Hindawi
BioMed Research International
Volume 2017, Article ID 1729301, 9 pages
https://doi.org/10.1155/2017/1729301

Research Article

Protein Function Prediction Using Deep Restricted Boltzmann Machines

Xianchun Zou, Guijun Wang, and Guoxian Yu

College of Computer and Information Science, Southwest University, Chongqing, China

Correspondence should be addressed to Guoxian Yu; gxyu@swu.edu.cn

Received 30 March 2017; Accepted 30 May 2017; Published 28 June 2017

Academic Editor: Peter J. Oefner

Accurately annotating biological functions of proteins is one of the key tasks in the postgenome era. Many machine learning based methods have been applied to predict functional annotations of proteins, but this task is rarely solved by deep learning techniques. Deep learning techniques recently have been successfully applied to a wide range of problems, such as video, images, and nature language processing. Inspired by these successful applications, we investigate deep restricted Boltzmann machines (DRBM), a representative deep learning technique, to predict the missing functional annotations of partially annotated proteins. Experimental results on *Homo sapiens*, *Saccharomyces cerevisiae*, *Mus musculus,* and *Drosophila* show that DRBM achieves better performance than other related methods across different evaluation metrics, and it also runs faster than these comparing methods.

1. Introduction

Proteins are the major components of living cells, they are the main material basis that form and maintain life activities. Proteins engage with various biological activities, such as catalysis of biochemical reactions and transport to signal transduction [1, 2]. High-throughput biotechniques produce explosive growth of biological data. Due to experimental techniques and the research bias in biology [3, 4], the gap between newly discovered genome sequences and functional annotations of these sequences is becoming larger and larger. The Human Proteome Project consortium recently claimed that we still have very little information about the cellular functions of approximately two-thirds of human proteins [5]. Wet-lab experiments can precisely verify functions of proteins, but it is time consuming and costly to do so. In practice, wet-lab techniques can only verify a portion of functions of proteins. In addition, it is difficult to efficiently verify functional annotations of massive proteins by wet-lab techniques. Therefore, it is important and necessary to develop computational models to make use of available functional annotations of proteins and a variety of types genomic and proteomic data, to automatically infer protein functions [2, 6].

Various computational methods have been proposed to predict functional annotations of proteins. These methods are often driven by data-intensive computational models. Data may come from amino acids sequences [7], protein-protein interactions [8], pathways [9], and multiple types of biological data fusion [10–12]. Gene Ontology (GO) is a major bioinformatics tool to unify gene products' attributes across all species, it uses GO terms to describe the gene products attributes [13], and these terms are structured in a directed acyclic graph (DAG). Each GO term in the graph can be viewed as a functional label and is associated with a distinct alphanumeric identifier, that is, GO:0008150 (biological process). GO is not static. Researchers and GO consortium contribute to updating GO as the revolved biological knowledge. Currently, most functional annotations of proteins are shallow and far from complete [3–5]. Given the true path rule of GO [13], if a protein is annotated with a GO term, then all the ancestor terms of that term are also annotated to the protein, but it is uncertain whether its descendant terms should be annotated to the protein or not. Therefore, it is more desirable to know the specific annotations of a protein, rather than the general ones, and the corresponding specific terms can provide more biological information than the shallow ones, which are ancestor terms of these specific terms. In this work,

we investigate to predict deep (or specific) annotations of a protein based on the available annotations of proteins.

Functional associations between proteins and GO structure have been directly employed to predict protein functions [14–18]. Functional annotations of proteins can be encoded by a protein function association matrix, in which each row corresponds to a protein and each column represents a type of function. King et al. [14] directly used decision tree classifier (or Bayes classifier) on the pattern of annotations to infer additional annotations of proteins. But these two classifiers need sufficient annotations and they get rather poor performance on specific GO terms, which are annotated to fewer than 10 proteins. Khatri et al. [15] used truncated single value decomposition (tSVD) to replenish the missing functions of proteins based on protein function matrix. This approach is able to predict missing annotations in existing annotation databases and improve prediction accuracy. But this method does not take advantage of the hierarchical and flat relationships between GO terms. Previous researches have demonstrated that the ontology hierarchy plays important roles in predicting protein function [2, 16, 18]. Done et al. [16] used a vector space model and a number of weighting schemes, along with latent semantic indexing approach to extract implicit semantic relationships between proteins and those between functions to predict protein functions. This method is called NtN [16]. NtN takes into account GO hierarchical structure and can weigh different GO terms situated at different locations of GO DAG [19]. Tao et al. [17] proposed a method called information theory based semantic similarity (ITSS). ITSS first calculates the semantic similarity between pairwise GO terms in a hierarchy and then sums up these pairwise similarity for pairwise GO terms annotated to two proteins. Next, it uses a kNN classifier to predict novel annotations of a protein. Yu et al. [18] proposed downward random walks (dRW) to predict missing (or new) functions of partially annotated proteins. Particularly, dRW applies downward random walks with restart [20] on the GO DAG, started on terms annotated to a protein, to predict additional annotations of the protein.

