TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 10/10 pp542–552
DOI: 10.26599/TST.2019.9010056
Volume 25, Number 4, August 2020

NetEPD: A Network-Based Essential Protein Discovery Platform

Jiashuai Zhang, Wenkai Li, Min Zeng, Xiangmao Meng, Lukasz Kurgan, Fang-Xiang Wu, and Min Li*

Abstract: Proteins drive virtually all cellular-level processes. The proteins that are critical to cell proliferation and survival are defined as essential. These essential proteins are implicated in key metabolic and regulatory networks, and are important in the context of rational drug design efforts. The computational identification of the essential proteins benefits from the proliferation of publicly available protein interaction datasets. Scientists have developed several algorithms that use these interaction datasets to predict essential proteins. However, a comprehensive web platform that facilitates the analysis and prediction of essential proteins is missing. In this study, we design, implement, and release NetEPD: a network-based essential protein discovery platform. This resource integrates data on Protein–Protein Interaction (PPI) networks, gene expression, subcellular localization, and a native set of essential proteins. It also computes a variety of node centrality measures, evaluates the predictions of essential proteins, and visualizes PPI networks. This comprehensive platform functions by implementing four activities, which include the collection of datasets, computation of centrality measures, evaluation, and visualization. The results produced by NetEPD are visualized on its website, and sent to a user-provided email, and they are available to download in a parsable format. This platform is freely available at http://bioinformatics.csu.edu.cn/netepd.

Key words: essential proteins; centrality; data integration; evaluation; visualization

1 Introduction

Rapid developments and the widespread use of high-throughput techniques have resulted in the accumulation of a large quantity of information on Protein–Protein Interactions (PPIs)[1–4]. Generally, these PPI data are represented as a graph, where nodes represent proteins and edges denote interactions between proteins[5]. Essential proteins are a very important type of proteins and play important roles in biological activities. If one of them has been removed, the organism cannot survive or develop[6]. The determination of essential proteins can help us to understand the minimum requirements of a cell[7]. In addition, it can provide theoretical support for finding new drugs and determining the underlying mechanism of diseases[8–11]. The centrality-lethality rule proposed by Jeong et al.[12] indicates that the essentiality of protein molecules is closely related to the degree of a node in the PPI networks and provides a theoretical basis to identify and study essential proteins. Numerous topological features extracted from the PPI networks have been used to predict essential proteins[13–17]. Moreover, other types of relevant information, such as gene expression data and protein domains, have been used to predict essential proteins[18–22].

The large number and breadth of the abovementioned

• Jiashuai Zhang, Wenkai Li, Min Zeng, Xiangmao Meng, and Min Li are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China. E-mail: jiashuaizhang@csu.edu.cn; lwktechnology@csu.edu.cn; zengmin@csu.edu.cn; mxmanhui@csu.edu.cn; limin@mail.csu.edu.cn.
• Lukasz Kurgan is with the Department of Computer Science, Virginia Common-wealth University, Richmond, VA 23284-2512, USA. E-mail: lkurgan@vcu.edu.
• Fang-Xiang Wu is with the Division of Biomedical Engineering and Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, S7N 5A9, Canada. E-mail: faw341@mail.usask.ca.
* To whom correspondence should be addressed.
Manuscript received: 2019-09-11; accepted: 2019-09-16

efforts to characterize and predict essential proteins motivate the development of resources that would support these endeavors. To this end, we design, implement, and release NetEPD: a network-based essential protein discovery platform. While several tools that can support some of the activities related to the prediction of essential proteins have been developed, they have a range of limitations. First, most of them are desktop applications and this requires the end users to install and run them locally on their hardware. The installation process could be cumbersome, especially when environment variables have to be configured, and this could present an unsurmountable challenge for less computer-savvy users. Second, they rarely provide access to preloaded data, and instead require the end users to preprocess and load PPI networks, gene expression data, and other relevant datasets. Third, they do not offer facilities to compare the results of essential protein predictions. Overall, none of these tools offers a comprehensive suite of features that would cover the computation of a broad range of topological features, preloaded datasets, supports of assessment activities, and visualization of the resulting networks online. Section 3.2 provides a detailed comparative analysis. Our solution addresses these shortcomings. NetEPD is a convenient webserver that does not require installation and performs all computations on the server side. It incorporates commonly used and preprocessed datasets (PPI networks, gene expression datasets, and annotations of subcellular locations of genes), and integrates the calculations of over 20 topological features in order to support the prediction of essential proteins. NetEPD also uses a dataset of essential proteins to evaluate and compare predicted results by different algorithms and provides multiple options to visualize PPI networks. In short, NetEPD is a comprehensive and convenient platform that supports research toward the prediction and characterization of the essential proteins.

2 Implementation

NetEPD is designed and developed as a freely available webserver. The scope and architecture of NetEPD are summarized in Fig. 1. Users only need a modern web browser and an internet connection to access and use our resource. They can process their queries and view the results directly at the NetEPD website (http://bioinformatics.csu.edu.cn/netepd). Our platform was designed and developed using Java technology, which makes it secure and portable. More specifically, the webserver was implemented with HTML5, CSS3, and JavaScript using the Spring and Hibernate development framework. This type of framework also facilitates maintenance and futures expansion for the NetEPD platform.

2.1 Datasets integrated into NetEPD

Research has shown that several types of information are useful for the identification of essential proteins[23,24]. For instance, Li et al.[23] combined gene expression, orthology, and subcellular localization to augment PPI networks with spatial and temporal information, showing that such augmentation is beneficial for the prediction of essential proteins. Correspondingly, NetEPD includes these spatiotemporal characteristics. In particular, our platform contains four types of information: essential proteins, PPI networks, gene expression data, and

