Small sets of interacting proteins suggest functional linkage mechanisms via Bayesian analogical reasoning

Edoardo M. Airoldi1,∗, Katherine A. Heller2,∗ and Ricardo Silva3,∗

1Department of Statistics and FAS Center for Systems Biology, Harvard University, Cambridge, MA 02138,
2Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02114, USA and
3Department of Statistical Science, University College London, 1-19 Torrington Place, London WC1E 6BT, UK

ABSTRACT

Motivation: Proteins and protein complexes coordinate their activity to execute cellular functions. In a number of experimental settings, including synthetic genetic arrays, genetic perturbations and RNAi screens, scientists identify a small set of protein interactions of interest. A working hypothesis is often that these interactions are the observable phenomena of some functional process, which is not directly observable. Confirmatory analysis requires finding other pairs of proteins whose interaction may be additional phenotypical evidence about the same functional process. Extent methods for finding additional protein interactions rely heavily on the information in the newly identified set of interactions. For instance, these methods leverage the attributes of the individual proteins directly, in a supervised setting, in order to find relevant protein pairs. A small set of protein interactions provides a small sample to train parameters of prediction methods, thus leading to low confidence.

Results: We develop RBSets, a computational approach to ranking protein interactions rooted in analogical reasoning; that is, the ability to learn and generalize relations between objects. Our approach is tailored to situations where the training set of protein interactions is small, and leverages the attributes of the individual proteins indirectly, in a Bayesian ranking setting that is perhaps closest to propensity scoring in mathematical psychology. We find that RBSets leads to good performance in identifying additional interactions starting from a small evidence set of interacting proteins, for which an underlying biological logic in terms of functional processes and signaling pathways can be established with some confidence. Our approach is scalable and can be applied to large databases with minimal computational overhead. Our results suggest that analogical reasoning within a Bayesian ranking problem is a promising new approach for real-time biological discovery.

Availability: Java code is available at: http://www.gatsby.ucl.ac.uk/~rbas/.
Contact: airoldi@fas.harvard.edu; kheller@mit.edu; ricardo@stats.ucl.ac.uk

1 INTRODUCTION

Functional mechanisms in the cell involve cascades of interactions among gene products, mostly proteins and small molecules. In recent years, a number of large-scale efforts have collected and organized data produced by the community in publicly available, online databases. Human curators summarize the current state of our understanding of the cell’s functional landscape, for instance, by categorizing genes’ and proteins’ functional roles, distinguishing those that have been experimentally verified from those that are more probable using well established computational methods (Ashburner et al., 2000; Finn et al., 2008; Kanchisa and Goto, 2000; Letunic et al., 2006; Mewes et al., 2004; Mulder et al., 2007; SGD). However, the current understanding of signaling and regulatory dynamics that instantiate functions in the cell is far from complete. Functional discovery is key. The main challenges are the scale of the problem and the costs of the necessary experimental validation (Evanisko, 2009). The space of possible interactions and their multiple functional roles is very large. Exploring it at random is inefficient—expensive and time consuming. A sensible approach is that of prioritizing the experiments, testing the most probable functions and interactions first (Schwartz et al., 2008).

Methods routinely used by biologists today to assist biological discovery boil down to assigning priority scores to protein–function pairs and to protein–protein–function triples, individually or in small sets. Here we consider a slightly different scenario, where we want to assign priority scores to protein–protein pairs in the absence of a clear functional implication. We propose a methodology to rank protein–protein pairs that are most similar to a given small set of protein–protein pairs of interest. We show that our approach, based on analogical reasoning, is able to identify additional protein–protein pairs that share functional implications with the set of pairs of interest. These results suggest that our method may be useful in establishing new functional implications, which are not necessarily well annotated in existing databases.

Bayesian analogical reasoning, the ability to learn and generalize relations between objects, seems particularly appropriate for biological discovery because it does not estimate feature weights and relations (during training) that are then fixed in stone to produce predictions. Rather, it calibrates a prior distribution on feature weights that takes into account how features relate to one another, but then updates these weights depending on the query, on the fly.

To illustrate the practical benefits of the proposed methodology, consider the following scenario. A certain cellular process is poorly understood and the corresponding gene ontology function is poorly annotated, say, protein kinases. Consider extent protein interaction prediction methods; they typically estimate the correlation structure among features of all known proteins that is useful to predict existing interactions, e.g., with a Bayesian network (Myers et al., 2005), store it on a web-server, and use it to rank interactions every time users submit a set of protein interactions. However, if we were to discover a few interactions with protein kinases and submit them as a query, extant systems would not be able to rank other interactions between kinases well, mainly for two reasons: (1) we
are submitting a small training set, and (ii) the correlations that were estimated during the training phase did not include kinase interactions, which were poorly annotated, and the corresponding ranking mechanism (feature weights in a classifier or conditional probability tables in a Bayesian network) will have to rely on poor estimates of the correlations. The proposed methodology on the other hand, addresses the issue of small training sets (i) by regularizing the estimates through the use of a prior distribution on the weights in a Bayesian estimation setting. It gets around the issue of poor annotations and poor parameters estimates during training (ii) by updating the feature weights on the fly using information in the query set of interactions. In this example, our method would be able to attribute more importance to features such as specific protein domains, other features of the protein amino-acid sequences and cellular localization in order to retrieve other kinase interactions with high probability. The proposed method is also computationally scalable to large databases.