A protein is often engaged with several biological activities and thus is annotated with several GO terms. Each term can be regarded as a functional label, and protein function prediction can be modeled as a multilabel learning problem [21, 22]. From this viewpoint, protein function prediction using incomplete annotations can be modeled as a multilabel weak learning problem [22]. More recently, Yu et al. [23] proposed a method called PILL to replenish missing functions for partially annotated proteins using incomplete hierarchical labels information. Fu et al. [24] proposed a method called dHG to predict novel functions of proteins using a directed hybrid graph, which is consisted with GO DAG, protein-protein interaction network, and available functional associations between GO terms and proteins. These aforementioned methods (except DRBM) can be regarded as shallow machine learning approaches [25]. They do not capture deep associations between proteins and GO terms.

In this paper, we investigate the recently widely applied technique, deep learning [25], to capture deep associations between proteins and GO terms, and to replenish the missing annotations of incompletely annotated proteins. For this investigation, we apply deep restricted Boltzmann machines (DRBM) to predict functional annotations of proteins. DRBM utilizes the archived annotations of four model species (*Homo sapiens*, *Saccharomyces cerevisiae*, *Mus musculus,* and *Drosophila*) to explore the hidden associations between proteins and GO terms and the structural relationship between GO terms. At the same time, it optimizes the parameters of DRBM. After that, we validate the performance of DRBM by comparing its predictions with recently archived GO annotations of these four species. The empirical and comparative study shows DRBM achieves better results than other related methods. DRBM also runs faster than some of these comparing methods.

The structure of this paper is organized as follows. Section 2 briefly reviews some related deep learning techniques that are recently applied for protein function prediction. Section 3 introduces the restricted Boltzmann machine and deep restricted Boltzmann machine for protein function prediction. The experimental datasets, setup, and results are discussed in Section 4. Conclusions are provided in Section 5.

2. Related Work

Some pioneers have already applied deep learning for some bioinformatics problems [26], but few works have been reported for protein function prediction. Autoencoder neural networks (AE) can process complex structural data better than shallow machine learning methods [25, 27, 28]. AE has been applied in computer vision [28], speech recognition [25, 27], and protein residue-residue contacts prediction [26]. Chicco et al. [29] recently used deep AE to predict protein functions. Experiments show that deep AE can explore the deep associations between proteins and GO terms and achieve better performance than other shallow machine learning based function prediction methods, including tSVD [29].

Deep AE takes much more time in fine-tuning network; if the network is very deep, it will lead to vanishing gradient problem. In this work, we suggest to use deep restricted Boltzmann machines (DRBM), instead of AE, to predict functional annotations of proteins. DRBM has rapid convergence speed and good stability. DRBM has been used to construct the deep belief networks [30], for speech recognition [31, 32], collaborative filtering [33], computational biology [34], and other fields. Recently, Wang and Zeng [34] proposed to predict drug-target interactions using restricted Boltzmann machines and achieved good prediction performance. More recently, Li et al. [35] used conditional restricted Boltzmann machines to capture high-order label dependence relationships and facilitate multilabel learning with incomplete labels. Experiments have demonstrated the efficacy of restricted Boltzmann machines on addressing multilabel learning with incomplete labels.

To the best of our knowledge, few teams investigate DRBM for large-scale missing functions prediction. For this

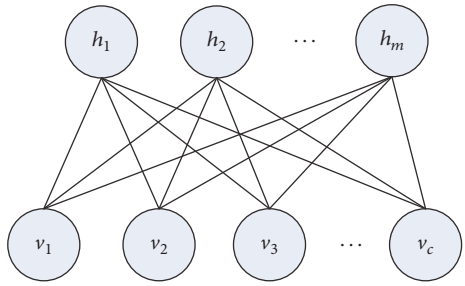

Figure 1: An RBM with binary hidden units (h_j) representing latent features and visible units (v_i) encoding observed data.

purpose, we study it for predicting functions of proteins of *Homo sapiens*, *Saccharomyces cerevisiae*, *Mus musculus*, and *Drosophila* and compare it with a number of related methods. The experimental results show that DRBM achieves better results than these comparing methods on various evaluation metrics.

