \qquad v_{n_i} + a_e \leq 1 + v_{n_j}$$
$$v_{n_j} + a_e \leq 1 + v_{n_i}$$

If a relevant internal edge $e = (n_i, n_j)$ represents inhibition ($h_e = 1$), the two interacting nodes must have opposite phenotypes ($v_{n_i} \neq v_{n_j}$).

$$\forall e = (n_i, n_j) \in \mathcal{E}^I \qquad h_e + v_{n_i} + v_{n_j} \leq 2$$
$$v_{n_i} + v_{n_j} \geq h_e$$

*In a relevant path, all edges are directed toward the interface.* In each relevant path $p$, the edges $\mathcal{E}(p)$ must be oriented toward the virus node at the end of the path. This direction is determined when the candidate path is generated in Step 2, and is given by $dir(p, e)$. The term including $I(\cdot)$, the indicator function, returns 1 if an edge's inferred direction corresponds to the direction required by the path.

$$\forall p \in \mathcal{P}, e \in \mathcal{E}(p) \qquad \sigma_p \leq I\left(d_e = dir(p, e)\right)$$

*The inferred subnetwork is acyclic.* While each candidate path is acyclic, it is possible to choose a union of paths that contains cycles. We argue that acyclic inferred subnetworks are more interpretable because they better describe the order of genes in paths and therefore differentiate upstream-acting factors from downstream-acting interfaces. Because searching for and prohibiting all cycles in the inferred subnetwork is an intractable task, we use an approximation that prohibits small cycles. First, we identify sets of edges that induce cycles of a restricted size, and then introduce constraints to the IP so that each possible cycle among relevant nodes must be broken.

We identify cycles among candidate nodes by performing depth-limited, depth-first searches through the candidate nodes in the background network, once per candidate node. If a node is encountered for the second time during the search, then the edges that were taken to get there are saved as a cycle. In our experiments, we search for cycles containing up to three edges. The following constraints require each cycle to be broken. A cycle is broken if either at least one edge in it is inferred to be irrelevant, or if at least one edge is inferred to be directed in the opposite direction of the other edges in the cycle. The precomputed set of possible cycles is $\mathcal{C}$, where each cycle $c$ has the set of edges $\mathcal{E}(c)$. In the second term, the direction of an undirected edge $e$ that would complete the cycle $c$ is given by $dir(c, e)$.

$$\forall c \in \mathcal{C} \qquad \left( \sum_{e \in \mathcal{E}(c)} 1 - x_e + \sum_{e \in \mathcal{E}(c) \cap \mathcal{E}^U} 1 - I(d_e = dir(c, e)) \right) \geq 1$$

*Interfaces are the most downstream nodes in the subnetwork.* Interfaces are meant to represent the host factors and processes that are closest to a direct interaction with the virus. So, we prohibit a predicted interface $n$ from being inferred to have any other downstream neighbors. Given a particular node $n$, the IP must choose between connecting $n$ to the virus (thus making it an interface), or inferring any other relevant outgoing edge from $n$. In the following constraints, we refer to the external edge (from $n$ to the virus) using $e_i$, and the internal edges using $e_j$. We use separate constraints for directed and undirected internal edges. In the constraint for undirected edges (listed second), the function $source(e, d_{e_j})$ returns the node $n$ that is the source of the undirected edge $e_j$ when the edge's direction is set to $d_{e_j}$.

For directed edges:

$$\forall e_i = (n, virus) \in \mathcal{E}^X, e_j = (n, n_k) \in \mathcal{E}^I - \mathcal{E}^U \qquad x_{e_i} + x_{e_j} \leq 1$$

For undirected edges:

$$\forall e_i = (n, virus) \in \mathcal{E}^X, e_j = (n, n_k) \in \mathcal{E}^I \cap \mathcal{E}^U \quad x_{e_i} + x_{e_j} + I\left(source(e_j, d_{e_j}) = n\right) \leq 2$$

*Specific nodes and edges whose relevance is supported by domain knowledge must be included in the inferred subnetwork.* When we have domain knowledge that a specific host factor or interaction is relevant to viral replication, we can use it to seed the inferred subnetwork by setting its relevance variable to 1 ($y_n = 1$ for nodes, $x_e = 1$ for edges). In the following

constraints, $\mathcal{N}^L \subset \mathcal{N}$ and $\mathcal{E}^L \subset \mathcal{E}$ represent the relevant nodes and edges drawn from the literature.

$$\forall e \in \mathcal{E}^L \qquad x_e = 1$$
$$\forall n \in \mathcal{N}^L \qquad y_n = 1$$

## 3.3 Results

Although it is not practicable to fully evaluate our inferred subnetworks, we can assess their validity using a number of quantitative and literature-based evaluations.

### 3.3.1 Cross-validated phenotype prediction

We first describe an experiment in which we assess the accuracy of our approach in predicting whether test genes with held-aside phenotypes are hits or not. We refer to this as the *hit-prediction* task. Previously, diffusion kernel methods like the one we use in our objective function have been successfully applied to this task, which is also called gene prioritization (Vanunu *et al.*, 2010; Murali *et al.*, 2011).

Using a leave-one-out methodology, we hold aside the measured phenotype for one gene at a time. The set of genes that are held-aside as test cases for the BMV data set includes 104 hits (49 **up** and 55 **down**) and 1074 **no-effect** genes. The test set for the FHV data set comprises 55 hits (48 **up** and 7 **down**) and 991 **no-effect** genes. We do not test weak-phenotype genes in this evaluation. When a given gene is held aside, it is treated as if its phenotype has the **unobserved** label, meaning that the inference process is used to predict whether or not the gene is relevant, and, if it is predicted to be relevant, its phenotype label. If the test case is included in the set of literature-curated interactions, then all interactions that involve the test case are held aside as well. We also recalculate the diffusion kernel scores for the entire network for each held-aside test case.

To predict the label of a held-aside node, we use our integer programming approach to infer an ensemble of subnetworks. An individual subnetwork may include the held-aside gene and provide a predicted **up** or **down** phenotype for it, or it may exclude the gene. We assess our confidence in whether the gene is a hit or not by determining the fraction of subnetworks in which it is predicted to have an **up** or **down** phenotype. When this fraction is the same for a set of cases, the node scores computed by the kernel are used as

a secondary measure of confidence. By varying a threshold on these confidence values, we can plot a precision-recall curve characterizing the predictive accuracy of our method. *Recall* is defined as the fraction of true hits in the test set that are predicted to be hits, and *precision* is defined as the fraction of predicted hits that are truly hits. In this context, we consider precision to be the more important of the two measures, as it is better to to avoid devoting follow-up experiments to false positives.

**Parameter settings**

For all experiments, candidate pathways are limited to a depth of three interactions, and 100 subnetworks are inferred for each ensemble. For the cycle-prohibiting constraint, we compute and disallow cycles of up to three edges. The default setting for $\alpha$, the fraction of inferred activating edges, is 0.9. We initially set $\gamma$, the maximum number of interfaces, to the minimum feasible number that can be used to consistently explain all hits. This is determined for each data set by running a slightly modified version of our IP in which the objective is to minimize the number of interfaces. We perform experiments assessing the effect of raising the level of $\gamma$ at four additional intervals of 25. For BMV, we perform experiments using $\gamma = [47, 72, 97, 122, 147]$; for FHV, we use $\gamma = [26, 51, 76, 101, 126]$. We also assess the effect of the other settings: prohibiting cycles, requiring edges and nodes supported by domain knowledge, and controlling the distribution of edge signs using the parameter $\alpha$. All experiments were performed using GAMS 23.9.3 (for constructing the IP) (GAMS Development Corporation, 2010) and the IBM ILOG CPLEX 12.4.0.1 (for solving the IP) (IBM, 2012). Both are commercial products that are currently available with reduced-cost or free licenses for academic use.

**Baselines for comparison**

We compare our method's precision-recall curve to the curve generated by the diffusion kernel (DK) scores. We also compare against two baselines that use local phenotype information: a hypergeometric test baseline and a nearest neighbor-baseline. In what we call the hypergeometric test baseline, we use the hypergeometric distribution to assign a $p$-value to each held-aside test case gene based on the proportion of hits among its first neighbors relative to the proportion of hits in the entire background network.

To acquire a ranking of the test cases, we sort them in ascending order of their hypergeometric $p$-value. Our second baseline is a naïve nearest-neighbor approach that uses information about both hit and weak viral phenotypes. For each query gene, we count the

number of adjacent genes that have either a hit or weak viral phenotype, and rank the test cases in descending order by this count.

We also perform a permutation test in order to estimate our method's ability to predict real viral phenotype hits using randomized input data. The purpose of this test is to estimate how much of our method's predictive accuracy is due to the topological properties (*e.g.*, degree, connectivity) of the held-aside genes in the background network, independent of true experimental data. For this test, we infer a subnetwork ensemble for each of 1,000 permuted sets of phenotype labels, and rank actual test cases by their average confidence over the 1,000 inferred subnetwork ensembles. We construct permuted phenotype label sets with approximately the same degree distribution as the original experimental phenotype labels, to control for the effect of degree on the likelihood that a node is predicted to be relevant. To maintain the degree distribution, we draw for each phenotype label a gene from the background network that has the same degree. If fewer than ten genes have the same degree, we expand our consideration to the genes with degree one higher or lower, and continue expanding until we have at least ten to draw from. Among the permuted phenotype label sets for BMV, on average 3.54 true hits (out of 104 in the background network) are retained as permuted hits; for FHV, on average 1.2 true hits (out of 55) are retained.

**Hit-prediction results**

Precision-recall curves for the hit-prediction task are presented in Figure 3.4. The horizontal line shown in each panel is the fraction of the test set that are hits, thus representing the level of precision that would be achieved by simply predicting that all held-aside genes are hits.

Figure 3.4A compares the diffusion kernel method to the two baselines that employ local phenotype information. For both the BMV and the FHV data sets, the nearest neighbor baseline performs quite poorly: this indicates that, locally, the weak phenotype information is not helpful for making predictions about a node's viral phenotype. The hypergeometric test baseline generally does not perform as well as the diffusion kernel on either data set, although its most highly confident predictions for BMV are more accurate. These results indicate that only a small number of hits can be predicted based only on their local neighborhood, and thus support the use of the diffusion kernel to help identify unassayed genes that might be involved in viral replication. The recall of both of these baselines is bounded by the number of hits that have other hits (or weak-phenotyped genes) among their neighbors.

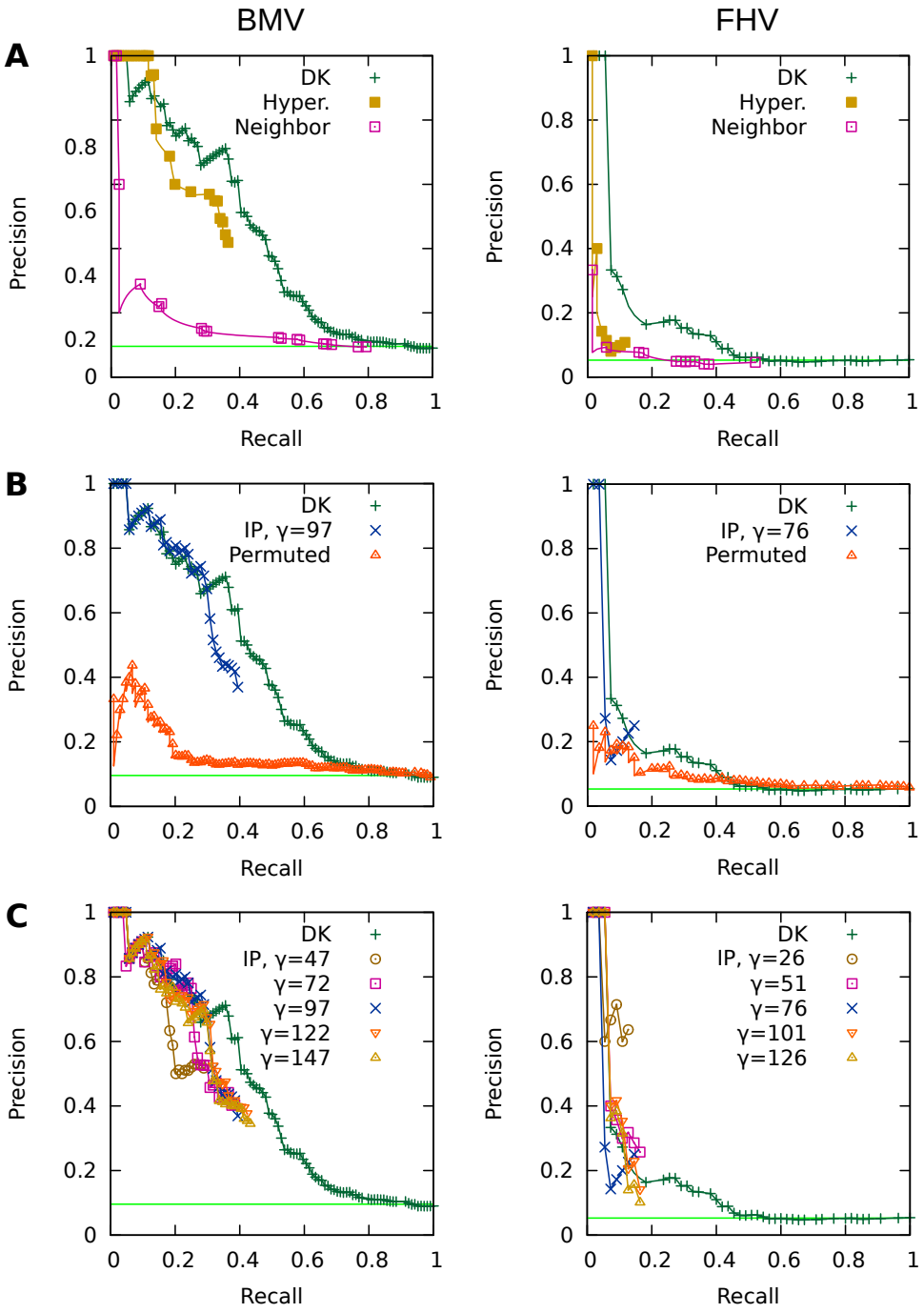



**Figure 3.4:** Precision-recall curves for the hit-prediction task. BMV at left, FHV at right. The horizontal line shows precision that would be achieved if all test cases were called hits.
**A** Comparison of the diffusion kernel method to the naïve baselines.
**B** Comparison of our IP approach to the diffusion kernel method and to random permutations.
**C** The effect of varying $\gamma$, the maximum number of interfaces allowed in the subnetwork inferred by the IP method.

Figure 3.4B compares our IP method, which uses the diffusion kernel, to the diffusion kernel alone and to the permutation-test baseline. We show the results achieved using the median tested number of interfaces ($\gamma = 97$ for BMV, $\gamma = 76$ for FHV). (We choose to show the $\gamma$ value from the middle of the tested range because, as we discuss later, the method's accuracy does not appear to be very sensitive to the number of allowed interfaces.) In the high-confidence range, our method is able to achieve comparable precision to the kernel method alone, despite the fact that it is making more detailed predictions by specifying interfaces and at least one directed path from each hit to an interface. Both our method and the diffusion kernel method easily surpass the permutation-test baseline's precision. Interestingly, the permutation test's precision is higher than the random guessing line in the low-recall region, suggesting that some hits are more central in the background network compared to **no-effect** genes.

We note that our method does not achieve the same level of recall as the diffusion kernel method. Whereas the diffusion kernel can reach high levels of recall because it propagates nonzero scores to all held-aside genes that are indirectly connected to a hit, the recall of our approach is limited by whether each held-aside gene is included in an inferred subnetwork or not. Our IP can only include a held-aside hit that (i) is used in at least one candidate path for another hit, and (ii) is useful for connecting hits to inferred interfaces. To some extent, we can increase recall by allowing more interfaces in the subnetworks, and by enlarging the number of subnetworks generated in the ensembles. Nevertheless, given the low precision of the diffusion-kernel predictions at high levels of recall, we argue that the recall differences between the two approaches are not of practical significance.

To assess the robustness of our IP with respect to the number of interfaces allowed, we vary $\gamma$ (the maximum number of interfaces) over five values that range from the minimum feasible number to one hundred more. Figure 3.4C presents precision-recall curves for this experiment. For the BMV data set, requiring the minimum number of interfaces results in ensembles that are the least accurate, but the other four values tested produce similar precision to each other, with recall increasing just slightly with $\gamma$. For the FHV data set, the minimum number of interfaces results in higher precision overall in comparison to higher values of $\gamma$, but lower precision in the highest-confidence range. Since the FHV curve represents only a small number of predictions, it is difficult to make strong conclusions based on it. However, the results of the experiment on both data sets suggest that, beyond the minimum allowed, the number of interfaces does not have a large effect on accuracy. For BMV, it appears to be best to use a moderate number of interfaces.

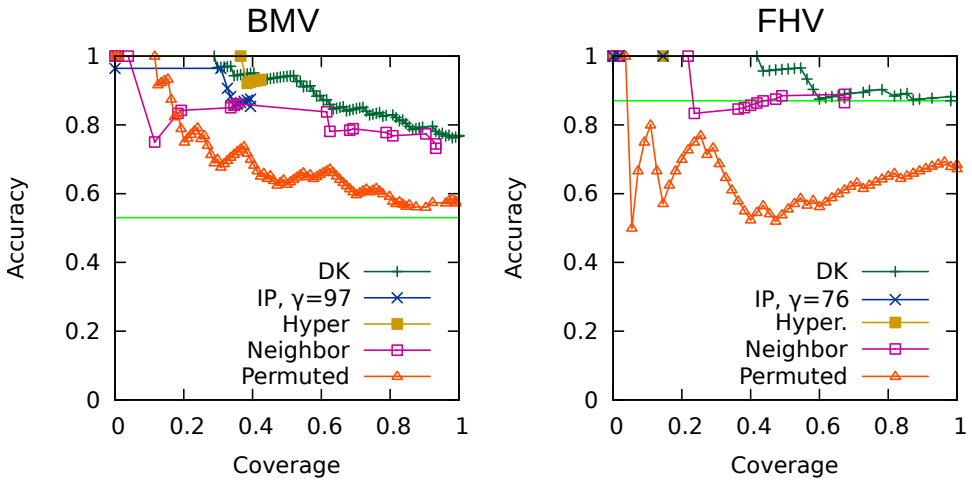



**Figure 3.5:** Accuracy-coverage curves for the sign-prediction task. BMV on the left, FHV on the right. The horizontal line indicates the accuracy that would be achieved by assigning the plurality phenotype label to every test case (**down** for BMV, **up** for FHV.)

### Sign-prediction task

As a secondary evaluation, we assess the accuracy of the methods in predicting the correct *sign* of the phenotype (**up, down**) for held-aside hits. We refer to this as the *sign-prediction* task. The methodology for this experiment is largely the same as for the previous one. We hold aside a given hit's phenotype (treating the gene as being **unobserved**), infer an ensemble of 100 subnetworks, and then predict the phenotype sign that is inferred by a plurality of subnetworks. The confidence in a predicted sign is given by the fraction of subnetworks in which the gene is predicted to take that sign. We compare the predictive accuracy of our approach to the diffusion kernel and the baselines considered in the previous experiment. We also tested a variant of the neighbor-voting baseline that employs the notion of consistency described in Section 3.2.2. That is, neighbors connected to the held-aside gene by unsigned and activating edges vote with their own phenotype, but neighbors connected by inhibiting edges vote with the phenotype of opposite sign. The consistency-based baseline performed no better than the simple neighbor-voting methods and thus we report the results only for the original baseline here.

We construct accuracy-coverage plots for our IP-based approach and both baselines. *Accuracy* is measured as the fraction of phenotype signs correctly predicted, and *coverage* is the fraction of hits (with either **up** and **down** phenotype) for which predictions are made. The hits are sorted by the algorithm's confidence in the predicted phenotype, and accuracy is plotted as coverage increases. The results of this experiment are presented in Figure 3.5.

For both data sets, the diffusion kernel method is the only one able to make predictions for the entire set of hits, and it achieves high accuracy. Our IP approach matches the diffusion kernel method in the high-confidence range for both data sets. The predictive accuracy of the hypergeometric test is comparable to the IP approach for both data sets. The neighbor-voting baseline is slightly better than our IP method for the FHV data, but inferior for BMV.

We also performed a number of other experiments to vary the method's parameters and compare to objective functions and constraints from related subnetwork inference approaches. These are explained in Section 3.3.7.

### 3.3.2 Phenotype prediction for unobserved host factors

One motivation for our inference approach is to make predictions about which unassayed host factors may be involved in viral replication. A number of host factors were unable to be assayed using the deletion or doxycycline-repressible mutant libraries, either because the mutant was not part of the library or did not grow under experimental conditions. As these factors cannot be assayed using high-throughput screens, there is a need to identify a high-confidence subset of them for further, lower-throughput experimentation. Toward this end, we use our approach to make predictions about host factors that were not assayed (or not successfully assayed) in the genome-wide BMV screens (Kushner *et al.*, 2003; Gancarz *et al.*, 2011). We collect an ensemble of 100 inferred subnetworks using all available phenotype data and allowing the use of 97 interfaces. We choose this number of interfaces because it is in the middle of the range tested in our cross-validation experiments, the results of which suggest that prediction accuracy is not significantly affected by a larger number of interfaces. We also seed the network with the literature-curated edges described in Section 3.2.1.

Out of 1,821 unassayed host factors in the background network, 221 are predicted to be relevant by any of the 100 inferred subnetworks in the ensemble, and 189 receive $\geq$ 0.75 confidence. Of these, 124 represent ORFs (about 9% of unassayed ORFs/putative ORFs), and 65 represent protein complexes (about 14% of represented protein complexes). Here we discuss independent evidence supporting a selection of these predictions.

In numerous cases, the predicted hits include members of pathways of protein complexes known to be involved in BMV replication. In these cases, the inferred subnetworks correctly expanded the relevant complexes with other known components or functional partners that were absent from the given hit sets for technical reasons, such as non-viability of the relevant mutant strain. One example is the inferred inclusion of previously un-

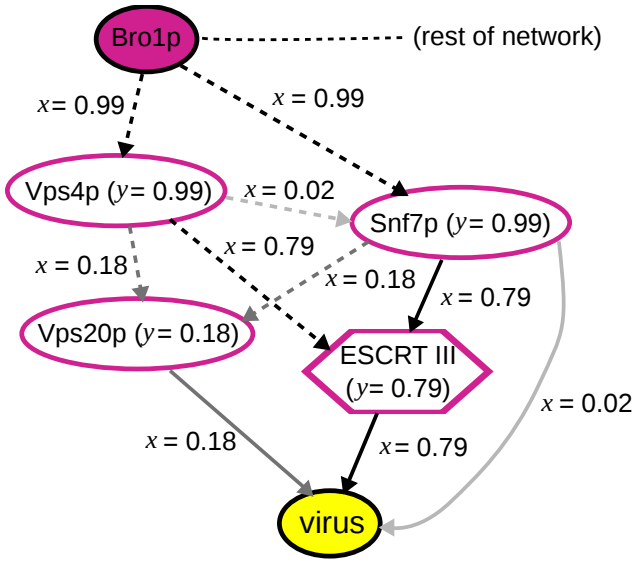



**Figure 3.6:** A component from the inferred subnetwork ensemble showing the predicted involvement of Snf7p and Vps4p in viral replication. For predictions made about node and edge relevance, confidence values <1.0 are indicated. For the unassayed nodes, the same phenotype label prediction was made in all solutions in which they appear; similarly, all solutions predicted the same direction for the undirected edges. Dashed edges indicate cases in which the edge's direction was not fixed in the background network. See Figure 1 for a key to the other network elements.

implicated components of the cellular ubiquitin-proteasome system, such as the 20S proteasome and components of the 19S regulator complex. While some experimental and literature-curated hits are associated with the proteasome, the predicted hits contribute several more proteasomal proteins. Recent additional experiments, including inhibitor studies and other approaches, have confirmed the involvement of the 20S proteasome, the 19S regulator and other factors in this system in multiple aspects of BMV RNA replication (B.G. and P.A., unpublished results).

Even more important biological validation of our results emerged from additional experimental studies. For example, our ensemble predicts the involvement of Snf7p and Vps4p, both at 0.99 confidence. These are proteins in the ESCRT pathway, which is involved in membrane bending and scission events in cell division, cell surface receptor down-regulation and other processes (Hurley & Hanson, 2010). Recent studies initiated independently of the work reported here have confirmed the predicted role of ESCRT pathway, and of Snf7p and Vps4p in particular, in facilitating BMV RNA replication (A. Diaz, X. Wang and P. Ahlquist, manuscript in preparation). The predicted relevant interactions involving Snf7p and Vps4p are shown in Figure 3.6.

A further example is provided by the inferred involvement of Xrn1p, a protein involved in RNA degradation. An independent study confirmed the strong impact of the gene *XRN1* on BMV replication by showing that a BMV mutant defective in modifying BMV RNA's by the addition of a 5′ $m^7G$ cap could not accumulate RNA in wild type yeast but did so in an *xrn1*$\Delta$ deletion strain (Ahola *et al.*, 2000).

### 3.3.3 Host-virus interface predictions

The subnetworks inferred using our method can be used to predict which host factors are closest to a direct interaction with the virus. For this evaluation, we predict a set of high-confidence host interfaces for BMV. The ability of our methods to predict physical interfaces between host protein networks and viral components is constrained by the limits of current background knowledge, as specifically represented by the input background network of interacting host proteins. Because of such external limitations, some predicted interfaces may not represent actual host-virus interfaces, but instead approximate the host component that would most likely connect with an actual interface if the relevant subnetwork were extended to include currently unrecognized interaction partners. We consider support from domain knowledge that the predicted interfaces are plausibly close to a direct interaction with a viral component.

To predict high-confidence interfaces, we infer an ensemble of 100 subnetworks for BMV-yeast interactions, applying the global constraint that only the minimum possible number of interfaces can be used (that is, the smallest number of interfaces such that the IP remains feasible; in this case, 47). We also seed the network with the literature-curated edges described previously, which include four interfaces. Over the entire ensemble, the total number of interfaces used by at least one subnetwork is 51. We designate as "high-confidence" those interfaces that (i) account for more than one hit (other than themselves), (ii) have greater than 0.75 confidence, and, (iii) are predicted to be an interface with an average of at least 0.75 confidence across all of the leave-one-out ensembles inferred using a minimum number of interfaces.

Our method predicts 14 novel high-confidence yeast interfaces for BMV, as shown in Table 3.6. We assessed these high-confidence interfaces for plausibility based on their annotated function in the Saccharomyces Genome Database (Cherry *et al.*, 2012).

The value of our subnetwork inference method is supported by the observation that several of the predicted 14 high-confidence interfaces are known interactors with BMV components and many more are closely associated with known interactors. Below we discuss available information on several classes of these predicted interfaces.

**Table 3.6:** High-confidence predicted interfaces. Yeast proteins and protein complexes that are predicted to affect BMV through a direct interaction.

| Function or location | Predicted interfaces |
|---|---|
| Membrane | Nem1p/Spo7p holoenzyme, Set3p complex, Tcb3p, UDP-N-acetylglucosamine complex |
| Ribosome | Dbp2p |
| Viral RNA and protein interactions | OCA complex, Ski complex, Smt3p, Ahp1p, 19/22S regulatory complex of proteasome, Cdc34p |
| mRNA transcription | Gcn5p, Sir4p, Tup1p |

**Membrane-associated interfaces**

Multiple predicted interface proteins have functions or localization related to endoplasmic reticulum (ER) membranes, the site of BMV genomic RNA replication (Restrepo-Hartwig & Ahlquist, 1999; Schwartz *et al.*, 2002). Of these, three reside on the ER (the Nem1p/Spo7p holoenzyme, Tcb3p, and the UDP-N-acetylglucosamine transferase complex, which consists of Alg13p and Alg14p). Such proteins may represent anchors for BMV RNA replication complexes, as the mechanisms that localize BMV RNA replication to ER membranes are not completely understood (Liu *et al.*, 2009). One of these potential interfaces, Tcb3p, normally resides predominantly on the cortical ER membrane near the cell periphery, rather than on the perinuclear ER membrane that is the major site of BMV RNA replication. However, it was recently shown that BMV RNA replication factor 1a interacts with and induces the relocalization of at least one class of cortical ER membrane proteins, the reticulons, to the perinuclear ER (Diaz *et al.*, 2010). In addition, the Nem1p/Spo7p phosphatase complex is involved in regulating phospholipid biosynthesis. Regulated synthesis of new lipids is critical to create a specific, expanded membrane compartment essential for RNA replication by BMV (Lee & Ahlquist, 2003; Zhang *et al.*, 2012) and other positive-strand RNA viruses, a number of which were recently shown to interact with lipid synthesis factors to actively promote lipid synthesis (Chukkapalli *et al.*, 2012). In addition, the Set3p complex is a histone modifier involved in regulating the secretory stress response. This complex might play a role in responding to the extensive occupation of the cell's ER membrane by BMV RNA replication complexes (Restrepo-Hartwig & Ahlquist, 1999; Schwartz *et al.*, 2002).

**Ribosome-associated interface**

One predicted interface is related to ribosomes, which directly interact with BMV genomic and subgenomic RNA to produce all BMV proteins, and thus regulate all steps of BMV replication and gene expression. Dbp2p is involved in processing ribosomal RNA (rRNA) precursors into mature form. The actual yeast-BMV interface might be this rRNA synthesis factor or its rRNA products, which interact with BMV RNAs in their primary role as key ribosomal components. Ribosomal-RNA-related proteins modulate ribosome abundance, which positively and negatively regulates the relative translation levels of different classes of mRNAs, including the competition between polyadenylated cellular mRNAs and non-polyadenylated mRNAs such as those of BMV RNAs (Wickner, 1996). Changes in ribosome synthesis rates, as well as more specific changes, could also alter the specific protein composition of ribosomes, which has can exert dramatic effects on the translation efficiencies of viral mRNAs (Barna, 2013).

**Interfaces implicated in viral RNA or protein interactions**

Additional predicted interface proteins are likely to interact with BMV RNAs or proteins. The Ski complex directs degradation of viral and cellular mRNAs, notably including preferential degradation of non-polyadenylated RNAs like those of BMV (Araki *et al.*, 2001; Wickner, 1996). Consistent with direct Ski-mediated degradation of BMV RNAs, knockout of Ski components increases BMV replication (Kushner *et al.*, 2003). Interestingly, Ski-mediated mRNA degradation involves the exosome, a complex also involved in the rRNA processing discussed above.

The Oca complex (Oca1p, Oca2p, Oca4p-6p and Siw14p) was predicted as an interface because knockouts of each of its genes produced significant BMV replication phenotypes. Two of its subunits, Siw14p and Oca1p, are tyrosine phosphatases. It was suggested previously that these phosphatases may play a role in undermining viral protein phosphorylation events that inhibit RNA replication complex assembly (Kim *et al.*, 2002; Kushner *et al.*, 2003).

Finally, multiple predicted interfaces (19/22s regulatory particle of the proteasome, Cdc34p, and Smt3p) are components of the ubiquitin-proteasome system, which covalently modifies proteins to direct their degradation, intracellular trafficking or other purposes. Many viruses encode proteins that interact with this system to modulate viral protein accumulation, targeting or function, or to direct the degradation of interfering cell proteins (Gao & Luo, 2006; Blanchette & Branton, 2009; Zhang *et al.*, 2009). As these precedents

include many other positive-strand RNA viruses, BMV may well do the same (Choi *et al.*, 2013).

**Interfaces involved in regulation of mRNA transcription**

The remaining predicted interfaces are involved in the regulation of mRNA transcription. As BMV is an RNA virus, these proteins are unlikely to be required for the virus' replication in its natural host. Instead, they may be artifacts of the DNA plasmid-directed experimental system used to artificially initiate BMV replication in yeast.

### 3.3.4 Impact of provided domain knowledge

One significant advantage of our approach is that it enables domain knowledge to be readily incorporated into the inferred subnetworks. Specifically, the IP can incorporate constraints that represent knowledge about host factors and interactions that are known to be involved in viral replication, thereby influencing decisions about the rest of the subnetwork. These constraints are shown in Section 3.2.2.

Here, we consider the effect of seeding the subnetworks with interactions from specific host pathways that are known to be involved in BMV replication. This set of domain knowledge, which we have elicitepd from the relevant literature, comprises 28 interactions among 24 host factors. It also specifies several host factors that should be treated as interfaces. For comparison, we also infer BMV subnetwork ensembles that do not use the literature-curated interactions. Seeding the subnetwork with these interactions does not have any apparent effect on hit-prediction accuracy, as we discussed earlier in Section 3.3.7. However, the interactions do appear to have an influence on their local neighborhoods. In examining the 97-interface BMV subnetwork ensemble, we observe a small number of cases in which the supplied interactions and interfaces serve to provide "anchors" that allow us to explain other, related hits.

One set of edges extracted from the literature connects the ubiquitin-proteasome pathway to membrane synthesis, and specifies that Ole1p is an interface to BMV. The inferred subnetwork identifies a connection between Ole1p, a fatty acid desaturase, and Acb1p, which is involved in transporting newly synthesized fatty acids; the relevant portion of the subnetwork is shown in Figure 3.7. The connection between the ubiquitin-proteasome pathway and Acb1p was not identified in any subnetwork inferred without the provided literature-based interactions. Furthermore, Ole1p is not inferred to be relevant at all without the provided interactions.

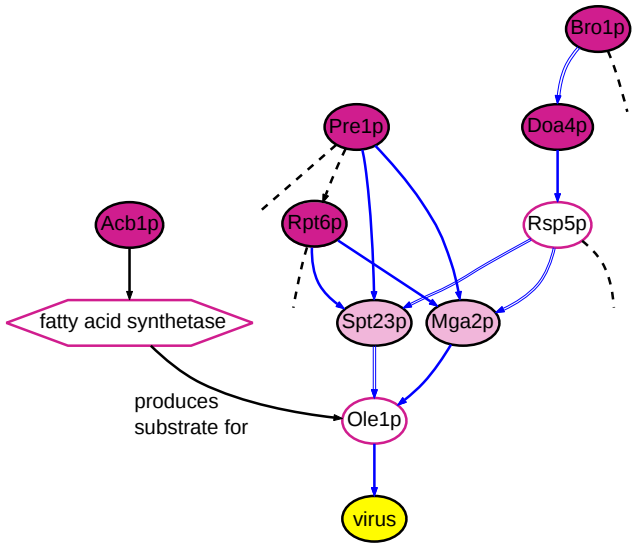



**Figure 3.7:** A component from the inferred subnetwork ensemble showing a connection between Acb1p and the literature-extracted ubiquitin-proteasome-system interactions. All node and edge predictions shown have confidence=1.0 in the ensemble. A dashed edge with no terminal indicates connections to the rest of the subnetwork. Edges extracted from literature are colored blue. Doubled blue edges (as from Rsp5p to Spt23p) indicate literature-extracted edges that were also present in the original background network. See Figure 3.1 for a key to the other network elements.

Another component from the literature specifies the chaperone protein Ydj1p is an interface. The inferred subnetwork, shown in Figure 3.8, identifies upstream connections from the hits Hsf1p and Ure2p to Ydj1p, which were not mentioned in the paper discussing Ydj1p's relationship to BMV (Tomita *et al.*, 2003). These inferred connections demonstrate that the inferred subnetwork can be used to predict relevant connections between well-understood components of the network and host factors that have not yet been studied in detail.

### 3.3.5 Gene Ontology analysis of inferred BMV subnetwork

To supplement our manual analysis of predicted hits and interfaces, we employ the Model Based Gene-Set Analysis (MGSA) tool (Bauer *et al.*, 2011) to evaluate the ability of the inferred subnetwork to better identify relevant functional categories than an analysis of the experimental hits alone. The MGSA method uses a Bayesian network to analyze the representation of all GO terms in a gene set at once. As output, it provides the marginal probability that each GO term accounts for the input gene set. We use MGSA to analyze first the experimental hits and literature-derived relevant genes for BMV that are present in

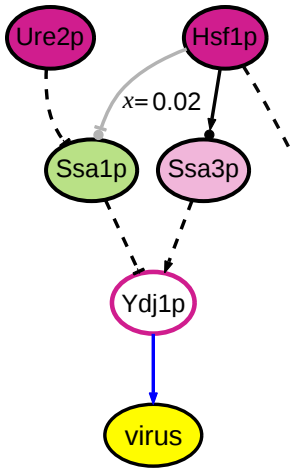



**Figure 3.8:** A component from the inferred subnetwork ensemble showing a connection between the literature-identified interface Ydj1p and two hits, Hsf1p and Ure2p. The blue edge from Ydj1p to the virus was originally extracted from literature. See Figure 3.1 for a key to the other network elements.

the background network, and second, the experimental hits combined with the predicted hits from the 97-interface inferred subnetwork. We use a probability threshold of 0.25 because we are willing to tolerate a degree of redundancy in the results, in exchange for the identification of a thorough list of representative GO terms.

We further assess the significance of each returned GO term by comparison to the subnetworks inferred from random data. For each GO term, we generate a $p$-value as the proportion of random subnetworks for which MGSA gives a greater or equal probability.

Table 3.7 presents the GO terms returned by MGSA with probability $\geq 0.25$ for the combined set of experimental and predicted hits. The "Experimental Hits" columns show the number of experimental hits associated with each GO term, and MGSA's probability that the GO term explains the experimental hits alone. Similarly, the "Predicted Hits" columns show the number of additional predicted hits associated with the GO term, and MGSA's probability that the GO term explains the combined experimental and predicted hit set. The "$p$-value" column shows the proportion of random subnetworks with equal or greater probability for the GO term as compared to the inferred subnetwork, with asterisks indicating $p < 0.05$.

As shown in Table 3.8, an additional 15 GO terms are identified by MGSA for the combined hit set, but are not identified for the experimental hits alone. A number of these GO terms represent only predicted hits. Eight of the GO terms receive a $p < 0.05$ from the random subnetwork analysis. This result indicates that our subnetwork inference method predicts hits that (i) are useful for amplifying weak functional signals among

the experimental hits, and (ii) are among themselves functionally coherent. Several of the amplified GO terms represent protein complexes or pathways that are recognized for their role in BMV replication. Deadenylation-dependent mRNA decapping factors are also known to be relevant (Noueiry *et al.*, 2003), and the perinuclear region of the cytoplasm is the cellular location in which BMV replicates (Restrepo-Hartwig & Ahlquist, 1999). Among the novel GO terms that contain no experimental hits are represented specific parts of the ubiquitin-proteasome system and ribosome synthesis, both of which we have noted are relevant to BMV replication.

### 3.3.6 Permutation analysis of inferred relevant complexes

One advantage of our method is that we explicitly include protein complexes as nodes in our background network. We propose that doing so allows the inferred subnetworks to provide useful information about cooperative interactions between proteins. We use two Monte Carlo tests to assess the degree to which the representation of complexes among inferred subnetworks, and specifically among inferred interfaces, is due to (i) topological properties of the background network and inference procedure, independent of the experimental data, and (ii) properties of the experimental data, independent of the inference procedure. We separately consider inferred relevant complexes and complexes that are or contain high-confidence interfaces.

#### Representation of complexes among experimental and predicted hits

When we considered the biological plausibility of predicted-but-unassayed host factors for BMV, we noted that a number of them are members of large protein complexes, including the proteasome and Mediator. The 97-interface inferred BMV subnetwork contains 65 protein complex nodes with confidence $\geq$ 0.75.

In a first permutation test, we assess whether the degree to which protein complexes are represented in the inferred subnetwork is simply due to properties of the underlying background network, independent of the experimental data. First, for each of the 65 complexes that are predicted to be relevant, we count how many of its constituent genes are experimental or predicted hits. Next, we estimate a distribution of random inferred subnetworks by applying a 0.75-confidence threshold to each of the 1,000 ensembles that we inferred from permuted data to generate baselines for the hit- and sign-prediction tests. For each complex, we record the fraction of the complex that is represented by permuted and predicted hits in each random subnetwork. Finally, we calculate a permutation-based

**Table 3.7:** Gene Ontology terms represented by both experimental and predicted BMV hits. GO terms that MGSA returns for both BMV experimental hits alone and for the inferred BMV subnetwork (with probability $\geq$ 0.25). In the "Experimental hits" column are shown the number of hits associated with the GO term, and MGSA's probability that the GO term explains the experimental hit set alone. In the "Predicted hits" column are shown the number of predicted hits associated with the GO term, and MGSA's probability that the GO term explains the combined experimental and predicted hit set. The column "*p*-value" shows the proportion of random subnetworks for which the MGSA probability of the GO term is greater than or equal to that of the inferred subnetwork; asterisks indicate $p < 0.05$.

| | | | Experimental hits | | Predicted hits | | | |
|---|---|---|---|---|---|---|---|---|
| GO ID | Description | Size | Count | Prob. | Count | Prob. | *p*-value | |
| 32447 | protein urmylation | 7 | 6 | 0.975 | 0 | 0.860 | 0.002 | * |
| 33588 | Elongator holoenzyme complex | 6 | 5 | 0.974 | 1 | 0.783 | 0.084 | |
| 55087 | Ski complex | 4 | 4 | 0.961 | 0 | 0.783 | $< 0.001$ | * |
| 32874 | positive regulation of stress-activated MAPK cascade | 3 | 3 | 0.954 | 0 | 0.694 | $< 0.001$ | * |
| 71782 | endoplasmic reticulum tubular network | 3 | 3 | 0.848 | 0 | 0.685 | $< 0.001$ | * |
| 5688 | U6 snRNP | 8 | 5 | 0.704 | 0 | 0.593 | 0.216 | |
| 445 | THO complex part of transcription export complex | 4 | 2 | 0.265 | 1 | 0.376 | 0.031 | * |
| 446 | nucleoplasmic THO complex | 4 | 2 | 0.307 | 1 | 0.363 | 0.038 | * |
| 32784 | regulation of DNA-dependent transcription, elongation | 3 | 2 | 0.288 | 1 | 0.292 | $< 0.001$ | * |
| 5732 | small nucleolar ribonucleoprotein complex | 9 | 5 | 0.260 | 0 | 0.272 | 0.049 | * |
| 3724 | RNA helicase activity | 5 | 3 | 0.356 | 1 | 0.264 | 0.005 | * |
| 71072 | negative regulation of phospholipid biosynthetic process | 2 | 2 | 0.369 | 0 | 0.262 | $< 0.001$ | * |
| 36083 | positive regulation of unsaturated fatty acid biosynthetic process by positive regulation of transcription from RNA Pol II promoter | 2 | 2 | 0.321 | 0 | 0.251 | $< 0.001$ | * |

**Table 3.8:** Additional Gene Ontology terms represented by the inferred BMV subnetwork. GO terms that MGSA returns for the inferred BMV subnetwork, but not for the BMV experimental hits alone (with probability $\geq$ 0.25). In the "Experimental hits" column are shown the number of hits associated with the GO term, and MGSA's probability that the GO term explains the experimental hit set alone. In the "Predicted hits" column are shown the number of predicted hits associated with the GO term, and MGSA's probability that the GO term explains the combined experimental and predicted hit set. The column "$p$-value" shows the proportion of random subnetworks for which the MGSA probability of the GO term is greater than or equal to that of the inferred subnetwork; asterisks indicate $p < 0.05$.

| | | | Experimental hits | | Predicted hits | | | |
|---|---|---|---|---|---|---|---|---|
| GO ID | Description | Size | Count | Prob. | Count | Prob. | $p$-value | |
| 42790 | transcription of nuclear large rRNA transcript from RNA polymerase I promoter | 17 | 3 | 0.022 | 5 | 0.977 | 0.001 | * |
| 502 | proteasome complex | 43 | 7 | 0.057 | 24 | 0.947 | 0.083 | |
| 70847 | core mediator complex | 20 | 0 | – | 8 | 0.873 | 0.301 | |
| 290 | deadenylation-dependent decapping of nuclear-transcribed mRNA | 9 | 3 | 0.143 | 2 | 0.798 | 0.021 | * |
| 48471 | perinuclear region of cytoplasm | 12 | 3 | 0.036 | 2 | 0.636 | 0.003 | * |
| 124 | SAGA complex | 20 | 0 | – | 7 | 0.631 | 0.105 | |
| 71629 | cytoplasm-associated proteasomal ubiquitin-dependent protein catabolic process | 4 | 0 | – | 3 | 0.621 | 0.001 | * |
| 34455 | t-UTP complex | 7 | 0 | – | 4 | 0.470 | 0.104 | |
| 30015 | CCR4-NOT core complex | 9 | 0 | – | 4 | 0.450 | 0.213 | |
| 33553 | rDNA heterochromatin | 9 | 0 | – | 5 | 0.430 | 0.014 | * |
| 34388 | Pwp2p-containing subcomplex of 90S preribosome | 6 | 0 | – | 3 | 0.426 | 0.100 | |
| 31146 | SCF-dependent proteasomal ubiquitin-dependent protein catabolic process | 17 | 0 | – | 6 | 0.390 | 0.104 | |
| 6750 | glutathione biosynthetic process | 2 | 0 | – | 2 | 0.315 | 0.016 | * |
| 6283 | transcription-coupled nucleotide-excision repair | 9 | 0 | – | 4 | 0.291 | 0.017 | * |
| 7584 | response to nutrient | 2 | 0 | – | 2 | 0.288 | 0.002 | * |

$p$-value for each complex as the proportion of permutations that contain as many or more members of the complex.

Out of 65 inferred relevant complexes, 16 are better represented ($p < 0.05$) by our inferred BMV subnetwork than by random inferred subnetworks. This result demonstrates that the representation of many of the complexes in our inferred subnetworks is not merely due to the topology of the background network. Among these 16 are two components of the proteasome: the 20S proteasome and its 19/22S regulatory particle. However, the Mediator complex, another large complex predicted to be relevant by our method, is not significantly better represented by our inferred subnetwork than by random subnetworks. While we cannot rule out the possibility that our method inferred it to be relevant simply because it is highly central in the background network, it is still possible that the Mediator is actually relevant to viral replication.

In a second Monte Carlo test, we consider whether the degree of involvement of *predicted* hits in relevant complexes is due to properties of the underlying background network and experimental data, independent of the *inference procedure*. For this test, we measure the representation of each complex by the set of predicted hits from the inferred BMV subnetwork. To estimate a random distribution of predicted hit sets, we draw 1,000 random subsets of the weak-phenotype genes and unassayed genes, in the same size and degree distribution as the set of predicted hits from the inferred BMV subnetwork. Again, we calculate the permutation-based $p$-value for a complex as the fraction of random predicted hit sets that contain as many or more members of the complex.

Eleven out of the 65 predicted relevant complexes are better represented ($p < 0.05$) by the inferred BMV subnetwork than by random hit predictions, including the two proteasome complexes as well as the Mediator. This result shows that our method can predict the inclusion of complexes that would not have been just as well-represented by randomly drawn genes.

Table 3.9 shows the predicted relevant complexes that are significant under either of these tests.

**Representation of complexes that are used as interfaces**

We use another pair of analogous Monte Carlo tests to examine the high-confidence, predicted BMV-yeast interfaces that are protein complexes or are members of complexes. Six high-confidence interfaces are themselves complexes, three of which are entirely composed of hits and are not considered during this analysis. Another five are genes are members of one or two complexes in the background network.

For the first test, we assess the representation of the interface-related complexes in 1,000 subnetworks inferred using permuted data and a minimum number of interfaces (the median was 52). Five out of the eight interfaces that are related to complexes that are better represented by the inferred BMV subnetwork than by random subnetworks ($p < 0.05$), including all three of the complexes that are used directly as interfaces.

For the second test, we assess the representation of the interface-related complexes by randomly drawn subsets of weak and unassayed genes. Four out of the eight complexes are significantly better represented by the inferred BMV subnetwork than by a random gene set ($p < 0.05$), including two out of the three complexes that are directly used as interfaces.

Taken together, the results of both of these analyses indicate that the complexes that our method predicts to be interfaces are well-supported by predicted hits and are not likely to be artifacts of the background network or chance. Table 3.10 shows the predicted relevant interfaces that can be accounted for by a complex that has a significant $p$-value in either of the Monte Carlo tests.

**Size of inferred subnetworks**

In Table 3.11 we provide the average number of predicted hits in the ensembles inferred using all experimental hits, separately reporting the number of predicted hits with weak phenotype labels (the “Relevant weak” column) and the number of unassayed predicted hits (“Relevant unassayed").

**Stability of the leave-one-out subnetworks**

To examine the robustness of our inference method, we measure how well the ensemble inferred using complete experimental data agrees with the ensembles inferred during the leave-one-out experiments. Specifically, we compare four types of predictions: (i) which nodes are relevant ($y_n$), (ii) the phenotype signs of relevant nodes ($v_n$ when $x_n = 1$), (iii) which nodes are interfaces ($x_e$ for edges from predicted interfaces to the virus), and (iv) the relevance of nodes that are predicted to be interfaces ($y_n$, considering only nodes that are ever predicted to be interfaces by any ensemble, but regardless of the confidence in that prediction).

