Bioinformatics, 38(13), 2022, 3395–3406
https://doi.org/10.1093/bioinformatics/btac322
Advance Access Publication Date: 16 May 2022
Original Paper

Systems biology

GLIDER: function prediction from GLIDE-based neighborhoods

Kapil Devkota[1], Henri Schmidt[1], Matt Werenski [1], James M. Murphy[2], Mert Erden[1], Victor Arsenescu [1] and Lenore J. Cowen [1,*]

[1]Department of Computer Science, Tufts University, Medford, MA 02155, USA and [2]Department of Mathematics, Tufts University, Medford, MA 02155, USA

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on July 28, 2021; revised on February 10, 2022; editorial decision on May 2, 2022; accepted on May 10, 2022

Abstract

Motivation: Protein function prediction, based on the patterns of connection in a protein–protein interaction (or association) network, is perhaps the most studied of the classical, fundamental inference problems for biological networks. A highly successful set of recent approaches use random walk-based low-dimensional embeddings that tend to place functionally similar proteins into coherent spatial regions. However, these approaches lose valuable local graph structure from the network when considering only the embedding. We introduce GLIDER, a method that replaces a protein–protein interaction or association network with a new graph-based similarity network. GLIDER is based on a variant of our previous GLIDE method, which was designed to predict missing links in protein–protein association networks, capturing implicit local and global (i.e. embedding-based) graph properties.

Results: GLIDER outperforms competing methods on the task of predicting GO functional labels in cross-validation on a heterogeneous collection of four human protein–protein association networks derived from the 2016 DREAM Disease Module Identification Challenge, and also on three different protein–protein association networks built from the STRING database. We show that this is due to the strong functional enrichment that is present in the local GLIDER neighborhood in multiple different types of protein–protein association networks. Furthermore, we introduce the GLIDER graph neighborhood as a way for biologists to visualize the local neighborhood of a disease gene. As an application, we look at the local GLIDER neighborhoods of a set of known Parkinson's Disease GWAS genes, rediscover many genes which have known involvement in Parkinson's disease pathways, plus suggest some new genes to study.

Availability and implementation: All code is publicly available and can be accessed here: https://github.com/kap-devkota/GLIDER.

Contact: cowen@cs.tufts.edu

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

1 Introduction

Function prediction, the prediction of appropriate GO functional labels for a protein of unknown function, based on the patterns of connection in a protein–protein interaction (PPI) (or association) network, is perhaps the most studied of the classical, fundamental biological network inference problems. Recently, embedding-based methods for function prediction have received a great deal of attention (Nelson *et al.*, 2019), where most of these methods logically decompose into two steps: (i) an *embedding step*, where the network is replaced by its low-dimensional representation, while retaining its implicit network features and (ii) a *classification step* where these embeddings are used for the purpose of multi-label classification, through the use of an appropriate machine-learning classifier.

For creating meaningful embeddings, network propagation, or diffusion methods have been found to be particularly effective (Cowen *et al.*, 2017). Indeed, it has been shown that for many types of protein–protein association (PPA) data (Choobdar *et al.*, 2019), diffusion-based methods are highly successful at creating embeddings that organize proteins based on their functions. Once an embedding that captures the implicit functional information is constructed, the entire machine-learning standard toolbox becomes available to perform the classification step; where one can employ anything from simple k nearest-neighbors (Cao *et al.*, 2014) ($k$nn), to support vector machines (Cho *et al.*, 2016), or beyond (Grover and Leskovec, 2016). However, some local information encoded directly in the links of the original network is destroyed by the embedding.

In this article, our focus is on creating a new graph-based similarity network that retains some of this local information while still giving us the global expressive power of embedding methods. Our similarity measure is a variant of GLIDE (Devkota *et al.*, 2020), a method, we introduced in 2020 for a different classical biological problem (link prediction). GLIDE combines a simple local score that captures relationships in the dense core, with a diffusion-based embedding that encapsulates the network structure in the periphery, creating a *quasi-kernel* (we note because of the local score component, the GLIDE similarity metric is not exactly a kernel, so we refer to it as a *quasi-kernel* in this article). Our new method, which we call GLIDER, uses a variant of GLIDE to create a new similarity network from the original graph. We demonstrate that this newly created GLIDER network has more functionally enriched local neighborhoods than the original network such that the application of a simple $k$nn classifier produces a significantly improved function prediction performance.

We show that this GLIDER network, equipped with an ordinary $k$nn classifier, produces state-of-the art GO functional label prediction, in each of the molecular function (MF), biological process (BP) and cellular component (CC) portions of the GO hierarchy, competing favorably even with methods using more sophisticated machine-learning classifiers. We compare its performance against competitor methods in cross-validation on a heterogeneous collection of four human PPA networks derived from the 2016 DREAM Disease Module Identification Challenge (Choobdar *et al.*, 2019). This heterogeneous network collection includes both classical PPI networks, a signaling network and a co-expression network (see Section 3.1 below for specific details). Moreover, we also used three composite protein association networks derived from the latest version of STRING(v 11.5) (Szklarczyk *et al.*, 2021) to compare the function prediction capabilities of our method with the existing state-of-the-art algorithms (the properties and the construction details of these three composite STRING networks are also provided in Section 3.1).

Additionally, the generation of functionally enriched neighborhoods facilitated by GLIDER naturally provides a graph-based visualization of protein functional neighborhoods. We examine the local GLIDER graph neighborhood for a set of GWAS genes from previous studies implicated in the pathology of Parkinson's Disease (PD) (Blauwendraat *et al.*, 2020; Nalls *et al.*, 2014, 2019). In the neighborhoods of proteins in the GLIDER-constructed network of these known PD genes, we find many genes already implicated in PD disease pathways, and also identify some interesting new candidates. We find that GLIDER is a powerful tool to explore function in biological networks.

2 Materials and methods

2.1 GLIDER

The GLIDER method constructs a graph that is based on a variant of our GLIDE similarity score (Devkota *et al.*, 2020). GLIDE combines a simple local score that captures relationships in the dense core, with a diffusion-based embedding that encapsulates the network structure in the periphery. For GLIDER networks, we pair a local score based on common neighbors with global score UDSED$^\gamma$, a variant of DSED$^\gamma$ from the original GLIDE paper [for comparative performance of alternative choices, including the local score L3 (Kovács *et al.*, 2019), that was best in many scenarios for the link prediction problem (Devkota *et al.*, 2020), but under-performs in our present context, see Supplementary Tables S2–S13]. We define these scores next.

Definition 2.1. DSE$^\gamma$ Embedding [from Devkota *et al.* (2020)]. Let $P \in \mathbb{R}^{N\times N}$ be a Markov transition matrix computed from a graph G with a unique stationary distribution π. Then, the DSE$^\gamma$ embedding is:

$$DSE^\gamma = I + \sum_{t=1}^{\infty} \gamma^t (P - W)^t, \tag{1}$$

where W is a constant matrix, whose rows are copies of the stationary distribution π and γ is a parameter satisfying $0 < \gamma \leq 1$, which is used to control the contribution of larger time-steps in the computation of the embedding. We set $\gamma = 1$ in all our experiments, as suggested in Devkota *et al.* (2020).

Definition 2.2. Global Score: UDSED$^\gamma$ Distance.

If DSE$^\gamma(p)$ and DSE$^\gamma(q)$ represent the DSE$^\gamma$ embeddings for the nodes p and q, respectively, we consider the (un-normalized) L2 distance between their DSE$^\gamma$ embeddings. Formally, this can be written as

$$\mathrm{UDSED}^\gamma(p,q) = \sqrt{\sum_{k=1}^{N} (\mathrm{DSE}^\gamma(p)_k - \mathrm{DSE}^\gamma(q)_k)^2}. \tag{2}$$

Definition 2.3. Local Score: Common Weighted Normalized.

Given nodes $p, q \in G$, the Common Weighted Normalized (CWN) score is

$$\mathrm{CWN}(p,q) = \frac{\sum_{r\in\mathcal{N}_p\cap\mathcal{N}_q}(w_{p,r} + w_{q,r})}{\sqrt{k(p)k(q)}},$$

where for any node $x \in G$, $\mathcal{N}_x$ is the neighbor set of x, $w_{x,y}$ is the weight of the edge (x, y) and $k(x)$ represents the weighted degree of x. Note that this is slightly different from the CW metric described in Devkota *et al.* (2020), because of the square roots in the denominator.

2.1.1 GLIDE score

Just as in Devkota *et al.* (2020), we define the following score between each pair of nodes:

$$GLIDE(p,q) = \exp\left(\frac{\alpha \cdot global(x_i,x_j)}{global(x_i,x_j) + \beta}\right) local(x_i,x_j) + global(x_i,x_j),$$

where GLIDER chooses $local(p,q) = \mathrm{CWN}(p,q)$ and $global(p,q) = 1/\mathrm{UDSED}^\gamma(p,q)$. We choose the default values of α and β as suggested by Devkota *et al.* (2020) ($\alpha = 0.1, \beta = 1000$), where these choices for α and β makes the local embedding dominant for ranking, while the global embedding is used to break ties and order nodes with the same strong local score. For the CWN local score, if nodes have no common neighbors, the first term is 0 and only the global score is used.