3. Methods

In this section, we will describe the deep restricted Boltzmann machines to predict missing GO annotations of proteins.

3.1. Restricted Boltzmann Machine. A restricted Boltzmann machine (RBM) is a network of undirected graphical model with stochastic binary units [32]. As shown in Figure 1, an RBM is a two-layer bipartite graph with two types of units, a set of visible units $v \in \{0, 1\}$, and a set of hidden units $h \in \{0, 1\}$. Input units and hidden units are fully connected; there is no connection between nodes in the same layer. In this paper, the number of visible units is equal to the number of GO terms, and these units take the protein function association matrix as inputs.

RBM is an unsupervised method; it learns one layer of hidden features. When the number of hidden units is smaller than that of visual units, the hidden layer can deal with nonlinear complex dependency and structure of data, capture deep relationship from input data [30], and represent the input data more compactly. Latent feature values are represented by the hidden units and visible units encode available GO annotations of proteins. Suppose there are c (the number of GO terms) visible units and m hidden units in an RBM. v_i $(i = 1, \ldots, c)$ indicates the state of the ith visible unit, where $v_i = 1$ means the ith term is annotated to the protein and $v_i = 0$ means the ith term is not associated with the protein. Binary variable h_j $(j = 1, \ldots, m)$ indicates the state of hidden unit, and $h_j = 1$ denotes the jth hidden unit which is active. Let W_{ij} be the weight associated with the connection between v_i and h_j. (v, h) is a joint configuration of an RBM.

The energy function capturing the interaction patterns between visual layer and hidden layer can be modeled as follows:

$$E(v, h \mid \theta) = -\sum_{i=1}^{C} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{C} \sum_{j=1}^{m} v_i W_{ij} h_j, \quad (1)$$

where $\theta = \{W_{ij}, a_i, b_j\}$ are parameters of RBM, while a_i and b_j are biases for the visible and hidden variables, respectively. $W \in \mathbb{R}^{c \times m}$ encodes the weights of connection between c visual variables and m hidden variables. Then, a joint probability configuration of v and h can be defined as

$$P(v, h) = \frac{\exp(-E(v, h))}{Z}, \quad (2)$$

where Z is a normalization constant or partition function, $Z = \sum_{v,h} e^{-E(v,h)}$. The marginal distribution over visible data is

$$P(v) = \frac{1}{Z} \sum_{h} e^{-E(v,h)}. \quad (3)$$

There is no connection between visible units (or hidden units) in an RBM; the conditional distributions over the visible and hidden units are given by logistic functions as follows:

$$P(v_i = 1 \mid h) = \sigma\left(a_i + \sum_j h_j W_{ij}\right) \quad (4)$$

$$P(h_i = 1 \mid v) = \sigma\left(b_j + \sum_i v_i W_{ij}\right), \quad (5)$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is a logistics sigmoid function.

It is difficult to train an RBM with a large number of parameters. To efficiently train an RBM and to optimize the parameters, we maximize the likelihood of visible data with respect to the parameters. To achieve this goal, the derivative of log probability of the training data derived from (4) can be adopted to incrementally adjust the weights as follows:

$$\frac{\partial \log p(v)}{\partial W_{ij}} = \left\langle v_i h_j \right\rangle_{\text{data}} - \left\langle v_i h_j \right\rangle_{\text{model}}, \quad (6)$$

where $\langle \cdot \rangle$ indicates expectations under the distribution. It is very easy to learn the log-likelihood probability of training data:

$$\Delta W_{ij} = \epsilon \left(\left\langle v_i h_j \right\rangle_{\text{data}} - \left\langle v_i h_j \right\rangle_{\text{model}} \right), \quad (7)$$

where ϵ controls the learning rate. Since there are no direct connections in the hidden layer of an RBM, so we can get an unbiased sample of $\langle v_i h_j \rangle_{\text{data}}$ easily. Unfortunately, it is difficult to compute an unbiased sample of $\langle v_i h_j \rangle_{\text{model}}$, since it requires exponential time. To avoid this problem, a fast learning algorithm, called Contrastive Divergence (CD) [36], is proposed by Hinton [37]. CD sets visible variables as training data. Then the binary states of hidden units are all computed in parallel using (5). Once the states have been chosen for the hidden units, a "reconstruction" is produced by setting each v_i to 1 with a probability given by (4). In addition, weights are also adjusted in each training pass as follows:

$$\Delta W_{ij} = \epsilon \left(\left\langle v_i h_j \right\rangle_{\text{data}} - \left\langle v_i h_j \right\rangle_{\text{recon}} \right). \quad (8)$$

$\langle v_i h_j \rangle_{\text{data}}$ is the average value over all input data for each update and $\langle v_i h_j \rangle_{\text{recon}}$ is the average value over reconstruction; it is considered as a good approximation to $\langle v_i h_j \rangle_{\text{model}}$.

