SSLPRED : PREDICTING SYNTHETIC SICKNESS LETHALITY*

NIRMALYA BANDYOPADHYAY†, SANJAY RANKA, AND TAMER KAHVECI

CISE Department, University of Florida,
Gainesville, FL 32611, USA
E-mail: {nirmalya†, ranka, tamer}@cise.ufl.edu
www.cise.ufl.edu

Two genes in an organism have a *Synthetic Sickness Lethality* (SSL) interaction, if their joint deletion leads to a lower than expected fitness. Synthetic Gene Array (SGA) is a technique that helps in identifying SSL values for pairs of genes in a given set of genes. SSL interactions are useful to discover the co-expressed gene groups in the regulatory and signaling networks. Also, they are used to unravel the pair of pathways (subset of physically interacting genes) that substitute the functions of each other. Generating an SGA entry is costly as it requires producing and monitoring a *double mutant* (a progeny with two mutated genes). Generating a comprehensive SGA can be very expensive as the number of gene pairs is quadratic in the number of genes of the corresponding organism.

In this paper, we develop a new method *SSLPred* to predict the SSL interactions in an organism. Our method is built on the concept of *Between Pathway Models (BPM)*, where majority of the SSL pairs span across the two functionally complementing pathways. We develop a regression based approach that learns the mapping between the gene expressions of single deletion mutant to the corresponding SGA entries. We compare our method to the one by Hescott et al. for predicting the GI (Genetic Interaction) score of *Saccharomyces cerevisiae* (*S. cerevisiae*) on four benchmark datasets. On different experimental setups, on average SSLPred performs significantly better compared to the other method.

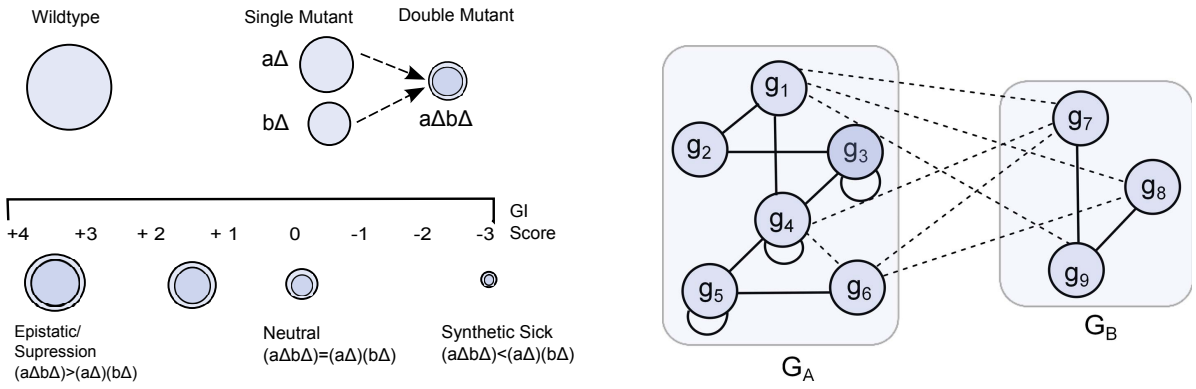
Keywords: Synthetic Sickness Lethality, Regression, Essential Genes

1. Introduction

Analysis of gene essentiality is a crucial problem to understand the roles of different genes at the molecular and genetic levels. A gene is defined *essential* if it is required for proper growth and sustenance of that organism. Essential genes have been thoroughly investigated using techniques such as single gene deletion screening for some low level organisms such as *Escherichia coli (E. coli)*.[1] Though identification of essential genes enlightens us about the functions of individual genes in an organism, it provides little conclusive information about the nature of their genetic relationships in gene regulatory and signaling networks. Recently, studies on *Synthetic Sickness Lethality (SSL)* opened up new directions in the areas of functional genomics. Two *non-essential* genes follow an SSL interaction if their joint deletion leads to a less than expected *fitness* for the organism. Here *fitness* denotes the growth and sustenance rate of an organism. An expected fitness corresponds to that of a double mutant when the two knocked out genes are not in an SSL interaction. Note that the fitness of an organism due to an SSL interaction can be less than (aggravating) or more than (alleviating) the expected fitness.[2] A genome wise catalog of SSL interactions enables in-depth molecular analysis, by creating a functional map of the cell, predicting functions and relations of the genes and deciphering complex regulatory relations from the global genetic network.[3]

The Synthetic Genetic Array (SGA)[4] and diploid-based synthetic lethality analysis on microarray (dSLAM)[5] are two pioneering approaches that enable systematic identification of SSL

*This work was supported partially by NSF under grants CCF-0829867 and IIS-0845439.



