SOFTWARE TOOL ARTICLE

PPInfer: a Bioconductor package for inferring functionally related proteins using protein interaction networks [version 1; referees: 1 approved with reservations]

Dongmin Jung¹, Xijin Ge²

¹Avison Biomedical Research Center, Yonsei University, Seoul, South Korea
²Department of Mathematics and Statistics, South Dakota State University, Brookings, SD, USA

Abstract

Interactions between proteins occur in many, if not most, biological processes. This fact has motivated the development of a variety of experimental methods for the identification of protein-protein interaction (PPI) networks. Leveraging PPI data available STRING database, we use network-based statistical learning methods to infer the putative functions of proteins from the known functions of neighboring proteins on a PPI network. This package identifies such proteins often involved in the same or similar biological functions. The package is freely available at the Bioconductor web site (http://bioconductor.org/packages/PPInfer/).

Keywords

graph mining, protein-protein interaction, pseudo-absence, semi-supervised learning, support vector machine

This article is included in the RPackage gateway.

This article is included in the Bioconductor gateway.
This article is included in the Machine learning: life sciences collection.

This article is included in the Interactive Figures collection.

**Corresponding author:** Xijin Ge (gexjin@gmail.com)

**Author roles:** Jung D: Conceptualization, Investigation, Methodology, Software, Visualization, Writing – Original Draft Preparation; Ge X: Conceptualization, Funding Acquisition, Investigation, Methodology, Validation, Visualization, Writing – Review & Editing

**Competing interests:** No competing interests were disclosed.

**Grant information:** This material is based upon work supported by the National Science Foundation/EPSCoR Grant Number IIA-1355423 and by the State of South Dakota. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Copyright:** © 2017 Jung D and Ge X. This is an open access article distributed under the terms of the Creative Commons Attribution Licence, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**How to cite this article:** Jung D and Ge X. PPInfer: a Bioconductor package for inferring functionally related proteins using protein interaction networks [version 1; referees: 1 approved with reservations] F1000Research 2017, 6:1969 (https://doi.org/10.12688/f1000research.12947.1)

**First published:** 07 Nov 2017, 6:1969 (https://doi.org/10.12688/f1000research.12947.1)
Introduction
The functions of many proteins remain unknown. This is a big challenge in functional genomics. As proteins that consist of the same protein complex are often involved in the same cellular process, the pattern of protein-protein interactions (PPIs) can give information regarding protein function. Thus PPI databases can be useful to predict protein function, complementing conventional approaches based on protein sequence analyses.

Many databases store information on protein-protein interactions as well as protein complexes. For example, the Biological General Repository for Interaction Datasets (BioGRID) includes over 500,000 manually annotated interactions. The STRING database aims to provide a critical assessment and integration of protein-protein interactions, including direct (physical) as well as indirect (functional) associations. The basic interaction unit in STRING is the functional association, i.e. a specific and productive functional relationship between two proteins.

Once the PPI has been obtained experimentally, there are numerous methods to analyze the network. A neighbor counting method for protein function prediction was developed. The theory of Markov random fields was used to infer functions using protein-protein interaction data and the functional annotations of its interaction protein partners. There are many algorithms that try to integrate multiple sources of data to infer functions. We propose a method that can infer the putative functions of proteins by solving classification problems and thus identifying closely connected proteins known to be involved in a certain process.

We use the kernel method for the graph as a similarity measure between proteins, instead of using the first or second level neighbors, and thus our proposed method provides scores or distances derived from a graph kernel. The main advantage of the proposed method is that the neighbors of a set of proteins can be ranked in terms of scores. Generally, we need two or more classes for classification problem. Although we only know the target, we want to apply semi-supervised learning techniques to the PPI through this package. Thus, the main idea of this method is how to find another class so that two classes can be used for classification problem. Eventually, we can classify proteins to identify such closely related proteins by using this package. Finally, functional enrichment analyses such as ORA and GSEA are incorporated to predict the protein function from the closely related proteins.