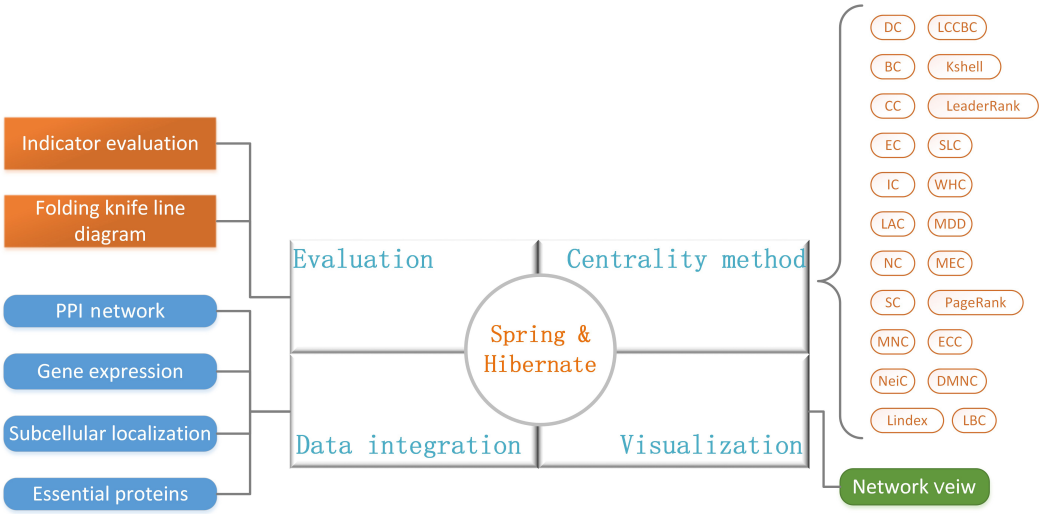


Fig. 1 Scope and architecture of the NetEPD resource. NetEPD combines two software frameworks: Spring and Hibernate, and implements four activities: collection of datasets, computation of network centrality measures, evaluation, and visualization.

subcellular localizations. These data are preloaded for two popular model species: house mouse (Mus musculus) and yeast (Sacchromyces cerevisiae). These data have been preprocessed by mapping to the UniProt database[25] to ensure the consistency between datasets.

PPI networks. NetEPD includes four PPI datasets which can be selected by the user for topological analysis. These datasets were collected from BioGRID[26], Database of Interacting Proteins (DIP)[27], Munich Information Center for Protein Sequences (MIPS)[28], and Molecular INTeraction database (MINT)[29] resources. We preprocessed these datasets to remove repeated protein–protein interactions and self-interactions. This is needed in the context of characterization and prediction of essential proteins. The contents of these four datasets are summarized in Table 1.

Gene expression data. Yeast gene expression data were downloaded from the gene expression omnibus database (No. GSE3431)[30] in the national center for biotechnology information and included in NetEPD. This dataset contains three metabolic cycles of yeast, each with 12 time intervals of approximately 25 min[31].

Subcellular localization data. Subcellular localization information of proteins was collected from the COMPARTMENTS database[32] and incorporated into NetEPD. Proteins are assigned into one of eleven cellular compartments: cytoskeleton, cytosol, endoplasmic reticulum, endosome, extracellular, mitochondrion, Golgi apparatus, lysosome, nucleus, peroxisome, and plasma.

Dataset of essential proteins. The list of essential protein data was obtained from the Database of Essential Genes (DEG)[33]. This dataset has 8393 eukaryotic proteins. We mapped 3224 of them to mouse and yeast proteomes. This protein list was loaded into NetEPD to assist with the characterization of essential proteins and assessment of their predictions.

2.2 Topological features generated from the PPI networks

One of the arguably strongest determinants that can be used to identify essential proteins is their topological feature in the PPI networks. In particular, node centrality, which quantifies the degree of connectivity of proteins in these networks, provides useful information to identify essential proteins. There are many ways to quantify the centrality values. NetEPD implements a broad selection of 22 popular centrality measures which are listed and briefly described in Table 2.

Table 1 Summary of the four PPI datasets included in NetEPD.

Dataset source	Number of proteins	Number of PPIs
BioGRID	63 885	852 865
DIP	26 600	72 823
MIPS	940	1151
MINT	32 668	93 742

Previous studies have shown that some biological information is related to more reliable PPI networks construction[51–55]. For instance, Wang et al.[54] proposed a model-based scheme for integrating gene expression and subcellular localization information to construct spatial and temporal active PPI networks. NetEPD also constructs spatiotemporal characteristic subnetworks[54,56] through three kinds of data, and then uses different centrality methods in subnetworks to predict essential proteins. It can not only reduce the noises or invalid neighbor nodes, which are adjacent to the protein nodes in the raw datasets, but also enhance the prediction precision of essential proteins.

An example of how NetEPD analyzes the yeast PPI network with the help of the centrality measures is shown in Fig. 2. This example demonstrates how the Degree Centrality (DC) measure can be used to rank input proteins, where this ranking can be used to annotate putative essential proteins. These results can be downloaded as a parsable excel file.

2.3 Support for assessment of predictions of essential proteins

NetEPD can be used to predict essential proteins using the information extracted from PPI networks (centrality measures), gene expression levels, and subcellular locations. These predictions can be compared against the preloaded list of essential proteins. NetEPD facilitates this comparison by implementing a variety of evaluation measures including sensitivity (SN), specificity (SP), Negative Predictive Value (NPV), Positive Predictive Value (PPV), accuracy (ACC), and F-measure (F).

$$\mathrm{SN} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{1}$$

$$\mathrm{SP} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{2}$$

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} \tag{3}$$

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4}$$

$$F = \frac{2\mathrm{SN} \cdot \mathrm{PPV}}{\mathrm{SN} + \mathrm{PPV}} \tag{5}$$

Table 2 List and description of centrality measures included in NetEPD.