2.1.2 The construction of the GLIDER network

Consider the complete graph on the nodes of the network, with edges weighted by their GLIDE score. The GLIDER network only retains a subset of these edges of high similarity, as follows: let $g_{\max}$ denote the GLIDE weight of the most similar node (and thus the heaviest edge) to node g. Let $G_{\min}$ denote the minimum value of $g_{\max}$ over all the nodes, i.e.

$$G_{\min} = \min_{g\in V} g_{\max}. \tag{3}$$

The construction of GLIDER(G) follows immediately by adding any node-pairs in V whose GLIDE score is greater than or equal to $G_{\min}$ (see Figure 1). Note that, the value of $G_{\min}$ is an intrinsic property of the original network and the GLIDE parameters, so no additional parameters need to be specified to generate GLIDER(G) from G.

2.2 Knn-based function prediction using the GLIDER network

For each node r, set $k_r = \min(d(r), k)$, where $d(r)$ denotes the degree of r in the GLIDER network, and k is a parameter of the method. Our function prediction method is simply a Majority Vote (MV) (Schwikowski *et al.*, 2000) of all the labels of all the labeled nodes in

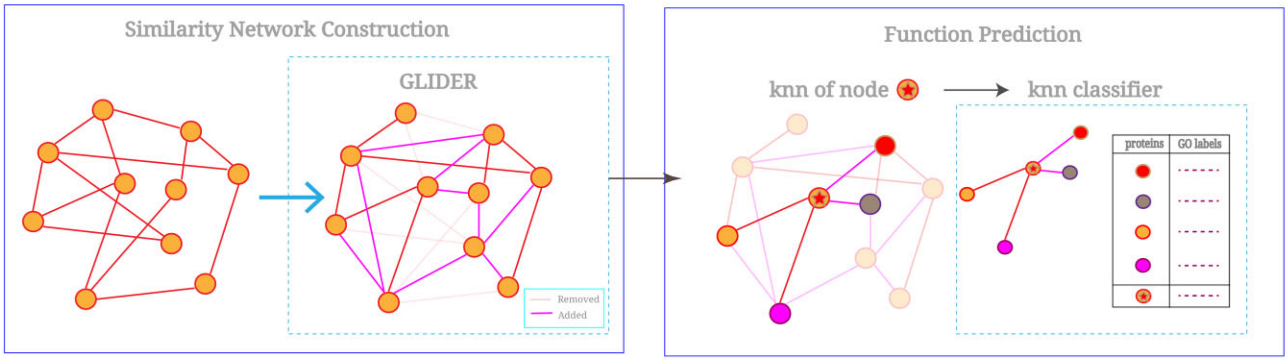

Fig. 1. Working schematic of GLIDER-*k*nn. The original graph is transformed into GLIDER(*G*) both adding and deleting edges. Then for each node (e.g. the starred node), the *k*-closest direct neighbors in GLIDER(*G*) vote for all their GO labels (created with BioRender.com)

the k_r-GLIDER neighborhood. If q in p's k_r-GLIDER neighborhood from the training set has multiple labels, q will vote for each of its labels with equal weight. p then is assigned all the labels that are above a given confidence threshold.

Let L be a function that given a protein p, returns $L(p)$, the set of functional labels associated with it. Given a version of GLIDE (with local and global measures fixed), let W_p denote the set consisting of the k_p closest GLIDE neighbors to p. Then, given a confidence threshold τ ($0 \leq \tau \leq 1$), GLIDER-*k*nn returns a list of functional labels of p as presented in Algorithm 1 (see also Figure 1). We remark that we extend the MV framework to multi-label function predictions by retaining labels that get second or third place votes (up to a confidence threshold τ). This will not change a percent accuracy metric that considers only the winning label (but allows us also measure performance of GLIDER and competing methods in a more complex framework, see Section 3.3).

2.2.1 Searching for the optimal value of *k*

Our experiments across the DREAM and STRING networks (see Supplementary Tables S2–S13) show that GLIDER-*k*nn is fairly robust to choice of k, and we can recommend setting k between 15 and 35 to get reasonable results on human networks, where we present results for GLIDER-25nn in Tables 2–4, and for ease of visualization, results for GLIDER-15 or GLIDER-20 in Section 4.2. In general, however, the choice of k plays an important role in the performance of GLIDER. In practice, the best k will depend both on the topology of the network, but also how well it has already been functionally annotated. In order to set k in a principled way that is robust for a variety of network settings, we use the training data to estimate the threshold of GLIDER-neighbors that still contain functional information as follows:

1. Construct GLIDER(G) from G.
2. Compute the average degree of nodes in GLIDER(G), call it d_{avg}. We consider potential settings of k between 1 and d_{avg}. We tried 20 different values for k for each network (see Supplementary Material for details).
3. Perform GLIDER-*k*nn on the *training set* proteins for each k; choose the value of k that maximizes the average accuracy on the training set in leave one out cross-validation.

Note that this k is the only parameter that GLIDER-*k*nn needs to accomplish its function predictions.

2.3 Competing methods

We consider the following competing function prediction methods:

Algorithm 1 GLIDER-*k*nn

Input: Protein p of unknown function, W_p = set of the k_p closest neighbors to p in GLIDER(G), where $G = (V, E)$ is the original graph, τ (a confidence threshold)
Output: A set L_p of predicted functional labels for p.

```
function GLIDER-knn($p, W_p, \tau$)
    Let $F$ be a set of all functional labels.
    For each $f \in F$, let vote($f, W_p$) count the number of times $f$ is present as a label in the proteins of $W_p$.
    Let $\text{vote}_{\max} = \max_f \text{vote}(f, W_p)$.
    Initialize $L_p = \emptyset$.
    $\forall f \in F$, if $\frac{\text{vote}(f)}{\text{vote}_{\max}} > \tau$ add $f$ to $L_p$.
    return $L_p$
end function
```

2.3.1 Simple MV

To label a node in the test set, this method simply has all direct neighbors in the training set vote for each of their labels, and assigns the node the label that receives the most votes (Schwikowski *et al.*, 2000). We generalize this method to also give a weighted confidence to second and third place labels (and so on), similar as in Lazarsfeld *et al.* (2021), in order to compute some of the performance measures we describe below. In particular, we divide the number of neighbors that vote for a label by the total number of voting neighbors, in order to give a confidence between 0 and 1 for each label appearing at least once among neighboring nodes (all other GO labels are voted with confidence 0).

2.3.2 Diffusion state distance-based KNN method

Distance-based KNN (DSD-*k*nn) (Cao *et al.*, 2013, 2014) is a kernel based function prediction method that uses random walks across multiple time-steps to compute a specialized network embedding called Diffusion State Embedding [actually, we use the variant that uses L2 distance instead of L1 distance, as in Cowen *et al.* (2021)]. After the embedding is produced, we can use the Gaussian Kernel to compute the similarity between two node embeddings in the network. We select K of the nearest nodes by their DSD similarity score, and have them vote on the node's function, in a manner similar to the MV method, above. After running DSD-*k*nn on different values of K (results provided in Supplementary Tables S14–S16), we found best results fairly stable for K in the range 20–35, so used $K = 25$ [rather than the recommended setting of $K = 10$ in Cao *et al.* (2014)] for comparative results below.

We also return a confidence in exactly the same way as described in Section 2.3.1 above.

2.3.3 node2vec method

The node2vec algorithm of Grover and Leskovec (2016) learns a low-dimensional embedding for nodes in a graph by optimizing a neighborhood-preserving objective. The algorithm accommodates various definitions of network neighborhoods by simulating biased random walks, utilizing hyperparameters (p and q) that must be trained for each network. After obtaining the node embeddings, a one-vs-rest logistic regression classifier is used to infer the function annotations of unlabeled nodes. Note that because node2vec is automatically a one-vs-rest logistic regression, the classifier simultaneously predicts multiple labels with confidence scores. We fixed the hyperparameters to be consistent with the optimal specifications outlined in Grover and Leskovec (2016): window-size = 10, num-walks = 10, dimension = 100, p, $q = 1$.

2.3.4 deepNF method

deepNF is a network fusion method based on multimodal deep autoencoders (MDA) to extract high-level features of proteins from multiple heterogeneous interaction networks (Gligorijević *et al.*, 2018). This method, which uses Random Walk with Restart to obtain a high dimensional structural information of the network(s), passes it to an MDA, resulting in a low-dimensional node representation. Function prediction from this low-dimensional representation is then done through one-vs-rest SVM classifier. As above, we automatically get multiple labels with confidence scores. We used default deepNF settings to generate the deepNF embeddings (MDA Hidden Dims = [1000, 500, 1000]).