Methods
The support vector machine (SVM) is one of most widely used methods for classification. Suppose we have a dataset in the real space and that each point in our dataset has a corresponding class label. A SVM is involved in a convex optimization problem to separate data points in the dataset according to their class, by maximizing distance between class and minimizing a penalty for misclassification for each class, at the same time. Unfortunately, the graph data is not in the real space. Cover’s theorem provides the useful idea behind a nonlinear SVM, which is to find an optimal separating hyperplane in high-dimensional feature space mapped by using a suitable kernel function, just as we did for the linear SVM in original space.

Graph (network) data is ubiquitous and graph mining tries to extract novel and insightful information from data. Graph kernels are defined in the form of kernel matrices, based on the normalized Laplacian matrix for a graph. The best-known kernel in a graph is the diffusion kernel. The motivation is that it is often easier to describe the local neighborhood than to describe the structure of the whole space. Another method is called a regularized Laplacian matrix and is widely used in areas such as spectral graph theory, where properties of graphs are studied in terms of their eigenvalues and vectors of adjacency matrices. Broadly speaking, kernels can be thought of as functions that produce similarity matrices. In the package, we choose only the regularized Laplacian matrix as a graph kernel for the PPI.

In many biological problems, datasets are often compounded by imbalanced class distribution, known as the imbalanced data problem, in which the size of one class is significantly larger than that of the other class. Many classification algorithms such as a SVM are sensitive to data with an imbalanced class problem, leading to a suboptimal classification. It is desirable to compensate the imbalance effect in model training for more accurate classification. A possible solution to this problem is to use the one-class SVM (OCSVM) by learning from the target class only. In one-class classification, it is assumed that only information of one of the classes, the target class, is available, and no information is available from the other class, known as the background. The OCSVM can be solely applied because we have only one class, the target. However, it is known that one-class classifiers seldom outperform two-class classifiers when the data from two class are available.

The strategy of this package is to make use of the OCSVM and classical SVM, sequentially. First, we apply the OCSVM by training a one-class classifier using the data from the known class only. Let n be the number of proteins in the
target class. This model is used to identify distantly related proteins among remaining \( N - n \) proteins in the background. Indeed, we do not know whether or not each of proteins in the background interacts with the proteins of the target. Thus, it does not always imply that proteins in the background do not interact with the target. In fact, their associations with the target are not observed to date yet. Proteins with zero similarity with the target class are extracted. Then they are potentially defined as the other class by pseudo-absence selection methods from spatial statistics. The target class can be seen as real presence data. The main idea of the proposed method is to adopt the pseudo-absence class. For the data to be balanced, assume that two classes contain the same number of proteins. Next, by the classical SVM, these two classes are used to identify closely related proteins whose scores ranging from -1 to 1 are close to 1 among remaining \( N - 2n \) proteins. Semi-supervised learning can be applied to make use of large unlabeled data and small labeled data. Some of these methods directly try to label the unlabeled data. Eventually, those found by this procedure can be functionally linked to the target proteins. This is usually based on the assumption that unannotated proteins have similar functions as their interacting proteins.

