

Explaining Effects of Host Gene Knockouts on Brome Mosaic Virus Replication

Deborah Chasman^{1,2}, Brandi Gancarz³, Paul Ahlquist³, Mark Craven^{2,1}

University of Wisconsin–Madison

¹Department of Computer Sciences

²Department of Biostatistics and Medical Informatics

³Institute for Molecular Virology

chasman@cs.wisc.edu, craven@biostat.wisc.edu

Abstract

Gene products are key players in the interaction networks within a cell. We analyze an experiment in which a yeast knockout library was assayed for the effects of host gene deletion on the replication of Brome Mosaic Virus (BMV). These observations, integrated with the partially known yeast interaction network, may be used to infer which host processes and gene products are involved in the mechanism of BMV's replication. We approach this task using Inductive and Abductive Logic Programming (ILP and ALP). We use ALP to abduce causal explanations for each observation, including possible host interfaces with BMV. Some notable aspects of our task that differ from previous work using abduction in systems biology include a highly incomplete background model and a large number of observations to explain. Additionally, we expect that there are many interfaces between the host cell and the virus, and that each abduced interface will serve to explain a handful of observations. We determine that ILP is unable to identify general, informative models that characterize host-virus interactions accurately. Using ALP, however, we are able to construct causal explanations that link multiple observations to the same host interface.

Introduction

A complex network of interactions within a cell determines its response to its environment. Many of the key players in these networks are gene products; i.e., proteins and RNAs that form important structures and catalyze reactions within the cell. We have only partial knowledge of these networks. However, in some cases we can shine a light into the “black box” by selectively turning off genes and observing the resulting change in the cell's response. There exists a library of strains of the yeast *Saccharomyces cerevisiae*, for example, in which each strain is defined by having a single gene deleted or modified to allow the gene's expression to be suppressed during an experiment [Winzeler et al., 1999]. Using these deletion strains, we can perform high-throughput assays by exposing each strain to the same conditions. These observations show us the ultimate effect of the loss of a gene on the

cell's response to the conditions of interest. By integrating these observations with a known partial network of intracellular gene interactions, we may be able to infer which cellular components and processes are involved in the response, and suggest a more complete model of the network.

Viral infection is one particular condition to which a cell might respond, and is one that is of particular relevance to human health. Brome Mosaic Virus (BMV) is a positive-strand RNA virus, a member of the same viral family as Hepatitis C and SARS. Understanding the BMV mechanism of replication may provide insight into the mechanisms of these high-profile, pathogenic viruses. The Ahlquist laboratory has artificially infected *Saccharomyces cerevisiae* knockouts with BMV [Kushner et al., 2003; Gancarz and Ahlquist, 2008]. By augmenting the virus with a luciferase reporter, it is possible to measure the amount of viral replication in an infected yeast colony. These assays have identified on the order of 100 host genes whose deletion or suppression significantly inhibits viral replication, and another 100 or so host genes whose knockouts encourage replication. We are interested in using computational methods to posit how these genes interact with one another, and with the virus, affecting its ability to replicate. Most genes are not directly involved in BMV's activity, but are instead part of some pathway that contains an actual interface with the virus. Our primary goal is to explain the causal chains that lead from deleted genes to potential interfaces with the virus.

The network of relationships among gene products and other molecules in *S. cerevisiae* is partially known, and is represented in various on-line databases [Christie et al., 2004; Stark et al., 2006; Pu et al., 2007]. For example, genes may encode proteins that form complexes with each other or otherwise physically interact. Or, they may catalyze reactions along the same pathway. We hypothesize that we can use these relationships to explain the BMV replication results in the yeast knockout assays. Because the background knowledge describing these networks of interactions has a rich relational structure, Inductive Logic Programming (ILP) and Abductive Logic Programming (ALP) represent compelling approaches to the problem.

As a preliminary investigation, we frame the problem as an inductive one. We wish to learn general rules for the concepts “gene deletions that significantly inhibit viral replication” and “gene deletions that significantly promote viral replication.”

Dividing our observations into positive and negative examples of these concepts, we use the ILP system Aleph [Srinivasan, 2007] to hypothesize clauses for these concepts in terms of our partial interaction network for *S. cerevisiae*.