2.3.5 MASHUP (single network)

MASHUP (Cho *et al.*, 2016), though designed for multiple networks, can also be used in a single network setting. The MASHUP network embedding is constructed by running a localized network diffusion process on the network to obtain the distribution for each node, followed by a dimension-reduction step. Similar to the deepNF method, MASHUP uses a one-vs-rest SVM classifier on the obtained low-dimensional embedding for function prediction. As with node2vec and deepNF, the classifier automatically produces multiple label predictions with confidence scores. We set the size of the reduced dimension to be 1000, which is within the recommended range of settings (5–10% of the network), as outlined in Cho *et al.* (2016). Also, we found the computation of the true MASHUP embedding to be infeasible for larger networks, so we instead used its SVD approximation, as described in Cho *et al.* (2016).

2.3.6 GLIDER-MASHUP

As seen from the description above, MASHUP embeds a graph, and then uses a one-vs-rest SVM classifier on the obtained low-dimensional embedding for function prediction. We wondered if replacing the original graph with the GLIDER graph, and then putting MASHUP's embedding and SVM pipeline downstream, would improve on our simple GLIDER-$k$nn classifier. Here, we have described how we set the k in $k$nn for GLIDER-$k$nn above; MASHUP sets its SVM parameters using linear weights based on the training data in a supervised manner as well. We show below that GLIDER-MASHUP does not improve on MASHUP in accuracy and $F1$ score, but it does in average Resnik score (see below, and further discussions in the Section 4.1.2).

3 Experimental setup

3.1 Networks

We test the efficacy of the similarity networks constructed through GLIDER on the four different benchmark networks from the recent DREAM disease module identification challenge (Choobdar *et al.*, 2019), and the latest version of the STRING human network [version 11.5, Szklarczyk *et al.* (2021)]. These human PPI and PPA networks are highly heterogeneous; DREAM1 is a heterogeneous PPA network derived from STRING (Szklarczyk *et al.*, 2015), DREAM2 is a more classical PPI derived from the Inweb database (Li *et al.*, 2017), DREAM3 is a signaling network derived from OmniPath (Türei *et al.*, 2016) (in the DREAM challenge, DREAM3 was presented as a directed network, but for this work, we considered an undirected version where all directed edges were made automatically bi-directional) and DREAM4 is a co-expression network based on Affymetrix HG-U133 Plus 2 arrays extracted from the GEO46. We summarize the graph properties of these networks in Supplementary Table S1. Additionally, in Supplementary Section S4, we use the GLIDER neighborhood measure to further explore natural differences in functional neighborhood cluster size for the different DREAM networks.

For evaluation using the STRING database, we extracted three sets of interactions from the STRING human network to generate three composite PPI networks. The first network, which we refer to as STRING-E, contains only the interactions labeled 'experimental', where all the associations involve actual physical binding of proteins. The second network, denoted as STRING-ED, contains interactions that are labeled either 'experimental' or 'database' (STRING labels physical interactions as 'database' if they are obtained from curated sources). The third network, referred to as STRING-EDC, further adds protein co-expression data into the 'STRING-ED'. The network properties of the three composite STRING networks are provided in the Supplementary Table S1.

3.2 Functional labels

We used GO Functional Labels for *Homo sapiens* (version: 2021-02-01, using the python package goatools). We considered the GO labels from each of the root hierarchies separately: MF, BP and CC, pruning both the most general and the most specific GO-terms as follows. We first removed GO-terms that are less than distance 5 in shortest path distance from their root node. We also removed GO-terms if the number of proteins annotated by that label is below 50. Table 1 shows the number of GO labels that satisfies the above restrictions for DREAM1–4 and the composite STRING networks.

3.3 Evaluation

We used three different evaluation metrics to compare performance of GLIDER and its competitors in 5-fold cross-validation. We compared the single top prediction using the oldest, classical simple percent accuracy measure. However, as is now the standard in the CAFA challenges, Jiang *et al.* (2016), Radivojac *et al.* (2013) and Zhou *et al.* (2019) recommend considering functional multi-label methods. We use two statistics in this regard: a hierarchy unaware $F1^*$ method (that nonetheless can capture label predictions at different specificities), and in order to take least common ancestors on the GO hierarchical DAG into account, we also utilized a Resnik-derived similarity score (Zhao and Wang, 2018), that generalizes to sets of genes as in Jiang *et al.* (2016).

3.3.1 Evaluation method 1: percent accuracy

This metric simply measures the percent of nodes whose top predicted functional label is correct, meaning it is among the set of true functional labels assigned to that node.

Table 1. The number of GO labels having their shortest path distance from the root nodes ≥5, and annotating at least 50 proteins, for DREAM1–4, STRING networks and GO hierarchies: MF, BP and CC

Networks	GO hierarchies		
	MF	BP	CC
DREAM1	45	272	86
DREAM2	38	218	72
DREAM3	28	120	31
DREAM4	38	213	71
STRING-E	47	277	89
STRING-ED	47	278	90
STRING-EDC	47	278	90

3.3.2 Evaluation method 2: hierarchy agnostic $F1^*$ method

This evaluation metric, which corresponds to the protein-centric evaluation method in the CAFA challenge (Radivojac *et al.*, 2013; Zhou *et al.*, 2019), scores a multi-label function prediction set, but still ignores the hierarchical nature of the GO annotations while scoring predictions. For a particular protein i, let T_i be the set representing its true GO annotation and $P_i(\tau)$ represent the set of GO annotations predicted by the Function Prediction method with likelihood greater than the confidence threshold τ. Then, we can compute the precision and recall for the protein i at the threshold τ as

$$\mathrm{prec}_i(\tau) = \frac{|P_i(\tau) \cap T_i|}{|P_i(\tau)|}, \tag{4}$$

$$\mathrm{recall}_i(\tau) = \frac{|P_i(\tau) \cap T_i|}{|T_i|}. \tag{5}$$

The average precision and recall for a particular confidence threshold τ is:

$$\mathrm{prec}(\tau) = \frac{1}{M}\sum_{i=1}^{M} \mathrm{prec}_{\alpha_\tau(i)}(\tau), \tag{6}$$

$$\mathrm{recall}(\tau) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{recall}_i(\tau), \tag{7}$$

where α_τ represents the set of all proteins, which have at least one GO annotation predicted at the confidence interval τ ($\alpha_\tau(i)$ represents its *ith* member), M is the size of the set $\alpha_\tau(i)$ and N is the total number of proteins in the test set.

We can then compute the $F1$ score at confidence τ, and $F1^*$ as

$$F1(\tau) = 2\frac{\mathrm{prec}(\tau)\cdot\mathrm{recall}(\tau)}{\mathrm{prec}(\tau)+\mathrm{recall}(\tau)}, \tag{8}$$

$$F1^* = \max_\tau F1(\tau). \tag{9}$$

3.3.3 Evaluation method 3: Resnik similarity metric

This metric models the hierarchical nature of the GO by introducing the information content of a GO-term (Jiang *et al.*, 2016) in the context of its ancestors. Let ℓ be a GO-term and L be the subgraph generated by all its ancestor labels, including ℓ. The information content of ℓ is defined formally as

$$i(\ell) = -\log(Pr(L)), \tag{10}$$

where the joint probability $Pr(L)$ is computed as

$$Pr(L) = \prod_{v\in L} Pr(v|\mathcal{P}(v)). \tag{11}$$

The term $Pr(v|\mathcal{P}(v))$, v being a GO-term and $\mathcal{P}(v)$ representing the parents of v, denotes the probability that we get v from $\mathcal{P}(v)$ after further ontological specialization. Expression (9) can be further simplified using (10) to obtain

$$i(\ell) = -\sum_{v\in L} Pr(v|\mathcal{P}(v)), \tag{12}$$

$$= \sum_{v\in L} ia(v). \tag{13}$$

The term $ia(v)$, referred to as information accretion of the annotation v, denotes the increase in the information obtained through the addition of child GO-term (v) to the set of its parent terms (or $\mathcal{P}(v)$).

Resnik similarity ($\mathrm{res}_{\mathrm{tt}}$) between two GO-terms, x and y, is

$$\mathrm{res}_{\mathrm{tt}}(\mathrm{x},\mathrm{y}) = i(\mathrm{lca}(x,y)), \tag{14}$$

where $\mathrm{lca}(x,y)$ represents the least common ancestor between x and y.

We next extend this similarity measure between GO-terms to a similarity metric between two sets of GO-terms, using the averaging scheme outlined in Pandey *et al.* (2008). Let X and Y be two GO-sets; then the Average Resnik Score (written as $\mathrm{res}_{ss}(X,Y)$) can be computed as in Pandey *et al.* (2008):

$$\mathrm{res}_{\mathrm{ss}}(X,Y) = \frac{\sum_{x\in X, y\in Y}\mathrm{res}_{tt}(x,y)}{|X||Y|}. \tag{15}$$

Let Q be the set containing all the test proteins, T_q be the true GO-terms and $P_q(\tau)$ be the predicted GO-terms at the confidence interval τ, for a protein $q\in Q$. Then, we compute the Resnik score as:

$$\mathrm{RES} = \max_\tau \frac{1}{|Q|}\sum_{q\in Q}\mathrm{res}_{\mathrm{ss}}(T_q, P_q(\tau)). \tag{16}$$

4 Results

4.1 Best local and global GLIDER variant

We tested alternative variants of the local and global GLIDE score, by evaluating GLIDER-*k*nn performance on the 4 DREAM networks. Furthermore, we used the parameter selection method described in Section 2.3.1 to choose for the optimal k from the range of options listed in the Supplementary Material (see Supplementary Section S6).

