BIOINFORMATICS ORIGINAL PAPER

Vol. 26 no. 7 2010, pages 912–918
doi:10.1093/bioinformatics/btq053

Systems biology

Advance Access publication February 12, 2010

# Gene function prediction from synthetic lethality networks via ranking on demand

Christoph Lippert[1,*], Zoubin Ghahramani[2] and Karsten M. Borgwardt[1]

[1]Machine Learning & Computational Biology Research Group, Max Planck Institutes, Tübingen, Germany and [2]Department of Engineering, University of Cambridge, Cambridge, UK

Associate Editor: Jonathan Wren

## ABSTRACT

**Motivation:** Synthetic lethal interactions represent pairs of genes whose individual mutations are not lethal, while the double mutation of both genes does incur lethality. Several studies have shown a correlation between functional similarity of genes and their distances in networks based on synthetic lethal interactions. However, there is a lack of algorithms for predicting gene function from synthetic lethality interaction networks.

**Results:** In this article, we present a novel technique called *kernelROD* for gene function prediction from synthetic lethal interaction networks based on kernel machines. We apply our novel algorithm to Gene Ontology functional annotation prediction in yeast. Our experiments show that our method leads to improved gene function prediction compared with state-of-the-art competitors and that combining genetic and congruence networks leads to a further improvement in prediction accuracy.

**Contact:** christoph.lippert@tuebingen.mpg.de

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

Received on October 30, 2009; revised on January 29, 2010; accepted on February 3, 2010

## 1 INTRODUCTION

*Synthetic lethality—concept and mechanisms*: synthetic lethal interactions have received much attention in the genetics community over recent years. A synthetic lethal interaction refers to a pair of genes which ‘interact’ in the following sense: while mutating each of the two genes individually does not cause lethality, a double mutation of both genes does show a lethal effect on the organism. Hence, synthetic lethality seems to indicate a compensatory effect of two genes, with one gene compensating for the deletion of the other and with lethal consequences only if both genes are deleted jointly.

The two main hypotheses to explain synthetic lethality between two genes A and B are the within- and between-pathway models (Boone *et al.*, 2007; Ye *et al.*, 2005b). The within-pathway hypothesis assumes that both genes A and B are part of the same pathway, and that the function of this pathway is diminished by the single mutations, but rendered below the viability threshold by their double mutation. The between-pathway hypothesis assumes that A and B act in parallel pathways that can compensate for defects in the other. These two main hypotheses can be extended in more complex models of synthetic lethality (e.g. see Ma *et al.*, 2008), and algorithms for detecting pathways within genetic networks have been defined based on these hypotheses (Kelley and Ideker, 2005; Ma *et al.*, 2008; Ulitsky and Shamir, 2007).

*Synthetic lethality and gene function*: the compensatory effect of genes in synthetic lethal interactions and the two main hypotheses to explain this phenomenon already hint at a strong link between gene function and synthetic lethal interaction. This link could be confirmed in several studies. Tong *et al.* (2004) report that 12 and 27% of synthetic lethal interaction pairs have identical or similar Gene Ontology (GO; Ashburner *et al.*, 2000) annotations, respectively. Ye *et al.* (2005a) report correlations between GO annotations of genes and their distances in so-called congruence networks which are derived from the genetic network. These congruence networks quantify similarity between two genes by means of a score that depends on their number of common neighbors in the genetic interaction graph. In a second study, Ye *et al.* (2005b) report correlations between GO annotations of gene pairs and this congruence score. This correlation is strongest for the GO annotations that refer to the biological process and the cellular component that these genes are part of, and weaker for their molecular function. Qi *et al.* (2008) defined diffusion kernels on genetic interaction networks whose scores were shown to correlate with semantic similarity of gene functions according to all three GO categories.

*Goals and scope of this article*: while previous studies focused on examining whether there is a correlation between genetic network structure and functional associations (Qi *et al.*, 2008; Tong *et al.*, 2004; Ye *et al.*, 2005a, b) or in discovering pathways (Kelley and Ideker, 2005; Ma *et al.*, 2008; Ulitsky and Shamir, 2007), our goal in this article is to define algorithmic machinery to rank all the genes in a genetic network based on their likelihood of belonging to a particular functional class, given a set of examples from this class. This ranking provides guidance in choosing promising targets for experimental function determination.

First, we study this problem of gene function prediction in yeast and for all three definitions of gene function provided by the GO (Ashburner *et al.*, 2000): biological process, molecular function and cellular component. Second, we assess the prediction accuracy of our method in comparison with that of state-of-the-art methods. Third, we study whether combining predictions based on genetic

[*]To whom correspondence should be addressed

and congruence networks improves accuracy, or whether both types of networks provide redundant information.

*Related work*: the approaches closest to the one presented here are those which unify several different data sources, including genetic networks, into a joint prediction of gene function or co-complex membership (Deng *et al.*, 2004; Lee *et al.*, 2006; Letovsky and Kasif, 2003; Qiu and Noble, 2008; Tian *et al.*, 2008). The approach presented here, however, differs conceptually from these earlier studies (Sharan *et al.*, 2007), as we describe in the following. Ranking on demand ranks all genes in a given network based on their likelihood of belonging to a set of example genes from the same functional class.

Related work on methods on ranking on demand is scarce. The family of techniques for transductive semi-supervised learning does not apply here, as we are dealing with one class, while these are designed for discriminative learning on two or more classes (Chapelle *et al.*, 2006; Tsuda *et al.*, 2005). A noteworthy exception are *Bayesian sets* (BS; Ghahramani and Heller, 2006) that use a model-based concept of a cluster (i.e. functional class of genes) to rank items (i.e. genes) using a score that evaluates the marginal probability that each item belongs to a cluster containing the query items. Another ranking on demand approach is *RankProp* (Weston *et al.*, 2004), which is designed for detecting remote homologs in a network of protein sequence similarities by propagating this sequence similarity through the graph. For a special choice of kernel, the random walk kernel, our kernel-based ranking procedure can imitate this information propagation idea from RankProp. Ranking on demand has been performed before for loss-of-function RNAi phenotype prediction on biological networks (Lee *et al.*, 2008). This approach is based upon the *guilt by association* (GBA) principle: the more edges between a node and the labeled examples, the higher its position in the ranking and the larger its probability of belonging to the same functional class. However, there are justified doubts that this successful technique will work on synthetic lethal interaction networks as well, because it only considers direct neighbors in a graph (nodes connected by an edge), and does not take the number of common neighbors into account, which has been shown to be correlated with functional similarity of two genes in genetic networks (Ye *et al.*, 2005b). A central point in our experiments is to find out how this GBA approach compares with our novel approaches to function prediction. In our experiments, we compare our kernel-based technique with BS and GBA, and explore the use of diffusion and random walk kernels within our framework.

