

SOFTWARE

Open Access



# Netpredictor: R and Shiny package to perform drug-target network analysis and prediction of missing links

Abhik Seal  and David J. Wild\*

## Abstract

**Background:** Netpredictor is an R package for prediction of missing links in any given unipartite or bipartite network. The package provides utilities to compute missing links in a bipartite and well as unipartite networks using Random Walk with Restart and Network inference algorithm and a combination of both. The package also allows computation of Bipartite network properties, visualization of communities for two different sets of nodes, and calculation of significant interactions between two sets of nodes using permutation based testing. The application can also be used to search for top-K shortest paths between interactome and use enrichment analysis for disease, pathway and ontology. The R standalone package (including detailed introductory vignettes) and associated R Shiny web application is available under the GPL-2 Open Source license and is freely available to download.

**Results:** We compared different algorithms performance in different small datasets and found random walk supersedes rest of the algorithms. The package is developed to perform network based prediction of unipartite and bipartite networks and use the results to understand the functionality of proteins in an interactome using enrichment analysis.

**Conclusion:** The rapid application development environment like shiny, helps non programmers to develop fast rich visualization apps and we believe it would continue to grow in future with further enhancements. We plan to update our algorithms in the package in near future and help scientist to analyse data in a much streamlined fashion.

**Keywords:** Prediction, Shortest-path, Enrichment analysis, R shiny, Drug-target

## Background

Identifying missing associations between drugs and targets provides insights into polypharmacology and off-target mediated effects of chemical compounds in biological systems. Traditional machine learning algorithms like Naive Bayes, SVM and Random Forest have been successfully applied to predict drug target relations [1–4]. However, using supervised machine learning methods requires training sets, and they can suffer from accuracy problems through insufficient sampling or scope of training sets. During the last years, the field of semi-supervised learning has been applied to methods based on graphs or networks. The data points are represented as vertices of a network, while the links between the vertices

depend upon the labeled information. Thus, it is desirable to develop a predictive model based on using both labeled and unlabeled information. Recently several machine learning techniques provides effective and efficient ways to predict DTIs. One way to formulate the problem of DTI prediction as a binary classification problem, where the drug-target pairs are treated as instances, and the chemical structures of drugs and the amino acid subsequences of targets are treated as features. Then, classical classification methods can be used, e.g., support vector machines (SVM) and regularized least square (RLS). Liu et al. [33] have developed PyDTI package which mainly focuses on neighborhood regularized logistic matrix factorization (NRLMF). NRLMF uses logistic matrix factorization and neighbourhood regularization to prediction drug target pairs. The PyDTI package provides access to other algorithms for drug target prediction such as

\*Correspondence: [djwild@indiana.edu](mailto:djwild@indiana.edu)  
School of Informatics and Computing, Indiana University Bloomington,  
Informatics West, Bloomington, 47408 Indiana, USA



NetLapRLS, BLM-NII, KBMF-2k, CMF implemented in a single package. Bajic [17] have developed DDR package which combines multiple different similarity measures in the drug space and protein target space and optimizes using average entropy measures. Peska [12] developed bayesian ranking approach for drug target prediction. The novelty of the approach comes from “per-drug ranking” optimization criteria, while projecting drugs and targets to a shared latent space. Most of these methods are command line based and they need to have prior programming expertise to start the analysis. Netpredictor solves this problem by building an intuitive UI and giving users an easy way to interaction and perform prediction based on their data. The main advantages of network-based methods are:

- They use label information and as well as unlabeled data as input in the form of vectors.
- Once can use multiple classes inside the network structure.
- It uses multitude of paths to compute associations.
- Network based methods mostly use transductive learning strategy, in which the test set is unlabelled but while computation it uses the information from neighbourhood.

With the advent of the R open source statistical programming language [13] and the gaining popularity of the RShiny package for interface development around R [14] it has become straightforward for programmers to create and deploy web applications on windows and Linux servers. R and RShiny have already used in several biomedical applications. Table 1 shows some of these. We used R and R shiny to create Netpredictor standalone and web application respectively, which is freely available and open source. The web application framework in R allows creation of a simple intuitive user interface with dynamic filters and real-time exploratory analysis. Shiny also allows integration of additional R packages, Javascript libraries and CSS for customization. Web applications are accessible via browser or can be run locally on the user’s computer. The R package described in this paper provides utilities to compute recommendations in a bipartite network and well as unipartite network based on HeatS [15], Random walk with Restart (RWR) [23, 24], Network based inference (NBI) [16, 21, 22] and combination of RWR and NBI (netcombo). In order to understand the topology of the network, the package also provides ways to compute bipartite network properties such as degree centrality, density of the network, betweenness centrality, number of sets of nodes and total number of interactions for given bipartite network. The package also performs graph partitioning such as bipartite community detection using the lpbrim algorithm [18, 20] and

**Table 1** Table shows some lifescience related applications developed in R and shiny

Shiny Web Applications	Description
rcellminer [5]	Analysis of molecular profiling and drug response data.
PACMEN [6]	Analysis of gene expression profiles and network topology of cancer .
SynRio [7]	Analysis of cyanobacterial genome and interactive genome visualization.
Rchemcpp [8]	Identifies structural analogs in large databases such as ChEMBL, Drugbank and CMAP.
GOPlot [9]	Functional analysis of gene expression data.
PEAX [10]	Exploration of clinical phenotype and gene expression association
Methylation Plotter [11]	Exploration of DNA methylation sites over genome.

visualization of communities, network permutations to compute the significance of predictions and performance of the algorithms based on user given data.

The ranked list of proteins can be also used to understand any protein proteins interactions exist among them using subgraph extraction or it can be used to understand neighbouring PPIs in the interactome. Such kind of networks helps in understanding of pathogenic and physiologic mechanisms that trigger the onset and progression of diseases. To dig deeper into such cases, the list of proteins can be used to perform Gene Ontology, disease and pathway enrichment to understand the mechanism of action of proteins and whether if that target is a suitable target or not.

