Predicting gene-disease associations using multiple species data

Nagarajan Natarajan[1]⋆ U. Martin Blom[2,3]⋆ Ambuj Tewari[1] John O. Woods[2]
Inderjit S. Dhillon[1] Edward M. Marcotte[2,4]

October 20, 2011

1. Department of Computer Science. University of Texas, Austin, Texas 78712, USA
2. Center for Systems and Synthetic Biology, Institute for Cellular and Molecular Biology, University of Texas, Austin, Texas 78712, USA
3. Program in Computational and Applied Mathematics, University of Texas, Austin, Texas 78712, USA
4. Department of Chemistry and Biochemistry. University of Texas, Austin, Texas 78712, USA
⋆ These authors contributed equally to this work.

1 Introduction

Correctly predicting genes associated with hereditary diseases is a first step to understanding the molecular mechanisms that lead to these diseases, and in the long run, to developing effective remedies for them. If a group of genes is already known to be associated with a disease, various “guilt-by-association” methods have shown themselves to be useful ways to find new candidate genes. In these, new candidate genes are predicted based on how many previously known disease genes they interact with.

Different kinds of interaction between genes can be used to determine their “guilt” for a given disease. The PRINCE method proposed by Vanunu et al. [27], RWRH method by Li & Patra [16], and the CIPHER tool proposed by Wu et al. [29] base their predictions on interactions from protein-protein interaction networks such as HPRD [26], and Goh et al. [5] construct a network where genes are connected if they are associated with the same disease.

One kind of network that has proven to be particularly useful for predicting biological function is the Bayesian functional interaction network, where a pair of genes is linked based on the integrated evidence from a wide array of information sources [14]. These have been used to associate genes with phenotypes in model organisms [20, 25], and a recently published network, HumanNet, has been used to refine predictions from genome-wide association studies [13]. Since functional gene interaction networks aggregate many different types of information, they can achieve much greater coverage than pure protein-protein interaction networks.

In the past decades, the growth of gene-phenotype associations in model species has been explosive, which suggests an alternative way to find candidate genes for human diseases. McGary et al.[21] used this treasure trove of information to find sometimes surprising connections between model species phenotypes and human diseases, by looking for pairs of human diseases and model phenotypes that share a higher than expected number of orthologous genes. In this way a number of new, and often surprising, model systems were found for human diseases. For instance, it seems like the human neural crest related developmental disorder Waardenburg syndrome shows some kind of functional similarity with gravitropism (the ability to detect what is up) in plants, and mammalian angiogenesis involves the same pathways as lovastatin sensitivity in yeast.

Both functional gene-gene interaction networks and the phenotypic information from distant species are valuable sources of information for predicting gene-disease interactions since they are *large-scale*, *independent* from each other and provide different *contexts* for the predictions, by showing which other genes they are functionally associated with, and which model organism phenotypes are exhibited when the gene is perturbed.

In this work, we combine the power of the functional gene-gene interaction networks with the phenotypic information from multiple species in a walk-based framework, and use a novel machine learning formulation called *PU learning* to infer the weights for different types of walks. We represent both genes and phenotypes as nodes in a graph, and connect two genes if they are functionally linked in HumanNet (weighted appropriately). Similarly, a gene is connected to a model organism phenotype if any of its orthologs in the model organism is known to be associated with the phenotype. In this way, the model organism phenotypes provide a new kind of links between genes, and we can leverage the independent information hidden in the model organism data.

We evaluate our methods on a number of diseases downloaded from the Online Mendelian Inheritance in Man (OMIM) project[1]. We demonstrate high recall for known diseases by cross-validation, and show that PU learning based methods outperform a state-of-the-art method that uses a similar walk-based framework.

2 Related Work

The problem of predicting gene-gene or gene-phenotype associations based on heterogneous sources of information has attracted the attention of the machine learning community recently. In particular, there is a whole line of work for predicting associations[16, 27, 6, 9, 29, 3, 2] based on the realization that genes and phenotypes can be modeled as nodes of a network, and thus graph-theoretic techniques can be brought to bear on the problem. One of the first systems that applied the idea of a random walk on the gene network to obtain a ranking of genes for a given phenotype was by Kohler et al. [9], where they treat the known gene-phenotype associations equally for restarting the walk. About the same time appeared CIPHER [29], another system that used integrated gene and phenotype networks and a linear regression model based on the idea that similar phenotypes arise from similar genes. However, the model does not effectively use the existing gene-phenotype associations. Later, Vanunu et al. [27] proposed a system PRINCE, which builds on CIPHER, and uses the similarities between phenotypes in order to obtain a "smoothed" restart vector. A different graph-based approach called Maximum-expectation Gene Cover was motivated by Karni et al. [6]: given a set of known gene-disease associations S, find a subset of a fixed size, such that the expected number of genes in the subset that are "close" to S is maximum (for a suitably defined notion of distance). More recently, Li and Patra [16] proposed a "heterogeneous" network model, incorporating gene-gene, gene-phenotype, and phenotype-phenotype networks. This model is closely related to our graph-based approach, as we will see in detail in Section 5.

It is important to emphasize here that none of the aforementioned graph-based approaches consider the problem in full generality, i.e. considering the *orthologous* gene relationships between species, and gene-phenotype networks of multiple species.

Modeling the link prediction as a supervised classification of pairs of nodes has been tried in the context of reconstructing gene networks. The notion of *PU learning* — learning from positive and unlabeled examples — was used in [3] for predicting gene-gene links, which employs a support vector machine classifier combined with Platt scaling [18] to obtain the conditional probability of being positive, given a pair of nodes. Here, gene expression profiles are used to obtain features.

While we were finishing the paper, we became aware of the work [23] that also uses the PU learning technique to train a classifier over gene-phenotype pairs. They do have multiple sources of information but the sources do not correspond to multiple species as is the case in our work. In [23], multiple sources are integrated using a multiple kernel learning (MKL) framework and information is shared across different diseases using a multi-task learning framework. In contrast, we integrate information from multiple species (and that from a gene-gene network) in a walk based framework. Our features are derived from various kinds of walks in a combined network consisting of all our information sources. In [23], their information sources (with one exception where a diffusion kernel is used) are not represented as a network. Instead, each source directly provides them with a vector of features. Another important difference is that we treat all diseases and phenotypes as part of a single task.

