2017

Collaborative Data Analytics towards Prediction on Pathogen-Host Protein-Protein Interactions

Huaming Chen
University of Wollongong, hc007@uowmail.edu.au

Jun Shen
University of Wollongong, jshen@uow.edu.au

Lei Wang
University of Wollongong, leiw@uow.edu.au

Jiangning Song
Monash University, jiangning.song@monash.edu

Publication Details

Chen, H., Shen, J., Wang, L. & Song, J. (2017). Collaborative Data Analytics towards Prediction on Pathogen-Host Protein-Protein Interactions. International Conference on Computer Supported Cooperative Work in Design (pp. 269-274). United States: IEEE.

Collaborative Data Analytics towards Prediction on Pathogen-Host Protein-Protein Interactions

Abstract
Nowadays more and more data are being sequenced and accumulated in system biology, which bring the data analytics researchers to a brand new era, namely 'big data', to extract the inner relationship and knowledge from the huge amount of data.

Keywords
data, prediction, analytics, interactions, towards, collaborative, protein-protein, pathogen-host

Publication Details
Chen, H., Shen, J., Wang, L. & Song, J. (2017). Collaborative Data Analytics towards Prediction on Pathogen-Host Protein-Protein Interactions. International Conference on Computer Supported Cooperative Work in Design (pp. 269-274). United States: IEEE.

Collaborative Data Analytics towards Prediction on Pathogen-Host Protein-Protein Interactions

Huaming Chen, Jun Shen, Lei Wang
School of Computing and Information Technology
University of Wollongong
Wollongong, NSW, Australia
Email: hc007@uowmail.edu.au,{jshen,leiw}@uow.edu.au

Jiangning Song
Department of Biochemistry and Molecular Biology
Monash University
Melbourne, Victoria, Australia
Email: Jiangning.Song@monash.edu

***Abstract*—Nowadays more and more data are being sequenced and accumulated in system biology, which brings the data analytics researchers to a brand new era, namely "big data", to extract the inner relationship and knowledge from the huge amount of data. Bridging the gap between computational methodology and biology to accelerate the development of biology analytics has been a hot area. In this paper, we focus on these enormous amounts of data generated with the speedy development of high throughput technologies during the past decades, especially for protein-protein interactions, which are the critical molecular process in biology. Since pathogen-host protein-protein interactions are the major and basic problems for not only infectious diseases but also drug design, molecular level interactions between pathogen and host play very critical role for the study of infection mechanisms. In this paper, we built a basic framework for analyzing the specific problems about pathogen-host protein-protein interactions (PHPPI), meanwhile, we also presented the state-of-art deep learning method results on prediction of PHPPI comparing with other machine learning methods. Utilizing the evaluation methods, specifically by considering the high skewed imbalanced ratio and huge amount of data, we detailed the pipeline solution on both storing and learning for PHPPI. This work contributes as a basis for a further investigation of protein and protein-protein interactions, with the collaboration of data analytics results from the vast amount of data dispersedly available in biology literature.**

***Index Terms*—big data; PHPPI; bioinformatics; machine learning**

I. Introduction

Arisen in many disciplines, including computer vision, economics, online resources, bioinformatics and so on, "Big Data", which consists of data with high volume, high velocity and high variety, is impacting every areas of our life. More and more researches are conducted on data mining and machine learning for uncovering and predicting related domain knowledge. For bioinformatics area, a category of the big data related research is omics, biomedical imaging, and biomedical signal processing [1], which means big data has become a main stream in not only genome and proteomics areas [2], but also biomedical medicine and imaging areas [3]. Specifically when more and more data become available, applying computational methodology to bioinformatics is under the spotlight of the academia.

Since proteomics is a main branch in bioinformatics, a natural benefit for big data analytics in proteomics is to understand and predict the knowledge for proteins, specifically for protein-protein interactions.

Most protein-protein interactions (PPI) are defined as intra-species PPI as these two interacting proteins are from the same species. Besides these interactions, we will focus on studying the inter-species PPI between pathogen and host. The inter-species PPI refers to the interactions between two proteins from two different species. Based on the experimentally verified data and supervised learning methodologies, classifying pairs of proteins as interacting or not, has been an intense research area in the bioinformatics[4].