In this article, we develop a statistical approach that leverages relational learning: given a set of protein pairs \( S = \{ A^{(1)}B^{(1)}, A^{(2)}B^{(2)}, \ldots, A^{(N)}B^{(N)} \} \) it measures how well other pairs \( A'B' \) fit in with the set \( S \). Our work addresses the question of whether the relation between proteins \( A \) and \( B \) is analogous to those relations found in \( S \). Such questions are particularly relevant in exploratory data analysis, where an investigator might want to search for protein pairs that are analogous to pairs in the query set of interest.

Our approach combines a similarity measure on function spaces with Bayesian analysis to produce a ranking of pairs. It requires data containing attributes of the proteins of interest and a link matrix specifying existing relationships; no further attributes of such relationships are necessary. We illustrate the potential of our method in a candidate pair and the features of objects in the query pairs. We show that our approach can work in practice even if a small set of protein pairs is provided, when evaluated on two functional categorization systems: GO and MIPS. Furthermore, we develop a variational inference algorithm that scales to very-large databases.

**Related problems and approaches:** similarity-based methods compute the score of a given protein pair as a function of observable attributes (e.g. corresponding genes’ expression levels) of the two proteins under examination. Possibly multiple functions are then assigned to each protein \( P_i \) using a guilt-by-association principle, e.g. by looking up in a curated database the most frequent functions of the proteins \( P_i \)’s that interact with \( P_i \) with high scores (Butte and Kohane, 2006; Clare and King, 2003; Margolin et al., 2006; Markowetz et al., 2007). The main obstacle to the success of these methods is that the similarity among proteins’ attributes is informative about a shared function only to a minor extent (Margolin and Califano, 2007). Clustering-based methods compute the score of a given protein pair as a function of topological properties of the interactions. The interactome is divided into clusters, or modules, and the inferred memberships of proteins-to-modules are used as attributes to predict protein functions in a number of ways (Adamczyk et al., 2006; AltshUl-Amin et al., 2006; Bader and Hougue, 2003; Enright et al., 2002; Sharan et al., 2005). Recent results, however, suggest that a simple non-clustering method that relies on the guilt-by-association principle is more accurate in predicting proteins’ functional roles (Song and Singh, 2009). The curation of the interactome is noisy and its degree of inaccuracy is higher than expected (Cosick et al., 2009). Proteins participate in the execution of multiple functional processes (Airoldi et al., 2008).

These are perhaps two of the factors that can explain this surprising result. Data integration methods compute the score of a given protein pair as a function that combines observed attributes, both of individual proteins and of protein pairs, from multiple data sources. These integrative methods have been the most successful to date in predicting protein function (Fraser and Marcotte, 2004; Guan et al., 2008; Llewellyn and Eisenberg, 2008) and networks of functional relationships between proteins (Hess et al., 2009; Huttererower et al., 2009; Ideker et al., 2002; Jensen et al., 2003; Jensen et al., 2009; Lee et al., 2004; Myers et al., 2005; Troyanskaya et al., 2003; von Mering et al., 2005) from large collections of data. The main drawback of these approaches is that they often involve a hodgepodge of different scores that only resemble a functional relation (Sterne and Smith, 2001). Multiple scores are usually combined using ad hoc considerations that are specific to the data under examination. The computational burden is substantial.

**Our contribution:** we introduce a new approach that determines similarity between protein pairs by essentially computing similarity between predictive functions that relate proteins pairs to functions. Our methodology is based on the idea of analogical reasoning.

The proposed methodology departs from the existing approaches in a few aspects. First, we explicitly address the ranking problem of which pairs of proteins are most similar to an input set of protein pairs. This is different from the typical protein pair prediction problem addressed in the literature. While we rely on similarity to rank protein pairs, we compute similarity between **predictive functions** that map pairs of proteins to functional links, rather than between attributes of functionally related proteins. This will require a description of the space where such functions live. Second, our methodology focuses on the case where little evidence is available, i.e. a small set of input pairs. This will require using prior functional knowledge to calibrate a prior on the space of functions that places mass on a most likely subspace of functions. Third, our methodology is rooted in Bayesian statistical methodology. It captures and updates prior knowledge about proteins’ functions stored in online databases within a hierarchical Bayesian model. It can easily be integrated in a computational pipeline for general use. Fourth, our variational inference algorithm scales to very-large databases.