3 Preliminaries

Two keys ingredients in our approach will be: (i) similarity measures between nodes of a network, and (ii) PU learning approaches. In this section, we set up notation and briefly describe some ideas from the literature that we will later use.

3.1 Similarity Measures

Suppose we are given an undirected graph with a possibly weighted (symmetric) adjacency matrix A. The edge weight $A_{i,j}$ reflects the strength of the connection between node i and node j. A natural question to ask is: how similar are two given nodes in the graph? One way to do this is by counting paths of different length that connect i to j. This has a natural connection to matrix powers since $(A^l)_{i,j}$ is exactly the number of paths of length l that connect i to j. We would like to put together these numbers corresponding to various values of l into a single number summarizing the similarity between i and j. For example, we could choose any sequence β_l of non-negative coefficients or weights (not to be confused with the *edge* weights $A_{i,j}$) and define the similarity

$$S_{i,j} = \sum_{l \geq 1} \beta_l (A^l)_{i,j} \ .$$

The similarity matrix itself is even simpler to write thanks to matrix notation:

$$S = \sum_{l \geq 1} \beta_l A^l \ . \tag{1}$$

If the sum above is only taken over finitely many values of l then any choice for β_l leads to a well-defined similarity measure. However, if the sum is over all values of l, then we need to choose β_l such that they rapidly decay as l grows. Otherwise, the infinite sum may not converge. Following the survey article[4], we can think of the (scalar) function

$$F(z) = \sum_{l \geq 1} \beta_l z^l$$

and think of S as simply $F(A)$ where the matrix function F is defined through the series expansion in (1).

Specific choices for β_l give us different similarity measures. A popular choice is $\beta_l = \beta^l$ for small enough β. This leads to the *Katz* measure [7]:

$$S^{katz} = \sum_{l \geq 1} \beta^l A^l \ .$$

The choice of weights β_l may seem like an arbitrary one. Indeed, it would be nice if the coefficients could be learned based upon the network at hand. This is exactly the approach we will adopt.

We will rely on supervised machine learning methods to learn the weights β_l. But the supervision or feedback we have in our data sets is one-sided: we only have positive examples of genes actually associated with phenotypes. The majority of gene-phenotype pairs are unlabeled i.e. we do not know whether or not there is an association. Fortunately, we have at our disposal a suite of supervised machine learning methods that is meant to solve exactly the problem we are facing: PU learning, i.e. learning from only Positive and Unlabeled examples.

3.2 PU Learning

The possibility of applying PU learning in the context of link prediction in (social) networks was already hinted to in [17]:

> There also has been some potentially relevant work in machine learning on classification when the training set consists only of a relatively small set of positively labeled examples and a large set of unlabeled examples, with no labeled negative examples. It is an open question whether these techniques can be fruitfully applied to the link-prediction problem.

Our work (see also [23]) can also be thought of as paving the way for arriving at an affirmative answer to the open question mentioned above.

Most PU learning approaches start by somehow identifying a set of *reliable negatives* among the unlabeled points. Then these reliable negatives, along with the given positives, can be fed into a standard supervised learning algorithm.

One of the state-of-the-art classification algorithms is the support vector machine (SVM). One way of adapting SVMs to the PU learning framework ([19, 3, 2]) is to randomly draw a sample of unlabeled examples and label them negative. Then we train a *biased* SVM to find a hyperplane of maximum margin that separates the positives from *pseudo-negatives.* The SVM is biased in the sense that false negatives (in the training data) are penalized much more heavily than the false positives. The bias makes sense because the positive examples are known to be positive, while the negatives were arbitrary and hence false positives are not to be penalized too heavily. For a biased SVM written in the primal form, the optimal hyperplane is given by the normal vector θ that is solution of:

$$\begin{array}{rl} \displaystyle\min_{\theta\in\mathbb{R}^d} & \|\theta\|^2 + C_- \sum_{i:y_i=-1} \xi_i + C_+ \sum_{i:y_i=+1} \xi_i \\ \text{subject to} & \quad \forall i,\ \xi_i \geq 0 \text{ and } \langle \Phi(x_i), \theta \rangle \geq 1 - \xi_i\ , \end{array} \tag{2}$$

for some $C_+ \gg C_-$. Here $x_i \in \mathcal{X}$ are the training examples, $y_i \in \{\pm 1\}$ are their labels. The feature mapping $\Phi : \mathcal{X} \to \mathbb{R}^d$ can, in general, map to a reproducing kernel Hilbert space (RKHS) but in our case, it will simply map to $\mathbb{R}^d$. After solving for θ, we can classify a test example $x \in \mathcal{X}$ by taking the sign of the score $\langle \Phi(x), \theta \rangle$.

4 Problem Formulation

Let $\mathcal{G}$ denote the set of genes. Let $\mathcal{P}_i, 1 \leq i \leq s$ be disjoint sets representing phenotypes corresponding to s species. Denote the sizes of the sets $\mathcal{G}$ and $\mathcal{P}_i$ by n_G and n_i respectively. Let $\mathcal{P} = \cup_i \mathcal{P}_i$ be set of all phenotypes and denote its size by n_P. Thus, $n_P = \sum_i n_i$.

Our input consists of the gene-gene interaction matrix $G \in \mathbb{R}^{n_G \times n_G}$ along with the matrices $P_i \in \{0, 1\}^{n_G \times n_i}$ encoding the known associations in the s species. A non-zero (g, p) entry in P_i means that gene g is known to be associated with phenotype p. In addition, a *target* species t is specified. This means the species for which we want to predict new gene-phenotype associations.

The desired output is simply a *ranked* list of genes for each phenotype $p \in \mathcal{P}_t$ of the target species. For any given phenotype p, the ranking method should rank the n_g genes in order of their degree of association with p. The ranking should be such that genes on top are the ones with the most (predicted) association.