### Implementation

The Netpredictor package can be used in two ways - either in standalone form and compelling web application running locally or on an Amazon cloud server [25]. The web applications accessible through the Internet and standalone package are functionally identical. More details regarding the package accessibility and the instructions on how to use it via the web application and run locally are given in the “Availability and requirements” section. The interface consists of two parts - a web interface and a web server. Both of these components are controlled by code that is written within the framework of Shiny application in R. RShiny uses “reactive programming” which ensures that changes in inputs are immediately reflected in outputs, making it possible to build a highly interactive tool. Within the RShiny package, ordinary controllers or

widgets are provided for ease of use for application programmers. Many of the procedures like uploading files, refreshing the page, drawing new plots and tables are provided automatically. The communication between the client and server is done over the normal TCP connection. The data traffic that is needed for many of web applications between the browser and the server is facilitated over the websockets protocol. This protocol operates separately using handshake mechanism between the client and server is done over the HTTP protocol. The duplex connection is open all the time and therefore authentication is not needed when exchange is done. In order for an RShiny app to execute, we have to create an RShiny server. RShiny follows a pre-defined way to write R scripts. It consists of server.R and ui.R, which need to be in same directory location. If a developer wants to customize the user interface shiny can also integrate additional CSS and Javascript libraries within the web application. The GUI consists of introduction page with tab panels shown in Fig. 1. The first tab, start prediction, consists of sidebar panels and a main output panel Fig. 2. The sidebar is used to upload the data and select the algorithms and its parameters. The start prediction tab consists of data upload, compute recommendations, compute network properties and visualization of user given data. The advanced analysis tab has two sections the statistical analysis section and permutation testing tab. We computed the recommendations of the Drugbank database using NBI and included the predictions results in the Drugbank search tab.

In the PPI Network tab consist of three functionalities namely one can search for protein interaction from a list of proteins, search for top-k PPI shortest paths using Yen's algorithm [19] using both weighted and un-

weighted graphs. The algorithm executes  $O(n)$  times Dijkstra algorithm to search paths for each of the  $k$  shortest paths, so its time complexity is  $O(kn(m+n\log n))$ , where  $n$  is the number of nodes and  $m$  is the number of edges. Shortest path graph algorithm has been widely adopted to identify genes with important functions in a network [26–30].

We also provide sub-graph extraction from the PPI datasets using a large list of proteins using Consensus-PathDB [31] and string [32] databases.

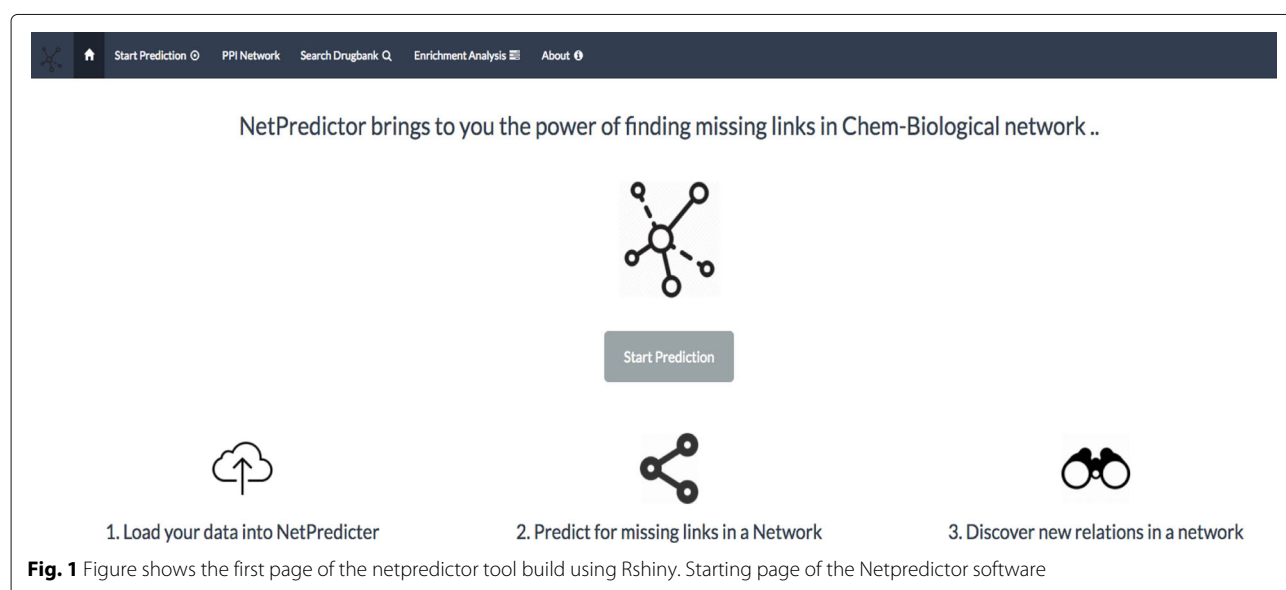
#### Main features of netpredictor standalone and web tool

The standalone R package application can perform prediction on unipartite networks using a set of different similarity measures between vertices of a graph in order to predict unknown edges (links) [34–36]. The prediction methods are classified into two categories:

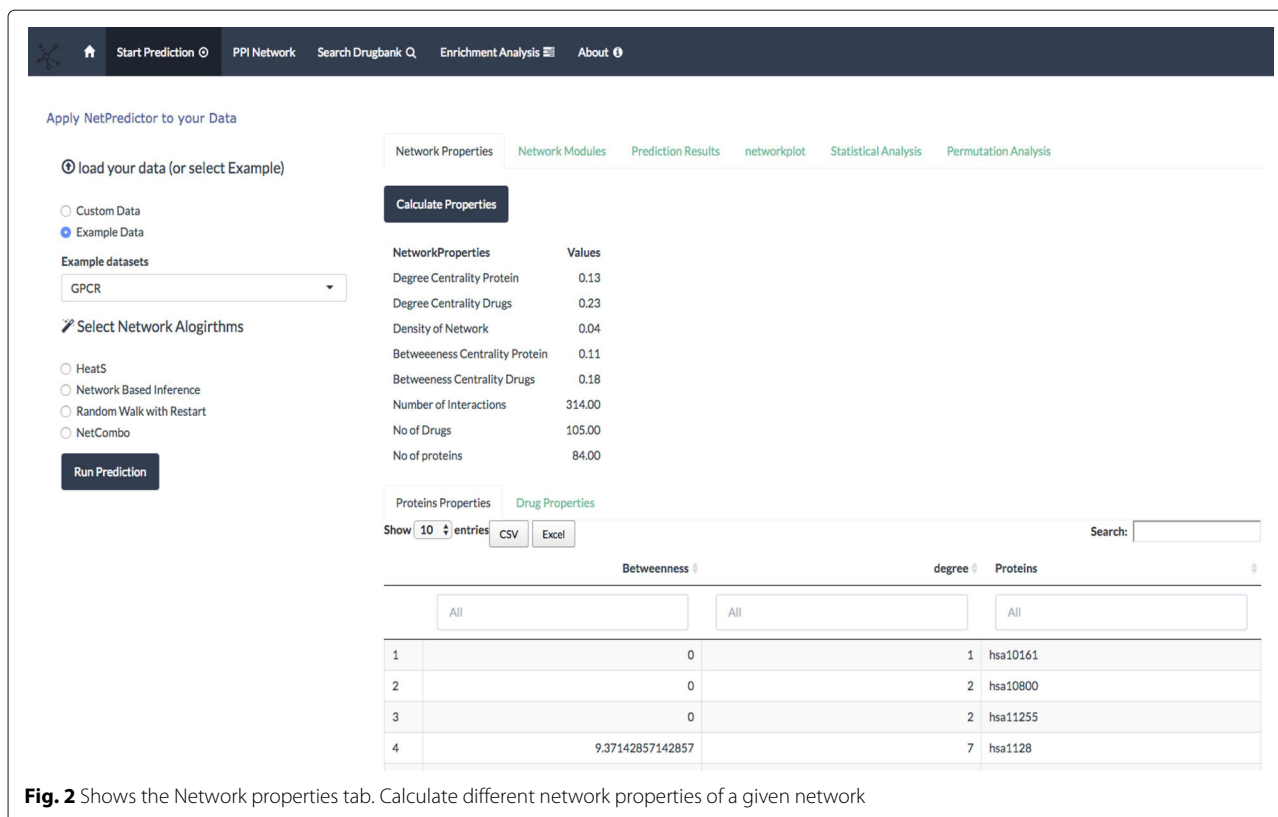
- Neighborhood based metrics and
- Path based metrics