Method	Short description	Reference
Degree Centrality (DC)	The degree of centrality of a node refers to the number of neighbor nodes directly connected to it.	[12]
Network Centrality (NC)	The network centrality of a node is the sum of the aggregation coefficients of all neighboring edges of the node.	[13]
Betweenness Centrality (BC)	The betweenness centrality of a node is the proportion of the shortest path through the node in all the shortest paths of the network.	[34]
Closeness Centrality (CC)	The closeness centrality of the node is inversely proportional to the sum of the shortest path from the node to all other nodes in the protein network.	[35]
Eigenvector Centrality (EC)	The eigenvector centrality of a node refers to the corresponding component of the main eigenvector of the network's adjacency matrix.	[35]
Semi-Local Centrality (SLC)	The semi-local centrality involves the fourth-order neighbor information of the node.	[36]
Local Average Connectivity (LAC)	The local average connectivity of a node indicates the public node relationship of the node and its neighbours.	[36]
Lindex	The value of a node's lindex is the largest integer of the neighbours that have at least k degrees.	[37]
Eccentricity (ECC)	The eccentricity value of a node is defined as the maximum of its shortest distance from other nodes in the network.	[38]
Neighborhood Connectivity (NeiC)	The neighborhood connectivity of a node is defined as the average connectivity of all neighboring nodes of the node.	[39]
Mapping Entropy Centrality (MEC)	This method is analogous to the concept of "information entropy" and is mainly defined by the degree centrality.	[40]
Localized Bridging Centrality (LBC)	The localized bridging centrality of a node is defined as the product of its own median centrality and the bridging coefficient.	[41]
Local Clustering Coefficient based on Degree Centrality (LCCDC)	The local clustering coefficient based on degree centrality is defined as the product of a node's degree centrality and the local clustering coefficient.	[42]
Subgraph Centrality (SC)	The subgraph centrality of a node refers to the total number of closed loops that the node participates in.	[43]
Weighted Index Centrality (WIC)	It is a method based on the weighted index of virtual nodes to evaluate the influence of node propagation in complex networks.	[44]
Maximum Neighborhood Component (MNC)	The MNC is defined as the maximum neighborhood component of a subgraph that consists of a node's neighborhoods.	[45]
Density Maximum Neighborhood Component (DMNC)	To better judge the criticality of nodes in biological networks, DMNC concept was proposed based on the MNC.	[45]
PageRank	The PageRank algorithm sorts nodes based on the link structure of the network.	[46]
LeaderRank	The LeaderRank algorithm was proposed by adding a "ground node" and the bidirectional edges with other nodes in the network.	[47]
K-shell decomposition (K-shell)	The k-shell decomposition method determines the influence of the nodes according to the position of the nodes in the network.	[48]
Mixture Degree Decomposition (MDD)	In the degree of mixture decomposition method, all nodes in the network are divided into different shells depending on their own residual degree and exhaustion degree node.	[49]
Information Centrality (IC)	The information centrality of a node essentially measures the average length of the harmonics of all paths with nodes as endpoints.	[50]

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\text{The number of proteins}} \tag{6}$$

where True Positive (TP) is the number of correctly predicted essential proteins, True Negative (TN) is the number of correctly predicted nonessential proteins, False Positive (FP) is the number of nonessential proteins incorrectly predicted as essential proteins, and False Negative (FN) is the number of essential proteins mispredicted as the nonessential proteins.

Moreover, NetEPD provides a jackknife line diagram to holistically compare the effectiveness of different predictions. First, the predictions are sorted in descending order for each prediction method, then the cumulative number of known essential proteins is determined by counting and the jackknife line is drawn. The jackknife line is a relationship between

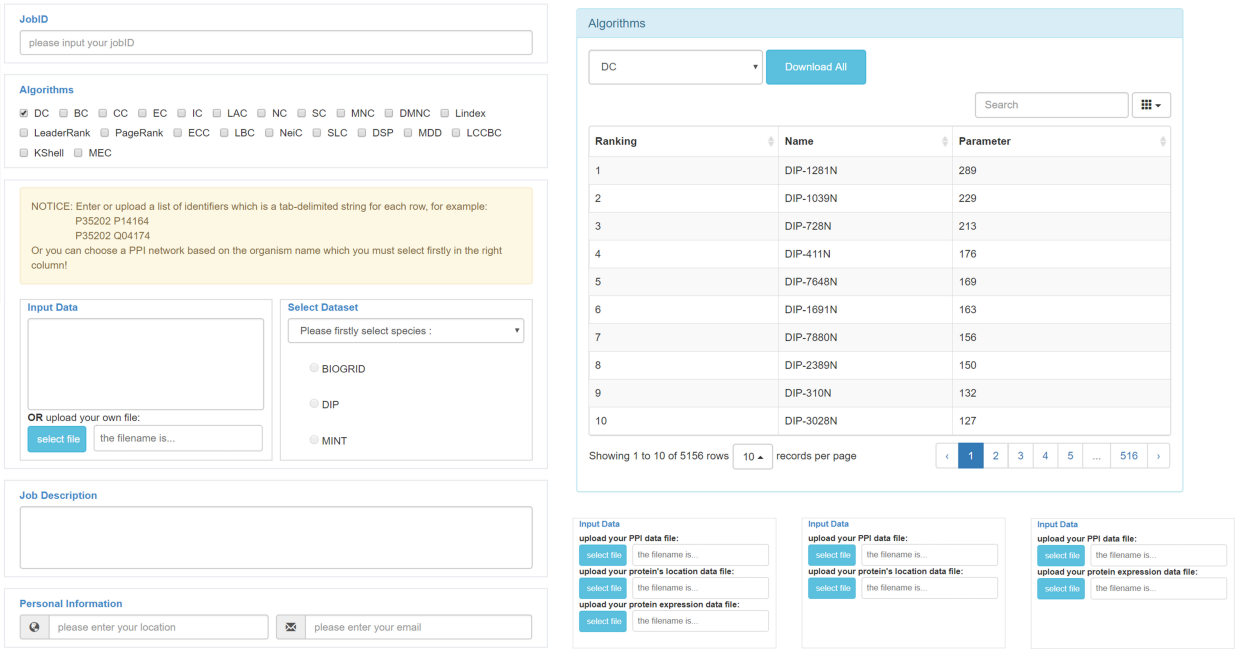


Fig. 2 An example of the use of the NetEPD platform. The panel on the left shows a screen that creates a new job. Input data can be either pasted into the "Input Data" panel or uploaded from a file. The panel on the right shows how the DC measure is used to rank input proteins.

the number of predicted essential proteins (x-axis) and the cumulative number of native essential proteins that match the putative essential proteins (y-axis).

Figure 3 shows an example of how NetEPD can be used to perform an assessment. Users can upload their list of essential proteins or use the preloaded list that was collected from the DEG resource. Next, the user must set up a cutoff to select putative essential proteins from a ranked list of proteins, where this ranking reflects the putative propensity for essential proteins. In Fig. 3, the top 10% proteins of the sorted proteins (515 proteins) are selected as the putative essential proteins and the proteins are sorted using three centrality measures: DC, LAC, and NC. After clicking "evaluate", NetEPD generates "Evaluation" panel for these three predictions. This panel includes a table with the six evaluation measures and the jackknife line diagram (Fig. 3). This diagram directly and conveniently compares effects of using multiple rankings on the quality of prediction of essential proteins.

