Fast integration of heterogeneous data sources for predicting
gene function with limited annotation

Sara Mostafavi1,2,* and Quaid Morris1,2,*
1Department of Computer Science and 2Center for Cellular and Biomolecular Research,
University of Toronto, Canada
Associate Editor: Jonathan Wren

ABSTRACT
Motivation: Many algorithms that integrate multiple functional
association networks for predicting gene function construct a
composite network as a weighted sum of the individual networks and
then use the composite network to predict gene function. The weight
assigned to an individual network represents the usefulness of that
network in predicting a given gene function. However, because many
categories of gene function have a small number of annotations, the
process of assigning these network weights is prone to overfitting.

Results: Here, we address this problem by proposing a novel
approach to combining multiple functional association networks.
In particular, we present a method where network weights are
simultaneously optimized on sets of related function categories. The
method is simpler and faster than existing approaches. Further, we
show that it produces composite networks with improved function
prediction accuracy using five example species (yeast, mouse, fly,
Escherichia coli and human).

Availability: Networks and code are available from:
http://morrislab.med.utoronto.ca/~sara/SW

Contact: smostafavi@cs.toronto.edu; quaid.morris@utoronto.ca

Supplementary information: Supplementary data are available at
Bioinformatics online.

Received on May 3, 2010; revised on April 25, 2010; accepted on
May 16, 2010

1 INTRODUCTION
The past decade has seen a dramatic increase in the quantity and
variety of publicly available genomic and proteomic data, and
a parallel increase in the number of computational methods to
integrate these heterogeneous data in generating predictions about
protein and gene function [see Noble and Ben-Hur (2007) for a review].
Many of these methods, often called gene (or protein) function prediction algorithms, use the same basic framework:
first, they generate so-called functional association networks that
capture information about shared gene (or protein) function implicit
in each dataset, then they integrate these networks to generate a
single composite network which they input, along with a set of
labels that describe gene function, to a kernel- or network-based
classification algorithm (e.g. Lanckriet et al., 2004; Marcotte et al.,
1999; Mostafavi et al., 2008; Myers and Troyanskaya, 2007; Tsuda
et al., 2005). Once trained, these classification algorithms assign
discriminant values to each gene that can then be thresholded to
generate hypotheses about the function of unlabeled genes.

The functional association network is a natural and widely
used representation for capturing information about shared gene
function from high-throughput data sources. In this representation,
nodes correspond to genes or proteins and the edges are weighted
according to the evidence implied by a given data source for
shared function of the connected nodes. These edge weights are
calculated using a similarity metric matched to a given data type; for
example, the Pearson’s correlation coefficient (PCC) is often used
to measure pairwise similarities between gene expression profiles.
Once calculated, it is relatively easy to translate these networks into
kernels for kernel-based learning methods [e.g. by using a diffusion
kernel (Kondor and Lafferty, 2002; Qi et al., 2008)].

An important step in predicting gene function is the construction
of a composite network from multiple functional association
networks. A common approach is to construct a function-specific
composite network as a weighted sum of the individual networks
such that the weight of each network is determined based on the
network’s predictiveness of a set of positively labeled genes that
are deemed to have the same specific function (Lanckriet et al.,
2004; Mostafavi et al., 2008; Tsuda et al., 2005). The positive gene
labels are derived from online databases such as Gene Ontology
(GO; Ashburner et al., 2000), KEGG (Kanehisa and Goto, 2000) and
Enzyme Commission (EC; Bairoch, 2000). These databases provide
a controlled vocabulary describing categories of gene function and
curated lists of genes annotated to these functions.

There are two challenges in constructing function-specific
composite networks. First, because many functional categories have
only a few annotations, it is difficult to assign network weights
without overfitting. Second, for an algorithm to be widely applicable
it must be fast and scalable to combine dozens of networks with over
10 000 nodes (genes) each.