**Use cases**

We need a list of proteins and the kernel matrix to infer functionally related proteins. With the STRING database for mouse, it is supposed that the target is the set of proteins in the Ras signaling pathway from the KEGG pathway (http://www.genome.jp/kegg-bin/show_pathway?mmu04014).

```r
library(PPInfer)
# download the kernel matrix from http://ge-lab.org/dm/K10090.rds
K.10090 <- readRDS("K10090.rds")

# remove prefix
rownames(K.10090) <- sub(".*\\.", ", rownames(K.10090))

# load target
library(limma)
kegg.mmu <- getGeneKEGGLinks(species.KEGG = "mmu")
index <- which(kegg.mmu[,2] == "path:mmu04014")
path.04014 <- kegg.mmu[index,1]

# infer functionally related proteins
path.04014.infer <- ppi.infer.mouse(path.04014, K.10090, input = "entrezgene",
       output = "entrezgene", nrow(K.10090))
genes <- path.04014.infer$top

# load gene sets for gene ontology
library(org.Mm.eg.db)
library(GO.db)
xx <- sapply(as.list(org.Mm.egGO2EG), unique)

# ORA with top 100 proteins
resORA <- ORA(xx, genes[1:100])
ORAsmmary <- na.omit(data.frame(resORA, AnnotationDbi::select(GO.db,
       rownames(resORA), "TERM")))
head(ORAsmmary)

p <- ORA.dotplot(ORAsmmary, category = "TERM", count = "Count", size = "Size",
       pvalue = "pvalue", sort = "pvalue", p.adjust.methods = "fdr", numChar = 60,
       top = 50) + scale_colour_gradient(low = "red", high = "yellow")

# interactive figure
library(plotly)
config(ggplotly(p), showLink = TRUE)

# GSEA
index <- !is.na(path.04014.infer$score)
genes <- path.04014.infer$top[index]
scores <- path.04014.infer$score[index]
```
scaled.scores <- as.numeric(scale(scores))
names(scaled.scores) <- genes
set.seed(1)
resGSEA <- fgsea(xx, scaled.scores, nperm = 1000)
GSEAsummary <- na.omit(data.frame(resGSEA, AnnotationDbi::select(GO.db,
resGSEA$pathway, "TERM")))
head(GSEAsummary)

# interactive figure
p <- GSEA.barplot(GSEAsummary, category = "TERM", score = "NES", pvalue = "padj",
top = 50, sort = "NES", decreasing = TRUE, numChar = 50) +
scale_fill_continuous(low = "red", high = "blue")

The tyrosine kinase receptor binds to a ligand in the extracellular space. Then its cytoplasmic domain undergoes
conformational change that forms dimerization, resulting in transphosphotyrosine. The SH2 domain recognizes this
phosphotyrosine. The Grb2 adaptor protein containing SH2 binds these receptors and recruits Sos, which is the
Ras-GEF inducing Ras, to release its GDP and bind a GTP instead. The Ras protein is activated by this guanine
nucleotide exchange factor. GTP-activated Ras can activate PI3K. Once PI3P is formed by PI3K, an Akt/PKB kinase
can become tethered via its PH domain. Once activated, Akt/PKB proceeds to phosphorylate a series of protein
substrates, leading to aiding survival by reducing the possibility of apoptotic suicide program. For this reason, several
GO terms about the RTK and PI3K pathways are shown in Figure 1. We can find cell migration due to the integrin
and FAK. MAP kinase activity, which is a downstream of Ras signaling and causes cell proliferation, is statistically
significant.

Figure 1. Top 50 categories and their adjusted p-values from the ORA with gene ontology, based on 100 proteins
most closely related to the target, Ras signaling pathway. This figure was generated with Plotly (https://plot.ly/).
In the GSEA, the gene list is sorted by the standardized score showing how much they are functionally close to the target. Genes with higher scores are functionally more related to the target. The positive values of the enrichment score indicate enrichment at the top of the ranked gene list. Proteins that are closely related to the target are enriched in categories with high enrichment scores. In other words, these proteins are depleted in categories with low enrichment scores. We can see the RTK, MAPK and Wnt signaling pathways in Figure 2. One possible reason is that Akt can activate β-catenin both indirectly and indirectly.