## 2 ALGORITHM

Our algorithm for function prediction on synthetic lethality networks focuses on the following scenario: given a network $\mathcal{G}$ of synthetic lethal interactions, one has experimentally determined a subset $\mathcal{D}_\mathcal{C}$ of genes that exhibit a particular function. The goal is then to rank all remaining genes in the genetic network based on their likelihood of having the same biological function (see Fig. 1 for a schematic illustration). In algorithmic terms, the problem of 'ranking of demand' can be defined as follows.

Problem Statement 1 (Ranking on demand). *Given a graph $\mathcal{G}$ with N nodes $\mathcal{V}=\{v_1,\ldots,v_N\}$ and a symmetric, undirected and unweighted adjacency matrix* **A**, *and a subset of nodes $\mathcal{D}_\mathcal{C}$ belonging to the same functional class $\mathcal{C}$. The problem of ranking on demand is to find a permutation $\pi:\mathbb{N}\to\mathbb{N}$ on the vertices in $\mathcal{V}\setminus\mathcal{D}_\mathcal{C}$ such that for $\mathbf{x}_i,\mathbf{x}_j\in\mathcal{V}\setminus\mathcal{D}_\mathcal{C}$*

$$\pi(i)\geq\pi(j)\Rightarrow \mathrm{P}(\mathcal{C}|\mathbf{x}_i)\geq \mathrm{P}(\mathcal{C}|\mathbf{x}_j), \tag{1}$$

*i.e. $\pi$ ranks the nodes in $\mathcal{V}\setminus\mathcal{D}_\mathcal{C}$ based on their likelihood of belonging to $\mathcal{C}$ themselves.*

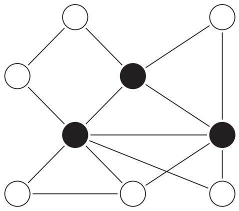

**Fig. 1.** Schematic illustration of 'Ranking on demand': given is a network of synthetic lethal interactions and a query set of genes with known common function (black nodes). The task is to rank all other genes (white nodes) based on their likelihood of exerting the same function.

*A kernel approach to ranking on demand*: it seems attractive to define a kernel-based approach for ranking on demand due to the efficiency of kernel machines in dealing with graph-structured data (Smola and Kondor, 2003) and in combining several different data sources (Schölkopf *et al.*, 2004).

The key observation that led to our novel algorithm was that *BS*, the Bayesian inference approach to ranking on demand (Ghahramani and Heller, 2006), computes a two-sample test statistic to score the likelihood of a gene (an item) to belong to a particular functional class (a cluster). A two-sample test tries to decide whether two samples, in our case **x** and $\mathcal{D}_\mathcal{C}$, have been generated by the same distribution or not. In more detail, BS compute a Bayes factor that compares two hypotheses: the hypothesis $\mathcal{H}_1$ that **x** and $\mathcal{D}_\mathcal{C}$ were generated by the same distribution versus the hypothesis $\mathcal{H}_2$ that they were generated by two different distributions.

The key idea in designing a kernel algorithm for ranking on demand is to compute a score that is based on a kernel-based two-sample test statistic rather than a Bayes factor. Such a kernel-based test statistic for the two-sample problem is the *Maximum Mean Discrepancy* (MMD) by Gretton *et al.* (2007).

*Ranking criterion*: the MMD is a criterion for measuring similarity between two distributions (population MMD) or between samples from two distributions (empirical MMD). The empirical MMD was shown to be equivalent to the distance between the means of two samples in a universal reproducing kernel Hilbert Space (Gretton *et al.*, 2007).

Theorem 1. *Let p and q be distributions and $X=\{\mathbf{x}_1,\ldots,\mathbf{x}_{m_1}\}$ and $Y=\{\mathbf{y}_1,\ldots,\mathbf{y}_{m_2}\}$ be samples from p and q, respectively. Let $\phi:\mathcal{X}\to\mathcal{H}$ be a mapping from input space to feature space, and let $k:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ be the associated kernel function. Then the empirical* MMD *can be computed as*

$$\mathrm{MMD}(\mathcal{H},X,Y):=\|\frac{1}{m_1}\sum_{i=1}^{m_1}\phi(\mathbf{x}_i)-\frac{1}{m_2}\sum_{i=1}^{m_2}\phi(\mathbf{y}_i)\|$$

**Algorithm 1** Kernel-based ranking on demand (kernelROD)

```
background: A graph $\mathcal{G}$ with nodes $\mathcal{V}$ and adjacency matrix $\mathbf{A}$, a kernel function $k(.,.)$ between nodes in $\mathcal{G}$
input: a query $\mathcal{D}_{\mathcal{C}} = \{\mathbf{x}_i\} \subset \mathcal{V}$
for all $\mathbf{x} \in \mathcal{V} \setminus \mathcal{D}_{\mathcal{C}}$ do
    $\text{score}(\mathbf{x}) = -\|\phi(\mathbf{x}) - \frac{1}{|\mathcal{D}_{\mathcal{C}}|} \sum_{\mathbf{x}_i \in \mathcal{D}_{\mathcal{C}}} \phi(\mathbf{x}_i)\|^2$
end for
output: return elements of $\mathcal{V} \setminus \mathcal{D}_{\mathcal{C}}$ sorted by descending score
```

$$= \left[ \frac{1}{m_1^2} \sum_{i,j=1}^{m_1} \mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) - \frac{2}{m_1 m_2} \sum_{i,j=1}^{m_1, m_2} \mathrm{k}(\mathbf{x}_i, \mathbf{y}_j) + \frac{1}{m_2^2} \sum_{i,j=1}^{m_2} \mathrm{k}(\mathbf{y}_i, \mathbf{y}_j) \right]^{\frac{1}{2}}. \tag{2}$$

The kernel-based estimator (2) follows from the fact that a kernel is an inner product between objects in feature space, i.e. $\mathrm{k}(\mathbf{x}_i, \mathbf{y}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{y}_j)$. The intuitive definition of a kernel is that it represents a similarity measure between objects $\mathbf{x}_i$ and $\mathbf{y}_j$.