Here, we investigate a number of network weighting schemes to
avoid overfitting. In particular, we propose a new approach that we
refer to as Simultaneous Weights (SWs). SW is based on our previous
algorithm, GeneMANIA (Mostafavi et al., 2008), which constructs
function-specific composite network by solving a constrained linear
regression problem. However, instead of assigning function-specific
network weights, we simultaneously optimize the weights on a group
of related function categories by solving a single-constrained linear
regression problem. We evaluate the impact of several regularization
schemes such as LASSO (Tibshirani, 1996), elastic net (Zou and
Hastie, 2005), ridge regularization on our previous weighting
scheme (Mostafavi et al., 2008) and SW. Compared with other
2 RELATED WORK

There are large number of algorithms that extend simple guilt-by-association when predicting gene function from a single network including (Karaoz et al., 2003; Nabieva et al., 2005; Vazquez et al., 2003). The approaches closest to those presented in this article are methods for integrating multiple functional association networks into one composite network with the goal of predicting gene function from the composite network. In the seminal work of Marcotte et al. (1999), a composite network is constructed as an unweighted sum of several functional association networks, each derived from a different data source. More recently, in Lanckriet et al. (2004) and Tsuda et al. (2005), the network weights are assigned to optimize the performance of support vector machine (SVM) and Gaussian random fields (GRFs), respectively, which use the composite network to predict gene function. In Mostafavi et al. (2008), we use linear regression to optimize an objective function involving the kernel target alignment (Cristianini et al., 2002) of the composite network and the class labels. Another method for combining multiple association network was presented in Myers and Troyanskaya (2007) where a combined network was constructed using a naive Bayes classifier.

The new approach that we present here, SW, extends GeneMANIA algorithm (Mostafavi et al., 2008) that was previously shown to have the state-of-art performance on yeast and mouse benchmark datasets (Mostafavi et al., 2008; Pena-Castillo et al., 2008). However, achieving good performance with the GeneMANIA algorithm in categories with a small number of annotations required a time-consuming regularization procedure. Here, we investigate how to improve the performance for function categories with few annotations without increasing computation time.

3 ALGORITHM

Following the framework of Mostafavi et al. (2008), our approach for predicting gene function from multiple networks consists of two steps: (i) it constructs a composite network from multiple functional association networks and (ii) it predicts gene function from a single composite network. Below, we first review the constrained linear regression problem solved by the GeneMANIA algorithm for assigning network weights; next we describe SW, our new approach for assigning network weights using related categories of gene function. Finally, we briefly review how gene function is predicted from a single composite network.

3.1 Combining networks with linear regression

We assume that we are given as input m networks, which we index by d, W_d ∈ R^{n×n}, W_d = W_d^T, where the (i,j)-th element of W_d, w_d^{ij} ≥ 0 for all i and j. We interpret w_d^{ij} as the strength of the evidence of co-functionality between genes i and j as derived from dataset d. Using annotation databases such as GO, for each GO term that describes a given category of gene function c, positive genes are defined as genes that are annotated to c and we consider all other genes as negatives: that is, we define a label vector z ∈ {+, −}, where positive and negative genes are labeled as +, −, respectively. Our goal is to construct a composite network as a weighted sum of the m networks W^* = ∑_d µ_d W_d, where µ_d is the weight assigned to network d, such that W^* can be used to predict other positive genes.

To assign the network weights, GeneMANIA solves a constrained linear regression problem by minimizing the least squares error between the composite network and the target network T which represents the pairwise functional relationships implied by the label vector:

\[ \tilde{\mu}^* = \arg\min_{\mu} \text{tr}(T - W^*)^T (T - W^*), \]

subject to W^* = ∑_d µ_d W_d, µ_d ≥ 0, d = [1, ..., m]

where the target network T has elements T_{ij} taking one of the two values: \( \pm \sqrt{\frac{1}{m}} \) if genes i and j are both positive and \( \pm \sqrt{\frac{m-2}{m}} \) for making predictions about gene function.

By using the fact that tr(CT) = vec(W^T) vec(T), where vec(W) is an operator that stacks the columns of matrix W atop of each other, we can write (1) as a non-negative linear regression problem:

\[ \tilde{\mu}^* = \arg\min_{\mu} \left( \tilde{r} - \Omega \mu \right)^T \left( \tilde{r} - \Omega \mu \right), \]

where \( \tilde{r} = vec(T), \Omega = [vec(W_1), ..., vec(W_m)] \) and \( \tilde{\mu} = [\mu_1, ..., \mu_m]^T \). In practice, we include a column of ones in \( \Omega \) and calculate a bias \( \mu_0 \) that we discard when constructing W^*.