We visualize the input data as describing a graph with a core consisting of the gene-gene network. This core has the s gene-phenotype bipartite networks hanging off it (see Figure 1). In order to put together the matrices G, P_i coming from heterogeneous sources of information, we need some constants λ_G and $\lambda_i, 1 \leq i \leq s$ to determine their relative contributions. We can then put all the input data into a single *combined* network whose adjacency matrix is given by:

$$C = \begin{bmatrix} \lambda_G G & \lambda_1 P_1 & \lambda_2 P_2 & \cdots & \lambda_s P_s \\ \lambda_1 P_1^\top & 0 & 0 & \cdots & 0 \\ \lambda_2 P_2^\top & 0 & 0 & \cdots & 0 \\ \vdots & 0 & 0 & \ldots & 0 \\ \lambda_s P_s^\top & 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} \lambda_G G & P \\ P^\top & 0 \end{bmatrix}\ , \tag{3}$$

where $P = [\lambda_1 P_1 \cdots \lambda_s P_s]$ is an $n_G \times n_P$ matrix. The matrix C is symmetric, as G is symmetric. Note that some or all of the zero matrices can be replaced with a network among the corresponding phenotypes, if available. In our experiments, we do not use any such network among phenotypes. Researchers have considered heterogeneous networks in similar settings ([29, 9, 6, 16]). However, none of these approaches

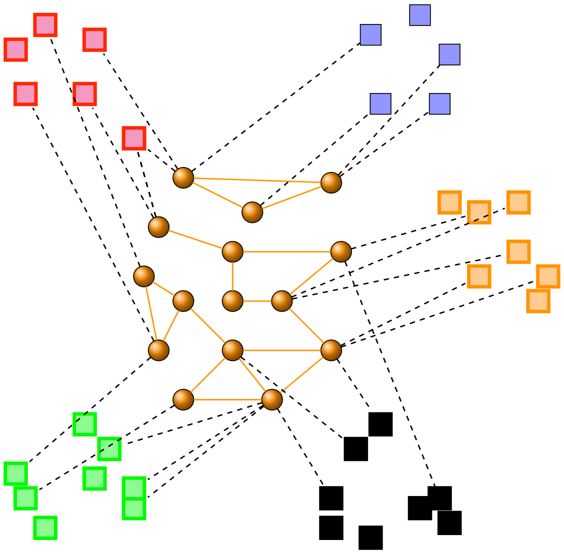


Figure 1: Combined network consisting of interactions among genes, and bipartite graphs between genes and phenotypes of different species. The square nodes represent phenotypes, while the circular nodes represent genes.

include multiple species phenotypes as we do. In fact, setting $\lambda_i = 0, i \neq t$ and $\lambda_t = \lambda_G = 1$, we get the heterogeneous network construction used in ([16]).

We can now pose the problem of predicting relevant associations in the target species t as a link prediction problem in the combined network C, focussing our attention on the links in P_t, i.e, evaluating the proposed methods only on the target phenotypes $\mathcal{P}_t$.

5 Methodology

Imagine using any of the popular ways to compute node-node similarity on the matrix C above. This will result in a $n_G \times n_P$ matrix which we can write in block form:

$$S = \begin{bmatrix} S_{GG} & S_{GP} \\ S_{PG} & S_{PP} \end{bmatrix} . \tag{4}$$

Most similarity measures are symmetric so we usually have $S_{PG} = S_{GP}^\top$. Some existing approaches rely on the simple but useful observation that the block S_{GP} can be used to rank the genes for any phenotype in $\mathcal{P}$ and therefore, in particular, for phenotypes in $\mathcal{P}_t$. The ranked list of genes can be obtained by simply considering the column of S_{GP} corresponding to a phenotype p of interest and then sorting the similarity values in decreasing order.

Let us now examine the Katz and Random Walk based similarity approaches in some more detail focusing on the particular form of our combined matrix C.

5.1 Katz

The general formula for the Katz similarity measure when specialized to the combined matrix written in block form gives:

$$S^{katz} = \begin{bmatrix} S^{katz}_{GG} & S^{katz}_{GP} \\ (S^{katz}_{GP})^\top & S^{katz}_{PP} \end{bmatrix} ,$$

where

$$S^{katz}_{GP} = (I - \beta\lambda_G G)^{-1} \cdot P \cdot \left[I - P^\top (I - \beta\lambda_G G)^{-1} P\right]^{-1} .$$

Note how the Katz similarity matrix for $\lambda_G G$ itself appears in the expression above. Let us set $S^{katz}_G = (I - \beta\lambda_G G)^{-1}$ to write the above as:

$$S^{katz}_{GP} = S^{katz}_G P \left[I - P^\top S^{katz}_G P\right]^{-1} .$$

Note that S^{katz}_G depends only on G and is different from S^{katz}_{GG}, the latter being the top-left block in the Katz similarity matrix of the *combined network*. This takes into account all kinds of paths in the combined network that start in $\mathcal{G}$ and end up in $\mathcal{P}$. Thus, it consists of an infinite sum that consists of (weighted) terms of the general form:

$$G^{i_1} \cdot (PP^\top)^{i_2} \cdot G^{i_3} \cdot (PP^\top)^{i_4} \cdots G^{i_{2k+1}} \cdot P ,$$

for $k \geq 0$ and $i_1, i_2, \ldots, i_{2k+1} \geq 0$.

5.2 Random Walk with Restarts

Another approach to incorporate heterogenous sources of information [16] is based on random walk model with restarts (RWR). The basic idea is to use the known genes for a phenotype as *seed nodes* to initialize a random walk in the combined network. At any time, the random walker either jumps to a neighboring node in the combined network or goes back to one of the seed nodes (hence the name "random walk with restarts"). Note that the setting in [16] corresponds to having gene-phenotype data for only one species. On the other hand, they also consider a phenotype-phenotype network obtained by mining text data about phenotypes. It is easy enough to extend their approach to handle multiple species. Similarly, as we have already noted above, it is straightforward to extend our approach to handle phenotype-phenotype interaction data, if available. Thus, for the purpose of comparison and discussion we consider both the Katz and RWR approaches in the setting of the current paper: namely, gene-phenotype data from multiple species but no phenotype-phenotype data.