As infectious diseases are major worldwide health concerns, which causes millions of illnesses and deaths every year, pathogen-host protein-protein interactions is considered as the key infection process at the molecular level. In the past few years, there has been an accumulation of experimentally identified interaction data generated by using in-lab methods, including small-scale biochemical, biophysical, genetic experiments and large-scale methods like yeast-two-hybrid analysis. However these in-lab methods are highly time and resources-consuming. The issues behind these in-lab methods are the high false positive rate and the huge quantity of possible interactions.

Since pathogen-host protein-protein interactions (PHPPI) reveal lots of information about the infection mechanisms between pathogen and host, a better understanding on PHPPI and instructions on further in-lab experiments design are the main goals for utilizing the computational methodologies. In this paper, our main research contributions are as follow, while the technical contributions are also discussed in this paper:

- a basic collaborative workflow-like framework for analyzing the specific problems by: curating the big PHPPI dataset in corporation with data analytics.
- the deep learning method was first time deployed on PHPPI data set, especially on the extremely large amount of data, while also being compared with various supervised machine learning methods, including support vector machine(SVM) and extreme learning machine(ELM).

The technical contributions in this paper include the collaborative data curation, storage and the implemented machine learning method, which will be discussed later. The rest of this

paper is organized as follows: Section II reviews the related work; Section III presents the framework of PHPPI learning; Section IV discusses the PHPPI big data set curation process and gives a brief introduction of these supervised machine learning methods; Section V is a detailed results analysis and discussion. Finally in Section VI we conclude with a discussion of these results for future PHPPI research direction.

II. Related Work

Protein-protein interactions are one of the main areas in bioinformatics as it is the basics of biological functions. Many systematic biologic experiments have been conducted to verify the PPI, which gives insights inside one single species, such as yeast [5]. Amongst these interactions, PHPPI could possibly reveal the information of the infection pathways and give the researchers much more knowledge between pathogen and host.

However, for a decent PHPPI research, there is currently not a comprehensive and structured database for research purpose. [6] presented a survey which detailed the research vision on pathogen-host protein-protein interactions. [7–10] reported the state-of-art studies of host-pathogen PPIs. A biological hypothesis that "similar pathogens target the same critical biological process in the host" was utilized across the learning models for several different kinds of pathogens. The authors built a common structure aiming at using the pathway information to compute the similarity between different kinds of pathogens while the host was considered only for human. However only a pairwise level multi-task model has been built which combined two different tasks. A solution for combining more tasks in the multi-task model has been proposed but not implemented in [10]. "Task" in [7–10] means a computational model to predict interactions between a specific kind of pathogen and host.

Besides multi-task model, some other supervised machine learning methods have also been utilized to facilitate the research of PPI. In [7], the authors utilized two pathogen-human datasets as source tasks and a third one as target task to achieve the transfer learning goal. In [11, 12], the extreme learning machine (ELM) model, aiming to get a faster training speed and a higher accuracy, was deployed for a balanced PPI dataset, which is exactly an intra-species PPI dataset. Also Naïve Bayes classification method in [13] gave a comprehensive study and prediction of PPI on yeast and humans via three-dimensional structural information. The algorithm, named PrePPI, used Bayesian statistics to show its ability to be comparable with high-throughput experiments combining the structural information with other functional clues. In the end of [13], it yielded over 30,000 high-confidence interactions for yeast and over 300,000 for humans.

Since all positive PPI are experimentally verified, there is only a small quantity of PPI being manually recorded and stored in different databases, including HPIDB [14], PATRIC [15], PHISTO [16], VirHostNet [17] and VirusMentha [18]. Owing to these earlier research efforts, these databases provide well sorted and structural experimentally verified PHPPI pairs information.

Furthermore, all proteins information related to pathogen and host species, while in this paper we mainly discuss about human as the host, could be fetched via Uniprot [19]. Uniprot provides verified details for both pathogen and host. Statistics of proteins and possible interactions number are listed in next section.