Unlike the other values of \( \mu_d \), \( \mu_0 \) is not constrained to be positive. Solving Equation (2) requires at most m iterations (though in practice the number of iterations is much smaller), each iteration involves solving a system of linear equations with m variables and a matrix-vector product. As m (the number of networks) tends to be smaller than 100, we can compute the network weights very fast (e.g. in seconds on a standard computer).

3.2 Combining networks with SWs

Although the above approach is fast, it often performs poorly in predicting categories that have a small number of annotations (as discussed in Section 5). In Mostafavi et al. (2008), it was shown that an L2 norm regularization (also known as ridge regression) to a mean weight prior, improves performance in such categories (Section 5.1.1). However, assessing this prior requires solving several regression problems. Here, we define a simple modification

1760
that improves the performance without increasing the computational time and show that it performs better than previous approaches. In particular, in SW, instead of assigning network weights for each category separately, we fit the network weights to a set of related function categories. To do so, we assign the network weights by solving the following problem:

\[
\tilde{\mu}^* = \arg\min_{\mu} \sum_{c=1}^{k} (\tilde{t}_c - \Omega \tilde{\mu})^T (\tilde{t}_c - \Omega \tilde{\mu}), \mu \geq 0
\]

where \( t_c \), for \( c = 1, \ldots, h \) are constructed from \( h \) positively labeled genes sets (categories) that are related to each other. Once we obtain \( \tilde{\mu}^* \), we construct \( W^* \) and use it to predict all \( h \) categories.

If we include all entries of \( \Omega \) (i.e. not excluding negative–negative pairs of genes for each category \( h \) as described above), we can then write the above problem as:

\[
\hat{\mu}^* = \arg\min_{\mu} -2\tilde{\tau}^T \hat{T}^* \mu + b\hat{T}^* \Omega \hat{\Omega} \mu
\]

where \( \tilde{\tau} = \sum_{c=1}^{h} \tilde{t}_c \) and we only need to solve the regression problem once to get the SWs. As such, for each category, \( T^*_c \) takes on one of the three possible values: \((\hat{T}^*_c)^2, (\hat{T}^*_c)^2, -(\hat{T}^*_c)^2\) when \( i, j \) are both negative, both positive and have the opposite signs, respectively, and \( \hat{T}^*_c \) is the number of positives (negatives) in category \( c \). As we will show, including the negative–negative pairs of genes in \( t_c \) and \( \Omega \) does not degrade the performance of the constructed composite network.

In our experiments, constructing \( \tilde{\tau} \) takes \(<5\) s with \( h=1000 \) on a standard computer (2.4 GHz Intel Core 2 Duo, 4 GB RAM). Further, for a given set of networks, we can always precompute \( \hat{\Omega}^2 \Omega \) and thus only need to calculate \( \hat{\Omega}^2 \Omega \) for a group of categories of interest. As we will show in Section 5, combining weights by SWs results in an improvement in the performance of the composite networks in predicting the relevant \( h \) gene categories while it reduces the computation time (as now we are only required to solve for the network weights once while predicting \( h \) categories).

### 3.3 Predicting protein function from a single network

We evaluate a composite network, \( W^* \), by its ability to predict a given gene function. As done in Mostafavi et al. (2008), we use the GRF's algorithm (Zhou et al., 2003) to predict gene function from a single composite network. In particular, given a label vector \( \tilde{y} \) where \( y_i \) represents the prior evidence for gene \( i \) having the function of interest, the GRF algorithm assigns a discriminant score \( \tilde{f}(i) = \sum_{j=1}^{n} w_{ij} (y_j - f_j) \) to each node (gene) \( i \) in the network which we can then threshold to classify the genes. In particular, \( y_i = [-1, 1, +1] \) where known negative and positive genes are assigned \(-1 \) and \(+1 \), respectively, and the unlabeled genes (i.e. the possibility set) are assigned a value \(-1 \leq k \leq +1 \), for example, \( k \) can be adjusted based on a gene’s annotations in GO (Mostafavi and Morris, 2009).