The primary focus of our study, however, is on using Abductive Logic Programming to infer explanations to account for the BMV replication observations. There are several notable aspects of our task that distinguish it from previous work using abduction for systems biology applications:

- The available background knowledge for our task is highly incomplete and likely contains some false-positive assertions.
- Our abductive task involves constructing explanations for a large number of observations (the results of hundreds of knockout/suppression experiments).
- It is likely that a large number of abduced predicates are required in order to explain the observations. That is, the virus may have many interfaces to its host cell.
- Our approach forms explanations that account for multiple observations, with each explanation consisting of logical clauses that share the same abduced predicate.

In our ALP investigation, in contrast to our ILP experiment, we want to construct a specific, causal explanation for each BMV replication observation, rather than a set of general clauses. These explanations require a vital piece missing from our background information: the actual host interfaces to the virus. These interfaces may be particular gene products, protein complexes, or small molecules produced by metabolic reactions. While the ALP literature typically uses the word “explanations” to describe only the set of abduced facts, we consider an explanation to consist of the entire chain of literals describing the relationship between the knocked-out gene and the viral interface. Given that our background knowledge is so incomplete, we liberally hypothesize multiple explanations (and consequently multiple ground abducibles) for each observation. Additionally, we attempt to find explanations that serve to explain multiple observations.

Related Work

Several studies have applied ALP, ILP, or a hybrid approach to systems biology. Ong et al. (2007) apply ILP to predict gene expression regulation in yeast using time series data. Their model uses the yeast interaction network to learn clauses that cover genes showing similar expression patterns over time. Our task does not include this temporal element. Tamaddoni-Nezhad et al. (2006) use an Abductive ILP approach: they abduce the effects of a toxin on rat metabolic enzymes, and use ILP to learn general clauses covering the abduced facts. ALP is also applied in the Robot Scientist project, which contains an abductive component used to complete a model of yeast metabolic pathways [Reiser et al., 2001].

One way in which our task differs from previous work in abduction, and work integrating it with ILP, is that we do not assume the background network is complete except for the abducible predicate. Our background interactions are not

necessarily active under the same conditions, and they are certainly incomplete. Additionally, we expect that the host cell and the virus interact in many unique ways. Consequently, we search for many unique, specific explanations, each covering a small group of observations, rather than a small set of general clauses.

Data and Representation

We apply two algorithms to our problem: ILP, as implemented in Aleph [Srinivasan, 2007], and ALP, as implemented in ProLogICA [Ray and Kakas, 2006], with some additions. Both algorithms require background information and examples of a target relation, which we summarize below.

Data

For our examples, we use observations of gene knockout effect on viral replication, and assign each observation to a class. The measured values represent the fold change in luciferase expression (implying viral replication) in a mutant as compared to the wildtype. The observations from the Ahlquist laboratory’s BMV assays cover 4887 genes, 615 of which are essential genes [Gancarz and Ahlquist, 2008], and 4272 of which are nonessential genes [Kushner et al., 2003].

We process the measurements into one value for each assayed knockout. The nonessential gene data includes measurements from at most two successful trials; we use the average of the two as our value for that gene. The essential gene data includes up to four measurements for each assayed gene. Two separate trials were performed for each gene, with measurements taken at two time points within each trial. We average each time point over its two trials, and then take as our value the time point with the greatest magnitude. We then convert these measurements into fold changes. Finally, we assign discrete labels to each measurement based on the following two relations.

Target Relations

We consider two target relations: one classifying genes whose deletion significantly increases viral replication (`up`), and another classifying genes whose deletion significantly decreases viral replication (`down`). Previous work has defined thresholds for what fold changes are considered significant [Kushner et al., 2003; Gancarz and Ahlquist, 2008]. For essential gene knockdowns, a significant fold change has a magnitude greater than 6.0. For nonessential gene knockouts, a significant fold change has a magnitude greater than 3.0. We summarize the two relations as follows:

- **Down.** Positive examples of this concept are genes whose deletion or suppression *significantly* inhibits viral replication. Negative examples are observations of *any* positive fold change (fold change greater than 0). Sample encoding: `replication(ygl048c, down)`. This division results in 122 positive examples and 1762 negative examples.
- **Up.** Positive examples of this concept are genes whose deletion or suppression *significantly* increases viral replication. Negative examples are observations of *any* negative fold change. Sample encoding:

`replication(yp1011c, up)`. This division results in 88 positive examples and 2769 negative examples.