Using this test statistic, Gretton *et al.* (2007) defined two-sample tests whose key concept can be summarized as follows: the larger the empirical MMD between $X$ and $Y$, the larger the probability that $p$ and $q$ are not the same. Hence $-$MMD (negative MMD), and equivalently $-\text{MMD}^2$, shows exactly the same behavior as the score in BS: the larger $-$MMD, the more likely it is that the two samples were generated by the same distribution.

*Computation*: to determine the ranking score of a gene $\mathbf{x}$, we compute the squared distance between $\mathbf{x}$ and the set of examples $\{\mathbf{x}_i\}$ in feature space,

$$\|\phi(\mathbf{x}) - \frac{1}{|\mathcal{D}_{\mathcal{C}}|} \sum_{\mathbf{x}_i \in \mathcal{D}_{\mathcal{C}}} \phi(\mathbf{x}_i)\|^2. \tag{3}$$

We then multiply this score by $-1$, as we want scores to be the largest for $\mathbf{x}$ that are likely to fit into the set of examples $\mathcal{D}_{\mathcal{C}}$. In terms of kernels, this score($\mathbf{x}$) can be rewritten as

$$-\left[ \mathrm{k}(\mathbf{x}, \mathbf{x}) - \frac{2}{|\mathcal{D}_{\mathcal{C}}|} \sum_{j=1}^{|\mathcal{D}_{\mathcal{C}}|} \mathrm{k}(\mathbf{x}, \mathbf{x}_j) + \frac{1}{|\mathcal{D}_{\mathcal{C}}|^2} \sum_{i,j=1}^{|\mathcal{D}_{\mathcal{C}}|} \mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) \right]. \tag{4}$$

We can drop the third term from (4) as it does not depend on $\mathbf{x}$ and hence does not affect the ranking. Hence, we have to compute $|\mathcal{D}_{\mathcal{C}}| + 1$ kernel values for each of the $N$ objects in our dataset to perform kernel-based ranking on demand. The entire procedure is summarized in Algorithm 1.

*Kernel design for function prediction on genetic networks*: the choice of kernel $\mathrm{k}(.,.)$, which can be thought of as a similarity measure between two genes, is crucial for applying kernel-based ranking on demand to function prediction on genetic networks. We used different kernel functions in our experiments and provide here biological interpretations of how they measure similarity between two nodes in a genetic network.

*Random walk kernels*: as synthetic lethal interactions have been shown to occur directly between functionally related genes (Tong *et al.*, 2004), we designed a kernel that captures direct neighborship between genes. The $p$-step random walk kernel exhibits this property (Smola and Kondor, 2003). It is defined as $\mathbf{K}_{\text{rw}} = (a\mathbf{I} - \tilde{\mathbf{L}})^p$, where $a$ is a scalar, $\mathbf{I}$ is the identity matrix and $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{A})\mathbf{D}^{-1/2}$ with $\mathbf{D}(i,i) = \sum_j \mathbf{A}(i,j)$ is the normalized graph Laplacian. This kernel measures similarity between two nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ in a graph in terms of the number of random walks from $i$ to $j$ in $p$ steps. The term $aI$ allows the random walk to remain in the same node in the next step with some probability (we will refer to $a$ as the restart parameter). The larger $a$, the larger this probability.

For $p = 1$ the random walk only takes one step, which means it reaches only the direct neighbors of a node and hence the associated kernel deems direct neighbors in the graph similar, which fits the observation by Tong *et al.* (2004). For $p = 2$ one may visit nodes that are direct or indirect neighbors of a given node, which means that the corresponding kernel deems direct neighbors and nodes with similar neighborhoods similar, combining both the observations by Tong *et al.* (2004) and Ye *et al.* (2005b). For $p > 2$, the kernel leads to a diffusion-like exploration of the graph and resembles the graph diffusion kernel by Qi *et al.* (2008), whose kernel values have been shown to correlate with similarity in gene function, and the information propagation approach RankProp (Weston *et al.*, 2004).

*Diffusion kernels*: for comparison, we also considered the diffusion kernel as defined by Kondor and Lafferty (2002) and used by Qi *et al.* (2008), which is defined as $\mathbf{K}_{\text{diffusion}} = \exp(-\beta\tilde{\mathbf{L}})$, where $\beta$ is a scalar, $\tilde{L}$ the normalized graph Laplacian as defined above and exp is the matrix exponentiation. This diffusion kernel deems nodes similar if they are in close proximity in the graph and connected by several paths. Qi *et al.* (2008) summed even- and odd-length path separately in their approach.

## 3 METHODS AND MATERIALS

*Dataset*: the BioGRID repository (Stark *et al.*, 2006), version 2.0.51, was parsed for a total of 11 998 directed synthetic lethal interaction pairs between $N = 2579$ *Saccharomyces cerevisiae* genes. By correcting for double counting bait-hit pairs, this results in a network of 10 791 undirected synthetic lethal interactions. Functional annotations for these genes were obtained from the *Saccharomyces* Genome Database (Cherry *et al.*, 1998) and follow the GO nomenclature (Ashburner *et al.*, 2000). Functional categories in the GO follow a general-to-specific ordering between terms of different depths in the GO graph. The annotations are transitive, meaning that if a gene is annotated with a term, this includes all terms that are more general. We make this explicit by annotating each gene by its GO terms and all GO terms that are more general than the annotated ones. We repeat this procedure for each of the three main GO branches: molecular function, biological process and cellular component.