By querying these data, we built a "golden standard dataset" including positive PHPPI and negative PHPPI for research based on a collaborative data analytics on similarity reduction and pairs selection. The experimentally verified PHPPI data provides the positive PHPPI however the negative PHPPI data is required under the consideration of supervised machine learning models.

Usually a balanced dataset, which ratio is nearly 1:1 between positive and negative PPI data, is built for classic model learning and further verification. For PHPPI, an imbalanced dataset with 1:100 is desired to prevent a biasing classifier towards wrong prediction. With regard to these issues, a well selected ratio is critical to build the PHPPI golden dataset. Normally researchers randomly sample the proteins of pathogen and host to curate the negative PHPPI pairs. Some proposed negative data sampling methods and selection strategies are also conducted [20].

In next section, we will extract and show the framework for PHPPI research, and conduct the experiments on several curated datasets.

III. Framework for PHPPI

As an important part of PHPPI study, a well-designed and structural process is important. Even though in [7–13, 20] there are several sections presenting a technical workflow for PPI study, currently there is not a comprehensive and detailed framework for PHPPI study considering data querying, data cleansing, feature representation and model selection. Specifically for data cleansing stage, we will give the details with collaborative data analytics later.

A. The PHPPI Framework

A framework for PHPPI normally consists of data querying, data cleansing, feature representation, and learning model selection. Shown as below Fig. 1 is a brief illustration of the framework.

B. Collaborative Data Analytics

PHPPI data are important for model learning and prediction performance. In this section, we will go through several collaborative data analytics steps, which includes data redundancy check, data storing and negative data sampling, to process data cleansing to build our dataset.

Since many database repositories across both academics and industry are open source, these experimentally verified positive PHPPI data is collected and recorded. To name a few here, HPIDB, PATRIC, PHISTO, VirHostNet and VirusMentha are several main repositories for PHPPI. As research on PPI has attracted much attention, recently Human Proteome Organization Proteomics Standards Initiative (HUPO-PSI) has also published the PSI-MI XML format to store a single, unified

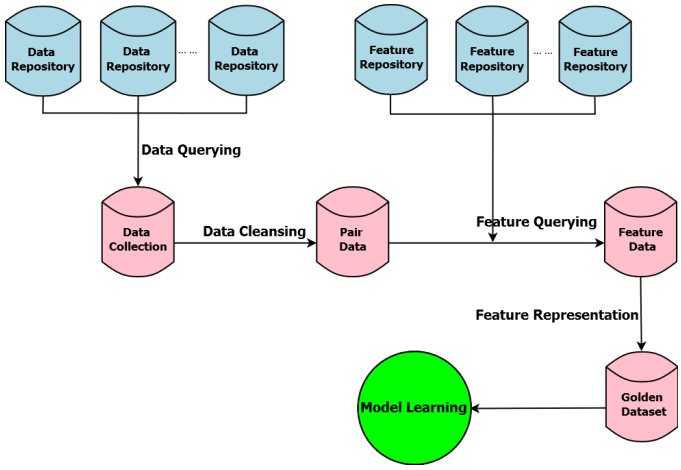

Fig. 1: A Brief Illustration of PHPPI Framework

format for PPI data. In this study, the data of PHPPI pair information is collected using XML format from several databases repositories. An exhaustive learning and prediction on PHPPI would query across several different database repositories to build a positive PHPPI dataset. To construct a negative PHPPI data set, the related proteins information is queried from Uniprot.

However, there are much data in these repositories being duplicated. Besides these directly duplicated PHPPI pairs, the homology between different proteins also needs to be considered as the PHPPI dataset contains the pairs that represent different pathogen proteins interacting with same host protein.

As [21] showed, to avoid the biasing classifier, a clustering method on these data is desired to build a dataset without much redundancy. In [21], the sequence information was utilized to obtain the clusters between proteins, which is named CD-HIT. It represents the similarity between proteins and helps us to avoid the redundancy during the data curation process.