(a) Synthetic Sickness Lethality (b) Between Pathway Models

Fig. 1. Figure 1(a) illustrates the concepts of synthetic sickness and lethality. A double mutant produced from the cross of two single mutants can have a specific fitness in a range of GI scores based of the relationship between the two genes. The single and double circles represent single and double mutants respectively. Here, the size of a circle corresponds to the observed fitness of the corresponding mutant. Based on whether the two genes have a epistatic, neutral of SSL interaction, the observed fitness of the double mutant can have more than, equal to or less than the expected fitness. In those cases, the GI score can be a significantly large positive, close to zero or significantly large negative number, respectively. Figure 1(b) depicts the concept of Between Pathway Models (BPM). The hypothetical BPM consists of two sub-networks (also called pathways) G_A and G_B who are functionally independent and complementing. The solid lines denote physical interactions, while the dashed directed lines stand for the SSL interactions. It is evident that the number of SSL edges between G_A and G_B is higher compared to the ones within the two groups.

interactions. Both methods require generation of double mutant strains and monitoring their growth. Each entry of SGA is a triple that consists of two genes and a GI (genetic interaction) score for those two genes. A score close to zero indicates that there is no SSL interaction between the two genes. For a gene-pair, a negative GI with a large magnitude indicates an aggravating SSL interaction. A significant large positive number denotes a higher chance of being alleviating relationship.[2] Figure 1(a) illustrates the concepts of synthetic sickness and lethality relationship. The EMAP strategy exploits the SGA technique by enabling colony sizes to be measured in an array format, thus quantifying genetic interactions in a high-throughput fashion.

Both SGA and dSLAM are costly techniques as for a pair of genes they require creation of two single mutant strains and crossing between them to produce a double mutant strain. For an organism with N genes, we need to generate and monitor the growth of $\frac{N(N-1)}{2}$ different double mutants. As a result, millions of double mutants need to be produced to tabulate all the genetic interaction scores for an organism that consists of thousands of genes. Creating such double mutants in wet-lab is an expensive and time consuming process. Therefore, we need an efficient method to predict whether there exists a synthetic lethality relation between two genes. Briefly, we can describe the problem considered in this paper as follows: *given two genes g_A and g_B what is the GI score between them?*

In order to predict the GI score between two genes, we incorporate the *genetic profile* of single mutant strains. This is a promising strategy for the number of single mutants can not be more than the number of genes. First, we elaborate on the term genetic profile. Consider single gene knockout dataset (also termed as single gene mutant data). Here, in each experiment a non-essential gene is knocked out from an organism. For each gene, expressions are obtained

before and after the knock-out and ratio of this after to before expression is calculated. Finally, the logarithm of that ratio is computed and tabulated. If the magnitude of a logarithm is large, it indicates that the expression of the corresponding gene changed significantly after the knockout of the non-essential gene under consideration. The genetic profile for a single mutant or a single gene knockout experiment consists of entries for all genes computed in the way described above.

In this paper our objective is to learn the GI scores of gene pairs with the help of *genetic profile* of single mutants. Formally we solve the following problem.

Problem *Let $\mathcal{V} = \{g_1, g_2, \cdots, g_M\}$ denote the set of genes in an organism. Assume that we are given the genetic profiles of K single mutant genes. $\mathcal{X}$ is a $K \times M$ matrix, where each row corresponds to the genetic profile of a single mutant. Let us represent the GI score of gene pairs g_a and g_b with $t_{a,b}$. Let $\mathcal{T}$ denote the set of all the available GI scores for that organism. For any gene pair (g_i, g_j) such that $g_i \in \mathcal{V}, g_j \in \mathcal{V}$, we would like to predict the GI score.* □

Before discussing our contribution in this paper, we summarize the Between Pathway Models (BPM), which is a building block of our model.[6] A BPM consists of two gene subnetworks (also called pathways) G_A and G_B, such that there are few SSL interactions within G_A and within G_B, but many of those between G_A and G_B. The opposite holds for the physical interaction edges. That is, many physical interactions exist within G_A and G_B, but few of them exist between G_A and G_B. Figure 1(b) depicts a hypothetical BPM. According to Kelley and Ideker, the two pathways in a BPM are functionally compensating due to the orientation of genetic and physical edges.[6] Now that we have introduced all the relevant building blocks, we discuss our contribution in this paper.

Contribution In this paper, we develop a new method *SSLPred* to predict the GI scores. To our knowledge, our method is the first one to predict GI scores using a mathematical machine learning based technique.

In accordance with the concept of BPM, we propose the following conjecture. If there is an SSL interaction between two genes and if these two genes belong to two pathways of a BPM, then knocking out one of them will change the expressions of most of the genes in both of the pathways in that BPM. The pathway containing the mutated gene is directly affected and dysfunctional as most of the consisting genes have a direct connection with the mutated gene through physical edges. The other pathway compensates for its affected pair, and due to the additional activities the genes in it change their expressions noticeably.

In our regression based method SSLPred, we develop a mapping between the genetic profiles of single mutants and the corresponding GI score. For every genetic interaction entry $(g_a, g_b, t_{a,b})$, such that either of g_a and g_b has been mutated in a single mutant gene experiment and $t_{a,b}$ is the GI score for g_a and g_b, we create a training sample. As we have already conjectured in the previous paragraph, if this genetic interaction entry represents an SSL, the mutated gene affects the expressions of all the genes in the corresponding BPM. Thus, we use the gene expression changes only from the pathways of that BPM to extract the features of the training point and correlate it with the corresponding GI score $t_{a,b}$ using a regression model. After we estimate the parameters of SSLPred, we are able to predict the GI score for a new pair of genes.

We compare our method to the one by Hescott et al.[7] in their ability to identify BPMs in the gene networks of *S. cerevisiae* on four benchmark datasets. On average SSLPred performs significantly better compared to the other method. We summarize our contribution as follows:

(1) According to our knowledge, SSLPred is the first *predictive method* to predict the GI score for a pair of genes. All other relevant computational methods are *descriptive*.
(2) The GI scores predicted by SSLPred assume a real value. This is more useful than a binary prediction, since it enables to conduct statistical analysis such as permutation tests and p-Value generation associated with the validation of benchmark BPMs.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 describes our method SSLPred. Section 4 presents the experimental results. Finally, Section 5 concludes the paper.