For neighbourhood based metrics the methods which are implemented are (i) common neighbours (ii) jaccard coefficient [37] (iii) cosine similarity (iv) hub promoted index [38] (v) hub depressed index (vi) Adamic Adar index [39] (vii) Preferential attachment [40] (viii) Resource allocation [41] (ix) Leicht-Holme-Nerman Index [42]. Similarly using path-based metrics one can compute paths between two nodes as similarity between node pairs. The methods are:

- The local path based metric [43] uses the path of length 2 and length 3. The metric uses the information of the nearest neighbours and it also uses



**Fig. 1** Figure shows the first page of the netpredictor tool build using Rshiny. Starting page of the Netpredictor software



the information from the nodes within length of 3 distances from the current node.

- (ii) The Katz metric [44] is based on similarity of all the paths in a graph. This method counts all the paths between given pair of nodes with shorter paths counting more heavily. Parameters are exponential.
- (iii) Geodesic similarity metric calculates similarity score for vertices based on the shortest paths between two given vertices.
- (iv) Hitting time [45] is calculated based on a random walk starts at a node  $x$  and iteratively moves to a neighbor of  $x$  chosen uniformly at random. The Hitting time  $H_{x,y}$  from  $x$  to  $y$  is the expected number of steps required for a random walk starting at  $x$  to reach  $y$ .
- (iv) Random walk with restart [16, 45, 46] is based on pagerank algorithm [47]. To compute proximity score between two vertexes we start a random walker at each time step with the probability  $1 - c$ , the walker walks to one of the neighbors and with probability  $c$ , the walker goes back to start node. After many time steps the probability of finding the random walker at a node converges to the steady-state probability.

The significance of interaction of links is based on random permutation testing. A random permutation test compares the value of the test statistic predicted data value to the distribution of test statistics

when the data are permuted. Supporting Information S1\_NetpredictorVignette provides tutorial for this netpredictor standalone R package. In the web application app one can load their own data or can use the given sample datasets used in the software. For the custom dataset option one needs to upload bipartite adjacency matrix along with the drug similarity matrix and protein sequence matrix. From the given datasets Enzyme, GPCR, Ion Channel and Nuclear Receptor in the application one can load the data and set the parameters for the given algorithms and start computations. The data structure the web application accepts matrix format files for computation.

A summary of the contents of each of the tabs shiny netpredictor application is reported in Table 2.

#### Start prediction tab

The *start prediction* tab is designed to upload a network in matrix format and compute its properties, searching for modules, fast prediction of missing interactions, visualization of bipartite modules and predicted network. For the custom dataset, in the input drug-target binary matrix, target nodes should be in rows and drug nodes in the columns. The drug similarity matrix and the target similarity should have the exact number of drugs and targets from the binary matrix. For HeatS, only the bipartite network is used to compute the recommendation of links.

**Table 2** Table shows the functions of tabs in Shiny web application

Interface Tabs	Description
Load data and select algorithms	The load data and selection of algorithms panel allows users to load custom data or example datasets in matrix format. Users need to upload the matrices binary drug-target bipartite network, drug –drug similarity and protein – protein similarity along with algorithm and parameters of choice.
Network Properties	The network properties several different properties of the bipartite graph such as the degree centrality of two types of nodes, density of the network, betweenness of two types of nodes, total number of interactions, count of each type of nodes.
Network Modules	Bipartite network modules are computed using the lpbim algorithm [49] the tab shows the bipartite nodes as tables and module network. Modules are dynamically updated based on input data.
Prediction Results	For a given dataset to compute the results one can select any one of the algorithms . The results are shown using the jquery Data Tables library. The table shows the drugs, targets, pvalues, outcome ( True/predicted interaction).
Network Plot	The network plot tab plots the computed predicted network. It uses visNetwork package which uses the vis.js library to generate network. The predicted interactions are marked in dashed lines and true interactions are marked with bold lines. Drop downs are provided to select specific nodes and groups.
Statistical Testing	The statistical testing tab tests the performance of the model based one the random removal of true links from the network based on the frequency of the drug-target associations. It measures the auac, auc, auctop(10%), bedroc and enrichment of links. Based on these scores we select which algorithm to use.
Permutation Testing	In permutation testing significance of the associations are calculated by, randomly permuting the matrices and and compute the significance using, standard normal distribution.

**Table 2** Table shows the functions of tabs in Shiny web application (*Continued*)

Search Drugbank	This tab allows users to search predicted drug target associations from the drugbank database from 5970 drugs and 3797 proteins from a total of 316645 predicted and 14167 true interactions.
Ontology and Pathway Search	This tab allows users to search for enrichment of Ontologies and Pathways using a given set of genes.

For RWR, NBI, and Netcombo all of these require three matrices. The default parameters are already being set for the algorithms. The main panel of the start prediction tab has four tabs that compute network properties, network modules, the prediction results and predicted network plot.

