Diffusion Component Analysis:
Unraveling Functional Topology in Biological Networks

Hyunghoon Cho[1], Bonnie Berger[1,2,*] and Jian Peng[1,2,3,*]

[1] Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA

[2] Department of Mathematics, MIT, Cambridge, MA, USA

[3] Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, IL, USA

[*] Corresponding authors: *bab@mit.edu* and *jianpeng@illinois.edu*

Abstract

Complex biological systems have been successfully modeled by biochemical and genetic interaction networks, typically gathered from high-throughput (HTP) data. These networks can be used to infer functional relationships between genes or proteins. Using the intuition that the topological role of a gene in a network relates to its biological function, local or diffusion-based "guilt-by-association" and graph-theoretic methods have had success in inferring gene functions. Here we seek to improve function prediction by integrating diffusion-based methods with a novel dimensionality reduction technique to overcome the incomplete and noisy nature of network data. In this paper, we introduce diffusion component analysis (DCA), a framework that plugs in a diffusion model and learns a low-dimensional vector representation of each node to encode the topological properties of a network. As a proof of concept, we demonstrate DCA's substantial improvement over state-of-the-art diffusion-based approaches in predicting protein function from molecular interaction networks. Moreover, our DCA framework can integrate multiple networks from heterogeneous sources, consisting of genomic information, biochemical experiments and other resources, to even further improve function prediction. Yet another layer of performance gain is achieved by integrating the DCA framework with support vector machines that take our node vector representations as features. Overall, our DCA framework provides a novel representation of nodes in a network that can be used as a plug-in architecture to other machine learning algorithms to decipher topological properties of and obtain novel insights into interactomes.[1]

---

[1]This paper was selected for oral presentation at RECOMB 2015 and an abstract is published in the conference proceedings.

1 Introduction

Cellular processes are executed through complex interactions between molecules, such as proteins, DNA, RNA and small compounds. Recently-developed high-throughput (HTP) experimental techniques, such as yeast two-hybrid screens and genetic interaction assays, have helped to identify molecular interactions in bulk and enabled researchers to piece together large-scale interaction networks from these data [1, 2, 3].

Protein-protein interaction and genetic interaction networks are the largest and most commonly-used datasets to study how molecules and their interactions determine the function of biological processes. However, networks generated from such data are often incomplete, noisy and difficult to interpret. Thus, we seek to design efficient algorithms to obtain accurate and comprehensive functional annotations of genes and proteins in these less-than-perfect networks.

Graph-theoretic methods have been proposed for the functional annotation of genes which use the intuition that interacting genes close together in network topology are more likely to perform similar functions [4, 5, 6, 7, 8, 9, 10, 11]. A type of random-walk diffusion algorithm, also known as random walk with restart (RWR), has been extensively studied in the context of biological networks and effectively applied to protein function prediction and enrichment analysis [4, 12, 13, 14, 15]. The key idea behind such algorithms is to propagate information along the network, in order to exploit both direct and indirect linkages between genes. Typically, a distribution of topological similarity is computed for each gene, in relation to other genes in the network, so that researchers can select the most related genes in the resulting distribution [12, 13, 14, 15] or, rather, select genes that share the most similar distributions [4]. Such diffusion-based methods have been shown to be useful for protein function prediction, mainly due to their ability to capture fine-grain topological properties, which can be less clear in the direct interactome neighborhood of a gene. Nevertheless, diffusion algorithms are still far from satisfactory, partially due to the noisy and incomplete nature of high-throughput data. Even the yeast protein interactome, the highest quality data gathered for any organism, includes a large number of false positive and false negative interactions [1, 16, 17, 18].

Dimensionality reduction techniques such as principle component analysis (PCA) have been effectively used for de-noising and improving accuracy in high dimensional biological data. The purpose of PCA is to reduce the dimensionality of a dataset by linearly projecting it onto a lower dimensional space, while retaining most of the original dataset's variance. In many predictive machine learning applications, using PCA to de-noise the input data has been shown to be effective because it makes the model more resistant to overfitting. However, to our best knowledge, little effort has been made in the spirit of dimensionality reduction and de-noising for improving diffusion-based function prediction approaches for biological networks. Here we seek to improve function prediction by integrating diffusion-based methods with a novel dimensionality reduction technique, specifically designed to capture the diffusion patterns of a network, to overcome the incomplete and noisy nature of network data.

In this paper, we propose Diffusion Component Analysis (DCA), a novel analytical framework that combines diffusion-based methods and dimensionality reduction to better extract topological network information in order to facilitate accurate functional annotation of genes or proteins. The key idea behind DCA is to obtain informative, but low-dimensional features, which encode the inherent topological properties of each node in the network. We first run a diffusion algorithm on a molecular network to obtain a distribution for each node which captures its relation to all other nodes in the network. We then approximate each of these distributions by constructing a multinomial logistic model, parametrized by low-dimensional feature vectors, for each node. Feature vectors of all nodes are obtained by minimizing the Kullback-Leibler (KL) divergence (relative entropy) between the diffusion and parameterized-multinomial logistic distributions. Akin

to PCA, which reveals the internal low-dimensional linear structure of the data that best explains the variance, DCA computes a low-dimensional vector-space representation for all nodes in the network such that the diffusion in the network can be best explained. Moreover, DCA can be naturally extended to integrate multiple heterogeneous networks by performing diffusion on separate networks and jointly optimizing feature vectors.

To evaluate the performance of DCA, we first applied DCA to predict protein function in yeast molecular interaction networks and achieved substantial improvement over existing methods. We next demonstrated the integrative capacity of DCA by using it to integrate heterogeneous networks based on genomic information, biochemical experiments and other resources from the STRING database and further improved functional annotation. In addition, we trained a support vector machine with our node-vector representation as features to be used as a metric for "distance between genes for improved protein function prediction, as SVMs performed better than k-nearest neighbor search. Overall, when tested on the STRING networks with third level functional annotations from MIPS, our DCA framework is able to achieve 71.29% accuracy, which is remarkably 12.31% higher than the previous state-of-the-art, diffusion-based method. Low-dimensional feature vector representation learned by DCA can readily be plugged into other existing machine learning algorithms to decipher functional properties of and obtain novel insights into interactomes.

2 Diffusion Component Analysis

Biological networks provide a useful topological structure over which information about certain nodes in a graph can be propagated to shed light on the properties of nodes that are less characterized. This relies on the assumption that the nodes that are "similar" or "close" to each other in the graph will tend to have similar properties, which is intuitively the case for biological functions or disease associations, where localized groups of interacting genes or proteins give rise to a specific molecular phenotype. Conventional approaches to finding nodes that are topologically associated with a given node involve either directly taking the neighborhood in the graph, or using localized diffusion models to select top nodes in the resulting distribution (which we refer to as *diffusion state*), or to select nodes with similar distributions. The diffusion-based methods are shown to be effective in various settings, mainly due to their ability to capture fine-grain topological properties that lie beyond the direct neighborhood of a node. Our DCA framework substantially advances previous diffusion-based methods by providing a way to extract topological information encoded in the diffusion states using a compact, low-dimensional vector representation of nodes in a graph (Figure 1).