We can write the GRFs algorithm in the following general form:

\[
\tilde{f}^* = \arg\min_{\tilde{f}} \sum_{i=1}^{n} \sigma_i (g_i - f_i)^2 + \sum_{i,j=1}^{n} w_{ij} (g_i - f_i)(g_j - f_j)
\]

where \( \sigma_i \) are model parameters, \( \Sigma \) is a diagonal matrix with \( \Sigma_{ii} = \sigma_i \), \( L = D - W \) is the graph Laplacian and \( D \) is a diagonal matrix with \( D_{ii} = \sum_{j=1}^{n} w_{ij} \). The above objective ensures that the discriminant scores remain close to their initial labels [first term in (3)] and that the discriminant scores of genes likely to share a function (measured by high \( w_{ij} \)) are similar to each other [second term in (3)]. As done in Mostafavi et al. (2008), we set \( k = \frac{1}{\sqrt{n}} \), the mean of the labels of the labeled nodes; this modification results in considerable performance improvement in unbalanced classification problems such as gene function prediction.

### 4 METHODS

In this section, we describe our benchmark datasets, how we construct functional association networks, our evaluation criterion and how we group function categories in SW (see the Supplementary Material for more detailed information).

#### 4.1 Yeast, fly, mouse, human and E.Coli datasets

We evaluate our methodology on benchmark networks in five species: yeast, fly, mouse, human and E.coli. For yeast, we constructed 44 networks that include interactions derived from gene expression, protein and genetic interaction [downloaded from BIOGRID (Stark et al., 2006) and protein localization. For mouse, we use the MouseFunc benchmark (Pena-Castillo et al., 2008), which consists of 10 networks and covers 21,603 mouse genes. For fly, we have constructed 38 networks from various gene expression data [downloaded from GEO (Edgar et al., 2002)], protein interaction [downloaded from BioGRID] and domain composition [downloaded from BioMart (Kasprzyk et al., 2004)] that cover 13,562 fly genes. For E.coli, we use seven networks from Hu et al. (2009) that include co-substrate and protein interactions for 4175 E.coli genes. Similarly, our human benchmark consists of eight networks constructed from various gene expression, protein interaction, domain composition and phenotype data and covers 13,281 human genes obtained from HPRD (Prasad et al., 2006).

#### 4.2 Functional association networks

We construct networks from each profile-based high-throughput data source using the PCC. For network-based data (e.g. protein interaction), we use both a direct interaction network and a correlation-based network using the PCC on the frequency- corrected data [as done in Mostafavi et al. (2008)]. For efficiency, we sparsify our correlation-based networks by setting by keeping the top \( K \) interactions for each gene and setting the rest to zero. See the Supplementary Material for more details. We then normalized all our networks by: \( W_i = D_i^{-1/2} W_d D_i^{-1/2} \) where \( D_i \) is the diagonal row sum matrix of \( W_i \). Similarly, we also normalize the combined network \( W^* \).

#### 4.3 Evaluation

To evaluate gene function prediction, we use the GO biological process (BP) function categories (Ashburner et al., 2000) for Saccharomyces cerevisiae
We have examined several methods for grouping GO categories when assigning SWs including grouping by (i) GO hierarchy (i.e. BP, CC and MF) (ii) GO hierarchy and number of annotations (iii) clustering of GO categories based on annotations and (iv) ancestor and descendant terms with ancestors having a maximum category size (300 annotations). We only report categories based on annotations and (iv) ancestor and descendant terms with

5.1 Performance on yeast networks

We first extensively compare the performance of SW and compare its performance to several other approaches: various regularized linear regression methods, the TSS algorithm and a simpler correlation-based method (described above), using the yeast benchmark networks. We then show analogous results using mouse, fly, human and E.coli benchmark data.