Bipartite network properties are calculated by transforming the network in to one-mode networks (contain one set of nodes) called projection of the network in which a bipartite network of drugs and proteins two drugs are connected if they share a single protein similarly two proteins are connected if they share a single drug molecule. Using the two-projected network of drugs and proteins we compute degree centrality, betweenness, total number of interactions, total number of each of the nodes and distribution of the drug and target nodes shown in Fig. 2. WE have implemented the visualization of cousts and betweenness histograms using the rCharts R package [48]. Bipartite network modules are computed using the lpbim algorithm [49] for which lpbim R package is used [20]. The algorithm consists of two stages. First, during the LP phase, neighboring nodes (i.e. those which share links) exchange their labels representing the community they belong to, with each node receiving the most common label amongst its neighbors. The process is iterated until densely connected groups of nodes reach a consensus of what is the most representative label, as indicated by the fact that the modularity is not increased by additional exchanges. Second, the BRIM algorithm (2) refines the partitions found with label propagation. HeatS and network based inference compute (NBI) recommendations using a bipartite graph , where a two phase resource transfer Information from set of nodes in A gets distributed to B set of nodes and then again goes back to resource A . This process allows us to define a technique for the calculation of the weight matrix  $W$ . HeatS uses only the drug target bipartite data matrix and NBI uses similarity matrices of drug chemical similarity matrix and protein similarity matrix. The random walk with restart (RWR) algorithm uses all the three different matrices to compute the recommendations. Netcombo computes

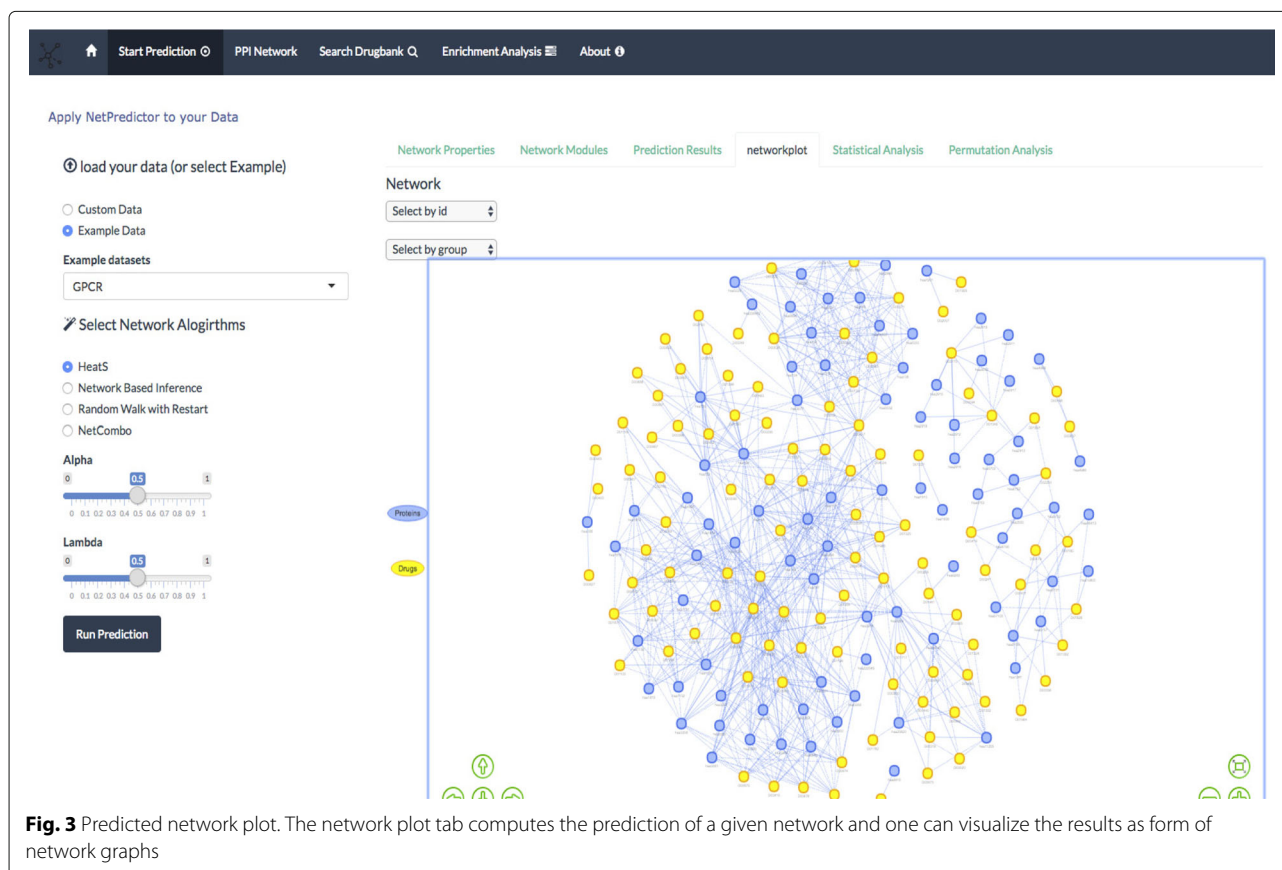
both NBI and RWR and then averages the scores. The prediction results tab shows the computed results using the javascript library DataTables [52]. The data table provides columns filters and search options. The network plot tab represent the network using the visNetwork R package [53]. The Network visualization is made using vis.js javascript library. Javascript libraries can be integrated using a binding between R and javascript data visualization libraries Fig. 3. The htmlwidgets library [54] can generate a web based plot by just calling a function that looks like any other R plotting function.

One can also perform advance analysis using two tabs namely - statistical analysis tab and permutation testing. The statistical analysis tab computes the performance of the algorithms. Three algorithms are network based inference, random walk with restart and netcombo can be used. One can randomly remove the true links from the network using frequency of the drug target interactions in the network. The performance of the algorithm is checked when the removed links are repredicted. The statistics used to evaluate the performance is AUAC, AUC, AUC-TOP(10%), Boltzmann-enhanced discrimination of ROC (BEDROC) [55] and enrichment factor(EF). The data table gets automatically updated for each of the computations. The results are reported in main panel using data tables.

The significance of interactions using random permutations can be computed for the given network using network based inference and random walk with restart. The networks are randomized and significance of the interactions are calculated based on standard normal distribution. The user needs to give total number of permutations to compute and the significant interactions to keep.