*Congruence network*: in addition to the network of direct synthetic lethal interactions, we also computed a genetic congruence network based on the undirected synthetic lethal interactions as described by Ye *et al.* (2005a, b). They define a genetic congruence score of two nodes $\mathbf{x}_i \neq \mathbf{x}_j$ as the negative logarithm of the probability that the number of shared true neighbors, $|\mathcal{N}^*_{\mathbf{x}_i} \cap \mathcal{N}^*_{\mathbf{x}_j}|$, in the graph is equal to or higher than the observed number of shared neighbors, $|\mathcal{N}_{\text{obs}}| = |\mathcal{N}_{\mathbf{x}_i} \cap \mathcal{N}_{\mathbf{x}_j}|$, of $\mathbf{x}_i$ and $\mathbf{x}_j$, that is

$$\text{congruence}(\mathbf{x_i}, \mathbf{x_j}) = -\log_{10} P\left(|\mathcal{N}^*_{\mathbf{x}_i} \cap \mathcal{N}^*_{\mathbf{x}_j}| \geq |\mathcal{N}_{\text{obs}}|\right),$$

where $P$ is hypergeometrically distributed:

$$P\left(|\mathcal{N}^*_{\mathbf{x}_i} \cap \mathcal{N}^*_{\mathbf{x}_j}| \geq |\mathcal{N}_{\text{obs}}|\right) = \sum_{l=|\mathcal{N}_{\text{obs}}|}^{\min(|\mathcal{N}_{\mathbf{x}_i}|, |\mathcal{N}_{\mathbf{x}_j}|)} \frac{\binom{|\mathcal{N}_{\mathbf{x}_i}|}{l} \binom{N - |\mathcal{N}_{\mathbf{x}_i}|}{|\mathcal{N}_{\mathbf{x}_j}| - l}}{\binom{N}{|\mathcal{N}_{\mathbf{x}_j}|}}.$$

In order to retain only significant edges, all edges $(\mathbf{x}_i,\mathbf{x}_j)$ with $P(\mathbf{x}_i,\mathbf{x}_j) > \frac{0.01}{N*(N-1)/2}$ were deleted, where the denominator equals the total number of possible edges in the congruence graph and corrects for multiple testing.

*Experimental setting*: gene function prediction experiments were performed on each of the three main GO branches for GO terms of GO depth (or GO *level*) 1, 2, 3 and 4, where depth is measured as the shortest distance to the root of the GO hierarchy. Depth 1 comprises the most general terms. For each of these terms, i.e. for each node on this level of the GO, we performed *5-fold stratified cross-validation* (5CV). The genes were randomly divided into five sets of equal size and equal proportions of members and non-members of a functional class. Then in each of five iterations, four sets were used for training and the remaining set was used as the query database. This ensures that each gene was used for testing exactly once.

*Ascertainment bias correction*: our experiment might suffer from an ascertainment bias (Supplementary Section 2.1), as bait genes from the same functional classes have a higher probability of being connected by an edge, and might artificially boost prediction accuracy if they appear in both training and test set. To avoid this effect, we make sure that bait genes with same function are either all in the training dataset or all in the test set.

*Comparison methods and parameter optimization*: we run kernelROD using a random kernel on the synthetic interaction network (synth), the congruence network (cong) and using the sum of both kernels (sum). We conduct the same three experiments for the diffusion kernel. We compare kernelROD with two ranking on demand approaches, GBA and BS, and to a two-class support vector machine (SVM) and a one-class SVM (SVM1) (for a full description—including the choice of negative examples for the SVM—see Supplementary Section 1). We set the parameters for all methods by 3CV on the training set. Parameters include the number of steps in the $p$-step random walk kernel ($p \in \{1,\dots,4\}$), the restart parameter $a$ which controls the influence of shorter random walks ($a \in \{2^1,\dots,2^{11}\}$), the diffusion parameter of the diffusion kernel ($\beta \in \{2^{-5},\dots,2^2\}$), weights for the convex combination of two kernels ($\lambda_1 \in \{0.1,\dots,0.9\}$ and $\lambda_2 = 1-\lambda_1$). In the case of the SVM there is also the parameter $C \in \{10^{-6},\dots,10^6\}$ and in the SVM1 a parameter defining the size of outlier quantile ($\nu \in \{10^{-6},\dots,10^0\}$).

*Evaluation criteria*: in order to assess the quality of the rankings, we computed the area under the receiver operator characteristic curve (AUC) and the area under the receiver operator characteristic for up to 50 false positives (AUC50) for each ranking. The AUC score reflects the probability that a randomly drawn positive example is ranked higher than a randomly drawn negative example. AUC50 reflects the probability that a randomly chosen positive example is ranked higher than one random example out of the 50 highest ranked negative examples. The latter criterion is more appropriate in a setting in which one is only interested in a small set of high confidence predictions that can be further evaluated experimentally. We report the AUC results in the main paper and the AUC50 results in Supplementary Material.

To quantify the performance of a method across tasks, we compute the $L_2$-distance between the results of this method and the best method on each task. A low $L_2$-distance indicates that a method is always close to the best performing one on each task.

## 4 RESULTS

We perform gene function prediction via cross-validation on a synthetic lethality network from yeast using kernelROD and several comparison methods: BS, GBA and SVM1 and two-class SVM (see Table 1 and Fig. 2, and Supplementary Tables 2 and 3 and Supplementary Figs 6 and 7 for all results).

*All methods are significantly better than random*: as a first check, we examined whether the results we had obtained with kernelROD, GBA and BS were better than random. For this purpose, we generated 1000 random rankings on each task and computed the AUC values for these random rankings. We then used this approximated null distribution of AUC values from random rankings to compute a $P$-value of the results of our methods. If the methods worked only as good as a random ranker, one would expect to obtain a distribution over $P$-values that is close to uniform. As can be seen from Figure 3 and Supplementary Figure 8, the distribution of $P$-values of all methods is skewed toward small $P$-values, and highly significantly different from that of a random ranker [Kolmogorov-Smirnov test (KS-test), $P=0$].

*kernelROD improves GO term prediction by ranking on demand*: kernelROD based on the combination of a random walk kernel on both the synthetic and the congruence network gives the best results across all GO branches and GO levels, that is, the lowest $L_2$ distance to the best method on each task in terms of AUC.

The improvement achieved by kernelROD is largest when looking at AUC50 scores (Supplementary Fig. 7): here kernelROD with a random walk kernel achieves the lowest $L_2$ distance on all three networks, that is the synthetic lethal interaction network, the congruence graph and on both. In terms of AUC, BS perform similarly well as kernelROD on the synthetic interaction network. This indicates that the top-ranked predictions of kernelROD tend to be more accurate than those of BS.

*Indirect interactions improve rankings*: to further understand why kernelROD leads to improved AUC scores, we examine the impact of the number of steps taken in its random walk kernel on its AUC scores. We compare kernelROD with a one-step random walk kernel and kernel with a two-step random walk kernel on the synthetic interaction network, for all GO branches and GO depths from 1 to 4 (Table 1). The two-step random walk kernel gives significantly better results than the one-step random walk kernel, reaching a higher AUC value in 10 out of 12 settings ($P=0.0032$; Binomial distribution with $n=12$, $P=0.5$). This indicates that it is useful to consider more than just direct interactions for gene ranking.