We consider two case studies where the underlying functional implications of the interactions we consider correspond to physical protein binding events and to signaling events within a metabolic pathway. We quantify the extent to which small sets of interacting proteins are suggestive of functional linkage mechanisms on the collection of pathways in KEGG and of functions in MIPS.

**2 METHODS**

To define an analogy is to define a measure of similarity between structures of related objects. In our setting, we need to measure the similarity between pairs of objects. The key aspect that distinguishes our approach from others is that we focus on the similarity between **predictive functions** that map pairs to links, rather than focusing on the similarity between the attributes of objects in a candidate pair and the features of objects in the query pairs.

As an illustration, consider an analogical reasoning question from a SAT-like exam where for a given pair (say, water:river) we have to choose, out of 5 pairs, the one that best matches the type of relation implicit in such a ‘query’. In this case, it is reasonable to say car:highway would be a better...
A prototypical analogical reasoning question is shown in Figure 1. The result of this effort is that taxonomies such as the Gene Ontology (Ashburner et al., 2004) are generated by two independent models: one classified all pairs \( (A, B) \) into the positive class \( R=1 \) and a random subset of data points as the negative class \( R=0 \) (e.g. Tumay, 2008).

In this setup, the event \( R=1 \) is equated with the event that \( X \) and the elements of \( S \) are i.i.d points generated by the same model. The event \( R=0 \) is the event by which \( X \) and \( S \) are generated by two independent models: one for \( X \) and another for \( S \). The parameters of all models are random variables that have been integrated out, with fixed (and common) hyperparameters. The result is the instantiation of (1) as

\[
\log P(X | R=1, S) - \log P(X | R=0, S) = \log \frac{P(X,S)}{P(X|S)}
\]

the Bayesian score function by which we rank points \( X \) given a query \( S \). The right-hand side was rearranged to provide a more intuitive graphical model. From this graphical model interpretation we can see that the score function is a Bayes factor comparing two models (Kass and Raftery, 1995).

Next, we describe how the Bayesian sets method can be adapted to define analogical similarity in biological networks settings.

### 2.3 Bayesian analogical similarity for protein pairs

Let \( A \) and \( B \) represent object spaces. To say that an interaction \( A \leftrightarrow B \) is analogous to \( S=\{A^{(1)} \leftrightarrow B^{(1)}, A^{(2)} \leftrightarrow B^{(2)}, \ldots, A^{(n)} \leftrightarrow B^{(n)}\} \) amounts to implicitly defining a measure of similarity between the pair \( A \leftrightarrow B \) and the set of pairs \( S \), where each query item \( A^{(i)} \leftrightarrow B^{(i)} \) corresponds to some pair \( A_i \leftrightarrow B_i \). However, this similarity is not directly derived from the similarity of the information contained in the distribution of objects themselves, \( \{A^{(i)}\} \subset A, \{B^{(i)}\} \subset B \). Rather, the similarity between \( A \leftrightarrow B \) and the set \( S \) is defined in terms of the similarity of the functions mapping the pairs as being linked. Each possible function captures a different possible relationship between the objects in the pair.

**Bayesian analogical reasoning:** Consider a space of latent functions in \( A \times B \rightarrow [0,1] \). Assume that \( A \) and \( B \) are two objects classified as linked by some unknown function \( f(A, B) \), i.e. \( f(A, B)=1 \). We want to quantify how similar the function \( f(A, B) \) is to the function \( g(\cdot, \cdot) \), which classifies all pairs \( (A^{(i)} \leftrightarrow B^{(i)}) \in S \) as being linked, i.e. where \( g(A^{(i)} \leftrightarrow B^{(i)})=1 \). The similarity depends on the observations \( [S,A,B] \) and our prior distribution over \( f(\cdot, \cdot) \) and \( g(\cdot, \cdot) \).

Functions \( f(\cdot, \cdot) \) and \( g(\cdot, \cdot) \) are unobserved, hence the need for a prior that will be used to integrate over the function space. Our similarity metric will be defined using Bayes factors, as explained next.
2.3.1 Scoring analogy of linkage functions using logistic regression

For simplicity, we will consider a family of latent functions that is parameterized by a finite-dimensional vector: the logistic regression function with multivariate Gaussian priors for its parameters.

For a particular pair \((A', B')\in \mathcal{A}\times \mathcal{B}\), let \(x' = (\Phi_1(A', B')\Phi_2(A', B'), \ldots, \Phi_d(A', B'))^T\) be a point on a feature space defined by the mapping \(\Phi: \mathcal{A}\times \mathcal{B}\to \mathbb{R}^d\). This feature space mapping computes a \(d\)-dimensional vector of attributes of the pair that may potentially be relevant to predicting the relation between the objects in the pair. Let \(L^S = \{0, 1\}\) be an indicator of the existence of a link or relation between \(A'\) and \(B'\) in the database. Let \(\Theta = (\theta_1, \ldots, \theta_d)^T\) be the parameter vector for our logistic regression model such that

\[P(L^S = 1 | x', \Theta) = \text{logistic}(\Theta^T x'),\]

and \(\text{logistic}(x) = \frac{1}{1 + e^{-x}}\) is the standard mapping from \(\mathbb{R}\to [0, 1]\).