### PPI network

In the current application we used human protein-protein interaction (PPI) data from both consensus-pathDB(CPDB) and string DB. The data sources are converted to igraph objects for faster loading and computation. We have implemented top-K shortest paths search using Yen's algorithm ([19]), with PPI in both the datasets. The multiple shortest path proteins can be enriched for reactome pathways using over-representation analysis. We also provide sub-graph extraction from the PPI datasets using a large list of proteins. can be useful for connecting sources to targets in protein networks, a problem that has been the focus of many studies in the past which include discovering genomic mutations that are responsible for changes in downstream gene expression [50] studying interactions between different cellular processes [51] and linking environmental stresses through receptors



to transcriptional changes. The details are of the PPI tab are discussed in the supplemental information.

The *drugbank tab* helps to search predicted interactions computed using NBI method using the drugbank database. One can search for targets given a specific drugbank ID and search for drugs given a specific hugo gene name. The *Enrichment Analysis* tab helps to search the relevant gene ontology terms, pathways and diseases for a given list of genes. A search can be made based on predicted proteins and in order to understand its function, location and pathway this tab can help to understand it. The level of ontology can also be given to the user input. We used biomart services using the biomART R package to convert genes names to entrez ids and then the clusterProfiler R package ([60]) to retrieve the gene ontology lists. The pathway enrichment is based on the ReactomePA R package ([61]).

### Search drugbank tab

The *drugbank tab* Fig. 4 helps to search predicted interactions computed using NBI method using the drugbank database [56]. One can search for targets given a specific drugbank ID and search for drugs given a specific hugo gene name [57]. In Fig. 3 the data table shows the drug target significant scores whether it is a true or predicted interaction, Mesh categories of drugs, ATC codes and groups (approved, illicit, withdrawn, investigational, experimental). Currently the drugbank search tab only supports data computed using Network based inference.

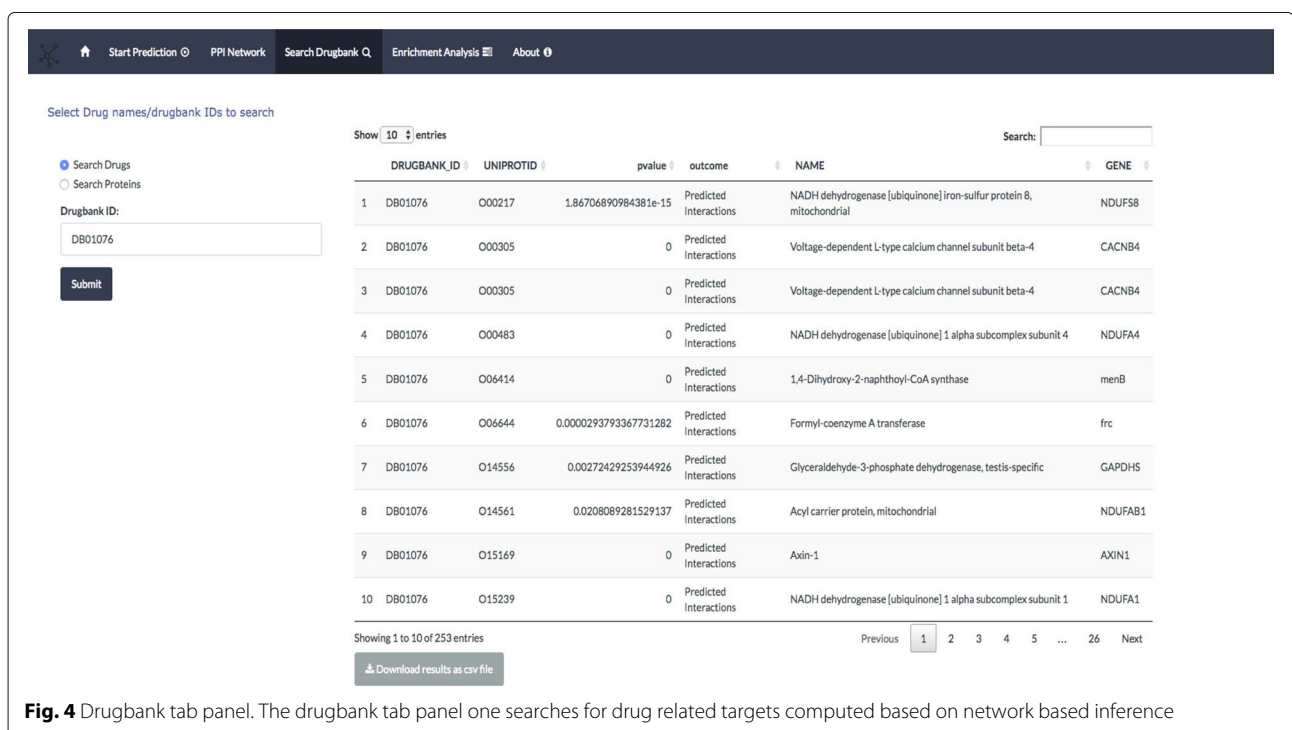
The computed results and the associated meta-data are stored in a sqlite database [58] for access through shiny data tables interface.