2.1 Random walk with restart review

The random walk with restart (RWR) method has been well-established for analyzing network structures. By allowing the restart of random walk in each step with a probability, RWR can take into consideration both local and global topology within the network to identify the relevant or important nodes in the network. Let A denote the adjacency matrix of a (weighted) molecular interaction network $G = (V, E)$ with n nodes, each denoting a gene or a protein. Each entry $B_{i,j}$ in the transition probability matrix B, which stores the probability of a transition from node i to node j, is computed as

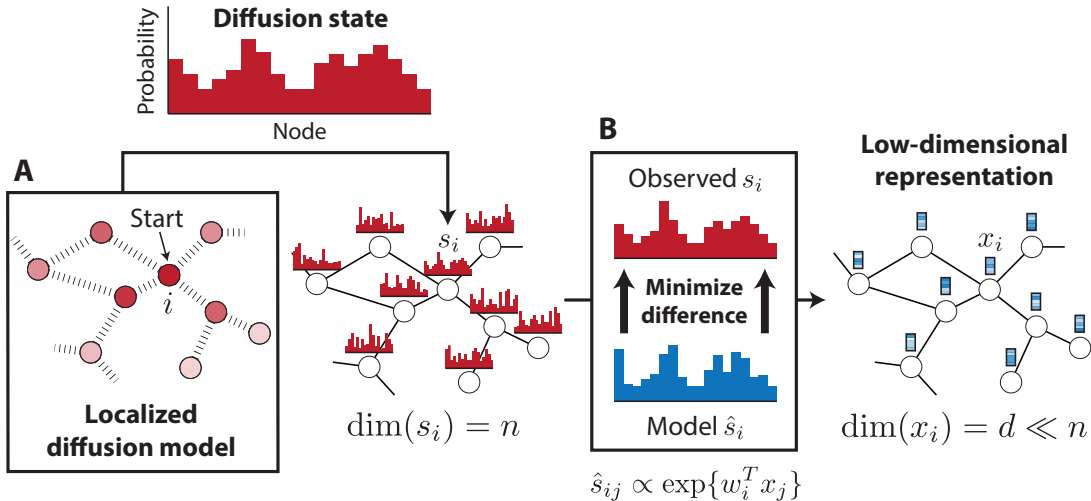$$B_{i,j} = \frac{A_{i,j}}{\sum_{j'} A_{i,j'}}.$$

Figure 1: **Diffusion Component Analysis.** (A) Random walks with restart are used to compute the diffusion state for each node in a network. (B) Low-dimensional feature vectors describing the topological properties of each node are obtained by minimizing the average KL-divergence between the observed diffusion states and the parameterized-multinomial logistic distributions.

Formally, the RWR from a node i is defined as:

$$s_i^{t+1} = (1 - p_r)s_i^t B + p_r e_i \tag{1}$$

where p_r is the probability of restart, controlling the relative influence of local and global topological information in the diffusion, with higher chances of restart placing greater emphasis on the local structure; e_i is a n-dimensional distribution vector with $e_i(i) = 1$ and $e_i(j) = 0, \forall j \neq i$; s^t is a n-dimensional distribution vector in which each entry holds the probability of a node being visited after t steps in the random walk. The first term in the above update corresponds to following a random edge connected to the current node, while the second term corresponds to restarting from the initial node i. At the fixed point of this iteration we obtain the stationary distribution s_i^∞. Consistent with a previous work [4], we define the *diffusion state* $s_i = s_i^\infty \in \Delta_n$ of each node i to the stationary distribution of random walk with restart (RWR) starting at each node, where Δ_n denotes the n-dimensional probability simplex. Intuitively, the jth element, s_{ij}, represents the probability of RWR starting at node i ending up at node j in equilibrium. When the diffusion states of two nodes are close to one another, it implies that they are in similar positions within the graph with respect to other nodes, which might suggest functional similarity. This insight provided the basis for several diffusion-based methods [4, 12, 19, 8] that aim to predict characteristics of genes or proteins by using the diffusion states to better capture topological associations. Instead of simply using the probability in the diffusion state, the diffusion state distance (DSD) approach, using $L1$ distances between diffusion states, achieved the state-of-the-art performance on predicting protein functions on yeast interactomes [4].

2.2 Novel dimensionality reduction

A key observation behind our approach is that the diffusion states obtained in this manner are still noisy, in part due to the low quality and high dimensionality of the original network data. Just a few missing or spurious interactions in the network can have significant impact on the result of

random walks, and the fact that most biological networks are incomplete greatly exacerbates this problem. Moreover, the high dimensionality of the diffusion state makes it not readily useful as topological features for network-based classification or regression tasks. With the goal of noise and dimensionality reduction, we approximate each diffusion state s_i with a multinomial logistic model based on a latent vector representation of nodes that uses far fewer dimensions than the original, n-dimensional state. Specifically, we compute the probability assigned to node j in the diffusion state of node i as

$$\hat{s}_{ij} := \frac{\exp\{w_i^T x_j\}}{\sum_{j'} \exp\{w_i^T x_{j'}\}}, \tag{2}$$

where $\forall i, w_i, x_i \in \mathbf{R}^d$ for $d \ll n$. Each node is given two vector representations, w_i and x_i. We refer to w_i as the context feature and x_i as the node feature of node i, both capturing the intrinsic topological properties in the network. If w_i and x_j are close in direction and with large inner product, node j should be frequently visited in the random walk starting from node i. Ideally, if the vector representation w and x is able to capture fine-grain topological properties, we can use it to retrieve genes with similar functions or use it as features for other network-based machine learning applications. While it is possible to enforce equality between these two vectors, decoupling them leads to a more manageable optimization problem and also allows our framework to be readily extended to the multiple network case, which is further described in the next section.