Complete results for BP, MF and CC hierarchies over DREAM1–4 networks appear in the Supplementary Tables S2–S13. Interestingly, we find the UDSED$^\gamma$ version performs significantly better than the DSED$^\gamma$ score from the original GLIDE paper. In addition, CWN versions of GLIDER were either slightly or significantly better than L3 versions (depending on the network and GO hierarchy). It is particularly interesting that the choice of CWN over L3 scores mostly improved the scores for function prediction. This is in contrast to what we found for the link prediction problem (Devkota *et al.*, 2020), where incorporation of details of interconnection structure as suggested by Kovács *et al.* (2019), helped improve performance in many settings. When all that is required is functionally enriched local neighborhoods, rather than the exact interconnection structure, as in the setting of this article, we find the simple normalized common neighbors measure better correlates with functional enrichment.

Across the board, in all experiments, we find DREAM1–3 produced much more meaningful results than DREAM4, regardless of which of the four versions of GLIDER was used, or how k was set (and replicated in results of competitor methods, see below).

Finally, we observed how the choice of the optimal GLIDER neighborhoods, obtained from the training scheme described in Section 2.2.1, differs for different DREAM networks, across all the GLIDER settings and GO hierarchies. Our observations for the optimal k under different modalities appear in Supplementary Figure S4. We see that, on dense networks, like DREAM1 and DREAM2, a smaller k neighborhood value is better. This pattern though, did not repeat for DREAM4, which although being a relatively dense network, required more GLIDER-neighbors for its optimal functional enrichment. One of the reasons behind this might be that co-expression networks, like DREAM4, capture weaker functional coherence between proteins. It might also be that the DREAM4 network is uniquely noisy, even among the co-expression networks.

For sparser regulatory networks like DREAM3, the optimal setting of k was relatively high. However, unlike in DREAM4, we assume this is more due to the sparsity of the original network than a weak signal. In fact, DREAM3 is in some sense the opposite of DREAM4, being a more curated but sparser set of high-confidence associations. The performance results for DREAM3 are on par with PPI networks like DREAM2 under all GO settings, and similar settings of k gave similar performance for all three versions of the STRING networks we tested (see Tables 2–4).

4.1.1 Comparison with other function prediction methods

We tested GLIDER against all competing methods described in Section 2.3. Tables 2–4 show that GLIDER-*k*nn, regardless of the

Table 2. Accuracy, *F*1 and Resnik score results on DREAM1–4 and STRING composite networks for different function prediction methods, using the MF category of GO, reporting mean and standard deviation over 5-fold cross-validation

Network	Metric	GLIDER-*k*nn	GLIDER-25nn	Majority-Vote	DSD-*k*nn	node2vec	deepNF(S)	MASHUP(S)	GLIDER-MASHUP
DREAM1	Accuracy	**0.671±0.017**	0.643 ± 0.013	0.356 ± 0.022	0.451 ± 0.011	0.439 ± 0.016	0.182 ± 0.011	0.605 ± 0.020	0.600 ± 0.007
DREAM2		**0.421±0.006**	0.418 ± 0.006	0.314 ± 0.012	0.364 ± 0.015	0.247 ± 0.011	0.267 ± 0.012	0.386 ± 0.010	0.384 ± 0.006
DREAM3		**0.407±0.015**	0.392 ± 0.016	0.229 ± 0.016	0.365 ± 0.016	0.253 ± 0.019	0.197 ± 0.010	0.366 ± 0.014	0.374 ± 0.017
DREAM4		0.281 ± 0.013	0.244 ± 0.018	**0.288±0.018**	0.249 ± 0.015	0.091 ± 0.015	0.186 ± 0.018	0.201 ± 0.015	0.175 ± 0.020
STRING-E		**0.702±0.003**	0.685 ± 0.010	0.375 ± 0.003	0.379 ± 0.006	0.449 ± 0.009	0.382 ± 0.015	0.636 ± 0.007	0.625 ± 0.011
STRING-ED		**0.698±0.009**	0.685 ± 0.010	0.410 ± 0.008	0.438 ± 0.010	**0.474±0.005**	0.384 ± 0.018	0.659 ± 0.007	0.624 ± 0.010
STRING-EDC		**0.670±0.009**	0.664 ± 0.004	0.432 ± 0.008	0.387 ± 0.010	0.425 ± 0.018	0.362 ± 0.011	0.660 ± 0.014	0.619 ± 0.008
DREAM1	*F*1	**0.663±0.009**	0.615 ± 0.003	0.406 ± 0.010	0.463 ± 0.011	0.360 ± 0.009	0.187 ± 0.014	0.580 ± 0.009	0.573 ± 0.011
DREAM2		**0.416±0.010**	0.415 ± 0.008	0.332 ± 0.011	0.361 ± 0.006	0.248 ± 0.010	0.236 ± 0.015	0.379 ± 0.008	0.365 ± 0.007
DREAM3		**0.401±0.018**	0.386 ± 0.018	0.262 ± 0.008	0.360 ± 0.019	0.263 ± 0.012	0.228 ± 0.028	0.377 ± 0.014	0.378 ± 0.015
DREAM4		0.285 ± 0.008	0.250 ± 0.006	**0.296±0.006**	0.259 ± 0.008	0.127 ± 0.003	0.210 ± 0.015	0.212 ± 0.016	0.188 ± 0.009
STRING-E		**0.677±0.013**	0.655 ± 0.010	0.400 ± 0.011	0.391 ± 0.004	0.384 ± 0.008	0.324 ± 0.099	0.598 ± 0.004	0.599 ± 0.003
STRING-ED		**0.698±0.011**	0.664 ± 0.013	0.438 ± 0.008	0.456 ± 0.010	0.401 ± 0.007	0.327 ± 0.009	0.614 ± 0.012	0.605 ± 0.009
STRING-EDC		**0.666±0.011**	0.637 ± 0.010	0.452 ± 0.016	0.404 ± 0.011	0.348 ± 0.008	0.322 ± 0.009	0.625 ± 0.005	0.588 ± 0.012
DREAM1	Resnik	**2.451±0.049**	2.388 ± 0.013	1.629 ± 0.029	1.870 ± 0.035	1.254 ± 0.041	0.859 ± 0.015	1.934 ± 0.051	1.998 ± 0.042
DREAM2		**1.740±0.041**	1.740 ± 0.035	1.316 ± 0.056	1.469 ± 0.056	0.936 ± 0.011	0.770 ± 0.027	1.301 ± 0.038	1.353 ± 0.045
DREAM3		**1.580±0.054**	1.567 ± 0.038	1.081 ± 0.056	1.515 ± 0.051	1.023 ± 0.032	0.837 ± 0.047	1.296 ± 0.038	1.390 ± 0.061
DREAM4		**1.309±0.042**	1.213 ± 0.021	1.295 ± 0.019	1.234 ± 0.008	0.773 ± 0.018	0.871 ± 0.079	0.925 ± 0.016	0.909 ± 0.027
STRING-E		**2.670±0.012**	2.583 ± 0.022	1.557 ± 0.029	1.621 ± 0.054	1.343 ± 0.020	1.056 ± 0.021	1.985 ± 0.028	2.095 ± 0.033
STRING-ED		**2.633±0.056**	2.593 ± 0.050	1.720 ± 0.028	1.820 ± 0.035	1.435 ± 0.024	1.069 ± 0.009	2.100 ± 0.031	2.155 ± 0.034
STRING-EDC		**2.552±0.044**	2.520 ± 0.009	1.742 ± 0.034	1.683 ± 0.016	1.251 ± 0.020	1.074 ± 0.029	2.136 ± 0.031	2.107 ± 0.010

Note: Best performance bolded. All method parameters set as described in Section 2.