We now apply the same score function underlying the Bayesian model from Section 2.2. However, instead of comparing objects by marginalizing over the parameters of their feature distributions, we compare functions for link indicators by marginalizing over the parameters of the functions.

Let \(L^B\) be the vector of link indicators for \(S\); in fact each \(L^B\in \{0, 1\}^{|S|\times |S|}\) has the value \(L=1\), indicating that every pair of objects in \(S\) is linked. Consider the following Bayes factors:

\[\frac{P(L^B = 1 | x', \Theta)}{P(L^B = 1 | x', \Theta')} = \frac{\prod_{(A, B)\in \mathcal{A}\times \mathcal{B}} \text{logistic}(\Theta^T x')} {\prod_{(A, B)\in \mathcal{A}\times \mathcal{B}} \text{logistic}(\Theta'^T x')} = \frac{\prod_{(A, B)\in \mathcal{A}\times \mathcal{B}} \text{logistic}(\Theta^T x')} {\prod_{(A, B)\in \mathcal{A}\times \mathcal{B}} \text{logistic}(\Theta'^T x')}\]

This is an adaptation of Equation (2) where relevance is defined now by whether \(L^B\) and \(L^S\) were generated by the same model, for fixed \(|x', \Theta|\). In one sense, this is a discriminative Bayesian sets model, where we predict links instead of modeling joint object features. Since we are integrating out \(\Theta\) for this particular parameter vector is needed. Thus, each pair \((A', B')\) is evaluated with respect to a query set \(S\) by the score function given in (4), rewritten after taking a logarithm and dropping constants as:

\[\text{score}(A', B') = \log P(L^B = 1 | x', \Theta) - \log P(L^B = 1 | x') - \log \text{logistic}(\Theta^T x')\]

The exact details of our procedure are as follows. We are given a relational database \((D_1, D_2, D_3, L, C)\). Dataset \(D_3\) is a sample of objects of type \(\mathcal{A}\) (\(\mathcal{B}\)). Relationship table \(L\) is a binary matrix modeled as generated from a logistic regression model of link existence. A query proceeds according to the following steps:

(i) the user selects a set of pairs \(S\) that are linked in the database, where the pairs in \(S\) are assumed to have some relation of interest;
(ii) perform Bayesian inference to obtain the corresponding posterior distribution for \(\Theta\) by \(P(\Theta|L^B = 1)\), given a Gaussian prior \(\phi(\Theta)\).
(iii) iterate through all linked pairs, computing the following for each: \(P(L^B = 1 | x', \Theta)\).

The integral presented above does not have a closed formula. Because computing the integrals by a Monte Carlo method for a large number of pairs would be unreasonable, we use a variational approximation (Airoldi, 2007; Jordan et al., 1999). Figure 3 presents a summary of the approach.

The suggested setup scales as \(O(K^2)\) with the feature space dimension, due to the matrix inversions necessary for (variational) Bayesian logistic regression (Jaakkola and Jordan, 2000). A less precise approximation to \(P(\Theta|L^B = 1)\) can be imposed if the dimensionality of \(\Theta\) is too high. However, it is important to point out that once the initial integral \(P(\Theta)\) is approximated, each score function can be computed at a cost of \(O(K^2)\).

Our analogical reasoning formulation is a relational model in that it models the presence and absence of interactions between objects. By conditioning on the link indicators, the similarity score between \(AB\) and \(CD\) is always

\[\text{Fig. 3. General framework of the procedure: first, a 'prior' over parameters \(\Theta\) for a link classifier is defined empirically using linked and unlinked pairs of points (the dashed edges indicate that creating a prior empirically is possible, but in practice we rely on this method). Given a query set \(S\) of linked pairs of interest, the system computes the predictive likelihood of each linked pair \(D^S\in D^\ast\) and compares it to the conditional predictive likelihood, given the query. This defines a measure of similarity with respect to \(S\) by which all pairs in \(D^\ast\) are sorted.}

\[\text{A function of pairs (A, B) and (C, D) that is not in general decomposable as similarities between A and C, and B and D.}

2.3.2 Empirical priors and calibration using biological databases

The choice of prior is based on the observed data, in a way that is equivalent to the choice of priors used in the original formulation of Bayesian sets (Ghahramani and Heller, 2005). Let \(\hat{\Theta}\) be the maximum-likelihood estimator (MLE) of \(\Theta\) using the relational database \((D_2, D_3, L)\). Since the number of possible pairs grows at a quadratic rate with the number of objects, we do not use the whole database for MLE. Instead, to get \(\hat{\Theta}\) we use all linked pairs as members of the ‘positive’ class \((L=1)\) and subsample unlinked pairs as members of the ‘negative’ class \((L=0)\). We subsample by sampling each object uniformly at random from the respective datasets \(D_2\) and \(D_3\) to get a new pair. Since link matrices \(L_{AB}\) are usually very sparse, in practice this will almost always provide an unlinked pair. Section 3 provides more details.