To build a negative dataset, randomly sampling method is deployed in this paper to generate a complete dataset. A better insight for these data would be achieved after data redundancy check and negative data sampling.

After data cleansing, an extracted PHPPI pair dataset from different database repositories is obtained. However, these data only shows the ID of interacting pairs between pathogen and host. To input the information of each unique proteins into our learning model, feature selection and representation are required.

Several key properties have been studied on the proteins, including sequence information, gene ontology, human interactome graph and gene expressions. These properties could be fetched from different databases, respectively Uniprot [19] for sequence information, Gene Ontology Consortium [22] for GO information, human protein reference database (HPRD) [23] for human interactome graph and gene expression database (GEO) [24] for gene expressions.

Since sequence information includes most of the singular protein information, in this study, we only use sequence information for the feature querying and representation based on the previous research [7–13, 20].

For different properties, it is required to represent the properties into the numerical form. Thus numerous studies have been conducted on feature representation [10, 25–28] for sequence and gene ontology information. Feature representation is still a hot and ongoing research area for bioinformatics researchers.

In sequence information, its representation would bring lots of information into the learning model since protein consists of 20 kinds of amino acids in different combination and length. “Sequence specifies structure” also tells a widely adopted view that knowledge of the sequence information would be adequate to represent a protein. Based on the electrostatic and hydrophobic properties of protein, 20 kinds of amino acids are categorized into seven groups. Shown as below Fig. 2 is an illustration of these groups.

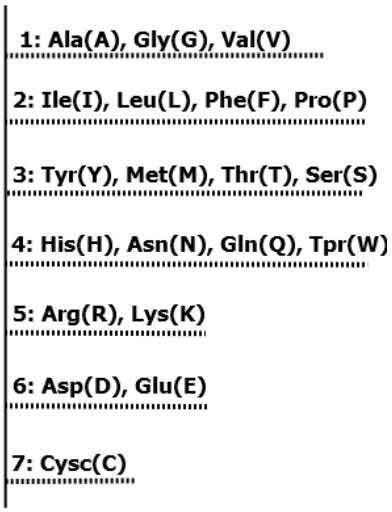

Fig. 2: Groups of Amino Acids

In this paper, Auto Covariance (AC) [26] is chosen as the representation methodology for sequence information. A brief introduction will be presented in next section for PHPPI data curation.

After a “PHPPI golden dataset” has been built, it is appropriate to deploy the supervised machine learning model to learn and predict from the dataset since all data are labelled to show interacting or not.

In this study, we applied deep learning method, which includes stacked denoising autoencoder and logistic regression layers, for a comprehensive study on the PHPPI prediction, while comparing with other methods SVM and ELM.

IV. PHPPI Dataset Curation

Following the framework shown in previous section, we first built the PHPPI dataset. The selected PHPPI database repositories are PATRIC and PHISTO. Inside these two database repositories, data is manually extracted and uploaded from related biology literatures. A statistics about bacteria species

combining PATRIC and PHISTO is shown in TABLE I. In TABLE I we only present the species that we used later in our model.

Bacteria Species	Positive Pairs Number	Clear Redundancy
Clostridium difficile	56	53
Escherichia coli	168	104
Bacillus anthracis	6073	3138

TABLE I: Statistics of PHPPI Data Set

We kept the small size datasets including 53 and 104 pairs of positive PHPPI, meanwhile, the large size dataset with 3138 pairs of positive PHPPI was remained to our later study. In this paper, we selected Clostridium difficile, Escherichia coli, and Bacillus anthracis for our further study. As shown in TABLE I, the number of positive PHPPI pairs decreases after data cleansing.

A. Feature Representation

Missing data is a mainly research problem in the Feature Querying stage. To avoid a large amount of missing data, in this paper we chose to query sequence information as the main feature.

As one feature representation algorithm using auto covariance based on sequence information, AC is popular for transforming numerical vectors to uniform matrices. It is because, even the length of different proteins would be different, the representing matrices would be the same after AC. In our paper, the length of each vector is set to 210 for each protein. In the pair-wise level, it has 420 features for each PHPPI pair.