Table 3. Accuracy, *F*1 and Resnik score results on DREAM1–4 and STRING composite networks for different function prediction methods, using the BP category of GO, reporting mean and standard deviation over 5-fold cross-validation

Network	Metric	GLIDER-*k*nn	GLIDER-25nn	Majority-Vote	DSD-*k*nn	node2vec	deepNF(S)	MASHUP(S)	GLIDER-MASHUP
DREAM1	Accuracy	**0.561±0.010**	0.544 ± 0.007	0.381 ± 0.003	0.476 ± 0.008	0.352 ± 0.017	0.273 ± 0.015	0.534 ± 0.009	0.521 ± 0.016
DREAM2		0.366 ± 0.005	0.363 ± 0.008	0.314 ± 0.006	**0.372±0.006**	0.225 ± 0.015	0.215 ± 0.008	0.3562 ± 0.0103	0.334 ± 0.009
DREAM3		0.338 ± 0.024	0.333 ± 0.019	0.255 ± 0.018	**0.342±0.022**	0.208 ± 0.013	0.200 ± 0.012	0.347 ± 0.026	0.341 ± 0.014
DREAM4		0.179 ± 0.011	0.157 ± 0.007	**0.180±0.005**	0.164 ± 0.009	0.076 ± 0.006	0.093 ± 0.014	0.142 ± 0.014	0.146 ± 0.010
STRING-E		**0.545±0.015**	0.521 ± 0.007	0.375 ± 0.003	0.353 ± 0.007	0.351 ± 0.010	0.273 ± 0.016	0.504 ± 0.012	0.505 ± 0.005
STRING-ED		**0.602±0.002**	0.573 ± 0.008	0.418 ± 0.010	0.401 ± 0.011	0.417 ± 0.005	0.300 ± 0.010	0.568 ± 0.008	0.547 ± 0.010
STRING-EDC		**0.560±0.008**	0.545 ± 0.008	0.406 ± 0.011	0.345 ± 0.014	0.375 ± 0.011	0.282 ± 0.013	0.521 ± 0.016	0.529 ± 0.018
DREAM1	*F*1	**0.484±0.008**	0.461 ± 0.006	0.364 ± 0.010	0.410 ± 0.008	0.272 ± 0.010	0.259 ± 0.021	0.440 ± 0.010	0.444 ± 0.009
DREAM2		0.317 ± 0.006	**0.320±0.005**	0.285 ± 0.004	0.301 ± 0.003	0.212 ± 0.004	0.200 ± 0.008	0.308 ± 0.005	0.285 ± 0.006
DREAM3		**0.306±0.012**	0.302 ± 0.011	0.244 ± 0.001	0.301 ± 0.007	0.211 ± 0.005	0.185 ± 0.012	0.296 ± 0.173	0.278 ± 0.014
DREAM4		0.171 ± 0.010	0.158 ± 0.004	**0.174±0.001**	0.162 ± 0.004	0.088 ± 0.001	0.106 ± 0.009	0.136 ± 0.007	0.146 ± 0.010
STRING-E		**0.479±0.005**	0.439 ± 0.007	0.400 ± 0.011	0.319 ± 0.008	0.292 ± 0.003	0.240 ± 0.006	0.433 ± 0.005	0.428 ± 0.007
STRING-ED		**0.512±0.004**	0.487 ± 0.004	0.365 ± 0.003	0.355 ± 0.004	0.334 ± 0.001	0.281 ± 0.005	0.474 ± 0.002	0.468 ± 0.010
STRING-EDC		**0.498±0.007**	0.463 ± 0.005	0.359 ± 0.004	0.308 ± 0.003	0.284 ± 0.003	0.253 ± 0.005	0.479 ± 0.002	0.448 ± 0.007
DREAM1	Resnik	**2.663±0.025**	2.615 ± 0.007	2.164 ± 0.063	2.465 ± 0.069	1.162 ± 0.036	0.751 ± 0.038	1.751 ± 0.015	1.876 ± 0.038
DREAM2		1.964 ± 0.035	**1.965±0.030**	1.885 ± 0.018	1.944 ± 0.042	1.090 ± 0.013	0.906 ± 0.020	1.321 ± 0.004	1.394 ± 0.028
DREAM3		1.894 ± 0.042	1.651 ± 0.052	1.627 ± 0.036	**1.959±0.048**	1.123 ± 0.011	1.128 ± 0.029	1.346 ± 0.046	1.426 ± 0.078
DREAM4		1.144 ± 0.042	1.155 ± 0.030	1.145 ± 0.022	**0.177±0.024**	0.729 ± 0.020	0.783 ± 0.084	0.865 ± 0.020	0.873 ± 0.017
STRING-E		**2.625±0.043**	2.189 ± 0.031	1.557 ± 0.029	2.011 ± 0.034	1.207 ± 0.045	0.905 ± 0.024	1.767 ± 0.028	1.897 ± 0.049
STRING-ED		**2.883±0.027**	2.787 ± 0.031	02.249 ± 0.018	2.238 ± 0.044	1.317 ± 0.042	0.978 ± 0.020	1.920 ± 0.013	2.079 ± 0.042
STRING-EDC		**2.731±0.045**	2.652 ± 0.033	2.134 ± 0.053	1.960 ± 0.036	1.199 ± 0.041	1.091 ± 0.058	2.008 ± 0.060	2.039 ± 0.047

Note: Best performance bolded. All method parameters set as described in Section 2.

Table 4. Accuracy, *F*1 and Resnik score results on DREAM1–4 and STRING composite networks for different function prediction methods, using the CC category of GO, reporting mean and standard deviation over 5-fold cross-validation

Network	Metric	GLIDER-*k*nn	GLIDER-25nn	Majority-Vote	DSD-*k*nn	node2vec	deepNF(S)	MASHUP(S)	GLIDER-MASHUP
DREAM1	Accuracy	0.596 ± 0.005	**0.599±0.008**	0.567 ± 0.005	0.585 ± 0.008	0.374 ± 0.011	0.330 ± 0.017	0.526 ± 0.003	0.517 ± 0.013
DREAM2		0.529 ± 0.016	0.527 ± 0.015	0.494 ± 0.007	**0.533±0.006**	0.218 ± 0.018	0.230 ± 0.012	0.410 ± 0.009	0.378 ± 0.010
DREAM3		**0.601±0.016**	0.596 ± 0.017	0.504 ± 0.015	0.595 ± 0.009	0.318 ± 0.015	0.518 ± 0.016	0.501 ± 0.016	0.464 ± 0.016
DREAM4		**0.483±0.005**	0.471 ± 0.005	0.471 ± 0.012	0.477 ± 0.008	0.091 ± 0.006	0.248 ± 0.056	0.282 ± 0.010	0.247 ± 0.007
STRING-E		**0.626±0.013**	0.625 ± 0.012	0.554 ± 0.013	0.555 ± 0.008	0.369 ± 0.009	0.356 ± 0.013	0.549 ± 0.010	0.545 ± 0.006
STRING-ED		**0.635±0.003**	0.629 ± 0.005	0.575 ± 0.002	0.576 ± 0.009	0.382 ± 0.010	0.329 ± 0.012	0.565 ± 0.009	0.553 ± 0.006
STRING-EDC		**0.626±0.007**	0.625 ± 0.009	0.567 ± 0.015	0.529 ± 0.013	0.371 ± 0.098	0.361 ± 0.020	0.594 ± 0.010	0.572 ± 0.017
DREAM1	*F*1	**0.556±0.008**	0.547 ± 0.008	0.544 ± 0.003	0.550 ± 0.007	0.327 ± 0.011	0.351 ± 0.018	0.469 ± 0.005	0.469 ± 0.002
DREAM2		0.497 ± 0.017	0.492 ± 0.014	0.474 ± 0.006	**0.532±0.006**	0.208 ± 0.010	0.264 ± 0.018	0.366 ± 0.008	0.334 ± 0.004
DREAM3		**0.544±0.010**	0.542 ± 0.007	0.471 ± 0.012	0.543 ± 0.002	0.334 ± 0.014	0.418 ± 0.021	0.436 ± 0.008	0.410 ± 0.016
DREAM4		0.450 ± 0.005	0.437 ± 0.007	**0.456±0.009**	0.445 ± 0.002	0.111 ± 0.005	0.310 ± 0.007	0.259 ± 0.009	0.234 ± 0.007
STRING-E		**0.579±0.007**	0.577 ± 0.005	0.546 ± 0.006	0.519 ± 0.007	0.341 ± 0.010	0.344 ± 0.007	0.485 ± 0.013	0.492 ± 0.003
STRING-ED		**0.588±0.006**	0.585 ± 0.005	0.559 ± 0.004	0.538 ± 0.005	0.350 ± 0.007	0.344 ± 0.010	0.499 ± 0.010	0.503 ± 0.010
STRING-EDC		**0.587±0.006**	0.584 ± 0.009	0.549 ± 0.003	0.499 ± 0.003	0.334 ± 0.013	0.359 ± 0.007	0.517 ± 0.013	0.509 ± 0.007
DREAM1	Resnik	**1.554±0.023**	1.483 ± 0.010	1.296 ± 0.009	1.422 ± 0.021	0.927 ± 0.009	0.708 ± 0.015	1.146 ± 0.022	1.239 ± 0.032
DREAM2		1.221 ± 0.023	1.232 ± 0.022	1.134 ± 0.018	**1.236±0.026**	0.822 ± 0.008	0.755 ± 0.015	0.976 ± 0.017	1.003 ± 0.016
DREAM3		1.089 ± 0.007	**1.108±0.014**	**1.011±0.021**	1.103 ± 0.031	0.884 ± 0.020	0.990 ± 0.058	1.042 ± 0.021	1.005 ± 0.023
DREAM4		1.032 ± 0.018	1.057 ± 0.016	1.020 ± 0.014	**1.083±0.015**	0.673 ± 0.016	0.761 ± 0.020	0.848 ± 0.026	0.848 ± 0.023
STRING-E		**1.592±0.021**	1.553 ± 0.015	1.222 ± 0.007	1.404 ± 0.018	0.931 ± 0.015	0.796 ± 0.027	1.226 ± 0.029	1.314 ± 0.023
STRING-ED		**1.642±0.034**	1.598 ± 0.038	1.298 ± 0.002	1.485 ± 0.045	0.987 ± 0.015	0.768 ± 0.022	1.260 ± 0.021	1.356 ± 0.023
STRING-EDC		**1.654±0.019**	1.596 ± 0.009	1.229 ± 0.015	1.357 ± 0.018	0.963 ± 0.022	0.927 ± 0.030	1.358 ± 0.032	1.381 ± 0.010