We use the prior \(P(\Theta)\sim N(0, cI)^{-1}\), where \(X(\Theta|V)\) is a normal of mean \(\Theta\) and variance \(V\). Matrix \(T\) is the empirical second moments matrix of the linked object features, although a different choice might be adequate for different applications. Constant \(c\) is a smoothing parameter set by the user. In all of our experiments, we set \(c\) to be equal to the number of positive pairs. A good choice of \(c\) might be important to obtain maximum performance, but we leave this issue as future work. (Wang et al., 2009) present some sensitivity analysis results.

Empirical priors are a sensible choice, since this is a retrieval, not a predictive, task. Basically, the entire dataset is the population, from which prior information is obtained on possible query sets. A data-dependent prior based on the population is important for an approach such as Bayesian sets, since deviations from the ‘average’ behavior in the data are useful to discriminate between subpopulations.

2.3.3 Extensions to continuous/multivariate relations

Although here we focused on measuring similarity of qualitative relationships, the same idea could be extended to continuous (or ordinal) measures of relationship, or relationships where each \(L^B\) is a vector. Several similarity metrics can be defined on this vector of continuous relationships. Given data on protein properties, one can easily modify our approach by substituting the logistic regression component with some multiple regression model.

3 RESULTS

The budding yeast is a unicellular organism that has become a de-facto model organism for the study of molecular and cellular biology (Botstein et al., 1997). There are about 6000 proteins in

![Image](https://academic.oup.com/bioinformatics/article-abstract/27/13/i374/181396)
the budding yeast, which interact in a number of ways (Cherry et al., 1997). For instance, proteins bind together to form protein complexes, the physical units that carry out most functions in the cell (Krogan et al., 2006). In recent years, significant resources have been directed to collect experimental evidence of physical proteins binding, in an effort to infer and catalogue protein complexes and their multifaceted functional roles (Fields and Song, 1989; Ho et al., 2002; Ito et al., 2000; Uetz et al., 2000). Currently, there are four main sources of interactions between pairs of proteins that target proteins localized in different cellular compartments with variable degrees of success: (i) literature curated interactions (Reguly et al., 2006); (ii) yeast two-hybrid (Y2H) interaction assays (Yu et al., 2008); (iii) protein fragment complementation (PCA) interaction assays (Tarassov et al., 2008); and (iv) tandem affinity purification (TAP) interaction assays (Gavin et al., 2006; Krogan et al., 2006). These collections include a total of about 12,292 protein interactions (Jensen and Bork, 2008), although the number of such interactions is estimated to be between 18,000 (Yu et al., 2008) and 30,000 (von Mering et al., 2002).

3.1 Design of experiments

We consider multiple functional categorization systems for the proteins in budding yeast. For evaluation purposes, we use individual proteins’ functional annotations curated by the Munich Institute for Protein Sequencing (MIPS) (Mewes et al., 2004), those by the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000), and those by the Gene Ontology consortium (GO) (Ashburner et al., 2000). We consider multiple collections of physical protein interactions that encode alternative semantics. Physical protein interactions in the MIPS curated collection measure physical binding events observed experimentally in Y2H and TAP experiments, whereas physical protein-to-protein interactions in the KEGG curated collection measure a number of different modes of interactions, including phosphoreilation, methylation and physical binding, all taking place in the context of a specific signaling pathway. So we have three possible functional annotation databases (MIPS, KEGG and GO) and two possible link matrices (MIPS and KEGG), which can be combined.

Our experimental pipeline is as follows. (i) Pick a database of functional annotations, say MIPS, and a collection of interactions, say MIPS (again). (ii) Pick a pair of categories, $M_1$ and $M_2$. For instance, take $M_1$ to be cytoplasm (MIPS 40.03) and $M_2$ to be cytoplasmic and nuclear degradation (MIPS 06.13.01). (iii) Sample, uniformly at random and without replacement, a set $S$ of 15 interactions in the chosen collection. (iv) Rank other interacting pairs according to the score in Equation (5) and, for comparison purposes, according to three other approaches to be described in Section 3.1.4. (v) The process is repeated for a large number of pairs $M_1 \times M_2$, and 5 different query sets $S$ are generated for each pair of categories. (vi) Calculate an evaluation metric for each query and each of the four scores. Report a comparative summary of results.

3.1.1 Protein specific features

The protein specific features were generated using the datasets summarized in Table 1 and an additional dataset (Qi et al., 2006), from which 20 gene expression attributes were obtained. Each gene expression attribute for a protein pair $P_i; P_j$

\begin{equation}
\hat{\Sigma}^{-1} = X_{POS}^T X_{POS}
\end{equation}

where $X_{POS}$ is the matrix containing the protein–protein features only of the linked pairs used in the MLE computation.