To arrive at a mathematical description of the RWR approach, let us look at the combined network again:

$$C = \begin{bmatrix} \lambda_G G & \lambda_1 P_1 & \lambda_2 P_2 & \cdots & \lambda_s P_s \\ \lambda_1 P_1^\top & 0 & 0 & \cdots & 0 \\ \lambda_2 P_2^\top & 0 & 0 & \cdots & 0 \\ \vdots & 0 & 0 & \ldots & 0 \\ \lambda_s P_s^\top & 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} \lambda_G G & P \\ P^\top & 0 \end{bmatrix} ,$$

Since we want to simulate a random walk, let us try to make this matrix (column) stochastic. This can be done in several ways. Assuming that G has non-negative entries, perhaps the simplest way is to divide each column by its ℓ_1-norm (sum of absolute values of entries) resulting in a column stochastic matrix.

Another way, more along the lines of [16] would be do normalize differently in the first n_G columns than the rest. Let

$$C_{:,g} = \begin{bmatrix} \lambda_G G_{:,g} \\ P^\top_{g,:} \end{bmatrix}$$

be one of the first n_G columns. We can normalize the top n_G entries and the bottom n_P separately and then take a weighted average of the two (with weighting given by yet another parameter $\lambda \in (0, 1)$):

$$C^N_{:,g} = \begin{bmatrix} \lambda \frac{G_{:,g}}{\|G_{:,g}\|_1} \\ (1-\lambda) \frac{P^\top_{g,:}}{\|P_{g,:}\|_1} \end{bmatrix}$$

with the understanding that if a gene in not known to be associated to any phenotype (i.e. $\|P_{g,:}\|_1 = 0$), then we will simply use $\lambda = 1$ for that gene. The rest of the n_P columns can be normalized in the usual way by their ℓ_1-norm.

In a nutshell, starting from the symmetric matrix C of the combined network, we arrive at a column normalized matrix C^N that will, in general, not be symmetric. Then we consider the evolution:

$$v_{T+1} = \beta C^N v_T + (1-\beta) P^N_{:,p}$$

where $P^N_{:,p}$ is simply a probability distribution with equal mass on all genes known to be associated with a phenotype p of interest. The genes are then ranked in the order of the mass that is assigned to them under the steady state distribution v_∞ of this evolution.

5.3 Relationship between Katz and Random Walks with Restarts

It is easily seen that the steady state vector v_∞ should satisfy

$$v_\infty = \beta C^N v_\infty + (1-\beta) P^N_{:,p}$$

which readily yields

$$v_\infty = (1-\beta)[I - \beta C^N]^{-1} P^N_{:,p} \ .$$

Since $P^N_{:,p}$ is a column of C^N and rankings do not depend on constant factors such as $\beta/(1-\beta)$, we see that this method ranks genes by considering the top-right $n_G \times n_P$ sized block of the matrix

$$\beta[I - \beta C^N]^{-1} C^N = \beta C^N + \beta^2 (C^N)^2 + \beta^3 (C^N)^3 + \dots \ .$$

which is *exactly* Katz but on the *normalized* matrix C^N instead of C itself.

5.4 A supervised approach to link prediction problem

The arbitrariness in the choices of parameters involved in the Katz and random walk based approaches (extended to the combined network C) is unsettling. To get rid of the arbitrariness, we would like to learn the weights based on the network itself. To this end, we conceive the problem of predicting potential gene-phenotype associations as a supervised learning problem, where we want to learn a classifier function whose input space consists of gene-phenotype pairs. In particular, by appropriately defining a feature space for the gene-phenotype pairs, we will see that learning a classifier in the constructed space also achieves learning coefficients for a truncated version of S^{katz}_{GP}.

Recall the following two key characteristics of our data set:

1. We do not know of any “negative associations”. For each phenotype, we have only partially observed associated genes.

2. There are a large number of unlabeled gene-phenotype pairs, with the prior knowledge that most of them are negative.

PU learning methods are thus natural for our setting, and we learn a *biased* SVM classifier using the positive and a sample of unlabeled links. Note that, though, we could in principle use any of the PU learning methods like [15].

5.5 Features Derived from Hybrid Walks

The embedding of gene-phenotype links is specified by the function Φ in equation (2). Recall that in the Katz measure, the weights for combining the contributions from walks of different lengths is fixed beforehand. Expanding the first few terms of the series S^{katz}_{GP}, we have

$$S^{katz}_{GP} = \beta P + \beta^2 (G^2 + PP^\top) P + \beta^3 (G^3 + GPP^\top + PP^\top G) P + \dots \tag{5}$$

We observe from (5) that, for a given degree (or length) of walk, there are multiple ways of obtaining hybrid walks, as given by the terms in the series. For a given gene-phenotype pair, different walks of the same degree, and walks of different degrees can be used as features for the pair. Thus learning a biased SVM provides an efficient way to learn the weights, and could help improve on the prediction performance of a particular choice of weights $(\beta, \beta^2, \beta^3, \dots)$.

More generally, the terms in the series (5) are the matrices in the expansion of the polynomial $(I + G + PP^\top)^k P$. Clearly, the dimensionality of the feature space $\mathcal{F}$ is exponential in k, the degree of the matrix polynomial, and make us vulnerable to the the curse of dimensionality because the examples are limited. However, taking cue from the fact that the weights of increasing walk lengths need to be heavily damped, we ignore higher order terms and thereby contain the dimensionality of the feature space d to a small constant.