5.1.1 Comparison of performance of SW with various function-specific linear regression methods

We first extensively compare the performance of SW in predicting gene function in yeast to that of GeneMANIA. In particular, as discussed in Section 4, one way to improve the performance of function-specific constrained linear regression in GeneMANIA is to use regularization; in fact, Mostafavi et al. (2008) showed that ridge regression (i.e. $\ell_2$ norm regularization) to a mean weight prior, where the mean weights refer to the average weight assigned to each network in a large number of function predictions, considerably improves the performance with the drawback of increasing the computation time to estimate the mean weights. Here, we investigate the effect of several forms of regularization on the performance of GeneMANIA algorithm where we find the network weights by solving the following problem:

$$J(\hat{\mu}) = \sum_{d=1}^{m} |\mu_d| + \frac{1}{2} \sum_{d=1}^{m} (\mu_d - \bar{\mu})^2$$

where $\bar{\mu}$ is the regularization function. In particular, we investigated the performance of four different regularizations:

(i) ridge with uniform prior, (ii) ridge with mean prior, (iii) LASSO and (iv) elastic net. In LASSO (Tibshirani, 1996), $J(\hat{\mu}) = \alpha_1 \sum_{d=1}^{m} |\mu_d|$, whereas in standard ridge regression

$$J(\hat{\mu}) = \alpha_2 \sum_{d=1}^{m} \mu_d^2$$

where $\alpha_1$ and $\alpha_2$ are regularization constants and determine the strength of the regularization. The elastic net regularization (Zou and Hastie, 2005) combines $\ell_2$ and $\ell_1$-norm penalties: $\alpha_1 \sum_{d=1}^{m} |\mu_d| + \alpha_2 \sum_{d=1}^{m} \mu_d^2$. In Zou and Hastie (2005), it was shown that the elastic net results in a sparse solution and often performs better than the LASSO.

For ridge with a prior, we define $J(\hat{\mu}) = \sum_{d=1}^{m} \frac{1}{2} (\mu_d - \bar{\mu})^2$, where $\bar{\mu}$ is a prior weight vector and $x_d$ determines the strength of the regularization on $\mu_d$. In Mostafavi et al. (2008), the mean weight prior was obtained as the average weight assigned to each category (using unregularized regression) in predicting all categories in the same GO hierarchy (we will refer to this method as ridge with mean prior). In addition, if we set $x_d = 1$ the network weights are shrunk to a uniform value, we call this second method ridge with uniform prior. In our experiments, we set $x_d = 1/\sum_{d=1}^{m} |\mu_d|$ thus, the strength of the regularizer is higher on sparser networks.

Figure 1a summarizes the performance of each method in five evaluation categories: predicting gene functions which have [3–10], [11–30], [31–100], [101–300] positive annotations and [3–300] (i.e. overall) positive annotations. In ridge with mean prior, we set the prior on each network’s weight to the average weight that network received in predicting all 1188 GO BP categories with 3–300 annotations. We used the LARS (Efron et al., 2004) algorithm to solve for the LASSO and elastic net solutions; we set the number of positive coefficients using $F$-statistics (Hastie et al., 2001). For elastic net, we set $\alpha_2 = 1e-6$ using CV. For SW, we used all 1188 BP GO categories to fit the networks weights. In Uniform, the network weights are all set to $1/m$ where $m$ is the number of networks.

Figure 1a shows that SW significantly outperforms ridge regression with mean prior overall in terms of ROC ($P = 0.0437$, Wilcoxon signed rank test) and slightly improves on the performance in terms of precision ($P = 0.0459$, Wilcoxon signed rank test). In addition, if we set $x_d = 1$ the network weights are shrunk to a uniform value, we call this second method ridge with uniform prior. In our experiments, we set $x_d = 1/\sum_{d=1}^{m} |\mu_d|$ thus, the strength of the regularizer is higher on sparser networks.

One explanation for the observed trend in Figure 1a is that regularization methods that shrink the network weights toward zero are too selective and often identify only a few relevant networks. For example, on average 45% (20/44), 54% (24/44) and 95% (42/44), 97% (43/44) of the networks are assigned a non-zero weight using LASSO, unregularized linear regression and ridge with mean prior, and SW, respectively (see Supplementary Fig. S1). Note that the best performing networks on their own are significantly worse than the combined data (Fig. 1b). SW results in a better measure of network relevancy and with the current available genomics and proteomics datasets, one integrated composite functional association network can sufficiently and accurately predict a broad range of functional relationships.