We measure the similarity, or agreement, between the predictions of two ensembles $E$ (complete experimental data) and $E'$ (one missing test case) as follows. Using the variable

**Table 3.9:** Enriched, predicted relevant, protein complexes. Complexes in this table received a $p$-value $< 0.05$ (indicated with *) in either the test based on inferring permuted subnetworks ("Permuted subnetworks"), or the test based on drawing random predicted hits ("Random predictions"). Complexes are listed in decreasing order of the number of predicted hits. The final three rows list complexes that are entirely composed of experimental hits.

| | | | | $p$-value | | | |
|---|---|---|---|---|---|---|---|
| Complex name | Size | Hits | Predicted hits | Permuted subnetworks | | Random predictions | |
| 19/22S regulator | 22 | 3 | 16 | $< 0.001$ | * | $< 0.001$ | * |
| Mediator | 25 | 2 | 11 | 0.243 | | $< 0.001$ | * |
| 20S proteasome | 14 | 2 | 10 | 0.007 | * | $< 0.001$ | * |
| Nsp1p complex | 4 | 1 | 3 | 0.02 | * | 0.017 | * |
| nuclear ubiquitin ligase complex | 4 | 1 | 3 | 0.04 | * | 0.021 | * |
| UTP B complex | 6 | 1 | 3 | 0.093 | | 0.006 | * |
| THO complex | 4 | 2 | 2 | 0.014 | * | 0.029 | * |
| karyopherin docking subcomplex of the Nuclear Pore Complex | 3 | 1 | 2 | 0.048 | * | 0.014 | * |
| Mot1p complex | 2 | 0 | 2 | 0.039 | * | 0.066 | |
| Decapping Enzyme Complex | 2 | 0 | 2 | 0.118 | | 0.018 | * |
| Elongator complex | 6 | 5 | 1 | 0.004 | * | 0.176 | |
| Set3p complex | 7 | 3 | 1 | $< 0.001$ | * | 0.378 | |
| UDP-N-acetylglucosamine transferase complex | 2 | 1 | 1 | 0.036 | * | 0.057 | |
| Bub1p/Bub3p complex | 2 | 1 | 1 | 0.036 | * | 0.107 | |
| DSIF complex | 2 | 1 | 1 | 0.048 | * | 0.215 | |
| Lge1p/Bre1p complex | 2 | 1 | 1 | 0.002 | * | 0.034 | * |
| Prs1p/Prs3p | 2 | 1 | 1 | 0.05 | | 0.045 | * |
| OCA complex | 6 | 6 | 0 | $< 0.001$ | * | – | |
| Ski Complex | 3 | 3 | 0 | 0.014 | * | – | |
| Nem1p/Spo7p complex | 2 | 2 | 0 | $< 0.001$ | * | – | |

**Table 3.10:** Significantly enriched complexes that account for predicted interfaces. If the complex contains an interface and is not a predicted interface itself, the protein name of the predicted interface is listed in the "Interface" column. The final three rows list complexes that are entirely composed of hits.

| | | | | | $p$-value | | | |
|---|---|---|---|---|---|---|---|---|
| Complex name | Interface | Size | Hits | Predicted hits | Permuted subnetworks | | Random predictions | |
| 19/22S regulator | | 22 | 3 | 14 | 0.002 | * | $< 0.001$ | * |
| SCF-Cdc4 complex | Cdc34p | 5 | 1 | 3 | 0.015 | * | 0.017 | * |
| Ada2p/Gcn5p/Ada3 transcription activator complex | Gcn5p | 5 | 0 | 3 | 0.027 | * | 0.051 | |
| Tup1p/Ssn6p complex | Tup1p | 2 | 0 | 2 | 0.05 | | 0.049 | * |
| Set3p complex | | 7 | 3 | 1 | $< 0.001$ | * | 0.174 | |
| UDP-N-acetylglucosamine transferase complex | | 2 | 1 | 1 | 0.029 | * | 0.034 | * |
| Nem1p/Spo7p complex | | 2 | 2 | 0 | $< 0.001$ | * | – | |
| OCA complex | | 6 | 6 | 0 | $< 0.001$ | * | – | |
| Ski Complex | | 3 | 3 | 0 | 0.009 | * | – | |

**Table 3.11:** Average number of predicted relevant weak-phenotype and unassayed genes in the inferred subnetworks.

| Data set | $\gamma$ | Relevant weak | Relevant unassayed |
|---|---|---|---|
| BMV | 47 | 44.32 | 112.13 |
| | 72 | 79.96 | 179.96 |
| | 97 | 107.36 | 203.56 |
| | 122 | 125.34 | 219.81 |
| | 147 | 141.50 | 235.18 |
| FHV | 26 | 20.91 | 50.10 |
| | 51 | 48.62 | 100.05 |
| | 76 | 65.84 | 129.79 |
| | 101 | 80.19 | 148.86 |
| | 126 | 93.68 | 171.00 |

**Table 3.12:** Stability of leave-one-out inferred subnetworks. Stability is calculated relative to ensembles inferred using all data. For both data sets, we show the stability of four types of predictions across each setting of $\gamma$.

| Data set | $\gamma$ | Node relevance | Node phenotype signs | Interface predictions | Interface relevance |
|---|---|---|---|---|---|
| BMV | 47 | 0.853 | 0.843 | 0.765 | 0.878 |
| | 72 | 0.909 | 0.796 | 0.734 | 0.907 |
| | 97 | 0.92 | 0.803 | 0.712 | 0.896 |
| | 122 | 0.928 | 0.814 | 0.761 | 0.897 |
| | 147 | 0.928 | 0.835 | 0.776 | 0.898 |
| FHV | 26 | 0.817 | 0.796 | 0.716 | 0.790 |
| | 51 | 0.891 | 0.742 | 0.742 | 0.867 |
| | 76 | 0.941 | 0.747 | 0.82 | 0.910 |
| | 101 | 0.856 | 0.795 | 0.696 | 0.788 |
| | 126 | 0.889 | 0.728 | 0.768 | 0.847 |

$y_n$ (node relevance) as an example, $p^E(y_n = 1)$ is $E$'s confidence that node $n$ is relevant.

$$similarity(E, E') = 1 - \frac{\sum_{n \in \mathcal{N} - \mathcal{N}^H} |p^E(y_n = 1) - p^{E'}(y_n = 1)|}{\sum_{n \in \mathcal{N} - \mathcal{N}^H} p^E(y_n = 1)}$$

We define stability as the mean similarity between the complete-data ensemble and each leave-one-out ensemble.

Stability results are shown in Table 3.12. Our method's predictions about node relevance for BMV are highly stable, with average agreement between the complete ensemble and leave-one-out ensembles at or above 90% for ensembles inferred using $\gamma = \{72, 97, 122, 147\}$ interfaces; for $\gamma = 47$ interfaces, the node relevance agreement is slightly lower at 85%. Phenotype sign predictions show slightly lower agreement (80–84%), as do predictions about which nodes are interfaces (71–78%). However, predicted interfaces are still likely to be deemed relevant across the set of ensembles, even if they are not as consistently predicted to be interfaces (88–91%). Overall, predictions for FHV are somewhat less stable than those for BMV, which may be due to the greater connectivity of BMV's hits compared to FHV's.

### 3.3.7 Varying components of the IP

As discussed in Chapter 2 and Section 3.1.1, several integer programming methods have been developed to infer signalling and regulatory networks from experimental data that comes in the form of source-target pairs. A key aspect of our approach is that it does not assume that targets are given. Instead, it infers the downstream interfaces. Existing IP approaches are therefore not directly applicable to our own task. However, we consider some components of existing methods that can be substituted into our integer program: namely, two alternative objective functions, and one alternative heuristic for inferring edge signs. Additionally, in this section, we explore the effect of varying some of the previously discussed parameters and constraints of our IP.

#### Alternative objective functions

While our objective function maximizes the total diffusion kernel score of relevant nodes, a common goal of other network inference methods is to maximize the number of paths that connect sources and targets. Edges or nodes may also be weighted, giving rise to a weight for each path. Inspired by these methods, particularly by the work by Ourfali *et al.* (2007) and Gitter *et al.* (2011), we consider two path-based objective functions as alternatives to the node-based objective function that we presented in Section 3.2.2).

*Maximize the total count of inferred relevant paths* (`MP-Count`):

$$\max \sum_{p \in \mathcal{P}} \sigma_p$$

*Maximize the total score of inferred relevant paths* (`MP-Score`). In advance of inference, we calculate the score of a path as the sum of the diffusion-kernel-derived scores of the nodes in the path:

$$\max \sum_{p \in \mathcal{P}} \text{score}(p)\, \sigma_p, \quad \text{where } \text{score}(p) = \sum_{n \in \mathcal{N}(p)} \text{score}(n)$$

We compare the predictive accuracy of these two path-based objective functions to our node-based objective function using the hit- and sign-prediction tasks that we described previously. Full results for both hit- and sign-prediction tasks for all levels of $\gamma$ are shown in Figures 3.9 and 3.10 (BMV) and Figures 3.11 and 3.12 (FHV). For both hit- and sign-prediction, the two path-based objective functions perform comparably to our node-based one; thus it does not appear that our IP method is very sensitive to the choice of objective function among the options tested.

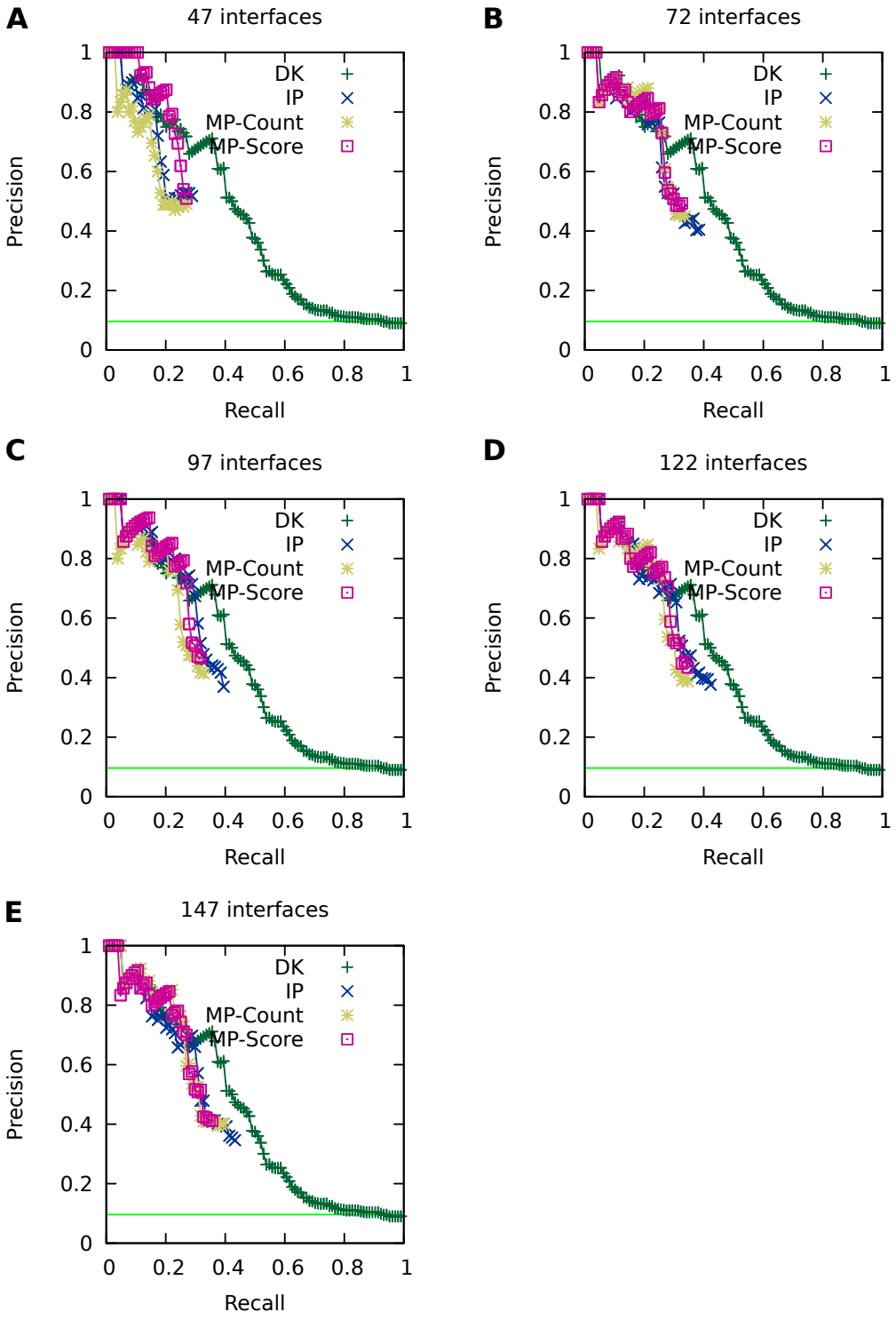



**Figure 3.9:** Hit-prediction results for alternative objective functions; BMV. Precision-recall curves comparing to path-based objective functions. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=47 (**A**), 72 (**B**), 97 (**C**), 122 (**D**), 147 (**E**).

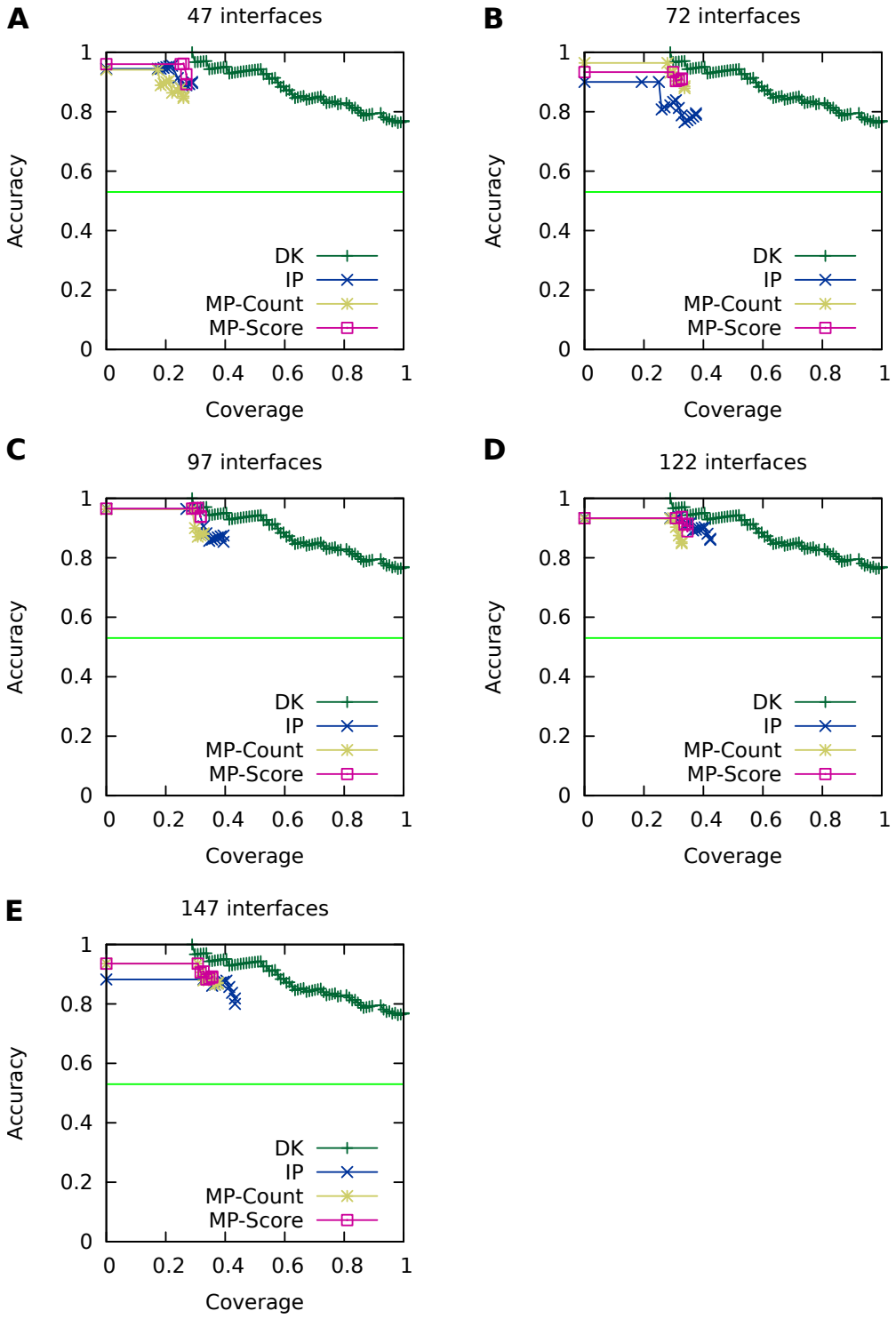



**Figure 3.10:** Sign-prediction results for alternative objective functions; BMV. Accuracy-coverage curves for alternative, path-based objective functions. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=47 (**A**), 72 (**B**), 97 (**C**), 122 (**D**), 147 (**E**).

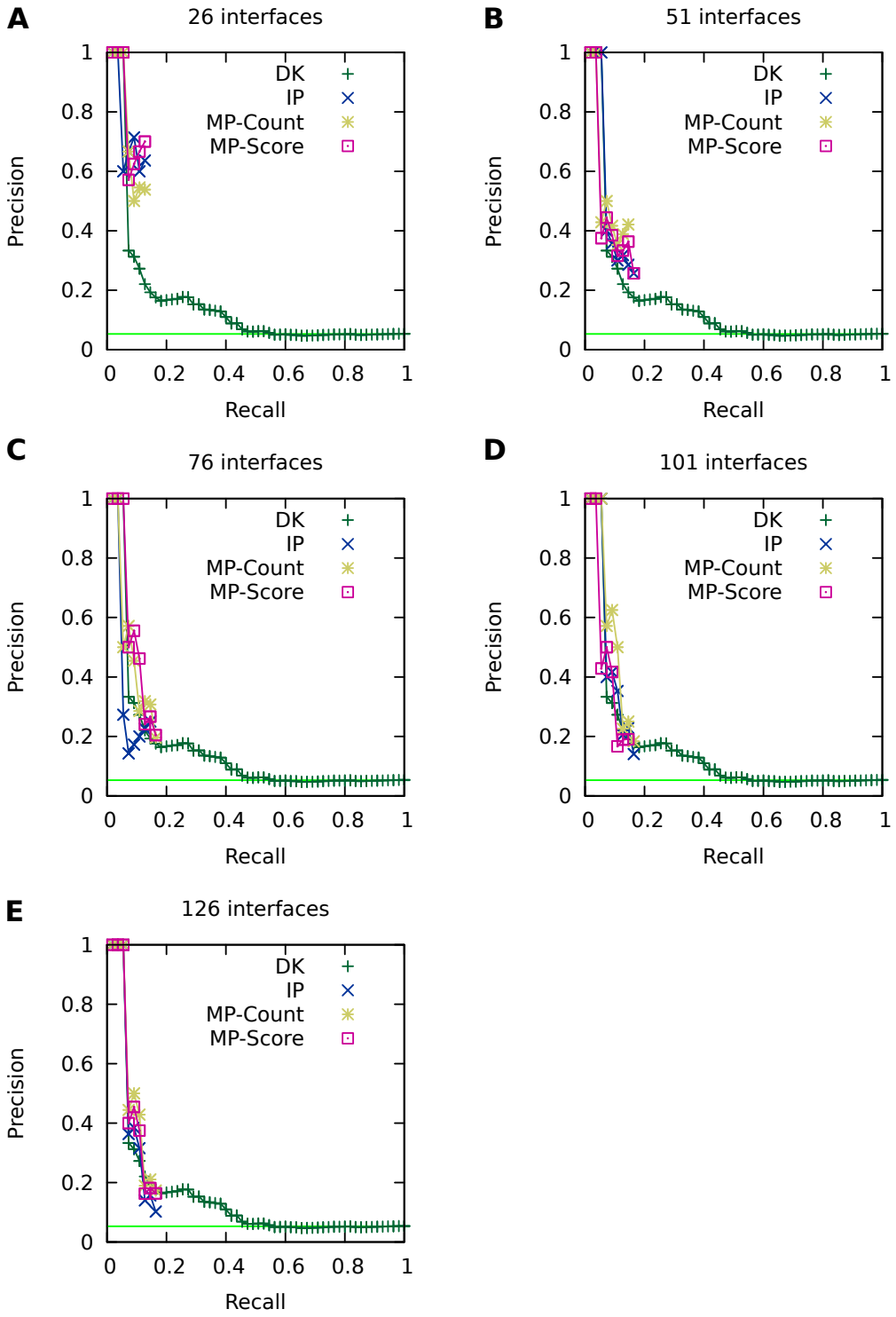



**Figure 3.11:** Hit-prediction results for alternative objective functions; FHV. Precision-recall curves comparing to path-based objective functions. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

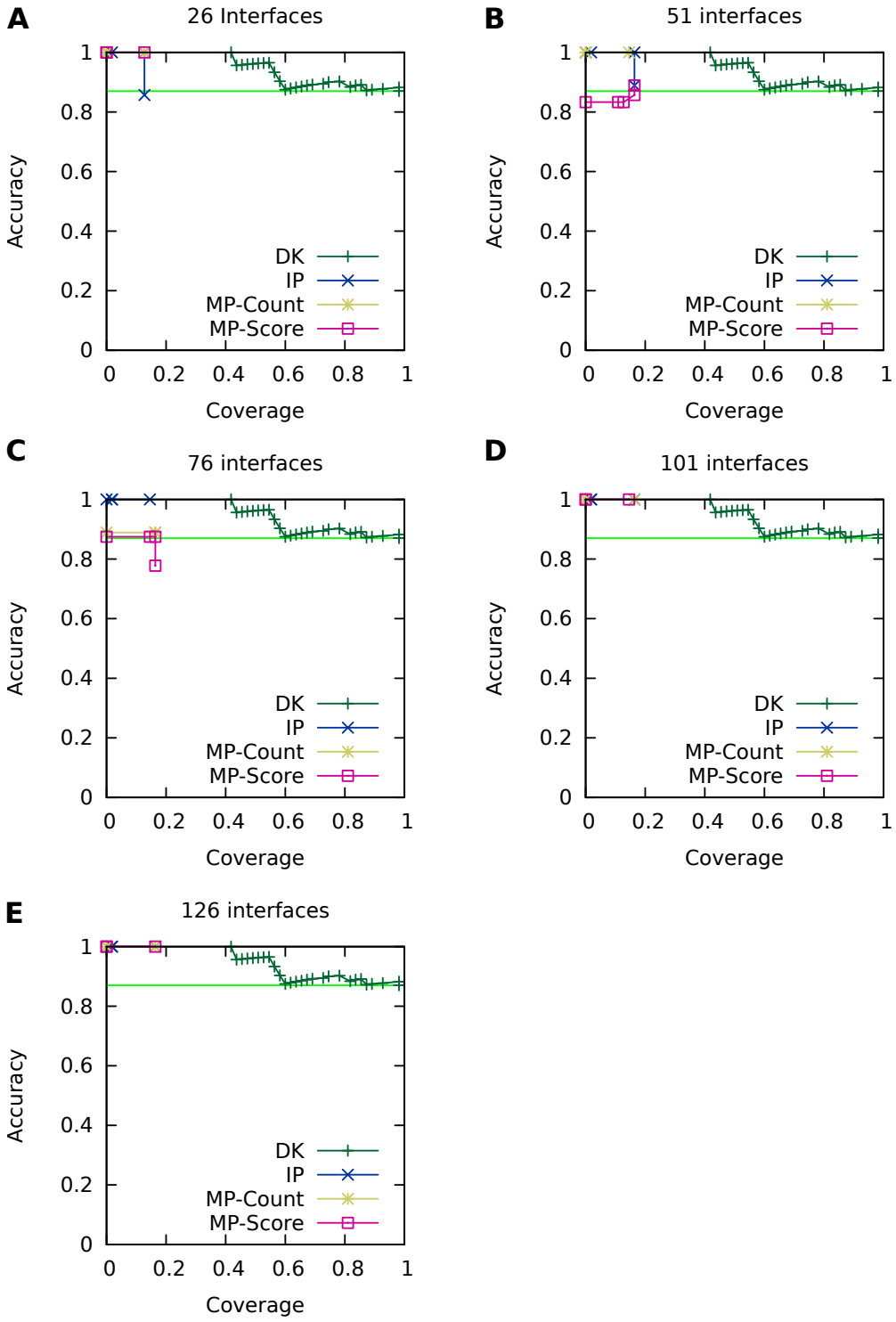



**Figure 3.12:** Sign-prediction results for alternative objective functions; FHV. Accuracy-coverage curves for alternative, path-based objective functions. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

**Alternative edge sign heuristic**

In their SPINE method for inferring signalling networks from source-target pairs, Ourfali *et al.* (2007) employ the assumption that each node is either a repressor or an activator. That is, all edges leaving from a node must have the same sign. Our own heuristic simply requires that, globally, at least 90% of edges must be activating. To compare the two, we constructed an alternate version of our IP that contains SPINE's heuristic. This is achieved with the introduction of two new variables per node and four new constraints per edge (and fewer for edges that have a fixed sign).

The two new node variables are $a_n$, which is set to 1 if the node is predicted to be an activator, and $h_n$, which is set to 1 if the node is predicted to be a repressor (in*h*ibitor). Each node can have at most one set:

$$\forall n \in \mathcal{N} \qquad\qquad y_n = a_n + h_n$$

If a node $n$ is an activator, then any relevant outgoing edge $e$ must be activating; likewise, if $n$ is an inhibitor, so are all of its outgoing edges. For $n$'s outgoing directed edges, this is fairly straightforward to constrain. (The set $\mathcal{E}^I - \mathcal{E}^U$ refers to directed edges.)

$$\forall e = (n_i, n_j) \in \mathcal{E}^I - \mathcal{E}^U \qquad\qquad \begin{aligned} a_e &\leq a_{n_i} \\ h_e &\leq h_{n_i} \end{aligned}$$

For undirected edges, however, an edge's sign is only tied to a node's activator/repressor status if the node is the source of the edge. $out(e, n)$ specifies the value of $d_e$ for which $n$ is the source of the edge.

$$\forall e = (n_i, n_j) \in \mathcal{E}^U \qquad\qquad \begin{aligned} a_e + out(e, n_i) &\leq 1 + a_{n_i} \\ h_e + I(d_e = out(e, n_i)) &\leq 1 + h_{n_i} \end{aligned}$$

We use the sign-prediction task to compare our heuristic to SPINE's (Figures 3.13 to 3.16). As shown in Figures 3.13 and 3.14, our global constraint results in higher sign-prediction accuracy than SPINE's locally-based constraints for nearly all numbers of interfaces. Under the SPINE heuristic, the majority of edges are still inferred to be activating. Among the ensembles that were used to generate the BMV SPINE sign curve in Figure 3.15C, the proportion of activating edges ranges from 0.73–0.84, with a median of 0.79. Accordingly, the SPINE heuristic's sign-prediction accuracy appears comparable to that of setting $\alpha$=0.8. It does not appear to have an effect on hit-prediction accuracy.

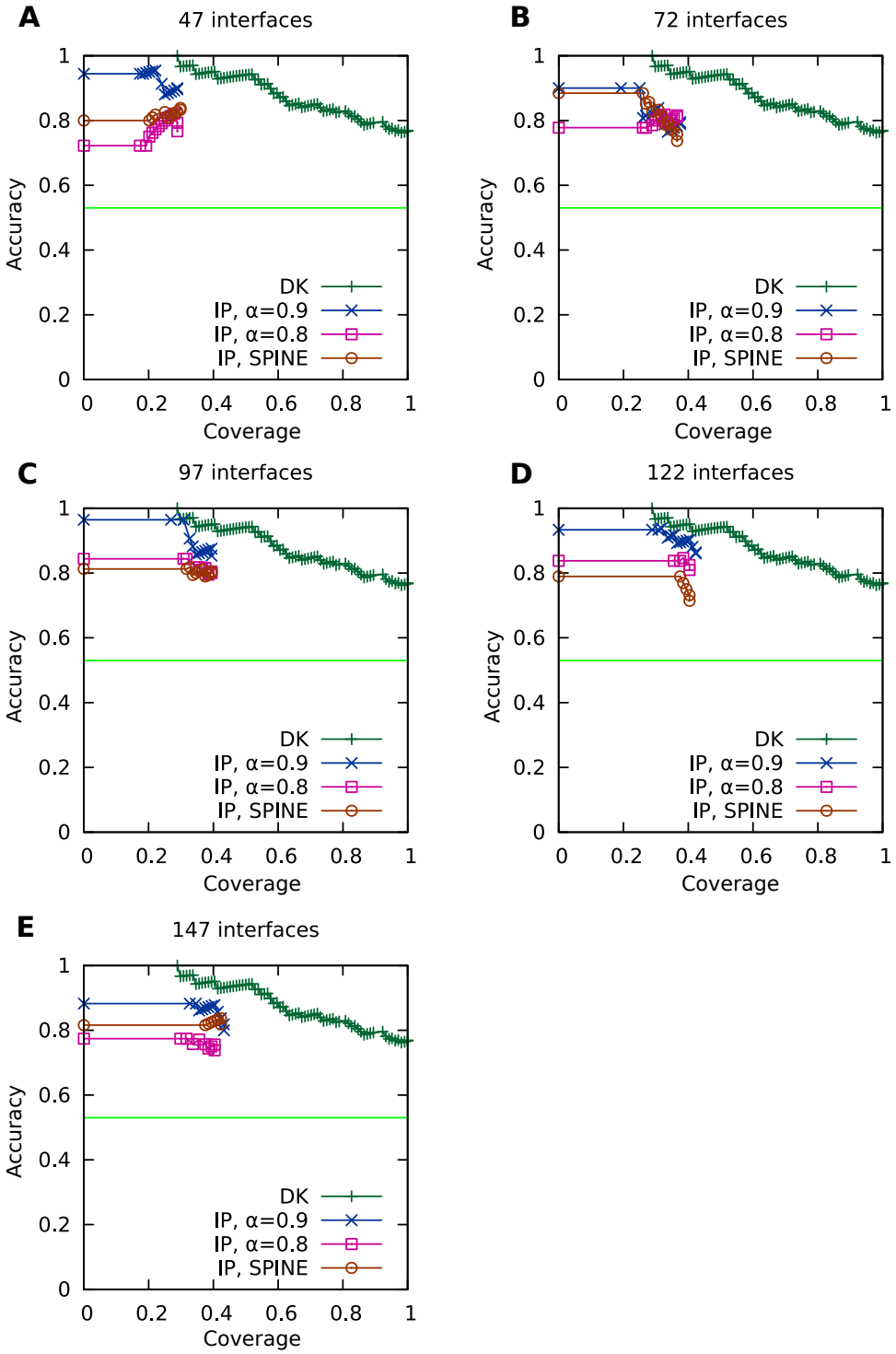



**Figure 3.13:** BMV sign-prediction results for the SPINE phenotype-sign heuristic. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

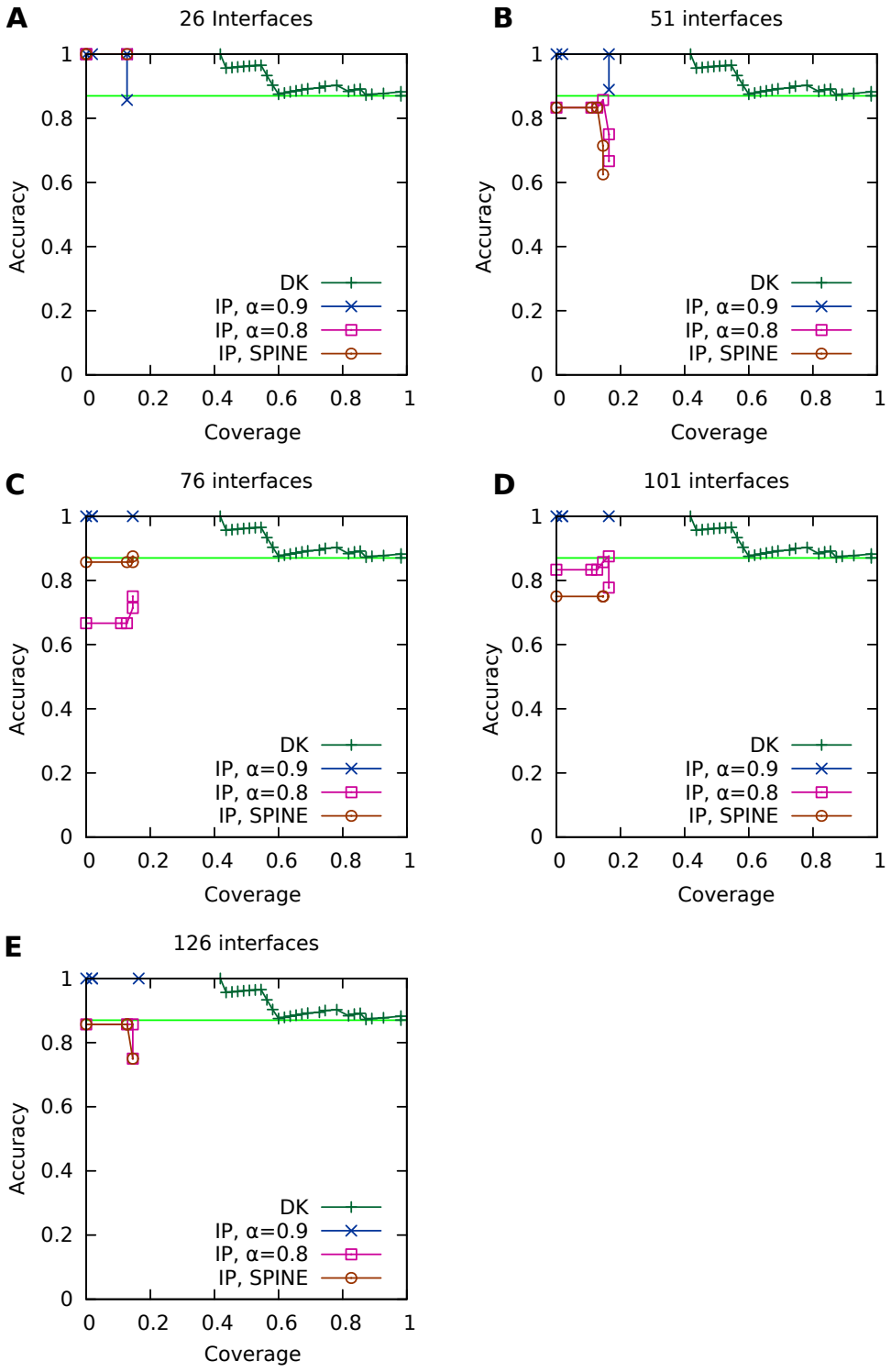



**Figure 3.14:** FHV sign-prediction results for the SPINE phenotype-sign heuristic. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

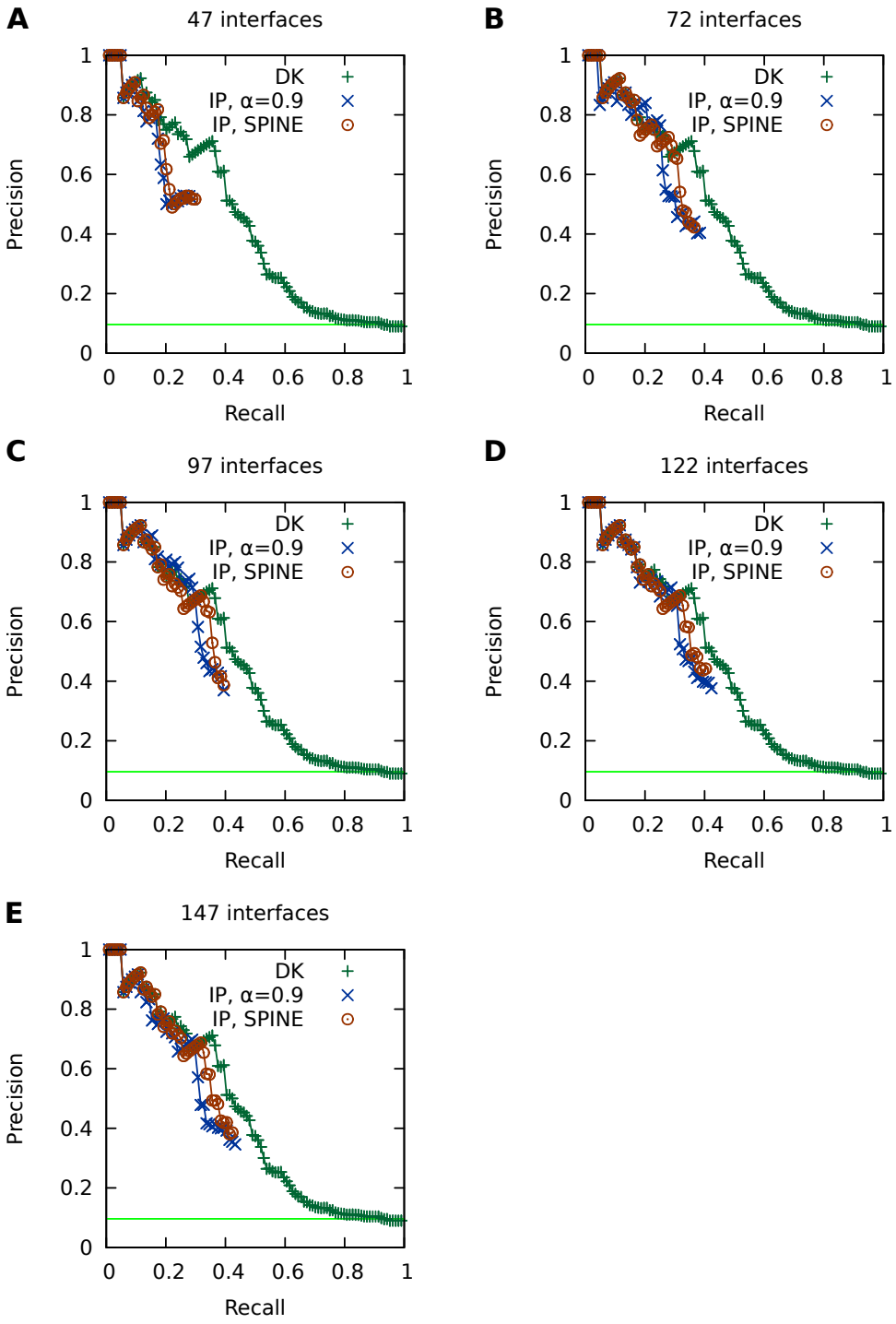



**Figure 3.15:** BMV hit-prediction results for the SPINE phenotype-sign heuristic. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

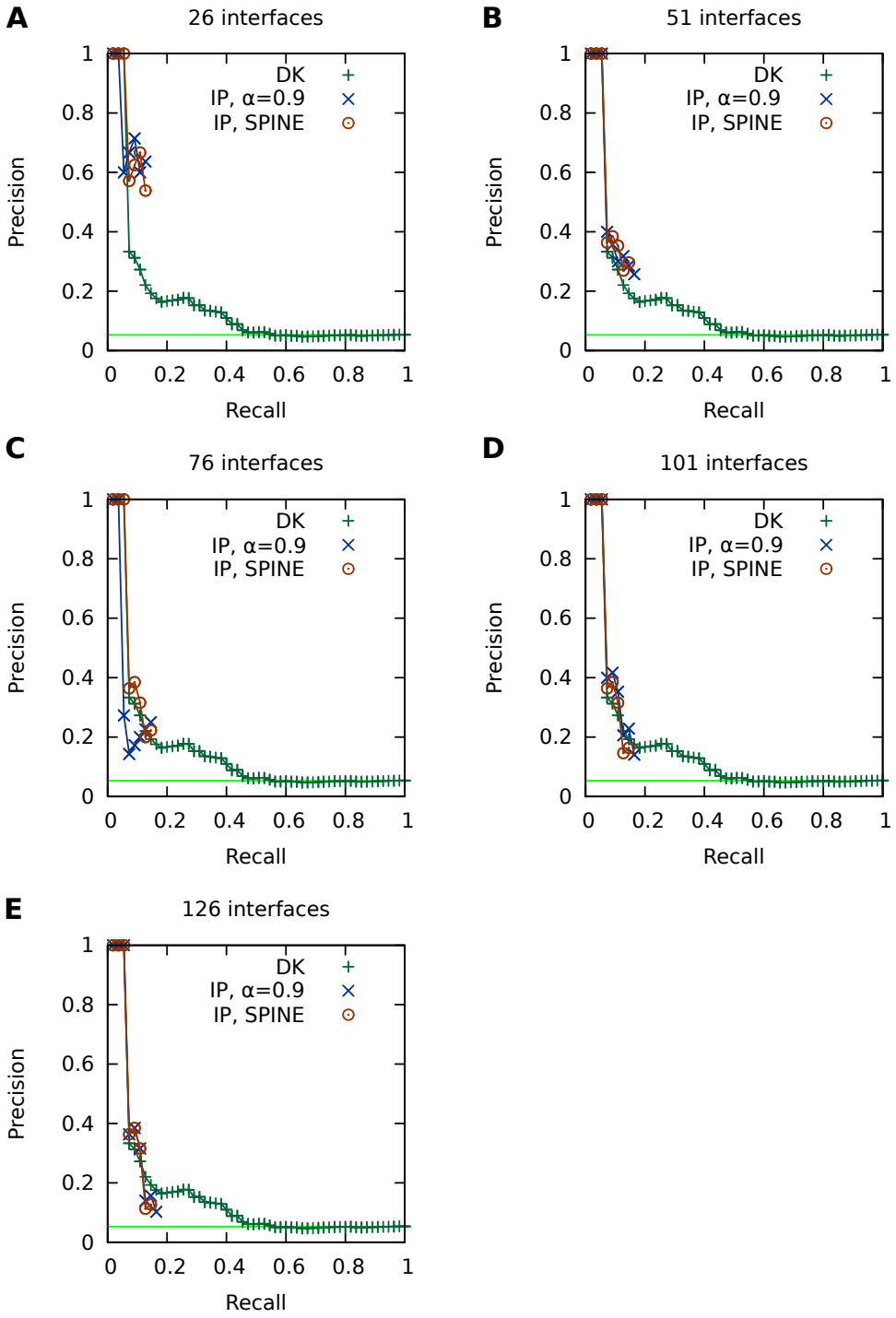



**Figure 3.16:** FHV hit-prediction results for the SPINE phenotype-sign heuristic. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

**Varying parameters and using additional constraints**

We also performed additional experiments to measure the effects of other aspects of our method. We summarize the results here, and provide precision-recall and accuracy-coverage curves in Figures 3.17 to 3.26.

- *Constraining edge signs (Figures 3.17 to 3.20).* We tested our method under two smaller values of the parameter $\alpha$, the proportion of activating edges: 0.7 and 0.8. On the sign-prediction task, setting $\alpha = 0.9$ results in the highest accuracy, and accuracy drops as $\alpha$ decreases. The hit-prediction accuracy of our method, however, appears to be fairly insensitive to the value of this parameter.

- *Prohibiting cycles in the inferred subnetworks (Figures 3.21 to 3.24).* We compared precision-recall curves for both data sets and several values of $\gamma$, both allowing and disallowing cycles in the inferred network. Disallowing cycles does not appear to have a strong or consistent effect on the method's precision. However, our rationale for prohibiting cycles is based on interpretability considerations.

- *Seeding the subnetwork with edges and interfaces from domain knowledge (Figures 3.25 and 3.26).* Seeding the subnetwork with literature-curated edges does not appear to have an effect on BMV hit- and sign-prediction accuracy. However, we believe that doing so is qualitatively useful, as it allows our method to make predictions about what additional hits might be explained by already-studied mechanisms.

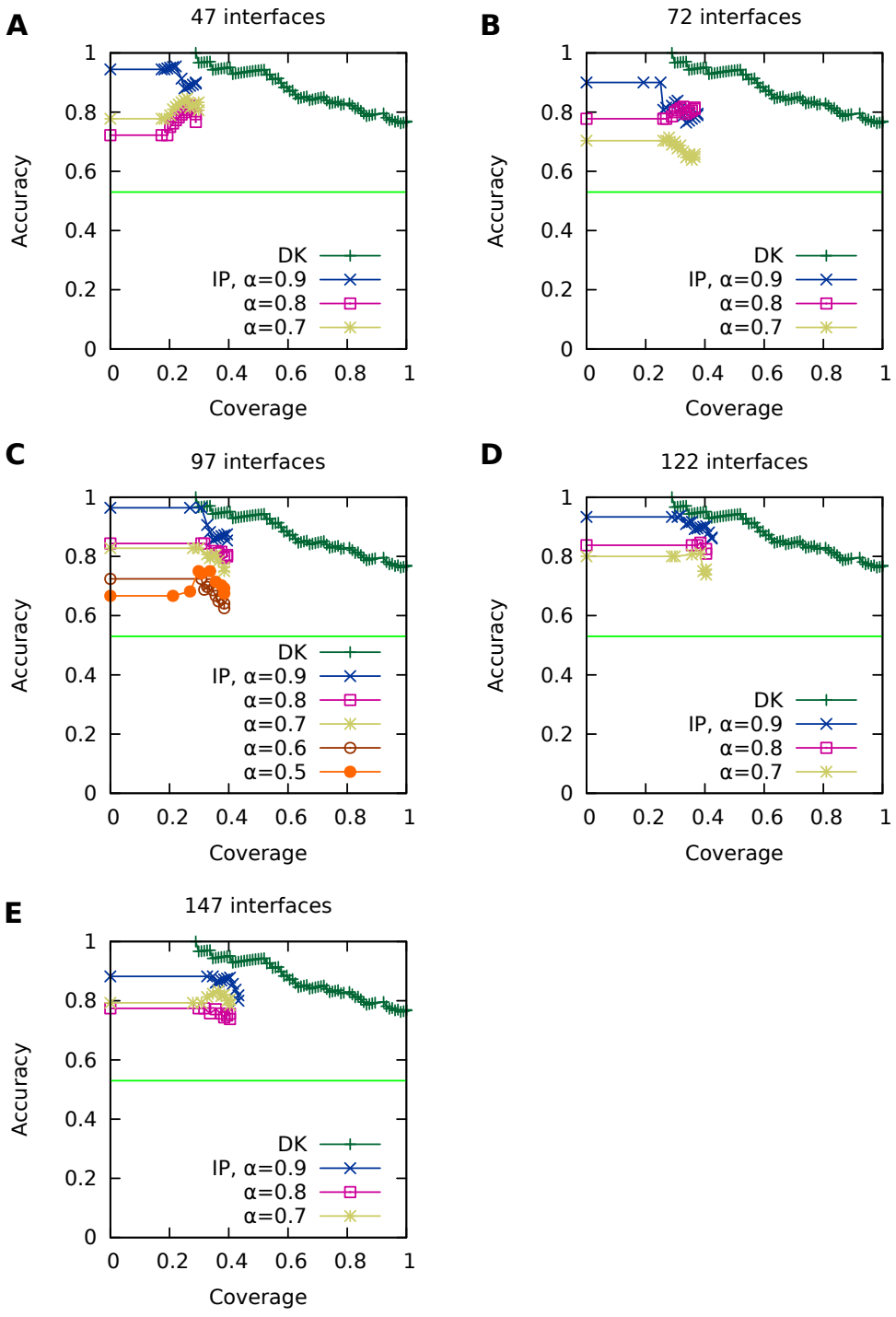



**Figure 3.17:** BMV sign-prediction results for varying $\alpha$, the required minimum fraction of activating edges in the inferred subnetwork. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

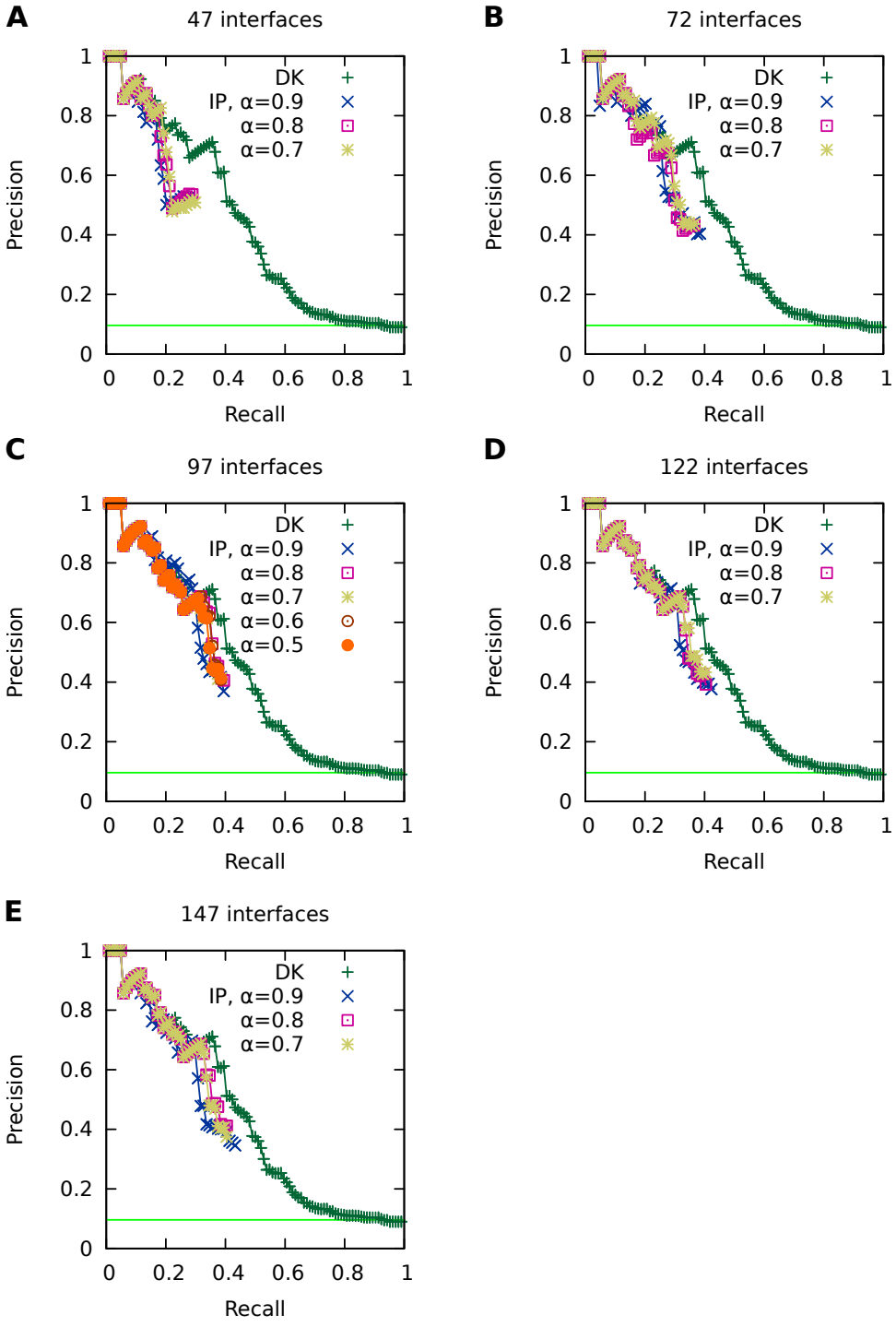



**Figure 3.18:** BMV hit-prediction results for varying $\alpha$, the required minimum fraction of activating edges in the inferred subnetwork. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