Given this model, we formulate the following optimization problem that takes a set of observed diffusion states $s = \{s_1, \ldots, s_n\}$ as input and finds the low-dimensional vector representation of nodes w and x that best approximates s according to the multinomial logistic model. To obtain w and x for all nodes, we use KL-divergence (relative entropy) as the objective function, which is a natural choice for comparing probability distributions, to guide the optimization:

$$\underset{w,x}{\text{minimize}} \; C(s, \hat{s}) = \frac{1}{n} \sum_{i=1}^{n} D_{\text{KL}}(s_i \| \hat{s}_i). \tag{3}$$

By writing out the definition of relative entropy and $\hat{s}$, we can express the objective as

$$C(s, \hat{s}) = \frac{1}{n} \sum_{i=1}^{n} \left[H(s_i) - \sum_{j=1}^{n} s_{ij} \left(w_i^T x_j - \log \left(\sum_{j'=1}^{n} \exp\{w_i^T x_{j'}\} \right) \right) \right], \tag{4}$$

where $H(\cdot)$ denotes the entropy. Similar to PCA which finds the low-dimensional linear structure of the data to best explain the data variance, DCA computes a low-dimensional vector-space representation for all nodes in the network data such that the diffusion on the network can be best explained.

To optimize this objective function, we computed the gradients with respect to the parameters w and x:

$$\nabla_{w_i} C(s, \hat{s}) = \frac{1}{n} \sum_{j=1}^{n} (\hat{s}_{ij} - s_{ij}) \, x_j, \tag{5}$$

$$\nabla_{x_i} C(s, \hat{s}) = \frac{1}{n} \sum_{j=1}^{n} (\hat{s}_{ji} - s_{ji}) \, w_j. \tag{6}$$

Both the objective function and the gradients can be computed in $O(n^2 d)$ runtime complexity. We use a standard quasi-Newton method L-BFGS [20] with these gradients to find the low-dimensional vector representations w and x corresponding to a local optimum of this optimization problem. We use uniform random numbers from $[-0.05, 0.05]$ to initialize the vectors and observed that the optimization can almost surely find good local optimal solutions.

4

2.3 Novel integration of heterogeneous networks

Our framework gives rise to a novel approach to integrating heterogeneous data sources. As huge amount of interaction data generated from wide variety of experimental and computational studies for molecular networks, integration approaches have been used to combine them to a network model and identify genes' functions [5, 8, 21, 22, 23, 24]. A common approach taken by previous work is based on weighting the networks in some manner to get an integrated network before analyzing the topological properties of each node [5, 23, 25]. For example, in the STRING database, confidence scores from heterogeneous sources including co-expression relationship, genomic context, experimental interactions and literature evidence, are combined by the Bayes method: $p_{i,j} = 1 - \prod_k(1 - p_{i,j}^{(k)})$, where $p_{i,j}^{(k)} \in [0,1]$ is the probability confidence of interaction (i,j) from network $k \in \{1, 2, ..., K\}$ derived by the same data source. However, by collapsing the networks into a single representative network, we lose important network-specific patterns. Performing random walk directly on the integrated network can be problematic, since edges from multiple sources are mixed.

Here, we naturally extend our DCA approach to integrate network data from diverse sources. We first perform random walks on each individual network k separately and obtain network-specific diffusion states $s_i^{(k)}$ for each node i. We also construct the multinomial distribution $\hat{s}_{ij}^{(k)}$ from the following logistic model,

$$\hat{s}_{ij}^{(k)} := \frac{\exp\{w_i^{(k)T}x_j\}}{\sum_{j'}\exp\{w_i^{(k)T}x_{j'}\}}, \tag{7}$$

where for each node i in network k, we assign it a network-specific context vector representation $w_i^{(k)}$, which also encodes the intrinsic topological properties of network dataset k; for node features x, we allow them to be shared across all K networks. Finally, we jointly optimize the following objective function,

$$\underset{w,x}{\text{minimize}}\; C(s,\hat{s}) = \sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}D_{\text{KL}}(s_i^{(k)}\|\hat{s}_i^{(k)}), \tag{8}$$

by the quasi-Newton L-BFGS method. Note that it is possible to weight the divergence term of each network differently, but we give equal importance to each network in this work for simplicity.

2.4 Evaluation of DCA in protein function prediction

To assess the quality of vector representations obtained by DCA, we considered the task of protein function prediction from protein interaction networks. In protein-protein interaction networks, functional relationship between proteins is believed to correlate with the proximity among proteins in the network structure, since interacting proteins are often involved within the same biological processes. Thus using functional annotations from well characterized proteins to predict the function of unlabeled proteins can be effective if we have a good way to compute the topological proximity or similarity between proteins. Given a distance metric that well captures the topological similarity of proteins, we are able to predict function of a protein from the known functions of closest proteins ranked by this distance metric. This makes function prediction a good example to evaluate how good the topological and diffusion information are encoded by DCA.

To predict protein function using DCA, we compute the similarity between any two proteins in the network based on their low-dimensional vector representations. For a pair of proteins i and j,

we used the cosine distance between their node features as following.

$$D_{cos}(i, j) = 1 - \frac{x_i^T x_j}{\|x_i\| \|x_j\|} \tag{9}$$

After the distances are computed, we use weighted majority vote with ten closest proteins to assign functions to unlabeled proteins. Unless otherwise noted, we use majority vote with cosine distances to carry out function prediction based on our vector representation of proteins.

Furthermore, since DCA gives low-rank features for each protein in the network, they can be readily used as feature vectors for machine learning algorithms in various applications, such as classification or clustering. To demonstrate the potential of DCA as a plug-in framework for existing machine learning algorithms, we formulate the protein function prediction task as a multi-label classification problem and applied an off-the-shelf support vector machine (SVM). Since there are proteins with multiple functional annotations, we train a SVM for each function category and then assign functions to a protein with annotations with highest output values from SVM. We denote this method as DCA-SVM.

3 Results

Here we evaluate the ability of our DCA framework to uncover functional relationships in the inter-actome of yeast, which is one of the few organisms for which a comprehensive annotation of proteins is available. With the combination of noise reduction via dimensionality reduction, improved integration of multiple heterogenous networks, and the use of support vector machines, our DCA framework is able to achieve 71.29% accuracy with five-fold cross-validation on the STRING networks with third level functional annotations from MIPS, which is remarkably 12.31% higher than the previous state-of-the-art diffusion state distance (DSD) method. The details of our experiments follow.

3.1 Networks and annotations

We obtained a collection of protein-protein interaction (PPI) networks of yeast from the STRING [25] database v9.1, which is based on a variety of data sources, including high-throughput interaction assays, curated PPI databases, and conserved coexpression. We excluded the network constructed from text mining of academic literature to prevent confounding caused by links based on functional similarity. The resulting collection consisted of six heterogenous networks over 6,400 proteins, with the number of edges varying from 1,361 to 314,013. Every edge in these networks is associated with a weight between 0 and 1 representing the probability of edge presence, which we factor into the calculation of transition probabilities in the random walk process. For methods that take a single network as input, we used STRING's approach to integrate the networks: we assign $p_{i,j} = 1 - \prod_k (1 - p_{i,j}^{(k)})$ as the probability of each edge in the combined network, where $p_{i,j}^{(k)}$ is the probability associated with the same edge in network k. 536,207 edges were produced after the integration.