B. PHPPI Dataset Statistics

The ratio between positive pairs and negative pairs has been discussed for a long time in the literature, in this paper, we chose ratio of 1:25, 1:50 and 1:100 to study the effect and performance of different models to yield more insights in PHPPI research.

Bacteria Species	Ratio 1:25	Ratio 1:50	Ratio 1:100
Clostridium difficile	1352	2652	5252
Escherichia coli	2548	4998	9898
Bacillus anthracis	73658	144483	286133

TABLE II: Detail Statistics of PHPPI Data Set

A brief conclusion about this PHPPI dataset is shown in TABLE II. TABLE II shows the available pairs number (including Clostridium difficile, Escherichia coli, and Bacillus anthracis) of each bacteria species.

C. Learning Model

In this paper, deep learning method was considered as our primary model since it was deemed more capable with big data sets, meanwhile several other general supervised learning models were also selected, including linear-kernel SVM and ELM.

SVM [29] is the most utilized model in many research disciplines. It aims to achieve a minimal structural risk to achieve a good performance, which has been successful in many real world applications. Basically, SVM is designed to classify by given a dataset of PHPPI denoted as $\{x_i,y_i\}$, i=1,2,...,N, where $x_i \in R^n$ and $y_i \in \{+1, -1\}$.

If we focus on high accuracy, and also consider the running time taken to train the classification model, ELM [30] would be an alternative based on the Moore-Penrose definition in this model.

By given (x_i, y_i), where $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T$ and $y_i = [y_{i1}, y_{i2}, ..., y_{im}]$, the learning procedure is presented as below with a L hidden neurons layer.

STEP 1 Fix the input weight w_i and bias b_i, i = 1,...,L
STEP 2 Calculate the hidden neurons output H
STEP 3 update β according to $\beta = H^*Y$, where H^* is the Moore-Penrose generalized inverse of the hidden neurons output and Y is the matrix of y_i

As deep learning has been getting more popular recently, its generalization in learning the relationships from data shows a promising future [1, 31]. Considering our curated big datasets, as we can see from the size and feature dimension of these five different bacteria species, we applied denoising autoencoders [32] as our unsupervised model, while at the top layer we chose logistic regression [33] as our classification model.

Denoising autoencoders is a training principle for unsupervised learning to represent the features through the deep neural network. It is motivated from autoencoders and is able to reconsutruct the input from a corrupting input. As discussed in [31], the denoising autoencoders could be stacked as stacked denoising autoencoder (SdA) to build a multi-layer network.

V. Result analyses and discussion

Since we have built up a "PHPPI golden dataset", we implemented the SdA, SVM and ELM based on "Tensorflow" [34], "libsvm" [35], "hpelm" [36] and "scikit-learn" [37]. Our system was built upon GPU "NVIDIA GTX970" and 64GB RAM, which allows us to achieve a extremely high processing speed. Our working system is Ubuntu 14.04.

To evaluate the performance and robustness of these models, the experiments were conducted with 10-fold cross-validation and the results were presented in terms of the precision, recall and F1 score. The accuracy value can not truly reflect the performance of these model since the datasets are highly skewed. Recall value represents the ratio of successful retrieval information out of the learning model. It is a critical factor to judge the system performance, specifically for an imbalanced dataset.

A basic calculation of precision and recall values are presented as:

$$Precision = TP/(TP + FP) \tag{1}$$

$$Recall = TP/(TP + FN) \tag{2}$$

in which "TP" represents true positive number, "FP" is false positive number and "FN" means false negative number.

The F1 score is calculated by:

$$F1 = 2 * Precision/(Precision + Recall) \tag{3}$$

At first we calculated the precision and recall values across these models. The statistics tables are shown in TABLE III-VIII . The ratio represents the ratio between positive samples and negative samples. "SVM" refers to linear-kernel SVM, "ELM" represents to ELM wihle "DL" is the stacked denoising autoencoders.