Note that both relations exclude a set of observations with uncertain classification: those with fold-changes between 0 and the positive or negative significance threshold.

Background Knowledge

To represent the known yeast network, we assemble logical representations of gene attributes and interactions in *S. cerevisiae*. These include genetic interactions, protein-protein interactions, post-translational modifications of transcription factors, GO annotations, metabolic pathways, protein complexes, and predicted protein complexes or functional units. Each dataset is encoded using either a binary or ternary relationship among atoms. Here is a brief summary of the relationships that make up our background information, along with a sample encoding.

- **Physical and genetic interactions.** These relationships from the BioGRID database [Stark et al., 2006] describe observed physical and genetic interactions between genes. Sample encoding: `physical(GeneA, GeneB)`, `genetic(GeneA, GeneB)`.
- **Genetic interactions from expression profiles.** These datasets from Rosetta Inpharmatics, Inc describe the quantitative effects of approximately 900 single gene knockouts on the expression levels of most other yeast genes [Hughes et al., 2000; Mnaimneh et al., 2004]. Hughes et al. include p -values calculated for each measurement. We considered a measurement to be significant if its p -value is less than or equal to .01. As Mnaimneh et al. do not include p -values for their measurements, we choose to keep those measurements with fold changes of magnitude two or greater from wildtype expression. Sample encoding: `upRegulates(GeneA, GeneB)` means that GeneA is necessary for transcription of GeneB. We observe a decrease in GeneB expression when GeneA is suppressed or deleted.
- **Post-translational modifications of transcription factors.** Transcription factors are proteins involved in the regulation of gene expression. Sometimes a transcription factor requires modification by another protein in order to activate it. The loss of the gene encoding either the transcription factor or the modifier may result in a different level of expression of the target gene, and may indirectly influence the interface with the virus. We use triplets from the PTM-Switchboard project [Everett et al., 2008]. Sample encoding: `tf_ptm(Modifier, TranscriptionFactor, TargetGene)`.
- **Metabolic pathways and protein complexes.** If a gene product is involved in a metabolic pathway, its deletion may influence activity downstream. Herrgård et al. (2008) have aggregated known pathways from multiple datasets into one consensus model. We include these pathway relationships between genes and metabolites in order to suggest a unified explanation for groups of genes whose knockouts result in similar effects and

which influence the same pathway. Sample encoding: `pathForward(A, B)`. A and B may be genes, molecules, or protein complexes. For example, A may be a gene catalyzing the reaction that produces molecule B. The pathways are not linear; there may be multiple Bs one step forward from any given A.

- **Protein complexes.** It is possible that the cell's interface with the virus is a protein complex. If this is the case, the knockout of a gene integral to the assemblage or functioning of the complex should inhibit viral replication. We include a collection of manually curated, literature-supported protein complexes from the CYC2008 project [Pu et al., 2008]. Sample encoding: `inComplex(GeneA, ComplexA)`.
- **GO annotations.** The Gene Ontology (GO) is a system for annotating genes with terms that describe known attributes of the genes [Ashburner et al., 2000]. The terms cover three categories: Cellular Component, Molecular Function, and Biological Process. Sample encoding: `go(GeneA, GO:1234)`.

Recent studies have predicted clusters of genes which may represent functional units. Many of these correspond to complexes previously reported in literature.

- **Predicted protein complexes.** Pu et al. (2007) have predicted a clustering based on high-throughput data of protein-protein interactions. Many of their clusters contain one or more previously reported complexes, while others predict complexes. Sample encoding: `inComplex(GeneA, YHPT_X)` means that GeneA is a member of cluster YHPT_X.
- **Modules and Complementing Module Pairs.** Ulitsky et al. (2008) present a method to cluster genes based on genetic interactions, endeavoring to predict functional units. Many of their resulting clusters, called modules, correspond to previously reported complexes. They also present pairs of modules, Complementing Module Pairs (CMPs), which may represent pairs of functionally redundant units. Sample encoding: `inModule(GeneA, ModuleA)`, `cmp(ModuleA, ModuleB)`.