Furthermore, we built a PPI network of 6,071 yeast proteins based on 135,374 physical interactions reported in the BioGRID [26] database v3.2.117. We considered every edge in the database to be undirected for simplicity, although our method can be easily applied to directed networks as well. For pairs of nodes with multiple edges, we merged the redundant edges and assigned a weight corresponding to the number of merged edges to produce a weighted graph. We used this network as an additional test case for our framework.

To train and test function prediction methods, we obtained functional annotations for yeast proteins from the Munich Information Center For Protein Sequences (MIPS) Functional Catalogue [27]. The biological functions in this database are organized in a three-layered hierarchy, where the top level (MIPS1) consists of 17 functions, the second (MIPS2) consists of 74 functions, and the third (MIPS3) consists of 154 functions. Importantly, each protein can have more than one associated function. All of our experiments were repeated for each level of annotations to test the methods in different challenging settings, with MIPS3 being the most difficult due to the largest number of candidate labels.

3.2 Performance metrics

For each function prediction method, we repeatedly held out 20% of the annotated proteins as the validation set and used the remaining 80% to predict their functions. We used two prediction performance metrics that are commonly used in this field. First, we calculate the *accuracy* by assigning top function to each protein in the validation set and measuring how often our prediction is one of the known functions of the protein. Second, we consider the micro-average *F1 score.* To calculate this metric, we assign top α predictions to each protein, construct a 2-by-2 contingency table for each function by treating it as a binary classification problem, and compute the F1 score using the combined table where each entry is summed across all functions. We used $\alpha = 3$ for the results presented in this paper, following previous work [4, 28].

3.3 DCA substantially improves functional annotation

The results for protein function prediction using the STRING network (without text mining) are summarized in Figure 2. We used restart probability of 0.5 for random walks and obtained 500-dimensional vector representation for each node via DCA. We observed that our performance is stable for different values of restart probability between 0.5 and 0.9. We compared our framework to two other baseline methods. The first baseline is *neighborhood majority vote* (NMV) [28], in which predictions are produced by prioritizing the functions based on the number of times they tag proteins that are directly connected to the target protein. This method is not able to make use of any information associated with proteins that are more than one hop away from the target. This issue is addressed by the second baseline we considered: *diffusion state distance* (DSD), which is the current state-of-the-art, diffusion-based method [4]. Here, the diffusion state of each node is obtained in a similar fashion as DCA. Then for each target protein, k most similar proteins, based on the L1 distance metric imposed over the diffusion states, are taken for a majority vote. In particular, each protein casts a vote for each of its associated functions with weight equal to the reciprocal of the distance to the target protein, and finally the functions are prioritized by the sum of these votes. We used $k = 10$, following the original work that introduced this method [4]. The diffusion states can capture longer-range topological properties in addition to the local neighborhood, which allows DSD to achieve a large improvement in prediction performance over NMV, as can be seen in Figure 2. However, through our DCA frameworks substantial advances, it achieves a significant improvement over DSD on all three levels of MIPS (Figure 2). This result highlights that, by reducing each diffusion state into a low-dimensional vector using DCA, one can capture functional associations between proteins more accurately.

One of the strengths of DCA is its ability to easily and principally integrate topological information from multiple networks. This integration is achieved by learning a canonical vector representation of proteins that explains their diffusion patterns across all networks, while simultaneously capturing network-specific effects using context-specific feature vectors (see 2.3). Intuitively, this
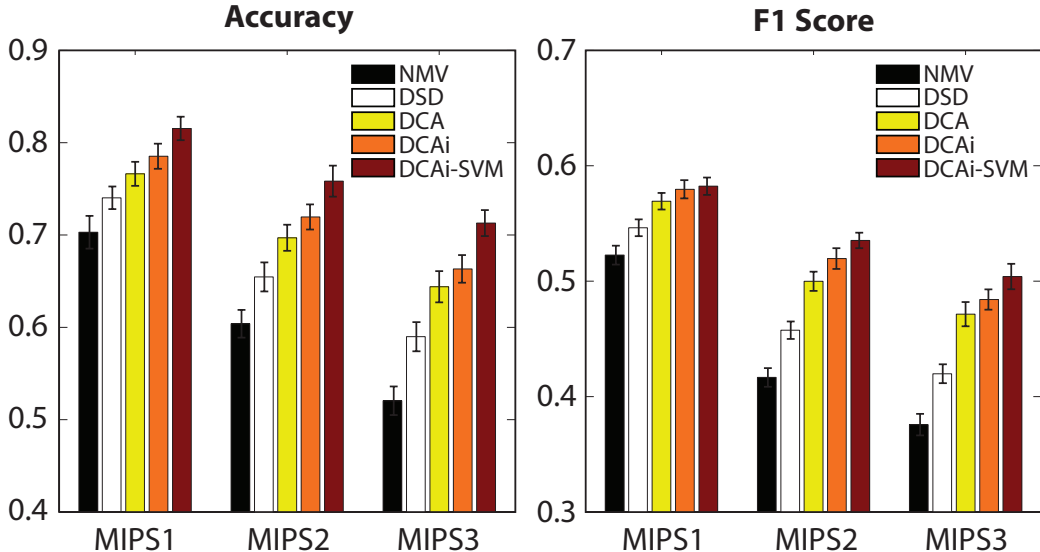
Figure 2: **Protein function prediction performance on yeast STRING networks.** DCA (diffusion component analysis) has substantially greater accuracy than both NMV (neighbor majority vote) and DSD (diffusion state distance) across all three MIPS levels. We used 500 dimensions for DCA, DCAi, and DCAi-SVM.

approach is able to capture more fine-grain topological patterns which would not be visible if we were to flatten the networks into a single representative network, as most do, before analyzing the topology. By integrating the STRING networks with our approach and using the canonical vectors to guide the majority vote for function prediction based on pairwise cosine distances (denoted as DCAi), we achieve improvement on all levels of MIPS on top of DCA with STRING's integration (Figure 2). This result suggests that functional relations can be better explained by individually modeling diffusion patterns within each network.