Note: Best performance bolded. All method parameters set as described in Section 2.

choice of the evaluation metric, almost always produces the best score for all three GO hierarchies. This pattern is more evident in dense, strongly connected PPI-adjacent networks like DREAM1 and the composite STRING networks, where GLIDER-*k*nn outpaces other methods by a significant margin. In DREAM2, even though the gap in performance is not as significant as that of DREAM1, GLIDER still outperforms the other methods in most of the evaluation metrics for MF and BP GO hierarchies (for CC, DSD-*k*nn slightly beats GLIDER). We see a similar pattern in DREAM3, where GLIDER is outperforming other methods in the MF hierarchy but for BP and CC, the results are very close between GLIDER and DSD-*k*nn. The exception is DREAM4, where the tables show GLIDER being overtaken by MV by a very small margin in MF and BP categories, and by DSD-*k*nn in the CC category. Note that absolute performance in DREAM4 is also much weaker especially for MF and BP: showing perhaps that co-expression networks contain less functionally relevant information than the actual PPI binding networks.

Similarly in the STRING composite networks, which are also dense networks, we find that the addition of ‘dataset’ and ‘coexpression’ edges decrease performance from the STRING-E network, which is not surprising, given the relative information content we saw for these different types of edges in the DREAM networks. Interestingly, MASHUP(S) performs best when all types of edges are included, leading us to postulate that either the MASHUP embedding or the more sophisticated SVM classifier can learn which edges are more reliable and incorporate that information.

Tables 2–4 also show that the GLIDER-*k*nn method is highly robust to the choice of *k* for the STRING and DREAM networks. In every experimental setting, we see GLIDER-25nn results being very close to, and in some cases slightly beating, the GLIDER-*k*nn scores obtained after training for *k* using the LOOCV method. This stability in the choice of *k* for the DREAM networks is important for our findings in Section 4.2 regarding the PD genes, where we fixed the size of *k* in our analysis.

In short, out of the 63 tests conducted to compare the performance of the different function prediction methods (characterized by rows in Tables 2–4), GLIDER produced the best results in 50 out of 63 experiments (≈80%), and in the remaining 13 experiments, GLIDER was almost always the strong second, with its score being very close to the top scoring method.

4.1.2 Comparing GLIDER-*k*nn and GLIDER-MASHUP

Because we were interested in whether our gain was coming from the GLIDER graph or the *k*nn classifier, we also chose to measure the performance of GLIDER-MASHUP as well (see Section 2.3.6), The results in Tables 2–4 show GLIDER-*k*nn significantly out-performing GLIDER-MASHUP in all the networks and GO categories (with the sole exception of BP for DREAM3, where neither performs as well as DSD-*k*nn). Evidently, the GLIDER local neighborhood is so strong in recovering function that a simple *k*nn classifier tends to do the job.

Furthermore, it is interesting to note that the addition of co-expression edges in the STRING-ED network resulted in a noticeable performance decline in GLIDER-MASHUP, exactly the opposite of what we observed for MASHUP(S). This decline can be attributed to GLIDER’s weakness in producing strong functional associations in heterogeneous networks, where edges can signify different meanings. So, the low scores of GLIDER-MASHUP on STRING-EDC (compared to STRING-E) is probably due to the negative returns from GLIDER counteracting the effectiveness of MASHUP while dealing with heterogeneous networks like STRING-EDC.

4.2 Biological case study: Parkinson’s Disease genes

4.2.1 A collection of disease–gene neighborhood subgraphs

We consider the GLIDER-15 neighborhood subgraphs for a set of genes known, based on GWAS studies, to be implicated in Parkinson’s Disease (PD). More specifically, a set of GWAS genes associated with PD was collected from the previously published literature (Blauwendraat *et al.*, 2020; Nalls *et al.*, 2014, 2019), where we considered the set of 40 GWAS genes from these papers that appear in all four of DREAM1–4 (see Table 5 for the gene names). For each of these PD GWAS genes, we looked at its 15 GLIDER neighborhood subgraph in each of DREAM1–4. For example, Figure 2 gives the subgraph of genes for DREAM1 and DREAM2. We further explore the characterization of this collection of 40 × 4 GLIDER neighborhood subgraphs, each anchored by a GWAS gene (cytoscape plots of all the 15 GLIDER neighborhood subgraphs of all 40 GWAS genes in Table 5 are available as Supplementary Material from the GLIDE github repository).

We first wished to compare the similarity and differences among the 40 subgraphs when switching between different DREAM networks. Supplementary Figure S3 shows the histogram of the average clustering coefficients for all the GLIDER subgraphs of GWAS genes in DREAM1–4. A cursory inspection of the histogram shows that the subgraphs on DREAM1 and DREAM4 were often significantly different from that of DREAM2 and DREAM3 in terms of graph connectivity. In fact, DREAM1 and DREAM4 subgraphs

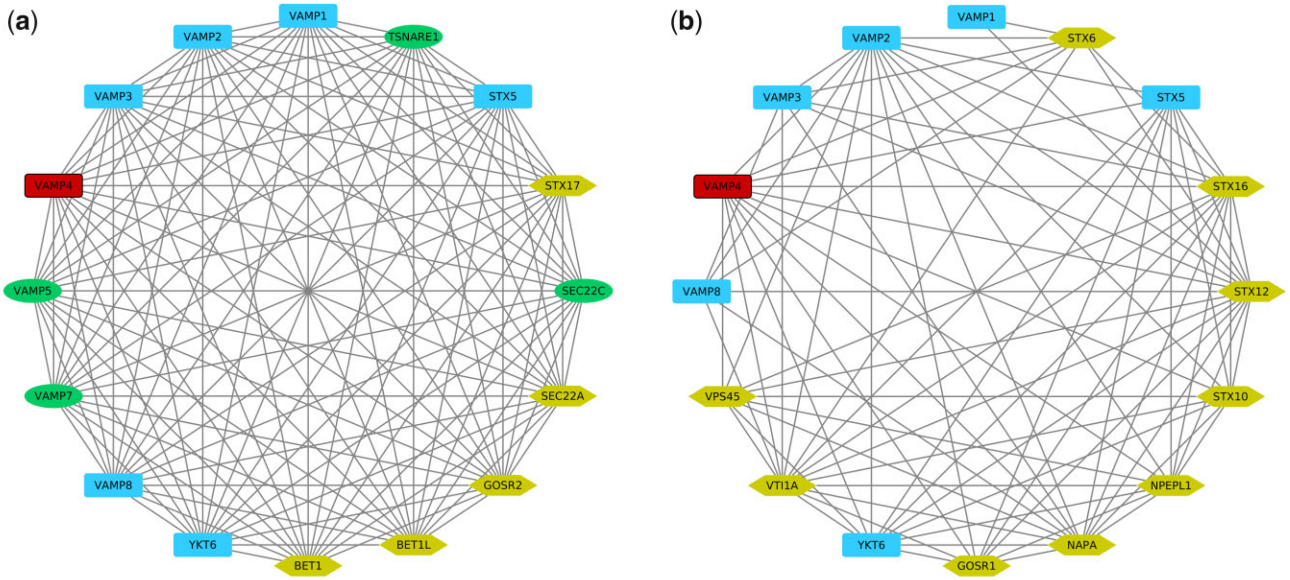

Fig. 2. GLIDER-neighbors and their induced subgraph for the protein VAMP4 in (**a**) DREAM1, and (**b**) DREAM2 networks. The number of top VAMP4 (bolded node in the figure) GLIDER neighbors *k* is set to 15. Note: rectangular nodes are present in both DREAM1 and DREAM2. The oval nodes in the DREAM1 subgraph (a) are absent in the whole of DREAM2. The hexagonal nodes are only present in one of the subgraphs in (a) and (b), even though these nodes are present in both DREAM1 and DREAM2

Table 5. List of 40 GWAS genes implicated for PD that are present in all the DREAM1–4 networks

BAG3	CTSB	HTRA2	PARK7	SREBF1
BCKDK	DLG2	KPNA1	PINK1	STK39
BRIP1	DYRK1A	MAP4K4	RIMS1	SYNJ1
CD19	EIF4G1	MAPT	RIT2	SYT11
CHRNB1	FBXO7	NOD2	SATB1	UBTF
CLCN3	FCGR2A	NSF	SETD1A	USB25
CNTN1	FYN	NUCKS1	SHEGL2	VAMP4
CRHR1	GBF1	PAM	SHEGL2	WNT3

were more likely to be highly connected compared to the rest. This can be largely explained by the fact that the number of edges in DREAM1 and DREAM4 were significantly greater than that of DREAM3 and DREAM4. The lack of good connectivity in DREAM3 subgraphs, which can be seen by comparing the histogram in Supplementary Figure S3c to the rest, was expected as DREAM3 is a fairly sparse network.