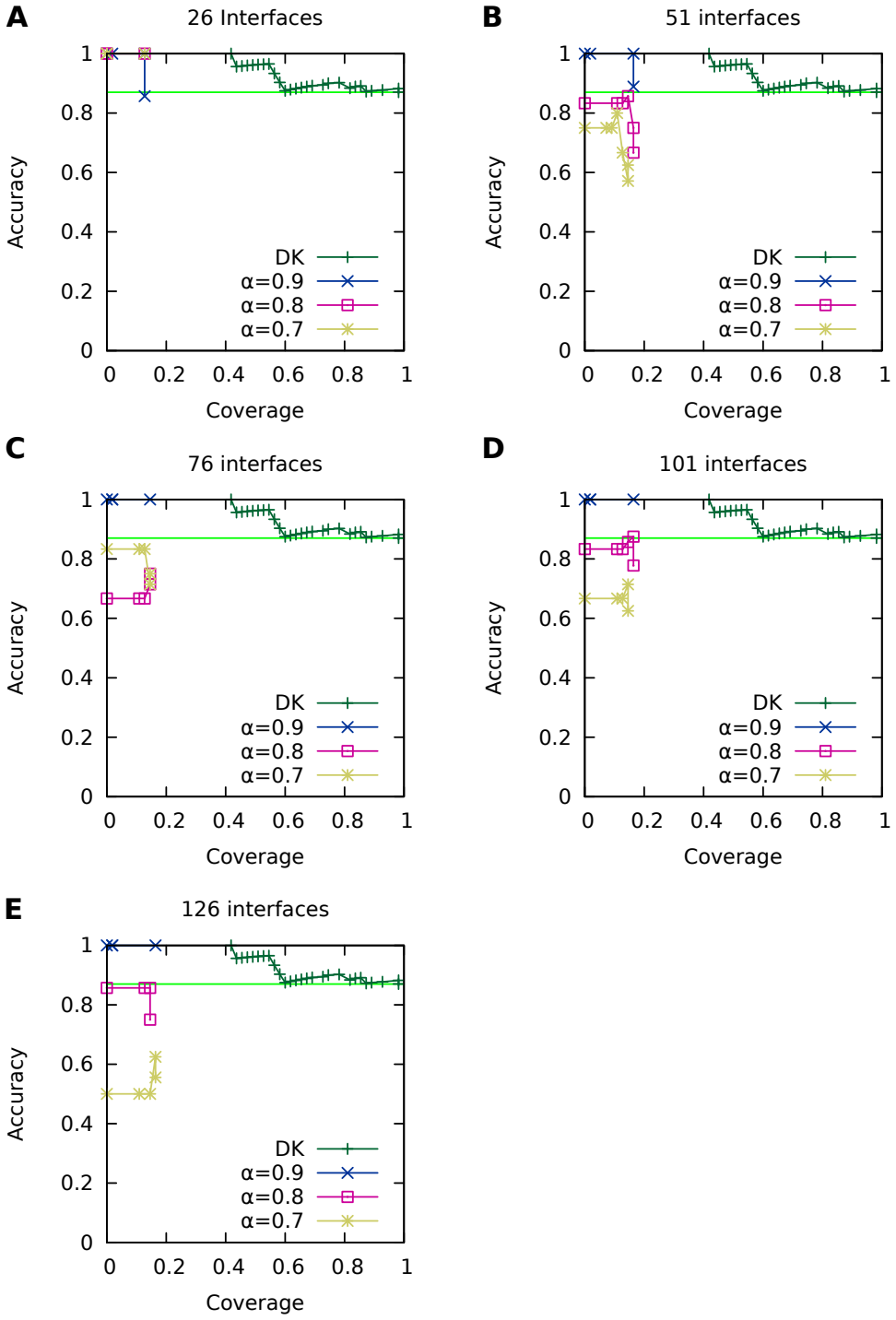



**Figure 3.19:** FHV sign-prediction results for varying $\alpha$, the required minimum fraction of activating edges in the inferred subnetwork. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

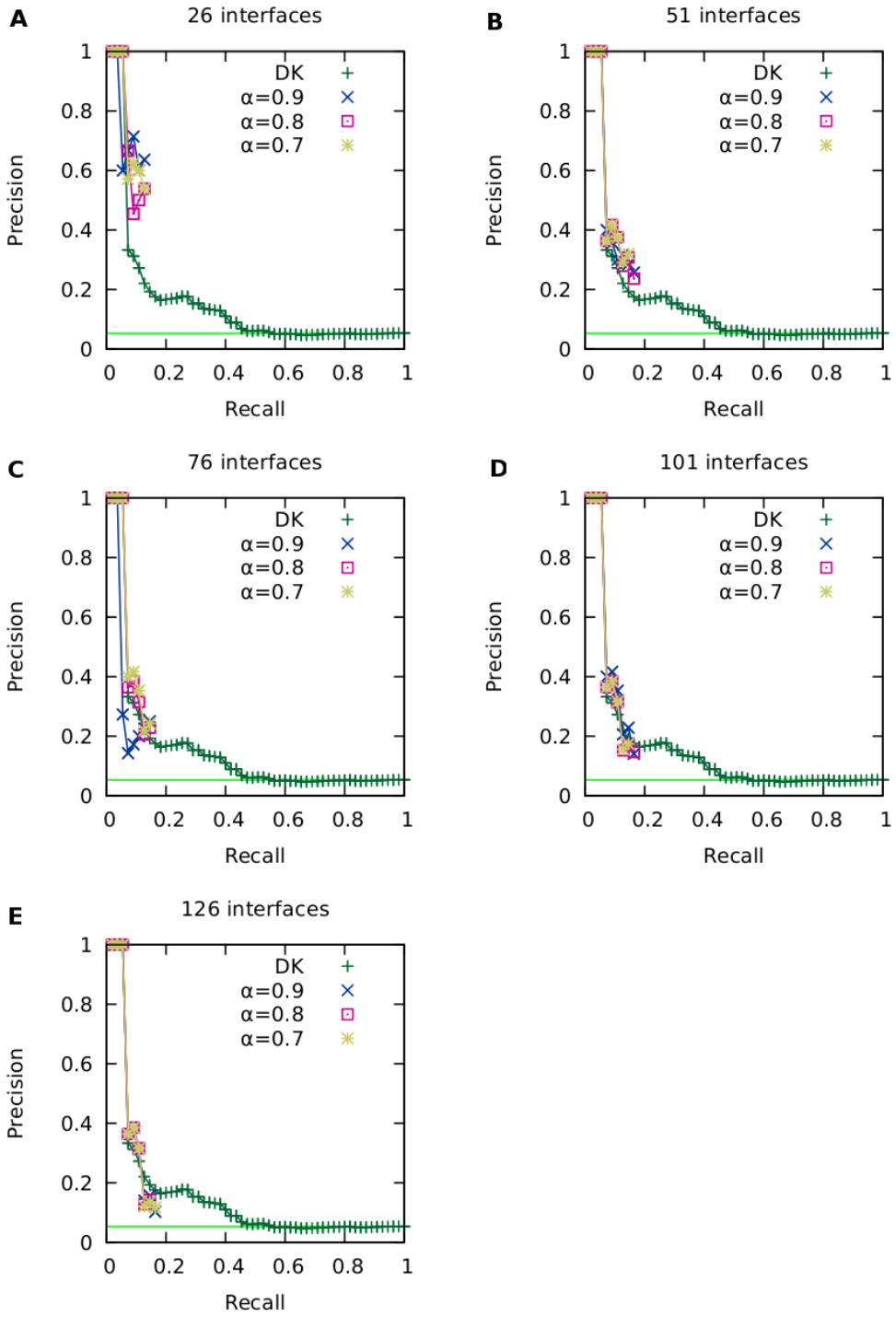



**Figure 3.20:** FHV hit-prediction results for varying $\alpha$, the required minimum fraction of activating edges in the inferred subnetwork. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

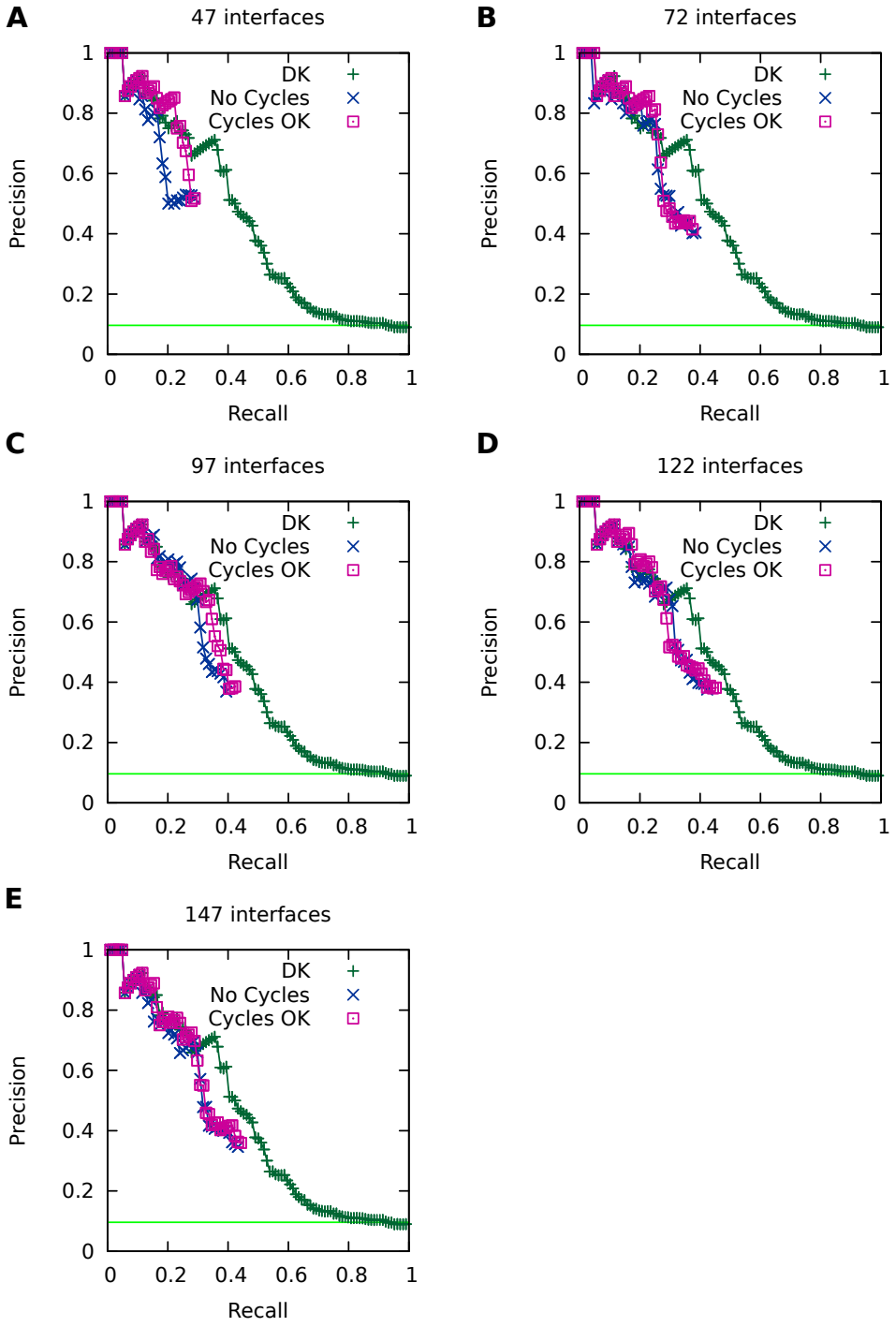



**Figure 3.21:** BMV hit-prediction results assessing accuracy of the cycle-prohibiting constraint. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

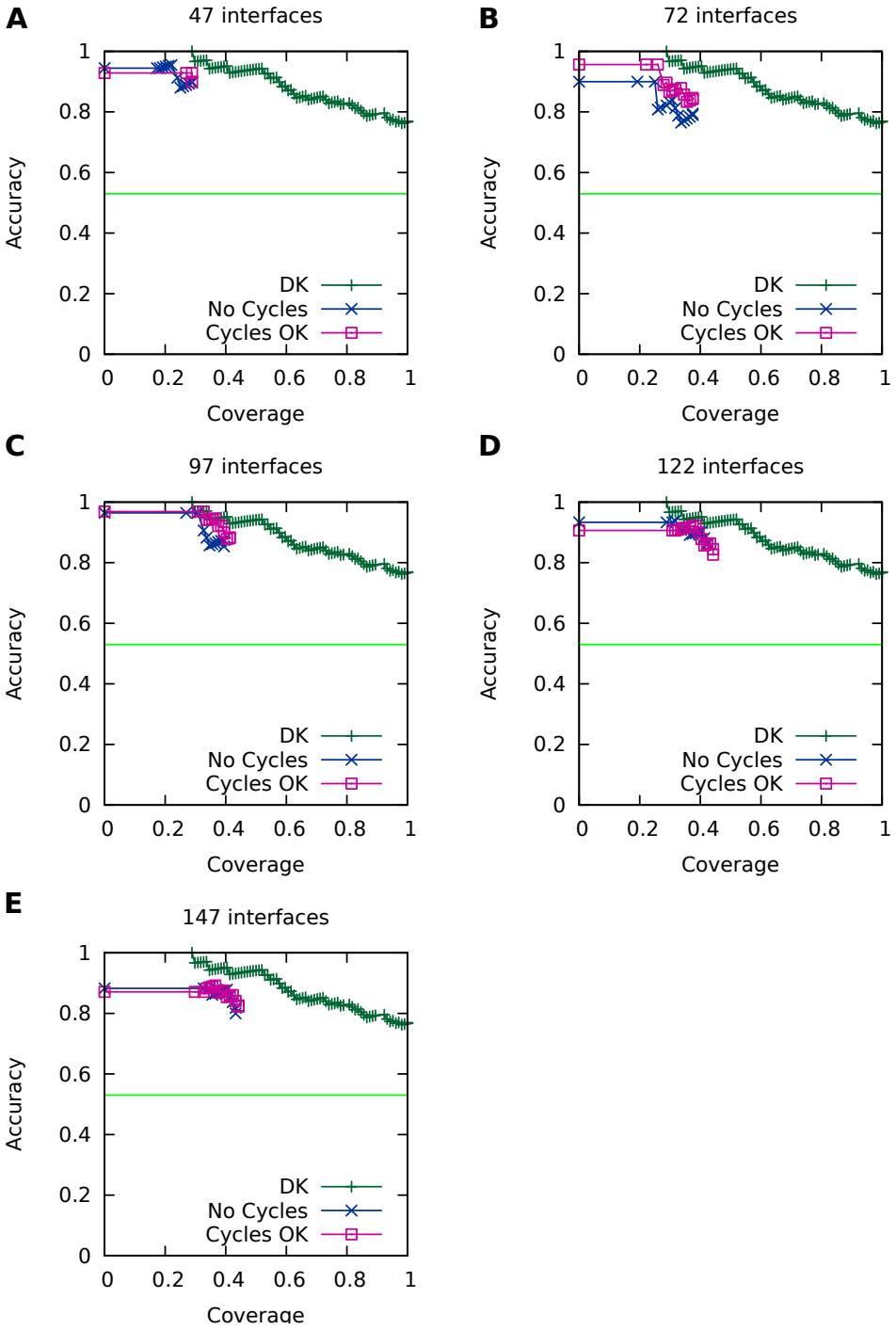



**Figure 3.22:** BMV sign-prediction results assessing accuracy of the cycle-prohibiting constraint. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

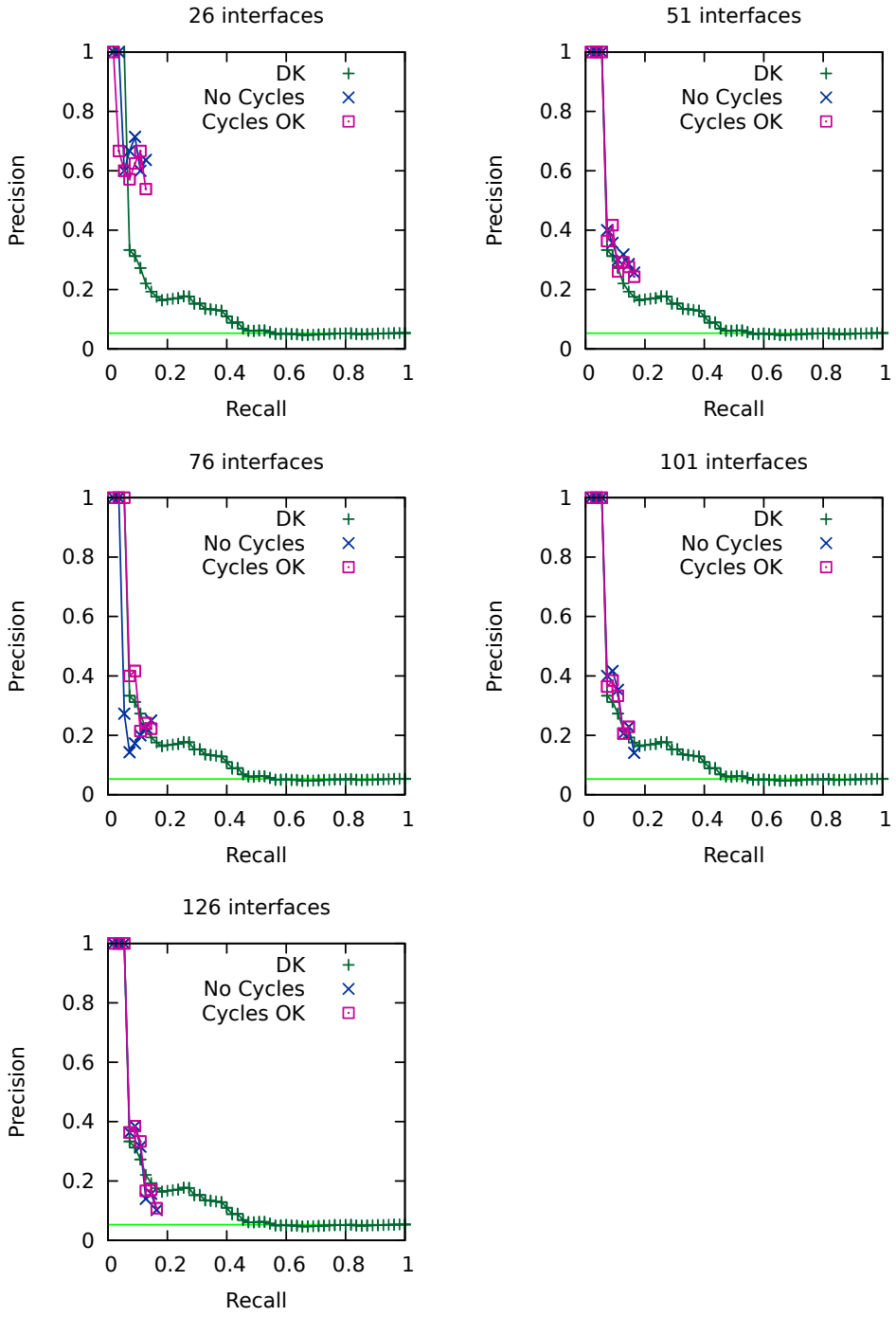



**Figure 3.23:** FHV hit-prediction results assessing accuracy of the cycle-prohibiting constraint. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

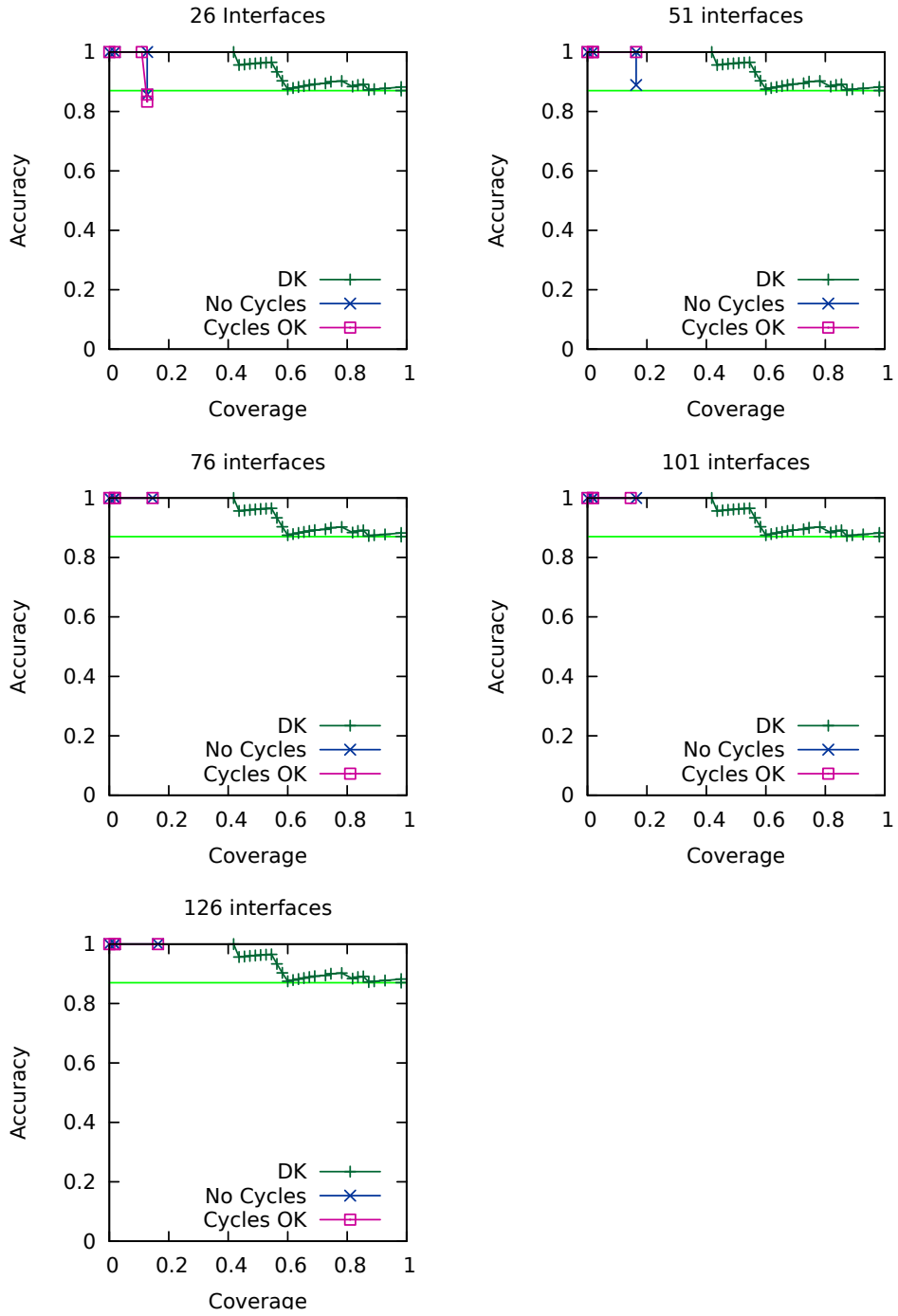



**Figure 3.24:** FHV sign-prediction results assessing accuracy of the cycle-prohibiting constraint. Results are provided at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

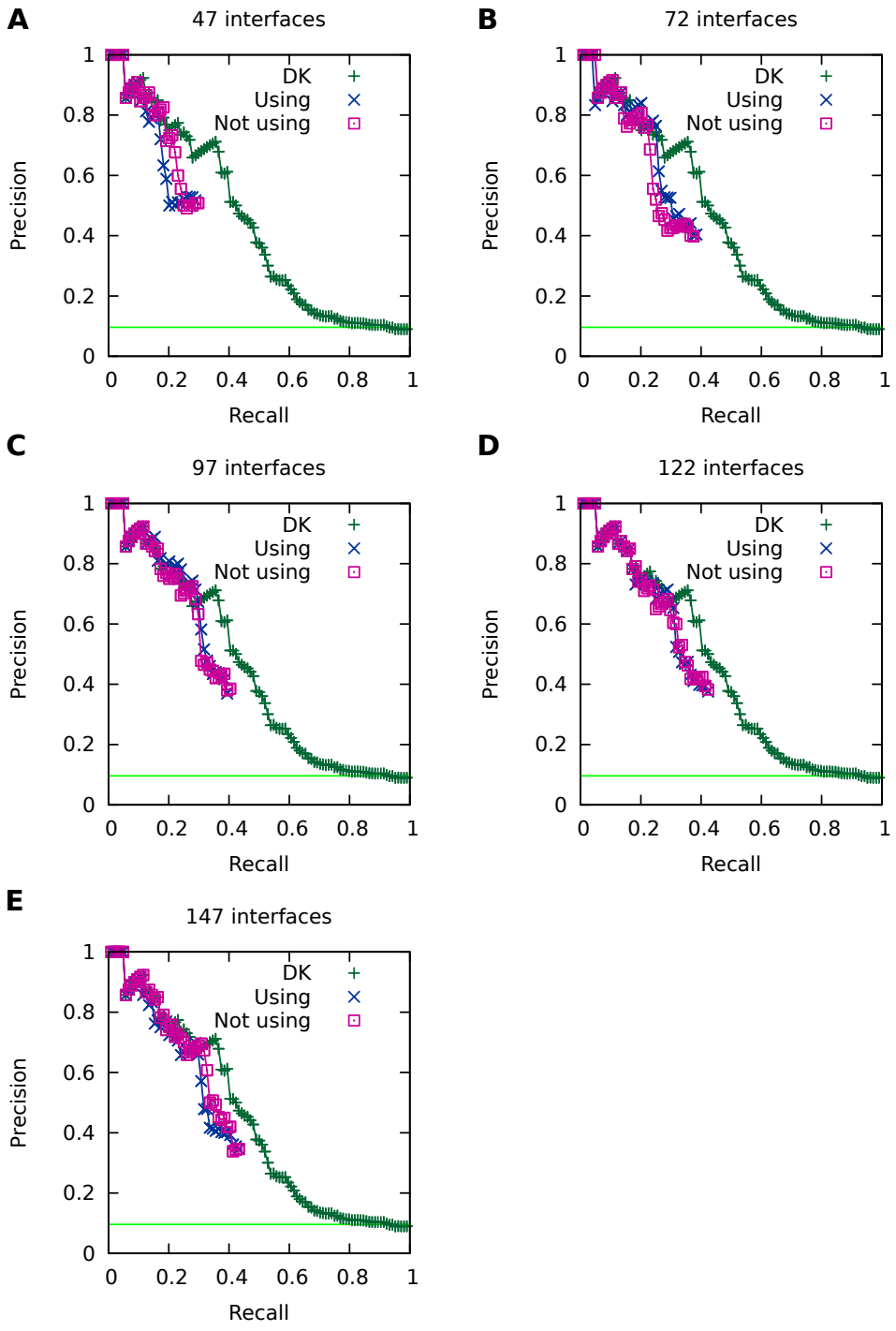



**Figure 3.25:** BMV hit-prediction results assessing effects of seeding the subnetwork with interactions from literature. Results are provided for BMV at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

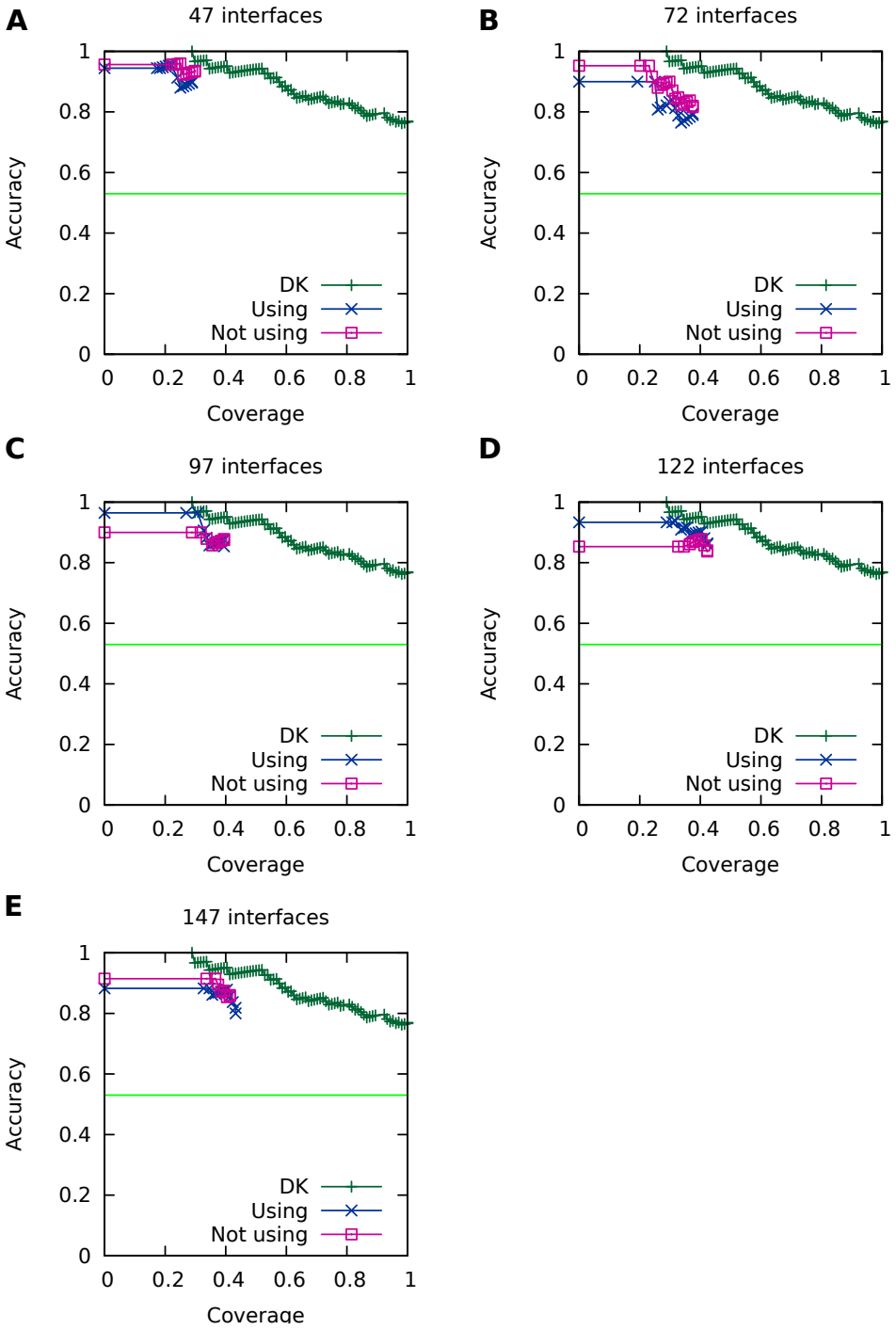



**Figure 3.26:** BMV sign-prediction results assessing effects of seeding the subnetwork with interactions from literature. Results are provided for BMV at all levels of $\gamma$ (the number of interfaces): $\gamma$=26 (**A**), 51 (**B**), 76 (**C**), 101 (**D**), 126 (**E**).

## 3.4 Discussion

In this chapter, we presented an application of our general method for subnetwork inference (Thesis Contribution 1) that aims to elucidate how viruses exploit their host cells (Thesis Contribution 2).

The interpretability of the inferred subnetworks (Thesis Contribution 3) is promoted by the following design decisions, each of which we assessed empirically or by a comparison to the literature.

- Our empirical evaluation demonstrates that, using a gene-prioritization method as a sub-component, our method is able to predict phenotypes for unassayed genes with accuracy that is comparable to the gene-prioritization method alone. However, beyond just prioritizing genes, the inferred subnetworks offer predictions for the mechanism by which suppression of a relevant gene affects viral replication. This entails predicting which host factors are host-virus interfaces, and inferring signed, directed, acyclic paths from each hit to an interface. Furthermore, by inferring an ensemble of subnetworks, rather than a single subnetwork, the approach is able to quantify its certainty about the relevance of various genes and interactions.

- We used our method to predict host-virus interfaces and additional relevant host genes for Brome Mosaic Virus, and performed a literature-based analysis of the predicted relevant host factors. While additional experimentation is necessary to confirm our predictions, a number of them are supported by domain knowledge. Among the predicted interfaces, many are known to bind or modify RNA, localize to the site of viral replication, or act in processes that have been previously identified as involved in viral replication. Similarly, many predicted hits are members of known relevant complexes, and a few are supported by independent experiments. These results are also supported by a Gene Ontology analysis which showed that our inferred subnetworks identify more relevant functional categories than the experimental data alone.

- Our approach uses known host intracellular interactions to infer ensembles of directed subnetworks which provide consistent explanations for phenotypes measured in genome-wide loss-of-function assays. This approach is able to represent a rich set of interaction types, in addition to domain knowledge about specific interactions that are known to be relevant. Among these heterogeneous biological data types are protein complexes. The results of our permutation experiments (Section 3.3.6)

suggest that the inclusion of several protein complexes in the inferred subnetworks, internally or as interfaces, is not merely due to the size or degree of the complexes in the background network.

- We impose constraints on the subnetwork to enforce edge sign consistency and acyclicity, both of which improve the interpretability of the subnetworks. Our studies show that these heuristics either improve or do not negatively effect the accuracy of the inferred subnetworks.
- When an expert-curated subnetwork is used to seed the inferred subnetwork, our inference method can be used to infer connections between existing knowledge and as-yet-unexplained hits. Including these edges does not reduce the accuracy of the inferred subnetworks.
- Our stability experiments demonstrate that the predictions made by our inferred networks have high levels of stability given small changes to the input data.

# 4 Providing interpretable views of inferred human-HIV interaction subnetworks

In this chapter, we extend our subnetwork inference method to infer human-HIV interaction subnetworks. As in Chapter 3, we combine a diffusion kernel method for gene prioritization with an integer linear programming method for subnetwork inference. Our method infers an ensemble of subnetworks that provide a confidence value for the relevance of each network element. We design our method with particular attention to the interpretability of the inferred subnetworks. Our contributions that are motivated by this goal include new representations of protein complexes and metabolic reactions in the background network and IP constraints, and a method for identifying customized views into the inferred subnetworks using sets of query genes. The inferred subnetworks achieve similar accuracy to the diffusion kernel method when tested on held-aside input data as well as externally derived relevant gene sets. They predict the relevance of some protein complexes, including some that were also identified in our study of Brome Mosaic Virus' interactions with yeast. The work in this chapter demonstrates that our method, originally developed to study the comparatively simple yeast system, can be extended to study human-relevant viruses in mammalian systems.

**The work in this chapter was performed in collaboration with the following co-authors:**

Eunju Park, Paul Ahlquist, Mark Craven.

## 4.1 Introduction

In this chapter, we discuss our efforts to extend our IP approach to infer interactions between Human Immunodeficiency Virus 1 (HIV) and a human cell host.

Because HIV is more widely studied than either BMV or FHV, this new setting provides a greater diversity and volume of the available experimental data. HIV is one of an increasing number of human viruses that have been studied using host-genome-wide RNA interference (RNAi) assays, which, similar to the yeast gene suppression experiments we studied in the previous chapter, are used to identify individual host genes whose suppression affects viral replication. Whereas for yeast we have to infer the host-virus interfaces, for HIV, instead, the interfaces are provided by catalogs of direct, physical interactions between human and HIV proteins. Other data sources, such as measurements of protein expression in HIV-infected cells, provide us lists of HIV-relevant genes that we can use to evaluate our inferred subnetworks. Many other databases provide human protein-protein interactions and other, richer data sources that describe protein complexes and biological pathways.

The transition from yeast to mammalian cells also brings new challenges. The background network is nearly an order of magnitude larger for humans than for yeast, and yet is known to be highly incomplete. Furthermore, the majority of the interactions are undirected. While data from several independent RNAi screens are available, the hit sets have very little overlap, suggesting that many truly relevant genes still remain to be identified.

Figure 4.1 provides an illustration of the subnetwork inference approach that we propose in this chapter. The inputs (Figure 4.1A and B) are a set of phenotype labels for host genes identified from RNAi assays, a set of host-virus protein-protein interactions, and a host background network. In contrast to Chapter 3, we have fewer categories of phenotype labels. As before, we populate our background network using a variety of data types, which include binary protein-protein interactions as well as protein complexes and reactions.

Our method produces as output an inferred subnetwork (Figure 4.1C) that is composed of paths, each of which is a linear chain of interactions that begins with a hit and ends with a direct interaction between a host interface protein and a viral protein. The subnetwork attempts to include all of the hits (B, E, L, K) and interfaces (L, N), and in doing so predicts a small number of additional hits (H, I) from among the host genes in the background network. The paths may also include protein complexes, reactions, and small molecules (here, reaction M and molecule F). In contrast to the previous chapter, the experimental data for HIV and human does not support predicting signs or directions on edges. Because

the majority of hits have the same phenotype label (84%, HDF), we also do not infer specific phenotype labels for the predicted hits.

As in the previous chapter, our method returns an ensemble of subnetworks, which allows us to assign a confidence value to every predicted subnetwork element. However, we propose a new method for generating the ensemble. To infer each subnetwork in the ensemble, we use a large subsample of the input experimental data, rather than the complete data. We are motivated by two hypotheses. First, that this approach will minimize the effect of noisy input data. Second, that the reduction in input data will make the resulting IPs easier to solve.

We find that the inferred subnetworks are very large and densely connected. To help biologists explore the subnetworks and generate hypotheses, we offer a method for identifying specialized views into the subnetworks. Given a query gene set (derived either by hand, or by using a literature search tool such as GADGET (Craven & Ziegler, 2011)), a view offers predictions about which inferred paths are most specifically relevant to the query genes.

We demonstrate that our subnetwork inference method is able to accurately predict held-aside data and known relevant human gene sets. Importantly, this work serves as a proof of concept that it is possible to extend our method from the relatively simple yeast domain to the more complex realm of mammalian cells, paving the way for future efforts to assist in the study of human-relevant viruses.

### 4.1.1 Related work

As it is an extension of the approach we describe in Chapter 3, the work in this chapter is also particularly related to methods for *subnetwork inference* (Chapter 2.2) and *subnetwork extraction* (Chapter 2.3). In particular, we note that Gitter *et al.* (2013) apply a subnetwork orientation method to the task of inferring human signaling pathways in response to influenza infection. As before, our method focuses on explaining the effect of RNAi hits on viral replication, rather than the effect of viral infection on the human transcriptional response.

Again, we use a diffusion kernel method for *candidate gene prioritization* (Chapter 2.4) as one part of the approach. A similar approach was previously proposed by Murali *et al.* (2011) for predicting HIV-relevant human genes from RNAi data and a background network. However, their method does not infer subnetworks.

We also propose a method for extracting context-specific components from the inferred subnetwork based on an query gene set. Our approach differs from *network filtering and*

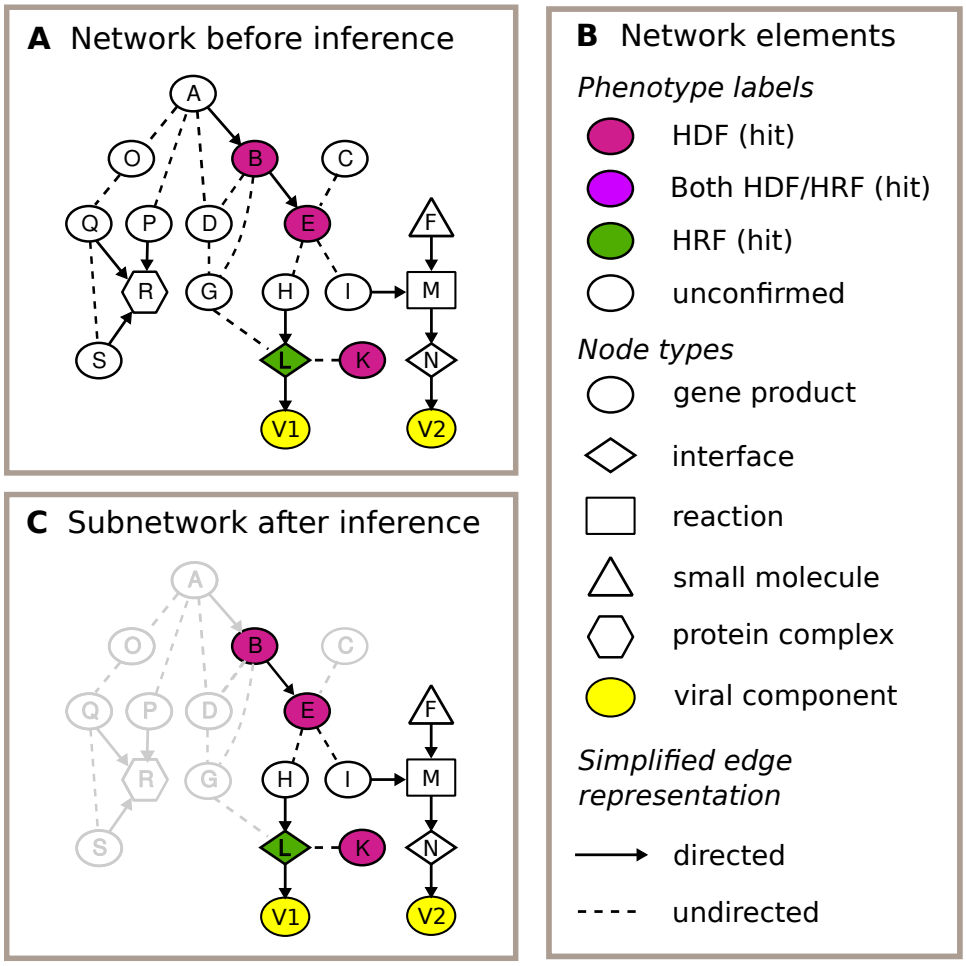



**Figure 4.1:** An illustration of the input and output to the human-HIV subnetwork inference method.
**A** Background network and observed phenotypes, before inference.
**B** Network elements, including phenotype labels, node types, and the simplified edge types. The phenotype label HDF means “HIV dependency factor", and is given to genes whose suppression inhibits HIV replication; it is analogous to the **down** label from the previous chapter. HRF is “HIV restriction factor", analogous to **up**.
**C** A subnetwork inferred from the input in panel **A**. Each hit is connected to a viral component via a known host-virus interface. A number of additional host factors are also predicted to be relevant. The edges that are predicted to be irrelevant are shown in grey.

*integration* (Chapter 2.6) in that we extract the context-specific components from the inferred subnetwork, rather than the background network. Filtering methods generally try to reduce the presence of nodes that are reported as relevant to many conditions, or that have very high degree. Similarly, we use a ranking function that uses the inferred subnetwork's path structure to predict which genes are most specifically related to the query gene set.

## 4.2 Materials and methods

### 4.2.1 Data

We represent human and HIV gene products using Entrez Gene identifiers, which in some cases we have mapped from more specific protein identifiers.

#### Host factors identified from RNAi screens

Several independent studies have used RNAi technology to identify human genes that are involved in HIV replication, either facilitating it (called **HIV dependency factors**, or HDFs) or inhibiting it (called **HIV restriction factors**, or HRFs). The intersection among the hit sets identified by each screen is quite small. However, when RNAi hit sets are viewed at the level of enriched pathways and functional categories, a much higher overlap is observed, suggesting that each screen captures only a small number of the true hits (Hao *et al.*, 2013). Therefore, we assemble a master hit set by pooling the hit sets reported in five publications. Our decision to pool the hits is also supported by an HDF prioritization study by Murali *et al.* (2011), who observe that using a three-study, pooled hit set as input to their network-based prioritization method achieves increased accuracy in comparison to using the three hit sets independently.

The hit data sets are summarized in Table 4.1. We note a few differences in this setting as compared to Chapter 3. Whereas in Chapter 3 we had access to complete screen data including weak fold-changes and were therefore able to assign five different phenotype labels to the genes in the background network, for these studies, we have access only to the hit sets from the RNAi screen and/or hits validated by subsequent screens. One study (Liu *et al.*, 2011) provides both primary RNAi screen hits as well as hits validated by a secondary screen. In this case, we use the primary RNAi hit set. Furthermore, several screens used criteria that could only identify hits of one particular sign; that is, only HDF or HRF. A small number of hits were given different phenotype labels from separate RNAi pools within the same study or between studies, suggesting that whether each hit assists or inhibit viral replication depends on additional factors such as cell type.

Therefore, we are motivated to use a reduced set of phenotype labels and make fewer types of predictions about nodes in the inferred subnetwork. Because of the high prevalence of HDFs in the pooled data (84%), and the presence of hits that are labeled both HDF and HRF, we do not attempt to predict phenotype signs. Because we know that each screen only identifies a subset of true hits, we do not rule out any of the human genes that were not identified by the screens in which they were tested. We apply one of two possible phenotype labels to the background network genes: **hit** (applied to HDFs, HRFs, and those receiving both labels) and **unconfirmed** (applied to all other genes).

In total, the background network contains 1,006 hits.

### Human-HIV interfaces

We compile a set of human-HIV interface proteins from interaction data provided by multiple studies and databases (Table 4.2). All of these data are provided as direct, physical interactions; each is between a human gene product and a specific HIV component. One of our sources, the NCBI Human, HIV-1 Protein Interaction Database, also contains indirect interactions, including genetic interactions. From this source, we use only the direct physical interactions as input to the method, and use the indirect interactions for evaluation. In total, the background network contains 1,693 interfaces, 195 of which are also RNAi hits.

### Background network

We assemble our background network from public databases that record protein-protein interactions, protein complexes, and reactions (Table 4.3). It is represented as a partially-directed graph. Each protein-protein interaction is represented as an edge in the graph, with directed interactions (such as kinase-substrate) represented as directed edges.

We draw both protein complexes and reactions from the Reactome collection of curated pathways (Croft *et al.*, 2014). Protein complexes are represented as separate nodes, with directed edges linking constituent genes to the complex node (Figure 4.2A). The reactions in Reactome include biochemical reactions as well as other types of biological events. We represent the reactions as separate nodes, with directed edges coming in from their inputs and catalysts, and directed edges going out to their products (Figure 4.2B). The inputs and outputs to the reactions may be molecules, gene products, or protein complexes, and the catalysts may be gene products or protein complexes.

The background network contains a total of 224,522 edges among 32,991 nodes, which include 10 HIV genes, 16,662 human genes, 6,464 human protein complexes, 5,567 reactions, and 4,288 small molecules.

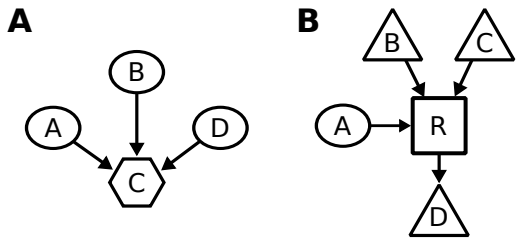



**Figure 4.2:** Representation of protein complexes and reactions in the background network.
**A** A complex is represented as a separate node (C). We add incoming directed edges from its constituent protein components (A,B,D).
**B** A reaction is represented as a separate node (R). We add incoming directed edges from its catalyst (A) and inputs (B,C), and outgoing edges to its output(s) (D).

**Table 4.1:** HIV-relevant hits from RNAi screens, divided into HIV dependency factors ('HDF'; RNAi suppresses viral replication), HIV restriction factors ('HRF'; RNAi enhances viral replication), and genes for which different siRNA pools for the same gene either suppress or enhance viral replication ('Both'). Screens marked with $^{D}$ ($^{R}$) use criteria that exclusively allow the discovery of dependency (restriction) factors. The quantities in this table include only hits that are represented in the background network.

| | Phenotype labels | | |
|---|---|---|---|
| Screen | HDF | HRF | Both |
| Brass *et al.* (2008)$^{D}$ | 241 | – | – |
| König *et al.* (2008)$^{D}$ | 271 | – | – |
| Zhou *et al.* (2008) | 188 | 1 | 19 |
| Yeung *et al.* (2009)$^{D}$ | 183 | – | – |
| Liu *et al.* (2011)$^{R}$ | – | 150 | – |

**Table 4.2:** Interfaces from human-HIV protein interaction databases.

| Source | Human genes | Interactions | Note |
|---|---|---|---|
| Jäger *et al.* (2012) | 381 | 443 | Retrieved from BioGRID, 02/2014 |
| BioGRID (Stark *et al.*, 2006) | | | Downloaded 02/2013 |
| PHISTO (Tekir *et al.*, 2013) | 981 | 1,286 | Downloaded 05/2013 |
| NCBI Human, HIV-1 protein interaction database (Fu *et al.*, 2009) | 1,305 | 1,646 | Direct interactions obtained from Roger Ptak (personal communication) |

**Table 4.3:** Intracellular interactions in the background network.

| Interaction | Source | Dir. | Count | Notes |
|---|---|---|---|---|
| Complex membership | Reactome (Croft *et al.*, 2014) | Y | 41,912 | Downloaded 05/2014 |
| Kinase-substrate | Newman *et al.* (2013) | Y | 1,797 | |
| Biochemical reactions and other events | Reactome (Croft *et al.*, 2014) | Y | 25,632 | Normal human pathways (omitting disease pathways); downloaded 03/2014 |
| Post-translational modification | BioGRID (Stark *et al.*, 2006) | Y | 3,905 | Downloaded 02/2013 |
| Protein-protein | BioGRID (Stark *et al.*, 2006) | N | 103,453 | Downloaded 02/2013 |
| Protein-protein | HIPPIE (Schaefer *et al.*, 2012) | N | 122,229 | Downloaded 02/2013 |
| Protein-protein | HPRD (Prasad *et al.*, 2009) | N | 39,132 | Release 9, 04/2010 |

**Supplementary interactions from literature**

The ESCRT (endosomal sorting complexes required for transport) complexes are involved in several processes involving membrane budding. HIV and other viruses are known to take advantage of ESCRT functions in order to facilitate the release of new virions from the host cell. Because this process is well-studied, we determined that it would make a useful case study for testing our method's capability for predicting useful connections between known processes and relevant genes whose role in HIV is yet undetermined. We therefore supplemented the background network with 26 interactions representing the relationship between the ESCRT pathway and HIV budding (Adell & Teis, 2011; Sundquist & Kräusslich, 2012).

### 4.2.2 Subnetwork inference

The output of our method is an ensemble of subnetworks, each of which explains the HIV-relevant hits by providing paths that connect them to host-virus interfaces. We encode our subnetwork desiderata using an integer linear program, and infer subnetworks by solving the program. Each subnetwork must:

- Provide at least one path from each hit to an HIV component
- Include nearly all available host-virus interfaces
- Predict a limited number of additional hits from among the unconfirmed human gene products

To accomplish the above, a subnetwork may do the following, in addition to using the subnetwork representation described in the previous chapter:

- Predict the inclusion of protein complexes that are overrepresented by known HIV-relevant human genes
- Predict the relevance of specific reactions from biological pathways

In Chapter 3, each subnetwork in the ensemble is inferred from the same input data, and represents a different optimal solution to the IP. In this chapter, we use a different procedure to generate an ensemble. For each subnetwork, we randomly hold aside a small fraction of the hits and interfaces, and use the IP to infer a single optimal solution for that input data. We hypothesize that this procedure mitigates the effect of noisy input data. Truly relevant nodes and edges should be confidently identified even when the input data

is slightly perturbed. Additionally, we propose that the reduction in the size of the input data considered for each subnetwork will make the IP more amenable to the solver.

The details of our inference method are as follows.

**Step 1: Assemble background network and HIV-relevant data.**

The inputs to our method are hits identified as HIV-relevant by RNAi assays (Table 4.1), human-HIV interfaces (Table 4.2), and a background network consisting of protein-protein and other direct interactions between human gene products and small molecules (Table 4.3).

**Step 2: Generate an ensemble of input data sets, candidate paths, and diffusion kernel scores.**

We proceed to create an ensemble of $n$ random subsamples from the HIV-relevant data sets, which are used as input to the IP in Step 3. In our experiments, we use $n$=100.

**For 1 to $n$:**

1. *Sample input hits and interfaces.* Create a subset of input data by randomly holding aside 25% of the hit and interface sets.

2. *Enumerate candidate paths for each hit in the sample.* Each path begins with a hit and terminates in an interaction between an interface and an HIV node. The depth of the search is limited for tractability; for our experiments, we search for paths that include up to three host nodes and an HIV node.

   Reactions are incorporated differently from the other interactions. When a reaction is included in a candidate path, all of the reaction's other inputs and catalysts are also included. When the length of the path is assessed, the entire reaction (inputs, reaction, and output) counts as only one interaction.

3. *Prioritize unconfirmed human genes using a diffusion kernel.* We use the regularized Laplacian kernel (Section 3.2.2) to calculate scores that represent the proximity and connectivity of each unconfirmed gene to the set of known HIV-relevant human genes. In this case, the HIV-relevant gene set includes *both* the sample's hits and interfaces. We calculate the kernel using a restricted version of the background network that only includes interactions between human gene products. We exclude protein complexes and reactions from the kernel and scoring because we wish to predict their relevance based on their inferred node content. As high-degree nodes, they may be likely to

**Table 4.4:** Integer program variables. Binary variables represent the status of nodes, edges, and paths in the network.

| Network elements | Variable | Interpretation | Values |
|---|---|---|---|
| Paths $p$ | $\sigma_p$ | Relevant | no=0, yes=1 |
| Edges $e$ | $x_e$ | Relevant | no=0, yes=1 |
| Nodes $n$ | $y_n$ | Relevant | no=0, yes=1 |

receive spuriously high kernel scores because of their placement in the background network.