To demonstrate the potential of DCA as a plug-in framework for machine learning algorithms, we cast the protein function prediction task as a multi-label classification problem and applied an off-the-shelf support vector machine (SVM) toolbox, LIBSVM [29], with the node representation from DCA as input features. We used the radial basis function kernel and performed a nested five-fold cross-validation within the training data to select the optimal parameters. The candidate values for parameter selection were $\{0.5, 0.25, 0.125\}$ for the variance parameter g and $\{0.5, 1, 2\}$ for the cost parameter C. With this pipeline, we trained a binary classifier for each function and obtained a per-class probability score for each protein in the validation set. Note that an SVM does not provide probability outputs by itself, but LIBSVM supports an additional routine to train a logistic model on top of the SVM output via cross-validation to estimate the probabilities. These probability scores are then used to prioritize candidate functions for each protein, after which the prediction performance was evaluated. Simply by using an off-the-shelf SVM in this manner with the vectors from DCAi (denoted as DCAi-SVM), we were able to obtain yet another substantial performance gain in function prediction on all three levels of MIPS, as shown in Figure 2. While one could also apply an SVM with the original, high-dimensional diffusion states as feature representation, we found that it performs significantly worse than training the SVM on the DCA vectors with STRING's integration (for fair comparison), denoted as DCA-SVM, which suggests
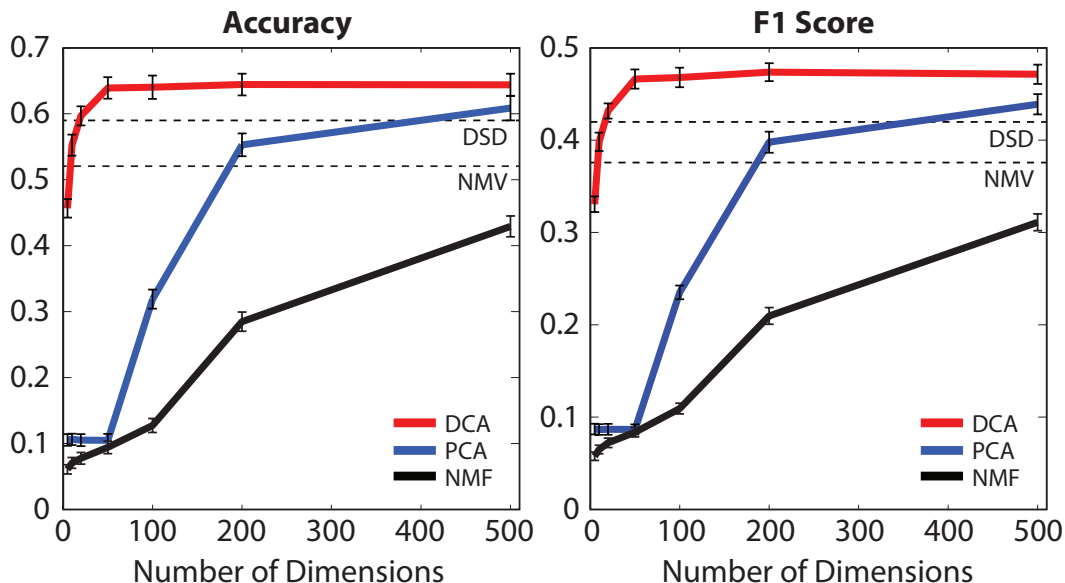
Figure 3: **Comparison with other dimensionality reduction methods.** DCA (diffusion component analysis) has much greater accuracy than principal component analysis (PCA) and non-negative matrix factorization (NMF) for different numbers of dimensions, especially for fewer dimensions. The results are based on majority vote prediction where the neighborhood is defined with cosine distance. STRING networks and MIPS3 annotations are used. Baseline performances of NMV and DSD are shown as constants.

that DCA is able to find a higher quality feature representation of the input data using much fewer dimensions (Supplementary Table 1). Furthermore, DCA-SVM performs worse than DCAi-SVM with our novel integration, implying that each of the improvements we made is necessary to achieve the best performance overall (Supplementary Table 1).

After observing the potential in using fewer dimensions to encode topological information, we then asked whether other conventional approaches to dimensionality reduction can achieve similar performance improvements. To this end, we compared DCA to principal component analysis (PCA) and non-negative matrix factorization (NMF) for different numbers of dimensions, and the results based on majority vote prediction based on cosine distances are summarized in Figure 3. We found that DCA significantly outperforms both PCA and NMF overall. The performance gap is even more noticeable for fewer number of dimensions (≤ 200), which suggests that DCA is effective at extracting meaningful topological associations even in a highly constrained setting, where the conventional approaches tend to suffer, presumably due to the fact that they aim to find a linear embedding to explain variance within the data without directly modeling the diffusion states as probability distributions. In addition, our results show that the performance of DCA is stable over a wide range of values for the number of dimensions, implying that our framework is robust to overfitting. Surprisingly, DCA can achieve performance comparable to DSD, which used $n = 6400$ dimensions, using only 20 dimensions. Furthermore, we found that training SVMs with feature representations from PCA and NMF perform relatively poorly, which provides additional evidence that DCA finds a good feature representation using a small number of dimensions (Supplementary Table 1).

In summary, compared to DSD, which is the current state-of-the-art function prediction method based on diffusion, we were able to achieve a 12.31% increase in accuracy and 8.43% increase in

the micro-average F1 score (see 3.2) on MIPS3, resulting in 71.29% accuracy and 50.40% F1 score, by using the full DCA framework, which combines our dimensionality reduction, a novel network integration approach and SVM classifiers. This is a notable improvement, given that DSD improves upon NMV — a naive method solely based on local topology — by only 6.94% in accuracy and 4.41% in F1 score on the same data. We also observed improved performance over DSD in a different yeast PPI network, constructed only from physical interactions in the BioGRID database (Supplementary Table 2).

4 Discussion

We have presented Diffusion Component Analysis (DCA), a novel analytical method for biological network analysis. DCA exploits the topology of the network by diffusion and then computes a low-dimensional but highly informative vector representation for nodes in the network to approximate the diffusion information. DCA can be naturally extended to integrating multiple heterogeneous networks by first performing diffusion on separate networks and then jointly optimizing feature vectors.

We have demonstrated DCA in exploiting functional topology in molecular networks by accurately predicting protein function from molecular interaction network data. We have demonstrated substantial improvements over the state-of-the-art diffusion-based approach and other dimensionality reduction approaches. DCA accurately encodes both local and global topological properties for all genes in the network, and thus is very useful for predicting gene functions that cannot be simply inferred from the direct interactome. Furthermore, low-dimensional vectors computed by DCA are a highly informative feature for describing the role of proteins in terms of functional topology, making them readily incorporable into existing machine learning methods.

In the future, we plan to pursue further improvements in function prediction with DCA. For example, in this work, we simply used the straightforward implementation of the supervised multilabel SVM, which is not ideal and prone to overfitting for species (e.g. Human and Mouse) with numerous functional labels but only sparsely annotated genes. We believe that functional label similarity [7, 30] and gene ontology hierarchy [31, 32, 33] can be applied with DCA to address the challenges of supervised multi-label function prediction. Other features, such as sequence or evolutionary information, which provide extra information, can be naturally combined with DCA. We also hope to explore other network-based applications, including identifying functional modules, network-guided enrichment analysis, discovering new gene ontology terms from molecular network data and network-based approaches for cancer genomics [12, 34, 35, 36, 37, 38].