2. Background

Recent studies on synthetic sickness and lethality analysis opened up new directions in the areas of functional genomics. These works can be classified into two categories, namely experimental and computational. Experimental methods include Synthetic Gene Array (SGA),[4] dSLAM (diploid-based synthetic lethal analysis with microarrays)[5] and EMAP (epistatic miniarray profile).[8] We describe them in detail in Section 1. We summarize the computation methods next.

Kelly and Ideker introduced Between Pathway Models (BPM) by combining synthetic sickness and lethality informations from EMAP data with information on protein-protein, protein-DNA or metabolic networks.[6] Hescott et al.[7] proposed a new method to validate BPMs using single gene deletion microarray data. They evaluated the quality of the BPMs from four different studies and described how their methods might be extended to refine BPM pathways. Kelley and Kingsford developed a new method called Expected Graph Compression to identify compensatory pathways (BPMs) by clustering genes into modules and establishing relationships between those modules.[9]

Though researches in these two avenues enriched our understanding of gene interactions and gene networks, we did not find any predictive model to predict the GI scores. In this paper, we proposed a new regression based method SSLPred to predict GI scores.

3. Methods

In this section, we discuss our method in detail. Section 3.1 describes the notation and formulates the problem. Section 3.2 explains our conjectures which guide our model and the rationale behind it. Section 3.3 discusses the feature extraction and regression model.

3.1. *Problem Formulation and Notation*

In this section, we mathematically formulate the problem, and for that purpose, we describe the relevant notation. We group our notation in three classes based on three related entities. These are gene network, single mutant data and SGA. Here, gene network stands for gene interaction network, specifically the union of gene regulatory and signaling network.

(1) Gene network. The gene network is a union of gene regulatory and signaling networks that can be modeled as a set of genes and the directed edges (i.e., interactions) connecting these genes. Here, an edge between two genes denotes different kinds of genetic interactions such as activation, inhibition and phosphorylation. Let us denote the set of all M genes by $\mathcal{V} = \{g_1, g_2, \cdots, g_M\}$. We denote the set of all edges in the gene network by $\mathcal{W} = \{(g_i, g_j) | g_i \in \mathcal{V}, g_j \in \mathcal{V}\}$, where $(g_i\ ,\ g_j)$ implies a directed interaction from g_i to g_j. thus, $G = (\mathcal{V}, \mathcal{W})$ defines

the gene network.

(2) Single mutant dataset. In a single mutant, one gene is mutated in an organism and gene expression is obtained before and after the mutation. *Single deletion mutant* (also known as *single gene knockout*) is an important kind of gene mutant, where one gene is knocked out from an organism. In a single mutant dataset, each entry contains the logarithm of the ratio of the expressions of a gene after the gene knockout to that of the same gene before the gene knockout.[10]

Let $e'_{h,j}$ and $e_{h,j}$ denote the expressions of the gene g_j after and before the mutation of g_h respectively. We define the genetic profile of the organism when gene g_h is mutated by $X_h = \{x_{h,j} | x_{h,j} = ln(e'_{h,j}/e_{h,j}), j \in \{1, 2, \cdots, M\}\}$. Let $\mathcal{H} \subseteq \mathcal{G}$ be the set of genes that have been mutated in total. We define the single mutant genetic profile of N genes as $\mathcal{X} = \{X_h | g_h \in \mathcal{H}\}$.

(3) Synthetic Gene Array. An SGA is a set of triples, $\mathcal{T} = \{(g_i, g_j, t_{i,j}) | i, j \in \{1, 2, \cdots, M\}, i \leq j\}$, where $t_{i,j}$ is a real number that corresponds to the ratio of the observed fitness to the expected fitness when the organism has both gene g_i and g_j knocked out. A value with a large magnitude implies a potential SSL edge. A positive and a negative value stand for alleviating and aggravating relations, respectively.

Problem Formulation *Given a gene network $\mathcal{G}$, the single mutant dataset $\mathcal{X}$ and the SGA dataset $\mathcal{T}$, find the mapping $\Upsilon : \mathcal{X}, \mathcal{G} \longrightarrow \mathcal{T}$ which minimizes a predetermined risk function.*

Risk function is a measure of expected miss prediction rate. In this paper, while estimating the mapping function Υ, we minimize least square error in order to minimize expected miss prediction rate. Based on the mapping learned, we would predict the GI score $t_{i,j}$ for a new double mutant whose two genes g_i and g_j haven been mutated.

3.2. *Between Pathway Conjectures*

In this section, we describe our two conjectures that are central to SSLPred and the rationale for them. These two conjectures are built on the concepts of BPMs. Incorporating the structure and properties of BPMs into our model to improve its prediction accuracy was the motivation behind these conjecture.