Species-wise features Notice that $PP^\top = \sum_{i=1}^{s} P_i P_i^\top$. We can further decompose the features to species level, which enables learning the contribution of each species to the prediction. Thus for a gene-gene link (g, g'), we have a feature $(P_i P_i^T)_{g,g'}$ for each species i, in addition to other G-only and hybrid features from the expansion of the matrix polynomial. We will see later in the experiments that using species-wise features not only lends interpretability but also improves the accuracy of the predictions, as compared to combining features of same degree.

5.6 Algorithm

A very recent work by Mordelet et al. [24] uses bagging technique to obtain an aggregate classifier in both transductive and inductive settings, using positive and unlabeled examples. Our problem is a typical transductive setting, i.e. the universal set of gene-phenotype pairs which must be labeled are available to the algorithm during training phase. It is natural to do bagging in the PU learning step to reduce the variance in the classifier that is induced due to arbitrary labeling of a random sample as negative. Let T be the number of bootstraps. Let n_+ denote the number of positive examples in the training data. Different classifiers can be combined using a bagging algorithm as follows:

```
initialize $f(x) = 0$ and $n(x) = 1, \forall x \in \mathcal{U}$.
for $i = 1, 2, \dots, T$:
    1. Draw a bootstrap sample $\mathcal{U}_t \subseteq \mathcal{U}$ of size $n_+$.
    2. Train a classifier $f_t$ using the positive training examples and $\mathcal{U}_t$ as negative examples.
    3. For any $x \in \mathcal{U} - \mathcal{U}_t$ update: $f(x) \leftarrow f(x) + f_t(x)$ and $n(x) \leftarrow n(x) + 1$.
return $s(x) = f(x)/n(x)$, $\forall x \in \mathcal{U}$.
```

We use biased SVM as the classifier in step 2 for all the iterations, and the scoring function $f_t(x)$ for the tth iteration is simply the distance of the point x from the hyperplane and is given by the standard dot product,

$$f_t(x) = \langle \theta_t, \Phi(x) \rangle$$

where θ_t is the normal to the hyperplane learnt using the random bootstrap in the tth iteration. In contrast to the traditional SVM classifiers that classify a pair as positive or negative based on the sign of $\langle \theta_t, \Phi(x) \rangle$, we use the value as a score under the assumption that the further a point is on the positive side of the hyperplane, the more likely it is to be a true positive.

We would also like to note that the biased SVM framework enables us to classify all gene-phenotype pairs with a single training phase, thereby making the best use of the relation between different phenotypes, and use the scoring function $s(x)$ to make predictions for every phenotype during the test phase.

6 Experiments

We use a 3-fold testing methodology, i.e. we split the known associations in the target species t into 3 mutually exclusive and exhaustive subsets, and evaluate different methods by training with two of the three subsets, and testing on the hold-out set. We hold each subset once, and report the results averaged over the 3 runs. We use two types of performance measures:

1. The average AUC (area under ROC curve) per phenotype. It is the average AUC for predicting the held-out test genes for a particular phenotype. For each phenotype, this quantity is calculated for each of its held-out sets of known genes, and averaged over the three splits.

2. The average rank per phenotype. By this we mean the average rank (over 3 splits) a test gene for the phenotype obtains when the gene was in a hold-out set.

Data set

The data sets for the experiments are collected from multiple sources including [11, 12, 8, 10]. Detailed description on the extraction of the data sets can be found in [22]. In particular, when a human gene is known to occur as multiple *orthologs* in a species, a single gene that is representative of the entire set of orthologs is retained. The gene-gene interaction network is obtained from sources independent of gene-phenotype networks (for its construction, see [13]). We have gene-phenotype associations collected from literature for nine different species: human (*Home sapiens*), plant (*Arabidopsis thaliana*), worm (*Caenorhabditis elegans*), fly (*Drosophila melanogaster*), mouse (*Mus musculus*), yeast (*Saccharomyces cerevisiae*), *Escherichia coli*, zebrafish (*Danio rerio*) and chicken (*Gallus gallus*). The human phenotypes of interest are human diseases from the OMIM database [1]. The sizes of the networks are shown in Table 1.

Index	Species	Acronym	Phenotypes	# Associations
1	Human	*Hs*	1435	3571
2	Plant	*At*	1137	12010
3	Worm	*Ce*	744	30519
4	Fly	*Dm*	2503	68525
5	Mouse	*Mm*	4662	75199
6	Yeast	*Sc*	1243	73284
7	Zebrafish	*Dr*	1143	4500
8	E.coli	*Ec*	324	72846
9	Chicken	*Gg*	1188	22150

Table 1: Different species used for inferring gene-phenotype associations in the combined network model, and sizes of the gene-phenotype networks for the species, restricted to orthologs of human genes. The total number of genes is 12231.

Walk-based methods

The two primary walk-based methods that are evaluated are the random walk with restart (RWR) method proposed in Li and Patra [16], and Katz on the combined graph C. The parameters that affect Katz scores are $k, \beta, \lambda_G, \lambda_i, 1 \leq i \leq 6$. Here k is the maximum length of walks considered. Estimating a set of (locally) optimal parameters is computationally expensive. We use the truncated Katz version (up to the third degree in (5)) for the experiments, and we set $k = 3$. We find that the quality of the predictions get better from $k = 2$ to $k = 3$, and for higher values of k the performance improvement is negligible. Even for $k = 3$, the number of hybrid paths between a gene and a phenotype can be really high (up to about 10^6). Typically, β is chosen small enough that the contribution from the higher order paths is made negligible. We set $\beta = 10^{-6}$ which we have found to work reasonably well on networks of similar size and sparsity [28]. Also, we weigh all the heterogeneous edges equally, i.e. $\lambda_G = 1$ and $\lambda_i = 1$ for all i. We compare our method to two versions of RWR method in [16]. First, the method as is from the paper, i.e. using only G and P_t networks, where t is the target species. For all our experiments, we set $t = Hs$. Second, we extend the RWR method to multiple species. The results are presented in Figure 2. One immediate observation is that using information from multiple other species is helpful for prediction of associations in the target species. While extended version of RWR does well on the average over all phenotypes, Katz on the combined network retrieves genes in top-100 for more phenotypes than the other two methods. This is interesting and arguably more important to the biologists, since the concern here is predicting with high precision, when the number of predictions required are small (of order 100).