Hypothesizing Clauses using ILP

The first question we consider is whether our experimental observations of BMV replication can be explained by general rules induced using a learning approach such as ILP. That is, we want to assess the ability of ILP to learn meaningful clauses that characterize the up and down classes in terms of the relationships represented in our assembled background information.

We use the Aleph ILP system [Srinivasan, 2007] for the experiments reported in this section. Because the proportion of positive to negative examples is greatly skewed, we define a cost function for Aleph in which a positive example covered by a clause has twice as much weight as a negative example. We restrict Aleph to suggesting clauses covering at least three positive examples, and search for clauses up to length four.

To evaluate the ability of the learning algorithm to induce descriptions that capture meaningful generalizations of the

Table 1: Results from the ILP experiment. Shown are precision (P), recall (R), accuracy (Acc.), and F_1 -measure for the target relations, based on the results of twenty-fold cross-validation. We also show the p -value for F_1 calculated from the results of the permutation test.

Relation	P	R	Acc.	F_1	p -value
up	0.040	0.441	0.670	0.073	0.05
down	0.080	0.541	0.557	0.137	0.02

positive examples, we employ a twenty-fold cross-validation methodology. In this procedure, we run the ILP algorithm twenty times, each time leaving out $\frac{1}{20}$ of the examples for a test set. We evaluate the predictive accuracy of our learned models by measuring precision, recall, and F_1 (the harmonic mean of precision and recall).

To assess whether the learned clauses represent specifically how the host genes influence viral replication, as opposed to simply characterizing groups of genes that are related in some way, we use a permutation testing methodology. Permutation testing here involves comparing the learning algorithm’s predictive accuracy on the given data to its accuracy on random permutations of the observation labels. For both observation classes up and down, we randomly partition the data into positive and negative sets 100 times, keeping the class sizes in the same proportion as in the original data. We perform twenty-fold cross validation on each partition to acquire an F_1 score for that partition. The 100 resulting F_1 scores represent our null distribution.

Results

Table 1 summarizes the performance of the ILP algorithm on the test sets in terms of precision, recall, accuracy, and F_1 . The rightmost column in the table shows the p -value for F_1 as determined by the permutation test. With respect to the F_1 -measure, we can reject the null hypothesis at the level of $p \leq 0.05$ for both the up and down relations. This result suggests that the learned Aleph models are capturing some meaningful information about how the host genes interact with the virus. However, the low precision of the learned clauses indicates that the ILP approach is not able to characterize the host-virus interactions with high accuracy.

In light of the low precision of models induced by Aleph in these experiments, it may be more informative to examine individual clauses produced by Aleph than to consider the set of clauses as a whole. Most clauses learned for both target concepts contain only literals from the Gene Ontology. These clauses do not give us any particular explanatory advantage; a tool such as the GO::Termfinder [Boyle et al., 2004] would be better used to find enriched GO terms among our positive examples, as it also assesses the statistical significance of shared terms. Other clauses identify entire protein complexes represented by the positive examples for up or down. Some of these complexes may provide insight into the mechanism of the virus.

In summary, the results of this experiment suggest that it may not be fruitful to use a standard inductive approach to explain the viral replication observations. We conjecture that

Table 2: The ALP task.

- **Given:**
 - A set of observations, e^+ , from the positive set of down
 - A logical encoding of the known partial network, B
 - An abducible predicate representing what is missing from the background information, `interface(x)`
 - A set of clauses for traversing B from an observation in e^+ to the abducible predicate
- **Do:**
 - Construct explanations for observations from e^+ , using terms from B , and ending with grounded hypotheses for `interface(x)`

Table 3: Example explanation produced by ALP.

```
replication(efb1, down) :-
  inComplex(efb1, cyc_121),
  interface(cyc_121).
```

the ILP approach does not produce high-accuracy models due to (i) the degree of incompleteness in the background knowledge, and (ii) the likelihood that there are many distinct interfaces between the virus and the host cell.

Hypothesizing Explanations using ALP

The goal of our second investigation is to determine if we can account for multiple down observations using a single abducted predicate. For this investigation, we focus on the relation down, because it is clinically relevant. The predicate that represents the missing piece in our understanding of the relationship between a knockout gene and viral replication is `interface(x)`, the actual host interface with the virus and the final step in an explanation. This interface may be a gene, a protein complex, or a molecule produced during a metabolic reaction. Table 2 describes the task.