**Step 3: Infer an ensemble of subnetworks.**

For each of the sampled input data sets, we separately infer a subnetwork by optimizing the integer linear program that is discussed in the following section. Taken together, the subnetworks form an ensemble that can be used to provide a confidence value for the relevance of each node, interaction, and path.

In our experiments, we model the IP using GAMS 23.9.3 (GAMS Development Corporation, 2010), and use the IBM ILOG CPLEX 12.4.0.1 (IBM, 2012) MIP solver for optimization.

**IP variables and notation**

The input is represented as sets of nodes $\mathcal{N}$, edges $\mathcal{E}$, and candidate paths $\mathcal{P}$. $\mathcal{E}(p)$ and $\mathcal{N}(p)$ refer to the edges and nodes in a particular candidate path $p$, $\mathcal{N}(e)$ refers to the nodes in a particular edge $e$, and $\mathcal{E}(n)$ refers to the edges that touch a particular node $n$. We denote an edge between nodes $n_i$ and $n_j$ as $(n_i, n_j)$.

We define various subsets of the nodes and edges using experimental data. From $\mathcal{N}$, $\mathcal{N}^H \subseteq \mathcal{N}$ is the set of hits, $\mathcal{N}^I$ is the set of interfaces derived from host-virus protein interaction data, $\mathcal{N}^U$ is the set of non-hit, non-interface human gene products (*u*nconfirmed), $\mathcal{N}^C$ is the set of protein complexes, $\mathcal{N}^R$ is the set of reactions, and $\mathcal{N}^V$ is the set of viral components. Among edges $\mathcal{E}$, $\mathcal{E}^X$ contains e*x*ternal edges between the host and virus, and $\mathcal{E}^I$ are edges that are *i*nternal to the host.

We use binary variables to represent the presence or absence of each node, edge, and path in an inferred subnetwork (summarized in Table 4.4). In this setting, we do not encode phenotype signs, edge signs, or edge directions as we did in the last chapter.

**Global objective function and constraint**

We know that the set of hits is incomplete, and would like to predict which other host genes are involved. However, we need a way to limit the size of the inferred subnetwork so as to not include everything. In this IP, we control the size of the inferred subnetwork by putting a limit, $\delta$, on the number of predicted hits that can be included in the subnetwork, and use an objective function to include those that maximize a predicted relevance score, provided by a diffusion kernel. Next, we apply a second objective function to maximize the number of paths among the confirmed and predicted relevant genes. The purpose of this second objective function is to ensure that the inferred subnetwork contains all possible edges between relevant nodes, rather than an arbitrary selection.

The following steps explain our optimization procedure in detail.

*Step 1: Predict a limited number of unconfirmed nodes that are proximal and connected to hits.* In a constraint, we limit the number of unconfirmed genes that can be predicted to be relevant. This constraint considers only unconfirmed human genes ($\mathcal{N}^U$); protein complexes and reactions are predicted based on their node content according to specific constraints. In our experiments, we vary the number of additional genes to predict.

$$\sum_{n \in \mathcal{N}^O} y_n \leq \delta$$

The first objective function is analogous to the objective function from the IP in Chapter 3, and we use it to predict the relevance of additional human gene products that are useful for providing paths that connect hits to interfaces. We optimize for predicting hits with high diffusion kernel scores (represented with the function score$(\cdot)$).

$$\max \sum_{n \in \mathcal{N}^U} \text{score}(n)\ y_n$$

*Step 2: Identify relevant nodes.* After solving this objective function, we have identified a set of nodes that are predicted to be relevant. We set each relevance variable $y_n$ to its inferred value $\hat{y_n}$.

$$y_n = \hat{y_n}$$

*Step 3: Maximize paths among relevant nodes.* The second objective function maximizes the number of predicted relevant paths among the nodes identified in the previous step. Doing

so does not change the node content, but ensures that the inferred subnetwork represents all possible relevant edges among the relevant nodes.

$$\max \sum_{p \in \mathcal{P}} \sigma_p$$

**IP constraints**

The objective functions and the size-limiting parameter control the size of the inferred subnetwork. The following constraints describe how nodes and edges are incorporated into paths, including special constraints to incorporate different data types.

*Nearly all hits and interfaces must be included in the inferred subnetwork.* Depending on the size of $\delta$, it may not be possible for all hits and interfaces to be accounted for by the inferred subnetwork. Therefore, we allow a small fraction $\epsilon$ of hits and interfaces to be left out. In our experiments, we use $\epsilon = 0.05$.

$$\sum_{n \in \mathcal{N}^H} y_n \geq (1 - \epsilon)|\mathcal{N}^H|$$
$$\sum_{n \in \mathcal{N}^I} y_n \geq (1 - \epsilon)|\mathcal{N}^I|$$

*All edges in a relevant path are relevant.* A relevant edge $e$ (where $x_e = 1$) must be in at least one relevant path (for which $\sigma_p = 1$); we refer to all paths $p$ for an edge $e$ as $\mathcal{P}(e)$. For a relevant path $p$, all of its edges $\mathcal{E}(p)$ must be relevant.

$$\forall e \in \mathcal{E} \qquad x_e \leq \sum_{p \in \mathcal{P}(e)} \sigma_p$$
$$\forall p \in \mathcal{P}, e \in \mathcal{E}(p) \qquad \sigma_p \leq x_e$$

*All nodes in a relevant edge are relevant.* A node $n$ is relevant (that is, $y_n = 1$) if one if its edges $e \in \mathcal{E}(n)$ is relevant ($x_e = 1$). For a relevant edge $e$, both of its nodes $\mathcal{N}(e)$ are also relevant.

$$\forall n \in \mathcal{N} \qquad y_n \leq \sum_{e \in \mathcal{E}(n)} x_e$$
$$\forall e \in \mathcal{E}, n \in \mathcal{N}(e) \qquad x_e \leq y_n$$

*To be deemed relevant, a protein complex must include over-representation of relevant nodes.* We use this constraint to discourage the change inclusion of large, highly-connected complexes. For a protein complex to be predicted to be relevant, the fraction of known or predicted relevant genes among its members should be relatively high. We provide the required relevant fraction, $0 \geq \beta \leq 1$, as a parameter to the method. In the following inequality, we use $\mathcal{N}(c)$ to refer to the set of genes that form a complex $c$.

$$\forall c \in \mathcal{N}^C, \;\; \beta \leq 1 \qquad\qquad \sum_{n \in \mathcal{N}(c)} y_n + (1 - y_c)|\mathcal{N}(c)| \geq \beta|\mathcal{N}(c)|$$

In our experiments, we choose $\beta$ to be two times higher than the fraction of known and predicted relevant genes in the entire human genome. Because we restrict the number of predicted hits to $\delta$, and by assuming that the genome contains 20,000 genes, we set $\beta$ as follows:

$$\beta = 2\frac{|\mathcal{N}^H \cup \mathcal{N}^I| + \delta}{20,000}$$

Many members of protein complexes may not have been included in candidate paths. Often, the complex representation constraint would preclude many of those complexes from being included in the inferred subnetwork. For edges between proteins and complexes that are not in candidate paths, we provide another constraint that allows the edge (and consequently the protein) to be included in the inferred subnetwork only if the complex is.

$$\forall c \in \mathcal{N}^C, \forall e = (n, c) \in \mathcal{E} \qquad\qquad x_e \leq y_c$$

*For a reaction to be relevant, all of its inputs and catalysts must be relevant.* We use this constraint to encode reactions as 'and' functions. It is assumed that in order for the reaction to proceed, all of the inputs and catalysts must be present in the inferred subnetworks. This constraint also ensures that all relevant components of a reaction are shown in the inferred subnetwork. If a reaction has multiple outputs, only the relevant ones will be included due to the structure of the candidate paths. We use $\mathcal{N}(r)$ to refer to all of the inputs and catalysts of a reaction $r \in \mathcal{N}^R$.

$$\forall r \in \mathcal{N}^R, \forall n \in \mathcal{N}(r) \qquad\qquad y_r \leq y_n$$

**Defining a consensus subnetwork**

By defining confidence thresholds for the relevance of nodes, edges, or paths, we can extract high-confidence, **consensus subnetworks** for further analysis. It is fairly straightforward to extract a consensus node set.

To predict additional high-confidence hits, a choice of threshold may be informed by inspecting precision-recall curves assessing the accuracy of predictions made for held-aside hits, as we perform in Section 4.3.1.

However, some considerations should be kept in mind when choosing consensus edges and paths. Because each subnetwork was inferred from a subset of the input data, we find that the distributions of edge and path confidence values are shifted left compared to the node confidence values. (The path confidence values are most dramatically shifted.) Therefore, the same confidence threshold should not necessarily be applied to nodes, edges, and paths. This is an expected and perhaps desirable result, as higher-order relationships between nodes are expected to be less likely than the individual nodes themselves. Within inferred ensembles, we find many paths for which all of the nodes and edges have confidence values that are much higher than the confidence value assigned to the path. We also find that many known hits and interfaces receive lower confidence than some high-confidence predicted hits.

A consensus subnetwork can be chosen using a different procedure depending on the task. If we wish to study a limited set of very high confidence paths, then we may choose the subnetwork by thresholding the path confidence values. If instead it would be useful to make predictions about more of the input data, we propose identifying consensus paths as those that contain known hits and interfaces as well as high-confidence predicted nodes and edges. This procedure for choosing a consensus subnetwork requires the choice of two confidence thresholds, one each for nodes and edges.

1. Identify the set of high-confidence predicted hits by thresholding the node confidence values.

2. Identify the set of paths that a) consist entirely of predicted hits and known hits and interfaces and b) that have a minimum edge confidence value above the provided threshold.

We recommend setting the node confidence threshold rather high so as to include a conservative number of additional hits. We recommend setting the edge confidence threshold somewhat lower. Both confidence values can be tuned.

### 4.2.3 Custom views into the inferred subnetwork ensemble

Considering that the hits and interfaces together comprise more than two thousand genes, even the most stringently-defined consensus subnetworks will be very large and difficult to manually inspect. Therefore, we propose a method for targeted browsing to allow a viewer to focus on the parts of the subnetwork that are most relevant to a cellular process or gene set of their interest. Given a set of query nodes assembled by the viewer, and a consensus subnetwork that has been identified by specifying a node or edge confidence threshold on the inferred subnetwork, this method extracts a **view** into the inferred subnetwork. The view consists of consensus paths that contain query nodes and known or predicted host factors whose role in viral replication is predicted to be similar to that of the queries.

#### Similarity function

First, we rank every node $n$ in the consensus subnetwork for its predicted functional similarity to the given query set $N^Q$. Nodes are considered functionally similar if they share relevant paths. Our similarity function, $s(n, \mathcal{N}^Q)$, resembles Jaccard similarity. It measures the fraction of consensus paths that contain both $n$ and at least one query node $q \in \mathcal{N}^Q$, out of all paths that contain either $n$ or any query node $q \in N^Q$. Let $\mathcal{P}_c(n)$ be the set of consensus paths that contain a node $n$.

$$s(n, \mathcal{N}^Q) = \frac{\left| \mathcal{P}_c(n) \cap \bigcup_{q \in \mathcal{N}^Q} \mathcal{P}_c(q) \right|}{\left| \mathcal{P}_c(n) \cup \bigcup_{q \in \mathcal{N}^Q} \mathcal{P}_c(q) \right|}$$

#### View creation

After ranking the consensus nodes according to this similarity function, we take the top $k$ as predicted additions to the query set. To construct a view, we extract all of the consensus paths that consist exclusively of query nodes, predicted additions, and HIV components. If specific HIV components are included in the query set, we may optionally decide not to include paths that contain other HIV components that are not in the query set.

#### Variations on the similarity function

Our proposed similarity function is easy to tailor for more specific queries. We suggest a few variations.

**Ranking for specific order relationships.** Because the relevant paths also predict the order of nodes, we can construct modified similarity functions in order to rank genes for ordered relationships to the query set. As a motivating example, we consider the case of the ESCRT pathway, one of a limited number of human pathways whose involvement in HIV replication is fairly well-mapped. Given that the direct interfaces between HIV and the ESCRT pathway are known, we may wish to query the inferred subnetwork for other host factors whose role in HIV replication is specifically mediated through upstream interactions with the ESCRT pathway. In the numerator of this ordered similarity function, $s_{\rightarrow}(n, \mathcal{N}^Q)$, we consider only relevant paths in which the node $n$ appears upstream of any query node $q$. Let $\mathcal{P}_r(n \rightarrow q)$ be the set of relevant paths in which $n$ appears upstream of $q$.

$$s_{\rightarrow}(n, \mathcal{N}^Q) = \frac{\left| \bigcup_{q \in \mathcal{N}^Q} \mathcal{P}_c(n \rightarrow q) \right|}{\left| \mathcal{P}_c(n) \cup \bigcup_{q \in \mathcal{N}^Q} \mathcal{P}_c(q) \right|}$$

This function can be generalized to rank according to other order relationships. For example, we may define $s_{\leftarrow}(n, \mathcal{N}^Q)$ in order to predict which nodes $n$ are more exclusively downstream of a query set, such as for predicting which interfaces explain a subset of upstream RNAi hits.

**Ranking with confidence-weighted paths.** Another variation on the similarity function avoids the step of defining a consensus threshold for the inferred ensemble, allowing us to see the range of confidence values in the extracted view. Instead of counting paths in the numerator and denominator of the similarity function, we weight each path $p$ by a confidence value $\text{conf}(p)$. To weight a path $p$, we may use the inferred ensemble's confidence that $\sigma_p = 1$, or, we may be more generous and set the confidence of the path as the confidence in its minimally-confident edge or node.

$$s_{\text{conf}}(n, \mathcal{N}^Q) = \frac{\sum_{p \in \mathcal{P}_c(n) \cap \bigcup_{q \in \mathcal{N}^Q} \mathcal{P}_c(q)} \text{conf}(p)}{\sum_{p \in \mathcal{P}_c(n) \cup \bigcup_{q \in \mathcal{N}^Q} \mathcal{P}_c(q)} \text{conf}(p)}$$

## 4.3 Results

In this section, we evaluate the inferred subnetworks by measuring how accurately they predict known relevant genes, as well as by inspecting some of the specific predictions. For these experiments, we infer ensembles of 100 subnetworks, varying the parameter $\delta$ (the number of additional unconfirmed genes that can be predicted to be hits) over the range $\delta$={50, 100, 500, 1000, 5000}.

### 4.3.1 Cross-validated hit prediction

Because we generate the inferred subnetwork ensemble from subsets of the input data, we can estimate our method's accuracy by checking each inferred subnetwork for the presence of the hits and interfaces that have been held aside from their input. In the absence of a set of confirmed irrelevant genes, we use the set of all unconfirmed background-network human genes as the set of 'negatives'. We exclude protein complexes and small molecules from this assessment, as we have no significant experimental data concerning the relevance of those entities. Therefore, our assessments provide a lower bound on the true accuracy of the method, as several of the unconfirmed human genes are likely to be true undiscovered hits.

For each hit and interface, we calculate its confidence value as its frequency of being included in the inferred subnetworks when held aside from the input data. On average, each hit and interface is held aside from 25 out of 100 inferred subnetworks. We calculate confidence values for unobserved genes as their frequency of inclusion over the entire ensemble. Genes that receive the same confidence value are secondarily sorted by their diffusion kernel score. After ranking all of these human genes, we plot a precision-recall curve for the ensemble.

We compare our inference method to several baseline approaches that rank the human genes according to features from the ensemble of input data sets and the background network.

- The *diffusion kernel* (DK) baseline ranks genes according to their average diffusion kernel score across the ensemble of input data sets in which they are not provided as hits or interfaces.

- The *average-candidate-paths* (Avg. paths) baseline ranks genes according to the average number of candidate paths in which they appear in data sets in which they are held aside.

- The *average-candidate-hits* (Avg. hits) baseline ranks genes according to the average number of hits that appear upstream of them in candidate path sets in which they are held aside.
- The *average-candidate-interfaces* (Avg. ints) baseline ranks genes according to the average number of interfaces that appear downstream of them in candidate path sets in which they are held aside.
- The *degree* (Degree) baseline ranks genes according to their degree in the human-gene-only background network, which we used as input to the diffusion kernel method.

We also compare the diffusion kernel baseline to another set of diffusion kernel scores calculated from samples that hold aside only one hit or interface at a time, rather than 25%. For unconfirmed genes, the scores are calculated when all hits and interfaces are used as input. The purpose of this experiment is to measure whether the data sampling technique itself results in a change in accuracy.

In Figure 4.3, we compare the baselines and the IP results on the basis of their precision-recall curves. In each figure, the horizontal green line represents the prevalence of hits and interfaces among all genes in the background network.

Figure 4.3A compares the diffusion kernel baseline (DK) to the three candidate-path-based approaches (avg. paths, avg. hits, avg. ints) and the node degree baseline. The diffusion kernel provides the highest precision, although the simple degree baseline achieves quite high precision as well. It appears that many hits and interfaces are hubs in the background network. The three candidate-path-based baselines do not achieve as high precision.

Figure 4.3B shows the results for the IP method at all tested values of $\delta$. When the inference is allowed to add only a small number (50, 100) of additional genes to the input hits and interfaces, it chooses mostly from a small set of hits, resulting in high precision but low recall. As $\delta$ increases, the IP method achieves similar precision to the DK scores for increasing levels of recall. The results therefore recommend the use of intermediate values of $\delta$ (500 and 1000), as those curves achieve a slight precision advantage over the DK curve for high-confidence predictions.

Figure 4.3C measures the effect of varying the input data for each subnetwork in the ensemble. The curves compare performance of the sampled diffusion kernel scores, with 25% of hits and interfaces held aside, to a version in which only one hit or interface is held aside. The curves are nearly identical until about the level of recall=0.60, after which the 'Leave-25%-out' curve dominates. It appears that the sampling procedure reduces the effect of noisy input data by filtering out low-confidence, unconfirmed genes.

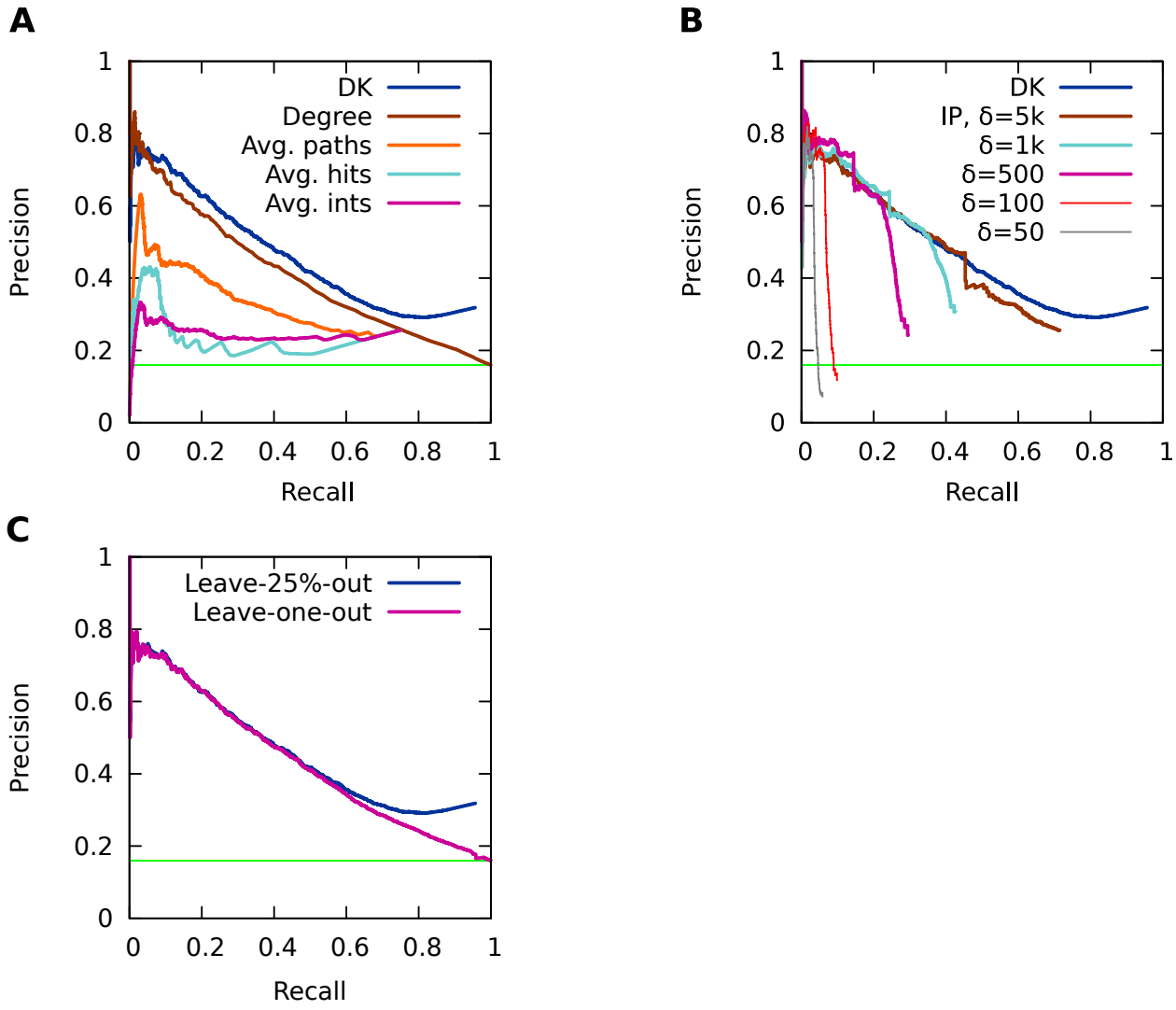



**Figure 4.3:** Precision-recall curves for hit-prediction experiments. The horizontal green line in each figure represents the prevalence of known hits and interfaces in the background network.
**A** Comparison of the diffusion kernel to baselines from candidate paths and the background network.
**B** Performance of the IP method at different values of $\delta$, the number of allowable predicted, unconfirmed genes.
**C** Assessment of the input-data-sampling approach. 'Leave-25%-out' averages the diffusion kernel scores that we input to our IP method, whereas 'Leave-one-out' scores are generated by holding aside only one phenotype label at a time.

### 4.3.2 Tractability of ensemble generation

To assess the tractability of our proposed method for generating ensembles by sampling the input data, we attempt to compare to the method we used in Chapter 3. In that approach, we find multiple solutions to the IP using all input data. After a reasonable amount of time, CPLEX is unable to find full ensembles of solutions for any tested $\delta$ except 5000, and in the process threatens to use nearly all of the 256 GB of memory available on our system. In contrast, each of our sampled IPs solves within, on average, 0.75-1.5 hours, using between 5 and 25 GB of memory. These results support our hypothesis that generating an ensemble by sampling the input data is more computationally tractable than doing so by requesting multiple IP solutions using the complete input data, at least on the data set that we consider here.

### 4.3.3 Inclusion of independently derived HIV-relevant gene sets

We also assess the IP method and baselines for their ability to recapitulate four other sources of HIV-relevant human genes. We assembled four independent gene sets for this evaluation, which are summarized in Table 4.5. First, we sought to represent genes that are relevant to T-cells, which are the cells in which HIV replicates. We identified a data set that provides human genes expressed this cell type, as well as another that identifies human proteins measured in human T-cells actually infected with HIV. Next, we gathered lists of genes whose association with HIV is recorded in the literature. We use one expert-curated list of human genes with known, indirect interactions with HIV, and one computationally-curated list of human genes that appear in at least two PubMed abstracts with variations on the keyword 'HIV'.

We plot precision-recall curves for each HIV-relevant gene set, taking the gene set as the positive class and all other human genes as the negative class. As in the hit-prediction experiment, confidence values for each gene are calculated only from data sets or inferred subnetwork ensembles in which the gene is not provided as input.

A comparison between the baselines and one representative subnetwork ensemble ($\delta$=500) are shown in Figure 4.4. All three methods are capable of predicting relevant genes. With the exception of the set of genes expressed in T-cells, the DK method and the IP-inferred subnetworks outperform the average-candidate-paths baseline. In Figure 4.5, we show the performance of the IP-inferred subnetworks as varying numbers of predicted hits ($\delta$) are allowed. As in the hit-prediction results, we see a significant gap between the performance of the method at low and high values of $\delta$. It appears to be useful to allow the method to predict at least 500 additional relevant genes.

**Table 4.5:** HIV-relevant gene sets used to evaluate the IP method. The 'Count' column gives the number of genes in the set that are present in the background network. The 'Hits' and 'Interfaces' columns give the number of hits and interfaces present in the gene set and background network.

| Description | Source | Count | Hits | Interfaces |
|---|---|---|---|---|
| Human genes expressed in T-cells, minus housekeeping genes | HIPPIE database (Schaefer *et al.*, 2012) | 1,912 | 317 | 668 |
| Human proteins expressed in HIV-infected T-cells | MOPED database, `yan_hiv` dataset (Montague *et al.*, 2014) | 924 | 93 | 373 |
| Human genes that indirectly interact with HIV | NCBI HIV-1, Human Protein Interaction Database (Fu *et al.*, 2009); indirect interactions obtained from Roger Ptak | 1,612 | 193 | 628 |
| Human genes associated with the query 'HIV' in at least two PubMed abstracts | GADGET search tool (Craven & Ziegler, 2011) | 2,992 | 209 | 1,119 |

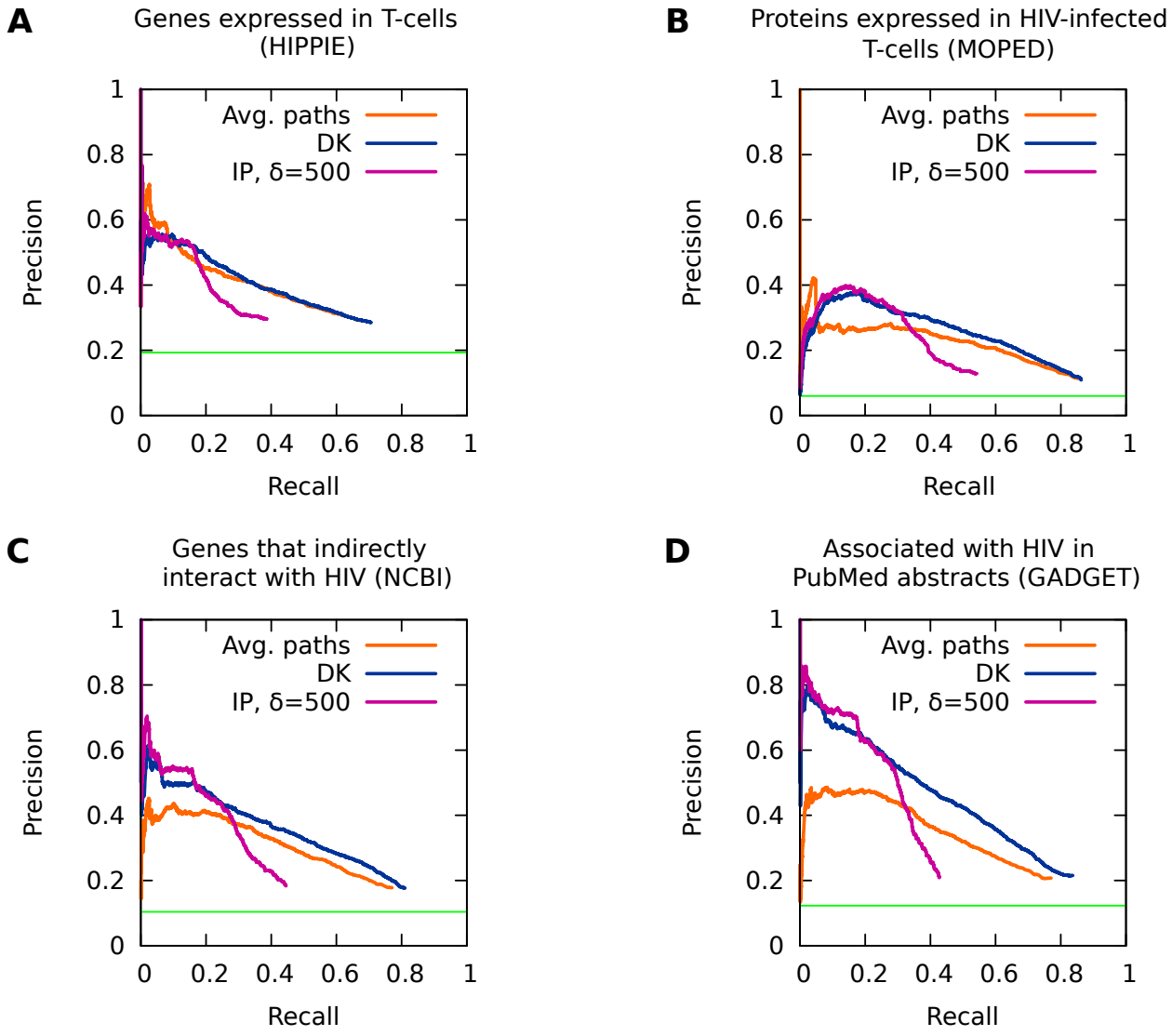



**Figure 4.4:** Precision-recall curves for relevant gene prediction experiments, comparing the performance of a representative IP-inferred subnetwork ensemble to the diffusion kernel (DK) and average-candidate-paths baseline (Avg. paths). The horizontal green line in each figure represents the prevalence of the likely relevant genes in the background network.
**A** Positives are human genes expressed in T-cells, minus housekeeping genes.
**B** Positives are human proteins expressed in HIV-infected T-cells.
**C** Positives are human genes that indirectly interact with HIV.
**D** Positives are human genes that co-occur with 'HIV' in at least two PubMed abstracts.

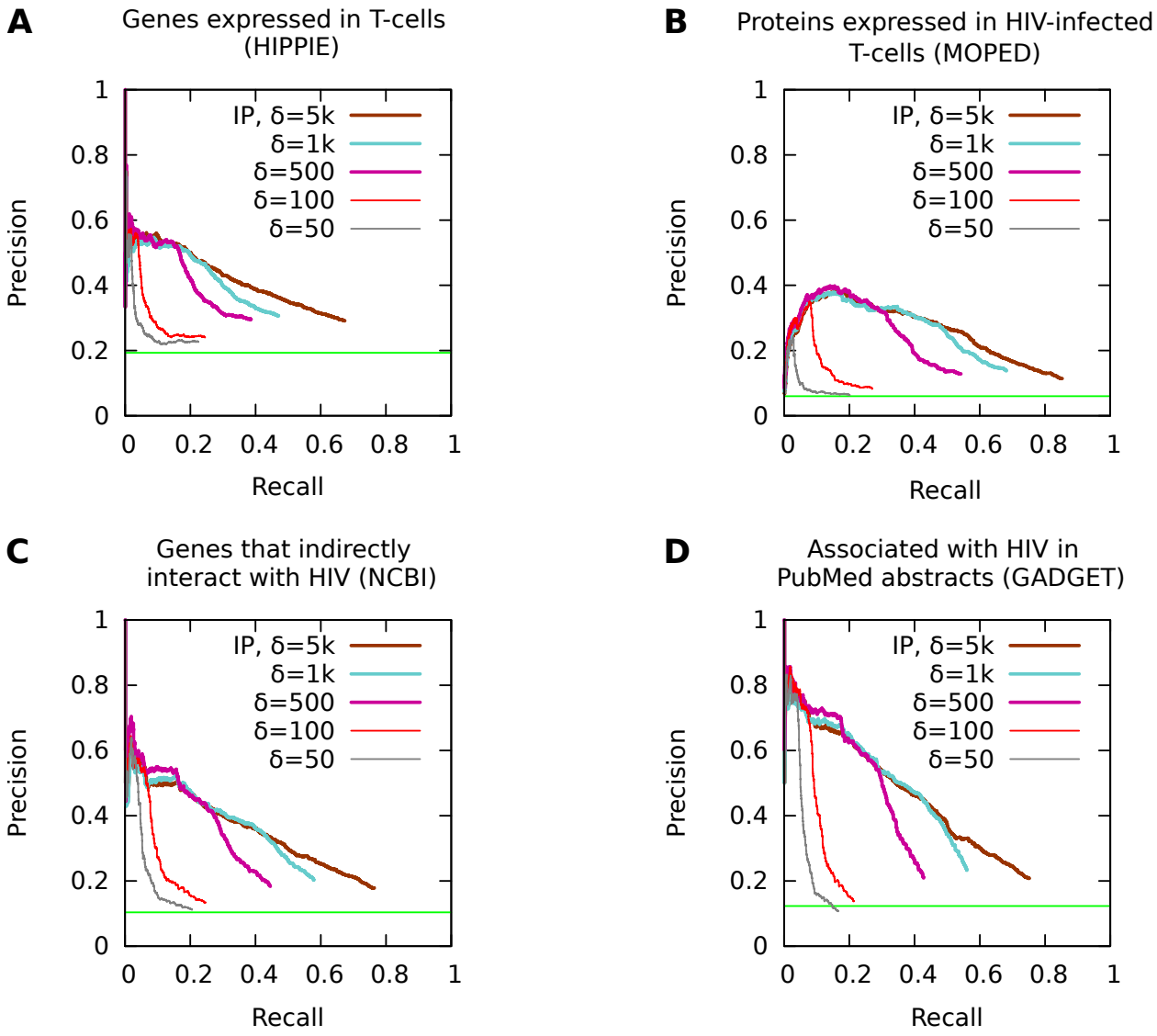



**Figure 4.5:** Precision-recall curves for relevant gene prediction experiments, showing the performance of the IP approach using increasing values of $\delta$. The horizontal green line in each figure represents the prevalence of the likely relevant genes in the background network.

**A** Positives are human genes expressed in T-cells, minus housekeeping genes.

**B** Positives are human proteins expressed in HIV-infected T-cells.

**C** Positives are human genes that indirectly interact with HIV.

**D** Positives are human genes that co-occur with 'HIV' in at least two PubMed abstracts.

### 4.3.4 Extracted view of the ESCRT pathway

Here we show an example of using our view extraction method to predict which additional host factors affect HIV through the ESCRT pathway (discussed in Section 4.2.1). This pathway is known to be hijacked by HIV to facilitate the release of new virions from the host cell. Our queries are the host factors from our manually-gathered HIV/ESCRT pathway (Section 4.2.1).

First, we infer an ensemble of 100 subnetworks, using $\delta$=500 and adding the requirement that the ESCRT interactions must be present in all inferred subnetworks. We define a consensus subnetwork by thresholding node confidence $\geq 0.90$ and edge confidence $\geq 0.75$. Then, we use the ordered version of our path-similarity-based ranking function (Section 4.2.3) to rank all consensus nodes (input hits, interfaces, and predicted hits) based on how frequently they appear in consensus paths upstream from the ESCRT query nodes. We take the top ten from this list as predicted additions to the ESCRT pathway. Finally, we expand the input HIV-ESCRT view by assembling those consensus paths that consist entirely of query genes, predicted additions, and HIV components. Nine of the top ten predicted additions are included in these paths. Seven of the predicted additions are either hits or interfaces, and two are predicted hits. The view therefore generates the hypothesis that suppressing any of the genes in this list will alter HIV replication as well as the function of the ESCRT pathway.

Figure 4.6 and Figure 4.7 show the input ESCRT interactions and the expanded ESCRT view, respectively. Table 4.6 lists the additional host factors that our method predicts are relevant to HIV replication specifically through interactions with the ESCRT pathway.

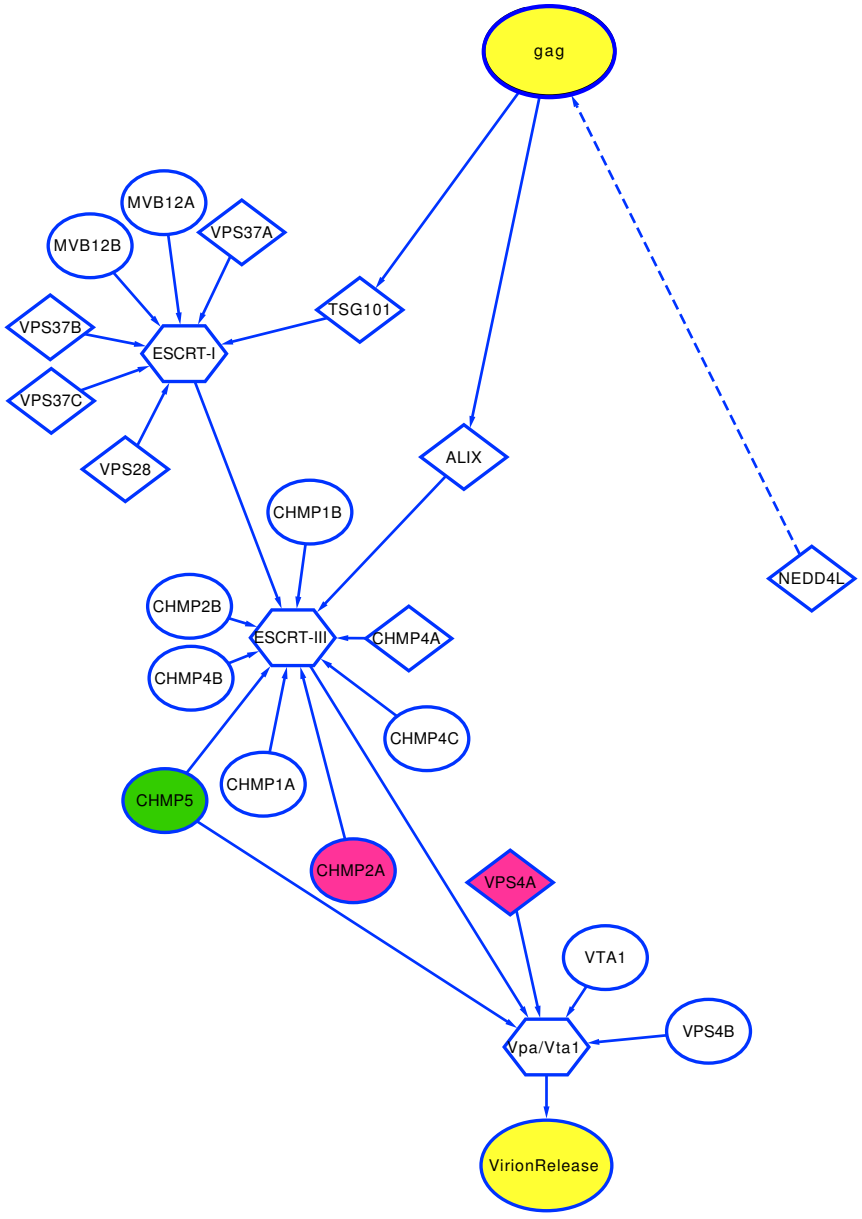



**Figure 4.6:** Input ESCRT-HIV pathway as provided to the IP method. Input edges and borders of input nodes are shown in blue. See Figure 4.1 for a key to the other graphical elements.

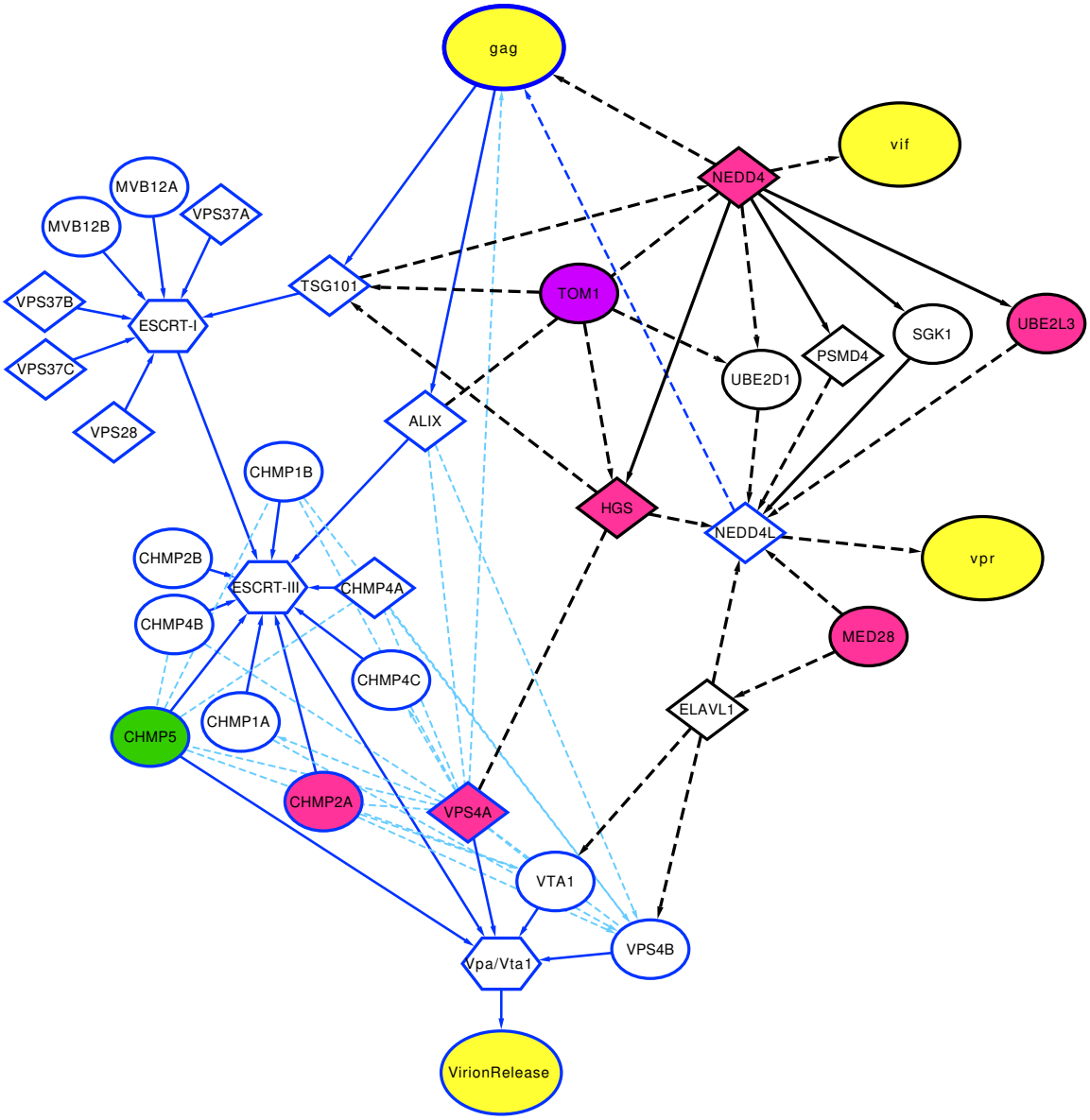



**Figure 4.7:** Expanded view of ESCRT-HIV pathway, including input nodes as well as the top nine host factors that the ranking function predicts are additions to the pathway. Input edges and borders of input nodes are shown in blue, inferred relevant edges between input nodes are shown in lighter blue, and inferred relevant nodes and edges are shown in black. Undirected edges are dashed, while directed edges are solid. When undirected edges are used in only one direction by the relevant paths for this view, that direction is indicated with an arrowhead. See Figure 4.1 for a key to the other graphical elements.

**Table 4.6:** Predicted additions to the ESCRT-HIV pathway. We show the top nine host factors whose known or predicted role in HIV replication is posited to be mediated by interactions with the ESCRT pathway.

| Symbol | Description | Phenotype label |
|---|---|---|
| ELAVL1 | ELAV like RNA binding protein 1 | Interface |
| HGS | hepatocyte growth factor-regulated tyrosine kinase substrate | HDF, interface |
| MED28 | mediator complex subunit 28 | HDF |
| NEDD4 | neural precursor cell expressed, developmentally down-regulated 4, E3 ubiquitin protein ligase | HDF, interface |
| PSMD4 | proteasome (prosome, macropain) 26S subunit, non-ATPase, 4 | HDF, interface |
| SGK1 | serum/glucocorticoid regulated kinase 1 | – |
| TOM1 | target of myb1 (chicken) | Both HDF, HRF |
| UBE2D1 | ubiquitin-conjugating enzyme E2D 1 | – |
| UBE2L3 | ubiquitin-conjugating enzyme E2L 3 | HDF |

### 4.3.5 Predicted relevant protein complexes and reactions

To investigate the outcome of our special constraints for protein complexes and reactions, we inspected the predictions made by the inferred subnetworks (with $\delta$=500). We find 12 protein complexes (comprised of nine unique gene sets), and no reactions, with confidence $\geq 0.90$. The predicted relevant protein complexes are shown in Table 4.7; we have collapsed complexes with duplicate node content into the same entry. We show the number of hits and interfaces in each complex, as well as the number of predicted hits (with confidence $\geq$ 0.90). The same component may be counted both as a hit and interface.

While the method appears to have identified several of the complexes because they contain many hits or interfaces, a few complexes include predicted hits as well. Some of the predicted complexes are relevant to transcriptional elongation, which our analyses also identified as relevant to BMV-yeast interactions.

The reactions that were included in any subnetwork all received confidence $\leq 0.65$. Therefore, including the Reactome pathways in the background network does appear to be somewhat useful, but not enough that any of the reactions are included with high confidence. It is possible that a modified representation of this data may be more useful.

**Table 4.7:** Inferred HIV-relevant human protein complexes (confidence $\geq 0.90$). For each complex, we note its intracellular location according to Reactome, a high-level description of its function, its total number of components ('Size'), as well as the number of components that are hits ('Hits'), interfaces ('Ints.'), and predicted hits with confidence $\geq 0.90$ ('Pred.').

| Complex name | Function | Size | Hits | Ints. | Pred. |
|---|---|---|---|---|---|
| Pol II transcription complex/Promoter Escape Complex [nucleoplasm] | Transcription | 41 | 11 | 38 | 0 |
| Elongation complex [nucleoplasm] | Transcription elongation | 44 | 14 | 40 | 0 |
| Processive/paused/arrested elongation complex [nucleoplasm] | Transcription elongation | 33 | 11 | 31 | 0 |
| Exon Junction Complex [nucleoplasm] | mRNA processing | 110 | 27 | 61 | 16 |
| Spliceosomal active C complex with lariat containing, 5-end cleaved pre-mRNP:CBC complex [nucleoplasm] | mRNA processing | 108 | 27 | 60 | 16 |
| TAP:3-polyadenylated, capped mRNA complex [nucleoplasm] | mRNA processing | 25 | 7 | 8 | 5 |
| NOTCH1 Coactivator Complex [nucleoplasm] | T-cell development | 12 | 3 | 5 | 0 |
| EGF:p-6Y-EGFR:GRB2:p-5Y-GAB1:SHP2 [plasma membrane] | Epidermal growth | 4 | 2 | 1 | 0 |
| EGF:p-6Y-EGFR:CBL:CIN85 [plasma membrane] | Epidermal growth | 4 | 2 | 0 | 2 |

## 4.4 Discussion

This chapter details a proof-of-concept extension of our subnetwork inference method (Thesis Contribution 1) to assist in the study of HIV replication in human cells (Thesis Contribution 2). These inferred subnetworks accurately predict held-aside hits and interfaces, as well as independently gathered lists of HIV-relevant human genes. Additionally, we make several contributions toward improving the interpretability of the inferred subnetworks (Thesis Contribution 3):

- Our method offers scientists a way to combine data from RNAi experiments and human-HIV protein interaction assays into ensembles of inferred subnetworks. These are relevant data types that are being gathered at large scales, and which are difficult to comprehend by manual inspection. Our inferred subnetworks predict paths that connect the RNAi hits to human-HIV protein interactions, and predict which unconfirmed human genes are also relevant to HIV.
- We provide another method for generating ensembles by varying the input data provided to the inference method. In support of this method, we observe some improvement in the accuracy of low-confidence predictions of the diffusion kernel when we use a large sample of the input data as compared to the entire data set. In our experiments, we also find that this method is computationally feasible given our data and resources. In contrast, we were unable to generate ensembles of multiple subnetworks from the complete data set using the CPLEX solver.
- The background network provided to our method represents many types of intracellular interactions. In addition to binary protein-protein interactions, we also include protein complexes, in order to provide an improved representation of biological knowledge. Our inference method includes a mechanism for including only over-represented complexes in an inferred subnetwork. Over-representation is tested based on both input and inferred relevant genes, thereby allowing the identification of complexes that may not have been identified from the input data alone. Our experiments show that the inferred subnetworks do make use of protein complexes.
- To overcome the difficulty of interpreting large inferred subnetworks, we propose a method for extracting a 'view' that shows connections between a set of query host genes and additionally predicts which other genes are specifically, functionally similar to the queries. These views are useful for assisting in the browsing and interpretation of the inferred subnetworks, as well as for generating testable hypotheses.

# 5 Inferring the salt-responsive subnetwork for yeast stress

This chapter represents a third extension of our general method to a different biological problem: inferring the signaling pathways that orchestrate the yeast response to salt stress. Again, we use a variety of biological interactions to build our background network. A new aspect of this work compared to our host-virus research is that the condition-specific data incorporates three different types: a set of source-target pairs and two independently-derived hit sets. Each set comes from an experiment that observes a different aspect of the stress response. We use specialized constraints and objective functions in the integer linear program to govern how each data set can be integrated into the inferred subnetworks.

Our application promotes interpretability through the content and representation of the input data, and the constraints of the model. Additionally, we present a method for querying the inferred subnetworks about what role each gene may play in coordinating different aspects of the stress response.

This project is a collaboration between computational and biological researchers. In addition to evaluating the inferred subnetworks using computational methods and comparison to the literature, we have been able to validate the subnetworks with additional experimental results. Many of the predictions made by the subnetworks are accurately reflected by these new data. Furthermore, the inferred subnetworks are being used to guide continuing experimental work.

**This work is under submission with the following citation:**
Deborah Chasman and Yi-Hsuan Ho[1], Corey M. Nemec, David B. Berry, Matthew E. MacGilvray, James Hose, Anna E. Merrill, M. Violet Lee, Jessica L. Will, Joshua J. Coon, Aseem Z. Ansari, Mark Craven, Audrey P. Gasch. Coordination and interconnectivity in the inferred stress-activated signaling network from yeast.

The computational methods for this chapter are the work of myself and my advisor, Mark Craven. The biological contributions and literature-based analyses are due to our co-authors, led by Audrey Gasch, with Yi-Hsuan Ho as first author in genetics. As this work was performed collaboratively, many of the analyses represent contributions from multiple authors. The complete biological methods and results are left to the original publication.

[1]Both first authors contributed equally

## 5.1 Introduction

All cells respond to stress by orchestrating complex responses customized for each situation. When grown in optimal conditions, *Saccharomyces cerevisiae* maintains high expression of growth-related genes and low transcription of stress-defense genes, in part via nutrient responsive TOR and RAS-regulated Protein Kinase A (PKA) signaling (Smets *et al.*, 2010; Broach, 2012). Suboptimal conditions suppress these pathways in an unknown manner while activating stress-specific signaling networks that coordinate changes in transcription and translation, protein function, and metabolic fluxes with transient arrest of growth and cell cycle progression. How these disparate physiological processes are coordinated is poorly understood but likely critical for surviving and acclimating to stressful conditions.