3.1.3 Evaluation metrics

We propose an objective measure of evaluation that is used to compare different algorithms. Consider a query set $S$, and a ranked response list $R = [R_1, R_2, R_3, \ldots, R^S]$ of protein–protein pairs. Every element of $S$ is a pair of proteins $P_i; P_j$ such that $P_i$ is of class $M_1$ and $P_j$ is of class $M_2$, where $M_1$ and $M_2$ are classes from either MIPS, KEGG or GO Ontology. In general, proteins belong to multiple classes. The retrieval algorithm that generates $R$ does not receive any information concerning the MIPS, KEGG or GO taxonomy. $R$ starts with the linked protein pair that is judged most similar to $S$, followed by the other protein pairs in the population, in decreasing order of similarity. Each algorithm has its own measure of similarity.

The evaluation criterion for each algorithm is as follows: as before, we generate a precision-recall curve and calculate the area under the curve (AUC). We also calculate the proportion (TOP10),
Bayesian analogical reasoning identifies latent functional linkage mechanisms

among the top 10 elements in each ranking, of pairs that match the
original \( \{ M_1, M_2 \} \) selection (i.e. a ‘correct’ \( P_i P_j \) is one where \( P_i \)
is of class \( M_1 \) and \( P_j \) of class \( M_2 \), or vice-versa. Notice that each protein
belongs to multiple classes, so both conditions might be satisfied.) Since a researcher is only likely to look at the top ranked pairs, it makes sense to define a measure that uses only a subset of the ranking. AUC and TOP10 are our two evaluation measures.

The original classes \( \{ M_1, M_2 \} \) are known to the experimenter but not known to the algorithms. Our criterion is rather stringent, in the sense it requires a perfect match of each \( \mathbf{R}^i \) with the MIPS, KEGG or GO categorization. There are several ways by which a pair \( \mathbf{R}^i \) might be analogous to the relation implicit in \( \mathbf{S} \), and they do not need to agree with MIPS, GO or KEGG. Still, if we are willing to believe that these standard categorization systems capture functional organization of proteins at some level, this must lead to association between categories given to \( \mathbf{S} \) and relevant sub-populations of protein–protein interactions similar to \( \mathbf{S} \). Therefore, the corresponding AUC and TOP10 are useful tools for comparing different algorithms even if the actual measures are likely to be pessimistic for a fixed algorithm.

3.1.4 Competing algorithms

We compare our method against a variant of it and two similarity metrics widely used for information retrieval: (i) the cosine score (Manning et al., 2008), denoted by \( \text{COS} \); (ii) the nearest neighbor score, denoted by \( \text{NNS} \); (iii) the relational maximum-likelihood sets score, denoted by \( \text{MLS} \). The nearest neighbor score measures the minimum Euclidean distance between \( \mathbf{R}^i \) and any individual point in \( \mathbf{S} \), for a given query set \( \mathbf{S} \) and a given candidate point \( \mathbf{R}^i \). The relational MLS is a variation of \( \text{RBSets} \) where we initially sample a subset of the unlinked pairs (10 000 points in our setup) and, for each query \( \mathbf{S} \), we fit a logistic regression model to obtain the parameter estimate \( \hat{\theta}_{\text{MLE}} \). We also use a logistic regression model fit to the whole dataset (the same one used to generate the prior for \( \text{RBSets} \)), giving the estimate \( \hat{\theta}_{\text{MLE}} \). A new score, analogous to (5), is given by

\[
\log \frac{P(Y_i = 1 | X^i, \hat{\theta}_{\text{MLE}})}{1 - P(Y_i = 1 | X^i, \hat{\theta}_{\text{MLE}})} = \log \frac{P(Y_i = 1 | X^i, \hat{\theta}_{\text{MLE}})}{1 - \hat{\theta}_{\text{MLE}}},
\]

i.e. we do not integrate out the parameters or use a prior, but instead the models are fixed at their respective estimates.

Neither COS or NNS can be interpreted as measures of analogical similarity, in the sense that they do not take into account how the protein pair features \( \mathbf{X} \) contribute to their interaction\(^3\).

3.2 Analysis of physical interactions (MIPS)

For this batch of experiments, we use the MIPS network of protein–protein interactions to define the relationships. In the initial experiment, we selected queries from all combinations of MIPS classes for which there were at least 50 linked pairs \( P_i P_j \) in the network that satisfied the choice of classes. Each query set contained 15 pairs. After removing the MIPS-categorized proteins for which we had no feature data, we ended up with a total of 6125 proteins and 7788 positive interactions. We set the prior for \( \text{RBSets} \) using a sample of 225 842 pairs labeled as having no interaction, as selected by (Q et al., 2006).