Conjecture 1. Let $\mathcal{B}$ denote a BPM, consisting of two pathways G_A and G_B. Also, consider an SSL edge $\mathcal{S} = \{g_a, g_b\}$ such that $g_a \in G_A$ and $g_b \in G_B$. Then, mutating g_a will significantly alter the expressions of many genes in G_A and G_B. □

Since, g_a is connected to most other genes in G_A through physical interactions, altering the expression level of g_a will affect the expression of all the genes connected to g_a as they regulate each other. This effect will propagate through the gene network and eventually may change the expressions of many genes in G_A. Eventually g_a will severely affect G_A and prohibit it from working properly. Since G_A and G_B constitute a BPM, G_B will compensate this loss by changing the expression of the genes in G_B. Thus, mutating g_a eventually changes the expressions of the genes in both G_A and G_B.

From this conjecture, we conclude that there is a mapping between an SGA entry $(g_a, g_b, t_{a,b})$ and the corresponding single mutant dataset X_h, $g_h \in \{g_a, g_b\}$. This implies a non-trivial mapping, if the SGA entry corresponds to an SSL or epistatic relationship and we have a higher chance to find both g_a and g_b embedded in two pathways of a BPM. In that case, most of the genes in that BPM are supposed to have their expression changed in the single mutant dataset and an

appropriate regression method can correlate the changes in the single mutant gene expressions and the corresponding GI score.

Before stating the second conjecture, we define a relevant term, *neighbor*. We say that, gene g_b is an rth layer incoming neighbor of gene g_a in the directed gene network, if the shortest path from g_b to g_a consists of r directed edges. In that case, g_a is a rth layer outgoing neighbor of g_b. Figure 2 depicts the incoming and outgoing neighbors for the two genes in a genetic interaction.

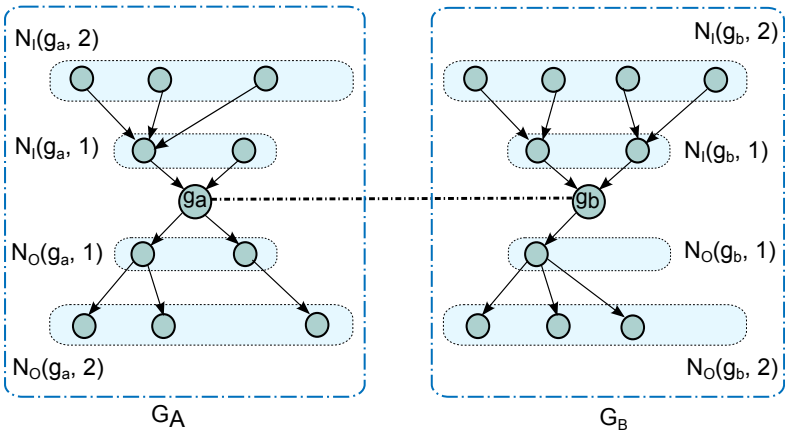


Fig. 2. This figure depicts the layered neighbor structure around a gene interaction edge (g_a, g_b). $N_{IN}(g_h, r)$ denotes the set of incoming neighbors of gene g_h at layer r. The set outgoing neighbor $N_{OUT}(g_h, r)$ is defined similarly. The example contains only up to 2 layers for each direction and gene. The dotted rectangles denote the putative BPM (G_A, G_B) around the gene interaction edge.

CONJECTURE 2. Let $\mathcal{B}$ denote a BPM, consisting of two pathways G_A and G_B. Consider an SSL edge $\mathcal{S} = \{g_a, g_b\}$ such that $g_a \in G_A$ and $g_b \in G_B$. If the expression of g_a changes significantly, then in G_A expression change is most prominent in the first layer of neighbors of g_a and gradually decreases with increasing layers. Similarly, in G_B the effect is most prominent for g_b and gradually decreases with increasing layers of neighbors. □

In brief, our conjecture is that the effect of a gene knockout eventually wanes away through the gene network. The rationale behind this is that the neighbors that are close to g_a and g_b have a higher chance of being connected only to the nodes of $\mathcal{B}$. The genes in the distant neighborhood have a greater possibility to take part in other pathways. Hence, the closer nodes are more susceptible to undergo a major effect, while the distant neighbors are supposed to be partially screened from that affect due to their activity in the other pathways. Based on these two conjectures, we build a regression based model that we describe in the next section.

3.3. *Regression based solution*

This section describes the customized regression based approach that we developed to build the mapping $\Upsilon : \mathcal{X}, \mathcal{G} \longrightarrow \mathcal{T}$, where $\mathcal{X}$, $\mathcal{G}$ and $\mathcal{T}$ denote the single mutant gene expression, gene network and GI score, respectively. Based on the two conjectures in Section 3.2, we extract a set of features for training and testing samples.

We start from the SGA and for each entry $(g_a, g_b, t_{a,b})$, we create a sample point provided either g_a or g_b has been mutated in the single mutant dataset available to us, otherwise we discard that SGA entry. Without losing the generality, assume that g_a has been mutated in this case. Thus, we extract the feature functions from the single mutant data $X_a = \{x_{a,1}, x_{a,2}, \cdots, x_{a,M}\}$. In designing the set of features, we leverage the information from gene networks by incorporating the two conjectures in our solution. According to the first conjecture, the mutated gene is suppose to perturb only the genes in the host BPM. Thus, while processing the SGA entry $(g_a, g_b, t_{a,b})$,

Table 1. The table summarizes the feature functions of the regression model and the corresponding parameters. Feature function represents the set of different features for the regression. A parameter quantifies the strength of the corresponding feature function.