*Combination of networks improves prediction of GO terms*: in 8 out of 12 experiments, the best AUC result is achieved using kernelROD with a sum of kernels on both the synthetic and the congruence network. In three out of the other four experiments, a random walk kernel on the synthetic network or the congruence network yields the highest AUC score. These results indicate that both data integration is beneficial for function prediction and that direct synthetic lethal interactions and shared neighbors in the synthetic lethal interaction network are both indicative of joint function.

*Performance varies for different GO levels and branches*: next we examine whether our prediction results differ significantly between different GO branches and for different levels in the GO hierarchy (see Table 1 and the plots of AUC versus GO level in Fig. 2). In Figure 2, we compare kernelROD to GBA and BS.

The AUC results across GO branches do not differ significantly, they are all clustered in the range from 0.55 to 0.65. On GO molecular function (Fig. 2a), results on the congruence graph improve with GO depth, whereas the methods on the synthetic interaction graph show no uniform trend. On GO molecular function (Fig. 2b), results of all methods on the synthetic network deteriorate from GO level 1 to 3, whereas the methods on the congruence graph improve simultaneously. On GO biological process (Fig. 2c), we observe an increase in AUC for deeper levels of the GO hierarchy for all methods expect for GBA on the synthetic interaction network. This indicates that across all GO branches, the gene function classes tend to be the more clustered in the congruence network, the deeper

**Table 1.** Results for GO term prediction using genetic interaction networks: Average AUC and standard deviations for GO term prediction on three GO branches