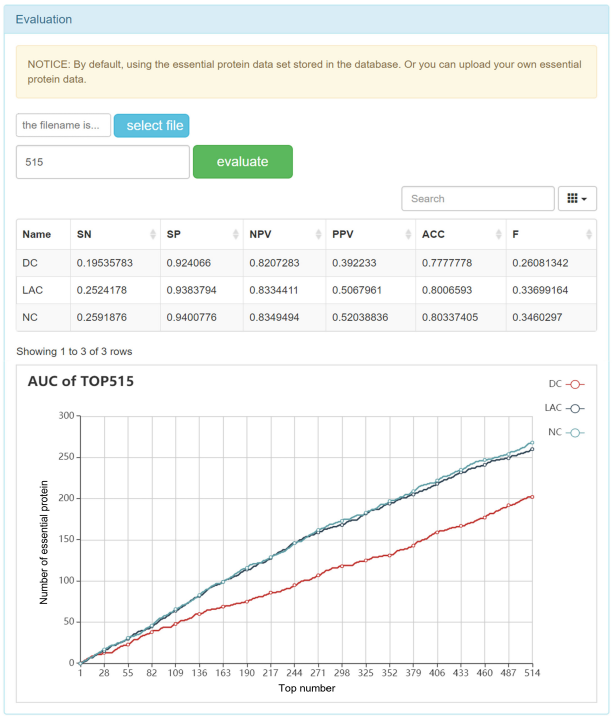


Fig. 3 Evaluation of predictions of essential proteins based on the DC, LAC, and NC centrality measures. The table in the top panel provides values of the six evaluation measures including SN, SP, NPV, PPV, ACC, and F. The jackknife line diagram is shown in the bottom panel.

2.4 Visualization

The visualization of the underlying PPI network provides useful clues to understand the topological properties of selected (essential) proteins. NetEPD uses the Cytoscape.js graphical library[57] to visualize PPI networks. Four different types of visual representations that are available in NetEPD are shown in Fig. 4. They include classical view (top-right corner), grid (bottom-right corner), concentric (middle bottom), and breadth-first layout (bottom left). NetEPD also provides an

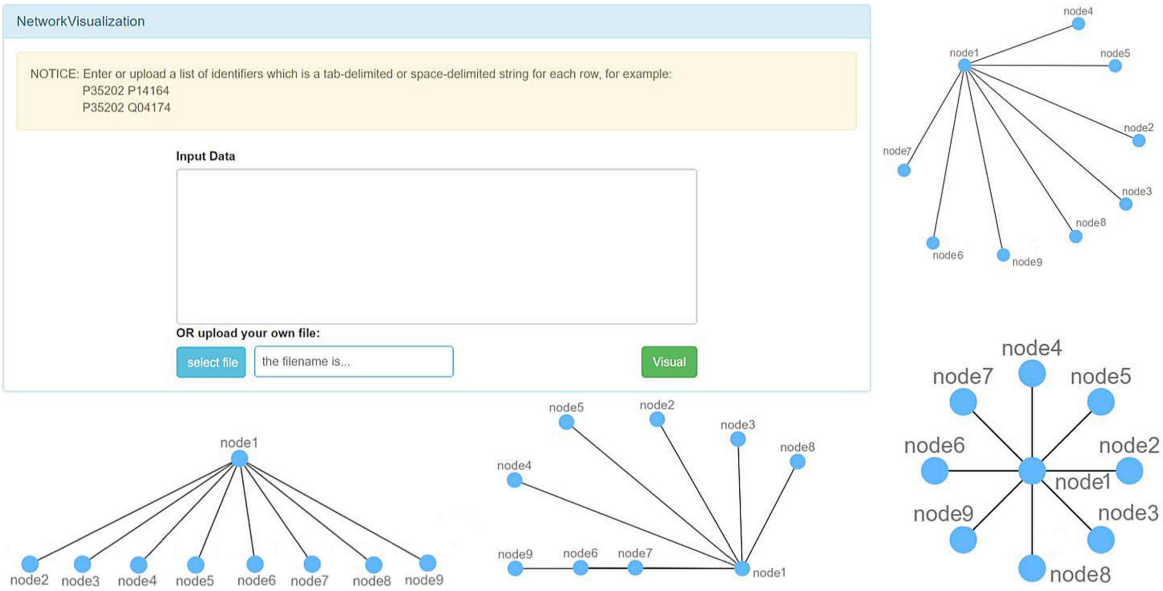


Fig. 4 Visualization of the PPI networks in NetEPD.

image export functionality that saves the resulting visual representation in png.

3 Workflow and Comparison with Similar Tools

3.1 Workflow

The NetEPD's workflow to characterize and predict essential proteins includes four steps:

Step one. Create and set up a new job using the interface shown in Fig. 2. This requires completing of the following five sub-steps: (1) Enter self-defined identifier in the "JobID" field. (2) Select the centrality measures to be used for the prediction of the essential proteins in the "Algorithms" panel. (3) Upload or select the preloaded PPI network, gene expression data, and subcellular localization data. (4) Provide a brief description of your job using the "Job Description" panel. (5) Provide your location and email, which are used for notification when the submitted job is completed. NetEPD offers four types of centrality calculations: PPI network centrality, spatial network centrality, temporal network centrality, and spatiotemporal network centrality. Users select a particular type of calculation by providing the corresponding data: PPI network centrality is computed when only the PPI data are available; spatial network centrality is calculated when the PPI and subcellular localization data are provided; temporal network centrality is used when the PPI and gene expression data are available; and spatiotemporal network centrality is computed when all three sources of data are present. The preloaded datasets (BioGRID, DIP, MIPS, and MINT) have been preprocessed to map proteins and remove redundant data. Selection of these datasets requires the user to also select the corresponding organism.

Step two. Submit the job by clicking the "submit" button. An incorrect setup of the job triggers an error message that explains how to fix the problem. A correctly set up job is redirected into a first-come-first-serve queue of jobs. Upon reaching the top of the queue, the job is executed and results are produced. The user is notified by email when the job is completed. The notification email includes a hyperlink to the webpage with the results.

Step three. Evaluate the results. Users can assess the candidate essential proteins computed in Step two with the known essential proteins, which are either provided by the user or preloaded by NetEPD. This selection can be made using the interface shown in Fig. 3. After specifying the criteria to select putative essential proteins in the field next to the "evaluate" button, the user clicks this button to generate the assessment. The assessment includes values of six performance measures (SN, SP, NPV, PPV, ACC, and F) and the jackknife line diagram is shown in the bottom panel (Fig. 3).