Feature Function	Parameter	Description
$\Psi_{IN}(N_{IN}(g,r))$	$w_{IN}(g,r)$	For incoming neighbors of rth layer for gene g.
$\Psi_{OUT}(N_{OUT}(g,r))$	$w_{OUT}(g,r)$	For outgoing neighbors of rth layer for gene g.
$\Psi(g_b)$	w	For gene g_b when considering (g_a, g_b), g_a is knocked out.
	w_0	A constant representing the bias of the regression.

we consider only the genes from G_A and G_B and discard the ones from $\mathcal{G} - (G_A \cup G_B)$. We use the GI score $t_{a,b}$ as the label of the training sample.

Note that while we create the features for a training point, all the data we have is the single mutant data, GI scores and the gene networks. However, for a specific pair of genes (g_a, g_b) we do not know the set of genes that consists of the putative BPM $\mathcal{B} = (G_A, G_B)$ around the gene pair. Rather, we are suppose to validate that information using our model. In fact, if the SGA entry does not correspond to an SSL, there may not be a *real* BPM for the pair (g_a, g_b). To circumvent this problem, we assume the BPM as part of our model rather an input to the model.

Specifically, we use the concept of *rth layer neighbors*, introduced in Section 3.2. Let R represent the maximum number of layers to construct G_A and G_B. (Usually, R will be set by the user.) Let us denote the rth layer incoming and outgoing neighbors of gene g_h by $N_{IN}(g_h, r)$ and $N_{OUT}(g_h, r)$ respectively. Figure 2 demonstrates the layered structure of incoming and outgoing neighbors for a pair of genes. We define the putative BPM pathway for g_h as the union of the sets of incoming and outgoing neighbors of g_h up to the layer Rth given by,

$$G_H = \bigcup_{r=1}^{R} (N_{IN}(g_h, r) \cup N_{OUT}(g_h, r)) \tag{1}$$

In a comprehensive SGA data each GI score is a real valued number that varies in the range of two small numbers such as -5 to +5. However, if the score $t_{a,b}$ has a small magnitude (close to zero), the gene pair (g_a, g_b) may not have an SSL/epistatic interaction and may not be part of a BPM. Since the GI score $t_{a,b}$, which is the label of the regression model is real valued, we still shall use this sample point to train our model. However, the regression model is expected *not* to discover any interesting pattern of a BPM in the gene expression, and will adjust its parameters accordingly.

To incorporate the second conjecture, we design the features of the regression in a layered approach that directly depends on the concept of layered neighbors introduced in Section 3.2. We denote the feature function associated with the incoming neighbors of layer r of gene g by $\Psi_{IN}(N_{IN}(g,r))$ and the corresponding regression parameter by $w_{IN}(g,r)$. Similarly, the feature function and parameters for the rth layer outgoing neighbor are given by $\Psi_{OUT}(N_{OUT}(g,r))$ and $w_{OUT}(g,r)$, respectively. Thus, for $J \in \{IN, OUT\}$ we state that the feature function $\Psi_J(N_J(g_c,r))$ corresponds to neighbors of gene g_c in direction J at layer r. Given that g_h has been knocked out and we are considering the neighborhood of g_c, $\Psi_J(N_J(g_c,r))$ can be defined as follows,

$$\Psi_J(N_J(g_c,r)) = \frac{\sum_{g_i \in N_J(g_c,r)} |x_{h,i}|}{|N_J(g_c,r)|} \tag{2}$$

We define another feature function for g_b by $\Psi(g_b)$ and the corresponding parameter by w.

However, we do not create any feature function to capture the expression of g_a, since g_a is mutated and its expression may not be available for inspection. Finally, we create the last parameter w_0 that acts as a bias constant in the model. Table 1 summarizes the feature functions and the corresponding parameters. By aggregating all these feature functions, we can fit the SGA entry $(g_a, g_b, t_{a,b})$, where g_a is knocked out in the single mutant data as,

$$y_{a,b} = w_0 + w\Psi(g_b) + \sum_{\substack{r \in \{1,2,\cdots,R\}, \\ J \in \{IN,OUT\},\ c \in \{a,b\}}} w_J(g_c, r)\Psi_J(N_J(g_c, r)) \tag{3}$$

Parameter Estimation. When the ratio of number of samples to that of the parameters is small (typically less than 20), the estimated value of the parameters experience high variance due to overfitting of data.[11] To alleviate this problem, we augment a regularization term on top of the regression model. Specifically, we aim to minimize the difference between the parameter values at neighbor levels r and $r+1$, to smoothen the decaying of gene expression change. Formally, the regularization term can be written as,

$$Q = \sum_{\substack{r \in \{1,2,\cdots,R-1\}, \\ J \in \{IN,OUT\},\ c \in \{a,b\}}} |w_J(g_c, r+1) - w_J(g_c, r)| \tag{4}$$

We augment this regularization term with the objective function when estimating the parameters. Using least square error approach, we estimate the parameters of the regression by minimizing the following,

$$E = \sum_{a,b \in \{1,2,\cdots,M\},\ a<b} (t_{a,b} - y_{a,b})^2 + \lambda Q \tag{5}$$

A detailed discussion on the simplification of the regularization term can be found in Section 3 of Tibshirani et al.[12] We use the interior-point method to solve this parameter estimation problem.[13] The value of λ is estimated using five fold cross validation.