4.2.2 Edge density of the gene neighborhood subgraphs does not correlate with functional enrichment

The average edge density of the GWAS subgraphs in a particular DREAM network did not correlate consistently with functional enrichment. We used the FuncAssociate 3.0 API (Berriz *et al.*, 2009) to calculate the functional enrichment of the collection of 40 GLIDER neighborhood GWAS subgraphs, calling a subgraph enriched if it returned at least one GO Functional label with an adjusted *P*-value of $P < 0.05$. Table 6 plots the percentage of the 40 GWAS genes whose GLIDER neighborhood subgraph of closest k genes was found by FuncAssociate to be enriched for at least one GO label, for each of DREAM1–4 and $k = 5$, 10, 15 and 50. The percent functionally enriched ranges from a high of 98% for DREAM1 to a low of 48% for DREAM4. When we extend out to the neighborhood of 50 closest GLIDE genes, over 95% of the 40 gene neighborhoods show functional enrichment in DREAM1–3, and 87.5% do so for DREAM4.

4.2.3 Case analysis of two PD GWAS genes: VAMP4 and PINK1

We first look in more depth at the neighborhood subgraph of VAMP4. We find that the neighborhood subgraphs of VAMP4 in DREAM1 and DREAM2 have > 1/3 of their genes in common (rectangular (blue) nodes in Fig. 2). Figures 2 and 3 were created with Cyctoscape (Shannon *et al.*, 2003). The oval (green) genes are missing in the DREAM2 network all together; the hexagonal (gold) genes in Figure 2 are present in both DREAM1 and DREAM2, but either present in the GLIDER-15 subgraph for DREAM1 and not DREAM2, or in DREAM2 but not DREAM1.

We went to the literature to see what was known about the genes in these subgraphs and their disease and pathway involvement. For VAMP4, there is substantial overlap between the neighborhood subgraphs for DREAM1–2, and in particular, many genes involved in both subgraphs are implicated in SNARE complexes. In particular, STX5, GOSR1, YKT6 and BET1L make up the Cis-Golgi SNARE complex, and VAMP4 itself along with STX16, STX6 and VT1A make up the Trans-Golgi SNARE complex (Climer *et al.*, 2015). There is increasing evidence that these SNARE complexes, that regulate ER-Golgi transport, become disregulated in PD (Ahmadpour *et al.*, 2020; Martínez-Menárguez *et al.*, 2019; Rendón *et al.*, 2013).

Table 6. Table of the fraction of the GWAS genes whose GLIDE neighbors enriched at least one GO label, using FuncAssociate (version 3.0), when the number of GLIDE neighbors is *k*

Network	$k = 5$	$k = 10$	$k = 15$	$k = 50$
DREAM1	0.90	0.98	0.95	0.98
DREAM2	0.63	0.80	0.73	0.95
DREAM3	0.68	0.70	0.75	0.98
DREAM4	0.50	0.48	0.65	0.875

We note that the DREAM3 subgraph for VAMP4 shows no functional enrichment for *any* GO-term with FuncAssociate, involves no DREAM3 edges, and probably just indicates that many of the relevant VAMP4 connections are missing from the very sparse DREAM3.

PINK1 is chosen as an example that is very different than VAMP4 in that the DREAM1 and DREAM2 GLIDER-15 neighborhoods identify completely different sets of genes; when we extend to the GLIDER-20 neighborhood (see Fig. 3), the important PARK2 (also called Parkin) known to interact with PINK1, shows up in both the DREAM1 and DREAM2 neighborhoods of PINK1. Mutations in PARK2 associate with inherited early-onset recessive PD (Djarmati *et al.*, 2004; Huttenlocher *et al.*, 2015). The GLIDER-15 neighborhood of PINK1 in DREAM1 also contains TOMM70A, NIPSNAP1 and MARCH5, all of which have been linked to PD, and are involved in autophagy and clearance of damaged mitochondria, a process with increasing evidence of a centralized role in PD. More specifically, Bertolin *et al.* (2013) showed PD-causing PARK2 mutations weakened or disrupted the molecular interaction between PARK2 and TOMM70A; Abudu *et al.* (2019) showed that NIPSNAP1 has a role in recognition of damaged mitochondria, as well as demonstrated that zebrafish lacking a functional Nipsnap1 display Parkinsonism.

Koyano *et al.* (2019) showed that the initial step in PARK2 recruitment is delayed following depletion of the mitochondrial E3, MARCH5. They propose a model in which the initial step in PARK2 recruitment and activation requires protein ubiquitylation by MARCH5 with subsequent PINK1-mediated phosphorylation.

The GLIDE neighborhood of PINK1-PARK2 in DREAM3 consists of entirely different genes from DREAM1, but a large subset also appear to have strong known associations with PD. Intermediate-length polyQ expansions (>24 Qs) of ATXN2 were found in seven ADPD patients and no controls (Yamashita *et al.*, 2014). Jo *et al.* (2020) suggest that AIMP2 contributes to PD pathogenesis. The orphan G-protein-coupled receptor 37 is a substrate of Parkin, and its insoluble aggregates accumulate in brain tissue samples of PD patients, including Lewy bodies and neurites (Marazziti *et al.*, 2009). Abnormal accumulation or turnover of RanBP2 and its substrates, may contribute to neuronal cell death in PD (Um *et al.*, 2006). VDAC1 is necessary for PINK1/Parkin-directed autophagy of damaged mitochondria (Geisler *et al.*, 2010). Grossmann *et al.* (2020) show the functional interaction of RHOT1 with other PD gene products linked to mitochondrial quality control.

5 Discussion

We introduced GLIDER, a simple function prediction method based on the GLIDE quasi-kernel, and showed its utility for function prediction in a heterogeneous collection of human PPA networks. A case study of GLIDER neighborhoods of known PD disease genes was presented, supporting involvement of SNARE complexes and mitochondrial autophagy in PD disease processes.

The DREAM networks were deliberately kept universal so they could be applied to a wide range of different human traits and conditions (Choobdar *et al.*, 2019); thus in DREAM1, e.g. genes in the GLIDER-15 neighborhood of PD GWAS gene BCKDK include PDK1–4, four isoforms of PDK, that have very different tissue expression profiles (Shi and McQuibban, 2017). PDK2 is ubiquitously expressed and has been shown to be a key regulator of PINK1/PARKIN-mediated mitophagy, a key pathway disregulated in PD (Shi and McQuibban, 2017). PDK4 is also highly expressed in brain tissue, but PDK1 is expressed almost exclusively in heart tissue (Di *et al.*, 2010) whereas PDK3 has only been found expressed in kidney and testes (Bowker-Kinley *et al.*, 1998). Thus PDK1 and PDK3 are unlikely to be associated with PD, a class of false positives that would be eliminated by running GLIDER instead on tissue-specific networks (Magger *et al.*, 2012) customized for brain.

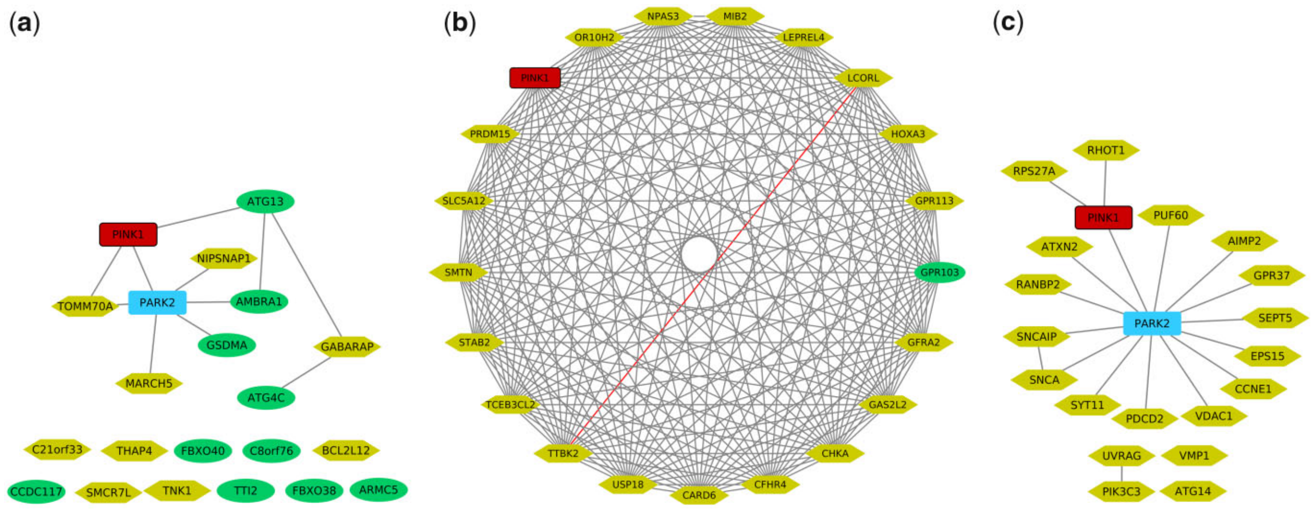

Fig. 3. GLIDER-neighbors and their induced subgraph for the protein PINK1 in (**a**) DREAM1, (**b**) DREAM2 and (**c**) DREAM3 networks. The number of PINK1 (bolded node in the figure) neighbors was chosen to be 20. Note: the hexagonal nodes [in (a–c)] also appear as nodes in DREAM1. The oval nodes [in (a)] are absent in DREAM2 while the oval nodes in (b) (GPR103) is absent in DREAM1

Acknowledgements

We thank Rohit Singh, Sam Sledzieski, Donna Slonim, the Tufts BCB group and the anonymous referees for helpful suggestions that greatly improved the quality of the article.