# gene sets for GO and KEGG pathway
GO.TERM <- AnnotationDbi::select(GO.db, names(xx), "TERM")[,2]
names(xxx) <- GO.TERM
library(KEGG.db)
pathway.id <- unique(kegg.mmu[,2])
yy <- list()
for(i in 1:length(pathway.id))
{
  index <- which(kegg.mmu[,2] == pathway.id[i])
  yy[[i]] <- kegg.mmu[index,1]
}
library(Category)
names(yy) <- getPathNames(sub("[:alpha:]+.", "", pathway.id))
yy[which(names(yy) == "NA")]<- NULL

# remove duplicate categories
yy[intersect(names(xx), names(yy))]<- NULL

# GSEA
set.seed(1)
GSEAsummary <- fgsea(c(xx, yy), scaled.scores, nperm = 1000)
index <- which(GSEAsummary[,1] == names(yy[1]))
groups <- 0
for(index:(index-1))<- "GO"
groups[index:nrow(GSEAsummary)] <- "KEGG"
index <- match(data.frame(GSEAsummary[,1])[,1], names(c(xx, yy)))

# network visualization
g <- enrich.net(GSEAsummary, c(xx, yy)[index], node.id = "pathway", numChar = 60,
pvalue = "pval", edge.cutoff = 0.2, pvalue.cutoff = 0.05, degree.cutoff = 1,
n = 200, group = groups, vertex.label.cex = 0.6, vertex.label.color = "black")

# interactive figure
vs <- V(g)
es <- as.data.frame(get.edgelist(g))
Nv <- length(vs)
Ne <- length(es[1]$V1)

# create nodes
L <- layout.kamada.kawai(g)
Xn <- L[,1]
Yn <- L[,2]
group <- ifelse(V(g)$shape == "circle", "GO", "KEGG")
network <- plot_ly(x = ~Xn, y = ~Yn, type = "scatter", mode = "markers",
  marker = list(color = V(g)$color, size = V(g)$size*2,
    symbol= ~V(g)$shape, line = list(color = "gray", width = 2)),
  hoverinfo = "text", text = paste("<br">", group, "</br>", names(vs))) %>%
  add_annotations( x = ~Xn, y = ~Yn, text = names(vs), showarrow = FALSE,
    font = list(color = "gray", size = 10))
# create edges
edge_shapes <- list()
for(i in 1:Ne)
{
  v0 <- es[i,]$V1
  v1 <- es[i,]$V2
  index0 <- match(v0, names(V(g)))
  index1 <- match(v1, names(V(g)))
  edge_shape <- list(type = "line", line = list(color = "#030303",
                        width = E(g)$width[i]), x0 = Xn[index0], y0 = Yn[index0],
                       x1 = Xn[index1], y1 = Yn[index1])
  edge_shapes[[i]] <- edge_shape
}

# create network
axis <- list(title = "", showgrid = FALSE, showticklabels = FALSE, zeroline = FALSE)
h <- layout(network, title = "Enrichment Network", shapes = edge_shapes,
            xaxis = axis, yaxis = axis)
config(h, showLink = TRUE)

In Figure 3, the network visualization of the functional enrichment analysis is displayed. Nodes indicate GO terms and KEGG pathways. The connection between two nodes depends on the proportion of overlapping genes of corresponding two categories. The size of nodes is proportional to the number of genes in their categories. The more significant categories are, the less transparent their nodes are. This network may be useful to see the overview of the functional enrichment.

![Network visualization](image)

**Figure 2.** Top 50 categories from the GSEA with scaled scores of genes from the target, Ras signaling pathway. This figure was generated with Plotly ([https://plot.ly](https://plot.ly)).
Discussion
The proposed method is highly dependent on protein interaction networks. There are many prominent databases that include extensive information on protein interactions, and interactions are also reported in thousands of literature references. Therefore, the choice of data can be critical. Another issue is a technical problem. The one-class support vector machine can be sensitive to the choice of kernels and parameters. However, there is potential in inferring putative functions of proteins from PPIs, complementing conventional methods based on protein sequence analyses.