Step four. Visualize the results. Users can visualize the topology of the PPI network by clicking the "network" button. Four different network layouts are available (Fig. 4). Users can interact with the resulting graphs using a mouse to zoom in and out (using the mouse wheel), to select and recolor specific proteins and interactions. The resulting network graphs can be

exported in png format into the user's workspace.

3.2 Comparison with other essential protein prediction tools

There are several tools for PPI network analysis that can be used to support the prediction of essential proteins. CytoNCA[58] is a Cytoscape plugin that calculates eight centrality measures and provides a visual representation of the PPI networks. CentiServer[59] is an online resource that focuses on the computation of a comprehensive set of network centrality measures without support for other tasks such as visualization and evaluation. CentiScaPe[60] and Network Analyzer[61] are both based on Cytoscape, and they are geared toward the topological analysis of PPI networks with no support for computation and evaluation of essential protein predictions. Hubba[45] is a web-based service that finds key nodes in the molecular interaction networks, such as PPI networks, but it lacks the ability to evaluate and visualize the results. NetworkX[62] is a toolkit in the Python language that focuses on the characterization and visualization of network topologies.

Table 3 compares these six tools with NetEPD based on five characteristics: the number of included topological measures, available online, inclusion of preloaded data, evaluation of predictions, and visualization of PPI networks. The currently available tools offer either two or three of these features. NetEPD is the only platform that offers all five features. The key advantages of NetEPD include the following:

(1) It features a web-based implementation, which eliminates the need for installation and execution on the end users' hardware.

(2) It enables the integration of PPI network, gene expression, and subcellular localization data for two organisms. The availability of multiple types of datasets, which cover temporal and spatial characteristics, allows for more accurate prediction of essential proteins.

(3) It allows for the implementation of a diverse set of six evaluation measures and provides a jackknife line diagram for convenient comparison of predictions.

(4) It features four modes to visualize the topology of PPI networks.

4 Conclusion

NetEPD is a feature-rich and user-friendly web-based tool for the characterization and prediction of essential proteins and the visualization of PPI networks. The platform combines data on the PPI networks, gene expression, and subcellular localization. NetEPD implements a variety of centrality measures that are implemented to incorporate the spatial (localization) and temporal (expression) data, which provide useful information for the accurate prediction of essential proteins. Our platform computes a broad set of evaluation indicators to quantify the comparison of the predictions of essential proteins with a preloaded (or user-defined) list of native essential proteins. NetEPD also visualizes PPI networks using several layouts to facilitate understanding of the topological features that are characteristic to the essential proteins.

Inspired by the recent successes of deep learning approaches in the analysis of protein datasets[63–67], we plan to extend the NetEPD framework with a deep learning-based module that is based on recently published predictors[67]. Additionally, there are some network construction methods from the perspective of network regulation[68–70], cellular signal transduction[71], and network control[72, 73], which can help to improve the accuracy of essential protein prediction. NetEPD now has datasets of two species (Mus musculus and Saccharomyces cerevisiae) derived from BioGRID, DIP, MIPS, and MINT. In the furture, we will add more species, such as human, into the NetEPD from the Human Protein Reference Database (HPRD)[74], STRING[75], and IntAct[76]. This will further assist users to accurately predict the essential proteins.

Table 3 Comparison of key features offered by different tools for PPI networks analysis.

Tool	Number of included topological measures	Available online	Inclusion of preloaded data	Evaluation of predictions	Visualization of PPI networks
CytoNCA	8	✗	✗	✓	✓
CentiServer	55	✓	✗	✗	✗
CentiScaPe	12	✗	✗	✗	✓
Network Analyzer	11	✗	✗	✗	✓
Hubba	6	✓	✓	✗	✗
NetworkX	10	✗	✗	✗	✓
NetEPD	22	✓	✓	✓	✓

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61832019, 61622213, and 61728211) and the 111 Project (No. B18059).

References

[1] S. J. Wodak, J. Vlasblom, A. L. Turinsky, and S. Y. Pu, Protein–protein interaction networks: The puzzling riches, *Current Opinion in Structural Biology*, vol. 23, no. 6, pp. 941–953, 2013.

[2] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, et al., A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae, *Nature*, vol. 403, no. 6770, pp. 623–627, 2000.

[3] X. Peng, J. Wang, W. Peng, F. X. Wu, and Y. Pan, Protein–protein interactions: Detection, reliability assessment and applications, *Brief. Bioinform.*, vol. 18, no. 5, pp. 798–819, 2017.

[4] A. Buntru, P. Trepte, K. Klockmeier, S. Schnoegl, and E. E. Wanker, Current approaches toward quantitative mapping of the interactome, *Front. Genet.*, vol. 7, p. 74, 2016.

[5] A. L. Barabási and Z. N. Oltvai, Network biology: Understanding the cell's functional organization, *Nature Reviews Genetics*, vol. 5, no. 2, pp. 101–113, 2004.

[6] E. A. Winzeler, D. D. Shoemaker, A. Astromoff, H. Liang, K. Anderson, B. Andre, R. Bangham, R. Benito, J. D. Boeke, H. Bussey, et al., Functional characterization of the S. cerevisiae genome by gene deletion and parallel analysis, *Science*, vol. 285, no. 5429, pp. 901–906, 1999.

[7] N. Vishveshwara, M. E. Bradley, and S. W. Liebman, Sequestration of essential proteins causes prion associated toxicity in yeast, *Mol. Microbiol.*, vol. 73, no. 6, pp. 1101–1114, 2009.

[8] N. Judson and J. J. Mekalanos, TnAraOut, a transposon-based approach to identify and characterize essential bacterial genes, *Nature Biotechnology*, vol. 18, no. 7, pp. 740–745, 2000.

[9] G. Lamichhane, M. Zignol, N. J. Blades, D. E. Geiman, A. Dougherty, J. Grosset, K. W. Broman, and W. R. Bishai, A postgenomic method for predicting essential genes at subsaturation levels of mutagenesis: Application to Mycobacterium tuberculosis, *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 12, pp. 7213–7218, 2003.

[10] C. G. Zhang, Essential functions of iron-requiring proteins in DNA replication, repair and cell cycle control, *Protein & Cell*, vol. 5, no. 10, pp. 750–760, 2014.