\(^3\)As a consequence, none uses negative data. Another consequence is the necessity of modeling the input space that generates \( \mathbf{X} \), a difficult task given the dimensionality and the continuous nature of the features.

for each tentative query set \( \mathbf{S} \) of categories \( \{ M_1, M_2 \} \), we scored and ranked pairs \( P_i P_j \) such that both \( P_i \) and \( P_j \) were connected to some protein appearing in \( \mathbf{S} \) by a path of no more than two steps, according to the MIPS network. The reasons for the filtering are 2-fold: to increase the computational performance of the ranking since fewer pairs are scored; and to minimize the chance that undesirable pairs would appear in the top 10 ranked pairs. Tentative queries would not be performed if after filtering we obtained fewer than 50 possible correct matches. Trivial queries, where filtering resulted only in pairs in the same class as the query, were also discarded. The resulting number of unique pairs of categories \( \{ M_1, M_2 \} \) was 931 classes of interactions. For each pair of categories, we sampled our query set \( \mathbf{S} \) 5 times, generating a total of 4655 rankings per algorithm.

We run two types of experiments. In one version, we give to \( \text{RBSets} \) the data containing only the 45 (continuous) microarray measurements. In the second variation, we provide to \( \text{RBSets} \) all 61 variables, including the 16 sparse binary indicators. However, we noticed that the addition of the 16 binary variables hurts \( \text{RBSets} \) considerably. We conjecture that one reason might be the degradation of the variational approximation. Including the binary variables hardly changed the other three methods, so we choose to use the 61 variable dataset for the other methods.

Table 2 summarizes the results of this experiment. We show the number of times each method wins according to both the AUC and TOP10 criteria. The number of wins is presented as divided by 5, since fewer pairs are scored; and to minimize the chance that undesirable pairs would appear in the top 10 ranked pairs. Tentative queries would not be performed if after filtering we obtained fewer than 50 possible correct matches. Trivial queries, where filtering resulted only in pairs in the same class as the query, were also discarded. The resulting number of unique pairs of categories \( \{ M_1, M_2 \} \) was 931 classes of interactions. For each pair of categories, we sampled our query set \( \mathbf{S} \) 5 times, generating a total of 4655 rankings per algorithm.

For each tentative query set \( \mathbf{S} \) of categories \( \{ M_1, M_2 \} \), we scored and ranked pairs \( P_i P_j \) such that both \( P_i \) and \( P_j \) were connected to some protein appearing in \( \mathbf{S} \) by a path of no more than two steps, according to the MIPS network. The reasons for the filtering are 2-fold: to increase the computational performance of the ranking since fewer pairs are scored; and to minimize the chance that undesirable pairs would appear in the top 10 ranked pairs. Tentative queries would not be performed if after filtering we obtained fewer than 50 possible correct matches. Trivial queries, where filtering resulted only in pairs in the same class as the query, were also discarded. The resulting number of unique pairs of categories \( \{ M_1, M_2 \} \) was 931 classes of interactions. For each pair of categories, we sampled our query set \( \mathbf{S} \) 5 times, generating a total of 4655 rankings per algorithm.

We run two types of experiments. In one version, we give to \( \text{RBSets} \) the data containing only the 45 (continuous) microarray measurements. In the second variation, we provide to \( \text{RBSets} \) all 61 variables, including the 16 sparse binary indicators. However, we noticed that the addition of the 16 binary variables hurts \( \text{RBSets} \) considerably. We conjecture that one reason might be the degradation of the variational approximation. Including the binary variables hardly changed the other three methods, so we choose to use the 61 variable dataset for the other methods.

Table 2 summarizes the results of this experiment. We show the number of times each method wins according to both the AUC and TOP10 criteria. The number of wins is presented as divided by 5, since fewer pairs are scored; and to minimize the chance that undesirable pairs would appear in the top 10 ranked pairs. Tentative queries would not be performed if after filtering we obtained fewer than 50 possible correct matches. Trivial queries, where filtering resulted only in pairs in the same class as the query, were also discarded. The resulting number of unique pairs of categories \( \{ M_1, M_2 \} \) was 931 classes of interactions.

<table>
<thead>
<tr>
<th>Method</th>
<th>#AUC</th>
<th>#TOP10</th>
<th>#AUC.S</th>
<th>#TOP10.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>240</td>
<td>294</td>
<td>219</td>
<td>277</td>
</tr>
<tr>
<td>NNS</td>
<td>42</td>
<td>122</td>
<td>28</td>
<td>75</td>
</tr>
<tr>
<td>MLS</td>
<td>105</td>
<td>270</td>
<td>52</td>
<td>198</td>
</tr>
<tr>
<td>RBSets</td>
<td>542</td>
<td>556</td>
<td>578</td>
<td>587</td>
</tr>
</tbody>
</table>

The first two columns, #AUC and #TOP10, count the number of times the respective method obtains the best score according to the AUC and TOP10 criteria, respectively, among the 4 approaches. This is divided by the number of replications of each query type (5). The last two columns, #AUC.S and #TOP10.S are ‘smoothed’ versions of this statistic: a method is declared the winner of a round of 5 replications if it obtains the best score in at least 3 out of the 5 replications. The top table shows the results when only the continuous variables are used by \( \text{RBSets} \), and in the bottom table when the discrete variables are also given to \( \text{RBSets} \).