Software availability
The PPInfer package is available at: http://bioconductor.org/packages/PPInfer/
Source code is available at: https://github.com/Bioconductor-mirror/PPInfer
Archived source code as at the time of publication: https://doi.org/doi:10.5281/zenodo.1035128
License: Artistic-2.0

Competing interests
No competing interests were disclosed.

Grant information
This material is based upon work supported by the National Science Foundation/EPSCoR Grant Number IIA-1355423 and by the State of South Dakota.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.
References


Open Peer Review

Current Referee Status: ❓

Referee Report 22 November 2017

https://doi.org/10.5256/f1000research.14038.r27695

Cathy H. Wu 1, Karen E. Ross 2
1 Center for Bioinformatics & Computational Biology (CBCB), University of Delaware, Newark, DE, USA
2 Department of Biochemistry and Molecular & Cellular Biology, Georgetown University Medical Center, Washington, DC, USA

The article describes a new R package, PPInfer, for inferring functions of proteins based on their connectivity in a PPI network. Discovery of new functions for proteins is an important problem and the method as described is an interesting and theoretically sound approach.

The authors provide detailed example code for using PPInfer. Unfortunately, when trying to reproduce the use case, there was an error in the first step:

> K.10090 <- readRDS("K10090.rds")
Error in gzfile(file, "rb") : cannot open the connection
In addition: Warning message:
In gzfile(file, "rb") :
cannot open compressed file 'K10090.rds', probable reason 'No such file or directory'

Therefore, we couldn't test the rest of the use case.

Also, in general, the use case would benefit from more text explanation so that readers understand the purpose of the tool and how to interpret the results. Specifically:

1. What is the meaning of the gene ratio on the x-axis in Figure 1?

2. In Figure 3, are the nodes enriched KEGG pathways and GO terms as determined by GSEA? What was the threshold for deciding which nodes should appear in the graph? What is the biological message of this graph?

3. A discussion of the functionally related proteins themselves (not just the enriched terms/pathways) would be helpful. How do these proteins functionally interact with the ras pathway proteins in the STRING network? Perhaps you could discuss the STRING confidence scores or types of evidence for these interactions.

4. It would be helpful to have more discussion of exactly what we learn from this approach. Can we infer that the proteins identified in the use case might be involved in the Ras pathway? It seems like based on the GO/KEGG annotation for the proteins, we could already have assumed they had
something to do with signal transduction even without knowing that they are closely associated with Ras pathway proteins. Are there any examples of proteins whose association with the Ras pathway was surprising? Are there any proteins about which little is known or whose GO/KEGG annotation does not indicate that they would be associated with the Ras pathway?

Please note that for questions 3, 4, and 5 on the referee report form: we were unable to fully evaluate this as we were unable to run the software. We have therefore assigned the answer "no" to reflect this.

Is the rationale for developing the new software tool clearly explained?
Partly

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
No

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
No

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
No

Competing Interests: No competing interests were disclosed.

We have read this submission. We believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however we have significant reservations, as outlined above.

Author Response 01 Dec 2017

Dongmin Jung, Avison Biomedical Research Center, Yonsei University, South Korea

Thank you for taking the time to read and review our paper. In the revised version, we have improved the clarity of the figures and the text in several places, according to your suggestions.

Necessary files are available at https://zenodo.org/record/1066236. Please, download K10090.rds and install the current version of this package.

In Figure 1, The horizontal axis represents the proportion of our interesting genes in a certain functional category.

In Figure 3, the 200 most significant categories from the GSEA are used, with the cutoff of 0.05 and 0.2 for p-values and edges, respectively. For GO terms, the two largest subnetworks are involved in transcription and signal transduction. We can see that Ras is related to tumorigenesis via the PI3K-Akt pathway from KEGG.
For the discussion of the functionally related proteins themselves, two figures are added. One is the heatmap for the relationship between top proteins and pathways in Figure 4. The other is visualization of PPI among proteins in Figure 5. Please, see the text for further discussion.

**Competing Interests:** No competing interests were disclosed.