Supervised methods

In this section, we present the results of biased SVM based methods and compare them with walk-based methods. As for the biased SVM method, we study two types of feature construction: a) Constructing features from hybrid walks, combining phenotypes of all species together, and b) Constructing features from phenotypes of individual species (and walks involving those phenotypes). The results are presented in Figure 3. We observe that the biased SVM variant that uses features that distinguish species perform the best by all measures of performance. The PU learning based methods in Figure 3 perform better than the walk-based methods like Katz and random walk with restarts.

Penalty parameters In equation (2), $C_+ \gg C_-$ are the penalties on misclassified positives and negatives respectively. The weights control the relative widths of the margins on either sides of the hyperplane. As C_+ increases from 0 to ∞, the margin on the side of the positive examples shrinks, and as $C_+ \to \infty$, the classifier makes *no* mistake on the positive examples. The ratio C_+/C_- determines the "weight" of a positive example, and we want this to be a very high value. In our experiments, we set $C_- = 1$ and $C_+ = 10^5$, which is found to be the best by cross-validation[1].

7 Conclusion and Future work

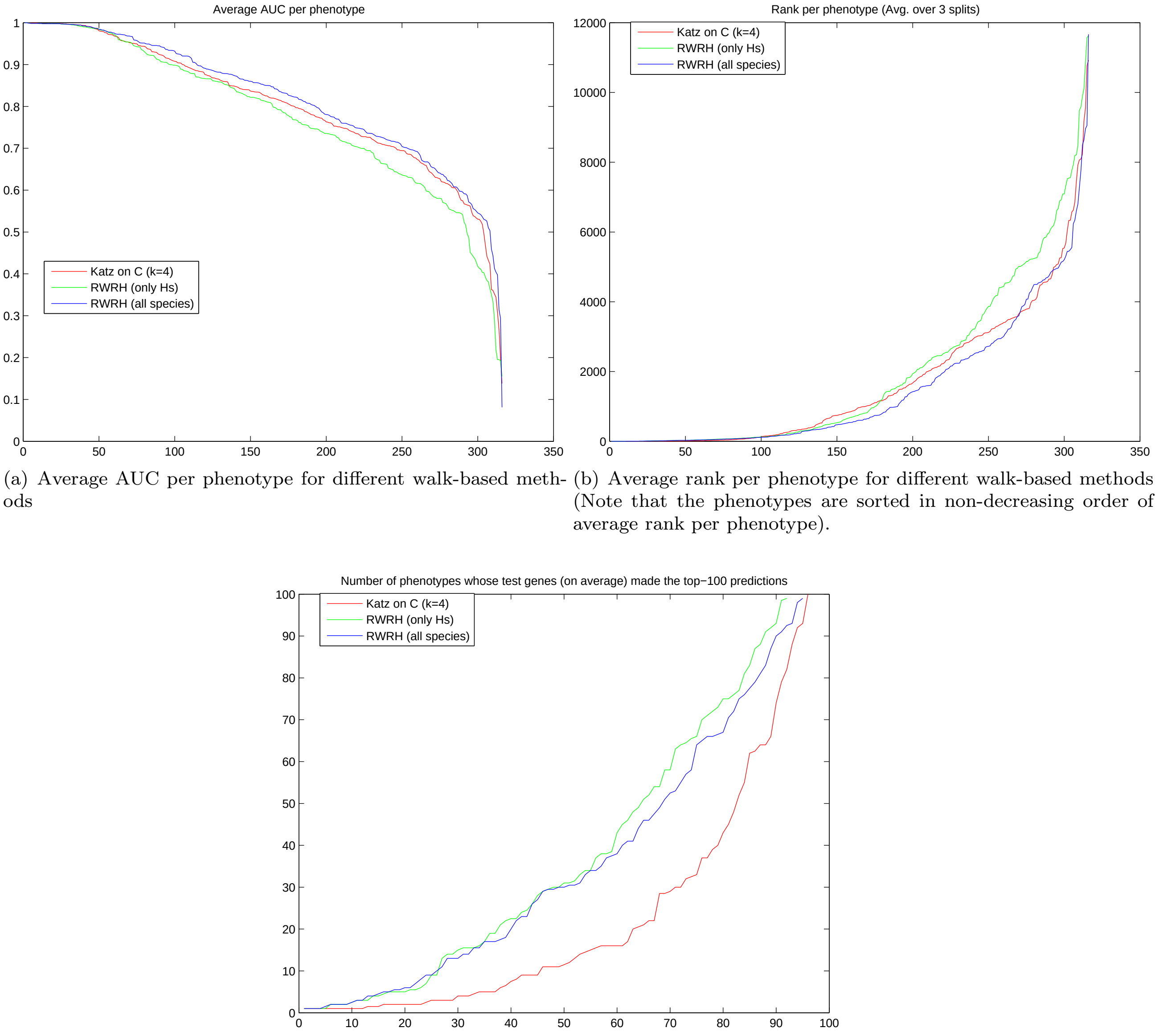
We have shown that observations on phenotypes of multiple species can be exploited through orthologous relationships to human phenotypes, to prioritize genes for human phenotypes. We have also generalized walk-based methods to include multiple species, and proposed PU learning based solution to learn the weights for walks of different lengths. Evaluation on OMIM human phenotypes show that the learnt weights give the best recall rates, and outperform random-walk based methods that use fixed weights. While k-fold validation based evaluation is good for comparisons, the real test of the proposed method is biological validation of the recommendations. We are currently pursuing this direction. We are also planning to make the system publicly accessible through a website, where biologists and other researchers can submit known associations for a phenotype, and query the system for recommendations.

As we mentioned in the related works section, the recent work [23] uses the multitask approach in machine learning where different diseases or phenotypes are treated as different learning tasks which may or may not be coupled to each other. In contrast, we treat all the gene-phenotypes as being part of a single task. It will be interesting to study if there is an optimal operating point between the two extremes: separate tasks for each phenotype and a single task for all phenotypes.