Bacteria Species	SVM	ELM	DL
Clostridium difficile	95.10 ± 7.84	60.61 ± 11.52	96.07 ± 6.02
Escherichia coli	63.95 ± 18.20	47.36 ± 11.21	48.54 ± 15.80
Bacillus anthracis	N/A	100 ± 0	76.12 ± 6.30

TABLE III: Precision Result on Ratio 1:25

Bacteria Species	SVM	ELM	DL
Clostridium difficile	97.14 ± 5.71	97.14 ± 5.71	95.71 ± 6.55
Escherichia coli	37.65 ± 16.26	47.18 ± 14.41	47.06 ± 16.84
Bacillus anthracis	N/A	0.91 ± 0.52	51.56 ± 5.97

TABLE IV: Recall Result on Ratio 1:25

Bacteria Species	SVM	ELM	DL
Clostridium difficile	88.81 ± 11.90	96.35 ± 7.51	88.35 ± 9.30
Escherichia coli	66.83 ± 27.80	36.12 ± 27.31	45.02 ± 13.08
Bacillus anthracis	N/A	70.00 ± 45.83	86.00 ± 8.22

TABLE V: Precision Result on Ratio 1:50

Bacteria Species	SVM	ELM	DL
Clostridium difficile	95.71 ± 6.55	97.14 ± 5.71	97.14 ± 5.71
Escherichia coli	23.53 ± 12.89	10.59 ± 9.04	38.24 ± 14.47
Bacillus anthracis	N/A	0.36 ± 0.36	36.98 ± 4.72

TABLE VI: Recall Result on Ratio 1:50

Bacteria Species	SVM	ELM	DL
Clostridium difficile	86.14 ± 13.14	96.07 ± 6.02	87.00 ± 9.22
Escherichia coli	50.0 ± 37.82	15.0 ± 32.02	50.94 ± 17.18
Bacillus anthracis	N/A	0 ± 0	93.83 ± 4.50

TABLE VII: Precision Result on Ratio 1:100

Bacteria Species	SVM	ELM	DL
Clostridium difficile	90.00 ± 11.16	78.57 ± 24.12	97.14 ± 5.71
Escherichia coli	8.24 ± 6.55	1.18 ± 2.35	38.82 ± 13.21
Bacillus anthracis	N/A	0 ± 0	23.83 ± 3.25

TABLE VIII: Recall Result on Ratio 1:100

From these tables, we could find out that, with a consideration of the ratio between positive samples and negative samples, the deep learning method achieved best performance among the larger datasets. We present the F1 score in TABLE IX-XI to get a better insight of the models performance.

From these F1 results, deep learning method, specifically the SdA model deployed, achieved the best performance for prediction of PHPPI. The symbol "N/A" denoted that the time required for training was beyond control and we dismissed it in this situation.

For Clostridium difficile, the models and the feature representation method would not affect much on the F1 value since the total sample size is small. Also the positive samples from the negative samples could be correctly classified. However, for Bacillus anthracis, which contain much more positive pairs, the SdA model achieved the best result on these three different ratio settings.

Bacteria Species	SVM	ELM	DL
Clostridium difficile	95.89 ± 5.43	73.88 ± 9.09	95.70 ± 4.76
Escherichia coli	46.38 ± 17.50	42.71 ± 10.71	47.18 ± 15.25
Bacillus anthracis	N/A	1.80 ± 1.01	61.04 ± 4.45

TABLE IX: F1 Result on Ratio 1:25

Bacteria Species	SVM	ELM	DL
Clostridium difficile	91.53 ± 6.85	96.55 ± 5.48	92.12 ± 5.37
Escherichia coli	33.96 ± 17.13	16.14 ± 13.32	40.17 ± 12.41
Bacillus anthracis	N/A	0.71 ± 0.73	51.36 ± 4.47

TABLE X: F1 Result on Ratio 1:50

Bacteria Species	SVM	ELM	DL
Clostridium difficile	87.15 ± 9.16	78.57 ± 24.12	91.48 ± 5.93
Escherichia coli	13.84 ± 10.71	2.16 ± 4.32	43.70 ± 14.38
Bacillus anthracis	N/A	0 ± 0	37.90 ± 4.17

TABLE XI: F1 Result on Ratio 1:100

VI. Conclusion

In this study, we present a comprehensive framework for PHPPI research problem, which could possibly be extended to other systems which might have the same attributes.