| Five times 5-fold cross-validation with correction for ascertainment bias | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AUC | | GO cellular component | | | | | | | | GO molecular function | | | | | | | | GO biological process | | | | | | | | |
| GO level | | 1 | | 2 | | 3 | | 4 | | 1 | | 2 | | 3 | | 4 | | 1 | | 2 | | 3 | | 4 | | |
| Method | Kernel | Avg. | SD | Avg. | SD | Avg. | SD | Avg. | SD | Avg. | SD | Avg. | SD | Avg. | SD | Avg. | SD | Avg. | SD | Avg. | SD | Avg. | SD | Avg. | SD | L2 |
| Synthetc lethal interaction network | | | | | | | | | | | | | | | | | | | | | | | | | | |
| kROD | RW | 0.619 | 0.131 | 0.614 | 0.171 | 0.611 | 0.197 | 0.626 | 0.201 | 0.621 | 0.169 | 0.590 | 0.176 | 0.575 | 0.219 | 0.590 | 0.218 | 0.600 | 0.125 | 0.615 | 0.154 | 0.604 | 0.190 | 0.611 | 0.194 | **0.040** |
| kROD | Diff | 0.549 | 0.149 | 0.585 | 0.165 | 0.599 | 0.202 | 0.611 | 0.205 | 0.592 | 0.172 | 0.568 | 0.173 | 0.569 | 0.219 | 0.588 | 0.212 | 0.546 | 0.132 | 0.576 | 0.161 | 0.583 | 0.188 | 0.600 | 0.197 | **0.065** |
| kROD | 1step | 0.617 | 0.126 | 0.561 | 0.202 | 0.568 | 0.214 | 0.578 | 0.207 | 0.569 | 0.176 | 0.605 | 0.176 | 0.572 | 0.221 | 0.578 | 0.242 | 0.614 | 0.117 | 0.594 | 0.176 | 0.592 | 0.189 | 0.602 | 0.201 | **0.065** |
| kROD | 2step | **0.630** | 0.137 | 0.628 | 0.162 | 0.612 | 0.199 | 0.618 | 0.202 | 0.627 | 0.159 | 0.591 | 0.170 | 0.584 | 0.219 | 0.602 | 0.212 | 0.605 | 0.121 | 0.608 | 0.156 | 0.601 | 0.189 | 0.615 | 0.194 | **0.038** |
| GBA | – | 0.584 | 0.133 | 0.596 | 0.182 | 0.558 | 0.203 | 0.586 | 0.188 | 0.576 | 0.191 | 0.563 | 0.181 | 0.553 | 0.224 | 0.561 | 0.232 | 0.578 | 0.108 | 0.597 | 0.161 | 0.573 | 0.179 | 0.597 | 0.196 | **0.070** |
| BS | – | 0.581 | 0.137 | 0.637 | 0.154 | 0.595 | 0.214 | 0.609 | 0.206 | 0.625 | 0.176 | 0.611 | 0.167 | 0.601 | 0.218 | 0.596 | 0.225 | 0.573 | 0.103 | 0.597 | 0.151 | 0.598 | 0.182 | 0.618 | 0.188 | **0.045** |
| SVM1 | RW | 0.548 | 0.121 | 0.583 | 0.123 | 0.563 | 0.173 | 0.613 | 0.165 | 0.524 | 0.113 | 0.555 | 0.155 | 0.550 | 0.162 | 0.574 | 0.137 | 0.543 | 0.087 | 0.558 | 0.120 | 0.546 | 0.141 | 0.573 | 0.157 | **0.085** |
| SVM1 | Diff | 0.547 | 0.147 | 0.605 | 0.105 | 0.583 | 0.146 | 0.625 | 0.157 | 0.532 | 0.104 | 0.547 | 0.158 | 0.544 | 0.168 | 0.565 | 0.159 | 0.536 | 0.095 | 0.557 | 0.122 | 0.559 | 0.135 | 0.580 | 0.170 | **0.081** |
| SVM | RW | 0.588 | 0.091 | 0.622 | 0.116 | 0.607 | 0.149 | 0.631 | 0.157 | 0.600 | 0.095 | 0.589 | 0.149 | 0.605 | 0.165 | 0.584 | 0.161 | 0.585 | 0.090 | 0.612 | 0.130 | 0.594 | 0.134 | 0.625 | 0.150 | **0.043** |
| SVM | Diff | 0.584 | 0.098 | 0.631 | 0.110 | 0.600 | 0.166 | 0.634 | 0.169 | 0.604 | 0.111 | 0.609 | 0.141 | 0.602 | 0.161 | 0.597 | 0.156 | 0.599 | 0.101 | 0.606 | 0.131 | 0.592 | 0.132 | 0.626 | 0.153 | **0.040** |
| SVM sim | RW | – | | – | | 0.578 | 0.151 | 0.610 | 0.160 | – | | – | | 0.541 | 0.160 | 0.557 | 0.162 | – | | – | | 0.552 | 0.128 | 0.574 | 0.158 | **0.083** |
| SVM sim | Diff | – | | – | | 0.590 | 0.164 | 0.599 | 0.164 | – | | – | | 0.541 | 0.155 | 0.583 | 0.151 | – | | – | | 0.555 | 0.135 | 0.584 | 0.157 | **0.077** |
| SVM dis | RW | – | | – | | 0.580 | 0.150 | 0.609 | 0.158 | – | | – | | 0.542 | 0.159 | 0.568 | 0.168 | – | | – | | 0.544 | 0.149 | 0.568 | 0.153 | **0.084** |
| SVM dis | Diff | – | | – | | 0.582 | 0.180 | 0.612 | 0.168 | – | | – | | 0.553 | 0.158 | 0.582 | 0.170 | – | | – | | 0.555 | 0.140 | 0.575 | 0.157 | **0.076** |
| Congruence network | | | | | | | | | | | | | | | | | | | | | | | | | | |
| kROD | RW | 0.576 | 0.128 | 0.642 | 0.172 | 0.655 | 0.219 | **0.679** | 0.206 | 0.582 | 0.183 | 0.602 | 0.188 | 0.604 | 0.227 | 0.621 | 0.235 | 0.608 | 0.116 | 0.632 | 0.167 | 0.638 | 0.189 | **0.655** | 0.204 | **0.027** |
| kROD | RW-f | 0.579 | 0.121 | 0.643 | 0.168 | **0.659** | 0.217 | **0.680** | 0.206 | 0.594 | 0.164 | 0.598 | 0.194 | 0.602 | 0.228 | 0.621 | 0.233 | 0.607 | 0.116 | 0.634 | 0.162 | 0.636 | 0.193 | **0.657** | 0.200 | **0.025** |
| kROD | Diff | 0.545 | 0.129 | 0.629 | 0.169 | 0.653 | 0.218 | 0.675 | 0.205 | 0.600 | 0.165 | 0.595 | 0.192 | 0.597 | 0.225 | 0.606 | 0.234 | 0.592 | 0.121 | 0.625 | 0.164 | 0.628 | 0.194 | 0.648 | 0.205 | **0.035** |
| kROD | Diff-f | 0.559 | 0.119 | 0.629 | 0.168 | 0.655 | 0.214 | 0.674 | 0.204 | 0.602 | 0.168 | 0.597 | 0.192 | 0.597 | 0.227 | 0.605 | 0.234 | 0.595 | 0.122 | 0.625 | 0.164 | 0.627 | 0.193 | 0.648 | 0.204 | **0.032** |
| GBA | – | 0.521 | 0.117 | 0.578 | 0.173 | 0.620 | 0.218 | 0.636 | 0.207 | 0.517 | 0.187 | 0.565 | 0.170 | 0.573 | 0.214 | 0.589 | 0.238 | 0.551 | 0.123 | 0.570 | 0.166 | 0.589 | 0.188 | 0.608 | 0.203 | **0.073** |
| SVM1 | RW | 0.539 | 0.100 | 0.579 | 0.144 | 0.573 | 0.155 | 0.609 | 0.148 | 0.525 | 0.136 | 0.551 | 0.152 | 0.536 | 0.167 | 0.567 | 0.162 | 0.541 | 0.088 | 0.559 | 0.107 | 0.557 | 0.135 | 0.585 | 0.149 | **0.086** |
| SVM1 | Diff | 0.560 | 0.115 | 0.601 | 0.107 | 0.573 | 0.155 | 0.606 | 0.167 | 0.523 | 0.141 | 0.565 | 0.159 | 0.550 | 0.150 | 0.575 | 0.145 | 0.539 | 0.088 | 0.562 | 0.120 | 0.564 | 0.136 | 0.588 | 0.150 | **0.080** |
| SVM | RW | 0.578 | 0.115 | 0.634 | 0.112 | 0.602 | 0.161 | 0.644 | 0.154 | 0.590 | 0.101 | 0.598 | 0.155 | 0.619 | 0.158 | 0.582 | 0.160 | 0.593 | 0.101 | 0.611 | 0.132 | 0.593 | 0.137 | 0.628 | 0.146 | **0.042** |
| SVM | Diff | 0.572 | 0.105 | 0.636 | 0.112 | 0.588 | 0.165 | 0.650 | 0.160 | 0.593 | 0.100 | 0.599 | 0.154 | 0.607 | 0.166 | 0.597 | 0.151 | 0.587 | 0.095 | 0.604 | 0.136 | 0.596 | 0.130 | 0.626 | 0.148 | **0.043** |
| SVM sim | RW | – | | – | | 0.578 | 0.172 | 0.614 | 0.167 | – | | – | | 0.565 | 0.148 | 0.569 | 0.158 | – | | – | | 0.555 | 0.129 | 0.590 | 0.150 | **0.074** |
| SVM sim | Diff | – | | – | | 0.596 | 0.161 | 0.614 | 0.159 | – | | – | | 0.551 | 0.150 | 0.563 | 0.152 | – | | – | | 0.548 | 0.127 | 0.577 | 0.153 | **0.077** |
| SVM dis | RW | – | | – | | 0.577 | 0.165 | 0.610 | 0.160 | – | | – | | 0.561 | 0.155 | 0.583 | 0.160 | – | | – | | 0.538 | 0.124 | 0.568 | 0.150 | **0.081** |
| SVM dis | Diff | – | | – | | 0.592 | 0.159 | 0.619 | 0.154 | – | | – | | 0.554 | 0.159 | 0.586 | 0.147 | – | | – | | 0.544 | 0.128 | 0.573 | 0.156 | **0.075** |
| Combination of kernels | | | | | | | | | | | | | | | | | | | | | | | | | | |
| kROD | RW | 0.608 | 0.135 | **0.660** | 0.169 | **0.665** | 0.215 | 0.669 | 0.214 | **0.643** | 0.172 | **0.619** | 0.176 | 0.608 | 0.227 | **0.636** | 0.222 | **0.627** | 0.122 | **0.647** | 0.161 | **0.647** | 0.192 | **0.655** | 0.204 | **0.009** |
| kROD | Diff | 0.562 | 0.127 | 0.639 | 0.171 | **0.663** | 0.220 | **0.677** | 0.215 | **0.641** | 0.167 | 0.610 | 0.192 | 0.603 | 0.229 | 0.624 | 0.222 | 0.603 | 0.134 | 0.635 | 0.165 | 0.639 | 0.195 | **0.657** | 0.205 | **0.024** |
| SVM1 | RW | 0.525 | 0.143 | 0.591 | 0.138 | 0.590 | 0.149 | 0.608 | 0.163 | 0.538 | 0.126 | 0.555 | 0.156 | 0.547 | 0.154 | 0.577 | 0.162 | 0.525 | 0.084 | 0.551 | 0.122 | 0.553 | 0.138 | 0.578 | 0.158 | **0.084** |
| SVM1 | Diff | 0.553 | 0.116 | 0.599 | 0.119 | 0.581 | 0.171 | 0.630 | 0.154 | 0.541 | 0.111 | 0.573 | 0.159 | 0.549 | 0.154 | 0.573 | 0.165 | 0.536 | 0.093 | 0.560 | 0.131 | 0.561 | 0.134 | 0.590 | 0.155 | **0.076** |
| SVM | RW | 0.585 | 0.110 | 0.635 | 0.111 | 0.607 | 0.163 | 0.655 | 0.152 | 0.608 | 0.103 | 0.599 | 0.146 | **0.624** | 0.155 | 0.598 | 0.156 | 0.595 | 0.098 | 0.617 | 0.125 | 0.604 | 0.125 | 0.639 | 0.145 | **0.034** |
| SVM | Diff | 0.594 | 0.108 | 0.637 | 0.110 | 0.615 | 0.163 | 0.651 | 0.163 | 0.601 | 0.140 | 0.610 | 0.149 | 0.611 | 0.168 | 0.601 | 0.172 | 0.603 | 0.097 | 0.618 | 0.128 | 0.606 | 0.125 | 0.639 | 0.150 | **0.032** |
| SVM sim | RW | – | | – | | 0.585 | 0.160 | 0.616 | 0.165 | – | | – | | 0.543 | 0.168 | 0.564 | 0.158 | – | | – | | 0.550 | 0.131 | 0.578 | 0.149 | **0.080** |
| SVM sim | Diff | – | | – | | 0.580 | 0.160 | 0.624 | 0.165 | – | | – | | 0.540 | 0.156 | 0.574 | 0.155 | – | | – | | 0.555 | 0.129 | 0.592 | 0.155 | **0.075** |
| SVM dis | RW | – | | – | | 0.560 | 0.173 | 0.611 | 0.163 | – | | – | | 0.554 | 0.156 | 0.589 | 0.146 | – | | – | | 0.547 | 0.133 | 0.572 | 0.156 | **0.082** |
| SVM dis | Diff | – | | – | | 0.565 | 0.161 | 0.611 | 0.170 | – | | – | | 0.543 | 0.155 | 0.576 | 0.158 | – | | – | | 0.552 | 0.130 | 0.573 | 0.156 | **0.083** |
| random | – | 0.500 | 0.114 | 0.500 | 0.150 | 0.500 | 0.188 | 0.500 | 0.177 | 0.500 | 0.150 | 0.500 | 0.154 | 0.500 | 0.199 | 0.500 | 0.206 | 0.500 | 0.105 | 0.500 | 0.139 | 0.500 | 0.166 | 0.500 | 0.175 | **0.146** |