After the parameter estimation step of the regression method is complete, SSLPred is prepared to predict the GI score for a test sample. For a pair of test genes, we extract the set of features in the same way as that of a training point. Plugging those extracted features and the estimated parameters in Equation 3, we obtain the predicted GI score.

4. Experiments

In this section, we describe the experiments and discusses the results. Section 4.1 describes the datasets we used for the experiments. Section 4.2 demonstrates SSLPred with another relevant method recently published by Hescott et al.[7,14]

4.1. *Datasets*

As implied in Section 3.1, we classify the datasets into three different categories, namely, single gene mutant data, SGA data and gene networks. We decided on collecting datasets for *S. cerevisiae*, since this is a well researched organism of yeast with extensive datasets available for all these three categories.[3,10,15] We extracted BPMs from four studies, which were also used by

Hescott et al. to validate their method.[14] We employed these BPMs as gold standards in this paper. Next, we describe these datasets in detail.

(1) **Gene network BioGRID.** The dataset maintains one of the most comprehensive gene networks for S. cerevisiae.[15] We collected 155,287 the genetic interactions in total from this database.
(2) **Single mutant data.** In this paper, we collected 287 single gene knockout experiments from the compendium of expression profile of *S. cerevisiae* developed by Hughes et al.[10] Each experiment contains 6,316 entries. Every entry contains the the logarithm of after to before ratio of expressions of a gene as described in Section 3.1.
(3) **SGA data.** Costanzo et al. generated a genome scale SGA profile for *S. cerevisiae* with neatly 5.4 millions of genetic interactions out of nearly 75% genes.[3] Out of this comprehensive profile, we selected GI scores for 370,913 interactions such that for every edge, at least one of the two consisting genes was knocked out in the gene knockout experiments.
(4) **BPMs.** We obtained four sets of BPMs, all are of *S. cerevisiae*, itemized in the following – *Kelley-Ideker*,[6] *Ulitsky-Shamir*,[16] *Brady et al.*,[17] and *Ma et al.*[18] We denote a dataset using the authors' names of the corresponding paper. The numbers of BPMs that contain three or more genes in each pathway in these datasets are 160, 36, 959 and 54 respectively.

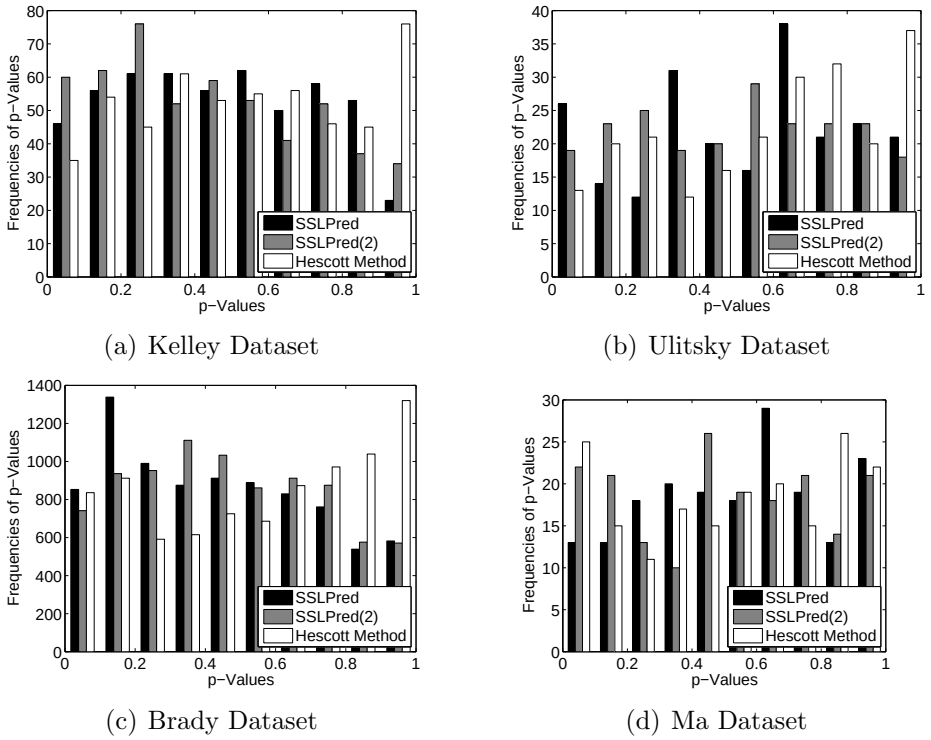
4.2. *Comparison with Hescott's Method*

This section describes the comparison between SSLPred and the method proposed by Hescott et al.[7,14] Hescott et al. employs microarray expression data of single gene knockout experiments to identify BPMs. Though their method does not predict GI score, according to our knowledge, this is the only published method that integrates the concept of single gene mutants and between pathway motifs.

Before coming to the main discussion, we describe how we create a matrix of predicted GI scores using five-fold cross validation. We divide the 287 knockout experiments into nearly equal five groups, each of them being a 57×6316 matrix. For each fold of cross validation, we use four out of five groups to create sample training points along with the corresponding GI scores as described in Section 3.3. If for a gene pair the corresponding GI score is not available, we discard the that sample point. After training, we create test points from the left-out one group and predict the test scores for them. Repeating this process in a five fold cross validation fashion, we predict GI scores for all possible pairs of genes from the 287×6316 matrix. Now that we have the predicted GI matrix which we denote by $\mathcal{T}_P$, we discuss how we employ it for comparison between SSLPred and the one proposed by Hescott et al.