---

1We picked $\alpha_2$ from the set {1e-8, 1e-6, 1e-4, 1e-2, 1e-1, 1} by examining the mean ROC using 3-fold CV.
We note that the absence of the lower bound results in a decrease in Tsuda et al. in predicting BP categories. When applicable (in the case of protein and genetic interaction) we combined all networks derived from the same publication (e.g. direct and correlation network). The combined network was constructed using SW. Regression weights; however, the redundancy between the networks is not accounted for. For example, the protein interaction networks are drawn from separate publications tend to include similar information and the average of weights assigned to these networks by correlation weights is higher than that of linear regression. As expected, the mean weight assigned by linear regression to individual networks is similar to SW for that network. In general, consistent with previous studies (Marcotte et al., 1999), we observed that all methods assign a high proportion of the network weights to the networks derived from gene expression datasets and the protein localization dataset.

5.2 Performance on fly, mouse, human and E.coli benchmarks

We also investigated the performance of unregularized linear regression, SW and uniform network combination with uniform weights (Uniform) in predicting BP categories with [3–10] \( n=1188 \) annotations. Error bars show one standard error. (b) Mean precision of combined and individual data sources (separated by publications) in predicting BP categories. When applicable (in the case of protein and genetic interaction) we combined all networks derived from the same publication (e.g. direct and correlation network). The combined network was constructed using SW.

Fig. 1. (a) Comparison of performance of LASSO, elastic net (ElasticNet), unregularized linear regression (Unregularized), ridge with uniform prior (Ridge (Uniform)), ridge with mean prior (Ridge (mean)), SW and a network combination with uniform weights (Uniform) in predicting BP categories with [3–10] \( n=1188 \) annotations. Error bars show one standard error. (b) Mean precision of combined and individual data sources (separated by publications) in predicting BP categories. When applicable (in the case of protein and genetic interaction) we combined all networks derived from the same publication (e.g. direct and correlation network). The combined network was constructed using SW.

Fig. 2. Performance of SW, TSS and correlation in predicting gene function in yeast according to BP categories.

### 5.1.2 Comparison of SW with TSS and correlation-based network weights

We also compared the performance of SW with two other methods: TSS algorithm (Tsuda et al., 2005) and a simpler correlation-based network weighting method (Fig. 2). In the correlation network weighting, each network is assigned a weight that is inspired by the Kernel Target Alignment score—we set \( \mu_d = \sum_{w_i} \mu_i w_i = \sum_{i=1}^{n-\beta} |w_i| \). Unlike the linear regression methods, correlation-based weighting does not account for the redundancy between the networks. The TSS algorithm (Tsuda et al., 2005) assigns the network weights by optimizing the performance of the GRFs algorithm with the resulting composite network. In our experiments, we set the regularization parameters of the TSS algorithm by CV to \( c_0=0.5 \) and \( c_1=1 \). As done in code provided in Tsuda et al. (2005), we also set a lower bound of 0.01 on \( \mu_d \). We note that the absence of the lower bound results in a decrease in the performance of the TSS algorithms. As shown in Figure 2, SW significantly outperforms correlation-based network weights and TSS in all evaluation categories.

To further understand the differences between these various approaches, we compare the network weights that were assigned to individual networks. As shown in Figure 3, we observed that the TSS algorithm tends to be very selective, often assigning large weights to a few networks and a very low weight (the weight lower bound) to the rest. The correlation weights are similar to the linear regression weights; however, the redundancy between the networks is not accounted for. For example, the protein interaction networks are drawn from separate publications tend to include similar information and the average of weights assigned to these networks by correlation weights is higher than that of linear regression. As expected, the mean weight assigned by linear regression to individual networks is similar to SW for that network. In general, consistent with previous studies (Marcotte et al., 1999), we observed that all methods assign a high proportion of the network weights to the networks derived from gene expression datasets and the protein localization dataset.
We have introduced a new network weighting scheme for combining significantly better than both of the other methods, asterisk indicates that the differences were significant only between SW and unregularized.