[11] F. H. Zhang, H. Song, M. Zeng, Y. H. Li, L. Kurgan, and M. Li, DeepFunc: A deep learning framework for accurate prediction of protein functions from protein sequences and interactions, *Proteomics*, doi: 10.1002/pmic.201900019.

[12] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai, Lethality and centrality in protein networks, *Nature*, vol. 411, no. 6833, pp. 41–42, 2001.

[13] J. X. Wang, M. Li, H. Wang, and Y. Pan, Identification of essential proteins based on edge clustering coefficient, *IEEE/ACM Transactions on Computational Biology & Bioinformatics*, vol. 9, no. 4, pp. 1070–1080, 2012.

[14] X. Y. Li, W. K. Li, M. Zeng, R. Q. Zheng, and M. Li, Network-based methods for predicting essential genes or proteins: A survey, *Briefings in Bioinformatics*, doi: 10.1093/bib/bbz017.

[15] G. S. Li, M. Li, J. X. Wang, Y. H. Li, and Y. Pan, United neighborhood closeness centrality and orthology for predicting essential proteins, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, doi: 10.1109/TCBB.2018.2889978.

[16] X. R. Liu, Z. Y. Hong, J. Liu, Y. Lin, A. Rodríguez-Patón, Q. Zou, and X. X. Zeng, Computational methods for identifying the critical nodes in biological networks, *Briefings in Bioinformatics*, doi: 10.1093/bib/bbz011.

[17] G. S. Li, M. Li, W. Peng, Y. H. Li, Y. Pan, and J. X. Wang, A novel extended Pareto optimality consensus model for predicting essential proteins, *J. Theor. Biol.*, vol. 480, pp. 141–149, 2019.

[18] X. J. Lei, J. Zhao, H. Fujita, and A. D. Zhang, Predicting essential proteins based on RNA-Seq, subcellular localization and GO annotation datasets, *Knowledge-Based Systems*, vol. 151, pp. 136–148, 2018.

[19] W. Kim, Prediction of essential proteins using topological properties in GO-pruned PPI network based on machine learning methods, *Tsinghua Science and Technology*, vol. 17, no. 6, pp. 645–658, 2012.

[20] W. Peng, J. X. Wang, Y. J. Cheng, Y. Lu, F. X. Wu, and Y. Pan, UDoNC: An algorithm for identifying essential proteins based on protein domains and protein-protein interaction networks, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 2, pp. 276–288, 2015.

[21] Y. T. Fan, X. H. Hu, X. W. Tang, Q. Ping, and W. Wu, A novel algorithm for identifying essential proteins by integrating subcellular localization, in *Proc. 2016 IEEE Int. Conf. Bioinformatics and Biomedicine*, Shenzhen, China, 2016, pp. 107–110.

[22] B. H. Zhao, J. X. Wang, X. Y. Li, and F. X. Wu, Essential protein discovery based on a combination of modularity and conservatism, *Methods*, vol. 110, pp. 54–63, 2016.

[23] M. Li, Z. B. Niu, X. P. Chen, P. Zhong, F. X. Wu, and Y. Pan, A reliable neighbor-based method for identifying essential proteins by integrating gene expressions, orthology, and subcellular localization information, *Tsinghua Science and Technology*, vol. 21, no. 6, pp. 668–677, 2016.

[24] J. W. Luo and Y. Qi, Identification of essential proteins based on a new combination of local interaction density and protein complexes, *PLoS One*, vol. 10, no. 6, p. e0131418, 2015.

[25] M. Magrane and UniProt Consortium, UniProt knowledgebase: A hub of integrated protein data, *Database*, vol. 2011, p. bar009, 2011.

[26] A. Chatr-Aryamontri, R. Oughtred, L. Boucher, J. Rust, C. Chang, N. K. Kolas, L. O'Donnell, S. Oster, C. Theesfeld,

A. Sellam, et al., The BioGRID interaction database: 2017 update, *Nucleic Acids Research*, vol. 45, no. D1, pp. D369–D379, 2017.

[27] I. Xenarios, E. Fernandez, L. Salwinski, X. J. Duan, M. J. Thompson, E. M. Marcotte, and D. Eisenberg, DIP: The database of interacting proteins: 2001 update, *Nucleic Acids Research*, vol. 29, no. 1, pp. 239–241, 2001.

[28] H. W. Mewes, D. Frishman, U. Güldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Münsterkötter, S. Rudd, and B. Weil, MIPS: A database for genomes and protein sequences, *Nucleic Acids Research*, vol. 30, no. 1, pp. 31–34, 2002.

[29] A. Chatr-Aryamontri, A. Ceol, L. M. Palazzi, G. Nardelli, M. V. Schneider, L. Castagnoli, and G. Cesareni, MINT: The molecular interaction database, *Nucleic Acids Research*, vol. 35, no. suppl. 1, pp. D572–D574, 2007.

[30] T. Barrett, D. B. Troup, S. E. Wilhite, P. Ledoux, D. Rudnev, C. Evangelista, I. F. Kim, A. Soboleva, M. Tomashevsky, and R. Edgar, NCBI GEO: Mining tens of millions of expression profiles-database and tools update, *Nucleic Acids Research*, vol. 35, no. suppl. 1, pp. D760–D765, 2007.

[31] B. P. Tu, A. Kudlicki, M. Rowicka, and S. L. McKnight, Logic of the yeast metabolic cycle: Temporal compartmentalization of cellular processes, *Science*, vol. 310, no. 5751, pp. 1152–1158, 2005.

[32] J. X. Binder, S. Pletscher-Frankild, K. Tsafou, C. Stolte, S. I. O'Donoghue, R. Schneider, and L. J. Jensen, COMPARTMENTS: Unification and visualization of protein subcellular localization evidence, *Database*, vol. 2014, p. bau012, 2014.

[33] H. Luo, Y. Lin, F. Gao, C. T. Zhang, and R. Zhang, DEG 10, an update of the database of essential genes that includes both protein-coding genes and noncoding genomic elements, *Nucleic Acids Research*, vol. 42, no. D1, pp. D574–D580, 2014.

[34] S. Narayanan, The betweenness centrality of biological networks, Master dissertation, Virginia Tech, CV, USA, 2005.

[35] P. Bonacich, Power and centrality: A family of measures, *American Journal of Sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.