At the level of gene expression, stressed yeast activate condition-specific transcript changes that provide specialized stress defenses. These responses are typically regulated by condition-specific transcription factors (TFs) and upstream signaling pathways that are activated under limited circumstances (Hohmann & Mager, 2003). In addition to these specialized responses, stressed yeast cells also activate the common **environmental stress response** (ESR) (Gasch *et al.*, 2000; Causton *et al.*, 2001). The ESR includes 300 induced (iESR) genes that are broadly involved in stress defense and 600 repressed-ESR (rESR) genes that together encode ribosomal proteins (RPs) and proteins involved in ribosome biogenesis/protein synthesis (RiBi). While the complete set of ESR regulators remains elusive, it is clear that the program is regulated by different upstream signaling factors under different situations (Gasch *et al.*, 2000, 2001; Gasch, 2003).

Transcript changes triggered by acute stress are in fact not required to survive the initial stressor, since ablating transcription or translation with drugs or mutation have no affect on acute-stress tolerance (Berry & Gasch, 2008; Westfall *et al.*, 2008). Instead, stress-induced changes to transcription and translation are required for acquisition of tolerance to subsequent stress (Berry & Gasch, 2008; Westfall *et al.*, 2008; Mitchell *et al.*, 2009; Berry *et al.*, 2011). Therefore, screens for **transcriptome** regulators based on single-stress sensitivities have likely missed many signaling proteins, rendering stress-dependent signaling networks incomplete. Although several isolated pathways are well characterized, how signaling is integrated through a single cellular system is poorly understood.

Here we present a computational approach to infer the complete sodium chloride (NaCl)-activated signaling network from a combination of data types. A key feature of our approach is that we incorporate several large-scale datasets (including mutant transcriptome profiles, phospho-proteome changes, and gene fitness contributions) that our collaborators generated under the same culture system in cells responding to acute NaCl stress.

Stress responses are highly context-dependent (Berry *et al.*, 2011), and therefore differences in culturing conditions (including strain, medium, osmolytes, and stress dose) can produce different downstream responses and even involve different regulators (Van Wuytswinkel *et al.*, 2000; O'Rourke & Herskowitz, 2004). Therefore, while there have been many insightful prior studies characterizing the salt response in yeast (*e.g.* Hirasawa *et al.* (2006); Capaldi *et al.* (2008); Melamed *et al.* (2008); Halbeisen & Gerber (2009); Martínez-Montañés *et al.* (2010); Warringer *et al.* (2010); Miller *et al.* (2011); Causton *et al.* (2001); Westfall *et al.* (2008); Soufi *et al.* (2009)), we restrict our analysis to datasets generated by our collaborators under identical culturing conditions. We use an integer linear programming (IP) approach to integrate and interpret three disparate datasets by inferring a signaling subnetwork. The novel facets of our computational approach include a means to integrate these varied data sources, using new types of input paths to the IP, and a multi-part objective function. The resulting subnetwork has generated many new insights into stress signaling.

### 5.1.1 Related work

Our computational method infers the stress-activated signaling subnetwork, both to implicate missing regulators and to understand their connections. As we discuss in Chapter 2, extensive previous work has approached the challenge of inferring signaling networks using diverse, large-scale biological data.

Extensions to *regulatory network reconstruction* (Chapter 2.1) focusing on the osmotic response include the work of Gat-Viks et al. (2006; 2007), whose probabilistic method learns parameters to describe the regulatory relationships between known regulators of the Hog pathway, assuming a known network topology.

As in the rest of this thesis, the method we present here is most closely related to *subnetwork inference* methods (Chapter 2.2) that take as input source-target pairs and perform inference using an integer linear program. Unlike the previous chapters, in this chapter our primary data source is a set of source-target pairs from single mutants profiled under stress conditions. During subnetwork inference, our approach predicts the relevance of each node and edge, but not edge signs. In particular, we compare to the work of Gitter *et al.* (2013), who present a combined probabilistic/IP method to discern signaling in the potassium chloride-responsive subnetwork from time-series expression data. However, this work focuses on analyzing transcriptome data only, without additional data types (*e.g.*, proteomics and genetic contributions) as considered here.

Previously, *subnetwork extraction* methods (Chapter 2.3) have been proposed to integrate heterogeneous data types, including phosphoproteomic and transcriptomic data, to iden-

tify signaling pathways. One example is the work of Huang *et al.* (2013), who apply their prize-collecting Steiner tree method to identify oncogene-induced signaling regulators. However, unlike in our setting, they use independently derived sets of potantial upstream regulators and downstream targets. They do not use source-target pairs in which downstream transcriptomic changes are tied to perturbations of specific upstream regulator genes. Additionally, they do not infer directed subnetworks.

A significant contribution of our method is that we provide means to selectively integrate the datasets by the use of four types of paths between gene products in the background network. Each dataset is prioritized separately by a series of objective functions. Other related methods use a single objective function. Related subnetwork inference methods essentially maximize the number of paths between sources and targets (Gitter *et al.*, 2011; Ourfali *et al.*, 2007; Yeang *et al.*, 2004, 2005). In contrast, our method's preference for sparse inferred subnetworks is also employed by subnetwork extraction methods based on the prize-collecting Steiner tree algorithm (Huang & Fraenkel, 2009, 2012; Huang *et al.*, 2013; Yosef *et al.*, 2009) and flow-based algorithms (Lan *et al.*, 2011; Yeger-Lotem *et al.*, 2009). However, those methods require the use of a weight parameter to trade off between subnetwork sparsity and the inclusion of known relevant proteins. Another contribution of our method is the way in which we represent uncertainty about the underlying network. We assign a confidence value to each protein and interaction according to its frequency of occurrence in an ensemble of optimal inferred subnetworks. This is similar to the score used by Yeang *et al.* (2005), who actually enumerate all optimal solutions, doing so is practically intractable for our input and our model. In contrast, Ourfali *et al.* (2007) assign confidence values based on the change in objective value when each protein or interaction is individually excluded, and Gitter *et al.* (2011) present several methods for ranking paths based on input experimental data and local topological features of the inferred subnetworks.

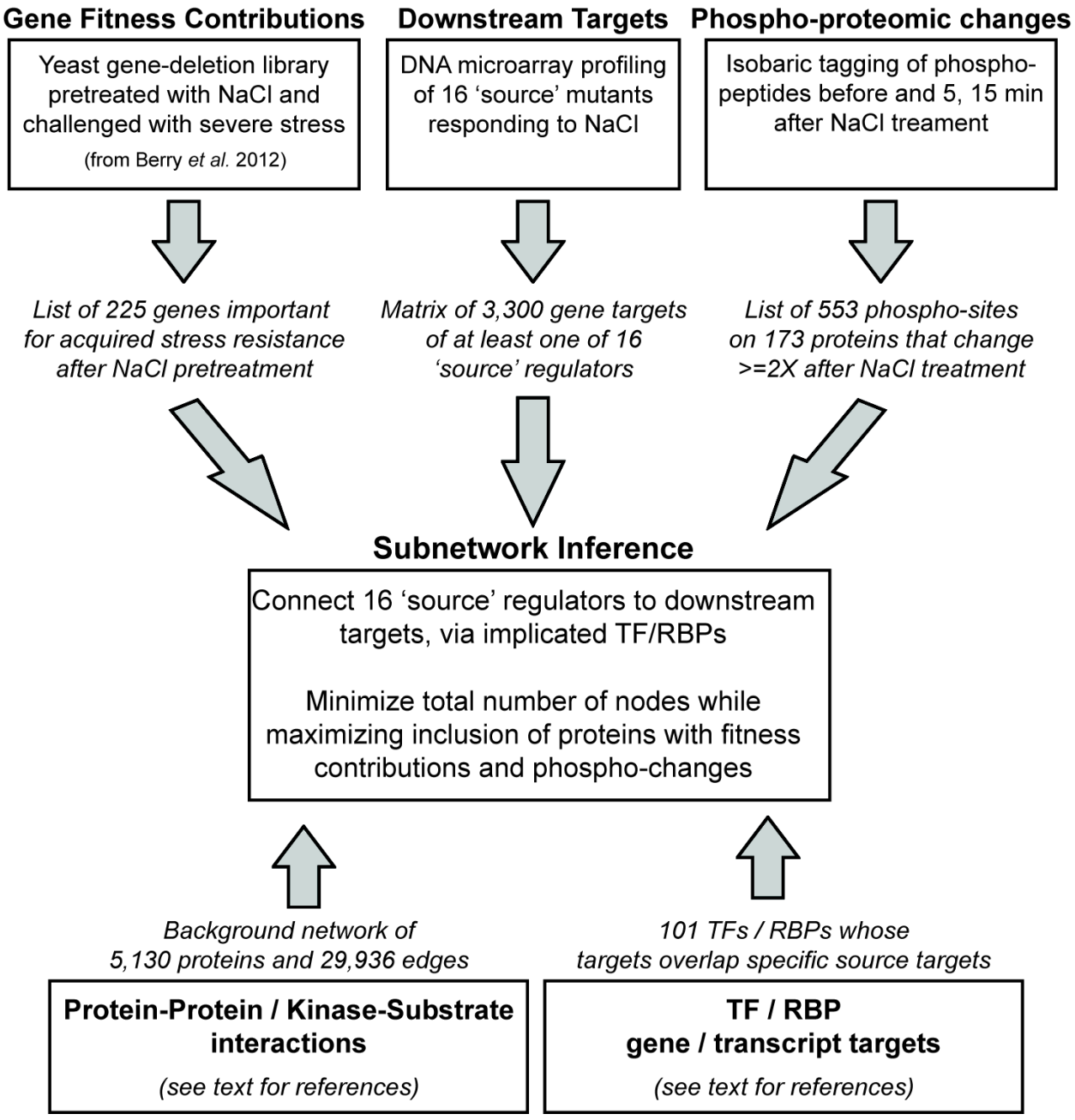



**Figure 5.1:** Overview of the experimental data generation and analysis to generate input for the subnetwork inference method. Figure contributed by Audrey Gasch.

## 5.2 Materials and methods

This project generated and assembled several rich sources of data, outlined in Figure 5.1. The inputs to our subnetwork inference method (the middle box in the figure) are three salt-response-specific data sets (top of the figure), a background network consisting of

protein-protein interactions and RNA-regulatory interactions (bottom of the figure), and a small set of salt-relevant relationships identified from the literature (not pictured).

In Section 5.2.1, we describe the three salt-response-specific data sets and the relationships from literature. Then, in Section 5.2.2, we walk through an overview of the entire subnetwork inference method. In Section 5.2.3, we give the details of the background network. Sections 5.2.4 and 5.2.5 give the complete details of the subnetwork inference method.

### 5.2.1 Salt-response data

First, we describe the three major input data sets. Detailed information about the biological experimental methodology used to gather the data is available in the original publication.

#### Fitness-contribution hits

The Gasch lab previously identified 225 genes important for acquired stress resistance after NaCl pretreatment (Berry *et al.*, 2011), including known signaling proteins activated by NaCl. For each of the identified relevant mutants, we apply the label **fitness-contribution hit** to the gene product, because of the mutation's negative effect on yeast fitness under salt stress.

#### Source-target pairs

Because only a fraction of NaCl-dependent transcript changes are important for acquired stress resistance, the mutant fitness assay misses many upstream transcriptome regulators. Therefore, to implicate the complete upstream signaling subnetwork, the Gasch lab profiled NaCl-dependent expression changes in deletion mutants of 16 of the fitness-contribution hits. Together, these experiments generate a matrix of regulator-gene target predictions that encompasses 3,300 genes (Table 5.1). A third of the target genes are affected by $\geq 2$ regulator mutants, and there is significant overlap in several target-gene sets. These results hint at the complex upstream signaling that controls the NaCl-responsive transcriptome.

From this matrix, we extract a set of source-target pairs, each consisting of a single signaling protein (*source*) and a gene that is dysregulated in the mutant under salt stress (*target*). We divide the set of pairs into two categories, based on whether or not the source is a known transcription factor (TF) or RNA-binding protein (RBP) or not. For non-TF/RBP sources, the effect on the targets is assumed to be indirect, and mediated by interactions between other gene products. For sources that are TFs/RBPs, it is assumed that they affect their targets directly.

**Table 5.1:** Gene targets identified in source regulator mutants. The two 'Targets' sub-columns give the number of genes with smaller ('Defective') or larger ('Amplified') expression changes compared to the wild type strain. Note that this table includes non-coding RNAs. Table contributed by Audrey Gasch.

| | | Targets | |
|---|---|---|---|
| Mutant | Replicates | Defective | Amplified |
| *hog1*Δ | 3 | 1,378 | 565 |
| *pde2*Δ | 3 | 517 | 59 |
| *mck1*Δ | 3 | 794 | 101 |
| *msn2*Δ | 3 | 184 | 26 |
| *rim101*Δ | 3 | 75 | 227 |
| *gpb2*Δ | 2 | 202 | 37 |
| *rim15*Δ | 2 | 438 | 106 |
| *npr2*Δ | 2 | 75 | 69 |
| *npr3*Δ | 2 | 184 | 89 |
| *swc3*Δ | 2 | 108 | 257 |
| *swc5*Δ | 2 | 84 | 55 |
| *whi2*Δ | 2 | 118 | 201 |
| *pph3*Δ | 2 | 235 | 21 |
| *sub1*Δ | 2 | 431 | 97 |
| *tpk1*Δ | 2 | 35 | 96 |
| *ygr122w*Δ | 2 | 106 | 502 |

**Phospho-proteomic hits**

Because much of signal transduction occurs post-translationally, the Coon lab measured changes to the phospho-proteome before and at 5 and 15 min after NaCl treatment, using chemical isobaric tags for phospho-peptide quantification. Nearly 600 of 1937 identified phospho-sites (mapping to 973 proteins) show a $\geq$ 2-fold change in phosphorylation, roughly split between sites with increased and decreased modification. We apply the label **phospho-proteomic hit** to each of the 324 proteins that showed a consistently-signed, $\geq$ 2-fold change at any phosphorylation site.

**Receptor-source pairs from literature**

Our method can take advantage of domain knowledge about the salt stress response in order to provide a scaffold for the inferred subnetwork. Here, we want to capture the upper-most stress sensors that may otherwise be missed in connecting sources to their downstream targets. We identified well-known indirect relationships between two transmembrane receptors, Sln1 and Sho1, and one of the sources, Hog1 (Saito & Tatebayashi, 2004). We provide this information to our method as *receptor-source* pairs: (Sln1, Hog1) and (Sho1, Hog1).

### 5.2.2 Overview of the subnetwork inference method

To integrate and interpret the three data sets, we designed an integer linear programming-based (IP) approach to infer the subnetwork of directed paths from interrogated *source* regulators to their *target* genes. We illustrate the approach in Figure 5.2.

We start with a *background network* of directed and undirected intracellular interactions representing protein-protein, kinase-substrate, and gene regulatory interactions between proteins and genes/mRNAs (Figure 5.2A, left side). For each interrogated source regulator, we identify candidate transcription factors (TFs) and RNA-binding proteins (RBPs) whose known binding targets significantly overlap with the source's targets (Figure 5.2A, right side). We then enumerate all possible directed, linear candidate paths (using an iterative deepening search up to a given length) that connect each of the 16 interrogated source regulators to the majority of their targets (Figure 5.2B). The final interaction in each path represents the regulation of the target gene by a candidate TF or RBP. Other candidate paths connect proteins required for fitness (Figure 5.2B, blue nodes), proteins with NaCl-dependent phosphorylation changes (yellow nodes), and two known upstream sensors (pink nodes; described in Section 5.2.1).

The candidate paths serve as input to the IP, which encodes the relevance of each network element as a binary variable and characterizes possible subnetworks using a set of linear constraints over these variables (Figure 5.2C). Subnetwork inference is performed by choosing a union of relevant, directed paths that optimize a series of successively applied objective functions that aim to connect experimentally implicated proteins, allowing the sparing inclusion of additional proteins. Because many distinct subnetworks may score equally well, we use the IP to identify an ensemble of high-scoring subnetworks. Each protein, interaction, and path is assigned a confidence value based on its frequency of appearing across the ensemble.

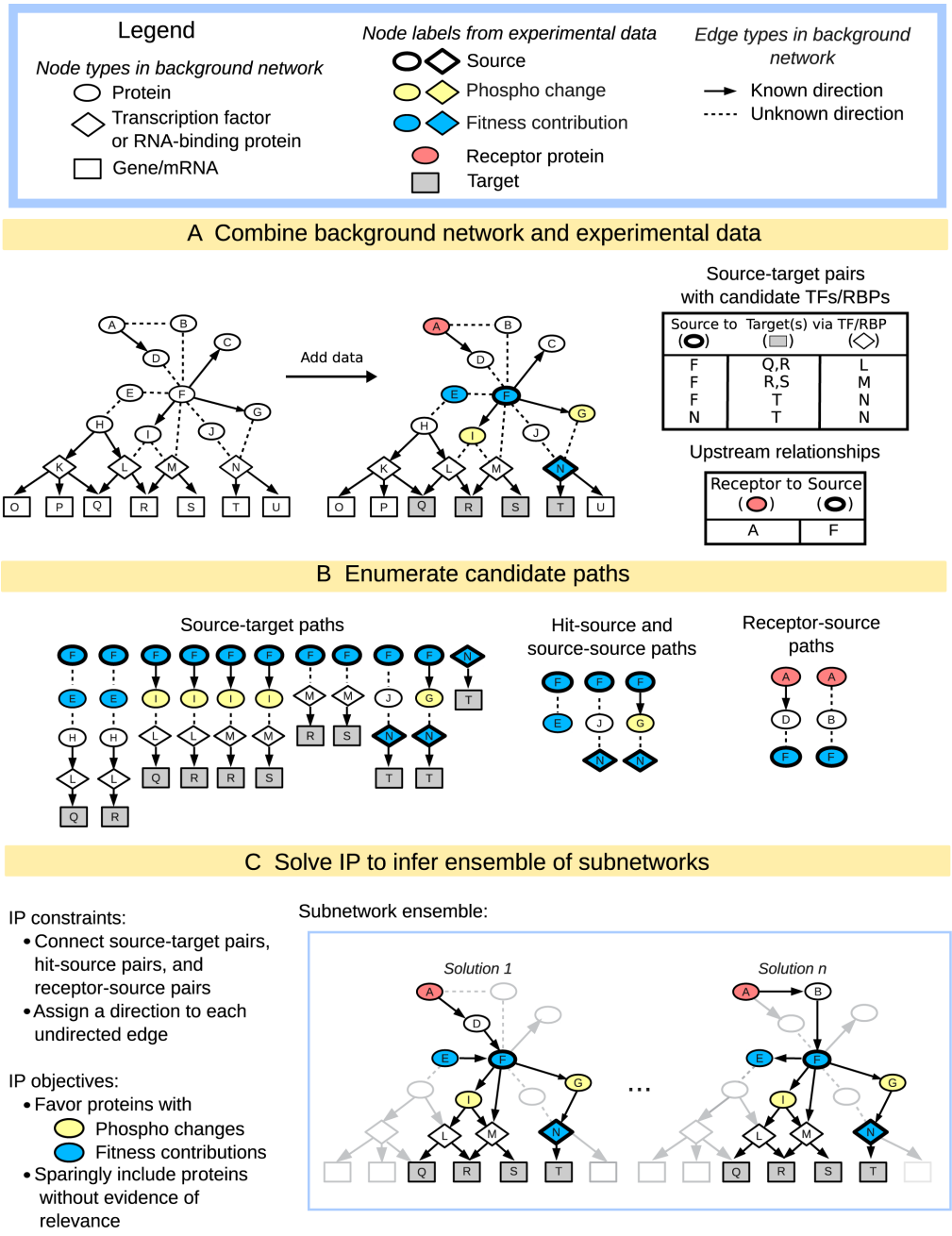



**Figure 5.2:** Overview of the subnetwork inference method.
**A** The input to the method includes a background network of yeast interactions combined with experimental data that describes the yeast salt stress response, including proteins with phospho-changes (yellow), proteins for genes that contribute to fitness under additional stress (blue), and two known upstream regulators (pink).
**B** The three different types of paths that we enumerate using the background network and experimental data, where 'hit' refers to proteins identified in the original fitness screen or with significant changes in phosphorylation.
**C** The IP for subnetwork inference and the output ensemble of inferred subnetworks.

### 5.2.3 Background network

To construct our background network, we use a variety of binary interactions that are relevant to intracellular signaling and gene expression regulation. The background network represents protein-protein (including kinase-substrate), protein-DNA, and protein-RNA interactions gathered from public databases.

#### Data types

Physical protein-protein interactions (PPIs) are sourced primarily from BioGRID (Stark *et al.*, 2006); these include both undirected PPIs and directed post-translational modifications, such as ubiquitination. In order to restrict the background network to high-confidence interactions, we only retrieve interactions that are supported by at least two separate experimental methods. In order to specifically capture signaling pathways, we supplement the PPI network with additional high-confidence protein-protein interactions involving kinases and phosphatases (Breitkreutz *et al.*, 2010; Fasolo *et al.*, 2011; Sharifpoor *et al.*, 2011). Additional directed kinase-substrate and phosphatase-substrate interactions are identified from BioGRID (contributed by Ptacek *et al.* (2005)) and the KID database (Sharifpoor *et al.*, 2011).

As in Chapter 3, we also use binary interactions from metabolic pathway data (Heavner *et al.*, 2012). Instead of including small molecules in the backround network, we convert the metabolic pathways into edges between enzymes that catalyze adjacent metabolic reactions. We also use interactions that represent which proteins are components of annotated protein complexes (Heavner *et al.*, 2012; Pu *et al.*, 2009).

Using protein-protein interactions, we also infer undirected interactions between protein complexes and between a complex and a single protein. If more than 50% of the possible protein-protein interactions between two protein complexes (or between a complex and a protein) are present in the data, then we add an undirected edge between the two complexes (or complex and protein).

We represent the direct regulation of the target genes/mRNAs using protein-DNA (Abdulrehman *et al.*, 2011; Everett *et al.*, 2009; Guelzim *et al.*, 2002; MacIsaac *et al.*, 2006; Venters *et al.*, 2011) and protein-RNA (Hogan *et al.*, 2008; Scherrer *et al.*, 2010; Tsvetanova *et al.*, 2010) interactions from several publications. We supplement these with two sources of protein-DNA interactions gathered under stress conditions (Huebert *et al.*, 2012; Ni *et al.*, 2009).

After manual inspection of the background network neighborhoods of the interrogated mutants, we added a set of 17 missing interactions between the mutants and nearby regulators based on known interactions in the literature.

**Representation**

While the types of biological interactions in the background network are rich and diverse, we use a simplified representation as input to the computational method. The background network is represented as a graph, in which nodes represent genes and gene products, and edges represent interactions. A gene may be represented as two separate nodes in the background network: one representing the protein, and, for targets, one representing the DNA or mRNA. Each interaction may have a direction: for example, transcriptional regulatory interactions are directed, but most protein-protein interactions are not. (Protein-protein interactions representing post-translational modifications, such as phosphorylation, are directed.)

We represent each protein complex as an additional node, and add a directed edge to the complex from each of its component proteins.

Finally, we collapse Dot6 and its paralog Tod6 into the same protein node in the background network, because they have been observed to function redundantly and because our measured targets were identified in a *dot6*$\Delta$*tod6*$\Delta$ double-mutant strain. We accomplish this by replacing the two separate proteins with a new protein node, named Dot6/ Tod6.

In total, the background network consists of 5,130 nodes representing proteins or protein complexes and 6,481 nodes representing DNA or RNA. There are 29,936 unique, interacting pairs of proteins (27% having a known direction) and 260,365 unique pairs of one protein and one nucleic acid sequence. Information about the provenance of the background network is provided in Table 5.2. The background network itself (minus transcriptional regulatory edges that do not involve experimental targets) is available in Cytoscape format as supplementary material to the original publication.

**Table 5.2:** Provenance of interactions in the background network. Notes about database extraction procedures are given in italicized text.

| Interaction | Source | Directed | Count |
|---|---|---|---|
| | *Protein-protein interactions* | | |
| General protein-protein | Stark *et al.* (2006) (BioGRID) | Both | 17,306 |
| | *Interactions supported by at least two categories of experimental methods; treated phosphorylation edges separately; downloaded 3/2011* | | |
| PPIs involving kinases and phosphatases | Breitkreutz *et al.* (2010) (YeastKinome.org) | No | 989 |
| | Fasolo *et al.* (2011) | No | 1,028 |
| | Sharifpoor *et al.* (2011) (Yeast KID) | No | 138 |
| | *Interactions without evidence of direct phosphorylation and annotated with p<0.01; downloaded 9/2012* | | |
| Kinase-substrate interactions | Sharifpoor *et al.* (2011) | Yes | 414 |
| | *Annotated with p < 0.01 and one of the following evidence codes: "(LTP in vitro kinase assays OR In vitro phosphorylation site mapping (Mass Spec, Phospho-specific antibodies by Western, in vitro site-directed mutagenesis) OR In vivo site-directed mutagenesis in substrate showing same biological consequence as the kinase delete OR Reduction in phospho-peptide in vivo by mass-spec OR In vivo phosphorylation site mapping using phospho-specific antibodies (Western blot) or by phospho-peptide mapping)"; downloaded 9/2012* | | |
| | Ptacek *et al.* (2005); Stark *et al.* (2006) | Yes | 5,315 |
| | *All interactions from Ptacek* et al. *(2005), plus low-throughput phosphorylation/dephosphorylation interactions from BioGRID, March 2011* | | |
| Manually curated | This work | Both | 17 |
| | *Hand-constructed after inspection of neighborhoods of interrogated sources* | | |

Continued on next page

| Interaction | Source | Directed | Count |
|---|---|---|---|
| | *Protein-nucleic acid interactions* | | |
| Protein-DNA | Abdulrehman *et al.* (2011); Everett *et al.* (2009); Guelzim *et al.* (2002); MacIsaac *et al.* (2006); Venters *et al.* (2011) | Yes | 259,565 |
| Osmotic-stress specific, protein-DNA | Huebert & Gasch (2012) | Yes | 1,225 |
| Salt-stress specific protein-DNA | Ni *et al.* (2009) | Yes | 2,144 |
| Protein-RNA | Hogan *et al.* (2008); Scherrer *et al.* (2010); Tsvetanova *et al.* (2010) | Yes | 17,868 |
| | *Other interaction types* | | |
| Between metabolic enzymes | Heavner *et al.* (2012) | Both | 1,153 |
| | *Created binary interactions from enzymes reported to catalyze adjacent reactions; reported as "directed" if the reaction was annotated as "irreversible"* | | |
| Complex membership | Heavner *et al.* (2012); Pu *et al.* (2009) | Yes | 2,183 |
| | *Directed interactions from protein to complex* | | |
| Inferred complex-complex interactions | Heavner *et al.* (2012); Pu *et al.* (2009), Stark *et al.* (2006) | No | 22 |
| | *Interaction between two complexes inferred if >50% possible protein pairs have interactions in BioGRID* | | |
| Inferred complex-protein interactions | Heavner *et al.* (2012); Pu *et al.* (2009), Stark *et al.* (2006) | No | 1,128 |
| | *Interaction between complex and protein inferred if >50% possible protein pairs have interactions in BioGRID* | | |

### 5.2.4 Details of the subnetwork inference method

In this section, we present a detailed description of the three steps of our inference method, illustrated in Figure 5.2.

**Step 1: Gather the experimental data and a background network.**

Figure 5.2A shows the input to the method: three data sets from experiments that interrogate the yeast salt stress response, and a background network of intracellular interactions. The background network contains three types of nodes: proteins (shown as ellipses), TFs and RNA-binding proteins (diamonds), and genes/mRNAs (rectangles). The experimental data comes in the following forms:

- *Source-target pairs,* each consisting of one interrogated *source* regulator (bold outlined nodes in the figure) and one of its identified downstream gene/mRNA targets (rectangles)
- *Fitness-contribution hits*, shown in blue
- *Phospho-proteomic hits*, shown in yellow
- *Receptor-source pairs*, where necessary, which represent directed relationships gathered from the literature; receptors are shown in pink

We refer to the fitness-contribution hits and phospho-proteomic hits together as *hits*. The sets are not mutually exclusive.

**Step 2: Generate candidate paths to explain experimental observations.**

The inferred subnetwork must provide directed paths between each source and all of its targets, and so we first use the background network to enumerate all possible candidate paths. Each candidate source-target path must end with an interaction that represents regulation of the target mRNA: either the binding of a transcription factor (TF) to the target's gene, or the binding of an RNA-binding protein (RBP) to the target's mRNA. We also enumerate other kinds of candidate paths in order to capture connections among salt-responsive proteins that may be missed by cataloging only source-target paths. In Figure 5.2B, we show the three kinds of paths that we identify:

- *Source-target paths*, which connect interrogated sources to their targets

- *Fitness-contribution hit-source* and *source-source paths*, which identify connections between potential regulators
- *Receptor-source paths*, which offer paths between upstream stress sensors and downstream sources

**Candidate source-target paths.** To account for the effect of a source mutant on its dysregulated targets, we must provide directed candidate paths that begin with the source and end with an interaction that represents the binding of a TF (or RBP) to the target gene (or mRNA). In this section we describe our process for enumerating these paths.

*For sources that are TFs.* For the five sources that are themselves transcription factors (Dot6/Tod6, Msn2, Rim101, Swc3, and Swc5), we assume that they bind their targets directly. For these sources, each source-target path consists of only one interaction.

*For sources that are not TFs.* For the remaining sources that are not transcription factors, we enumerate longer paths. First, we identify which TFs and RBPs present in the background network could plausibly account for the effect of each source on its targets; we call these *candidate* TFs and RBPs. We consider a relationship between a source and a TF/RBP if a) the binding data for the TF/RBP was gathered under salt stress conditions (Ni *et al.*, 2009) or b) the set of genes bound by the TF/RBP are significantly over-represented in the source's targets. Over-representation is decided by a $p$-value $< 0.05$, given by a hypergeometric test of the overlap between the two sets. Because we are only assembling a candidate list of TFs/RBPs for each source, we do not correct for multiple testing at this step.

After identifying candidate TFs and RBPs for each source, we enumerate possible directed paths between all source-target pairs through the background network, terminating with an interaction between one of the candidate TFs/RBPs and the target. Because the experimental data represents transcriptional changes that happen on a short time scale, the interactions in the path (except for the last) are limited to either interactions between proteins or between an RBP and the protein node for an mRNA that it binds. In other words, we do not allow connections from target genes/mRNAs to proteins, which would require both transcription and subsequent translation of the protein encoded by the target.

We search for paths using an iterative deepening search, and limit the total number of interactions in a path for the sake of tractability. We first enumerate all paths of up to three interactions. If by doing so we are unable to reach at least 50% of the source's candidate TFs/RBPs, we search out until that goal percentage is reached, allowing up to five interactions total. Applying this process to our data, we reach on average 78% of

each source's targets and 75% of each source's candidate TFs/RBPs. Table 5.3 shows the coverage of each source's targets and candidate TFs/RBPs by the candidate paths.

**Candidate fitness-contribution hit-source and source-source paths.** In addition to explaining the effects of the sources on their targets, we are also interested in identifying other connections among salt-responsive proteins that may not lie along source-target paths: among fitness-contribution hits and sources, and among the sources themselves. We enumerate short candidate paths (up to two interactions) between each fitness-contri-bution hit and each source, and between each pair of sources. For each pair, we search for paths that proceed in both directions. We do not allow protein-DNA interactions in these paths, but do allow edges between RBPs and the proteins translated from their target mRNAs.

**Candidate paths to explain indirect relationships from domain knowledge (receptor-source paths).** In this step, we enumerate directed paths of up to five interactions that connect the two upstream receptors, Sho1 and Sln1, to Hog1. As we do for the hit-source and source-source paths, we only allow protein-protein and RBP-protein interactions in these paths.

The nodes and edges in the candidate paths are indicated in the Cytoscape files provided as supplementary material with the original publication. Experiments varying the lengths of these paths are discussed in Section 5.3.8.

**Step 3: Solve an IP to infer an ensemble of subnetworks.**

To perform inference, we construct an integer linear program that consists of a set of constraints and objective functions that describe a subnetwork that conforms to the following desiderata:

- Each source is connected to of all of its targets by directed paths
- Any provided upstream relationships (receptor-source pairs) are explained by directed paths
- Directed paths reveal connections between fitness-contribution hits and sources that are proximal in the background network
- Each edge is assigned only one direction
- Proteins with phospho-proteomic or fitness evidence are favored for inclusion in the subnetwork

- The subnetwork includes a minimal number of nodes that are not supported by experimental evidence

The IP itself is discussed in the following section (Section 5.2.5).

Because both the experimental data and the background network are incomplete, there are many possible inferred subnetworks that explain the experimental data equally well. To quantify our confidence in the relevance of each protein and interaction, we infer an ensemble of optimal subnetworks. Confidence in a protein (interaction, path) is calculated as the fraction of the subnetworks in the ensemble that predict that the protein (interaction, path) is relevant.

Figure 5.2C shows two inferred relevant subnetworks for the given example. In each subnetwork, the predicted relevant edges (interactions) and nodes (proteins) are indicated with black outlines; rejected edges and nodes are in grey. A direction has been inferred for each formerly undirected interaction that is predicted to be relevant.

**Table 5.3:** Coverage of each source's targets and candidate TF/RBPs by the candidate paths. 'Prop.' columns give the proportion of targets covered by candidate paths. Sources marked with the message 'Self' are sources that are themselves TFs or RBPs; for these sources, all targets were covered by the addition of inferred regulatory interactions.

| | Targets | | | Candidate TF/RBPs | | |
|---|---|---|---|---|---|---|
| Source | Covered | Total | Prop. | Covered | Total | Prop. |
| Gpb2 | 174 | 234 | 0.74 | 12 | 17 | 0.71 |
| Hog1 | 1843 | 1912 | 0.96 | 41 | 44 | 0.93 |
| Mck1 | 861 | 886 | 0.97 | 48 | 52 | 0.92 |
| Npr2 | 71 | 131 | 0.54 | 9 | 12 | 0.75 |
| Npr3 | 218 | 267 | 0.82 | 16 | 28 | 0.57 |
| Pde2 | 526 | 573 | 0.92 | 26 | 41 | 0.63 |
| Pph3 | 209 | 247 | 0.85 | 12 | 22 | 0.55 |
| Rim15 | 513 | 530 | 0.97 | 36 | 41 | 0.88 |
| Sub1 | 422 | 515 | 0.82 | 15 | 19 | 0.79 |
| Tpk1 | 53 | 124 | 0.43 | 6 | 10 | 0.6 |
| Whi2 | 167 | 296 | 0.56 | 8 | 9 | 0.89 |
| Dot6/Tod6 | – | 259 | – | – | Self | – |
| Msn2 | – | 209 | – | – | Self | – |
| Rim101 | – | 302 | – | – | Self | – |
| Swc3 | – | 365 | – | – | Self | – |
| Swc5 | – | 138 | – | – | Self | – |

### 5.2.5 IP notation and variables

The salt-specific signaling subnetwork is inferred by solving an integer linear program (IP, for short). We encode the relevance of each node, edge, and candidate path, and the direction of each edge, as binary variables. We characterize possible subnetworks using a set of linear constraints over those binary variables. Subnetwork inference is performed by choosing a union of relevant, directed paths that together satisfy our constraints and optimize a series of successively applied objective functions.

The values of some variables were determined by data provided as input to the inference process (for example, directions of directed edges), while others are inferred by solving the IP.

#### Notation (Table 5.4)

The input to the method is represented as a graph of nodes $\mathcal{N}$, edges $\mathcal{E}$, and candidate paths $\mathcal{P}$ . A node represents either a protein or a target gene/mRNA. Protein nodes may belong to one or more of the following subsets: sources $\mathcal{N}^S$, fitness-contribution hits $\mathcal{N}^F$, phospho-proteomic hits $\mathcal{N}^P$, and known membrane receptors $\mathcal{N}^R$. The set $\mathcal{N}^T$ describes targets, and for a given source node $n$, $\mathcal{N}^T(n)$ is the set of its targets.

The set of edges is $\mathcal{E} = (\mathcal{E}^D \cup \mathcal{E}^U)$, where $\mathcal{E}^D$ is the set of directed edges and $\mathcal{E}^U$ is the set of undirected edges. We denote an edge $e$ between nodes $n_i$ and $n_j$ as $e = (n_i, n_j)$. $\mathcal{N}(e)$ refers to the nodes connected by a particular edge $e$, and $\mathcal{E}(n)$ refers to the edges that touch a particular node $n$.

We consider four subsets of candidate paths $\mathcal{P}$: source-target paths between sources and their targets $\mathcal{P}^{ST}$, hit-source paths between fitness-contribution hits and sources $\mathcal{P}^{FS}$, source-source paths $\mathcal{P}^{SS}$, and receptor-source paths $\mathcal{P}^{RS}$ that connect known receptor proteins to sources. Phospho-proteomic hits and additional fitness-contribution hits may appear in any of these paths.

To refer to the paths between a specific source $s$ and target $t$, we use the notation $\mathcal{P}^{ST}(s,t)$. We use the same notation to refer to other kinds of paths with specific endpoints: $\mathcal{P}^{FS}(f,s)$, $\mathcal{P}^{SS}(s_i,s_j)$, $\mathcal{P}^{RS}(r,s)$.

Each path $p$ specifies a direction for each of its undirected edges $e$, which is denoted as $dir(p,e)$. $\mathcal{E}(p)$ and $\mathcal{N}(p)$ refer to the edges and nodes in a particular path $p$.

#### Variables (Table 5.5)

The predicted relevance of a path $p$ is represented with the variable $\sigma_p$ which takes the value 1 if the path is included in the inferred subnetwork, and 0 if it is not. As many as

**Table 5.4:** Sets of network elements that are provided as input to the method.

| Network elements | Set | Description |
|---|---|---|
| Nodes | $\mathcal{N}$ | All nodes |
| | $\mathcal{N}^S$ | Source nodes |
| | $\mathcal{N}^R$ | Receptor nodes |
| | $\mathcal{N}^F$ | Fitness-contribution hits |
| | $\mathcal{N}^P$ | Phospho-proteomic hits |
| | $\mathcal{N}^T$ | Target nodes |
| | $\mathcal{N}^T(s)$ | Dysregulated targets of source $s$ |
| | $\mathcal{N}^S(r)$ | Sources that are downstream of receptor $r$ |
| | $\mathcal{N}(e)$ | Nodes in edge $e$ |
| Edges | $\mathcal{E}$ | All edges |
| | $\mathcal{E}^D$ | Directed edges |
| | $\mathcal{E}^U$ | Undirected edges |
| | $\mathcal{E}(n)$ | Edges that touch node $n$ |
| Paths | $\mathcal{P}$ | All paths |
| | $\mathcal{P}^{ST}$ | Source-target paths |
| | $\mathcal{P}^{ST}(s,t)$ | Source-target paths between source $s$ and target $t$ |
| | $\mathcal{P}^{RS}$ | Receptor-source paths |
| | $\mathcal{P}^{RS}(r,s)$ | Receptor-source paths between receptor $r$ and source $s$ |
| | $\mathcal{P}^{FS}$ | fitness-contribution hit-source paths |
| | $\mathcal{P}^{FS}(f,s)$ | fitness-contribution hit-source paths between hit $f$ and source $s$ |
| | $\mathcal{P}^{SS}$ | Source-source paths |
| | $\mathcal{P}^{SS}(s_i,s_j)$ | Paths between source $s_i$ and source $s_j$ |

two variables describe each edge. The predicted relevance of an edge $e$ is represented with the variable $x_e$, which takes the value 1 if the edge is in at least one relevant path. For undirected edges in the background network, the variable $d_e$ represents the inferred direction of the edge. Each node $n$ has one variable: $y_n$, representing whether or not the node is present in any relevant paths. Finally, for all pairs of sources $(n_i, n_j)$, and also for all pairs consisting of one source and one fitness-contribution hit, the variable $c_{ij}$ represents whether or not the relevant subnetwork provides a directed path between the two nodes in the pair.

**Table 5.5:** Integer program variables. Binary variables represent the status of nodes, edges, paths, and pairs in the network.

| Network elements | Variable | Interpretation | Values |
|---|---|---|---|
| Paths $p$ | $\sigma_p$ | Relevant | no=0, yes=1 |
| Edges $e$ | $x_e$ | Relevant | no=0, yes=1 |
| | $d_e$ | Direction | back=0, forward=1 |
| Nodes $n$ | $y_n$ | Relevant | no=0, yes=1 |
| Pairs $(n_i, n_j)$ | $c_{ij}$ | Connected | no=0, yes=1 |

### 5.2.6 IP constraints

The following linear constraints define a subnetwork that, at minimum, provides consistently directed paths between source-target pairs and receptor-source pairs. Additional constraints are used to count up the number of connected fitness-contribution hit-source pairs and source-source pairs. These counts are optimized during the optimization procedure.

*Provide at least one path between each source-target pair.* Each source must be connected to each of its targets by at least one relevant path. The following constraint requires that, for each source $s$, for each of its targets $t$, at least one source-target path $p \in \mathcal{P}^{ST}(s,t)$ from $s$ to $t$ must have $\sigma_p = 1$.

$$\forall s \in \mathcal{N}^S, t \in \mathcal{N}^T(s) \qquad \sum_{p \in \mathcal{P}^{ST}(s,t)} \sigma_p \geq 1$$

*Provide at least one path between each receptor-source pair.* Similar to the previous constraint, this one requires that for each receptor $r$ and each of its downstream sources $s$ there must be at least one receptor-source path $p \in \mathcal{P}^{RS}(r, s)$ for which $\sigma_p = 1$.

$$\forall r \in \mathcal{N}^R, s \in \mathcal{N}^S(r) \qquad \sum_{p \in \mathcal{P}^{RS}(r,s)} \sigma_p \geq 1$$

*Record whether or not there is a path between each fitness-contribution hit-source pair and source-source pair.* Rather than require that each of these pairs is connected, we use the optimization procedure to maximize the total count of connected pairs. We use the following constraints to count up the number of connected pairs.

If there is a relevant path between a fitness-contribution hit $f$ and a source $s$, set the variable $c_{fs} = 1$. If there are no paths, set it to 0.

$$\begin{aligned} &\forall f \in \mathcal{N}^F, s \in \mathcal{N}^S, && c_{fs} \geq \sigma_p \\ &\quad p \in \left(\mathcal{P}^{FS}(f, s) \cup \mathcal{P}^{FS}(s, f)\right) && \\ &\forall f \in \mathcal{N}^F, s \in \mathcal{N}^S && c_{fs} \leq \sum_{p \in (\mathcal{P}^{FS}(f,s) \cup \mathcal{P}^{FS}(s,f))} \sigma_p \end{aligned}$$

Similarly, if source $s_i$ is connected to source $s_j$ by at least one relevant path, set $c_{ij} = 1$; otherwise, set it to 0.

$$\begin{aligned} &\forall (s_i, s_j) \in \mathcal{N}^S \times N^S && c_{ij} \geq \sigma_p \\ &\qquad p \in \left(\mathcal{P}^{SS}(s_i, s_j) \cup \mathcal{P}^{SS}(s_j, s_i)\right) && \\ &\forall (s_i, s_j) \in \mathcal{N}^S \times N^S && c_{ij} \leq \sum_{p \in (\mathcal{P}^{SS}(s_i,s_j) \cup \mathcal{P}^{SS}(s_j,s_i))} \sigma_p \end{aligned}$$

*All edges in a relevant path are relevant.* For an edge $e$ to be relevant (that is, have $x_e = 1$), there must be at least one relevant path $p$ (having $\sigma_p = 1$) that contains it. Similarly, a relevant path $p$ must contain all relevant edges. The set $\mathcal{P}(e)$ refers to the paths that contain

$e$.

$$\forall e \in \mathcal{E} \qquad x_e \leq \sum_{p \in \mathcal{P}(e)} \sigma_p$$

$$\forall p \in \mathcal{P}, e \in \mathcal{E}(p) \qquad \sigma_p \leq x_e$$

*All nodes in a relevant edge are relevant.* A node $n$ is relevant only if it touches at least one relevant edge $e$. An edge $e$ can only be relevant if each of its nodes $n$ is relevant.

$$\forall n \in \mathcal{N} \qquad y_n \leq \sum_{e \in \mathcal{E}(n)} x_e$$

$$\forall e \in \mathcal{E}, n \in \mathcal{N}(e) \qquad x_e \leq y_n$$

*All paths must be uniquely directed.* For a relevant path $p$, all undirected edges in that path ($e \in \mathcal{E}(p) \cap \mathcal{E}^U$) must be uniquely oriented so that the path proceeds only in one direction. This required direction for each edge is determined when the candidate path is generated, and is given by $dir(p, e)$. For source-target paths, the required direction allows the path to proceed from the source to the target. The term including $I(\cdot)$, the indicator function, returns 1 if an edge's inferred direction corresponds to the direction that the path requires for it.

$$\forall p \in \mathcal{P}, e \in \mathcal{E}(p) \cap \mathcal{E}^U \qquad \sigma_p \leq I(d_e = dir(p, e))$$

**Solving the IP to find an ensemble of subnetworks**

An optimal inferred subnetwork satisfies two goals: it maximizes the inclusion of salt-response-relevant proteins that are supported by experimental evidence, and minimizes the number of additional nodes that are necessary for connecting each source to each target. To achieve this, we apply four successive objective functions. To accompany the following description, a diagram of the process is depicted in Figure 5.3.

To model and solve the IP, we used the GAMS modeling system v. 23.9.3 and the ILOG CPLEX solver v. 12.4.0.1. We provide our GAMS code as supplementary material to the original publication.

*Step 1: Maximize connections between hits and sources.* This involves solving the IP to identify `max_connections`, the maximum number of connections possible between pairs of sources, and between pairs of fitness-contribution hits and sources. The purpose of this step is to reveal proximal connections between salt-responsive proteins, whether or not they occur between sources and targets. The set $\left((\mathcal{N}^S \times \mathcal{N}^S) \cup (\mathcal{N}^F \times \mathcal{N}^S)\right)$ gives all source-source pairs and fitness-contribution-hit-source pairs, and the sum counts up the number of pairs that are connected by relevant paths.

$$\texttt{max\_connections} = \max \sum_{(n_i,n_j)\in((\mathcal{N}^S\times\mathcal{N}^S)\cup(\mathcal{N}^F\times\mathcal{N}^S))} c_{ij}$$

After optimizing for this criterion, we add a new constraint to the IP:

$$\sum_{(n_i,n_j)\in((\mathcal{N}^S\times\mathcal{N}^S)\cup(\mathcal{N}^F\times\mathcal{N}^S))} c_{ij} = \texttt{max\_connections}$$

*Step 2: Maximize inclusion of fitness and phospho hits.* Next, we solve the IP to identify `max_hits`, the maximum number of fitness-contribution hits and phosphor-proteomic hits that can be included in the relevant subnetwork. This step prioritizes the use of nodes with experimental evidence of being relevant to the salt stress response.

$$\texttt{max\_hits} = \max \sum_{n\in(\mathcal{N}^F\times\mathcal{N}^P)} y_n$$

After identifying the maximum number of hits that can be included in the subnetwork, subject to existing constraints, we add a new constraint to the IP:

$$\sum_{n\in(\mathcal{N}^F\times\mathcal{N}^P)} y_n = \texttt{max\_hits}$$

*Step 3: Minimize total nodes and find multiple solutions.* Now we solve the IP with a new objective function, which minimizes the number of nodes required to satisfy all of the constraints. The resulting subnetwork includes only those additional nodes that are required

to explain the experimental data.

$$\min \sum_{n \in \mathcal{N}} y_n$$

At this point, we find an ensemble of solutions to the IP, where each solution identifies a minimum set of nodes (while still satisfying all other constraints). The CPLEX solver allows for the identification of multiple solutions. First, the CPLEX solver uses a branch-and-cut algorithm to find one optimal solution; this algorithm entails maintaining a tree of linear relaxations of the IP. Next, the solver proceeds down previously rejected branches of the tree to identify additional optimal solutions with different variable settings. For our experiments, we identified 10,000 solutions. To assess this choice, we performed experiments varying the ensemble size; results are reported in Section 5.3.9.

*Step 4: Maximize the number of paths in each solution.* For each of the solutions identified in the previous step, we solve the IP again to maximize the number of relevant directed paths between the nodes included in the solution. This step does not change the node content of each solution, but instead reveals all possible directed paths that connect the nodes chosen in the previous step.

**For each solution:**

- First, we introduce constraints to fix each value of $y_n$ to its value from the previous solution, $\hat{y_n}$.

$$\forall n \in \mathcal{N} \qquad\qquad y_n = \hat{y_n}$$

- Next, we solve the IP to maximize the number of relevant paths:

$$\max \sum_{p \in \mathcal{P}} \sigma_p$$

At this point, we assemble the solutions into an ensemble of inferred subnetworks. Using the ensemble, we assign a confidence value for each prediction based on the number of solutions in the ensemble that support the prediction. We performed several experiments to assess the effect of each component of our four-part objective function, as well as their ordering. These can be found in Section 5.3.10.

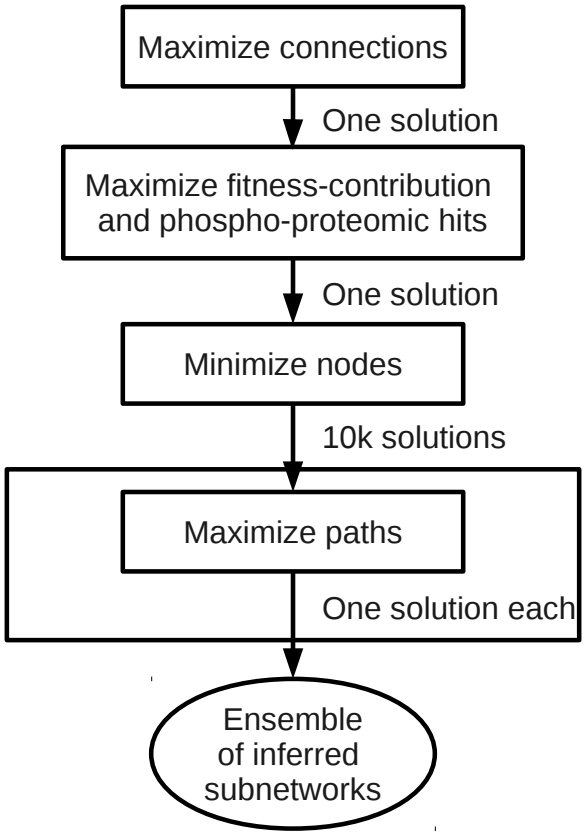



**Figure 5.3:** Diagram of the procedure for optimization in the IP.

## 5.3 Results

We use the ensemble of subnetworks inferred by the IP method to make predictions about different aspects of the salt-specific subnetwork. We inspect different sets of network elements depending on the prediction task. By applying a threshold on confidence values represented by the ensemble, we defined three high-confidence, or consensus, subnetworks:

- **Consensus nodes**, defined as the set of nodes with at least 75% confidence; used to make predictions about the identity of novel regulators of the salt response
- **Consensus edges**, defined as the set of edges with at least 75% confidence; used to make predictions about high-confidence direct interactions between consensus nodes

Unless specified otherwise, the unqualified *consensus subnetwork* refers to the set of consensus edges, which encompasses 380 nodes (predicted regulators) and 1131 edges (relevant interactions). This consensus subnetwork is depicted in Figure 5.4A.