A well designed framework and learning model are important for PHPPI research, and would further facilitate the exploration and understanding of PHPPI, which produces a better extraction of infectious mechanisms between pathogen and host. As nowadays the data is accumulated in an extraordinary speed, a suitable learning model for PHPPI research is also highly demanded.

We also study the effect of big dataset across several different supervised learning machines. It turns out for a highly skewed and extremely big dataset, the unsupervised learning method, specifically SdA model, is better at feature representation learning. The unsupervised learning model leads to a better prediction result comparing with others. Since that, the deep learning method is more capable of dealing with big dataset while also achieves a comparable result on small dataset based on our study.

In future research, we will examine our findings on larger dataset, which contains a higher dimensional feature and samples. As for a feature fusion, the gene ontology feature, human interactome graph feature and gene expression feature will also be discussed in the future.

Acknowledgment

This work is supported by the scholarship from the China Scholarship Council (CSC), while the first author pursues his PhD degree in the University of Wollongong.

References

[1] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *arXiv preprint arXiv:1603.06430*, 2016.

[2] C. S. Greene, J. Tan, M. Ung, J. H. Moore, and C. Cheng, "Big data bioinformatics," *Journal of cellular physiology*, vol. 229, no. 12, pp. 1896–1900, 2014.

[3] N. Savage, "Bioinformatics: big data versus the big c," *Nature*, vol. 509, no. 7502, pp. S66–S67, 2014.

[4] Y. Qi, O. Tastan, J. G. Carbonell, J. Klein-Seetharaman, and J. Weston, "Semi-supervised multi-task learning for predicting interactions between hiv-1 and human proteins," *Bioinformatics*, vol. 26, no. 18, pp. i645–i652, 2010.

[5] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *Proceedings of the National Academy of Sciences*, vol. 98, no. 8, pp. 4569–4574, 2001.

[6] H. Chen, J. Shen, L. Wang, and J. Song, "Towards data analytics of pathogen-host protein-protein interaction: a survey," in *Big Data (BigData Congress), 2016 IEEE International Congress on*. IEEE, 2016, pp. 377–388.

[7] M. Kshirsagar, J. Carbonell, and J. Klein-Seetharaman, "Multisource transfer learning for host-pathogen protein interaction prediction in unlabeled tasks," *NIPS Work. Mach. Learn. Comput. Biol.*, no. 1, pp. 3–6, 2013.

[8] M. Kshirsagar, S. Schleker, J. Carbonell, and J. Klein-Seetharaman, "Techniques for transferring host-pathogen protein interactions knowledge to new tasks," *Plants as alternative hosts for human and animal pathogens*, p. 63, 2015.

[9] S. Schleker, M. Kshirsagar, and J. Klein-Seetharaman, "Comparing human–salmonella with plant–salmonella protein–protein interaction predictions," *Plants as alternative hosts for human and animal pathogens*, p. 76, 2015.

[10] M. Kshirsagar, J. Carbonell, and J. Klein-Seetharaman, "Multitask learning for host–pathogen protein interactions," *Bioinformatics*, vol. 29, no. 13, pp. i217–i226, 2013.

[11] Z.-H. You, S. Li, X. Gao, X. Luo, and Z. Ji, "Large-scale protein-protein interactions detection by integrating big biosensing data with computational model," *BioMed research international*, vol. 2014, 2014.

[12] Z.-H. You, Y.-K. Lei, L. Zhu, J. Xia, and B. Wang, "Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis," *BMC bioinformatics*, vol. 14, no. 8, p. 1, 2013.