[36] M. Li, J. X. Wang, X. Chen, H. Wang, and Y. Pan, A local average connectivity-based method for identifying essential proteins from the network level, *Computational Biology & Chemistry*, vol. 35, no. 3, pp. 143–150, 2011.

[37] A. Korn, A. Schubert, and A. Telcs, Lobby index in networks, *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 11, pp. 2221–2226, 2009.

[38] P. Hage and F. Harary, Eccentricity and centrality in networks, *Social Networks*, vol. 17, no. 1, pp. 57–63, 1995.

[39] S. Maslov and K. Sneppen, Specificity and stability in topology of protein networks, *Science*, vol. 296, no. 5569, pp. 910–913, 2002.

[40] T. Y. Nie, Z. Guo, K. Zhao, and Z. M. Lu, Using mapping entropy to identify node centrality in complex networks, *Physica A: Statistical Mechanics and its Applications*, vol. 453, pp. 290–297, 2016.

[41] S. Nanda and D. Kotz, Localized bridging centrality for distributed network analysis, presented at the 17th Int. Conf. Computer Communications and Networks, St. Thomas, US Virgin Islands, USA, 2008.

[42] N. Meghanathan, A computationally lightweight and localized centrality metric in lieu of betweenness centrality for complex network analysis, *Vietnam Journal of Computer Science*, vol. 4, no. 1, pp. 23–38, 2017.

[43] E. Ernesto and J. A. Rodríguez-Velázquez, Subgraph centrality in complex networks, *Phys. Rev. E*, vol. 71, no. 5, p. 056103, 2005.

[44] S. B. Yu, L. Gao, Y. F. Wang, G. Gao, C. C. Zhou, and Z. Y. Gao, Weighted H-index for identifying influential spreaders, arXiv preprint arXiv: 1710.05272, 2017.

[45] C. Y. Lin, C. H. Chin, H. H. Wu, S. H. Chen, C. W. Ho, and M. T. Ko, Hubba: Hub objects analyzer-a framework of interactome hubs identification for network biology, *Nucleic Acids Research*, vol. 36, no. suppl. 2, pp. W438–W443, 2008.

[46] S. J. Kim and S. H. Lee, An improved computation of the PageRank algorithm, in *Advances in Information Retrieval*, F. Crestani, M. Girolami, and C. J. van Rijsbergen, eds. Berlin, Germany: Springer, 2002, pp. 73–85.

[47] L. Y. Lü, Y. C. Zhang, C. H. Yeung, and T. Zhou, Leaders in social networks, the delicious case, *PLoS One*, vol. 6, no. 6, p. e21202, 2011.

[48] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, Identification of influential spreaders in complex networks, *Nature Physics*, vol. 6, no. 11, pp. 888–893, 2010.

[49] A. Zeng and C. J. Zhang, Ranking spreaders by decomposing complex networks, *Physics Letters A*, vol. 377, no. 14, pp. 1031–1035, 2013.

[50] K. Stephenson and M. Zelen, Rethinking centrality: Methods and examples, *Social Networks*, vol. 11, no. 1, pp. 1–37, 1989.

[51] Q. H. Xiao, J. X. Wang, X. Q. Peng, F. X. Wu, and Y. Pan, Identifying essential proteins from active PPI networks constructed with dynamic gene expression, *BMC Genomics*, vol. 16, no. S3, p. S1, 2015.

[52] X. Q. Peng, J. X. Wang, J. Wang, F. X. Wu, and Y. Pan, Rechecking the centrality-lethality rule in the scope of protein subcellular localization interaction networks, *PLoS One*, vol. 10, no. 6, p. e0130743, 2015.

[53] M. Li, J. Yang, F. X. Wu, Y. Pan, and J. Wang, DyNetViewer: A Cytoscape app for dynamic network construction, analysis and visualization, *Bioinformatics*, vol. 34, no. 9, pp. 1597–1599, 2018.

[54] J. X. Wang, X. Q. Peng, M. Li, and Y. Pan, Construction and application of dynamic protein interaction network based on time course gene expression data, *Proteomics*, vol. 13, no. 2, pp. 301–312, 2013.

[55] M. Li, X. M. Meng, R. Q. Zheng, F. X. Wu, Y. H. Li, Y. Pan, and J. X. Wang, Identification of protein complexes by using a spatial and temporal active protein interaction network, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, doi: 10.1109/TCBB.2017.2749571.

[56] M. Li, W. K. Li, F. X. Wu, Y. Pan, and J. X. Wang, Identifying essential proteins based on sub-network partition and prioritization by integrating subcellular localization information, *Journal of Theoretical Biology*, vol. 447, pp. 65–73, 2018.

[57] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, Cytoscape: A software environment for integrated models of biomolecular interaction networks, *Genome Research*, vol. 13, no. 11, pp. 2498–2504, 2003.

[58] Y. Tang, M. Li, J. X. Wang, Y. Pan, and F. X. Wu, CytoNCA: A Cytoscape plugin for centrality analysis and evaluation of protein interaction networks, *Biosystems*, vol. 127, pp. 67–72, 2015.

[59] M. Jalili, A. Salehzadeh-Yazdi, Y. Asgari, S. S. Arab, M. Yaghmaie, A. Ghavamzadeh, and K. Alimoghaddam, CentiServer: A comprehensive resource, web-based application and R package for centrality analysis, *PLoS One*, vol. 10, no. 11, p. e0143111, 2015.

[60] G. Scardoni, M. Petterlini, and C. Laudanna, Analyzing biological network parameters with CentiScaPe, *Bioinformatics*, vol. 25, no. 21, pp. 2857–2859, 2009.

[61] Y. Assenov, F. Ramírez, S. E. Schelhorn, T. Lengauer, and M. Albrecht, Computing topological parameters of biological networks, *Bioinformatics*, vol. 24, no. 2, pp. 282–284, 2008.

[62] A. A. Hagberg, D. A. Schult, and P. J. Swart, Exploring network structure, dynamics, and function using NetworkX, presented at the 7th Python in Science Conf., Pasadena, CA, USA, 2008.

[63] J. X. Wang, H. L. Cao, J. Z. H. Zhang, and Y. F. Qi, Computational protein design with deep learning neural networks, *Scientific Reports*, vol. 8, no. 1, p. 6349, 2018.