References

[1] Online Mendelian Inheritance in Man, OMIM. *McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University (Baltimore, MD).* `http://omim.org/`.

[1] Fixing $C_- = 1$, $\log_{10} C_+$ was varied in the range $1, 2, \ldots, 10$

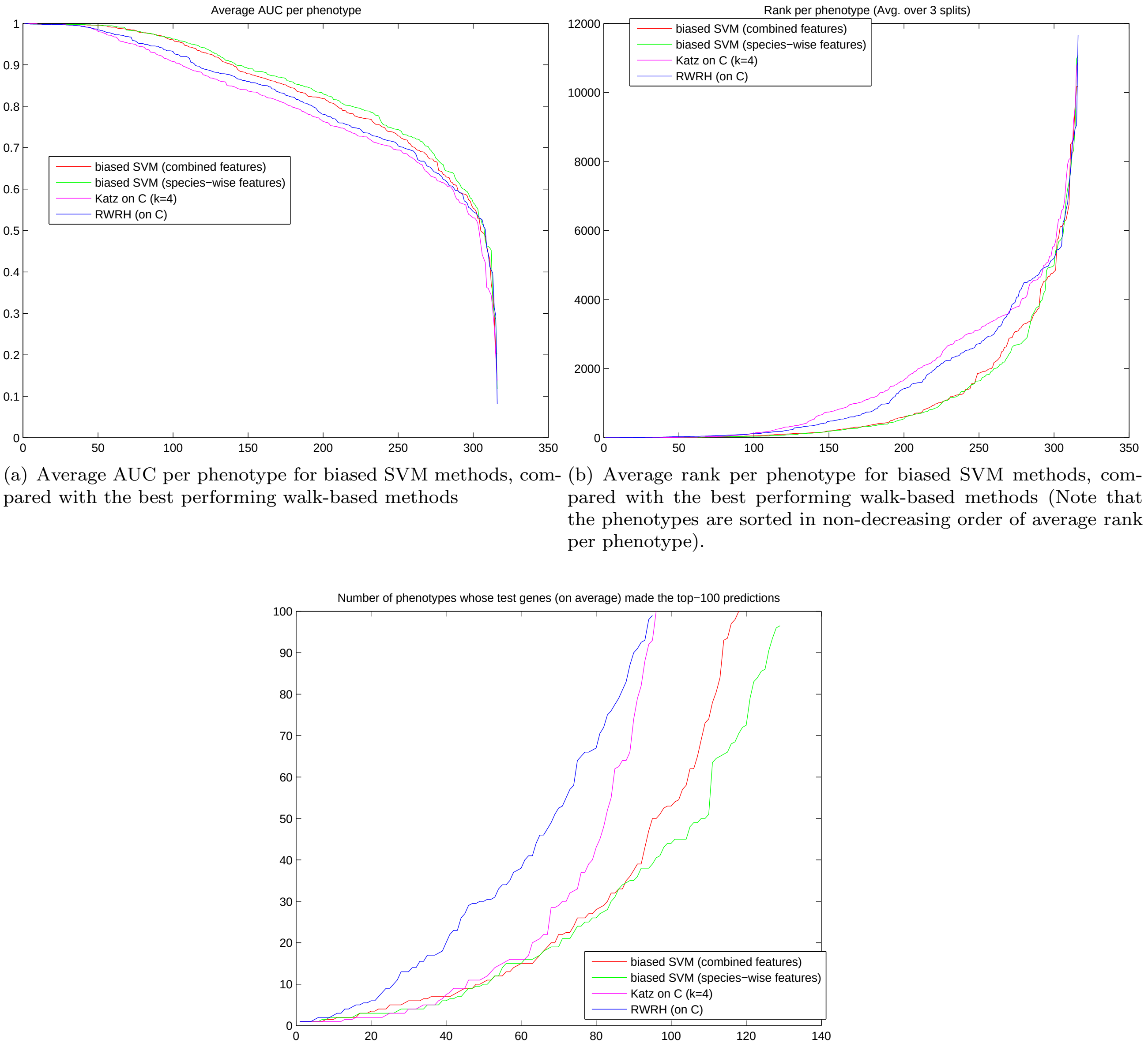


(a) Average AUC per phenotype for different walk-based methods

(b) Average rank per phenotype for different walk-based methods (Note that the phenotypes are sorted in non-decreasing order of average rank per phenotype).

(c) Number of phenotypes whose test genes are identified in the top k predictions, $k = 1, 2, \ldots, 100$ (Note that the phenotypes are sorted in non-decreasing order of average rank per phenotype).

Figure 2: Evaluation of different walk-based methods, with *Hs* as the target species. RWRH (only Hs) is the method in Li and Patra [16]. RWRH (all species) is the extension of the random walk with restart method to include all species. See Section 5.2 for details. All the results are averaged over 3-fold splits.



(a) Average AUC per phenotype for biased SVM methods, compared with the best performing walk-based methods

(b) Average rank per phenotype for biased SVM methods, compared with the best performing walk-based methods (Note that the phenotypes are sorted in non-decreasing order of average rank per phenotype).

(c) Number of phenotypes whose test genes are identified in the top k predictions, $k = 1, 2, \ldots, 100$ (Note that the phenotypes are sorted in non-decreasing order of average rank per phenotype).

Figure 3: Evaluation of biased SVM methods, with *Hs* as the target species. RWRH (all species) is the extension of the random walk with restart method in Li and Patra [16] to include all species (This is the best performing walk-based method as observed from Figure 2). All the results are averaged over 3-fold splits.

[2] Nitin Bhardwaj, Mark Gerstein, and Hui Lu. Genome-wide sequence-based prediction of peripheral proteins using a novel semi-supervised learning technique. *BMC Bioinformatics*, 11, 2010.

[3] Luigi Cerulo, Charles Elkan, and Michele Ceccarelli. Learning gene regulatory networks from only positive and unlabeled data. *BMC Bioinformatics*, 11, 2010.