Table 2. Number of times each method wins when querying pairs of MIPS classes using the MIPS protein–protein interaction network

\(^2\)As a consequence, none uses negative data. Another consequence is the necessity of modeling the input space that generates \( \mathbf{X} \), a difficult task given the dimensionality and the continuous nature of the features.
the continuous variables. For this reason, for the rest of this section all
analysis and experiments will consider only this case. Table 3 displays a pairwise comparison of the methods. In this Table, we show how often the row method performs better than the column method, among those trials where there was no tie. Again, RBSES
takes advantage of the relative performance of each algorithm. In Table 4, we show the proportion of correct hits among the top 10 for each algorithm for our queries using MIPS
categorization and also GO categorization, as explained in the next
section. About 14% of the time, all pairs in the top 10 pairs ranked
by RBSES were of the intended type, compared to 8% of the second
best approach.

3.2.1 Changing the categorization system A variation of this
experiment was performed where the protein categorizations do not
come from the same family as the link network, i.e. where we
used the MIPS network but not the MIPS categorization. Instead we
performed queries according to Gene Ontology categories. Starting
from 150 pre-selected GO categories (Myers et al., 2006), we once
again generated unordered category pairs \(\{M_1, M_2\}\). A total of 179
queries, with 5 replications each (a total of 895 rankings), were
generated and the results summarized in Table 5.

This is a more challenging scenario for our approach, which is
optimized with respect to MIPS. Still, we are able to outperform
other approaches. Differences are smaller, but consistent. In the
pairwise comparison of RBSES against the second best method,
CO, our method wins 62% of the time by the TOP10 criterion.

3.2.2 The role of filtering In both experiments with the MIPS
network, we filtered candidates by examining only a subset of the
proteins linked to the elements in the query set by a path of no more
than two proteins. It is relevant to evaluate how much coverage
of each category pair \(\{M_1, M_2\}\) we obtain by this neighborhood
selection. For each query \(S\), we calculate the proportion of pairs \(P_i/P_j\)
of the same categorization \(\{M_1, M_2\}\) such that both \(P_i\) and \(P_j\)
are included in the neighborhood. For the MIPS categorization, 93%
of the queries resulted in a coverage of at least 75% (24% of the
queries resulting in perfect coverage). Although filtering implies
that some valid pairs will never be ranked, the gain obtained by
reducing false positives in the top 10 ranked pairs is considerable
(results not shown) across all methods, and the computational gain
of reducing the search space is particularly relevant in exploratory
data analysis.

3.3 Analysis of signaling pathways (KEGG)
We repeated the same experimental setup, now using the KEGG
network to define the protein interactions. We selected proteins from

<table>
<thead>
<tr>
<th>Method</th>
<th>#AUC</th>
<th>#TOP10</th>
<th>#AUC.S</th>
<th>#TOP10.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>58</td>
<td>73</td>
<td>58</td>
<td>72</td>
</tr>
<tr>
<td>NNS</td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>MLS</td>
<td>26</td>
<td>55</td>
<td>13</td>
<td>38</td>
</tr>
<tr>
<td>RBSES</td>
<td>93</td>
<td>105</td>
<td>101</td>
<td>110</td>
</tr>
</tbody>
</table>

Columns #AUC, #TOP10, #AUC.S and #TOP10.S are defined as in Table 2.
A total of 6125 proteins was selected. The KEGG network is much

We presented a novel measure of similarity between biological

Distribution across all queries of the number hits in the top 10

Table 7. Distribution across all queries of the number hits in the top 10

Proportion of top hits using KEGG categories and links specified

The more skewed to the right, the better

4 CONCLUDING REMARKS

We presented a novel measure of similarity between biological

Bayesian analogical reasoning identifies latent functional linkage mechanisms

alologies that are functionally relevant among the top matches. We

This work can be expanded in many ways, including but not

Funding: National Science Foundation (under grants no. DMS-

Conflict of Interest: none declared.

REFERENCES

Adamcsek,B. et al. (2006) CFinder: locating cliques and overlapping modules in

Cognition


Nature

Nature Biotechnol.

Nature

Nature

Nature

Genet.

Bioinformatics

Nucleic Acids Res.

Science

Nature

Genet.

Nature

Nature

Nature

Nature

J. Mol. Biol.

Nat.


D281–D288.


Genet., 36, 559–564.

Gasch,A.P et al. (2000) Genomic expression programs in the response of yeast cells to


Gavin,A.-C. et al. (2006) Proteome survey reveals modularity of the yeast cell


263–297.

}\i381
SICD. Saccharomyces Genome Database. Available at http://www.yeastgenome.org (last accessed date January 3, 2011).