References

[1] Charlotte M Deane, Lukasz Salwinski, Ioannis Xenarios, and David Eisenberg. Protein interactions two methods for assessment of the reliability of high throughput observations. *Molecular & Cellular Proteomics*, 1(5):349–356, 2002.

[2] Eric M Phizicky and Stanley Fields. Protein-protein interactions: methods for detection and analysis. *Microbiological reviews*, 59(1):94–123, 1995.

[3] Assen Roguev, Marianna Wiren, Jonathan S Weissman, and Nevan J Krogan. High-throughput genetic interaction mapping in the fission yeast Schizosaccharomyces pombe. *Nature methods*, 4(10):861–866, 2007.

[4] Mengfei Cao, Christopher M. Pietras, Xian Feng, Kathryn J. Doroschak, Thomas Schaffner, Jisoo Park, Hao Zhang, Lenore J. Cowen, and Benjamin J. Hescott. New directions for

diffusion-based network prediction of protein function: Incorporating pathways with confidence. *Bioinformatics*, 30(12), 2014.

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

[6] Elena Nabieva, Kam Jim, Amit Agarwal, Bernard Chazelle, and Mona Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21(SUPPL. 1), 2005.

[7] Hua Wang, Heng Huang, and Chris Ding. Function-function correlated multi-label protein function prediction over interaction networks. *Journal of computational biology*, 20(4):322–343, 2013.

[8] Peggy I Wang, Sohyun Hwang, Rodney P Kincaid, Christopher S Sullivan, Insuk Lee, and Edward M Marcotte. RIDDLE: reflective diffusion and local extension reveal functional associations for unannotated gene sets via proximity in a gene network. *Genome biology*, 13(12):R125, 2012.

[9] Hon Nian Chua, Wing-Kin Sung, and Limsoon Wong. Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. *Bioinformatics*, 22(13):1623–1630, 2006.

[10] Tijana Milenkoviæ and Nataša Pržulj. Uncovering biological network function via graphlet degree signatures. *Cancer informatics*, 6:257, 2008.

[11] Ulas Karaoz, TM Murali, Stan Letovsky, Yu Zheng, Chunming Ding, Charles R Cantor, and Simon Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2888–2893, 2004.

[12] Enrico Glaab, Anaïs Baudot, Natalio Krasnogor, Reinhard Schneider, and Alfonso Valencia. EnrichNet: network-based gene set enrichment analysis. *Bioinformatics*, 28(18):i451—-i457, 2012.

[13] Sebastian Köhler, 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.

[14] Chung-Shou Liao, Kanghao Lu, Michael Baym, Rohit Singh, and Bonnie Berger. IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253—-i258, 2009.

[15] Saket Navlakha and Carl Kingsford. The power of protein interaction networks for associating genes with diseases. *Bioinformatics*, 26(8):1057–1063, 2010.

[16] Sean R Collins, Patrick Kemmeren, Xue-Chu Zhao, Jack F Greenblatt, Forrest Spencer, Frank C P Holstege, Jonathan S Weissman, and Nevan J Krogan. Toward a comprehensive atlas of the physical interactome of Saccharomyces cerevisiae. *Molecular & cellular proteomics : MCP*, 6(3):439–450, 2007.

[17] Anne-Claude Gavin, Patrick Aloy, Paola Grandi, Roland Krause, Markus Boesche, Martina Marzioch, Christina Rau, Lars Juhl Jensen, Sonja Bastuck, Birgit Dümpelfeld, Angela Edelmann, Marie-Anne Heurtier, Verena Hoffman, Christian Hoefert, Karin Klein, Manuela Hudak, Anne-Marie Michon, Malgorzata Schelder, Markus Schirle, Marita Remor, Tatjana Rudi, Sean Hooper, Andreas Bauer, Tewis Bouwmeester, Georg Casari, Gerard Drewes, Gitte Neubauer,

Jens M Rick, Bernhard Kuster, Peer Bork, Robert B Russell, and Giulio Superti-Furga. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440(7084):631–636, 2006.

[18] Nevan J Krogan, Gerard Cagney, Haiyuan Yu, Gouqing Zhong, Xinghua Guo, Alexandr Ignatchenko, Joyce Li, Shuye Pu, Nira Datta, Aaron P Tikuisis, Thanuja Punna, José M Peregrín-Alvarez, Michael Shales, Xin Zhang, Michael Davey, Mark D Robinson, Alberto Paccanaro, James E Bray, Anthony Sheung, Bryan Beattie, Dawn P Richards, Veronica Canadien, Atanas Lalev, Frank Mena, Peter Wong, Andrei Starostine, Myra M Canete, James Vlasblom, Samuel Wu, Chris Orsi, Sean R Collins, Shamanta Chandran, Robin Haw, Jennifer J Rilstone, Kiran Gandi, Natalie J Thompson, Gabe Musso, Peter St Onge, Shaun Ghanny, Mandy H Y Lam, Gareth Butland, Amin M Altaf-Ul, Shigehiko Kanaya, Ali Shilatifard, Erin O'Shea, Jonathan S Weissman, C James Ingles, Timothy R Hughes, John Parkinson, Mark Gerstein, Shoshana J Wodak, Andrew Emili, and Jack F Greenblatt. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*, 440(7084):637–643, 2006.

[19] Kathy Macropol, Tolga Can, and Ambuj K Singh. RRW: repeated random walks on genome-scale protein networks for local cluster discovery. *BMC bioinformatics*, 10(1):283, 2009.

[20] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

[21] Lourdes Peña Castillo, Murat Tasan, Chad L Myers, Hyunju Lee, Trupti Joshi, Chao Zhang, Yuanfang Guan, Michele Leone, Andrea Pagnani, Wan Kyu Kim, and Others. A critical assessment of Mus musculus gene function prediction using integrated genomic evidence. *Genome biology*, 9(Suppl 1):S2, 2008.

[22] Sohyun Hwang, Seung Y Rhee, Edward M Marcotte, and Insuk Lee. Systematic prediction of gene function in Arabidopsis thaliana using a probabilistic functional gene network. *Nature protocols*, 6(9):1429–1442, 2011.

[23] Insuk Lee, Shailesh V Date, Alex T Adai, and Edward M Marcotte. A probabilistic functional network of yeast genes. *Science*, 306(5701):1555–1558, 2004.

[24] Bo Wang, Aziz M Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nature methods*, 11(3):333–337, 2014.