Consider a BPM $\mathcal{B} = (G_A, G_B)$ obtained from a known sets of BPMs. Now, consider a gene $g_x \in G_A$. Hescott et al. ranks all the genes $\mathcal{G}$ of the organism with respect to g_x. Let us denote that rank by $\mathcal{G}_\Phi(g_x)$. Then, from that rank, it retrieves G_B and calculate the quality of retrieved G_B by a scoring method called *ClusterRankScore*. We now describe ClusterRankScore which is adapted from Gene Set Enrichment Analysis.[19]

ClusterRankScore accepts an ordered list of genes $\mathcal{L}$ and another set $\mathcal{C}$ as input. Then, it explore the distribution of $\mathcal{C}$ along $\mathcal{L}$. Intuitively, if $\mathcal{C}$ appears at the head or tail of $\mathcal{L}$, it is enriched with the specific properties represented by the ordered list $\mathcal{L}$. In the current context, consider a BPM $\mathcal{B} = \{G_A, G_B\}$. Let us knock out a gene g_a from G_A and measure the change



(a) Kelley Dataset (b) Ulitsky Dataset

(c) Brady Dataset (d) Ma Dataset

Fig. 3. Comparison of SSLPred with the method from Hescott et al. SSLPred and SSLPred(2) denote variants of SSLPred with at most one or two layers of neighbors, respectively. The X axis that ranges between zero and one represents the p-Values of the permutation tests. The Y axis represents the frequencies of the BPMs at a particular p-Value of the permutation test. The four sub-figures demonstrate that apart from on Ma dataset, SSLPred outperforms Hescott's method, as it maintains a higher frequency at the p-Value ranges between zero and 0.1.

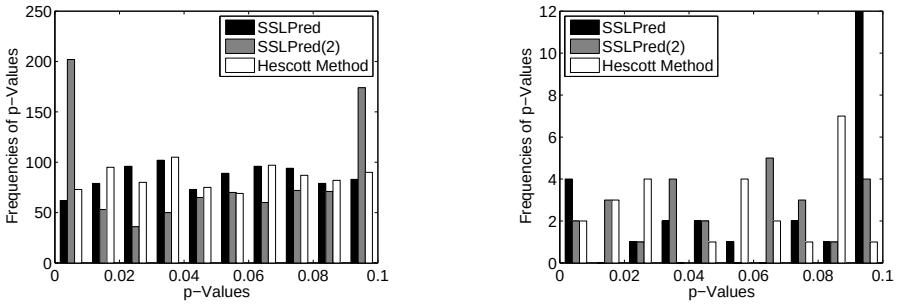
Table 2. BPMs with p-Values less than 0.l [%]

Method	BPMs			
	Kelley-Ideker	Ulitsky-Shamir	Brady et al.	Ma et al.
SSLPred	8.74	11.71	9.95	7.02
SSLPred (2)	11.4	8.55	8.64	11.89
Hescott et al.	6.65	5.85	9.75	13.51

of expressions for all the other genes. Hescott et al. now arrays the genes according to its own criteria. Here, this ordered list is $\mathcal{L}$ and the pathway G_B is $\mathcal{C}$. Thus, a correlation between the ordered gene list and the pathway G_B implies that the BPM $\mathcal{B}$ is validated by Hescott et al.

Using SSLPred we create a similar rank as follows. We obtain the predicted GI score of all the gene pairs (g_x, g_y), $g_y \in \mathcal{G}$ from the predicted GI matrix $\mathcal{T}_P$. Then we sort $\mathcal{G}$ in increasing value of the retrieved GI scores of (g_x, g_y). Let us denote the sorted list of genes by $\mathcal{G}_\Psi$. After this we calculate the ClusterRankScore of G_B based on $\mathcal{G}_\Psi$. Let us denote the ClusterRankScore of G_B with respect to Hescott et al. and SSLPred by $CRS_\Phi(g_x, G_B)$ and $CRS_\Psi(g_x, G_B)$, respectively.

To calculate the statistical significance of the two ClusterRankScore, we design separate permutation tests for each of them and calculate p-Values with respect to those permutation tests. Here the null hypothesis can be stated as $\mathcal{B}$ *is not a BPM*. A detailed account of *ClusterRankScore* and the permutation test can be found at Hescott et al.[7,14]



(a) Brady Dataset (b) Ma Dataset

Fig. 4. Comparison of SSLPred with the method from Hescott et al. SSLPred and SSLPred(2) denote variants of SSLPred with at most one or two layers of neighbors, respectively. The X axis, that ranges between zero and 0.l, represents the p-Values of the permutation tests. The Y axis represents the frequencies of the pathways at a particular p-Value of the permutation test. We display histograms for the very two datasets for which our method performs similar or worse in Figure 3. The two sub-figures demonstrate that on these two specific datasets, SSLPred maintains a higher frequency at the p-Value ranges between zero and 0.01.

Consider a BPM $\mathcal{B} = (G_A, G_B)$. For all the combinations, $(g_x, G_B), g_x \in G_A$ and $(g_y, G_A), g_y \in G_B$ we calculate the p-Values using the procedure described above. For every dataset, we plot the histograms of those p-Values. Since all the BPMs have been obtained from published literature, we assume them to be equivalent to a gold standard. Hence, in the histogram, an increased frequency of the BPMs with lower p-Values corresponds to a better quality of the BPM retrieval method.