[4] Ernesto Estrada and Desmond J. Higham. Network properties revealed through matrix functions. *SIAM Rev.*, 52:696–714, November 2010.

[5] K.I. Goh, M.E. Cusick, David Valle, Barton Childs, Marc Vidal, and A.L. Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685, 2007.

[6] Shaul Karni, Hermona Soreq, and Roded Sharan. A network-based method for predicting disease-causing genes. *Journal of Computational Biology*, 16:181–189, 2009.

[7] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43, 1953.

[8] W. K. Kim, C. Krumpelman, and E. M. Marcotte. Inferring mouse gene functions from genomic-scale data using a combined functional network/classification strategy. *Genome Biol.*, 9 Suppl 1:S5, 2008.

[9] Sebastian Kohler, Sebastian Bauer, Denise Horn, and Peter N Robinson. Walking the interactome for prioritization of candidate disease genes. *The American Journal of Human Genetics*, 82(4):949–958, 2008.

[10] I. Lee, B. Ambaru, P. Thakkar, E. M. Marcotte, and S. Y. Rhee. Rational association of genes with traits using a genome-scale gene network for Arabidopsis thaliana. *Nat. Biotechnol.*, 28:149–156, Feb 2010.

[11] I. Lee, B. Lehner, C. Crombie, W. Wong, A. G. Fraser, and E. M. Marcotte. A single gene network accurately predicts phenotypic effects of gene perturbation in Caenorhabditis elegans. *Nat. Genet.*, 40:181–188, Feb 2008.

[12] I. Lee, Z. Li, and E. M. Marcotte. An improved, bias-reduced probabilistic functional gene network of baker's yeast, Saccharomyces cerevisiae. *PLoS ONE*, 2:e988, 2007.

[13] Insuk Lee, U Martin Blom, Peggy I Wang, Jung Eun Shim, and Edward M Marcotte. Prioritizing candidate disease genes by network-based boosting of genome-wide association data. *Genome research*, 21(7):1109–21, July 2011.

[14] Insuk Lee, Shailesh V Date, Alex T Adai, and Edward M Marcotte. A probabilistic functional network of yeast genes. *Science (New York, N.Y.)*, 306(5701):1555–8, November 2004.

[15] Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, 2003.

[16] Yongjin Li and Jagdish Chandra Patra. Genome-wide inferring gene-phenotype relationship by walking on the heterogeneous network. *Bioinformatics/computer Applications in The Biosciences*, 26:1219–1224, 2010.

[17] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, New York, NY, USA, 2003.

[18] Hsuan-Tien Lin, Chih-Jen Lin, and Ruby Weng. A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, 68:267–276, 2007. 10.1007/s10994-007-5018-6.

[19] Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 179–, Washington, DC, USA, 2003. IEEE Computer Society.

[20] Kriston L McGary, Insuk Lee, and Edward M Marcotte. Broad network-based predictability of Saccharomyces cerevisiae gene loss-of-function phenotypes. *Genome biology*, 8(12):R258, January 2007.

[21] Kriston L McGary, Tae Joo Park, John O Woods, Hye Ji Cha, John B Wallingford, and Edward M Marcotte. Systematic discovery of nonobvious human disease models through orthologous phenotypes. *Proceedings of the National Academy of Sciences of the United States of America*, 107(14):6544–9, April 2010.

[22] Kriston L. McGary, Tae Joo Park, John O. Woods, Hye Ji Cha, John B. Wallingford, and Edward M. Marcotte. Systematic discovery of nonobvious human disease models through orthologous phenotypes. si materials and methods (supporting information). In *PNAS*, volume 107, 2010.

[23] F. Mordelet and J.-P. Vert. Prodige: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC Bioinformatics*, 12(389), October 2011.

[24] Fantine Mordelet and Jean-Philippe Vert. A bagging SVM to learn from positive and unlabeled examples. Technical Report hal-00523336, version 1, HAL, 2010.

[25] Sara Mostafavi, Debajyoti Ray, David Warde-Farley, Chris Grouios, and Quaid Morris. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome biology*, 9 Suppl 1:S4, January 2008.

[26] Suraj Peri, J Daniel Navarro, Ramars Amanchy, Troels Z Kristiansen, Chandra Kiran Jonnalagadda, Vineeth Surendranath, Vidya Niranjan, Babylakshmi Muthusamy, T K B Gandhi, Mads Gronborg, Nieves Ibarrola, Nandan Deshpande, K Shanker, H N Shivashankar, B P Rashmi, M a Ramya, Zhixing Zhao, K N Chandrika, N Padma, H C Harsha, a J Yatish, M P Kavitha, Minal Menezes, Dipanwita Roy Choudhury, Shubha Suresh, Neelanjana Ghosh, R Saravana, Sreenath Chandran, Subhalakshmi Krishna, Mary Joy, Sanjeev K Anand, V Madavan, Ansamma Joseph, Guang W Wong, William P Schiemann, Stefan N Constantinescu, Lily Huang, Roya Khosravi-Far, Hanno Steen, Muneesh Tewari, Saghi Ghaffari, Gerard C Blobe, Chi V Dang, Joe G N Garcia, Jonathan Pevsner, Ole N Jensen, Peter Roepstorff, Krishna S Deshpande, Arul M Chinnaiyan, Ada Hamosh, Aravinda Chakravarti, and Akhilesh Pandey. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome research*, 13(10):2363–71, October 2003.

[27] Oron Vanunu, Oded Magger, Eytan Ruppin, Tomer Shlomi, and Roded Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS Comput Biol*, 6, January 2010.

[28] Vishvas Vasuki, Nagarajan Natarajan, Zhengdong Lu, and Inderjit.S Dhillon. Affiliation recommendations using auxiliary friendship networks. In *RecSys '10: Proceedings of the third ACM conference on Recommender systems*, Barcelona, Spain, 2010.

[29] Xuebing Wu, Rui Jiang, Michael Q Zhang, and Shao Li. Network-based global inference of human disease genes. *Mol Syst Biol*, 4:189, 2008.