### Ontology and pathway search tab

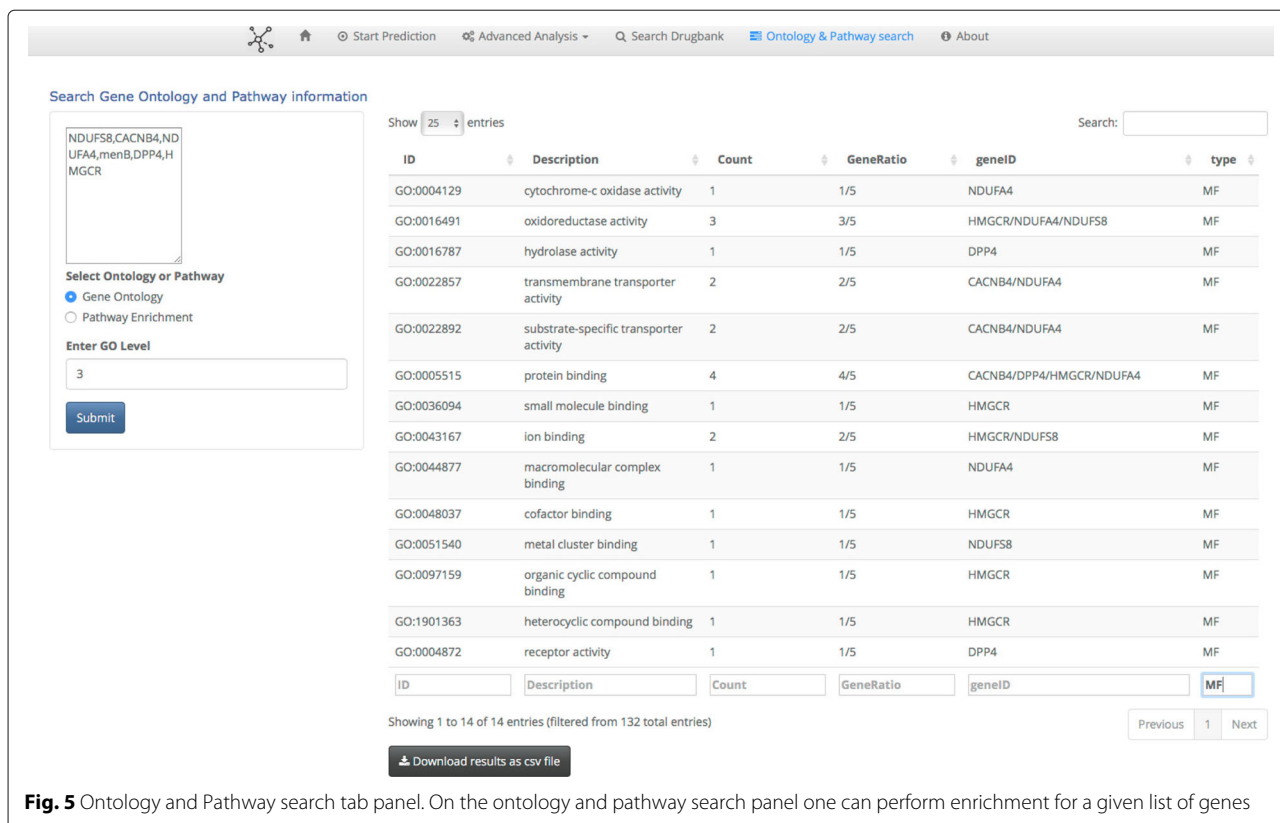
The *Ontology and pathway search tab* Fig. 5 helps to search the relevant gene ontology terms and pathways for a given set of genes. A search can be made based on predicted proteins and in order to understand its function, location and pathway this tab can help to understand it. The level of ontology can also be given to the user input. We used biomart services using the biomART R package [59] to convert genes names to entrez ids and then the clusterProfiler R package [60] to retrieve the gene ontology lists. The pathway enrichment is based on the ReactomePA R package [61].

### Results and discussion

In this section we illustrate the use of Netpredictor package in prediction of drug target interactions and analysis of networks. The information about the interactions between drugs and target proteins was obtained from Yamanishi et al. [62] where the number of drugs 212, 99, 105 and 27, interacting with enzymes, ion channels, GPCRs and nuclear receptors respectively. The numbers of the corresponding target proteins in these classes are 478, 146, 84 and 22 respectively. The numbers of the corresponding interactions are 1515, 776, 314 and 44. We performed both network based inference and Random



**Fig. 4** Drugbank tab panel. The drugbank tab panel one searches for drug related targets computed based on network based inference



**Fig. 5** Ontology and Pathway search tab panel. On the ontology and pathway search panel one can perform enrichment for a given list of genes

walk with restart on all of these datasets. To check the performance we randomly removed 20% of the interactions from each of the dataset and computed the performance 50 times and calculated the mean performance of each of these methods. The results are given in Table 3. Clearly, RWR supersedes its performance compared to network based inference in Enzyme and the GPCR dataset. However, computation of NBI algorithm takes less amount of time than RWR. For the drugbank tab we download the latest drugbank set version 4.3 and created a drug target interaction list of 5970 drugs and 3797 proteins We computed similarities of drugs using RDkit [64] ECFP6 fingerprint and local sequence similarity of proteins using smith waterman algorithm and normalized using the procedure proposed by Bleakley and Yamanishi [65] and integrated the matrices for network based inference computation. We ran the computations 50 times and kept the significant drug target relations ( $p \leq 0.05$ ) where a total of 316645 predicted interactions and 14167 true interactions present in the system.

**Conclusions**

In this paper we presented netpredictor, a standalone and web application for drug target interaction prediction. Netpredictor uses a shiny framework to develop web pages and the application can be accessed from web browsers. To set up the Netpredictor application locally

there are some additional requirements other than shiny which are given below,

- Firstly, the user has to have the R statistical environment installed, for which instructions can be found in R software home page.
- Secondly, the devtools R package [63] has to be installed. The package can be installed using devtools R package.
- Also for fast computation Microsoft R Open package needs to be installed which can be obtained from <https://mran.revolutionanalytics.com/documents/>

**Table 3** Table shows the performance of RWR and NBI on different datasets

AUAC	AUC	AUCTOP	BDR	EFC	Dataset	Method
0.934	0.899	0.577	0.506	8.38	Enzyme	RWR
0.823	0.882	0.274	0.252	5.066	GPCR	RWR
0.841	0.88	0.283	0.254	6.28	Ion Channel	RWR
0.585	0.698	0.018	0.089	0.295	Nuclear Receptor	RWR
0.834	0.879	0.636	0.429	4.633	Enzyme	NBI
0.767	0.466	0.281	0.217	4.3	GPCR	NBI
0.874	0.938	0.349	0.284	6.756	Ion Channel	NBI
0.487	0.52	0.049	0.05	0.309	Nuclear Receptor	NBI



[rro/installation/](#). Microsoft R Open includes multi-threaded math libraries to improve the performance of R. R is usually single threaded but if its linked to the multi-threaded BLAS/LAPACK libraries it can perform in multi-threaded manner. This usually helps in matrix multiplications, decompositions and higher level matrix operations to run in parallel and minimize computation times.

- After installing R, R open and shiny calling `shiny::runGitHub('Shiny_NetPredictor', 'abhik1368')`

This will load all the libraries need to run `netpredictor` in browser. The application can be accessed in any of the default web browsers. The `netpredictor` R package (<https://github.com/abhik1368/netpredictor>) and the Shiny Web application ([https://github.com/abhik1368/Shiny\\_NetPredictor](https://github.com/abhik1368/Shiny_NetPredictor)) is freely available. Users can follow the “Issues” link on the GitHub site to report bugs or suggest enhancements. In future the intention is to include Open Biomedical Ontologies for proteins to perform enrichment analysis. The package is scalable for further development integrating more algorithms.