An explanation takes the form of chain of relationships leading from a gene knockout to a viral interface. The example in Table 3 shows an explanation generated for the observation that viral replication is inhibited in the EFB1 knockout. In this case, the explanation is that the gene encodes a protein that is in the Complex 121 from the Cyc2008 database, and that complex is the viral interface. In EFB1-knockouts, the production of complex 121 may be prevented, thus suppressing the replication of the virus.

Background Knowledge and Model

As we would like to construct a specific, causal story for each observation, we limit our background information to terms that describe causal relationships between genes, complexes, and metabolites. We include the following in our background relations: protein complex membership, physical interactions, genetic interactions from expression profiles,

post-translational modifications of transcription factors, and metabolic pathway steps. We exclude genetic interactions for which we do not know the direction of the relationship. Similarly, we exclude GO annotations, which do not necessarily capture direct relationships between genes. Additionally, much of relevant information from the Cellular Component subontology should be redundant with the protein complex data.

We supply a simple logical model of the potential interactions between a gene and an interface. These clauses dictate how interactions from the background data may chain together to form explanations. Within these clauses, we enforce consistent behavior among genes in an explanation. For example, if we are tracing the path from a gene to the interface, no gene that appears along the path (including the final interface, if it is a gene) should belong to the set of up genes. Additionally, with respect to genetic interactions from expression profiles, we only allow those in which one knock-out results in a decrease in the expression of another gene. The clauses are presented in Table 4.

Methods

We use ProLogICA [Ray and Kakas, 2006] to perform abduction, generating all possible explanations for each observation. We have added a slight modification to the source code so that ProLogICA outputs the grounded intermediate goals satisfied in the process of abducing facts. We then process this output into a set of coherent clauses in the form of the one shown in Table 3.

We organize the explanations based on their shared components, or tails, using a process related to the *A Priori* algorithm for association rules [Agrawal and Srikant, 1994]. The *support* of a shared tail is the number of distinct observations found in explanations sharing that tail. We refer to a set of explanations sharing a tail as an *explanation group*. An example explanation group is shown in Table 5. The output of the ALP process, then, is a set of these explanation groups. We assess the power of this approach to explain multiple genes under the same explanation. Again, we use permutation tests to determine whether we cover significantly more genes under high-coverage explanation groups as we can with explanation groups learned on randomly labelled data. We score a learned set of explanation groups (all groups generated from a set of observations) with its gene coverage. Here, we define *coverage* as the number of genes that are covered by an explanation group that covers at least a minimum number of total genes. That minimum number we call *support*.

For each of the 100 permutation tests, we randomly draw 88 genes from our observation pool to label as up, and 122 genes to label as down. For these 100 partitions, we run the ALP process as described above to acquire a set of explanation groups. We score the set of explanation groups for several values of minimum support.

Results

At a maximum clause length of five, ProLogICA constructs 19,154 explanations for the 122 positive examples of down. 85 observations generate more than one explanation. (All observations generate at least a trivial explanation - that the

Table 4: Background model used in ALP experiments.

The observed gene itself may be an interface, or the gene may be directly related to an interface.

```

replication(G, down) :-
    interface(G).
direct(A,B) :-
    tf_ptm(A,_,B), A\==B,
    not replication(B, up).
direct(A,B) :-
    tf_ptm(_,A,B), A\==B,
    not replication(B, up).
direct(A,B) :-
    upRegulates(A, B, down), A\==B,
    not replication(B, up).
direct(A,B) :-
    physical(A,B), A=B,
    not replication(B, up).
direct(A,B) :-
    inModule(A,B).
direct(A,B) :-
    inComplex(A,B).

```

To chain multiple entities in an explanation:

```

replication(G,down) :-
    connect(G,A), interface(A).
connect(A,B) :-
    direct(A,B).
connect(A,B) :-
    not direct(A,B), direct(A,C),
    connect(C,B).