[13] Q. C. Zhang, D. Petrey, L. Deng, L. Qiang, Y. Shi, C. A. Thu, B. Bisikirska, C. Lefebvre, D. Accili, T. Hunter *et al.*, "Structure-based prediction of protein-protein interactions on a genome-wide scale," *Nature*, vol. 490, no. 7421, pp. 556–560, 2012.

[14] R. Kumar and B. Nanduri, "Hpidb-a unified resource for host-pathogen interactions," *BMC bioinformatics*, vol. 11, no. 6, p. 1, 2010.

[15] A. R. Wattam, D. Abraham, O. Dalay, T. L. Disz, T. Driscoll, J. L. Gabbard, J. J. Gillespie, R. Gough, D. Hix, R. Kenyon *et al.*, "Patric, the bacterial bioinformatics database and analysis resource," *Nucleic acids research*, p. gkt1099, 2013.

[16] S. D. Tekir, T. Çakır, E. Ardıç, A. S. Sayılırbaş, G. Konuk, M. Konuk, H. Sarıyer, A. Uğurlu, İ. Karadeniz, A. Özgür *et al.*, "Phisto: pathogen–host interaction search tool," *Bioinformatics*, vol. 29, no. 10, pp. 1357–1358, 2013.

[17] V. Navratil, B. de Chassey, L. Meyniel, S. Delmotte, C. Gautier, P. André, V. Lotteau, and C. Rabourdin-Combe, "Virhostnet: a knowledge base for the management and the analysis of proteome-wide virus–host interaction networks," *Nucleic acids research*, vol. 37, no. suppl 1, pp. D661–D668, 2009.

[18] A. Calderone, L. Licata, and G. Cesareni, "Virusmentha: a new resource for virus-host protein interactions," *Nucleic acids research*, p. gku830, 2014.

[19] U. Consortium *et al.*, "The universal protein resource (uniprot)," *Nucleic acids research*, vol. 36, no. suppl 1, pp. D190–D195, 2008.

[20] S. Mei and H. Zhu, "A novel one-class svm based negative data sampling method for reconstructing proteome-wide htlv-human protein interaction networks," *Scientific reports*, vol. 5, p. 8034, 2015.

[21] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.

[22] G. O. Consortium *et al.*, "Gene ontology consortium: going forward," *Nucleic acids research*, vol. 43, no. D1, pp. D1049–D1056, 2015.

[23] R. Goel, H. Harsha, A. Pandey, and T. K. Prasad, "Human protein reference database and human proteinpedia as resources for phosphoproteome analysis," *Molecular bioSystems*, vol. 8, no. 2, pp. 453–463, 2012.

[24] T. Barrett, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, M. Holko *et al.*, "Ncbi geo: archive for functional genomics data setsupdate," *Nucleic acids research*, vol. 41, no. D1, pp. D991–D995, 2013.

[25] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, and H. Jiang, "Predicting protein–protein interactions based only on sequences information," *Proceedings of the National Academy of Sciences*, vol. 104, no. 11, pp. 4337–4341, 2007.

[26] Y. Guo, L. Yu, Z. Wen, and M. Li, "Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences," *Nucleic acids research*, vol. 36, no. 9, pp. 3025–3030, 2008.

[27] M. N. Davies, A. Secker, A. A. Freitas, E. Clark, J. Timmis, and D. R. Flower, "Optimizing amino acid groupings for gpcr classification," *Bioinformatics*, vol. 24, no. 18, pp. 1980–1986, 2008.

[28] Z. Du, L. Li, C.-F. Chen, S. Y. Philip, and J. Z. Wang, "G-sesame: web tools for go-term-based gene similarity analysis and knowledge discovery," *Nucleic acids research*, p. gkp463, 2009.

[29] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[30] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[32] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[33] J. M. Hilbe, *Logistic regression models*. CRC press, 2009.

[34] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous systems, 2015," *Software available from tensorflow. org*, vol. 1, 2015.

[35] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[36] A. Akusok, K.-M. Björk, Y. Miche, and A. Lendasse, "High-performance extreme learning machines: a complete toolbox for big data applications," *IEEE Access*, vol. 3, pp. 1011–1025, 2015.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.