[25] Andrea Franceschini, Damian Szklarczyk, Sune Frankild, Michael Kuhn, Milan Simonovic, Alexander Roth, Jianyi Lin, Pablo Minguez, Peer Bork, Christian von Mering, and Others. STRING v9. 1: protein-protein interaction networks, with increased coverage and integration. *Nucleic acids research*, 41(D1):D808—-D815, 2013.

[26] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. BioGRID: a general repository for interaction datasets. *Nucleic acids research*, 34(suppl 1):D535—-D539, 2006.

[27] Andreas Ruepp, Alfred Zollner, Dieter Maier, Kaj Albermann, Jean Hani, Martin Mokrejs, Igor Tetko, Ulrich Güldener, Gertrud Mannhaupt, Martin Münsterkötter, and Others. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic acids research*, 32(18):5539–5545, 2004.

[28] Benno Schwikowski, Peter Uetz, and Stanley Fields. A network of protein-protein interactions in yeast. *Nature biotechnology*, 18(12):1257–1261, 2000.

[29] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[30] Emre Sefer and Carl Kingsford. Metric labeling and semi-metric embedding for protein annotation prediction. In *Research in computational molecular biology*, pages 392–407. Springer, 2011.

[31] Wyatt T Clark and Predrag Radivojac. Information-theoretic evaluation of predicted ontological annotations. *Bioinformatics*, 29(13):i53–i61, 2013.

[32] Sara Mostafavi and Quaid Morris. Using the Gene Ontology hierarchy when predicting gene function. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 419–427. AUAI Press, 2009.

[33] Xiaoyu Jiang, Naoki Nariai, Martin Steffen, Simon Kasif, and Eric D Kolaczyk. Integration of relational and hierarchical network information for protein function prediction. *BMC bioinformatics*, 9(1):350, 2008.

[34] Alex J Cornish and Florian Markowetz. SANTA: quantifying the functional content of molecular networks. *PLoS computational biology*, 10(9):e1003808, 2014.

[35] Janusz Dutkowski, Michael Kramer, Michal A Surma, Rama Balakrishnan, J Michael Cherry, Nevan J Krogan, and Trey Ideker. A gene ontology inferred from molecular networks. *Nature biotechnology*, 31(1):38–45, 2013.

[36] Matan Hofree, John P Shen, Hannah Carter, Andrew Gross, and Trey Ideker. Network-based stratification of tumor mutations. *Nature methods*, 10(11):1108–15, 2013.

[37] Igor Ulitsky and Ron Shamir. Identification of functional modules using network topology and high-throughput data. *BMC systems biology*, 1:8, 2007.

[38] Nir Yosef, Lior Ungar, Einat Zalckvar, Adi Kimchi, Martin Kupiec, Eytan Ruppin, and Roded Sharan. Toward accurate reconstruction of functional protein networks. *Molecular systems biology*, 5(1), 2009.

Supplementary Table 1

Summary of function prediction performance on STRING networks for all three levels of MIPS annotations.

NMV denotes Neighborhood Majority Vote. DSD denotes Diffusion State Distance method of Cao et al., 2014.

DS-SVM denotes training SVM on the full n-dimensional diffusion states.

The numbers appended to DCA (diffusion component analysis), PCA (principal component analysis), NMF (nonnegative matrix factorization) represent the number of dimensions used for each dimensionality reduction method. DCAi incorporates our novel network integration approach.

STRING	MIPS1				MIPS2				MIPS3			
	Accuracy		F1 score		Accuracy		F1 score		Accuracy		F1 score	
	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
NMV	70.30%	1.77%	52.26%	0.82%	60.39%	1.50%	41.66%	0.81%	52.05%	1.55%	37.57%	0.93%
DSD	74.03%	1.22%	54.63%	0.73%	65.46%	1.57%	45.75%	0.76%	58.99%	1.58%	41.98%	0.82%
DS-SVM	76.26%	1.16%	56.56%	0.63%	69.20%	1.82%	48.87%	0.51%	64.80%	1.78%	45.69%	0.69%
DCA-5	65.32%	1.29%	48.45%	0.73%	53.30%	1.44%	36.95%	0.92%	45.66%	1.41%	33.06%	0.85%
DCA-10	72.05%	1.24%	52.85%	0.72%	62.48%	1.54%	43.74%	0.95%	55.25%	1.60%	39.82%	0.99%
DCA-20	74.28%	1.31%	54.39%	0.77%	65.52%	1.53%	46.35%	0.93%	59.68%	1.45%	43.16%	0.82%
DCA-50	76.77%	1.41%	56.39%	0.78%	69.09%	1.57%	49.23%	0.91%	63.92%	1.63%	46.62%	1.04%
DCA-100	76.55%	1.26%	56.69%	0.73%	69.43%	1.51%	49.25%	0.97%	64.03%	1.77%	46.79%	1.06%
DCA-200	76.89%	1.35%	56.92%	0.80%	69.78%	1.32%	49.85%	0.90%	64.44%	1.65%	47.37%	0.97%
DCA-500	76.63%	1.30%	56.93%	0.72%	69.70%	1.41%	49.99%	0.83%	64.40%	1.82%	47.14%	1.05%
DCA-500-SVM	78.75%	0.84%	58.05%	0.62%	74.55%	1.08%	52.36%	0.64%	69.91%	1.82%	48.73%	0.59%
PCA-5	31.74%	1.25%	29.02%	0.74%	16.38%	1.05%	12.98%	0.63%	10.53%	0.89%	8.70%	0.58%
PCA-10	31.21%	1.45%	28.86%	0.80%	16.42%	1.08%	13.06%	0.59%	10.62%	0.86%	8.65%	0.58%
PCA-20	31.34%	1.37%	28.85%	0.63%	16.23%	1.09%	13.06%	0.55%	10.51%	0.91%	8.68%	0.59%
PCA-50	31.51%	1.51%	28.95%	0.69%	16.22%	1.09%	13.00%	0.55%	10.47%	0.98%	8.66%	0.55%
PCA-100	54.90%	1.44%	42.68%	0.75%	39.97%	1.41%	28.89%	0.81%	31.89%	1.45%	23.51%	0.74%
PCA-200	71.34%	1.35%	52.45%	0.79%	61.65%	1.49%	42.76%	0.87%	55.29%	1.73%	39.78%	1.14%
PCA-500	74.06%	1.11%	54.47%	0.75%	67.51%	1.49%	47.00%	0.91%	60.87%	1.73%	43.89%	1.10%
PCA-500-SVM	74.29%	1.84%	54.13%	0.67%	66.92%	1.20%	45.94%	0.86%	59.73%	1.87%	40.75%	0.60%
NMF-5	25.05%	1.34%	25.15%	0.71%	9.94%	0.82%	9.68%	0.49%	6.07%	0.69%	5.75%	0.45%
NMF-10	25.85%	1.37%	26.05%	0.82%	11.12%	0.91%	10.26%	0.55%	7.05%	0.82%	6.49%	0.46%
NMF-20	26.51%	1.11%	26.55%	0.59%	11.49%	1.05%	11.07%	0.60%	7.75%	0.89%	7.21%	0.51%
NMF-50	27.02%	1.36%	27.17%	0.69%	13.45%	1.09%	12.32%	0.55%	9.41%	0.95%	8.36%	0.51%
NMF-100	31.08%	1.25%	29.72%	0.81%	16.53%	1.17%	14.42%	0.62%	12.72%	1.06%	10.92%	0.58%
NMF-200	47.92%	1.48%	39.19%	0.75%	33.58%	1.40%	25.01%	0.77%	28.48%	1.46%	20.96%	0.90%
NMF-500	59.45%	1.48%	45.39%	0.77%	48.07%	1.55%	34.06%	0.82%	42.92%	1.58%	31.10%	0.91%
NMF-500-SVM	62.57%	1.20%	45.59%	0.46%	51.35%	1.74%	33.93%	0.72%	44.94%	2.19%	31.26%	1.36%
DCAi-500	78.53%	1.36%	57.96%	0.79%	71.95%	1.36%	51.96%	0.90%	66.33%	1.50%	48.41%	0.88%
DCAi-500-SVM	81.54%	1.27%	58.23%	0.75%	75.83%	1.68%	53.53%	0.67%	71.29%	1.41%	50.40%	1.10%