Data Availability

The data underlying this article are available in Supplementary material at Bioinformatics online, and in Github at https://github.com/kap-devkota/GLIDER. The networks were derived from sources in the public domain: STRING (https://string-db.org/), and Disease Module Identification Dream Challenge (https://www.synapse.org/#!Synapse:syn7543745)

Funding

This work was supported in part by the National Science Foundation [DMS 1812503, CCF 1934553 to L.J.C.].

Conflict of Interest: none declared.

References

Abudu,Y.P. *et al.* (2019) NIPSNAP1 and NIPSNAP2 act as "eat me" signals to allow sustained recruitment of autophagy receptors during mitophagy. *Autophagy*, **15**, 1845–1847.

Ahmadpour,D. *et al.* (2020) Hitchhiking on vesicles: a way to harness age-related proteopathies? *FEBS J.*, **287**, 5068–5079.

Berriz,G.F. *et al.* (2009) Next generation software for functional trend analysis. *Bioinformatics*, **25**, 3043–3044.

Bertolin,G. *et al.* (2013) The TOMM machinery is a molecular switch in PINK1 and PARK2/PARKIN-dependent mitochondrial clearance. *Autophagy*, **9**, 1801–1817.

Blauwendraat,C. *et al.* (2020) The genetic architecture of Parkinson's disease. *Lancet Neurol.*, **19**, 170–178.

Bowker-Kinley,M.M. *et al.* (1998) Evidence for existence of tissue-specific regulation of the mammalian pyruvate dehydrogenase complex. *Biochem. J.*, **329**, 191–196.

Cao,M. *et al.* (2013) Going the distance for protein function prediction. *PLoS One*, **8**, e76339.

Cao,M. *et al.* (2014) New directions for diffusion-based prediction of protein function: incorporating pathways with confidence. *Bioinformatics*, **30**, i219–i227.

Cho,H. *et al.* (2016) Compact integration of multi-network topology for functional analysis of genes. *Cell Syst.*, **3**, 540–548.

Choobdar,S. *et al.*; DREAM Module Identification Challenge Consortium. (2019) Assessment of network module identification across complex diseases. *Nat. Methods*, **16**, 843–852.

Climer,L.K. *et al.* (2015) Defects in the COG complex and COG-related trafficking regulators affect neuronal Golgi function. *Front. Neurosci.*, **9**, 405.

Cowen,L. *et al.* (2017) Network propagation: a universal amplifier of genetic associations. *Nat. Rev. Genet.*, **18**, 551–562.

Cowen,L. *et al.* (2021) Diffusion state distances: multitemporal analysis, fast algorithms, and applications to biological networks. *SIAM J. Math. Data Sci.*, **3**, 142–170.

Devkota,K. *et al.* (2020) GLIDE: combining local methods and diffusion state embeddings to predict missing interactions in biological networks. *Bioinformatics*, **36**, i464–i473.

Di,R.M. *et al.* (2010) PDK1 plays a critical role in regulating cardiac function in mice and human. *Chin. Med. J.*, **123**, 2358–2363.

Djarmati,A. *et al.* (2004) Detection of Parkin (PARK2) and DJ1 (PARK7) mutations in early-onset Parkinson disease: Parkin mutation frequency depends on ethnic origin of patients. *Hum. Mutat.*, **23**, 525.

Geisler,S. *et al.* (2010) PINK1/Parkin-mediated mitophagy is dependent on VDAC1 and p62/SQSTM1. *Nat. Cell Biol.*, **12**, 119–131.

Gligorijević,V. *et al.* (2018) deepNF: deep network fusion for protein function prediction. *Bioinformatics*, **34**, 3873–3881.

Grossmann,D. *et al.* (2020) The emerging role of RHOT1/Miro1 in the pathogenesis of Parkinson's disease. *Front. Neurol.*, **11**, 587.

Grover,A. and Leskovec,J. (2016) node2vec: scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD, San Francisco, California USA*. pp. 855–864. ACM.

Huttenlocher,J. *et al.* (2015) Heterozygote carriers for CNVs in PARK2 are at increased risk of Parkinson's disease. *Hum. Mol. Genet.*, **24**, 5637–5643.

Jiang,Y. *et al.* (2016) An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol.*, **17**, 184.

Jo,A. *et al.* (2020) Deubiquitinase USP29 governs MYBBP1a in the brains of Parkinson's disease patients. *J. Clin. Med.*, **9**, 52.

Kovács,I.A. *et al.* (2019) Network-based prediction of protein interactions. *Nat. Commun.*, **10**, 1240.

Koyano,F. *et al.* (2019) Parkin recruitment to impaired mitochondria for nonselective ubiquitylation is facilitated by MITOL. *J. Biol. Chem.*, **294**, 10300–10314.

Lazarsfeld,J. *et al.* (2021) Majority vote cascading: a semi-supervised framework for improving protein function prediction. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 1.

Li,T. *et al.* (2017) A scored human protein–protein interaction network to catalyze genomic interpretation. *Nat. Methods*, **14**, 61–64.

Magger,O. *et al.* (2012) Enhancing the prioritization of disease-causing genes through tissue specific protein interaction networks. *PLoS Comput. Biol.*, **8**, e1002690.

Marazziti,D. *et al.* (2009) Induction of macroautophagy by overexpression of the Parkinson's disease-associated GPR37 receptor. *FASEB J.*, **23**, 1978–1987.

Martínez-Menárguez,J.Á. *et al.* (2019) Golgi fragmentation in neurodegenerative diseases: is there a common cause? *Cells*, **8**, 748.

Nalls,M.A. *et al.*; Alzheimer Genetic Analysis Group. (2014) Large-scale meta-analysis of genome-wide association data identifies six new risk loci for Parkinson's disease. *Nat. Genetics*, **46**, 989–993.

Nalls,M.A. *et al.*; International Parkinson's Disease Genomics Consortium. (2019) Identification of novel risk loci, causal insights, and heritable risk for Parkinson's disease: a meta-analysis of genome-wide association studies. *Lancet Neurol.*, **18**, 1091–1102.

Nelson,W. *et al.* (2019) To embed or not: network embedding as a paradigm in computational biology. *Front. Genet.*, **10, 381**.

Pandey,J. *et al.* (2008) Functional coherence in domain interaction networks. *Bioinformatics*, **24**, i28–i34.

Radivojac,P. *et al.* (2013) A large-scale evaluation of computational protein function prediction. *Nat. Methods*, **10**, 221–227.

Rendón,W.O. *et al.* (2013) Golgi fragmentation is Rab and SNARE dependent in cellular models of Parkinson's disease. *Histochem. Cell Biol.*, **139**, 671–684.

Schwikowski,B. *et al.* (2000) A network of protein-protein interactions in yeast. *Nat. Biotechnol.*, **18**, 1257–1261.

Shannon,P. *et al.* (2003) Cytoscape: A software environment for integrated models of biomolecular interaction networks. Genome Res., **13**, 2498–2504. https://doi.org/10.1101/gr.1239303.

Shi,G. and McQuibban,G.A. (2017) The mitochondrial rhomboid protease PARL is regulated by PDK2 to integrate mitochondrial quality control and metabolism. *Cell Rep.*, **18**, 1458–1472.

Szklarczyk,D. *et al.* (2015) STRINGv10: protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.*, **43**, D447–D452.

Szklarczyk,D. *et al.* (2021) The string database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Res.*, **49**, D605–D612.

Türei,D. *et al.* (2016) OmniPath: guidelines and gateway for literature-curated signaling pathway resources. *Nat. Methods*, **13**, 966–967.

Um,J.W. *et al.* (2006) Parkin ubiquitinates and promotes the degradation of RanBP2. *J. Biol. Chem.*, **281**, 3595–3603.

Yamashita,C. *et al.* (2014) The evaluation of polyglutamine repeats in autosomal dominant Parkinson's disease. *Neurobiol. Aging*, **35**, 1779.e17–1779.e21.

Zhao,C. and Wang,Z. (2018) GOGO: an improved algorithm to measure the semantic similarity between gene ontology terms. *Sci. Rep.*, **8**, 15107.

Zhou,N. *et al.* (2019) The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol.*, **20**, 244.