Figure 3 compares SSLPred with the other method. SSLPred and SSLPred (2) denote variants of SSLPred with at most one or two layers of neighbors, respectively. The X axis, that ranges between zero and one, represents the p-Values of the permutation tests. The Y axis represents the frequencies of the BPMs at a particular p-Value of the permutation test. SSLPred outperforms Hescott et al. by 100%, 31%, 2% for Ulitsky, Kelley and Bradly dataset when the p-Value is equal to or smaller than 0.1. For SSLPred (2) the corresponding numbers are 46%, 71% and -12% respectively. For Dataset Ma, Hescott et al. is better by 48% and 12% than SSLPred and SSLPred (2) respectively. If we relax the p-Value to 0.3 we observe that SSLPred (2) performs better than Hescott et al. by 24%, 12%, 10% and 48% for Ulitsky, Brady, Ma and Kelley respectively. It can be concluded that that apart from on Ma dataset, SSLPred outperforms Hescott's method, since it maintains a higher frequency at the p-Value ranges between zero and 0.1. For Kelley and Ulitsky dataset SSLPred outperforms with a high margin between 0 to 0.1 p-Value range. For Ma and Brady datasets, the two methods perform very competitively on an average. Also, its is difficult to compare between two variants of SSLPred. Though these two variants are in close competition, SSLPred (2) has a slightly better advantage over SSLPred. Table 2 summarizes the results in Figure 3 by tabulating the percentage of BPMs with p-Values less than or equal to 0.1.

Figure 4 highlights a special case of the frequency distribution when the p-Value is restricted to be less than or equal to 0.01. We are specifically interested in the very two datasets Brady and Ma for which our method performs similar or worse when we restrict the p-Values to 0.1. Since a lower p-value implies a lower chance of false positive detection, these results are important

in determining the superiority of the competing methods. Here also we observe that SSLPred (2) has a better accuracy compared to Hescott et al. in identifying larger number of small p-Value BPMs. For Brady dataset SSLPred (2) outperforms the other method by 177%. For Ma dataset both of them detect two BPMs. This concludes that, our method demonstrates superior accuracy in validating more BPMs with very low p-Values (≤ 0.01). We also conducted a third set of experiments SSLPred (3) with highest layer of neighbors $R = 3$. However, SSLPred (3) performed poorly compared to Hescott et al. We believe that most BPMs are of small sizes with a diameter of less that equal to four edges as indicated in Kelley et al.[9] Hence, assuming a bigger size BPMs with $R = 3$ compromises the accuracy of our method.

Code. All the code developed in this paper is available from `http://bioinformatics.cise.ufl.edu/projects/SSLPred.html`.

5. Conclusion

In this paper, we developed a new method SSLPred to predict SSL interactions in an organism. Our method is built on the concept of Between Pathway Models, where majority of the SSL pairs span across the two functionally complementing pathways. We developed a regression based approach that learns the mapping between the gene expressions of single deletion mutant to the corresponding synthetic gene array.

We compared our method to the one by Hescott et al. for predicting the GI scores of *S. cerevisiae* on four benchmark datasets. On different experimental setups, on average SSLPred performed significantly better compared to the other method.

References

1. T. Baba, T. Ara and M. H. et al., *Mol Syst Biol* **2**, p. 2006.0008 (2006).
2. P. Beltrao, G. Cagney and N. Krogan, *Cell* **141**, 739 (2010).
3. M. Costanzo, A. Baryshnikova and J. B. et al., *Science* **327**, 425 (2010).
4. A. Tong, M. Evangelista and A. P. et al., *Science* **294**, 2364 (2001).
5. X. Pan, D. Yuan and D. X. et al, *Mol Cell* **16**, 487 (2004).
6. R. Kelley and T. Ideker, *Nat Biotechnol* **23**, 561 (2005).
7. B. Hescott, M. Leiserson and L. C. et al., *J Comput Biol* **17**, 477 (2010).
8. S. Collins, K. Miller and N. M. et al., *Nature* **446**, 806 (2007).
9. D. Kelley and C. Kingsford, *J Comput Biol* **18**, 379 (2011).
10. T. R. Hughes, M. J. Marton and A. R. J. et al, *Cell* **102**, 109 (2000).
11. V. S. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods*, 1st edn. (John Wiley & Sons, Inc., New York, NY, USA, 1998).
12. R. Tibshirani, M. Saunders and S. R. et al., *Journal of the Royal Statistical Society Series B* , 91 (2005).
13. M. H. Wright, *Bull. Amer. Math. Soc. (N.S)* **42**, 39 (2005).
14. B. J. Hescott, M. D. M. Leiserson and L. C. et al., Evaluating between-pathway models with expression data, in *RECOMB*, 2009.
15. B. Breitkreutz, C. Stark and T. R. et al., *Nucleic Acids Res* **36**, D637 (2008).
16. I. Ulitsky and R. Shamir, *BMC Syst Biol* **1**, p. 8 (2007).
17. A. Brady, K. Maxwell and N. D. et al., *PLoS One* **4**, p. e5364 (2009).
18. X. Ma, A. Tarone and W. Li, *PLoS One* **3**, p. e1922 (2008).
19. A. Subramanian, P. Tamayo and V. M. et al., *Proc Natl Acad Sci U S A* **102**, 15545 (2005).