### 5.3.1 Precision-recall analysis

To assess the predictive accuracy of the inferred consensus subnetwork, we assembled a list of known NaCl regulators (true positives) and another list of unlikely regulators (likely negatives) that included metabolic enzymes and exclusively subcellular proteins.

We defined true positives as genes that have been previously identified as relevant to the Hog signaling pathway based on literature curation (de Nadal & Posas, 2010; Tiger *et al.*, 2012), genes with 'osmotic' or 'osmolarity' in their Saccharomyces Genome Database (SGD) (Cherry *et al.*, 2012) annotations, and genes with 'stress regulator' in their SGD annotations, if they are also linked to the osmotic response in at least one publication. In all, this identifies 112 true positives.

Likely negatives are defined as genes with no evidence for nuclear localization and whose GO compartment annotation is 'mitochondrion', 'mitochondrial envelope', 'peroxisome', 'vacuole', 'Golgi', and/or 'endoplasmic reticulum'. We add to this list all proteins annotated in SGD as 'metabolic enzymes'. We then remove 32 well-known signaling proteins, many of which are already on the true positive list. This process identifies 1,865 likely negative proteins for the network assessment.

Among these test cases, the background network contains 108 positives and 1512 likely negatives. In order to separate out the effect of the experimental hits on predictive accuracy, we omitted all fitness contribution and phospho-proteomic hits from the test cases, leaving 70 true positives and 1416 likely negatives.

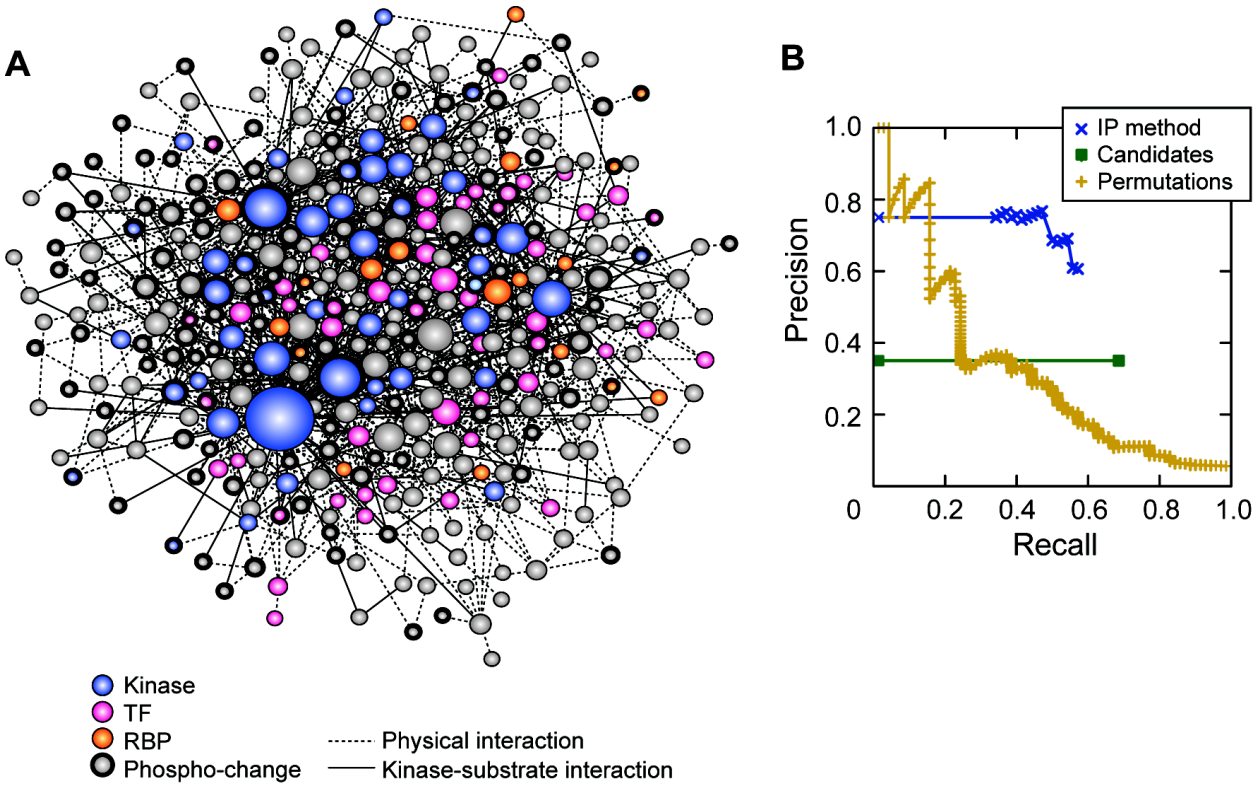



**Figure 5.4:** Inferred NaCl-activated signaling network and precision-recall curves.
**A** Inferred consensus subnetwork at 75% confidence, where node size indicates degree (number of connections) and color represents kinases (blue), TFs (pink), and RBPs (orange) according to the key. Nodes representing proteins with phospho-changes are outlined in bold. Figure contributed by Audrey Gasch.
**B** Precision-recall curves. The blue curve represents the performance of our IP method. The green curve indicates the performance of all candidate enumerated paths (i.e. the input to the IP method), and the yellow curve represents the performance of the method on randomized permutations of the input network.

For each test case (true positive or likely negative), we measure the inferred subnetwork ensemble's confidence that it is relevant to the salt response. This is calculated as the fraction of the 10,000 solutions in which the test case appears as a protein node in the subnetwork.

We plot a precision-recall curve by ranking the test cases and calculating *precision* and *recall* at decreasing confidence thresholds. Precision is defined as the fraction of true positives (known NaCl regulators) out of all test cases that are above the threshold, and recall is defines as the fraction of the total true positives that are above the threshold.

We compare our ensemble's precision-recall curve to two baselines, which we refer to as the *candidate* baseline and *permuted* baseline:

- The *candidate* baseline represents the accuracy of enumerating candidate paths without performing inference. For this baseline, we compute the total precision and recall of the test cases using the complete set of protein nodes present in candidate paths.

- The *permuted* baseline assesses the degree to which the predictive accuracy is merely a result of topological properties in the background network. For this baseline, we infer 1,000 ensembles using permuted experimental data. We generate the permuted data as follows. For each of 1,000 permutations, we randomly draw a set of sources, proteins with fitness defects, and proteins with phospho-changes from the background network, equal in number and degree distribution to the true experimental data. To generate receptor-source pairs, we randomly draw two proteins from the background network and pair each with a randomly chosen source. To generate permuted source-target pairs, for each source, we randomly draw an equal number of targets from the entire background network. Now, for each permutation of the data, we infer an ensemble of 1,000 solutions, and measure the confidence of each test case as the average confidence over all 1,000 ensembles.

Figure 5.4B presents the precision-recall curves for the inferred ensemble and the tested baselines. The inferred ensemble (blue curve) achieves substantially higher accuracy than the enumerated candidate paths provided as input to the IP method (green line). This highlights the power of the inference process over simply enumerating all paths.

The permuted baseline (yellow curve) achieves high accuracy in the low-recall range, suggesting that some regulators are highly central in the background network. However, our inferred ensemble significantly outperforms the permuted baseline at higher levels of recall; thus, our method's accuracy is not simply due to properties of the background network's topology.

### 5.3.2 Enrichment of the consensus node subnetwork with known relevant proteins

Because our curated list of true positives is known to be incomplete, we additionally evaluate our consensus protein node set (excluding target genes/mRNAs) against several external sources of likely-relevant genes and interactions. We gathered two sets of genes that are related to cell signaling and stress: a) kinases and phosphatases, and b) proteins retrieved from SGD's YeastMine (Balakrishnan *et al.*, 2012; Cherry *et al.*, 2012) with the queries 'stress regulation' or 'nutrient regulation'. We also test for enrichment with essential genes from SGD (Cherry *et al.*, 2012), and for genetic interactions from BioGRID (Stark *et al.*,

2006). We also test for over-representation of the true positives and under-representation of the likely negatives that we gathered for the precision-recall analysis.

Using the hypergeometric test, we test for the enrichment of relevant gene sets in the consensus node subnetwork relative to the candidate network (the network given by the set of candidate paths), the background network, and the consensus subnetworks inferred from permuted data. To separate out the effect of experimental hits on the enrichment score, we exclude them from the consensus nodes and the external gene sets. After filtering, 160 consensus nodes, 736 candidate nodes, and 4703 background network nodes remain. With experimental hits removed, the external gene sets consist of 147 kinases and phosphatases, 237 general stress proteins, 910 essential genes, 70 true positives, and 1416 likely negatives.

Additionally, we test for enrichment of genetic interactions based on the assumption that genetic interactions are more likely to occur between functionally related genes. For this test, we do not omit experimental hits. We extracted 141,507 genetic interactions reported in BioGRID (Stark *et al.* (2006), downloaded February 2013) and assume a total of 16 million possible interactions among 5,700 yeast genes. Enrichment for genetic interactions is calculated using the number of pairs of nodes in the subnetwork that have a reported genetic interaction.

We compare the consensus node subnetwork against the permuted subnetworks as follows. For each ensemble inferred from permuted data, we define a consensus node set using a 75% confidence threshold, and, for each external gene/interaction set, measure the proportion of each consensus node set that is contained in the external set. For each external gene set, we calculate the $p$-value as the fraction of the 1,000 permuted node sets that have an equal or higher proportion than the true consensus node set. (For the likely negatives, we count the number of permuted consensus node sets with equal or lower representation.) We do not filter out hits for this experiment, as the hit sets are different for each permutation.

We find support for the consensus node subnetwork in the non-random inclusion of specific protein functional groups. When compared to the background network, to the enumerated candidate pathways used as input to the IP, and to the permuted subnetworks, the inferred consensus node subnetwork is enriched for proteins annotated as stress proteins (background $p$ = 5e-21, candidate $p$ = 2e-6, permutations $p$ = 0.007). It is also enriched for genetic interactions (background and candidate $p \approx 0$; permutations $p$ = 0.003), which suggests that the consensus proteins may have a higher rate of functional similarity. The consensus node subnetwork is also slightly enriched for kinases (relative to the candidate paths and background network) and for essential genes (relative to the background network). However, it is not enriched for kinases or essential genes relative to the permuted

subnetworks, suggesting a bias in the background network for connectivity of kinases and essential genes.

In Table 5.6, we show the proportion of the consensus subnetwork, candidate network, background network, and permuted subnetworks that is represented in each external gene set, and the $p$-values for comparisons to the consensus node subnetwork.

**Table 5.6:** Enrichment analysis results. The 'Prop.' columns give the proportion of each subnetwork that is in the relevant gene or interaction set; for the 'Permutations' row, this value is the average over all 1,000 permutations. The '$p$-value' columns are calculated by a comparison to the consensus subnetwork (derived from the 10,000-solution ensemble). Asterisks (*) indicate a result that is significant at $p < 0.05$. For comparisons to the candidate and background networks, $p$-values are calculated using the hypergeometric test. For comparisons to the permuted ensembles, $p$-values are calculated as the fraction of permutations having an equal or greater proportion of relevant genes (or lower propoerion of likely negatives) compared to the consensus subnetwork.

| | Relevant gene sets | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | True positives | | Likely negatives | | Kinases and phosphatases | | General stress proteins | | Essential genes | | Genetic interactions | |
| Subnetwork | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value |
| Consensus nodes | 0.156 | | 0.050 | | 0.200 | | 0.269 | | 0.294 | | 0.051 | |
| Candidate network | 0.065 | 2e-6 * | 0.121 | 2e-4 * | 0.124 | 0.001 * | 0.145 | 2e-6 * | 0.319 | 0.810 | 0.033 | $\approx 0$ * |
| Background network | 0.015 | 5e-20 * | 0.302 | 6e-17 * | 0.032 | 8e-18 * | 0.050 | 5e-21 * | 0.194 | 0.001 * | 0.008 | $\approx 0$ * |
| Permutations | 0.072 | 0.002 * | 0.133 | 0.007 * | 0.183 | 0.309 | 0.185 | 0.007 * | 0.348 | 0.899 | 0.035 | 0.003 * |

### 5.3.3 Experimental validation of predicted regulators and targets

The inferred subnetwork includes many predicted genes that have not previously been linked to the NaCl response. We refer to these genes as *predicted regulators*. To test some of the novel predictions, we compare the consensus path subnetwork's predictions to actual measurements of osmo-dependent transcriptome changes in fourteen mutants lacking the predicted regulators. The Gasch lab chose the fourteen predicted regulators with a preference for kinases and phosphatases. They include a) proteins that are activated by other stresses, but which have not yet been implicated in the NaCl response (Yak1, Bck1, Pho85), b) proteins with little known function in stress-dependent gene regulation (Cdc14, Kin2, Nnk1, Scd6, Arf3), and c) several others that are either poorly characterized or for which other datasets exist to test downstream effects (CK2 subunits Cka2, Cka1, Ckb1/2, Tpk2, and Bem1).

The Gasch lab assayed NaCl-responsive gene expression in ten mutants, focusing on kinases and phosphatases not known to respond to NaCl, and two RBPs (Scd6 and Arf3). As before, targets are identified at $q < 0.05$ from `limma` $q$-value analysis. A summary of the identified targets for these new sources is given in Table 5.7. We also acquire similar data for Rpd3, Bem1, Gal11, and Tpk2 from previous studies probing the osmotic response (Gitter & Bar-Joseph, 2013; Alejandro-Osorio *et al.*, 2009).

These experiments produced additional source-target pairs, which we compare to the predictions made by the consensus paths subnetwork. We assess the accuracy of up to three types of predictions about each predicted regulator:

- *Defective NaCl response.* First, we test whether the mutant exhibits a defective transcriptome response. A defective response is defined by a significant defect in at least fifty genes.

- *Downstream targets.* Next, we test whether the measured targets of the mutant overlap significantly with the targets predicted by the consensus paths. This is scored by hypergeometric test, assuming a total population of 3,330 genes. For mutants whose predicted targets do not significantly overlap with measured targets, we also measure the overlap between the NaCl-predicted targets and gene targets affected in an unstressed mutant.

- *In-path relationships.* Finally, we test whether the mutant significantly shares targets with any other interrogated nodes that are predicted to be in the same paths. From the nodes predicted to lie in each regulator's path (based on the consensus-paths network), we identify those that are a) either TFs or RBPs with known direct binding targets or

b) have NaCl-responsive downstream targets measured in this study. We then score the enrichment of each predicted node's known targets within the measured targets of the interrogated regulator, taking $p <$ 1e-6 from the hypergeometric test as significant. Because the test lacks statistical power for large gene groups, we score enrichment against the total list of measured targets, as well as separate lists of induced and repressed targets with defective or amplified expression changes. This gives a lower bound for the number of supported in-path nodes, since the hypergeometric test has lower statistical power for small gene groups (including known targets of several regulators).

The accuracy of the predictions for each new interrogated mutant are provided in Table 5.8.

- *Defective NaCl response.* All but one of the mutants (93%) display a defect in osmo-responsive expression. The exception is the mutant lacking Scd6, a gene that is implicated in translational regulation and binds HOG-encoding transcripts (Tsvetanova *et al.*, 2010; Iwaki & Izawa, 2012).

- *Downstream targets.* Furthermore, the predicted targets of 80% of these regulators overlap significantly ($p <$ 1e-3) with their measured targets, highlighting the accuracy of regulator-target predictions.

- *In-path relationships.* To garner support for the subnetwork's structure, we investigate the overlap between genes affected in each interrogated mutant and targets of signaling proteins predicted to lie in the interrogated regulator's paths. Using stringent scoring, we find support for 30-100% of nodes in most paths (53% on average). The targets of several of the in-path nodes are marginally enriched (1e-5 $< p <$ 0.01) among measured targets of interrogated regulators, even though they do not meet our stringent threshold. It is also possible that regulators that serve redundant roles are difficult to identify, since single-gene knockouts may not identify all of the downstream targets.

Together, these results provide strong support for the validity of the inferred consensus subnetwork.

**Table 5.7:** Gene targets identified in validation mutants. 'Targets' columns give the number of genes with smaller ('defective') or larger ('amplified') expression changes compared to the wild type strain. Note this table includes non-coding RNAs that were excluded from the inference. * *cdc14-3* was compared to its isogenic wild type. Table contributed by Audrey Gasch.

| | | Targets | |
|---|---|---|---|
| Mutant | Replicates | Defective | Amplified |
| *cdc14-3** | 3 | 929 | 346 |
| *nnk1*Δ | 1 | 94 | 278 |
| *bck1*Δ | 1 | 107 | 169 |
| *yak1*Δ | 1 | 226 | 248 |
| *kin2*Δ | 1 | 52 | 266 |
| *pho85*Δ | 1 | 614 | 342 |
| *cka2*Δ | 2 | 155 | 63 |
| *cka1*Δ | 2 | 58 | 133 |
| *ckb1*Δ*ckb*Δ | 2 | 129 | 176 |
| *arf3*Δ | 2 | 466 | 331 |
| *scd6*Δ | 2 | 0 | 0 |

**Table 5.8:** Validation of predicted regulators.
* = data from Gitter *et al.* (2013). ^ T0 = unstressed mutant. + = data from Alejandro-Osorio *et al.* (2009). NA = no targets predicted, by nature of the node inclusion. Table contributed by Audrey Gasch.

| Predicted regulators | Defective NaCl transcriptome response | Significant overlap in measured vs. predicted targets ($p$-value) | Number of in-path validated nodes | Number of scorable in-path nodes | Percent validated in-path nodes |
|---|---|---|---|---|---|
| Arf3 | Yes | Yes ($p$ =6e-5) | 2 | 2 | 100% |
| Bck1 | Yes | NA | 2 | 4 | 50% |
| Bem1* | Yes | NA | 1 | 4 | 25% |
| Cdc14 | Yes | Yes (2e-14) | 14 | 18 | 78% |
| Cka2 | Yes | Yes (3e-3) | 7 | 10 | 70% |
| Ckb1/2 | Yes | No (1); overlap with Cka2 targets ($p$ =3e-12) | 8 | 21 | 38% |
| Gal11* | Yes | Yes ($p$ =2e-4) | 5 | 10 | 50% |
| Kin2^ | Yes | No ($p$ =0.57); marginal overlap with T0-affected ($p$ = 0.005) | 3 | 17 | 18% |
| Nkk1 | Yes | No ($p$ =1) | | | |
| Pho85 | Yes | Yes ($p$ =5e-8) | 14 | 21 | 67% |
| Rpd3+ | Yes | NA | | | |
| Scd6 | No | | | | |
| Tpk2* | Yes | Yes ($p$ =5e-20) | 10 | 24 | 42% |
| Yak1 | Yes | Yes ($p$ =1e-10) | 5 | 10 | 50% |

### 5.3.4 Interconnectivity among new and novel players in the inferred signaling subnetwork

Our collaborators explored the inferred consensus subnetwork for new insights into stress signaling. The inferred subnetwork captures known and new players, and suggests interconnectivity between pathways. Many expected pathways are captured, including the canonical HOG, PKA, and TOR pathways. The consensus subnetwork includes members of other stress-activated pathways not previously linked to the NaCl response, such as PKC, Pho85, Rim15 pathways and GSK-3 kinase Mck1 (Figure 5.5A). The Coon and Gasch labs tested the involvement of these pathways by analyzing phospho-proteome changes and mutant transcriptome profiles. They found that members of all of these pathways show NaCl-dependent phospho-changes, and cells lacking specific pathway members (including *BCK1, YAK1, PHO85, RIM15,* and *MCK1*) had defects in NaCl-dependent expression changes.

The subnetwork also includes the STE mating pathway, which shares upstream components with the Hog network and is known to be suppressed by Hog1 signaling (O'Rourke & Herskowitz, 1998; Shock *et al.*, 2009; Patterson *et al.*, 2010; McClean *et al.*, 2007; Zarrinpar *et al.*, 2004; Marles *et al.*, 2004; Nagiec & Dohlman, 2012). The inclusion of the mating pathway indicates that some connections in the consensus subnetwork could represent signaling suppression that prevents crosstalk to other pathways.

The structure of the subnetwork reveals surprising cross-connectivity between previously defined pathways. To examine the connections suggested by the subnetwork, our collaborators first identified pathways with members included in the consensus subnetwork, and defined membership of the pathways based on the literature. Next, we counted the number of direct interactions in the background network between the predicted relevant members of those pathways (Figure 5.5B). We use the background network edges, rather than the inferred subnetwork edges, for this experiment because our focus is to study direct interactions between the regulators rather than their relationship to downstream effects under salt stress.

Many of the pathways appear to be intricately connected, with Tor1 and PKA pathways linked to the greatest number of other pathways. We also identified the individual nodes that themselves are connected to many pathways. We define integration points as the nodes with the greatest number of connections to distinct pathways (Figure 5.5C). Nearly half of the top ten integration nodes are kinases or phosphatases, including Mck1 and cell-cycle regulator Cdc28 (which regulates RP genes under optimal conditions (Chymkowitch *et al.*, 2012) but is suppressed during osmotic shock (Alexander *et al.*, 2001; Bellí *et al.*, 2001;

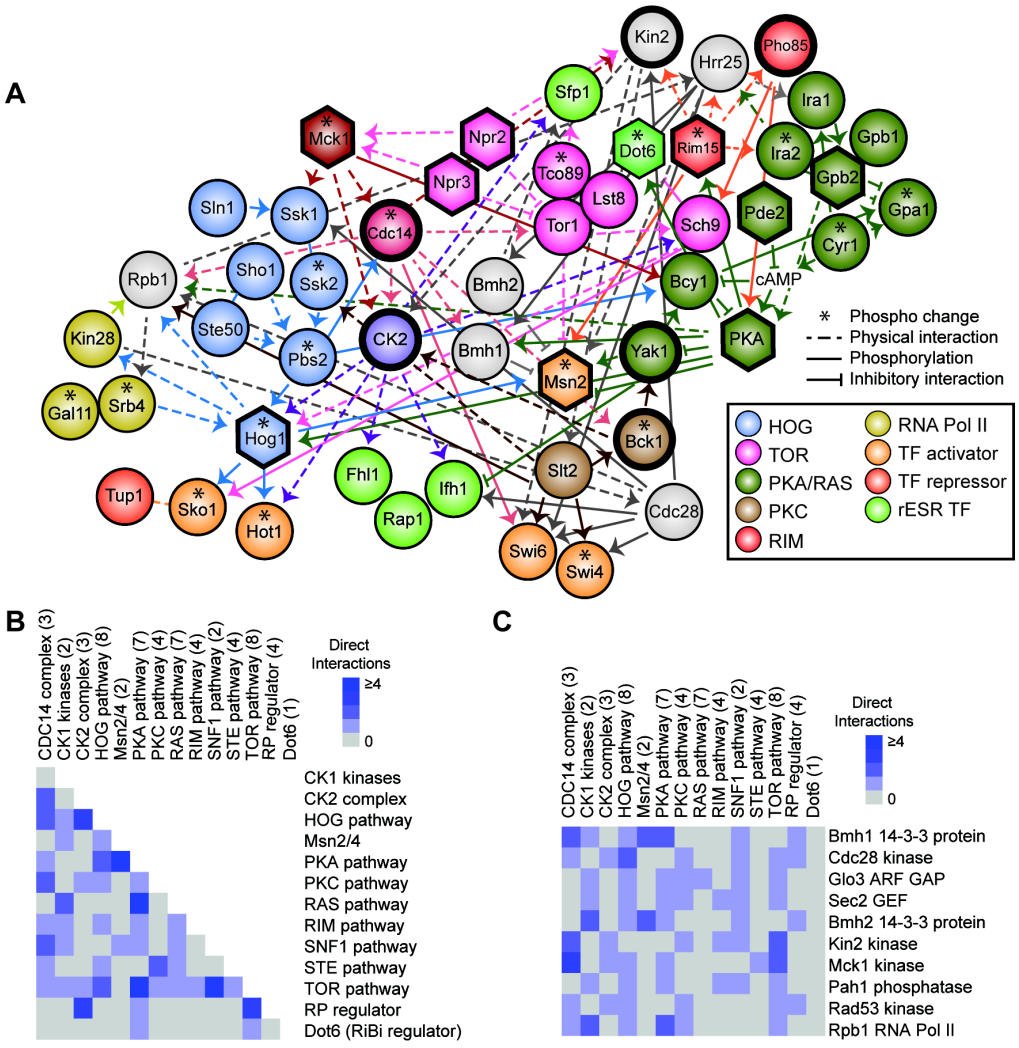



**Figure 5.5:** Connectivity between known pathways. Figure contributed by Audrey Gasch.

**A** A graphical representation of a subregion of the inferred subnetwork, highlighting proteins in known pathways according to the key. Hexagons represent source regulators profiled for input to the IP inference, nodes outlined in bold indicate validated players in the NaCl response, and asterisks represent proteins with phospho-changes upon NaCl treatment. Dashed edges represent physical interactions and solid arrows indicate kinase-substrate relationships. Edge directionality is as predicted by the inference, and edge color is according to the edge's source node. Inhibitory edges were taken from the literature.

**B** Connectivity between known pathways, where the blue color represents the number of interactions (including interactions from the background network) between any members of two pathways. Pathway membership is indicated in parentheses.

**C** The top 15-ranked integrator nodes with connections to the greatest number of different pathways, as shown in panel **B**.

Adrover *et al.*, 2011)). The 14-3-3 proteins Bmh1 and Bmh2 are also included, confirming their known role as signaling cofactors.

Several of the integration points are also hubs of high degree within the consensus subnetwork. While 11 of the top 15 most connected (highest degree) nodes are kinases or phosphatases, the remaining four are known regulatory cofactors – including stress-activated ubiquitin (Ubi4), Sumo (Smt3), and Bmh1 – and the core subunit of RNA polymerase (Pol II), Rpb1. Modification of the Rpb1 carboxyl-terminal domain (CTD) is the basis for the so-called CTD code of transcriptional regulation (Buratowski, 2003; Zhang *et al.*, 2012), making it a logical downstream integration point for complex upstream signaling.

Our collaborators performed additional experimental work to probe some of the predicted relevant nodes and relationships shown in the consensus subnetwork. We summarize the results here, and a complete account is available in the original publication.

**Cdc14.** The Cdc14 phosphatase is a key regulator of mitotic exit with little known role in stress responses. However, the inferred consensus subnetwork predicts that it is involved in the salt stress response. Cdc14 was previously shown to interact with Tor1 and stress-regulated kinases, hinting that it dampens stress signaling (Breitkreutz *et al.*, 2010). The Gasch lab tested a *cdc14-3* mutant in the validation experiments described previously in Section 5.3.3, and observed that the gene suppression results in a significant defect in NaCl-dependent gene induction and repression compared to the identically treated isogenic wild type. Furthermore, the subnetwork's predictions about its gene targets and the targets of other nodes in Cdc14's paths were well-supported by the measured targets.

To further explore the prediction that Cdc14 is an integration point, the Gasch lab performed additional experiments focusing on Cdc14's predicted regulatory connections. In the process, they found that Cdc14 is critical for coordinating distinct facets of the NaCl response.

**CK2 kinase complex.** The Gasch lab also performed further experiments into the role of the CK2 kinase complex, which regulates the iESR repressor Nrg1 and is known to affect stress-specific splicing and abundance of ribosome-related transcripts (Berkey & Carlson, 2006; Rudra *et al.*, 2007; Bergkessel *et al.*, 2011). The Gasch lab found that cells lacking either catalytic subunit (Cka1 or Cka2) or both regulatory subunits (Ckb1 and Ckb2) display a defective NaCl response, but that the defect differs for each mutant: *cka2*$\Delta$ cells display a defect in iESR gene induction, whereas the *cka1*$\Delta$ and *ckb1*$\Delta$*ckb2*$\Delta$ mutants instead produce amplified repression of rESR genes. The gene targets predicted by the subnetwork significantly overlap with the measured targets, as well as do the targets of many of the scorable nodes in their predicted paths (reported in Table 5.8).

**The Rpb1 CTD.** Our collaborators perfomed additional experiments to test the subnetwork's prediction that proteins interacting with Rpb1 might represent novel Rpb1-CTD kinases. Their results confirm the Rpb1-CTD as an important signaling target involved in regulating transcription under stress.

Thus, the inferred consensus subnetwork allowed our team to identify new players in the salt response.

### 5.3.5 New insights into ESR regulation and coordination

The Gasch lab is especially interested in how distinct modules in the ESR – including the induced ESR (iESR) and two repressed ESR modules (rESR) modules, RP and RiBi – are regulated and coordinated. Here we present a method for querying the consensus subnetwork to predict which nodes are most important for orchestrating the signals that control the modules. We evaluate our predictions using a comparison to the literature.

#### A salt-relevant ESR consensus subnetwork

We use the consensus paths subnetwork to define a salt-relevant ESR consensus paths subnetwork as follows. This subnetwork predicts which ESR module (or modules) are regulated by each node.

First, we gather three clusters of genes, defined by Gasch *et al.* (2000), based on expression profiles under multiple stress conditions: the iESR and the two rESR sub-clusters, RiBi and RP. Using the protein-nucleic acid interactions from the background network, we identify potential transcriptional regulators of the three ESR gene clusters. These are TFs and RBPs whose targets are enriched for a cluster as determined by hypergeometric test (significant at FDR=0.1, calculated by the Benjamini-Hochberg procedure). For iESR targets, we identify 25 total potential TFs/RBPs, of which 22 are TFs and three are RBPs. We find 16 TFs and 10 RBPs for the combined rESR clusters.

Next, we extract the consensus source-target paths (having confidence $\geq 75\%$) that end in an interaction between an ESR-relevant TF/RBP and ESR-relevant target gene (of the same cluster). For each protein node in each ESR-relevant consensus path, we assign a label based on the ESR cluster(s) represented by the downstream ESR-relevant TF/RBPs. These labels are used to perform the coloring in Figure 5.6. Finally, we remove the targets that were not a member of any ESR cluster.

**Summary.** Of the 178 nodes thereby implicated in ESR regulation, over half are predicted (Figure 5.6A), and several confirmed, to lie upstream of all three ESR modules. In contrast to common upstream nodes, which are enriched for kinases compared to the consensus

subnetwork ($p$ = 2.6e-7), the nodes that are predicted to be exclusive to iESR regulation are enriched for TFs ($p$ = 5e-5), while rESR regulators show a preponderance of RBPs ($p$ = 1e-5), implicating regulated RNA stability for these genes. Many more regulators and regulatory connections are unique to the iESR versus RP and RiBi modules, the latter being the largest group (Figure 5.6B). This is consistent with the extensive redundancy in iESR control (Gasch, 2003) and hints at a more monolithic regulation of rESR expression during times of adversity. Intriguingly, many direct regulators of iESR genes fall in the upstream paths of RP regulation, but most are not direct binders of RP promoters or transcripts (Figure 5.6B, nested circle). One possibility is that these TFs are artifactually predicted in RP paths to accommodate missing physical interactions in the background network. However, it is notable that the *msn2*$\Delta$ mutant responding to NaCl displays a subtle but significant defect in expression across the group of RP genes, supporting a more direct effect.

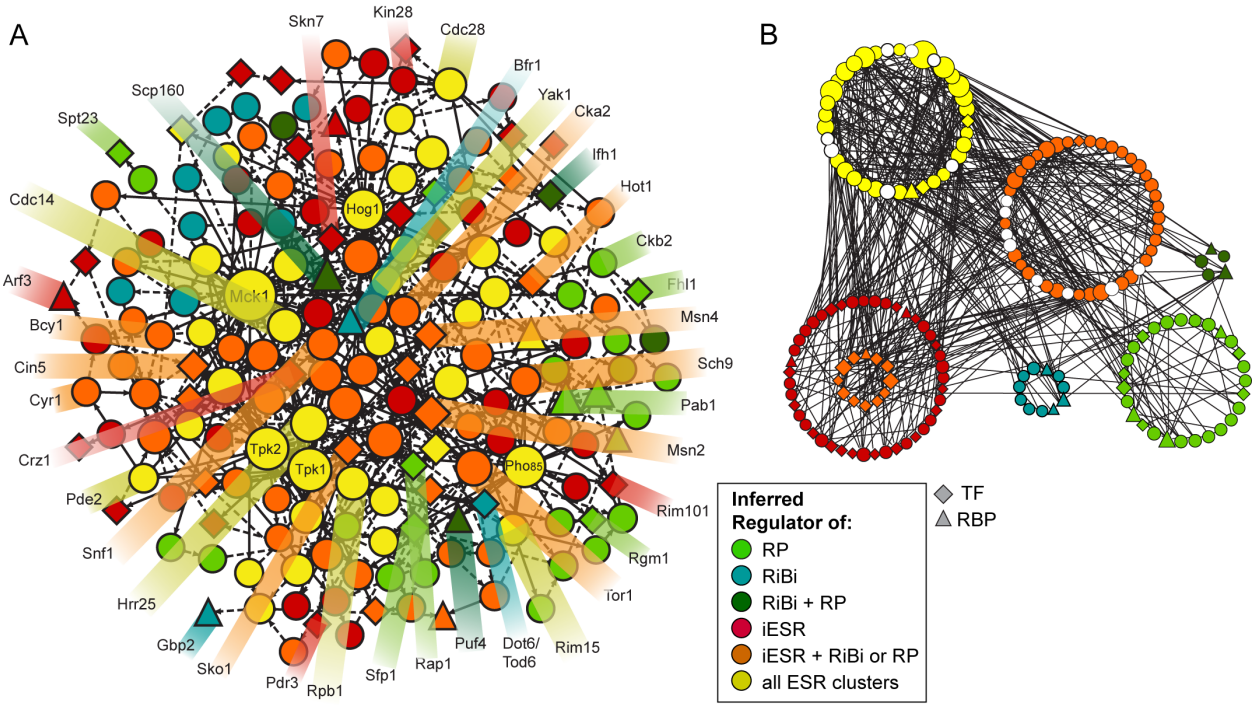



**Figure 5.6:** Inferred ESR regulatory subnetwork. Figure contributed by Audrey Gasch.
**A** Regulators predicted to lie upstream of iESR, RP, and/or RiBi ESR modules are color-coded according to the key and sized according to degree (number of connections). Diamonds and triangles represent TFs and RBPs, respectively.
**B** Same as **A** but organized by ESR regulatory potential. Top-ranked bifurcation nodes discussed in the text are colored white. Nested nodes represent iESR TFs that also fall upstream of RP paths.

**A score for ranking putative ESR bifurcation points**

To understand how cells coordinate repression of growth-related genes with induction of stress genes in the ESR, we present a score to rank candidate **bifurcation points**. These are nodes that a) are upstream of many genes from both modules (yellow and orange nodes in Figure 5.6) but b) have outgoing paths that relatively cleanly divide the iESR and rESR genes. We define a bifurcation score, $B(n)$, that is related to the concept of information gain ratio (Quinlan, 1986). $B(n)$ is calculated as follows, and an illustration of the process is provided in Figure 5.7.

First, we define the count $C(T)$, which counts the number of bits required to represent the cluster membership of all of the targets in a set of targets $T$. Considering the clusters $c \in$ {iESR, rESR}, let $T^c(n)$ be the set of targets downstream of $n$ that belong to the ESR cluster $c$, and let $T(n)$ be the set of all ESR targets downstream of $n$.

$$C(T(n)) = -\sum_{c \in \{\text{iESR, rESR}\}} |T^c(n)| \log_2 \frac{|T^c(n)|}{|T(n)|}$$

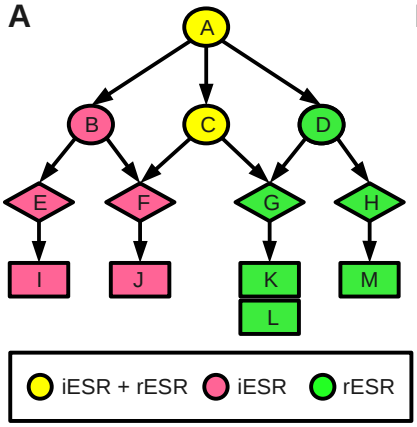
An ideal bifurcation point would have a high $C(T(n))$ compared to the paths that emanate from it. To perform this comparison, we next calculate $C(\cdot)$ for each of the paths downstream from $n$. If the subnetwork were a tree, $n$'s targets would simply be partitioned by $n$'s children. However, since the paths leading out from $n$'s children may converge on the same targets, we instead partition $T(n)$ into disjoint subsets of targets, each of which is reachable via a unique combination of $n$'s children. We refer to $n$'s outgoing partitions as $P_1(n)...P_m(n)$.

After having calculated $C(P_i(n))$ for each partition of $n$'s targets, we then calculate the information gain, $I(n)$, which measures the number of bits that are saved by partitioning the targets that are downstream of $n$:

$$I(n) = C(T(n)) - \sum_{i=1}^{m} C(P_i(n))$$

Finally, to calculate the bifurcation score $B(n)$, we normalize $I(n)$ by the *split information* $S(n)$, which measures the number of bits required to describe the partition assignment of one of n's targets. $I(n)$ is strongly biased toward nodes whose outgoing partitions split each target each into its own partition. The normalized score $B(n)$ prioritizes nodes that

A




B

| Target set | $C(T)$ |
|---|---|
| $T$(A) | 4.86 |
| $T$(C) | 2.76 |
| $T$(B only) | 0.0 |
| $T$(B and C) | 0.0 |
| $T$(C and D) | 0.0 |
| $T$(D only) | 0.0 |
| $T$(F) | 0.0 |
| $T$(G) | 0.0 |

C

| $n$ | $I(n)$ | $S(n)$ | $B(n)$ |
|---|---|---|---|
| A | 4.86 | 1.92 | 2.53 |
| C | 2.76 | 0.92 | 3.00 |

**Figure 5.7:** Example of ESR bifurcation score calculation.
**A** Example ESR network with nodes labeled according to which ESR targets are downstream. Nodes A and C are clearly candidate bifurcation points: they are upstream of both clusters, and we can see that the paths leading out from them divide the two clusters. However, choosing and ranking possible bifurcation points in the ESR consensus subnetwork is not easy to do by visual inspection.
**B** Calculation of $C(T)$, count, for each candidate bifurcation point (A and C) and their outgoing partitions. At a given node, the outgoing partitions are defined by combinations of the node's children. Node A's children define four outgoing partitions: ([B only], [B and C], [C and D], [D only]), and C's children define two outgoing partitions ([F], [G]). This table shows all of the values of $C(\cdot)$ that must be calculated for the example network: that of the two candidate bifurcation points, A and C, and each of their outgoing partitions.
**C** Final scores for the candidate bifurcation points. Note that while node A has superior information gain $I(\text{A})$ compared to C, its bifurcation score $B(\text{A})$ is penalized by its high split information. Node C is therefore the higher-ranked bifurcation point.

have a small number of (relatively) cleanly-split outgoing paths and many downstream targets.

$$S(n) = -\sum_{i=1}^{m} \frac{|P_i(n)|}{|T(n)|} \log_2 \frac{|P_i(n)|}{|T(n)|}$$

$$B(n) = \frac{I(n)}{S(n)}$$

The consensus ESR subnetwork allows for the ranking of 92 candidate bifurcation points (yellow and orange nodes in Figure 5.6). The full ranking is available in the original publication.

Remarkably, a third of the top fifteen ranked proteins associate with Pol II (including Rpb3, Spt5, Sub1, and Ask10). This result suggests that a key regulatory step is the reallocation of transcriptional capacity from highly transcribed rESR genes to stress-activated iESR genes. Of the remaining nodes, half are linked to cAMP signaling and include adenylate cyclase Cyr1, cAMP response regulator Bcy1, and phosphodiesterase Pde2. In validation of the subnetwork prediction, the Gasch lab found that cells lacking the cAMP-degrading PDE2 display aberrantly high abundance of rESR transcripts at the expense of iESR induction, paralleling the mutant's sensitivity to multiple stresses (Berry *et al.*, 2011). A recent publication proposes that cAMP levels play a key role in dictating whether translational capacity is directed toward growth versus other processes (such as stress defense) (You *et al.*, 2013). Taken together with the predicted bifurcation points, this suggests that reallocation of cellular resources is the driving force coordinating rESR repression with iESR induction.

### 5.3.6 The orthologous human network is enriched for disease related genes

Striking the correct balance between growth rate and stress defense is fundamental for proper cellular function, and improper balance is thought to be a critical driver in diseases such as cancer (Jones & Thompson, 2009). In this section, we discuss an analysis performed by our collaborators to investigate the potential of the inferred consensus subnetwork to provide insight into phenotypes in humans and mice. The following enrichment analyses consider both the input and predicted relevant yeast genes.

First, our collaborators identify the set of human genes that are orthologous to the yeast NaCl-responsive consensus node subnetwork. Using the RSD method of Wall *et al.* (2003), they identify 1,619 reviewed human genes. Comparing this set to the COSMIC database v67 (Forbes *et al.*, 2011) reveals an enrichment for genes linked to cancer, mostly through somatic mutation. Of the 35 human genes in the COSMIC dataset with yeast orthologs, eight were orthologous to nodes in the consensus node network, representing a 2.5-fold enrichment above chance ($p = 0.0068$).

They also identify orthologous proteins found in mice, using methods outlined by McGary *et al.* (2010). This approach looks across orthologous gene sets for their association with species-specific phenotypes, known as 'phenologs'. Phenologs emerge when a pathway is conserved across evolution but has evolved in its cellular function; thus, disruption of the pathway can produce different phenotypes in different species (Woods *et al.*, 2013; McGary *et al.*, 2010; Cha *et al.*, 2012). The phenolog database (Woods *et al.* (2013), `http://phenologs.org`) reveals strong enrichment among our inferred consensus

subnetwork for mouse orthologs required for pre/perinatal viability, normal growth rate and body size, and male and female fertility (FDR < 5%). These phenotypes indicate that, like the yeast subnetwork, the orthologous network in mammals is also linked to growth decisions.

### 5.3.7 Stability analysis to compare different versions of the IP

One measurement we use to assess the results of changing different aspects of the IP or input data is the stability of the predictions made by the inferred ensemble. We compare a pair of inferred ensembles based on the Jaccard similarity between their predictions. We calculate the similarity between two ensembles $E$ and $E'$ as follows. Using the variable $y_n$ (node relevance) as an example, $p^E(y_n = 1)$ is $E'$s confidence that node $n$ is relevant. $\mathcal{N}$ refers to the set of all nodes (proteins).

$$similarity(E, E') = \frac{\sum_{n \in \mathcal{N}} |p^E(y_n = 1) - p^{E'}(y_n = 1)|}{\sum_{n \in \mathcal{N}} p^E(y_n = 1) + p^{E'}(y_n = 1) - |p^E(y_n = 1) - p^{E'}(y_n = 1)|}$$

In the following experiments, we compare the ensembles based on four variable types: node relevance, edge relevance, edge direction, and path relevance. Sources, targets, and upstream receptors are omitted from the node stability calculations because their relevance variables are fixed. When calculating the similarity of edge directions, only edges that are undirected in the background network are counted.

### 5.3.8 Testing the effect of variations on candidate path length

In this section, we discuss experiments that measure the effect of increasing and decreasing the maximum length of the candidate source-target and hit-source paths by one interaction. For each variation, we infer an ensemble of 1,000 subnetworks, and compare the results to the original consensus ensemble on the basis of precision-recall and stability analyses (Figure 5.8), as well as enrichment analysis on the nodes with $\geq$ 75% confidence (Table 5.9). Because the candidate path sets are different between the original, complete IP and the new variations, we do not measure the stability of paths for these comparisons, and only compare node and edge relevance and edge direction.

**Source-target paths (Figure 5.8A-B)**. In the original IP, we enumerate paths of up to five interactions, stopping search early at the depth at which at least 50% of candidate TFs/RBPs

for a given source are reached. We test the effect of this early-stopping option by inferring subnetworks using candidate paths enumerated at three different lengths: three, four, or five interactions for all sources. At a path length of five, we are unable to complete the IP solution portion of the method due to the large number of paths generated by two sources, Hog1 and Mck1. As a compromise, we stop the search for those two sources at four, but search for paths of length five for all other sources.

In general, the precision-recall curves and most enrichment scores do not appear to be very sensitive to the path length in the range tested; however, we see some patterns. The ensemble inferred from candidate paths of length four appears to be nearly identical to the original IP under all measures, which is unsurprising, as search terminates at that length for most sources in the original setting. Stopping all search at length three results in a slight increase in precision, a slight decrease in recall, and a significant increase in genetic interactions. Stopping all search at length five results in a slight decrease in precision and a significant decrease in genetic interactions. As shown by the relative lengths of the ends of the bars in the stability analysis figure (Figure 5.8B), increasing candidate path length also results in increased size of inferred subnetworks. The node content of the inferred ensembles is fairly similar, with more variation shown in edge relevance and direction.

**Hit-source and source-source paths (Figure 5.8C-D)**. For the original IP, we enumerate candidate paths consisting of up to two edges between the fitness-contribution hits and the tested sources. To assess the sensitivity of the method to this path length, we test two alternatives: allowing only direct interactions (no intermediate nodes), and allowing longer paths of up to three edges (two intermediate nodes).

The shorter path length confers increased precision in the precision-recall curve; however, the 75%-confidence nodes from this subnetwork do not have statistically significantly different proportions of the gene sets tested for enrichment. There is a small but statistically significant increase in genetic interactions in the ensemble with shorter paths. This subnetwork includes 49 fitness-contribution hits, in contrast to the 106 included in the original consensus subnetwork, and is somewhat smaller.

The longer path length results in a large decrease in precision, weakly significant decreases in enrichment of true positives and genetic interactions, and a corresponding weak-ly significant increase in essential gene enrichment and likely negatives. This subnetwork includes 151 fitness-contribution hits and is much larger overall.

Considering these results, it appears that limiting the length of candidate hit-source and source-source paths is beneficial: longer paths result in larger subnetworks with decreased precision and decreased enrichment with other relevant gene and interaction sets. The preferred stringency of this limit may depend on how the inferred subnetwork will be used

to guide further experiments. The shortest length path provides an increase in accuracy at the expense of not being able to provide connections to as many fitness-contribution hits.

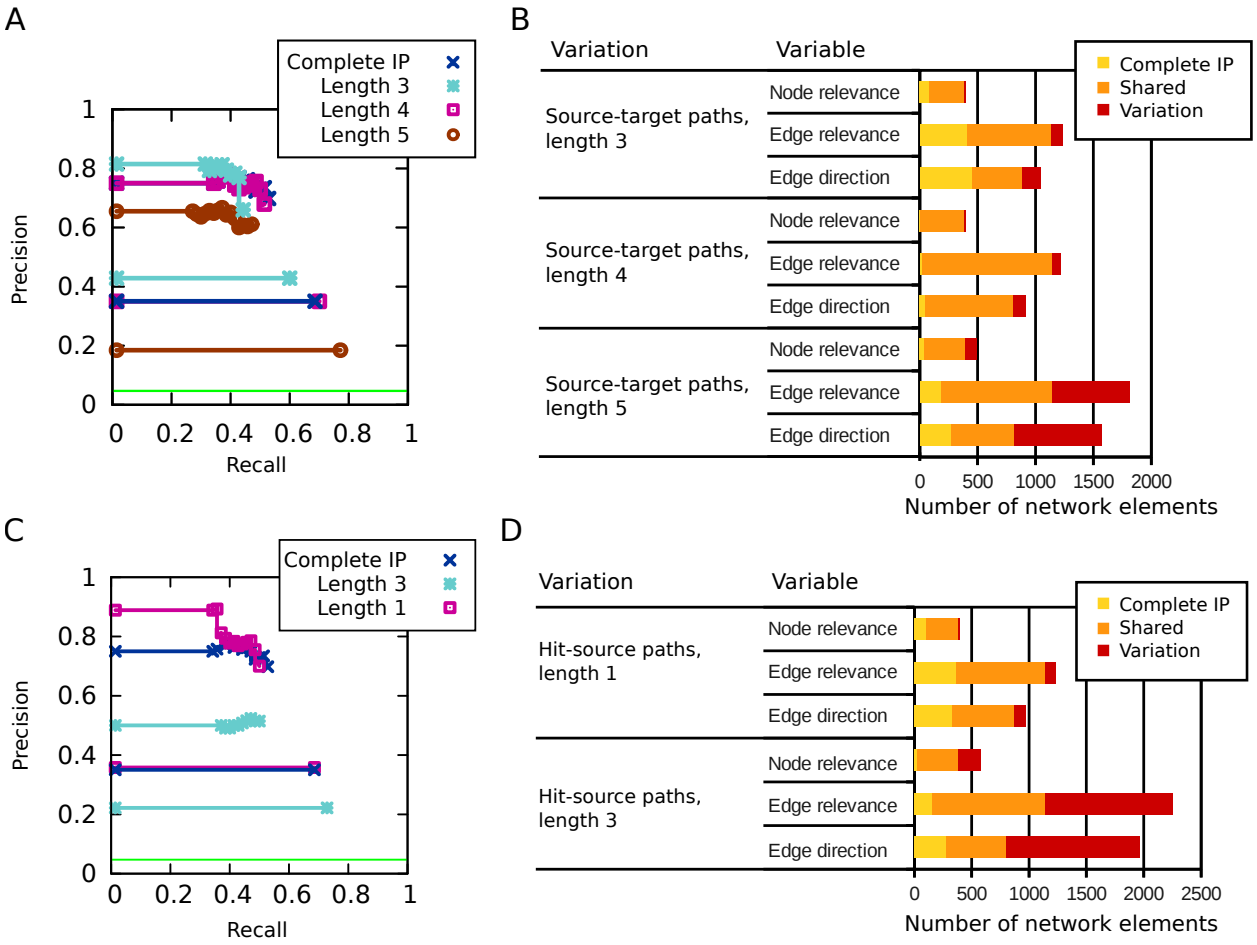



**Figure 5.8:** Testing variations on the length of candidate paths. Precision-recall curves and stability results derived from changing the length of candidate paths. For each variation, we show two PR curves in the same color. The top curve represents the inferred ensemble, while the bottom curve represents the performance of candidate paths provided as input to the inference method. The horizontal green line in PR plot indicates the precision achieved by predicting that all test cases are positive examples. In the stability analysis panel, each bar represents the total number of elements in the union of the complete ensemble with an ensemble with a different path length. The central portion of the bar, “Shared", represents the number of elements that are in the intersection of both ensembles. The end caps, labeled “Complete IP" and “Variation", represent the number of elements unique to each ensemble. The longer the center is relative to the ends, the more similar are the two ensembles in their predictions. In contrast to previous stability analyses, we do not compare the ensembles based on path relevance, as the input sets of paths are different.