```

A special case. Pathway relationships are only causal within the context of a metabolic pathway. Once our explanation enters a pathway, all downstream entities in the explanation must be steps forward along that pathway.

```

connect(A,B) :-
    connectPathOnly(A,B).
connectPathOnly(A,B) :-
    directPath(A,B).
connectPathOnly(A,B) :-
    not directPath(A,B), directPath(A,C),
    connectPathOnly(C,B).
directPath(A,B) :-
    pathForward(A,B), A\==B,
    not replication(B, up).

```

Table 5: An explanation group sharing the tail `interface(ole1)`, which covers five observations.

```

replication(pre1,down):-
  physical(pre1,rpt4),
  upRegulates(rpt4,ole1,down), interface(ole1).
replication(rpt6,down):-
  physical(rpt6,rpn8),
  upRegulates(rpn8,ole1,down), interface(ole1).
replication(rpt6,down):-
  physical(rpt6,rpt4),
  upRegulates(rpt4,ole1,down), interface(ole1).
replication(rpt6,down):-
  physical(rpt6,rpt2),
  upRegulates(rpt2,ole1,down), interface(ole1).
replication(rpt6,down):-
  physical(rpt6,erv25), physical(erv25,ole1),
  interface(ole1).
replication(ubp6,down):-
  physical(ubp6,rpt4),
  upRegulates(rpt4,ole1,down), interface(ole1).
replication(ufd4,down):-
  physical(ufd4,rpt4),
  upRegulates(rpt4,ole1,down), interface(ole1).
replication(yip5,down):-
  physical(yip5,rsp5),
  tf_ptm(rsp5,spt23,ole1), interface(ole1).

```

Table 6: Results from the ALP experiment and permutation tests. For each row, explanations groups covering at least a minimum number of genes are considered; this number is in the Minimum Support column. The Coverage column displays the coverage of the set of explanation groups produced using the actual data. We calculate the p -value to be the number of random partitions with a coverage at or above that of the actual data. For minimum support of 2-13, no random partition scored higher than the actual data.

Minimum Support	Coverage	p -value
2	106	< 0.01
3	83	< 0.01
4	74	< 0.01
5	67	< 0.01
6	62	< 0.01
7	46	< 0.01
8	39	< 0.01
9	39	< 0.01
10	36	< 0.01
11	33	< 0.01
12	31	< 0.01
13	31	< 0.01
14	22	0.01

knockout gene is the interface.) With 10,769 explanations, the observation for the gene YGL048C/RPT6 generates by far the most. This gene regulates the expression of many other genes, and it has high degree in the background network. The highest number of observations sharing one tail is 14. With respect to explanation groups, we assemble 5,924 groups covering at least two genes; this collection covers 106 genes out of 122. The coverage of explanation groups at other levels of minimum support, as well as the p -values for the permutation tests, are shown in Table 6. For all values of minimum support from 2-14, $p \leq 0.05$. The number of genes we can cover using high-coverage explanation groups is significant at the 95% confidence level. This suggests that our model is capturing information about the host-virus interactions, rather than other, irrelevant interactions between genes.

Example Explanation Groups

We partially recover a subnetwork of interactions previously identified by the Ahlquist group. This subnetwork indicates that OLE1’s involvement in lipid metabolism is important to BMV replication. The underlying explanation, as described by Gancarz and Ahlquist (2008), is as follows: RSP5 tags transcription factor SPT23 for activation by the proteasome. The proteasome activates SPT23, allowing SPT23 to enter the nucleus and activate the transcription of OLE1.

Our OLE1 explanation group tells part of the same story. Table 5 shows the explanation group sharing the tail `interface(ole1)`, while Figure 1 depicts the group graphically. RSP5 modifies SPT23, which activates transcription of OLE1. Several genes in the proteasome (RPT4, RPT2, RPN8) up-regulate OLE1. While we do not have observations for the effects of these genes on viral replication, they physically interact with other genes in the proteasome (RPT6, PRE1, UBP6, UFD4) that are examples of the down relation. Additionally, the explanation suggests that the decrease in viral replication in the YIP5 knockout may be related to its physical interaction with RSP5. It also suggests another path from the proteasome to OLE1: RPT6 has a physical interaction with ERV25, a gene with inconclusive inhibitory effect on the virus. ERV25 in turn interacts with OLE1. It is worth noting that these explanations highlight the incompleteness of our background network. Genetic interactions from expression profiles relate the proteasome to OLE1 in one step, rather than through SPT23.