## Availability and requirements

**Project name:** shiny\_Netpredictor

**Project home page:** <https://github.com/abhik1368/ShinyNetPredictor>

**Operating system(s):** Platform independent

**Programming language:** R

**Other requirements:** R environment including digest and tools packages. Tested on R version 3.4

**License:** GNU GPL

**Any restrictions to use by non-academics:** no restrictions

## Abbreviations

ATC: Anatomical therapeutic chemical; ECFP: Extended connectivity fingerprints; NBI: Network based inference; PPI: Protein - protein interactions; RWR: Random walk with restart

## Acknowledgements

Authors wish to thank anonymous reviewers for their critiques and constructive comments which significantly improved this manuscript. Authors also wish to acknowledge these individuals for their comments on this project: Dr. Yong Yeol Ahn and Dr. Ying Ding. Authors would also like to thank BMC editors who have waived 50% of the article processing fee.

## Funding

No funding was received for this study.

## Authors' contributions

Conceived and designed the experiments and tool: AS. Performed the experiments: AS. Analyzed the data: AS. Contributed reagents/materials/analysis tools: AS, DJW. Wrote the paper: AS, DJW. Interpreted the results, drafted the manuscript and contributed to revisions: AS, DJW. Read and approved the final manuscript: AS, DJW.

## Ethics approval and consent to participate

Not Applicable.

## Consent for publication

Not Applicable.

## Competing interests

The authors declare that they have no competing interests.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 25 January 2018 Accepted: 18 June 2018

Published online: 16 July 2018

## References

1. Cao DS, Liang YZ, Yan J, Tan GS, Xu QS, Liu S. PyDPI: Freely Available Python Package for Chemoinformatics, Bioinformatics, and Chemogenomics Studies. *J Chem Inf Model*. 2013;53(11):3086–3096. <https://doi.org/10.1021/ci400127q>.
2. Cao DS, Liang YZ, Deng Z, Hu QN, He M, Xu QS, Zhou GH, Zhang LX, Zx Deng, Liu S. Genome-Scale Screening of Drug-Target Associations Relevant to Ki Using a Chemogenomics Approach. *PLoS one*. 2013;8(4):e57680.
3. van Westen GJP, Wegner JK, Ilzerman AP, van Vlijmen HWT, Bender A. Proteochemometric modeling as a tool to design selective compounds and for extrapolating to novel targets. *Med Chem Comm*. 2010;2:16–30.
4. Paricharak S, Cortés-Ciriano I, Ilzerman AP, Malliavin TE, Bender A. Proteochemometric modelling coupled to in silico target prediction: an integrated approach for the simultaneous prediction of polypharmacology and binding affinity/potency of small molecules. *J Cheminformatics*. 2015;7:15.
5. Luna A, Rajapakse VN, Sousa FG, Gao J, Schultz N, Varma S, Reinhold W, Sander C, Pommier Y. rcellminer: exploring molecular profiles and drug response of the NCI-60 cell lines in R. *Bioinformatics*. 2016;32(8):1272–1274.
6. Ghazanfar S, Yang JY. Characterizing mutation-expression network relationships in multiple cancers. *Comput Biol Chem*. 2016;63:73–82.
7. Lakshmanan K, Peter AP, Mohandass S, Varadharaj S, Lakshmanan U, Dharmar P. SynRio: R and Shiny based application platform for cyanobacterial genome analysis. *Bioinformatics*. 2015;11(9):422–5.
8. Klambauer G, Wischenbart M, Mahr M, Unterthiner T, Mayr A, Hochreiter S. Rchemcpp: a web service for structural analoging in ChEMBL, Drugbank and the Connectivity Map. *Bioinformatics*. 2015;31(20):3392–4.
9. Walter W, Sánchez-Cabo F, Ricote M. GOplot: an R package for visually combining expression data with functional analysis. *Bioinformatics*. 2015;31(17):2912–4.
10. Hinterberg MA, Kao DP, Bristow MR, Hunter LE, Port JD, Görg C. Peax: interactive visual analysis and exploration of complex clinical phenotype and gene expression association. *Pac Symp Biocomput*. 2015:419–30. [https://doi.org/10.1142/9789814644730\\_0040](https://doi.org/10.1142/9789814644730_0040).
11. Mallona I, Diez-Villanueva A, Peinado MA. Methylation plotter: a web tool for dynamic visualization of DNA methylation data. *Source Code Biol Med*. 2014;9:11. <https://doi.org/10.1186/1751-0473-9-11>. eCollection 2014.
12. Peska L, Buza K, Koller J. Drug-target interaction prediction: A Bayesian ranking approach. *Comput. Methods Programs Biomed*. 2017;152:15–21.
13. R Core Team. R: A Language and Environment for Statistical Computing. 2013. Available from: <http://www.r-project.org/>.
14. Chang W, Cheng J, Allaire J, Xie Y, McPherson J. shiny: Web Application Framework for R. 2015. R package version 0.11.1. Available from: <http://CRAN.R-project.org/package=shiny>.
15. Zhou T, et al. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proc Natl Acad Sci USA*. 2010;107:4511–5.
16. Zhou T, et al. Bipartite network projection and personal recommendation. *Phys Rev E Stat Nonlin Soft Matter Phys*. 2007;76:046115.
17. Olayan RS, Ashoor H, Bajic VB. DDR: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. *Bioinformatics*. 2018;34(7):1164–73. <https://doi.org/10.1093/bioinformatics/btx731>.
18. Liu X, Murata T. Community Detection in Large-Scale Bipartite Networks. *IEEE Comput Soc*. 2009;1:50–57.
19. Yen JY. Finding the K Shortest Loopless Paths in a Network. *Management Sci*. 1971;17(11):712–716.
20. Poisot T. lpbtrim: Optimization of bipartite modularity using LP-BRIM (Label propagation followed by Bipartite Recursively Induced Modularity). R package version 1.0.0 2015.