**A** Precision-recall analysis for varying source-target path length
**B** Stability analysis for varying source-target path length
**C** Precision-recall analysis for varying hit-source path length
**D** Stability analysis for varying hit-source path length

**Table 5.9:** Enrichment analysis of subnetworks inferred from varied candidate path lengths. 'Prop' columns show the proportion of each consensus subnetwork that is composed of the relevant gene set being tested. $P$-values are calculated by a two-tailed $z$-test of proportions comparing the complete consensus subnetwork (derived from the 1,000 solution ensemble) to the consensus subnetwork inferred using a different path length (also derived from 1,000-solution ensembles). Asterisks (*) indicate significance at $p < 0.05$. For significant results, bolded proportions and $p$-values indicate comparisons in which the original, complete IP has a higher proportion of relevant genes/interactions (or lower proportion of likely negatives). Italicized proportions and p-values indicate the opposite.

| | Relevant gene sets | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | True positives | | Likely negatives | | Kinases and phosphatases | | General stress proteins | | Essential genes | | Genetic interactions | |
| IP Version | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value |
| Complete IP (1k sols.) | 0.168 | | 0.054 | | 0.197 | | 0.257 | | 0.293 | | 0.052 | |
| Source-target paths, length 3 | 0.176 | 0.857 | 0.046 | 0.751 | 0.221 | 0.616 | 0.260 | 0.968 | 0.328 | 0.518 | *0.057* | *≈ 0 ** |
| Source-target paths, length 4 | 0.168 | 1.000 | 0.048 | 0.803 | 0.204 | 0.891 | 0.258 | 1.000 | 0.287 | 0.904 | 0.052 | 0.504 |
| Source-target paths, length 5 | 0.124 | 0.260 | 0.071 | 0.517 | 0.195 | 0.960 | 0.225 | 0.484 | 0.343 | 0.327 | **0.044** | **≈ 0 *** |
| Source-hit paths, length 1 | 0.191 | 0.604 | 0.038 | 0.524 | 0.214 | 0.731 | 0.260 | 0.970 | 0.282 | 0.836 | *0.061* | *≈ 0 ** |
| Source-hit paths, length 3 | **0.100** | **0.039 *** | 0.107 | 0.055 | 0.184 | 0.724 | 0.207 | 0.223 | *0.387* | *0.048 ** | **0.041** | **≈ 0 *** |

### 5.3.9 Testing the effect of ensemble size

As it is practically infeasible to identify all optimal solutions to the IP, we specify the number of subnetworks in the ensemble as an input to our method. The question arises of how the ensemble's predictions change as more solutions are identified. In this analysis, we compare the predictions made by the ensemble of 10,000 solutions to two smaller ensembles of 100 and 1,000 solutions each. As shown in Figure 5.9A, the precision-recall curves for each ensemble are approximately the same, although the recall of low-confidence predictions increases slightly as more solutions are gathered. The representation of general stress proteins, kinases and phosphatases, and genetic interactions are also approximately the same across the different ensemble sizes.

Additionally, we compare the different sizes of ensembles based on the Jaccard similarity between their predictions, as described in Section 5.3.7. We compare the original ensemble of 10,000 solutions to the smaller ensembles of 100 and 1,000 solutions based on their predictions of node relevance, edge relevance, edge direction, and path relevance. For all comparisons, the similarity between the 10,000-solution ensemble and the smaller ensemble is very high ($\geq$ 90%). In Figure 5.9B, we provide a visualization of the results of each comparison, showing the number of elements that are shared or unique to each ensemble.

All together, our stability analysis results demonstrate that the distribution of solutions captured by CPLEX is not changed significantly by the collection of more solutions within the range tested.

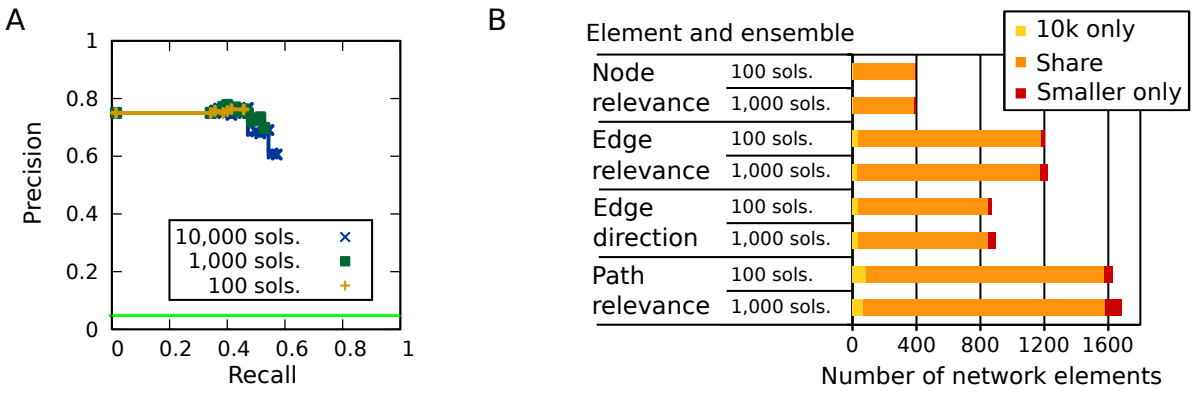



**Figure 5.9:** Stability analysis results.
**A** Precision-recall curves for inferred ensembles of three sizes. The horizontal green line in each panel indicates the precision achieved by predicting that all test cases are positive examples.
**B** Similarity of confidence values for different types of network elements (nodes, edges, paths) between different sizes of ensembles. Each bar represents the total number of elements in the union of the complete, 10,000-solution ensemble with a smaller ensemble. The central portion of the bar, 'Shared', represents the number of elements that are in the intersection of both ensembles. The end caps, labeled '10k only' and 'Smaller only', represent the number of elements unique to each ensemble. The longer the center is relative to the ends, the more similar are the two ensembles in their predictions.

### 5.3.10 Assessing the contributions of IP components through lesion testing

Our approach combines multiple sources of experimental data using a multi-step optimization procedure. It is useful to investigate the contribution of each ingredient to the predictions made by the inferred subnetwork ensemble. To do so, we run lesioned versions of the IP, in each of which one component of the optimization procedure or one experimental data source is held aside. For each lesioned IP, we infer an ensemble of solutions, which we refer to as a *lesioned ensemble*. We evaluate the lesioned ensembles using the same computational evaluation metrics that we apply to the ensemble inferred by the complete IP (the *complete ensemble*): namely, accuracy in predicting held-aside test cases, and enrichment of relevant, externally-derived gene and interaction sets. Experimental hits are omitted from the test cases and the relevant, external gene sets. As an additional evaluation, we compare the lesioned ensembles to the complete ensemble based on the similarity of their predictions, as in the stability analysis presented in Section 5.3.9.

Having observed that predictions are highly stable regardless of the size of the ensemble (Section 5.3.9), we perform the lesion experiments using an ensemble size of 1,000. Correspondingly, we compare the lesioned ensembles to the complete ensemble with 1,000 solutions.

Figure 5.10A-F show comparative precision-recall curves for each lesioned ensemble. In Table 5.10, we show the results of the enrichment analyses, including the proportion of each lesioned consensus subnetwork that is contained in each external gene and interaction set. For each relevant set, we compare its representation in the lesioned consensus to the complete consensus using a two-tailed $z$-test of proportions.

We discuss the contribution of each lesioned component:

**Maximizing the number of connections between hits and sources (Figure 5.10A).** Removing this step alone slightly increases the predictive accuracy of the inferred ensemble, but does not significantly change the proportions of the consensus nodes that are represented by any relevant gene sets. We suggest three possible explanations for the change in precision: a) the experimental hit set is noisy or includes proteins that are required for stress survival but not for signaling *per se*, b) the set of test cases is relatively small and based on focused laboratory experimentation, and is thus likely to be incomplete, and c) removing this objective function component leaves the resulting IP less constrained, allowing greater variation between the different solutions in the ensemble and more stratification between high-confidence predictions. There is also a small but statistically significant

change in the proportion of genetic interactions. Despite the small change in precision that it confers, including this step assists in the interpretation of the experimental data, as it reveals proximal connections between fitness-contribution hits and sources that are not captured by source-target paths.

**Maximizing the number of experimental hits (Figure 5.10B).** Even though the hit nodes themselves are not considered in the accuracy analysis, maximizing their inclusion results in more accurate choices among the other nodes: the lesioned IP has a slightly lower PR curve compared to the complete IP. The representation of relevant gene sets does not change significantly; however, the lesioned consensus has a higher proportion of genetic interactions than the complete consensus node set.

**The combined contribution of maximizing connections and hits (Figure 5.10C).** These two objective components are partially redundant, as maximizing the number of connections indirectly also maximizes the inclusion of hits. Therefore, we assess the effect of removing both components. This results in a slight increase in precision. Additionally, while the lesioned consensus node set does not differ in the representation of relevant gene sets, it does contain a significantly higher proportion of genetically interacting pairs of nodes. The next lesion we report attempts to tease apart the contributions of the two experimental hit sets.

**Maximizing the inclusion of either of the two hit sets (Figure 5.10D).** To assess the hit sets separately, we perform two lesions: separately holding aside the fitness-contribution hits and the phospho-proteomic hits. The inclusion of phospho-proteomic hits in the IP appears to be useful for identifying higher precision subnetworks, demonstrating that some of the information contained in this hit set is congruent with current knowledge about the salt stress response. Conversely, holding aside the entire set of fitness-contribution hits (modulo the sources that are fitness-contribution hits) results in a moderate increase in precision, suggesting that this set of hits is noisy or partially disjoint to existing knowledge. Therefore the lack of fitness-contribution hits may be primarily responsible for the increased precision observed when both types of hits are held aside. Neither lesioned consensus node set shows a significant change in the representation of relevant node sets, though the lesioning of phospho-proteomic hits results in a statistically significant increase in genetic interactions. It should be noted that most genetic interactions have been measured under standard growth conditions; it is likely that NaCl-specific genetic interactions are not represented in the test set.

Even though the predictive accuracy drops slightly when fitness-contribution hits are maximized, our results may actually represent novel discoveries, as the set of true positives

is heavily biased toward HOG pathways proteins. Regardless, we are willing to tolerate a slight drop in accuracy in exchange for increased interpretability. We believe that including the fitness-contribution hits in the IP is still useful; after all, part of the purpose of this method is to aid in the interpretation of the experimental hits. The resulting subnetwork reveals connections between the fitness-contributions and other components of the salt stress response, which may inspire further experimentation.

**Minimizing the total number of nodes (Figure 5.10E).** Removing this step results in decreased precision and enrichment across the board, demonstrating that minimizing nodes is useful for reducing false positives.

**Connecting receptor-source pairs (Figure 5.10F).** Including the receptor-source pairs appears to significantly increase recall, but does not appear to have a great effect on precision or enrichment with relevant gene sets or genetic interactions. We believe that including these pairs makes the inferred subnetwork more interpretable by showing connections to well-understood interactions in the stress response.

Similar to our evaluation of the stability of the complete IP across different ensemble sizes, we also report the similarity between the complete and lesioned ensembles. A visualization of the differences and intersections between the ensembles' predictions is shown in Figure 5.11. Except when nodes are not minimized, the lesioned ensembles contain fewer nodes, edges, and paths, and are mostly subsets of the complete ensemble. (In the figure, we observe that the size of the left-hand portion of each bar generally dominates the right-hand portion.) When nodes are not minimized, the reverse is observed. Path relevance and edge direction show the most variablility, demonstrated by the longer right-hand ends to those bars.

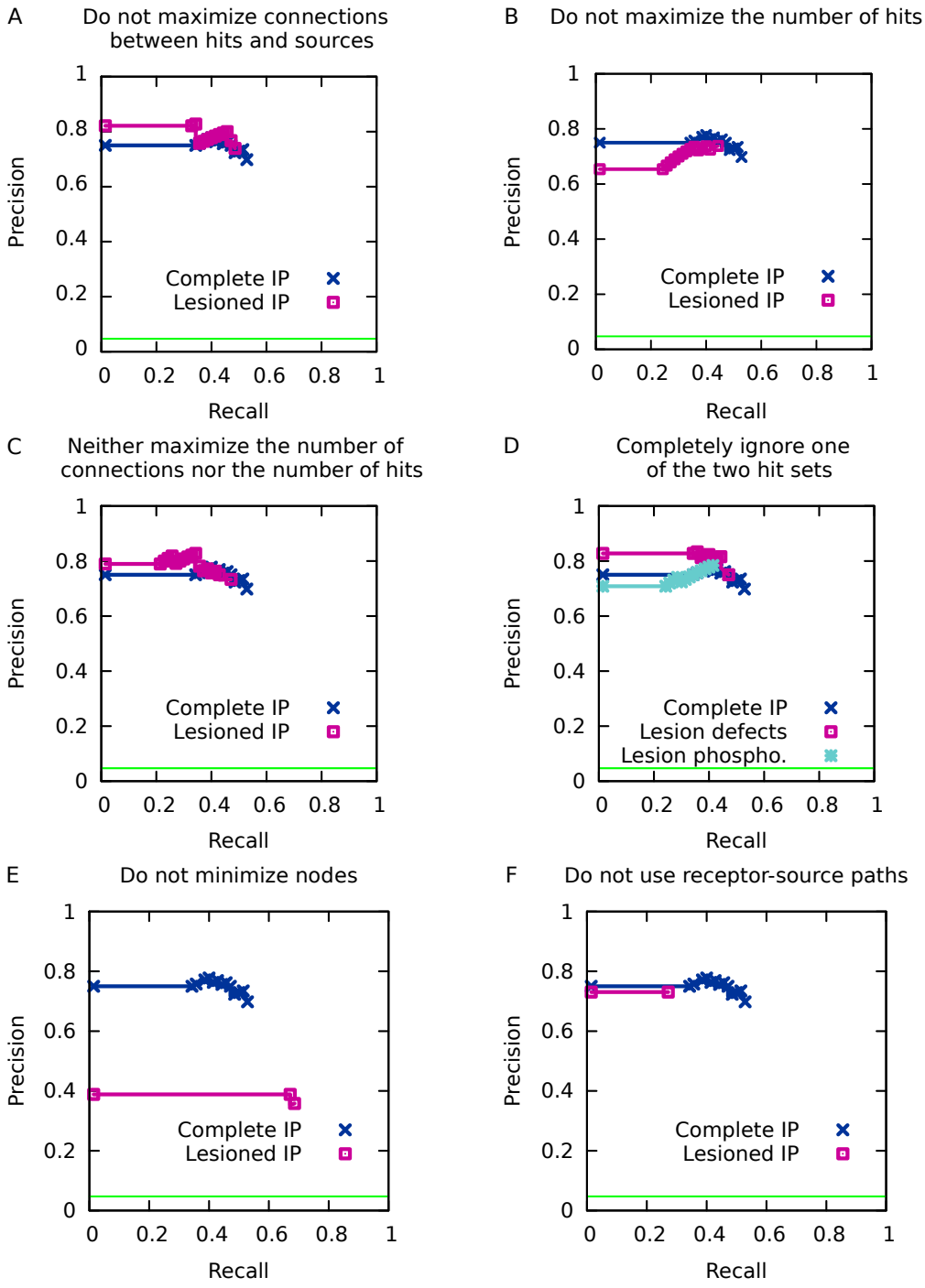



**Figure 5.10:** Precision-recall curves for lesioned IPs. The horizontal green line in each panel indicates the precision achieved by predicting that all test cases are positive examples. The lesioned components are:
**A** maximization of connections,
**B** maximization of hits,
**C** maximization of both connections and hits,
**D** an entire set of experimental node labels (fitness-contributions or phospho-proteins),
**E** minimization of nodes,
**F** inclusion of receptor-source pairs curated from literature.

**Table 5.10:** Enrichment analysis of lesioned IPs. 'Prop' columns show the proportion of each lesioned or complete consensus subnetwork that is composed of the relevant gene set being tested. $P$-values are calculated by a two-tailed $z$-test of proportions comparing the complete consensus subnetwork (derived from the 1,000 solution ensemble) to the consensus subnetwork inferred using a different path length (also derived from 1,000-solution ensembles). Asterisks (*) indicate significance at $p < 0.05$. For significant results, bolded proportions and $p$-values indicate comparisons in which the original, complete IP has a higher proportion of relevant genes/interactions (or lower proportion of likely negatives). Italicized proportions and $p$-values indicate the opposite.

| | Relevant gene sets | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | True positives | | Likely negatives | | Kinases and phosphatases | | General stress proteins | | Essential genes | | Genetic interactions | |
| IP Version | Prop. | $p$-val. | Prop. | $p$-val. | Prop. | $p$-val. | Prop. | $p$-val. | Prop. | $p$-val. | Prop. | $p$-val. |
| Complete IP (1k sols.) | 0.168 | | 0.054 | | 0.197 | | 0.257 | | 0.293 | | 0.052 | |
| Do not maximize connections | 0.179 | 0.801 | 0.057 | 0.901 | 0.200 | 0.958 | 0.250 | 0.881 | 0.279 | 0.775 | **0.047** | **≈0 *** |
| Do not maximize hits | 0.131 | 0.380 | 0.066 | 0.664 | 0.182 | 0.738 | 0.255 | 0.968 | 0.263 | 0.554 | *0.061* | *≈0 ** |
| Neither maximize connections nor hits | 0.183 | 0.757 | 0.043 | 0.700 | 0.172 | 0.613 | 0.237 | 0.709 | 0.247 | 0.426 | *0.071* | *≈0 ** |
| Do not use fitness-defects | 0.202 | 0.454 | 0.047 | 0.774 | 0.202 | 0.933 | 0.256 | 0.974 | 0.310 | 0.757 | 0.050 | 0.242 |
| Do not use phospho-hits | 0.133 | 0.409 | 0.052 | 0.937 | 0.193 | 0.913 | 0.252 | 0.911 | 0.259 | 0.510 | *0.060* | *≈0 ** |
| Do not minimize nodes | **0.067** | **0.000 *** | **0.113** | **0.024 *** | **0.121** | **0.009 *** | **0.147** | **0.001 *** | 0.315 | 0.591 | **0.033** | **≈0 *** |
| Do not use receptor-source paths | 0.133 | 0.394 | 0.049 | 0.845 | 0.182 | 0.724 | 0.273 | 0.762 | 0.273 | 0.687 | 0.052 | 0.553 |

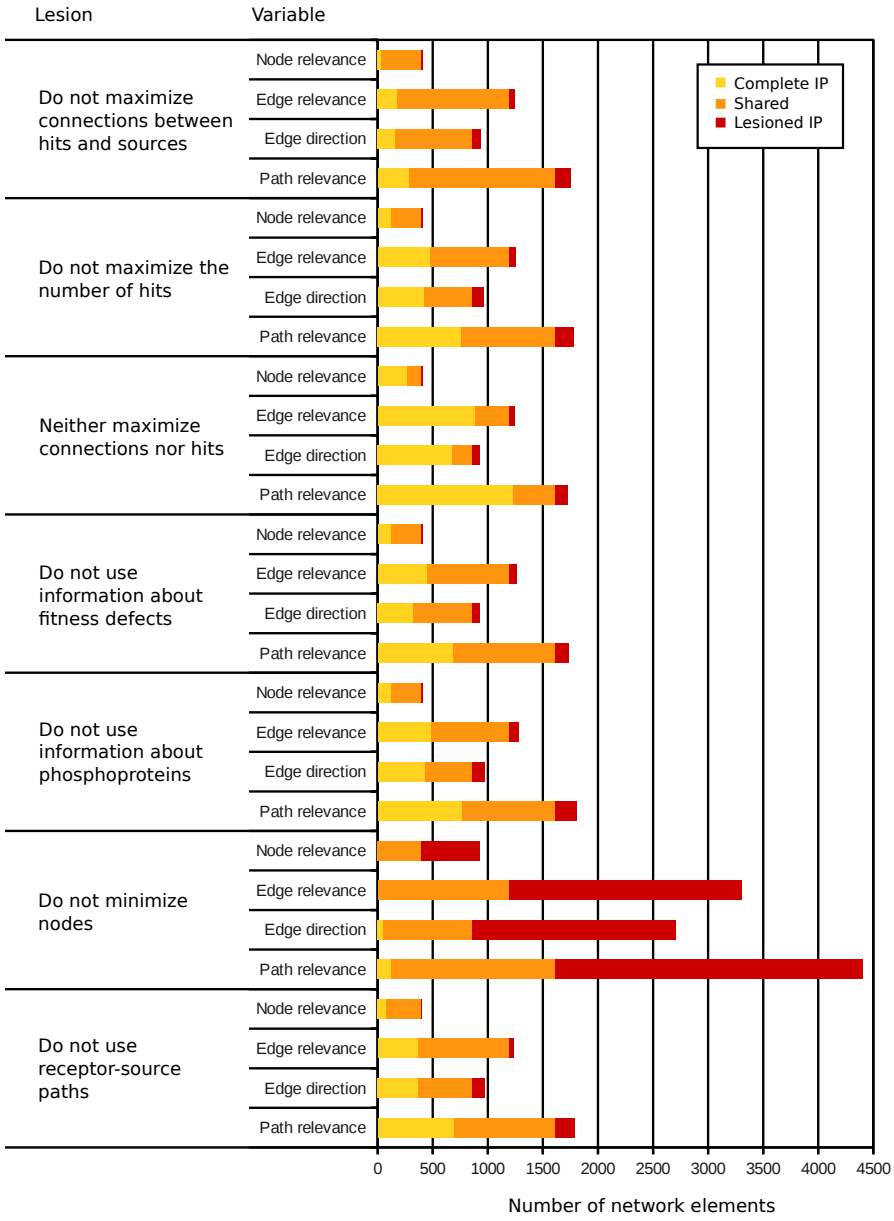



**Figure 5.11:** Similarity of ensembles inferred by the lesioned IPs. Comparison of the complete ensemble to the lesioned ensembles on the basis of the confidence values given to each type of network element (nodes, edges, paths). Each bar represents the total number of elements in the union of the complete ensemble with a lesioned ensemble. The central portion of the bar, “Shared", represents the number of elements that are in the intersection of both ensembles. The end caps, labeled “Complete IP" and “Lesioned IP", represent the number of elements unique to each ensemble. The longer the center is relative to the ends, the more similar are the two ensembles in their predictions.

### 5.3.11 Assessing our choice of objective function procedure order

Our four objective functions can be summarized as representing two concepts: explain a *maximal* amount of the experimental data using a *minimal* subnetwork. To assess the sensitivity of the subnetwork inference method to the ordering of these concepts, we perform an experiment in which we reverse the order. The altered sequence of objective functions proceeds as follows, and can be compared to the sequence shown in Figure 5.3:

1. Minimize nodes (one solution)
2. Maximize connections (one solution)
3. Maximize fitness-contribution and phospho-proteomic hits (one thousand solutions)
4. For each previous solution, maximize paths (one solution each)

Using this ordering of the objective functions, we infer an ensemble of 1,000 subnetworks, and compare the results to the original consensus ensemble using precision-recall and stability analysis (Figure 5.12), and enrichment analysis (Table 5.11). Compared to the original IP, the IP with reversed objective function components (which we name “min-then-max") achieves lower recall of the known regulators, no significantly different proportions of relevant gene sets, and a significant increase in genetic interaction enrichment. The resulting subnetworks are also overall smaller, as shown by the relative lengths of the colored portions of the bars in the stability analysis chart in Figure 5.12B.

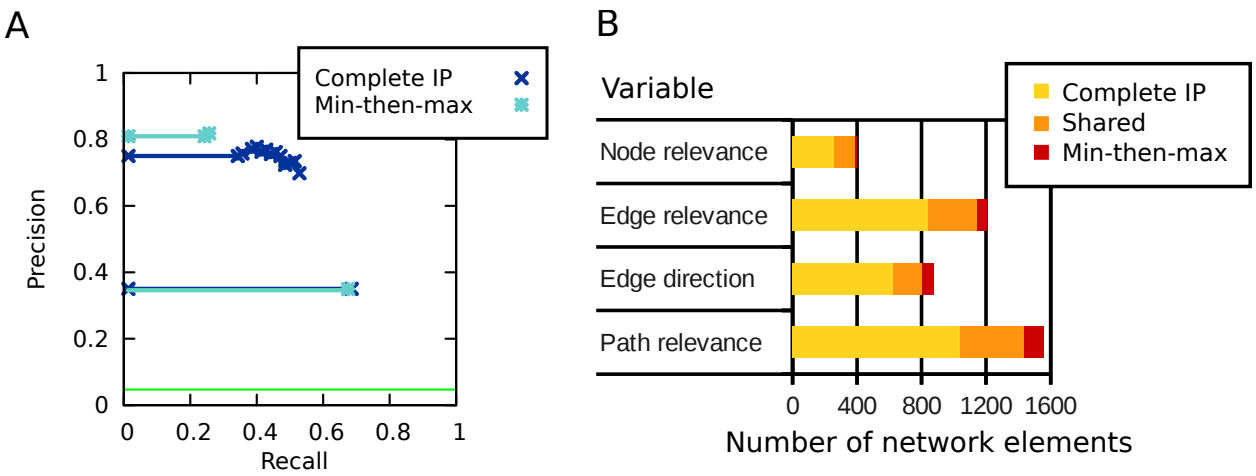



**Figure 5.12:** Evaluating a reordering of objective function components.Precision-recall curves and stability results derived from reversing the order in which the objective function components are solved.
**A** Precision-recall analysis. For each ensemble, two PR curves are shown, one for each variation on the objective function procedure. The top curve represents the performance of the inferred subnetwork, while the bottom curve represents the performance of input candidate paths. The curves for candidate paths are identical, as the input was the same for both versions. The horizontal green line in each panel indicates the precision achieved by predicting that all test cases are positive examples.
**B** Stability analysis. Each bar represents the total number of elements in the union of the complete ensemble with a lesioned ensemble. The central portion of the bar, “Shared", represents the number of elements that are in the intersection of both ensembles. The end caps, labeled “Complete IP" and “Min-then-max", represent the number of elements unique to each ensemble. The longer the center is relative to the ends, the more similar are the two ensembles in their predictions.

**Table 5.11:** Enrichment analysis of reordered IPs. 'Prop' columns show the proportion of each consensus subnetwork that is composed of the relevant gene set being tested. $P$-values are calculated from a two-tailed $z$-test of proportions comparing the complete consensus subnetwork (derived from 1,000 solutions) to the consensus subnetwork inferred using a different ordering of the objective function procedure (also derived from 1,000 solutions). Asterisks (*) indicate significance at $p < 0.05$. For the significant result, the italicized proportion (and $p$-value) indicates a comparison in which the original, complete IP has a lower proportion of relevant genetic interactions.

| | Relevant gene sets | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | True positives | | Likely negatives | | Kinases and phosphatases | | General stress proteins | | Essential genes | | Genetic interactions | |
| IP Version | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value | Prop. | $p$-value |
| Complete IP (1k sols.) | 0.168 | | 0.054 | | 0.197 | | 0.257 | | 0.293 | | 0.052 | |
| Min-then-max | 0.181 | 0.787 | 0.043 | 0.686 | 0.170 | 0.586 | 0.255 | 0.969 | 0.245 | 0.398 | *0.076* | *$\approx 0$ ** |

## 5.4 Discussion

In this chapter, we presented an application of our subnetwork inference method (Thesis Contribution 1) to assist in a highly collaborative study of the yeast salt stress response (Thesis Contribution 2). Our method performs well under our multifaceted computational evaluation, and we demonstrate that it can be used to generate important insights into biology that stand up to experimental validation.

Several aspects of our work support the goal of improving interpretability of subnetwork inference methods (Thesis Contribution 3).

- We offer methods for making predictions not only about the relevance of gene products, but also their role in the stress response. The representation of the inferred subnetworks, as well as methods we offer for querying them, can assist a biologist in translating the subnetworks' predictions into testable hypotheses. The path-based structure of the inferred subnetworks allows for the prediction of which gene targets will be affected by an additional gene suppression experiment. The results of our the Gasch lab's validation experiments strongly support the subnetworks' accuracy.

- The inferred consensus subnetwork identifies new regulators in the NaCl response and also provides a glimpse into how regulators are connected in a single cellular signaling system. The extensive physical connectivity between what are traditionally considered distinct pathways suggests much greater signaling integration than previously realized. The inferred subnetwork has prompted our collaborators to perform further experimental inquiry into some of these connections.

- Our novel bifurcation scoring method and inferred consensus ESR subnetwork can be used to predict in which parts of the stress response a given node is involved. Specifically, we used the scoring method to predict which nodes are most important for coordinating the trade-off between cell growth and stress defense.

- We incorporate input data from multiple biological experiments or domain knowledge that consider different aspects of the stress response. To do so, we search for multiple kinds of candidate paths, and employ a multi-part objective function to selectively incorporate each path and data type. The results of our lesion experiments (Section 5.3.10) demonstrate that our selective optimization procedure does not degrade the accuracy of the inferred subnetworks compared to a simpler version of the IP. We propose that our representation of mutiple path types improves the interpretability of the subnetworks.

- The background network represents many types of biological interactions. To account for the observed expression changes, we employ both protein-DNA-binding interactions, which have typically been used in subnetwork inference methods, as well as protein-RNA-binding interactions, which have not. The use of both types allows the inferred subnetworks to predict whether the observed expression changes are controlled at the point of transcription or through altered mRNA stability.

# 6 Dénouement

This thesis presented a method for inferring relevant subnetworks from a large background network, given an incomplete set of relevant nodes and ordered relationships between relevant nodes. Although we were motivated by specific biological problems, our method could be applied to finding relevant subnetworks within other types of relational or graph data, such as supply chains and social networks. We describe the problem abstractly here.

**Given:**

- a large, partially-directed network of nodes and edges, $G = (\mathcal{N}, \mathcal{E})$, in which some nodes represent higher-order relationships between nodes (*and* and *or* relations), some edges have binary labels (signs), and many nodes and edges are assumed to be irrelevant
- a set of known relevant nodes, or hits, $\mathcal{N}^H \subseteq \mathcal{N}$, assumed to be incomplete
- a set of ordered pairs of relevant nodes, $(n_i, n_j) \in \mathcal{N}^H \times \mathcal{N}^H$, assumed to be incomplete
- additional binary labels (signs) on a subset of the relevant nodes, assumed to be incomplete

**Do:** Identify the distribution of inferred relevant subnetworks of $G$ that each:

- includes all (or nearly all) relevant nodes $n \in \mathcal{N}^H$, and predicts the inclusion of missed relevant nodes
- allows all (or nearly all) of the ordered pairs to be connected in their given order by directed paths through the relevant subnetwork
- generously includes edges to connect relevant nodes
- posits a single direction for each relevant edge
- infers the values of missing sign labels on relevant nodes and edges
- is subject to additional constraints on:
  - how the node and edge sign labels may be combined by relevant nodes and edges
  - how nodes that represent higher-order relationships may be inferred to be relevant

  - the inclusion of specific subsets of nodes or edges; for example, limits on the number of edges in a particular subset that can be predicted to be relevant

In each of Chapters 3 to 5, we considered a variation on this general problem. Here we summarize the main computational requirements of the general task, and how we approached each one.

- **A mechanism for requiring the relevant subnetwork to provide paths that connect input ordered pairs.** To do so, we enumerated candidate paths through the background network. Our inference method chooses a subset of these paths.
- **A method for constraint solving.** We chose to represent the problem as solving an integer linear program. The relevance and binary labels on nodes and edges, the directions of edges, and the relevance of candidate paths, are represented with binary variables. The program specifies linear constraints on the variables. We used objective functions and global constraints to impose heuristic preferences for how many additional nodes and edges are predicted to be relevant.
- **A method for estimating the distribution of relevant subnetworks.** We explored two methods for estimating this distribution by gathering an ensemble of subnetworks. One was to find many solutions to the IP using a branch-and-cut algorithm. The other was to assemble an ensemble of solutions by varying the subset of the input data that was used to infer each subnetwork.

## 6.1 Summary of contributions

We demonstrated the utility of our subnetwork inference method through three successful applications to different biological problems. Our work improves upon the state of the art of biological subnetwork inference methods by improving the interpretability of the inferred subnetworks, in particular. In this summary, we sort our contributions into three general categories, based on the stage of the inference process in which they are introduced: at the level of *input data scope and representation*, in the *design of the inference method*, and in the *generation and evaluation of predictions from the inferred subnetworks*.

### 6.1.1 Scope and representation of input data

Our method is useful for integrating and explaining heterogeneous experimental data types that are of great interest to biological researchers.

- In Chapters 3 and 4, we presented a method for inferring subnetworks based on viral phenotype labels from yeast deletion, yeast knockdown, and mammalian RNAi experiments. We believe that Chapter 3 represents the first attempt to infer host-virus interaction subnetworks, including host-virus interfaces, from these data (and a background network) alone.

- In Chapter 5, we applied our method to three experimental data sets that each provide a different perspective on the cell's stress response: mutant transcription data in the form of directed source-target pairs, and phosphoproteomic and mutant fitness data in the form of additional node labels.

- The basic graph representation of the background network can be used for many kinds of binary interactions between genes or proteins, with or without signs or directions.

- We believe that Chapter 5 contains the first use of RNA-binding data in a source-target-pair-based subnetwork inference method. By incorporating information about both transcription factors and RNA-binding proteins, our method is able to make predictions about whether the effect of a source on its targets is controlled at the point of transcription, versus mRNA stability.

- We also offered more detailed representations for special types of data, including protein complexes and metabolic reactions. Incorporating these types of relationships into the inferred subnetworks improves their interpretability: the behavior of protein complexes in biological pathways is generally more relevant than the behavior of each individual protein. In Chapter 3, we showed that our method is able to predict the relevance of protein complexes, and our analysis shows that this putative relevance is nor merely an artifact of the background network topology. We believe that our work represents the first attempt to incorporate non-binary protein complexes directly into a path-based subnetwork inference method.

- Like those inferred by related methods, our subnetworks are based on candidate paths. This representation is versatile, and allows for the prediction of mechanistic paths to explain, or predict, multiple kinds of indirect relationships between gene products. Our work expands the types of paths that have been considered by related methods. In Chapter 3, we used them to predict host-virus interfaces, and in Chapter 4, to connect the RNAi hits to known host-virus interfaces. In Chapter 5, we used candidate paths not only to connect source-target pairs, but also to identify connections between

fitness-contribution hits, as well as between surface receptors and key signaling regulators.

### 6.1.2 Design of the inference method

Our inference method was carefully designed to infer plausible, mechanistic subnetworks. Several aspects of our method improve the interpretability of the inferred subnetworks compared to related approaches.

- In Chapters 3 and 4, we combined a gene prioritization method (a diffusion kernel) with a subnetwork inference method. This combination leverages the advantages of both approaches. The inferred subnetworks accurately predict relevant genes, and organize them into interpretable, mechanistic paths.
- We developed biologically-motivated IP constraints to improve the interpretability of the inferred subnetworks. Similar to previous subnetwork inference methods, we require subnetworks to be directed (Chapters 3 and 5) or signed (Chapter 3). We also proposed new constraints that, to our knowledge, are novel.
  - In Chapter 3, we provided constraints on the total proportion of activating edges in the subnetworks, constraints that enforce that the subnetworks are acyclic, and constraints that require the predicted interfaces to be most-downstream nodes in the subnetwork. We argue that these constraints improve the interpretability of the subnetworks and our experimental results show that they do not reduce their accuracy.
  - In Chapter 4, we proposed constraints that allow over-represented protein complexes to be included in the inferred subnetworks.
  - In Chapter 4, we provided a representation of metabolic reactions as 'and' functions.
- For each application, we constructed biologically-motivated objective functions. In Chapters 4 and 5, we optimize multiple objective functions sequentially in order to selectively integrate different data types. While other related approaches generally combine multiple objectives into a single, weighted objective function, our sequential procedure allows us to specify an order to the objective functions without choosing a weight parameter on each one.
- In the cases considered here, the available experimental data and background network are incomplete. Therefore, many optimal subnetworks are possible. To distinguish

between high- and low-confidence predictions, our method infers an ensemble of subnetworks, rather than a single one. We proposed two techniques for generating the ensemble:

1. Finding multiple solutions to the integer linear program (Chapters 3 and 5). We observed that the inferred subnetworks' predictive accuracy is fairly robust to changes in the the size of the ensemble.
2. Randomly holding aside a small fraction of the input data used to infer each subnetwork in the ensemble (Chapter 4). Our hypothesis is that this sampling approach should reduce the effect of spurious interactions among the input data. Our results suggest that it improves the accuracy of low-confidence predictions.

- Our IP representation provides a straightforward way to seed the inferred subnetwork with domain knowledge in the form of a partial subnetwork. The relevance variables for known relevant nodes, edges, or paths can be fixed prior to inference.

### 6.1.3 Generation of and evaluation of subnetwork predictions

The inferred subnetwork ensemble can be used to make several kinds of detailed predictions. Predictions include not only which genes are relevant, but also the specific role of each gene product and interaction in the condition of interest.

- By thresholding the confidence values in the inferred subnetwork ensemble, a user can extract a high-confidence consensus subnetwork.
- In Chapter 4, we proposed a method for extracting a view from a consensus subnetwork. This view allows the user to focus on the parts of the inferred subnetwork that are most relevant to a set of input genes of their choice. We provide variations on the basic view-extraction method that can be used to identify specific, ordered relationships within the consensus subnetwork.
- In Chapter 5, we proposed a bifurcation score to make predictions about how the induced and repressed components of the environmental stress response are coordinated. This approach is generally applicable to other subnetworks inferred from source-target pairs if the targets can be divided into multiple clusters.

We demonstrated that in each application, our subnetwork method performs well under a series of different evaluations.

- In all chapters, we demonstrated our method's ability to predict held-aside input data and known relevant gene sets.
- In Chapters 3 and 5, we used permutation analyses to measure the degree to which the accuracy of our predictions is due to the topological properties of the input data in the background network. Although our inference method strongly outperforms the permuted baseline, we observe that the background network topology confers some predictive capability. We concur with Gillis & Pavlidis (2012) that accuracy results for network-based prediction methods should be presented in comparison to the results of a baseline method that takes the network topology into account.
- In Chapter 3, we defined consensus subnetworks in order to predict both additional relevant genes, as well as which host factors are host-virus interfaces. Several of these predictions are supported by the literature, or are known to be involved in relevant cellular pathways.
- In Chapter 5, we were fortunate to have our collaborators validate several predictions through further experimentation. The experiments confirm that nearly all of the tested, predicted regulators are involved in the salt stress response. Our method accurately predicts many of their affected targets, and in some cases, their position in the subnetwork relative to other known regulators. Furthermore, the inferred subnetworks suggest interesting connections that have sparked further experimental inquiry.

## 6.2 Future work

This research offers many promising directions for future research.

### 6.2.1 Open computational challenges in our approach

One of the computational bottlenecks in our approach is the enumeration of candidate paths, which becomes intractable as the requested search depth and size of the background network increases. As we transitioned from the yeast background network to human, we observed that increasing the numbers of candidate paths, with relevance variables tightly interdependent with each other and with the node and edge variables, challenged the IP solver. It may be informative to characterize the properties of background networks and input data that are most influential on the ease or difficulty of the resulting IP. It also is worth considering if it is possible to avoid generating candidate paths. Previous

work has studied the problem of efficiently orienting graphs based on input ordered pairs without enumerating paths (Charikar *et al.*, 1999; Arkin & Hassin, 2002; Medvedovsky *et al.*, 2008; Silverbush *et al.*, 2011; Silverbush & Sharan, 2014). It may be possible to extend these efficient orientation algorithms to also impose the greater variety of constraints that we consider, such as edge sign consistency and our proposed constraint for including only over-represented protein complexes.

Another opportunity lies in the sub-problem of identifying (or estimating) the distribution of relevant subnetworks. We have considered two methods: one based on finding multiple solutions to the IP, and one based on sampling the input data. With regard to the first method, our stability experiments in Chapter 5.3.9 suggested that varying the ensemble size did not appreciably change the predictions made by the ensembles. However, our results gave no indication of how thoroughly the CPLEX solver explored the solution space. In the development of our work in Chapter 4, we found that requesting multiple solutions for the the complete HIV IP was very challenging for CPLEX in terms of both time and memory required. Our sampling approach was more tractable in both respects. Moving forward, it may be useful to explore further ways for gathering an ensemble of subnetworks. Some ideas include varying the objective function or constraints applied during the inference of each individual subnetwork. It may also be worthwhile to study how to estimate how well an ensemble of subnetworks represents the space of possible relevant subnetworks. The fields of multi-objective programming and constraint logic programming may be relevant to future work in this area.

### 6.2.2 Comparative analysis of subnetworks

In this dissertation, we evaluated the subnetworks from each application separately. However, it may be useful to compare inferred subnetworks that represent different, related conditions: across different stresses or stress dosages, or across different viral strains in the same host. Just as genome sequences of different species have been compared to gain insight into their evolutionary history, so have interaction networks. Methods have been developed to quantify the frequency of regulatory motifs (Shen-Orr *et al.*, 2002) and to identify conserved structures (Kelley *et al.*, 2004; Sharan *et al.*, 2005; Koyutürk *et al.*, 2006; Tian *et al.*, 2007). These methods could be applied to the subnetworks inferred using our method.

### 6.2.3 Active learning and experimental design

Most subnetwork inference and candidate gene prioritization methods, ours included, seek to identify the most *likely* structure or set of candidates according to existing data. However, given that we know our data sets are incomplete and can induce multiple models that are equivalent according to our objective functions, it could be useful to reframe the problem as one of identifying which future experiments will give us the most *information* about the underlying structure and allow us to converge upon it with the least additional resources. In machine learning, we refer to this problem as *active learning*. Several approaches have addressed active learning of subnetwork structure, including edge signs, from single- or double-gene suppression experiments (Ideker *et al.*, 2000; Reiser *et al.*, 2001; King *et al.*, 2004; Pournara & Wernisch, 2004; Yeang *et al.*, 2005; Barrett & Palsson, 2006; Steinke *et al.*, 2007; King *et al.*, 2009; Szczurek *et al.*, 2009; Ray *et al.*, 2010) and to predict Gene Ontology annotations (Hibbs *et al.*, 2009). Another related task is to identify missing information from existing data sets. One category of methods identifies interactions that, if they were present, would enable the explanation of another data set: *e.g.*, allowing the completion of cliques (Yu *et al.*, 2006) or the connection of source-target pairs (Navlakha *et al.*, 2012) in a protein interaction background network. Other methods have been designed to identify incorrectly signed edges from signed source-target pairs (Gebser *et al.*, 2011; Melas *et al.*, 2013). Because our methods infer ensembles of subnetwork hypotheses that are equivalent with respect to the input data, they are well-suited to integration into an active learning framework for identifying the most useful next single- or double-gene suppression experiment (or batch of experiments) or for correcting inconsistencies.

### 6.2.4 Extension to new data types

Several opportunities for future work involve the integration of additional data types.

- In our studies of host-virus interactions, we use viral phenotype data from host gene suppression experiments performed using yeast deletion libraries, yeast knockdown libraries, and RNAi experiments. New methods for selectively suppressing, or otherwise controlling, gene expression in high-throughput experiments are continually under development; for example, CRISPR interference (Larson *et al.*, 2013). As data from these experiments becomes more widely available, it will become important to adapt our subnetwork inference method to take them as input.

- In Chapter 4, we discovered that the degree of a gene in the background network was a relatively strong predictor of relevance to HIV, and that our choice of graph

kernel (regularized Laplacian) did not provide very much more predictive benefit. Previous work has considered supervised learning methods for predicting host-virus interactions based on additional features, such as protein domain information (*e.g.*, Dyer *et al.* (2007); Tastan *et al.* (2009); Doolittle & Gomez (2010); Dyer *et al.* (2011)). Future work may investigate how to integrate different types of prioritization methods into our subnetwork inference method. The prioritization scores may be specific to the type of prediction, and may, for example, distinguish the direct interfaces from intermediate relevant genes.

- We explored a new representation for incorporating data about metabolic reactions and similar biological events. However, we did not observe very much incorporation of these data into the inferred subnetworks. Further work into understanding how to usefully integrate them may be very helpful.

- We observe a large gap between the biological information that is encoded in structured databases and the vast amounts of biological knowledge and experimental results that are available only in the literature. Being able to automatically extract information from text and use it to inform the inference method would greatly benefit the method's accuracy and interpretability. We believe it could be very fruitful to extend our method to incorporate information extracted by natural language processing tools.

- Our work in Chapter 4 only scrapes the surface of the available information about interactions between HIV and human cells. We have explored some data sources from experiments conducted on human cell culture. It may be useful to integrate information from orthogonal sources, such as the pharmacological mechanism of existing HIV therapies, or perhaps measurements of intracellular activity or other data from HIV-seropositive patients.