As noted previously, some clauses produced by Aleph during our ILP experiment suggest complexes or genes to further investigate. However, our ALP approach may produce richer explanations than ILP. We consider, for example, the YHPT_3 complex, which contains 40 genes and is involved in transcription from the promoters of RNA polymerases I, II, and III. While the YHPT_3 complex appears in both an ILP clause and in an ALP explanation group, the latter provides a larger context.

Figure 2 provides a graphical representation of an ALP explanation group for the abducible `interface(yhpt_3)`. This group was learned under a maximum clause length of four. ILP learned the clause `replication(A,down):-inComplex(A,yhpt_3)`, which covers the knockouts for RPB3, RPA34, RPC19, SPT4, RPC19, and DST1. While

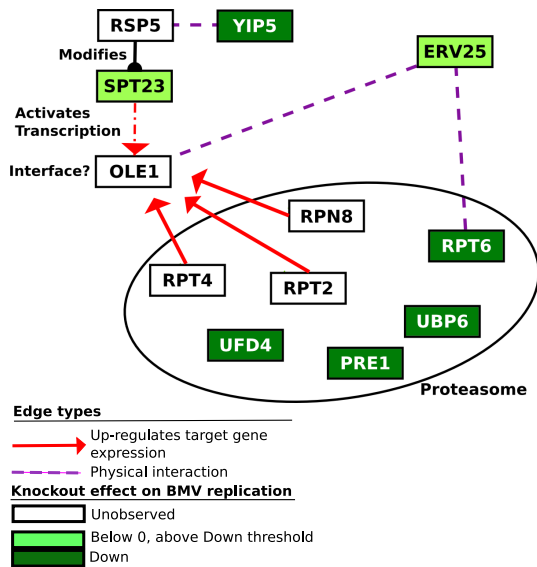


Figure 1: Graphical representation of the explanations with the tail interface (ole1). Five observations (YIP5, RPT6, UBP6, PRE1, UFD4) are covered by this interface. The genes involved in the proteasome complex are outlined by an ellipse. For clarity, we do not depict all of the physical interactions among the constituents of the proteasome. It should be noted that the proteasome comprises more genes than are pictured here; this image only contains those genes that appear in the explanations produced by ALP.

both representations cover the same complex, the explanation group also indicates that the knockouts for RPT6 and DOP1 may inhibit viral replication by inhibiting the expression of the genes RPB5, RPB10, RPA43, and RPA34 in the YHPT_3 complex.

Discussion

In summary, we investigated the application of ILP and ALP to our observations of yeast knockout effect on BMV replication. Our inductive approach was unable to learn general models to characterize virus-host factor interactions with high precision. Using ALP, we abduced explanation groups linking multiple host genes to the same interface with the virus. Based on the results of permutation tests, it appears these groups capture information about virus-host interactions. We were also able to recover an explanation group previously identified by the Ahlquist group.

Our investigation thus far has suggested many ideas for future work, both in application and algorithm development. As was highlighted by the OLE1 situation, it would be worthwhile to supplement our current background information with other types of intracellular relationships. For example, we may integrate additional transcription factors and post-translational modifications. It will be necessary to further develop the ALP algorithm, improving the method for ordering and grouping explanations. We also plan to investigate the possibility of integrating other data sources into the scoring of possible explanations: for example, genetic interaction

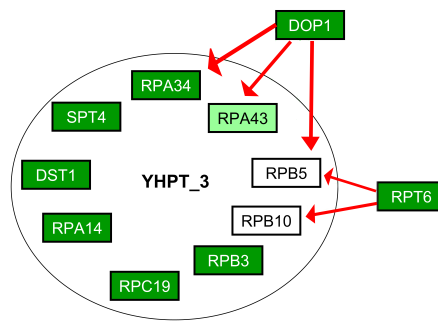


Figure 2: Graphical representation of explanations for eight down genes (RBP3, RPA34, RPC19, SPT4, RPB3, DST1, DOP1, RPT6) all sharing the tail interface (yhpt_3). YHPT_3 contains 40 genes; only genes involved in the ALP explanation group learned with a maximum clause length of four are depicted here.

data and the quantitative measurement of a knockout's effect on viral replication. Much current research focuses on the integration of ALP and ILP. It is possible that by running ILP again on our background knowledge, plus the abduced host-virus interfaces, we may hypothesize additional clauses relating a knockout gene to a proposed interface. These clauses may be used supplement the model used in ALP. Lastly, we hope in the future to investigate our explanation groups with wet-lab experiments. We envision an exchange between the computation of hypotheses and the experimental investigation thereof; however, we currently have no official plans.