21. Cheng F, et al. Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput Biol*. 2012;8:e1002503.
22. Alaimo S, Pulvirenti A, Giugno R, Ferro A. Drug-target interaction prediction through domain-tuned network-based inference. *Bioinformatics*. 2013;29(16):2004–8.
23. Chen X, et al. Drug–target interaction prediction by random walk on the heterogeneous network. *Mol BioSyst*. 2012;8:1970–8.
24. Seal A, Ahn Y, Wild DJ. Optimizing drug target interaction prediction based on random walk on heterogeneous networks. *J Cheminformatics*. 2015;7:40. <https://aws.amazon.com/documentation/ec2/>.
25. <https://aws.amazon.com/documentation/ec2/>.
26. Chen L, Huang T, Zhang YH, Jiang Y, Zheng M, Cai YD. Identification of novel candidate drivers connecting different dysfunctional levels for lung adenocarcinoma using protein–protein interactions and a shortest path approach. *Sci Rep*. 2016;6:29849.
27. Jiang M, Chen Y, Zhang Y, Chen L, Zhang N, Huang T, Cai YD, Kong XY. Identification of hepatocellular carcinoma related genes with k-th shortest paths in a protein–Protein interaction network. *Mol BioSyst*. 2013;9:2720–8.
28. Chen L, Xing Z, Huang T, Shu Y, Huang G, Li HP. Application of the shortest path algorithm for the discovery of breast cancer related genes. *Curr Bioinform*. 2016;11:51–8.
29. Li BQ, Huang T, Liu L, Cai YD, Chou KC. Identification of colorectal cancer related genes with mRMR and shortest path in protein–protein interaction network. *PLoS ONE*. 2012;7:e33393.
30. Chen L, Yang J, Huang T, Kong XY, Lu L, Cai YD. Mining for novel tumor suppressor genes using a shortest path approach. *J Biomol Struct Dyn*. 2016;34:664–75.
31. Kamburov A, Stelzl U, Lehrach H, Herwig R. The ConsensusPathDB interaction database: 2013 update. *Nucleic Acids Res*. 2013;41(D1):D793–800.
32. Szklarczyk D, Morris JH, Cook H, Kuhn M, Wyder S, Simonovic M, Santos A, Doncheva NT, Roth A, Bork P, Jensen LJ, von Mering C. The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Res*. 2017;45:D362–68.
33. Liu Y, Wu M, Miao C, Zhao P, Li X-L. Neighborhood Regularized Logistic Matrix Factorization for Drug-Target Interaction Prediction. *PLoS Comput Biol*. 2016;12(2):e1004760.
34. Liben-Nowell D, Kleinberg JM. The link prediction problem for social networks. *J Comput Aided Mol Des*. 2003;CIKM:556–9.
35. Hasan MA, Zaki MJ. A survey of link prediction in social networks. *Soc Netw Data Analytics*. 2011:243–75.
36. Liben-Nowell D, Kleinberg JM. The link prediction problem for social networks. *J Comput Aided Mol Des*. 2003;CIKM:556–9.
37. Jaccard P. Etude comparative de la distribution florale dans une portion des alpes et de jura. *Bull Soc Vaudoise Sci Nat*. 1901;37:547–79.
38. Ravasz E, Somera AL, Mongru DA, Oltvai ZN, Barabasi A-L. Hierarchical organization of modularity in metabolic networks. *Science*. 2002;297:1553.
39. Adamic LA, Adar E. Friends and neighbors on the web. *Soc Networks*. 2002;25(3):211–30.
40. Barabasi AL, Albert R. Emergence of scaling in random networks. *Science*. 1999;286:509–12.
41. Zhou T, Lu L, Zhang YC. Predicting missing links via local information. *Eur Phys JB*. 2010;71:623–30.
42. Leicht EA, Holme P, Newman MEJ. Vertex similarity in networks. *Phys Rev E*. 2006;73:026120.
43. Lu L, Jin CH, Zhou T. Similarity index based on local paths for link prediction of complex networks. *Phys Rev E*. 2009;046122:80.
44. Katz L. A new status index derived from sociometric analysis. *Psychometrika*. 1953;18:39–43.
45. Fouss F, Pirotte A, Renders J-M, Saerens M. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans Knowl Data Eng*. 2007;19:355–69.
46. Kohler S, Bauer S, Horn D, Robinson PN. Walking the interactome for prioritization of candidate disease genes. *Am J Hum Genet*. 82:949–958.
47. Langville AN, Meyer CD. *Google's pagerank and beyond: the science of search engine rankings*. Princeton University Press; 2012.
48. rCharts. [cited 4.1.2016]. Available from: <https://ramnathv.github.io/rCharts/>.
49. Barber M. Modularity and community detection in bipartite networks. *Phys Rev E*. 2007;76:066102.
50. Suthram S, Beyer A, Karp RM, et al. eQED: an efficient method for interpreting eQTL associations using protein networks. *Mol Syst Biol*. 2008;4:162. 10.1038/msb.2008.4.
51. Yosef N, Zalcvar E, Rubinstein AD, et al. ANAT: a tool for constructing and analyzing functional protein networks. *Sci Signal*. 2011;4(196):. p11. 10.1126/scisignal.2001935.
52. DataTables. [cited 4.1.2016]. Available from: <https://www.datatables.net/>.
53. visNetwork. [cited 4.1.2016]. Available from: <http://dataknowledge.github.io/visNetwork/>.
54. Htmlwidgets. [cited 4.1.2016]. Available from: <http://www.htmlwidgets.org/>.
55. Truchon J-F, Bayly CI. Evaluating VS methods: good and bad metrics for the early recognition problem. *J Chem Inf Model*. 2007;47:488–508.
56. Wishart DS, Knox C, Guo AC, Shrivastava S, Hassanali M, Stothard P, Chang Z, Woolsey J. DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res*. 2011;39(Database issue):D514–9. Epub 2010 Oct 6.
57. Gray KA, Yates B, Seal RL, Wright MW, Bruford EA. Genenames.org: the HGNC resources in 2015. *Nucleic Acids Res*. 2015;43(Database issue):D1079–85. <https://doi.org/10.1093/nar/gku1071>. Epub 2014 Oct 31.
58. <https://www.sqlite.org/>.
59. Durinck S, Moreau Y, Kasprzyk A, Davis S, De Moor B, Brazma A, Huber W. BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*. 2005;21:3439–40.
60. Yu G, Wang L, Han Y, He Q. clusterProfiler: an R package for comparing biological themes among gene clusters. *OMICS J Integr Biol*. 2012;16(5):284–7.
61. Yu G, He Q. ReactomePA: an R/Bioconductor package for reactome pathway analysis and visualization. *Mol BioSyst*. 2016;12:477–9.
62. Yamanishi Y, Araki M, Gutteridge A, Honda W, Kanehisa M. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*. 2008;24:i232–40.
63. Devtools by Hadley Wickham. <https://github.com/hadley/devtools>.
64. RDKit. Cheminformatics and Machine Learning Software. 2013. <http://www.rdkit.org>.
65. Bleakley K, Yamanishi Y. Supervised prediction of drug–target interactions using bipartite local models. *Bioinformatics*. 2009;25:2397–403.

**Ready to submit your research? Choose BMC and benefit from:**

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more [biomedcentral.com/submissions](https://biomedcentral.com/submissions)