Our results show that fitting the SWs to GO categories in the same hierarchy with a broad range of specificities (those with [3–300] annotations) outperform more specific groupings of the GO categories. Note that, because we adjust the target vector \( \hat{c} \) to balance the number of positives and negatives in each category \( c \), the larger GO categories contribute more to the overall target vector \( \hat{t} \); on the other hand, there are many more categories with [3–10] annotations.

In summary, we have demonstrated the feasibility and the utility of constructing a single composite network with SWs for predicting various GO categories. Unlike a fixed network combination with uniform weights, SWs account for noisy and redundant networks. This observation can in turn speed up gene function prediction from multiple networks.

### ACKNOWLEDGMENTS

We would like to thank Khalid Zuberi, Quentin Shao and Javier Diaz for their help with collecting the networks.

**Fundings:** Ontario Graduate Scholarship (partially funded to S.M.), Genome Canada Technology Development Project administered through Ontario Genomics Institute and a Natural Science and Engineering Operating grant (to Q.M.).

**Conflict of Interest:** none declared.

### REFERENCES

Adziboe-Abi, M. et al. (2000) Gene ontology: tool for unification of biology. Nat. Genet., 25, 25–29.

Baitsch,A. (2000) The enzyme database in 2000. Nucleic Acids Res., 28, 304–305.

Cristianini,N. et al. (2002) On kernel target alignment. In Proceedings of the Tenth Conference on Advances in Neural Information Processing Systems, Vancouver, BC, Canada, pp. 367–373.

Edgar,R. et al. (2002) Gene expression omnibus: NCBi gene expression and hybridization array data repository. Nucleic Acids Res., 30, 207–210.
Fast integration of heterogeneous data sources

Efron, B. et al. (2004) Least angle regression. Ann. Stat., 32, 407–499.

Hastie, T. et al. (2001) The Elements of Statistical Learning. Springer, NY, USA.

Hu, P. et al. (2009) Global functional atlas of escherichia coli encompassing previously uncharacterized proteins. PLoS Biol., 7, e96.

Kanehisa, K. and Goto, S. (2000) KEGG: Kyoto encyclopedia of genes and genome. Nucleic Acids Res., 28, 27–30.

Karaoz, U. et al. (2003) Whole-genome annotation by using evidence integration in functional-linkage networks. Proc. Natl Acad. Sci. USA, 101, 2888–2893.

Kaprzyk, A. et al. (2004) EnsMart: a generic system for fast and flexible access to biological data. Genome Res., 14, 160–169.

Kondor, R. and Lafferty, J. (2002) Diffusion kernels on graphs and other discrete structures. Int. Conf. Mach. Learn. (ICML), 11, 463–475.

Lanckriet, G. et al. (2004) A statistical framework for genomic data fusion. Bioinformatics, 20, 2626–2635.

Marcotte, E. et al. (1999) A combined algorithm for genome-wide prediction of protein function. Nature, 42, 83–86.

Mostafavi, S. and Morris, Q. (2008) Integrating evidence from multiple networks for gene function prediction. Genome Biol., 9 (Suppl. 1), S4.

Rocklin, R. et al. (2006) Learning gene functional classification from multiple data types. J. Comput. Biol., 9, 401–411.

Renaudin, P. et al. (2002) Learning gene functional classification from multiple data types. J. Comput. Biol., 9, 401–411.

Qi, Y. et al. (2008) Finding friends and enemies in an enemies-only network: a graph diffusion kernel for predicting novel genetic interactions and co-complex membership from yeast genetic interactions. Genome Res., 18, 1991–2004.

Stark, C. et al. (2006) BioGRID: a general repository for interaction datasets. Genome Biol., 7, D53.

Zhou, D. et al. (2004) Learning with local and global consistency. Adv. Neural Inf. Process. Syst., 16, 321–328.

Zhu, J. and Hastie, T. (2005) Regularization and variable selection via the elastic net. J. R. Stat. Soc. B., 67, 301–320.