Depth of the GO tree is denoted by integers 1–4. Best AUC values and AUC values not significantly worse ($\alpha = 5\%$) are marked in bold. 'L2' Euclidean distance to the best methods in individual experiments. (Methods: GBA; BS; kROD, kernelROD; SVM1; SVM, discriminative support vector machines; kernels: RW, random walk kernel; Dif, diffusion kernel; 1step and 2step, random walk kernels with steps restricted to $p = 1$ and $p = 2$; Dif, diffusion kernel; RW-f and Diff-f, kernels on the congruence graph without thresholding.) For complete results see supplementary Tables 1–2.

the GO level we look at. On the synthetic interaction network, we could not detect a similar trend.

*Choice of kernel*: finally, we assess to which degree the choice of kernel affects our gene prediction performance (Table 1). In terms of $L_2$-distance to the best performing method, the random walk kernel is better than the diffusion kernel on the synthetic network, the congruence network and their combination.

## 5 DISCUSSION

In this article, we have presented *kernelROD*, a novel ranking approach for gene function prediction from synthetic lethality networks. In function prediction in yeast, we observe that our kernel-based approach *kernelROD* outperforms state-of-the-art methods and that a combined random walk kernel on genetic networks and on congruence networks (Ye *et al.*, 2005a) often improves, and never harms prediction accuracy. Considering indirect interactions (walks of length 2) in the synthetic interaction network results in improved rankings compared with considering only direct interactions.

We could confirm that congruence networks are useful for function prediction from genetic networks, as reported by Ye *et al.* (2005a, b). We could also confirm that diffusion or random walk-based kernels are a promising approach to function prediction on genetic networks, as reported by Qi *et al.* (2008). Furthermore, we also established that random walk kernels achieve even better

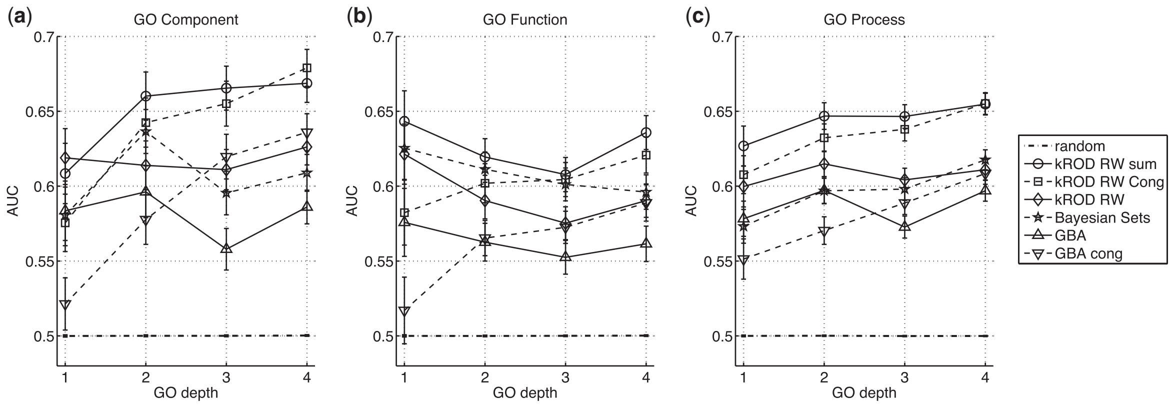

**Fig. 2.** AUC versus GO level. (**a–c**): average AUC values and standard errors on GO levels 1, 2, 3 and 4. kROD, kernelROD; RW, random walk kernel; cong, congruence network; sum, both networks.