[64] A. S. Rifaioglu, H. Atas, M. J. Martin, R. Cetin- Atalay, V. Atalay, and T. Doğan, Recent applications of deep learning and machine intelligence on in silico drug discovery: Methods, tools and databases, *Brief. Bioinform.*, doi: 10.1093/bib/bby061.

[65] C. S. Cao, F. Liu, H. Tan, D. S. Song, W. J. Shu, W. Z. Li, Y. M. Zhou, X. C. Bo, and Z. Xie, Deep learning and its applications in biomedicine, *Genomics, Proteomics & Bioinformatics*, vol. 16, no. 1, pp. 17–32, 2018.

[66] M. Zeng, F. H. Zhang, F. X. Wu, Y. H. Li, J. X. Wang, and M. Li, Protein–protein interaction site prediction through combining local and global features with deep neural networks, *Bioinformatics*, doi: 10.1093/bioinformatics/btz699.

[67] M. Zeng, M. Li, Z. H. Fei, F. X. Wu, Y. H. Li, Y. Pan, and J. X. Wang, A deep learning framework for identifying essential proteins by integrating multiple types of biological information, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, doi: 10.1109/TCBB.2019.2897679.

[68] R. Q. Zheng, M. Li, X. Chen, F. X. Wu, Y. Pan, and J. X. Wang, BiXGBoost: A scalable, flexible boosting based method for reconstructing gene regulatory networks, *Bioinformatics*, vol. 35, no. 11, pp. 1893–1900, 2019.

[69] X. Chen, M. Li, R. Q. Zheng, S. Y. Zhao, F. X. Wu, Y. H. Li, and J. X. Wang, A novel method of gene regulatory network structure inference from gene knock-out expression data, *Tsinghua Science and Technology*, vol. 24, no. 4, pp. 446–455, 2019.

[70] R. Q. Zheng, M. Li, X. Chen, S. Y. Zhao, F. X. Wu, Y. Pan, and J. X. Wang, An ensemble method to reconstruct gene regulatory networks based on multivariate adaptive regression splines, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, doi: 10.1109/TCBB.2019.2900614.

[71] M. Li, R. Q. Zheng, Y. H. Li, F. X. Wu, and J. X. Wang, MGT-SM: A method for constructing cellular signal transduction networks, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 2, pp. 417–424, 2019.

[72] Y. X. Hu, C. H. Chen, Y. Y. Ding, X. Wen, B. B. Wang, L. Gao, and K. Tan, Optimal control nodes in disease-perturbed networks as targets for combination therapy, *Nature Communications*, vol. 10, no. 1, p. 2180, 2019.

[73] M. Li, H. Gao, J. X. Wang, and F. X. Wu, Control principles for complex biological networks, *Briefings in Bioinformatics*, doi: 10.1093/bib/bby088.

[74] T. S. Keshava Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, et al., Human protein reference database-2009 update, *Nucleic Acids Research*, vol. 37, no. suppl. 1, pp. D767–D772, 2009.

[75] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork, et al., The STRING database in 2011: Functional interaction networks of proteins, globally integrated and scored, *Nucleic Acids Research*, vol. 39, no. suppl. 1, pp. D561–D568, 2011.

[76] S. Kerrien, B. Aranda, L. Breuza, A. Bridge, F. Broackes-Carter, C. Chen, M. Duesbury, M. Dumousseau, M. Feuermann, U. Hinz, et al., The IntAct molecular interaction database in 2012, *Nucleic Acids Research*, vol. 40, no. D1, pp. D841–D846, 2012.

Jiashuai Zhang received the BSc degree from Henan University, China in 2018. He is currently a postgraduate student in bioinformatics at Central South University. His currently research interests include bioinformatics, essential proteins and single cell visualization and analysis.

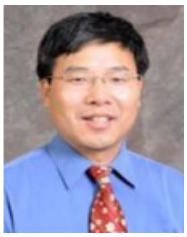
Wenkai Li received the BSc degree from Henan University in 2015, and the MS degree from Central South University in 2018. His research interests include bioinformatics, network analysis, and essential protein discovery.

Min Zeng received the BS degree from Lanzhou University in 2013 and the MS degree from Central South University in 2016. He is currently working toward the PhD degree in the School of Computer Science and Engineering, Central South University, China. His research interests include bioinformatics, machine learning, and deep learning.

Xiangmao Meng received the BSc degree and MSc degree from Jiangxi University of Science and Technology, China in 2012 and 2015, respectively. He is currently a PhD candidate in computer science at Central South University. His currently research interests include bioinformatics, complex network analysis, and data mining.

Lukasz Kurgan received the PhD degree in computer science from the University of Colorado at Boulder in 2003. He is the Qimonda endowed professor in the Department of Computer Science at the Virginia Commonwealth University. He joined the editorial board of *BMC Bioinformatics* in 2010 and currently he is the section editor for *Structural Bioinformatics*. His research interests are in structural bioinformatics of proteins and small RNAs, from single molecules through entireproteomes/genomes to projects that span thousands of proteomes/genomes. Highlights of recent research coming from his lab include release of widely used computational tools for high-throughput prediction of functional residues in protein sequences, tools that support target selection for structural genomics, and methods for functional characterization of intrinsic disorder in proteins. More details are available on the web site of his lab at http://biomine.cs.vcu.edu/.

Min Li received the PhD degree in computer science from Central South University, China in 2008. She is currently the vice dean and a professor at the School of Computer Science and Engineering, Central South University. Her research interests include computational biology, systems biology, and bioinformatics. She has published more than 80 technical papers in refereed journals such as *Bioinformatics*, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *Proteomics*, and conference proceedings such as BIBM, GIW, and ISBRA.



Fang-Xiang Wu received the BSc degree and MSc degree in applied mathematics, both from Dalian University of Technology in 1990 and 1993, respectively. He received the first PhD degree in control theory and its applications from Northwestern Polytechnical University in 1998, and the second PhD degree in biomedical engineering from University of Saskatchewan (U of S), Saskatoon, Canada in 2004. During 2004–2005, he worked as a postdoctoral fellow in the Laval University Medical Research Center (CHUL), Quebec City, Canada. He is currently a professor of the Division of Biomedical Engineering and the Department of Mechanical Engineering at the U of S. His current research interests include computational and systems biology, genomic and proteomic data analysis, biological system identification and parameter estimation, and applications of control theory to biological systems. He is serving as the editorial board member of five international journals, the guest editor of several international journals, and the program committee chair or member of several international conferences.