# Glossary

**background network** A graph of physical interactions between gene products. Although we know that available interaction data is incomplete, we use a background network to represent the possible space of physical interactions that could be involved in the phenotypes that we study. p. 3

**bifurcation point** A node in a signaling subnetwork that is important for coordinating complementary cellular responses. In our study of the environmental stress response, we define a bifurcation point as a node that a) is upstream of many genes that are either induced or repressed under stress, but b) has outgoing paths that relatively cleanly divide the induced target genes from the repressed target genes. p. 165

**consensus subnetwork** A high-confidence inferred subnetwork that is identified by thresholding the confidence values from an inferred subnetwork ensemble. p. 108

**deletion mutant** An organism (typically yeast), in which a specific gene or genomic region has been removed and replaced with a 'barcode' sequence that can be used for identifying successful cells in a pooled experiment. p. *see also* mutant

**differential expression** An expression measurement that is statistically significantly outside of the expected range. A common usage is to refer to genes that are differentially expressed in a mutant strain compared to the wild type. p. 15

**down** The viral phenotype label that we assign to a host gene whose suppression inhibits viral replication. p. 2, 31, 36

**downstream** Describes an ordered relationship between two gene products in a cellular pathway. If the action of gene A precedes the action of gene B, we say that gene B is downstream of gene A. p. *see also* upstream

**dox-repressible mutant** An organism (typically yeast), in which the promoter of a specific coding gene has been altered to allow for reversible, tunable adjustment of gene expression by the addition of docycycline. p. 3, *see also* mutant

**environmental stress response** A pattern of differential gene expression that is observed in yeast across many different stresses. p. 126

**fitness-contribution hit** The phenotype label that we assign to a yeast gene that is important for acquiring stress resistance after an initial salt stress. Identified as a single-gene mutant that tolerates a secondary stress less well than does the wild type. p. 130

**Gene Ontology** A resource for describing gene products according to terms in three hierarchical ontologies: Cellular Component, Molecular Function, and Biological Process. p. 3

**hit** A gene or gene product that has been experimentally implicated in a cellular phenotype of interest. p. 1

**HIV dependency factor** The viral phenotype label for a human gene whose suppression inhibits the replication of HIV. p. 99

**HIV restriction factor** The viral phenotype label for a human gene whose suppression enhances the replication of HIV. p. 99

**induction** Used to describe an increase in gene expression (or other measurement of cellular activity) compared to a control time-point or condition. p. 16

**inferred subnetwork** Produced by a subnetwork inference method. The nodes and edges from the background network that are predicted to be relevant to the phenotype of interest. The subnetwork also predicts additional information inferred about the phenotype labels of the nodes, the signs and directions of the edges, and the organization of the nodes into ordered paths. p. 4

**interface** One of the host factors that is most proximal to a direct interaction with a viral component; also referred to as *host-virus interface*. p. 4

**mutant** Describes the condition of having a specific genetic mutation; used in contrast to wild type. In this thesis, we primarily use the term mutant to refer to organisms whose genomes have been intentionally modified in order to suppress or alter the expression of a specific gene. Some examples include deletion mutants and dox-repressible mutants. p. 1, *see also* wild type

**no-effect** The viral phenotype label that we assign to a host gene whose suppression does not consistently inhibit or enhance viral replication. p. 32, 36

**path** A chain of physical interactions between gene products. We almost exclusively use the word to refer to acyclic paths. p. 4

**pathway** A sequence of biological events that result in a cellular phenotype. A pathway may not necessarily be completely ordered or linear. p. 2

**phenotype label** An experimentally-derived label that describes the magnitude and sign of the effect of a specific gene suppression experiment on a phenotype of interest; for example, a viral phenotype label describes the effect of a host gene suppression on viral replication. p. 2, 31

**phospho-proteomic hit** The phenotype label that we assign to a yeast protein that displays a significant increase or decrease in phosphorylation under NaCl stress as compared to normal conditions. p. 131

**repression** Used to describe a decrease in gene expression (or other measurement of cellular activity) compared to a control time-point or condition. p. 16

**RNA interference** A gene suppression method. Short RNAs that match a specific gene are introduced to the cell, triggering a targeted RNA degradation response. p. 1, 3

**sign** A sign on a directed edge describes the nature of the relationship between the interacting gene products: activating or inhibiting. A sign on a phenotype label similarly describes the nature of the gene suppression experiment on the phenotype. p. 2, 3

**sign consistency** (also, **consistent**) The idea that, for a particular signed and directed edge to be relevant to a phenotype, the phenotype labels of the interacting nodes must match the sign of the interaction. For edges that represent activation, the nodes should have the same phenotype label. For edges that represent repression, the nodes should have opposite phenotype labels. p. 5

**source-target pair** An experimental result that links the action of a source gene product to the expression of a target gene product. Often identified from differentially-expressed genes identified from single-gene mutants. The pair conveys a direction (that the source is upstream of the target) and sometimes a sign (whether the target was differentially repressed or induced in the source mutant). p. 15, 130

**transcriptome** The population of RNA products in a cell. p. 126

**unconfirmed** The viral phenotype label that we assign to all human genes that are not confirmed hits for HIV. The set includes genes that have not been assayed for HIV replication, as well as genes that have been assayed. p. 100

**unobserved** The viral phenotype label that we assign to yeast genes that have not been assayed for involvement in viral replication. p. 36

**up** The viral phenotype label that we assign to a host gene whose suppression enhances viral replication. p. 2, 31, 36

**upstream** Describes an ordered relationship between two gene products in a cellular pathway. If the action of gene A precedes the action of gene B, we say that gene A is upstream of gene B. p. 2, *see also* downstream

**view** Our term for a sub-set of a predicted, relevant subnetwork that is predicted to be most functionally similar to a set of provided query genes. p. 109

**weakly-up** The viral phenotype label that we assign to a host gene whose suppression weakly, but consistently, inhibits viral replication. p. 31, 36

**weakly-up** The viral phenotype label that we assign to a host gene whose suppression weakly, but consistently, enhances viral replication. p. 31, 36

**yeast deletion library** A genome-spanning collection of yeast nonessential gene mutants. p. 2

# Bibliography

Abdulrehman D, Monteiro PT, Teixeira MC, Mira NP, Lourenço AB, dos Santos SC, Cabrito TR, Francisco AP, Madeira SC, Aires RS, Oliveira AL, Sá-Correia I, Freitas AT (2011) YEASTRACT: providing a programmatic access to curated transcriptional regulatory associations in *Saccharomyces cerevisiae* through a web services interface. *Nucleic Acids Research* **39**: D136–D140

Adell MAY, Teis D (2011) Assembly and disassembly of the ESCRT-III membrane scission complex. *FEBS Letters* **585**: 3191–3196

Adrover MÀ, Zi Z, Duch A, Schaber J, González-Novo A, Jimenez J, Nadal-Ribelles M, Clotet J, Klipp E, Posas F (2011) Time-dependent quantitative multicomponent control of the $G_1$-S network by the stress-activated protein kinase Hog1 upon osmostress. *Science Signaling* **4**: ra63

Ahola T, den Boon JA, Ahlquist P (2000) Helicase and capping enzyme active site mutations in Brome Mosaic Virus protein 1a cause defects in template recruitment, negative-strand RNA synthesis, and viral RNA capping. *Journal of Virology* **74**: 8803–8811

Akutsu T, Miyano S, Kuhara S (1999) Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing* : 17–28

Akutsu T, Miyano S, Kuhara S (2000) Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function. *Journal of Computational Biology* **7**: 331–343

Alejandro-Osorio AL, Huebert DJ, Porcaro DT, Sonntag ME, Nillasithanukroh S, Will JL, Gasch AP (2009) The histone deacetylase Rpd3p is required for transient changes in genomic expression in response to stress. *Genome Biology* **10**: R57

Alexander MR, Tyers M, Perret M, Craig BM, Fang KS, Gustin MC (2001) Regulation of cell cycle progression by Swe1p and Hog1p following hypertonic stress. *Molecular Biology of the Cell* **12**: 53–62

Alon N, Yuster R, Zwick U (1995) Color-coding. *Journal of the Association for Computing Machinery (ACM)* **42**: 844–856

Araki Y, Takahashi S, Kobayashi T, Kajiho H, Hoshino S, Katada T (2001) Ski7p G protein interacts with the exosome and the Ski complex for 3′-to-5′ mRNA decay in yeast. *EMBO Journal* **20**: 4684–4693

Arkin EM, Hassin R (2002) A note on orientations of mixed graphs. *Discrete Applied Mathematics* **116**: 271–278

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G (2000) Gene Ontology: tool for the unification of biology. *Nature Genetics* **25**: 25–29

Avery L, Wasserman S (1992) Ordering gene function: the interpretation of epistasis in regulatory hierarchies. *Trends in Genetics* **8**: 312–316

Bader G, Hogue C (2003) An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* **4**: 2

Balakrishnan R, Park J, Karra K, Hitz BC, Binkley G, Hong EL, Sullivan J, Micklem G, Cherry JM (2012) YeastMine–an integrated data warehouse for *Saccharomyces cerevisiae* data as a multipurpose tool-kit. *Database (Oxford)* **2012**: bar062

Bar-Joseph Z, Gerber GK, Lee TI, Rinaldi NJ, Yoo JY, Robert F, Gordon DB, Fraenkel E, Jaakkola TS, Young RA, Gifford DK (2003) Computational discovery of gene modules and regulatory networks. *Nature Biotechnology* **21**: 1337–1342

Barna M (2013) Ribosomes take control. *Proceedings of the National Academy of Science USA* **110**: 9–10

Barrett CL, Palsson BO (2006) Iterative reconstruction of transcriptional regulatory networks: an algorithmic approach. *PLoS Computational Biology* **2**: e52

Bauer S, Gagneur J, Robinson PN (2010) GOing Bayesian: model-based gene set analysis of genome-scale data. *Nucleic Acids Research* **38**: 3523–3532

Bauer S, Robinson PN, Gagneur J (2011) Model-based gene set analysis for Bioconductor. *Bioinformatics* **27**: 1882–1883

Beckham CJ, Light HR, Nissan TA, Ahlquist P, Parker R, Noueiry A (2007) Interactions between Brome Mosaic Virus RNAs and cytoplasmic processing bodies. *Journal of Virology* **81**: 9759–9768

Bellí G, Garí E, Aldea M, Herrero E (2001) Osmotic stress causes a G1 cell cycle delay and downregulation of Cln3/Cdc28 activity in *Saccharomyces cerevisiae*. *Molecular Microbiology* **39**: 1022–1035

Bergkessel M, Whitworth GB, Guthrie C (2011) Diverse environmental stresses elicit distinct responses at the level of pre-mRNA processing in yeast. *RNA* **17**: 1461–1478

Berkey CD, Carlson M (2006) A specific catalytic subunit isoform of protein kinase CK2 is required for phosphorylation of the repressor Nrg1 in *Saccharomyces cerevisiae*. *Current Genetics* **50**: 1–10

Berry DB, Gasch AP (2008) Stress-activated genomic expression changes serve a preparative role for impending stress in yeast. *Molecular Biology of the Cell* **19**: 4580–4587

Berry DB, Guan Q, Hose J, Haroon S, Gebbia M, Heisler LE, Nislow C, Giaever G, Gasch A (2011) Multiple means to the same end: the genetic basis of acquired stress resistance in yeast. *PLoS Genetics* **7**: e1002353

Blanchette P, Branton PE (2009) Manipulation of the ubiquitin-proteasome pathway by small DNA tumor viruses. *Virology* **384**: 317–323

Blokh D, Segev D, Sharan R (2013) The approximability of shortest path-based graph orientations of protein-protein interaction networks. *Journal of Computational Biology* **20**: 945–957

Bonneau R, Reiss DJ, Shannon P, Facciotti M, Hood L, Baliga NS, Thorsson V (2006) The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology* **7**: R36

Börnigen D, Tranchevent LC, Bonachela-Capdevila F, Devriendt K, Moor BD, Causmaecker PD, Moreau Y (2012) An unbiased evaluation of gene prioritization tools. *Bioinformatics* **28**: 3081–3088

Brass AL, Dykxhoorn DM, Benita Y, Yan N, Engelman A, Xavier RJ, Lieberman J, Elledge SJ (2008) Identification of host proteins required for HIV infection through a functional genomic screen. *Science* **319**: 921–926

Breitkreutz A, Choi H, Sharom JR, Boucher L, Neduva V, Larsen B, Lin ZY, Breitkreutz BJ, Stark C, Liu G, Ahn J, Dewar-Darch D, Reguly T, Tang X, Almeida R, Qin ZS, Pawson T, Gingras AC, Nesvizhskii AI, Tyers M (2010) A global protein kinase and phosphatase interaction network in yeast. *Science* **328**: 1043–1046

Brem RB, Kruglyak L (2005) The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proceedings of the National Academy of Science USA* **102**: 1572–1577

Broach JR (2012) Nutritional control of growth and development in yeast. *Genetics* **192**: 73–105

Buratowski S (2003) The CTD code. *Nature Structural & Molecular Biology* **10**: 679–680

Butte AJ, Kohane IS (2000) Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing* : 418–429

Capaldi AP, Kaplan T, Liu Y, Habib N, Regev A, Friedman N, O'Shea EK (2008) Structure and function of a transcriptional network activated by the MAPK Hog1. *Nature Genetics* **40**: 1300–1306

Carter GW, Prinz S, Neou C, Shelby JP, Marzolf B, Thorsson V, Galitski T (2007) Prediction of phenotype and gene expression for combinations of mutations. *Molecular Systems Biology* **3**: 96

Causton HC, Ren B, Koh SS, Harbison CT, Kanin E, Jennings EG, Lee TI, True HL, Lander ES, Young RA (2001) Remodeling of yeast genome expression in response to environmental changes. *Molecular Biology of the Cell* **12**: 323–337

Cha HJ, Byrom M, Mead PE, Ellington AD, Wallingford JB, Marcotte EM (2012) Evolutionarily repurposed networks reveal the well-known antifungal drug thiabendazole to be a novel vascular disrupting agent. *PLoS Biology* **10**: e1001379

Charikar M, Chekuri C, Cheung T, Dai Z, Goel A, Guha S, Li M (1999) Approximation algorithms for directed Steiner Tree problems. *Journal of Algorithms* **33**: 73–91

Chen Y, Jiang T, Jiang R (2011) Uncover disease genes by maximizing information flow in the phenome-interactome network. *Bioinformatics* **27**: i167–i176

Cherry JM, Hong EL, Amundsen C, Balakrishnan R, Binkley G, Chan ET, Christie KR, Costanzo MC, Dwight SS, Engel SR, Fisk DG, Hirschman JE, Hitz BC, Karra K, Krieger CJ, Miyasato SR, Nash RS, Park J, Skrzypek MS, Simison M, *et al.* (2012) Saccharomyces Genome Database: the genomics resource of budding yeast. *Nucleic Acids Research* **40**: D700–D705

Cherry S, Doukas T, Armknecht S, Whelan S, Wang H, Sarnow P, Perrimon N (2005) Genome-wide RNAi screen reveals a specific sensitivity of IRES-containing RNA viruses to host translation inhibition. *Genes & Development* **19**: 445–452

Choi AG, Wong J, Marchant D, Luo H (2013) The ubiquitin-proteasome system in positive-strand RNA virus infection. *Reviews in Medical Virology* **23**: 85–96

Chukkapalli V, Heaton NS, Randall G (2012) Lipids at the interface of virus-host interactions. *Current Opinions in Microbiology* **15**: 512–518

Chymkowitch P, Eldholm V, Lorenz S, Zimmermann C, Lindvall JM, Bjørås M, Meza-Zepeda LA, Enserink JM (2012) Cdc28 kinase activity regulates the basal transcription machinery at a subset of genes. *Proceedings of the National Academy of Science USA* **109**: 10450–10455

Craven M, Ziegler M (2011) GADGET: Genes Associated by Documents, Genes, Events and Text. `http://gadget.biostat.wisc.edu`

Croft D, Mundo AF, Haw R, Milacic M, Weiser J, Wu G, Caudy M, Garapati P, Gillespie M, Kamdar MR, Jassal B, Jupe S, Matthews L, May B, Palatnik S, Rothfels K, Shamovsky V, Song H, Williams M, Birney E, *et al.* (2014) The Reactome pathway knowledgebase. *Nucleic Acids Research* **42**: D472–D477

Dahlhaus R, Eichler M (2003) Causality and graphical models in time series analysis. *Oxford Statistical Science Series* : 115–137

Danna E, Fenelon M, Gu Z, Wunderling R (2007) Generating multiple solutions for mixed integer programming problems. In *Proceedings of the 12th international conference on Integer Programming and Combinatorial Optimization*. Springer-Verlag, pp. 280–294

de Nadal E, Posas F (2010) Multilayered control of gene expression by stress-activated protein kinases. *The EMBO Journal* **29**: 4–13

Diaz A, Wang X, Ahlquist P (2010) Membrane-shaping host reticulon proteins play crucial roles in viral RNA replication compartment formation and function. *Proceedings of the National Academy of Science USA* **107**: 16291–16296

Dittrich MT, Klau GW, Rosenwald A, Dandekar T, Müller T (2008) Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics* **24**: i223–i231

Doolittle JM, Gomez SM (2010) Structural similarity-based predictions of protein interactions between HIV-1 and Homo sapiens. *Virology Journal* **7**: 82

Dutkowski J, Ideker T (2011) Protein networks as logic functions in development and cancer. *PLoS Computational Biology* **7**: e1002180

Dyer MD, Murali TM, Sobral BW (2007) Computational prediction of host-pathogen protein-protein interactions. *Bioinformatics* **23**: i159–i166

Dyer MD, Murali TM, Sobral BW (2011) Supervised learning and prediction of physical interactions between human and HIV proteins. *Infection, Genetics, and Evolution* **11**: 917–923

Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Science USA* **95**: 14863–14868

Ernst J, Plasterer HL, Simon I, Bar-Joseph Z (2010) Integrating multiple evidence sources to predict transcription factor binding in the human genome. *Genome Research* **20**: 526–536

Ernst J, Vainas O, Harbison CT, Simon I, Bar-Joseph Z (2007) Reconstructing dynamic regulatory maps. *Molecular Systems Biology* **3**: 74

Everett L, Vo A, Hannenhalli S (2009) PTM-Switchboard–a database of posttranslational modifications of transcription factors, the mediating enzymes and target genes. *Nucleic Acids Research* **37**: D66–D71

Fasolo J, Sboner A, Sun MGF, Yu H, Chen R, Sharon D, Kim PM, Gerstein M, Snyder M (2011) Diverse protein kinase interactions identified by protein microarrays reveal novel connections between cellular processes. *Genes & Development* **25**: 767–778

Faust K, Dupont P, Callut J, van Helden J (2010) Pathway discovery in metabolic networks by subgraph extraction. *Bioinformatics* **26**: 1211–1218

Forbes SA, Bindal N, Bamford S, Cole C, Kok CY, Beare D, Jia M, Shepherd R, Leung K, Menzies A, Teague JW, Campbell PJ, Stratton MR, Futreal PA (2011) COSMIC: mining complete cancer genomes in the Catalogue of Somatic Mutations in Cancer. *Nucleic Acids Research* **39**: D945–D950

Friedman N (2004) Inferring cellular networks using probabilistic graphical models. *Science* **303**: 799–805

Friedman N, Linial M, Nachman I, Pe'er D (2000) Using Bayesian Networks to analyze expression data. *Journal of Computational Biology* **7**: 601–620

Fu W, Sanders-Beer BE, Katz KS, Maglott DR, Pruitt KD, Ptak RG (2009) Human Immunodeficiency Virus Type 1, Human Protein Interaction Database at NCBI. *Nucleic Acids Research* **37**: D417–D422

GAMS Development Corporation (2010) General Algebraic Modeling System Version 23.6.5

Gancarz BL, Hao L, He Q, Newton MA, Ahlquist P (2011) Systematic identification of novel, essential host genes affecting bromovirus RNA replication. *PLoS ONE* **6**: e23988

Gao G, Luo H (2006) The ubiquitin-proteasome pathway in viral infections. *Canadian Journal of Physiology and Pharmacology* **84**: 5–14

Gasch A (2003) *Yeast Stress Responses*, chap. The Environmental Stress Response: a common yeast response to diverse environmental stresses. Springer-Verlag, pp. 11–70

Gasch AP, Huang M, Metzner S, Botstein D, Elledge SJ, Brown PO (2001) Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p. *Molecular Biology of the Cell* **12**: 2987–3003

Gasch AP, Spellman PT, Kao CM, Carmel-Harel O, Eisen MB, Storz G, Botstein D, Brown PO (2000) Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell* **11**: 4241–4257

Gat-Viks I, Shamir R (2007) Refinement and expansion of signaling pathways: the osmotic response network in yeast. *Genome Research* **17**: 358–367

Gat-Viks I, Tanay A, Raijman D, Shamir R (2006) A probabilistic methodology for integrating knowledge and experiments on biological networks. *Journal of Computational Biology* **13**: 165–181

Gebser M, Schaub T, Thiele S, Veber P (2011) Detecting inconsistencies in large biological networks with Answer Set Programming. *Theory and Practice of Logic Programming* **11**: 1–38

Geistlinger L, Csaba G, Küffner R, Mulder N, Zimmer R (2011) From sets to graphs: towards a realistic enrichment analysis of transcriptomic systems. *Bioinformatics* **27**: i366–i373

Gibbs DL, Baratt A, Baric RS, Kawaoka Y, Smith RD, Orwoll ES, Katze MG, McWeeney SK (2013) Protein co-expression network analysis (ProCoNA). *Journal of Clinical Bioinformatics* **3**: 11

Gillis J, Pavlidis P (2012) "Guilt by association" is the exception rather than the rule in gene networks. *PLoS Computational Biology* **8**: e1002444

Gitter A, Bar-Joseph Z (2013) Identifying proteins controlling key disease signaling pathways. *Bioinformatics* **29**: i227–i236

Gitter A, Braunstein A, Pagnani A, Baldassi C, Borgs C, Chayes J, Zecchina R, Fraenkel E (2014) Sharing information to reconstruct patient-specific pathways in heterogeneous diseases. *Pacific Symposium on Biocomputing* : 39–50

Gitter A, Carmi M, Barkai N, Bar-Joseph Z (2013) Linking the signaling cascades and dynamic regulatory networks controlling stress responses. *Genome Research* **23**: 365–376

Gitter A, Klein-Seetharaman J, Gupta A, Bar-Joseph Z (2011) Discovering pathways by orienting edges in protein interaction networks. *Nucleic Acids Research* **39**: e22

Guelzim N, Bottani S, Bourgine P, Képès F (2002) Topological and causal structure of the yeast transcriptional regulatory network. *Nature Genetics* **31**: 60–63

Guo Z, Wang L, Li Y, Gong X, Yao C, Ma W, Wang D, Li Y, Zhu J, Zhang M, Yang D, Rao S, Wang J (2007) Edge-based scoring and searching method for identifying condition-responsive protein-protein interaction sub-network. *Bioinformatics* **23**: 2121–2128

Halbeisen RE, Gerber AP (2009) Stress-dependent coordination of transcriptome and translatome in yeast. *PLoS Biology* **7**: e1000105

Hao L, He Q, Wang Z, Craven M, Newton MA, Ahlquist P (2013) Limited agreement of independent RNAi screens for virus-required host genes owes more to false-negative than false-positive factors. *PLoS Computational Biology* **9**: e1003235

Hao L, Lindenbach B, Wang X, Dye B, Kushner D, He Q, Newton M, Ahlquist P (2014) Genome-wide analysis of host factors in Nodavirus RNA replication. *PLoS ONE* **9**: e0095799

Hao L, Sakurai A, Watanabe T, Sorensen E, Nidom CA, Newton MA, Ahlquist P, Kawaoka Y (2008) Drosophila RNAi screen identifies host genes important for influenza virus replication. *Nature* **454**: 890–893

Heavner BD, Smallbone K, Barker B, Mendes P, Walker LP (2012) Yeast 5 - an expanded reconstruction of the *Saccharomyces cerevisiae* metabolic network. *BMC Systems Biology* **6**: 55

Hibbs MA, Myers CL, Huttenhower C, Hess DC, Li K, Caudy AA, Troyanskaya OG (2009) Directing experimental biology: a case study in mitochondrial biogenesis. *PLoS Computational Biology* **5**: e1000322

Hirasawa T, Ashitani K, Yoshikawa K, Nagahisa K, Furusawa C, Katakura Y, Shimizu H, Shioya S (2006) Comparison of transcriptional responses to osmotic stresses induced by NaCl and sorbitol additions in Saccharomyces cerevisiae using DNA microarray. *Journal of Bioscience and Bioengineering* **102**: 568–571

Hogan DJ, Riordan DP, Gerber AP, Herschlag D, Brown PO (2008) Diverse RNA-binding proteins interact with functionally related sets of RNAs, suggesting an extensive regulatory system. *PLoS Biology* **6**: e255

Hohmann S, Mager WH (eds.) (2003) *Yeast Stress Responses*. Springer-Verlag

Hsu DF, Taksa I (2005) Comparing rank and score combination methods for data fusion in information retrieval. *Information Retrieval* **8**: 449–480

Huang DW, Sherman BT, Lempicki RA (2009) Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research* **37**: 1–13

Huang SC, Clarke DC, Gosline SJC, Labadorf A, Chouinard CR, Gordon W, Lauffenburger DA, Fraenkel E (2013) Linking proteomic and transcriptional data through the interactome and epigenome reveals a map of oncogene-induced signaling. *PLoS Computational Biology* **9**: e1002887

Huang SSC, Fraenkel E (2009) Integrating proteomic, transcriptional, and interactome data reveals hidden components of signaling and regulatory networks. *Science Signaling* **2**: ra40

Huang SSC, Fraenkel E (2012) Swimming upstream: identifying proteomic signals that drive transcriptional changes using the interactome and multiple "-omics" datasets. *Methods in Cell Biology* **110**: 57–80

Huebert DJ, Gasch AP (2012) Defining flexible vs. inherent promoter architectures: the importance of dynamics and environmental considerations. *Nucleus* **3**: 399–403

Huebert DJ, Kuan PF, Keleş S, Gasch AP (2012) Dynamic changes in nucleosome occupancy are not predictive of gene expression dynamics but are linked to transcription and chromatin regulators. *Molecular and Cellular Biology* **32**: 1645–1653

Hurley JH, Hanson PI (2010) Membrane budding and scission by the ESCRT machinery: it's all in the neck. *Nature Reviews Molecular Cell Biology* **11**: 556–566

IBM (2012) IBM ILOG CPLEX Optimization Studio, Version 12.4.0.1

Ideker T, Ozier O, Schwikowski B, Siegel AF (2002) Discovering regulatory and signaling circuits in molecular interaction networks. *Bioinformatics* **18**: S233–240

Ideker TE, Thorsson V, Karp RM (2000) Discovery of regulatory interactions through perturbation: inference and experimental design. *Pacific Symposium on Biocomputing* **5**: 305–316

Imoto S, Higuchi T, Goto T, Tashiro K, Kuhara S, Miyano S (2003) Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *Proceedings of the IEEE Computer Society Bioinformatics Conference* **2**: 104–113

Iwaki A, Izawa S (2012) Acidic stress induces the formation of P-bodies, but not stress granules, with mild attenuation of bulk translation in *Saccharomyces cerevisiae*. *Biochemical Journal* **446**: 225–233

Jäger S, Cimermancic P, Gulbahce N, Johnson JR, McGovern KE, Clarke SC, Shales M, Mercenne G, Pache L, Li K, Hernandez H, Jang GM, Roth SL, Akiva E, Marlett J, Stephens M, D'Orso I, Fernandes J, Fahey M, Mahon C, *et al.* (2012) Global landscape of HIV-human protein complexes. *Nature* **481**: 365–370

Jones RG, Thompson CB (2009) Tumor suppressors and cell metabolism: a recipe for cancer growth. *Genes & Development* **23**: 537–548

Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M (2012) KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research* **40**: D109–D114

Kato T, Tsuda K, Asai K (2005) Selective integration of multiple biological data for supervised network inference. *Bioinformatics* **21**: 2488–2495

Kaufman A, Keinan A, Meilijson I, Kupiec M, Ruppin E (2005) Quantitative analysis of genetic and neuronal multi-perturbation experiments. *PLoS Computational Biology* **1**: e64

Kelley BP, Yuan B, Lewitter F, Sharan R, Stockwell BR, Ideker T (2004) PathBLAST: a tool for alignment of protein interaction networks. *Nucleic Acids Research* **32**: W83–W88

Kelley R, Ideker T (2005) Systematic interpretation of genetic interactions using protein networks. *Nature Biotechnology* **23**: 561–566

Kim SH, Palukaitis P, Park YI (2002) Phosphorylation of cucumber mosaic virus RNA polymerase 2a protein inhibits formation of replicase complex. *EMBO Journal* **21**: 2292–2300

Kim YA, Wuchty S, Przytycka TM (2011) Identifying causal genes and dysregulated pathways in complex diseases. *PLoS Computational Biology* **7**: e1001095

King RD, Rowland J, Oliver SG, Young M, Aubrey W, Byrne E, Liakata M, Markham M, Pir P, Soldatova LN, Sparkes A, Whelan KE, Clare A (2009) The automation of science. *Science* **324**: 85–88

King RD, Whelan KE, Jones FM, Reiser PGK, Bryant CH, Muggleton SH, Kell DB, Oliver SG (2004) Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature* **427**: 247–252

Köhler S, Bauer S, Horn D, Robinson PN (2008) Walking the interactome for prioritization of candidate disease genes. *American Journal of Human Genetics* **82**: 949–958

König R, Zhou Y, Elleder D, Diamond TL, Bonamy GMC, Irelan JT, Chiang CY, Tu BP, Jesus PDD, Lilley CE, Seidel S, Opaluch AM, Caldwell JS, Weitzman MD, Kuhen KL, Bandyopadhyay S, Ideker T, Orth AP, Miraglia LJ, Bushman FD, *et al.* (2008) Global analysis of host-pathogen interactions that regulate early-stage HIV-1 replication. *Cell* **135**: 49–60

Koyutürk M, Kim Y, Topkara U, Subramaniam S, Szpankowski W, Grama A (2006) Pairwise alignment of protein interaction networks. *Journal of Computational Biology* **13**: 182–199

Kulp DC, Jagalur M (2006) Causal inference of regulator-target pairs by gene mapping of expression phenotypes. *BMC Genomics* **7**: 125

Kushner DB, Lindenbach BD, Grdzelishvili VZ, Noueiry AO, Paul SM, Ahlquist P (2003) Systematic, genome-wide identification of host genes affecting replication of a positive-strand RNA virus. *Proceedings of the National Academy of Sciences USA* **100**: 15764–15769

Lan A, Smoly IY, Rapaport G, Lindquist S, Fraenkel E, Yeger-Lotem E (2011) ResponseNet: revealing signaling and regulatory networks linking genetic and transcriptomic screening data. *Nucleic Acids Research* **39**: W424–W429

Larson MH, Gilbert LA, Wang X, Lim WA, Weissman JS, Qi LS (2013) CRISPR interference (CRISPRi) for sequence-specific control of gene expression. *Nature Protocols* **8**: 2180–2196

Lee SA, Chan CH, Chen TC, Yang CY, Huang KC, Tsai CH, Lai JM, Wang FS, Kao CY, Huang CYF (2009) POINeT: protein interactome with sub-network analysis and hub prioritization. *BMC Bioinformatics* **10**: 114

Lee SI, Pe'er D, Dudley AM, Church GM, Koller D (2006) Identifying regulatory mechanisms using individual variation reveals key role for chromatin modification. *Proceedings of the National Academy of Sciences USA* **103**: 14062–14067

Lee WM, Ahlquist P (2003) Membrane synthesis, specific lipid requirements, and localized lipid composition changes associated with a positive-strand RNA virus RNA replication protein. *Journal of Virology* **77**: 12819–12828

Liang S, Fuhrman S, Somogyi R (1998) Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing* **3**: 18–29

Lippert C, Ghahramani Z, Borgwardt KM (2010) Gene function prediction from synthetic lethality networks via ranking on demand. *Bioinformatics* **26**: 912–918

Liu L, Oliveira NMM, Cheney KM, Pade C, Dreja H, Bergin AMH, Borgdorff V, Beach DH, Bishop CL, Dittmar MT, McKnight A (2011) A whole genome screen for HIV restriction factors. *Retrovirology* **8**: 94

Liu L, Westler WM, den Boon JA, Wang X, Diaz A, Steinberg HA, Ahlquist P (2009) An amphipathic alpha-helix controls multiple roles of brome mosaic virus protein 1a in RNA replication complex assembly and function. *PLoS Pathogens* **5**: e1000351

Liu M, Liberzon A, Kong SW, Lai WR, Park PJ, Kohane IS, Kasif S (2007) Network-based analysis of affected biological processes in type 2 diabetes models. *PLoS Genetics* **3**: e96

Lopes TJS, Schaefer M, Shoemaker J, Matsuoka Y, Fontaine JF, Neumann G, Andrade-Navarro MA, Kawaoka Y, Kitano H (2011) Tissue-specific subnetworks and characteristics of publicly available human protein interaction databases. *Bioinformatics* **27**: 2414–2421

Lu Y, Rosenfeld R, Simon I, Nau GJ, Bar-Joseph Z (2008) A probabilistic generative model for GO enrichment analysis. *Nucleic Acids Research* **36**: e109

Ma H, Schadt EE, Kaplan LM, Zhao H (2011) COSINE: COndition-SpecIfic sub-NEtwork identification using a global optimization method. *Bioinformatics* **27**: 1290–1298

MacIsaac KD, Wang T, Gordon DB, Gifford DK, Stormo GD, Fraenkel E (2006) An improved map of conserved regulatory sites for *Saccharomyces cerevisiae*. *BMC Bioinformatics* **7**: 113

Maeyer DD, Renkens J, Cloots L, Raedt LD, Marchal K (2013) PheNetic: network-based interpretation of unstructured gene lists in E. coli. *Molecular BioSystems* **9**: 1594–1603

Maraziotis IA, Dimitrakopoulou K, Bezerianos A (2007) Growing functional modules from a seed protein via integration of protein interaction and gene expression data. *BMC Bioinformatics* **8**: 408

Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera RD, Califano A (2006) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* **7 Suppl 1**: S7

Markowetz F (2010) How to understand the cell by breaking it: network analysis of gene perturbation screens. *PLoS Computational Biology* **6**: e1000655

Markowetz F, Bloch J, Spang R (2005) Non-transcriptional pathway features reconstructed from secondary effects of RNA interference. *Bioinformatics* **21**: 4026–4032

Marles JA, Dahesh S, Haynes J, Andrews BJ, Davidson AR (2004) Protein-protein interaction affinity plays a crucial role in controlling the Sho1p-mediated signal transduction pathway in yeast. *Molecular Cell* **14**: 813–823

Martínez-Montañés F, Pascual-Ahuir A, Proft M (2010) Toward a genomic view of the gene expression program regulated by osmostress in yeast. *OMICS* **14**: 619–627

McClean MN, Mody A, Broach JR, Ramanathan S (2007) Cross-talk and decision making in MAP kinase pathways. *Nature Genetics* **39**: 409–414

McGary KL, Park TJ, Woods JO, Cha HJ, Wallingford JB, Marcotte EM (2010) Systematic discovery of nonobvious human disease models through orthologous phenotypes. *Proceedings of the National Academy of Sciences USA* **107**: 6544–6549

Medvedovsky A, Bafna V, Zwick U, Sharan R (2008) An algorithm for orienting graphs based on cause-effect pairs and its applications to orienting protein networks. In *Proceedings of the 8th International Workshop on Algorithms in Bioinformatics*, Springer-Verlag, pp. 222–232

Melamed D, Pnueli L, Arava Y (2008) Yeast translational response to high salinity: global analysis reveals regulation at multiple levels. *RNA* **14**: 1337–1351

Melas IN, Samaga R, Alexopoulos LG, Klamt S (2013) Detecting and removing inconsistencies between experimental data and signaling network topologies using integer linear programming on interaction graphs. *PLoS Computational Biology* **9**: e1003204

Miller C, Schwalb B, Maier K, Schulz D, Dümcke S, Zacher B, Mayer A, Sydow J, Marcinowski L, Dölken L, Martin DE, Tresch A, Cramer P (2011) Dynamic transcriptome analysis measures rates of mRNA synthesis and decay in yeast. *Molecular Systems Biology* **7**: 458

Mitchell A, Romano GH, Groisman B, Yona A, Dekel E, Kupiec M, Dahan O, Pilpel Y (2009) Adaptive prediction of environmental changes by microorganisms. *Nature* **460**: 220–224

Mnaimneh S, Davierwala AP, Haynes J, Moffat J, Peng WT, Zhang W, Yang X, Pootoolal J, Chua G, Lopez A, Trochesset M, Morse D, Krogan NJ, Hiley SL, Li Z, Morris Q, Grigull J, Mitsakakis N, Roberts CJ, Greenblatt JF, *et al.* (2004) Exploration of essential gene functions via titratable promoter alleles. *Cell* **118**: 31–44

Montague E, Stanberry L, Higdon R, Janko I, Lee E, Anderson N, Choiniere J, Stewart E, Yandl G, Broomall W, Kolker N, Kolker E (2014) MOPED 2.5–an integrated multi-omics resource: multi-omics profiling expression database now includes transcriptomics data. *OMICS* **18**: 335–343

Murali TM, Dyer MD, Badger D, Tyler BM, Katze MG (2011) Network-based prediction and analysis of HIV Dependency Factors. *PLoS Computational Biology* **7**: e1002164

Nagiec MJ, Dohlman HG (2012) Checkpoints in a yeast differentiation pathway coordinate signaling during hyperosmotic stress. *PLoS Genetics* **8**: e1002437

Navlakha S, Gitter A, Bar-Joseph Z (2012) A network-based approach for predicting missing pathway interactions. *PLoS Computational Biology* **8**: e1002640

Navlakha S, Kingsford C (2010) The power of protein interaction networks for associating genes with diseases. *Bioinformatics* **26**: 1057–1063

Newman RH, Hu J, Rho HS, Xie Z, Woodard C, Neiswinger J, Cooper C, Shirley M, Clark HM, Hu S, Hwang W, Jeong JS, Wu G, Lin J, Gao X, Ni Q, Goel R, Xia S, Ji H, Dalby KN, *et al.* (2013) Construction of human activity-based phosphorylation networks. *Molecular Systems Biology* **9**: 655

Ni L, Bruce C, Hart C, Leigh-Bell J, Gelperin D, Umansky L, Gerstein MB, Snyder M (2009) Dynamic and complex transcription factor binding during an inducible response in yeast. *Genes & Development* **23**: 1351–1363

Nibbe RK, Koyutürk M, Chance MR (2010) An integrative -omics approach to identify functional sub-networks in human colorectal cancer. *PLoS Computational Biology* **6**: e1000639

Noueiry AO, Díez J, Falk SP, Chen J, Ahlquist P (2003) Yeast Lsm1p-7p/Pat1p deadenylation-dependent mRNA-decapping factors are required for Brome Mosaic Virus genomic RNA translation. *Molecular & Cellular Biology* **23**: 4094–4106

Novershtern N, Regev A, Friedman N (2011) Physical Module Networks: an integrative approach for reconstructing transcription regulation. *Bioinformatics* **27**: i177–i185

Ong IM, Glasner JD, Page D (2002) Modelling regulatory pathways in E. coli from time series expression profiles. *Bioinformatics* **18 Suppl 1**: S241–S248

Ong IM, Topper SE, Page D, Santos Costa V (2007) Inferring regulatory networks from time series expression data and relational data via inductive logic programming. In *Proceedings of the Sixteenth International Conference on Inductive Logic Programming*, vol. 4455 of *Lecture Notes in Computer Science*. Springer, pp. 366–378

O'Rourke SM, Herskowitz I (1998) The Hog1 MAPK prevents cross talk between the HOG and pheromone response MAPK pathways in Saccharomyces cerevisiae. *Genes & Development* **12**: 2874–2886

O'Rourke SM, Herskowitz I (2004) Unique and redundant roles for HOG MAPK pathway components as revealed by whole-genome expression analysis. *Molecular Biology of the Cell* **15**: 532–542

Ourfali O, Shlomi T, Ideker T, Ruppin E, Sharan R (2007) SPINE: a framework for signaling-regulatory pathway inference from cause-effect experiments. *Bioinformatics* **23**: i359–i366

Patterson JC, Klimenko ES, Thorner J (2010) Single-cell analysis reveals that insulation maintains signaling specificity between two yeast MAPK pathways with common components. *Science Signaling* **3**: ra75

Pavlidis P, Weston J, Cai J, Noble WS (2002) Learning gene functional classifications from multiple data types. *Journal of Computational Biology* **9**: 401–411

Peleg T, Yosef N, Ruppin E, Sharan R (2010) Network-free inference of knockout effects in yeast. *PLoS Computational Biology* **6**: e1000635

Pérez-Enciso M, Quevedo JR, Bahamonde A (2007) Genetical genomics: use all data. *BMC Genomics* **8**: 69

Pournara I, Wernisch L (2004) Reconstruction of gene networks using Bayesian learning and manipulation experiments. *Bioinformatics* **20**: 2934–2942

Prasad TSK, Goel R, Kandasamy K, Keerthikumar S, Kumar S, Mathivanan S, Telikicherla D, Raju R, Shafreen B, Venugopal A, Balakrishnan L, Marimuthu A, Banerjee S, Somanathan DS, Sebastian A, Rani S, Ray S, Kishore CJH, Kanth S, Ahmed M, *et al.* (2009) Human Protein Reference Database–2009 update. *Nucleic Acids Research* **37**: D767–D772

Ptacek J, Devgan G, Michaud G, Zhu H, Zhu X, Fasolo J, Guo H, Jona G, Breitkreutz A, Sopko R, McCartney RR, Schmidt MC, Rachidi N, Lee SJ, Mah AS, Meng L, Stark MJR, Stern DF, Virgilio CD, Tyers M, *et al.* (2005) Global analysis of protein phosphorylation in yeast. *Nature* **438**: 679–684

Pu S, Wong J, Turner B, Cho E, Wodak SJ (2009) Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Research* **37**: 825–831

Quinlan JR (1986) Induction of decision trees. *Machine Learning* **1**: 81–106

Ray O, Whelan K, King R (2010) Automatic revision of metabolic networks through logical analysis of experimental data. In de Raedt L (ed.) *Proceedings of the Nineteenth International Conference on Inductive Logic Programming*, vol. 5989 of *Lecture Notes in Artificial Intelligence*. Springer, pp. 194–201

Reiser PGK, King RD, Kell DB, Muggleton SH, Bryant CH, Oliver SG (2001) Developing a logical model of yeast metabolism. *Electronic Transactions in Artificial Intelligence* **5**: 233–244

Restrepo-Hartwig M, Ahlquist P (1999) Brome mosaic virus RNA replication proteins 1a and 2a colocalize and 1a independently localizes on the yeast endoplasmic reticulum. *Journal of Virology* **73**: 10303–10309

Rudra D, Mallick J, Zhao Y, Warner JR (2007) Potential interface between ribosomal protein production and pre-rRNA processing. *Molecular and Cellular Biology* **27**: 4815–4824

Sachs K, Perez O, Pe'er D, Lauffenburger DA, Nolan GP (2005) Causal protein-signaling networks derived from multiparameter single-cell data. *Science* **308**: 523–529

Saez-Rodriguez J, Alexopoulos LG, Epperlein J, Samaga R, Lauffenburger DA, Klamt S, Sorger PK (2009) Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Molecular Systems Biology* **5**: 331

Saito H, Tatebayashi K (2004) Regulation of the osmoregulatory HOG MAPK cascade in yeast. *Journal of Biochemistry* **136**: 267–272

Schadt EE, Lamb J, Yang X, Zhu J, Edwards S, Guhathakurta D, Sieberts SK, Monks S, Reitman M, Zhang C, Lum PY, Leonardson A, Thieringer R, Metzger JM, Yang L, Castle J, Zhu H, Kash SF, Drake TA, Sachs A, *et al.* (2005) An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics* **37**: 710–717

Schaefer MH, Fontaine JF, Vinayagam A, Porras P, Wanker EE, Andrade-Navarro MA (2012) HIPPIE: Integrating protein interaction networks with experiment based quality scores. *PLoS ONE* **7**: e31826

Schaefer MH, Lopes TJS, Mah N, Shoemaker JE, Matsuoka Y, Fontaine JF, Louis-Jeune C, Eisfeld AJ, Neumann G, Perez-Iratxeta C, Kawaoka Y, Kitano H, Andrade-Navarro MA (2013) Adding protein context to the human protein-protein interaction network to reveal meaningful interactions. *PLoS Computational Biology* **9**: e1002860

Scherrer T, Mittal N, Janga SC, Gerber AP (2010) A screen for RNA-binding proteins in yeast indicates dual functions for many enzymes. *PLoS ONE* **5**: e15499

Schulz MH, Devanny WE, Gitter A, Zhong S, Ernst J, Bar-Joseph Z (2012) DREM 2.0: Improved reconstruction of dynamic regulatory networks from time-series expression data. *BMC Systems Biology* **6**: 104

Schwartz M, Chen J, Janda M, Sullivan M, den Boon J, Ahlquist P (2002) A positive-strand RNA virus replication complex parallels form and function of retrovirus capsids. *Molecular Cell* **9**: 505–514

Scott J, Ideker T, Karp RM, Sharan R (2006) Efficient algorithms for detecting signaling pathways in protein interaction networks. *Journal of Computational Biology* **13**: 133–144

Scott MS, Perkins T, Bunnell S, Pepin F, Thomas DY, Hallett M (2005) Identifying regulatory subnetworks for a set of genes. *Molecular & Cellular Proteomics* **4**: 683–692

Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, Friedman N (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics* **34**: 166–176

Serviene E, Jiang Y, Cheng CP, Baker J, Nagy PD (2006) Screening of the yeast yTHC collection identifies essential host factors affecting tombusvirus RNA recombination. *Journal of Virology* **80**: 1231–1241

Serviene E, Shapka N, Cheng CP, Panavas T, Phuangrat B, Baker J, Nagy PD (2005) Genome-wide screen identifies host genes affecting viral RNA recombination. *Proceedings of the National Academy of Sciences USA* **102**: 10545–10550

Sharan R, Suthram S, Kelley RM, Kuhn T, McCuine S, Uetz P, Sittler T, Karp RM, Ideker T (2005) Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Science USA* **102**: 1974–1979

Sharifpoor S, Ba ANN, Young JY, van Dyk D, Friesen H, Douglas AC, Kurat CF, Chong YT, Founk K, Moses AM, Andrews BJ (2011) A quantitative literature-curated gold standard for kinase-substrate pairs. *Genome Biology* **12**: R39

Shen-Orr SS, Milo R, Mangan S, Alon U (2002) Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics* **31**: 64–68

Shih YK, Parthasarathy S (2012) A single source k-shortest paths algorithm to infer regulatory pathways in a gene network. *Bioinformatics* **28**: i49–i58

Shock TR, Thompson J, Yates JR, Madhani HD (2009) Hog1 mitogen-activated protein kinase (MAPK) interrupts signal transduction between the Kss1 MAPK and the Tec1 transcription factor to maintain pathway specificity. *Eukaryot Cell* **8**: 606–616

Silverbush D, Elberfeld M, Sharan R (2011) Optimally orienting physical networks. In *Proceedings of the 15th Annual International Conference on Research in Computational Molecular Biology (RECOMB)*. Springer-Verlag, pp. 424–436

Silverbush D, Sharan R (2014) Network orientation via shortest paths. *Bioinformatics* **30**: 1449–1455

Smets B, Ghillebert R, Snijder PD, Binda M, Swinnen E, Virgilio CD, Winderickx J (2010) Life in the midst of scarcity: adaptations to nutrient availability in *Saccharomyces cerevisiae*. *Current Genetics* **56**: 1–32

Smola A, Kondor R (2003) Kernels and Regularization on Graphs. In Schölkopf B, Warmuth M (eds.) *Proceedings of the Annual Conference on Computational Learning Theory and Kernel Workshop*, vol. 2777 of *Lecture Notes in Artificial Intelligence*. Springer, pp. 144–158

Soufi B, Kelstrup CD, Stoehr G, Fröhlich F, Walther TC, Olsen JV (2009) Global analysis of the yeast osmotic stress response by quantitative proteomics. *Molecular BioSystems* **5**: 1337–1346

Stark C, Breitkreutz BJJ, Reguly T, Boucher L, Breitkreutz A, Tyers M (2006) BioGRID: a general repository for interaction datasets. *Nucleic Acids Research* **34**

Steffen M, Petti A, Aach J, D'haeseleer P, Church G (2002) Automated modelling of signal transduction networks. *BMC Bioinformatics* **3**: 34

Steinke F, Seeger M, Tsuda K (2007) Experimental design for efficient identification of gene regulatory networks using sparse Bayesian models. *BMC Systems Biology* **1**: 51

Su J, Yoon BJ, Dougherty ER (2010) Identification of diagnostic subnetwork markers for cancer in human protein-protein interaction network. *BMC Bioinformatics* **11 Suppl 6**: S8

Sundquist WI, Kräusslich HG (2012) HIV-1 assembly, budding, and maturation. *Cold Spring Harbor Perspectives in Medicine* **2**: a006924

Suthram S, Beyer A, Karp RM, Eldar Y, Ideker T (2008) eQED: an efficient method for interpreting eQTL associations using protein networks. *Molecular Systems Biology* **4**: 162

Szczurek E, Gat-Viks I, Tiuryn J, Vingron M (2009) Elucidating regulatory mechanisms downstream of a signaling pathway using informative experiments. *Molecular Systems Biology* **5**: 287

Tamaddoni-Nezhad A, Chaleil R, Kakas A, Muggleton S (2006) Application of abductive ILP to learning metabolic network inhibition from temporal data. *Machine Learning* **64**: 209–230

Tastan O, Qi Y, Carbonell JG, Klein-Seetharaman J (2009) Prediction of interactions between HIV-1 and human proteins by information integration. *Pac Symp Biocomput* : 516–527

Tekir SD, Cakir T, Ardiç E, Sayilirbas AS, Konuk G, Konuk M, Sariyer H, Ugurlu A, Karadeniz I, Ozgür A, Sevilgen FE, Ulgen KÖ (2013) PHISTO: pathogen-host interaction search tool. *Bioinformatics*

Tian Y, McEachin RC, Santos C, States DJ, Patel JM (2007) SAGA: a subgraph matching tool for biological graphs. *Bioinformatics* **23**: 232–239

Tiger CF, Krause F, Cedersund G, Palmér R, Klipp E, Hohmann S, Kitano H, Krantz M (2012) A framework for mapping, visualisation and automatic model creation of signal-transduction networks. *Molecular Systems Biology* **8**: 578

Tomita Y, Mizuno T, Díez J, Naito S, Ahlquist P, Ishikawa M (2003) Mutation of host DnaJ homolog inhibits brome mosaic virus negative-strand RNA synthesis. *Journal of Virology* **77**: 2990–2997

Tsuda K, Shin H, Schölkopf B (2005) Fast protein classification with multiple networks. *Bioinformatics* **21 Suppl 2**: ii59–ii65

Tsvetanova NG, Klass DM, Salzman J, Brown PO (2010) Proteome-wide search reveals unexpected RNA-binding proteins in Saccharomyces cerevisiae. *PLoS ONE* **5**

Tu Z, Wang L, Arbeitman MN, Chen T, Sun F (2006) An integrative approach for causal gene identification and gene regulatory pathway inference. *Bioinformatics* **22**: e489–e496

Tuncbag N, Braunstein A, Pagnani A, Huang SSC, 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**: 124–136

Ulitsky I, Shamir R (2007) Identification of functional modules using network topology and high-throughput data. *BMC Systems Biology* **1**

Van Wuytswinkel O, Reiser V, Siderius M, Kelders MC, Ammerer G, Ruis H, Mager WH (2000) Response of *Saccharomyces cerevisiae* to severe osmotic stress: evidence for a novel activation mechanism of the HOG MAP kinase pathway. *Molecular Microbiology* **37**: 382–397

Vanunu O, Magger O, Ruppin E, Shlomi T, Sharan R (2010) Associating genes and protein complexes with disease via network propagation. *PLoS Computational Biology* **6**: e1000641

Vaske CJ, House C, Luu T, Frank B, Yeang CH, Lee NH, Stuart JM (2009) A factor graph nested effects model to identify networks from genetic perturbations. *PLoS Computational Biology* **5**: e1000274

Venters BJ, Wachi S, Mavrich TN, Andersen BE, Jena P, Sinnamon AJ, Jain P, Rolleri NS, Jiang C, Hemeryck-Walsh C, Pugh BF (2011) A comprehensive genomic binding map of gene and chromatin regulatory proteins in *Saccharomyces. Molecular Cell* **41**: 480–492

Verbeke LPC, Cloots L, Demeester P, Fostier J, Marchal K (2013) EPSILON: an eQTL prioritization framework using similarity measures derived from local networks. *Bioinformatics* **29**: 1308–1316

Wall DP, Fraser HB, Hirsh AE (2003) Detecting putative orthologs. *Bioinformatics* **19**: 1710–1711

Wang X, Diaz A, Hao L, Gancarz B, den Boon JA, Ahlquist P (2011) Intersection of the multivesicular body pathway and lipid homeostasis in RNA replication by a positive-strand RNA virus. *Journal of Virology* **85**: 5494–5503

Wang Z, He Q, Larget B, Newton MA (2013) A multi-functional analyzer uses parameter constraints to improve the efficiency of model-based gene-set analysis. Technical Report 1174, University of Wisconsin–Madison, Department of Statistics

Warringer J, Hult M, Regot S, Posas F, Sunnerhagen P (2010) The HOG pathway dictates the short-term translational response after hyperosmotic shock. *Molecular Biology of the Cell* **21**: 3080–3092

Westfall PJ, Patterson JC, Chen RE, Thorner J (2008) Stress resistance and signal fidelity independent of nuclear MAPK function. *Proceedings of the National Academy of Science USA* **105**: 12212–12217

Wickner RB (1996) Double-stranded RNA viruses of *Saccharomyces cerevisiae. Microbiology and Molecular Biology Reviews* **60**: 250–265

Winzeler EA, Shoemaker DD, Astromoff A, Liang H, Anderson K, Andre B, Bangham R, Benito R, Boeke JD, Bussey H, Chu AM, Connelly C, Davis K, Dietrich F, Dow SW, Bakkoury ME, Foury F, Friend SH, Gentalen E, Giaever G, *et al.* (1999) Functional characterization of the *S. cerevisiae* genome by gene deletion and parallel analysis. *Science* **285**: 901–906

Woods JO, Singh-Blom UM, Laurent JM, McGary KL, Marcotte EM (2013) Prediction of gene-phenotype associations in humans, mice, and plants using phenologs. *BMC Bioinformatics* **14**: 203

Yang L, Walker JR, Hogenesch JB, Thomas RS (2008) NetAtlas: a Cytoscape plugin to examine signaling networks based on tissue gene expression. *In Silico Biology* **8**: 47–52

Yeang CH, Ideker T, Jaakkola T (2004) Physical network models. *Journal of Computational Biology* **11**: 243–262

Yeang CH, Mak CH, McCuine S, Workman C, Jaakkola T, Ideker T (2005) Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome Biology* **6**: R62

Yeger-Lotem E, Riva L, Su LJ, Gitler AD, Cashikar AG, King OD, Auluck PK, Geddie ML, Valastyan JS, Karger DR, Lindquist S, Fraenkel E (2009) Bridging high-throughput genetic and transcriptional data reveals cellular responses to alpha-synuclein toxicity. *Nature Genetics* **41**: 316–323

Yeung ML, Houzet L, Yedavalli VSRK, Jeang KT (2009) A genome-wide short hairpin RNA screening of jurkat T-cells for human proteins contributing to productive HIV-1 replication. *Journal of Biological Chemistry* **284**: 19463–19473

Yosef N, Kaufman A, Ruppin E (2006) Inferring functional pathways from multi-perturbation data. *Bioinformatics* **22**: e539–e546

Yosef N, Ungar L, Zalckvar E, Kimchi A, Kupiec M, Ruppin E, Sharan R (2009) Toward accurate reconstruction of functional protein networks. *Molecular Systems Biology* **5**: 248

You C, Okano H, Hui S, Zhang Z, Kim M, Gunderson CW, Wang YP, Lenz P, Yan D, Hwa T (2013) Coordination of bacterial proteome with metabolism by cyclic AMP signalling. *Nature* **500**: 301–306

Yu H, Paccanaro A, Trifonov V, Gerstein M (2006) Predicting interactions in protein networks by completing defective cliques. *Bioinformatics* **22**: 823–829

Zarrinpar A, Bhattacharyya RP, Nittler MP, Lim WA (2004) Sho1 and Pbs2 act as coscaffolds linking components in the yeast high osmolarity MAP kinase pathway. *Molecular Cell* **14**: 825–832

Zhang J, Diaz A, Mao L, Ahlquist P, Wang X (2012) Host acyl coenzyme A binding protein regulates replication complex assembly and activity of a positive-strand RNA virus. *Journal of Virology* **86**: 5110–5121

Zhang L, Villa NY, McFadden G (2009) Interplay between poxviruses and the cellular ubiquitin/ubiquitin-like pathways. *FEBS Letters* **583**: 607–614

Zhou H, Xu M, Huang Q, Gates AT, Zhang XD, Castle JC, Stec E, Ferrer M, Strulovici B, Hazuda DJ, Espeseth AS (2008) Genome-scale RNAi screen for host factors required for HIV replication. *Cell Host & Microbe* **4**: 495–504

Zitnik M, Zupan B (2014) Matrix factorization-based data fusion for gene function prediction in baker's yeast and slime mold. *Pacific Symposium on Biocomputing* : 400–411