Acknowledgements

This work is supported by NIH/NLM grants T15-LM007359 and R01-LM07050.

References

- [Agrawal and Srikant, 1994] Agrawal, R., and Srikant, R. 1994. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc., 487–499.
- [Ashburner et al., 2000] Ashburner, M., et al. 2000. Gene Ontology: tool for the unification of biology. *Nature Genetics* 25(1):25–29.
- [Boyle et al., 2004] Boyle, E. I., et al. 2004. GO::TermFinder—open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics* 20(18):3710–3715.
- [Christie et al., 2004] Christie, K., et al. 2004. Saccharomyces genome database (SGD) provides tools to identify and analyze sequences from *Saccharomyces cerevisiae* and related sequences from other organisms. *Nucleic Acids Research* 32:D311–D314.
- [Everett et al., 2008] Everett, L., et al. 2008. PTM-Switchboard—a database of posttranslational modifications of transcription factors, the mediating enzymes and target genes. *Nucleic Acids Research* 37:D66–D71.
- [Gancarz and Ahlquist, 2008] Gancarz, B., and Ahlquist, P. 2008. Systematic identification of essential host genes affecting Brome Mosaic Virus RNA replication and gene expression. Poster.

- [Herrgård et al., 2008] Herrgård, M. J., et al. 2008. A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. *Nature Biotechnology* 26(10):1155–1160.
- [Hughes et al., 2000] Hughes, T. R., et al. 2000. Functional discovery via a compendium of expression profiles. *Cell* 102(1):109–126.
- [Kushner et al., 2003] Kushner, D. B.; Lindenbach, B. D.; Grdzelishvili, V. Z.; Noueiry, A. O.; Paul, S. M.; and 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 of the United States of America* 100(26):15764–15769.
- [Mnaimneh et al., 2004] Mnaimneh, S., et al. 2004. Exploration of essential gene functions via titratable promoter alleles. *Cell* 118(1):31–44.
- [Ong et al., 2007] Ong, I. M.; Topper, S. E.; Page, D.; and 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*, Springer Lecture Notes in Artificial Intelligence. 4455:366–378.
- [Pu et al., 2007] Pu, S., et al. 2007. Identifying functional modules in the physical interactome of *Saccharomyces cerevisiae*. *Proteomics* 7(6):944–960.
- [Pu et al., 2008] Pu, S., et al. 2009. Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Research* 37(3):825–31.
- [Ray and Kakas, 2006] Ray, O., and Kakas, A. 2006. Prologica: a practical system for abductive logic programming. In *Proceedings of the 11th International Workshop on Non-monotonic Reasoning*.
- [Reiser et al., 2001] Reiser, P. G. K.; King, R. D.; Kell, D. B.; Muggleton, S. H.; Bryant, C. H.; and Oliver, S. 2001. Developing a logical model of yeast metabolism. *Electronic Transactions Articles in Artificial Intelligence* 6.
- [Srinivasan, 2007] Srinivasan, A. 2007. *The Aleph Manual*. <http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/aleph.html>
- [Stark et al., 2006] Stark, C., et al. 2006. BioGRID: a general repository for interaction datasets. *Nucleic Acids Research* 34:D535–D539.
- [Tamaddoni-Nezhad et al., 2006] Tamaddoni-Nezhad, A.; Chaleil, R.; Kakas, A.; and Muggleton, S. 2006. Application of abductive ILP to learning metabolic network inhibition from temporal data. *Machine Learning* 64(1-3):209–230.
- [Ulitsky et al., 2008] Ulitsky, I., et al. 2008. From E-MAPS to module maps: dissecting quantitative genetic interactions using physical interactions. *Molecular Systems Biology* 4:209–221.
- [Winzeler et al., 1999] Winzeler, E., et al. 1999. Functional characterization of the *Saccharomyces cerevisiae* genome by gene deletion and parallel analysis. *Science* 285:901–906.