14

Supplementary Table 2

Summary of function prediction performance on BioGRID physical PPI networks for all three levels of MIPS annotations.

NMV denotes Neighborhood Majority Vote. DSD denotes Diffusion State Distance method of Cao et al., 2014.
DS-SVM denotes training SVM on the full n-dimensional diffusion states.
The numbers appended to DCA (diffusion component analysis), PCA (principal component analysis), NMF (nonnegative matrix factorization) represent the number of dimensions used for each dimensionality reduction method.

BioGRID	MIPS1				MIPS2				MIPS3			
	Accuracy		F1 score		Accuracy		F1 score		Accuracy		F1 score	
	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
NMV	51.82%	1.51%	43.53%	0.86%	44.33%	1.60%	32.30%	0.97%	40.60%	1.51%	30.91%	0.81%
DSD	67.42%	1.26%	51.15%	0.77%	55.32%	1.48%	40.97%	0.91%	51.08%	1.59%	38.64%	0.99%
DS-SVM	67.86%	1.54%	51.69%	0.82%	56.26%	2.09%	41.03%	0.99%	50.09%	1.44%	37.86%	0.68%
DCA-5	54.67%	1.41%	43.54%	0.78%	42.77%	1.43%	31.07%	0.93%	38.10%	1.32%	27.94%	0.81%
DCA-10	63.09%	1.38%	48.21%	0.75%	50.95%	1.51%	37.16%	0.90%	46.88%	1.62%	34.89%	1.03%
DCA-20	66.29%	1.37%	50.59%	0.73%	54.93%	1.71%	40.29%	0.95%	51.54%	1.51%	38.26%	0.94%
DCA-50	67.04%	1.52%	51.38%	0.83%	56.63%	1.33%	41.47%	0.94%	52.70%	1.59%	39.53%	1.08%
DCA-100	66.73%	1.42%	51.49%	0.80%	56.56%	1.55%	41.39%	0.97%	52.69%	1.62%	39.65%	1.06%
DCA-200	67.23%	1.65%	51.54%	0.89%	57.29%	1.40%	41.75%	0.88%	52.88%	1.63%	39.81%	1.05%
DCA-500	67.47%	1.45%	51.92%	0.75%	57.54%	1.32%	41.87%	0.81%	52.83%	1.67%	39.85%	1.00%
DCA-500-SVM	71.75%	1.37%	53.57%	1.13%	59.92%	1.01%	43.40%	0.83%	54.56%	1.31%	39.91%	1.10%
PCA-5	38.20%	1.57%	32.94%	0.83%	20.15%	1.31%	16.65%	0.72%	14.71%	1.12%	12.14%	0.65%
PCA-10	43.78%	1.48%	36.22%	0.79%	26.99%	1.40%	20.91%	0.76%	20.43%	1.24%	16.08%	0.76%
PCA-20	53.49%	1.40%	41.66%	0.76%	38.91%	1.43%	28.59%	0.87%	32.10%	1.59%	24.56%	0.94%
PCA-50	59.60%	1.64%	46.10%	1.06%	46.57%	1.72%	34.25%	1.03%	42.29%	1.57%	31.66%	1.03%
PCA-100	62.50%	1.44%	48.47%	0.78%	50.90%	1.52%	37.15%	0.89%	46.42%	1.59%	35.18%	1.06%
PCA-200	65.14%	1.38%	49.64%	0.81%	53.36%	1.54%	39.22%	0.88%	48.53%	1.71%	37.21%	1.10%
PCA-500	66.31%	1.42%	50.60%	0.82%	56.02%	1.35%	40.76%	0.82%	50.94%	1.50%	38.62%	0.97%
PCA-500-SVM	67.12%	1.17%	50.46%	0.63%	53.93%	1.06%	38.28%	0.69%	47.22%	1.44%	34.88%	0.93%
NMF-5	27.65%	1.27%	27.41%	0.80%	12.21%	0.90%	11.30%	0.58%	7.15%	0.89%	6.82%	0.47%
NMF-10	31.33%	1.27%	28.73%	0.73%	14.54%	0.97%	12.75%	0.57%	9.77%	0.99%	8.78%	0.58%
NMF-20	33.48%	1.23%	30.81%	0.68%	17.79%	1.16%	15.54%	0.64%	12.31%	1.05%	11.11%	0.63%
NMF-50	37.95%	1.44%	33.72%	0.82%	22.97%	1.12%	19.50%	0.70%	19.01%	1.04%	16.01%	0.72%
NMF-100	56.73%	1.55%	45.31%	0.85%	45.38%	1.44%	34.18%	0.89%	42.38%	1.61%	32.41%	1.19%
NMF-200	62.45%	1.58%	47.86%	0.88%	51.10%	1.52%	37.44%	0.95%	46.71%	1.53%	35.70%	1.04%
NMF-500	64.43%	1.54%	49.36%	0.84%	53.84%	1.42%	38.86%	0.90%	49.45%	1.49%	37.12%	0.95%
NMF-500-SVM	63.23%	1.12%	48.21%	0.80%	51.98%	1.68%	37.40%	0.98%	47.09%	1.18%	34.44%	0.80%