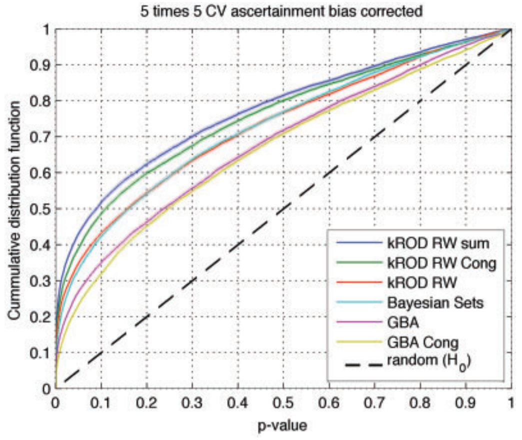

**Fig. 3.** Cumulative *P*-value distribution. Cumulative distribution functions of *P*-values estimated from an empirical null distribution of 1000 random rankings per AUC experiment. For all methods the distributions are significantly biased toward small values. (kROD, kernelROD; RW, random walk kernel; Cong, congruence network; sum, both networks) Best viewed in color.

results on congruence networks. Our best performing kernel, both on genetic and on congruence networks, is the random walk kernel, and it achieves its best results when integrating information both from the synthetic and the congruence network.

We make the following interesting observations which will motivate future research efforts: our ranking on demand approach outperforms discriminative SVMs in function prediction, reaching better results in terms of $L_2$ on the congruence graph and the combined graph, and similar ones on the synthetic graph. This indicates that *within-class similarity* seems to be more important than *between-class dissimilarity* for our task: genes with the same function seem to be in close proximity in the network (within-class similarity), but there are also connected genes with different functions (lack of between-class dissimilarity). The within-class similarity allows our ranking method to obtain good results, because genes with similar function will appear high in the ranking. The lack of between-class dissimilarity is likely to be the cause for the worse performance of discriminative methods, because functional classes cannot be distinguished based on proximity in the network. At the same time, this lack of between-class dissimilarity also keeps our method from reaching higher accuracy levels, as it leads to genes from other functional classes being ranked high. A topic of future research will be to add further data sources to our prediction system that increase the between-class dissimilarity in our feature representation of genes. Due to the closure properties of kernels, our kernel-based ranking algorithm will be a convenient framework for this task of data integration.

*Conflict of Interest*: none declared.

## REFERENCES

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat. Genet.*, **25**, 25–29.

Boone,C. *et al.* (2007) Exploring genetic interactions and networks with yeast. *Nat. Rev. Genet.*, **8**, 437–449.

Chapelle,O. *et al.* (eds) (2006) *Semi-Supervised Learning*. MIT Press, Cambridge, MA.

Cherry,J.M. *et al.* (1998) SGD: Saccharomyces genome database. *Nucleic Acids Res.*, **26**, 73–79.

Deng,M. *et al.* (2004) An integrated probabilistic model for functional prediction of proteins. *J. Comput. Biol.*, **11**, 463–475.

Ghahramani,Z. and Heller,K.A. (2006) Bayesian sets. In Weiss,Y. *et al.* (eds) *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA.

Gretton,A. *et al.* (2007) A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA.

Kelley,R. and Ideker,T. (2005) Systematic interpretation of genetic interactions using protein networks. *Nat. Biotechnol.*, **23**, 561–566.

Kondor,I.R. and Lafferty,J.D. (2002) Diffusion kernels on graphs and other discrete structures. In *Proceedings of the International Conference on Machine Learning*. Morgan Kaufmann, San Francisco, CA, pp. 315–322.

Lee,H. *et al.* (2006). Diffusion kernel-based logistic regression models for protein function prediction. *Omics J. Integrative Biol.*, **10**, 40–55.

Lee,I. *et al.* (2008) A single gene network accurately predicts phenotypic effects of gene perturbation in Caenorhabditis elegans. *Nat. Genet.*, **40**, 181–188.

Letovsky,S. and Kasif,S. (2003) Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, **19** (Suppl. 1), i197–i204.

Ma,X. *et al.* (2008) Mapping genetically compensatory pathways from synthetic lethal interactions in yeast. *PLoS ONE*, **3**, e1922.

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.

Qiu,J. and Noble,W.S. (2008) Predicting co-complexed protein pairs from heterogeneous data. *PLoS Comput. Biol.*, **4**, e1000054.

Schölkopf,B. *et al.* (2004) *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA.

Sharan,R. *et al.* (2007) Network-based prediction of protein function. *Mol. Syst. Biol.*, **3**, 88.

Smola,A.J. and Kondor,I.R. (2003) Kernels and regularization on graphs. In Schölkopf,B. and Warmuth,M.K. (eds) *Proceedings of the Annual Conference on Computational Learning Theory. Lecture Notes in Computer Science*. Springer, Heidelberg, pp. 144–158.

Stark,C. *et al.* (2006) BioGRID: a general repository for interaction datasets. *Nucleic Acids Res.*, **34**, D535–D539.

Tian,W. *et al.* (2008) Combining guilt-by-association and guilt-by-profiling to predict saccharomyces cerevisiae gene function. *Genome Biol.*, **9** (Suppl. 1), S7.

Tong,A. *et al.* (2004) Global mapping of the yeast genetic interaction network. *Science*, **303**, 808–813.

Tsuda,K. *et al.* (2005) Fast protein classification with multiple networks. *Bioinformatics*, **21** (Suppl. 2), ii59–ii65.

Ulitsky,I. and Shamir,R. (2007) Pathway redundancy and protein essentiality revealed in the Saccharomyces cerevisiae interaction networks. *Mol. Syst. Biol.*, **3**, 104.

Weston,J. *et al.* (2004) Protein ranking: from local to global structure in the protein similarity network. *Proc. Natl Acad. Sci. USA*, **101**, 6559–6563.

Ye,P. *et al.* (2005a) Commensurate distances and similar motifs in genetic congruence and protein interaction networks in yeast. *BMC Bioinformatics*, **6**, 270.

Ye,P. *et al.* (2005b) Gene function prediction from congruent synthetic lethal interactions in yeast. *Mol. Syst. Biol.*, **1**, 2005.0026.