3.2. Deep RBM. In this paper, we will use a fully connected restricted Boltzmann machine and consider learning a multilayer RBMs (as shown in Figure 2). In the network structure, each layer captures complicated correlations between hidden layer and its beneath layer.

DRBM is adopted for several reasons [38]. Firstly, DRBM, like deep belief networks, has the potential of learning internal representations that become increasingly complex; it is regarded as a promising way to solve complex problems [30]. Second, high-level representations can be built from large volume incomplete sensory inputs and scarce labeled data and then be used to unfold the model. Finally, DRBM can well propagate the uncertainty information and hence robustly deal with ambiguous inputs. Hinton et al. [30] introduced a greedy, layer-by-layer unsupervised learning algorithm that consists of learning a stack of RBMs. After the stacked RBMs have been learned, the whole stack can be viewed as a single probabilistic model. In this paper, we use that greedy algorithm to optimize the parameters of DRBM. DRBM greedily trains a stack of more than two RBMs, and the modification only needs to be used for the first and last RBMs in the stack. Retraining consists of learning a stack of RBMs; each RBM has only one layer of feature detectors. The learned feature activation of one RBM is used as the input data to train the next RBM in the stack. After that, these RBMs are popped up (or unfolded) to create a DRBM. Through the above training, we can optimize the parameters of DRBM and then take the outputs of the network as the results of protein function prediction.

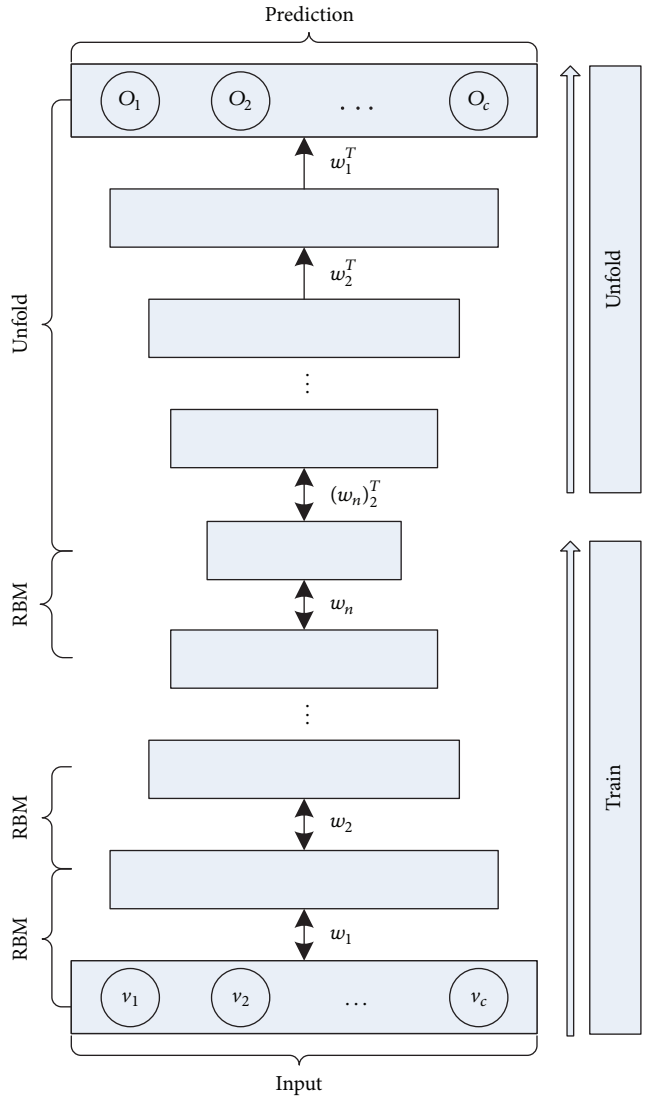

Figure 2: Network architecture of DRBM.

4. Result and Discussion

4.1. Datasets and Experimental Setup. To study the performance of DRBM on predicting missing GO annotations of incompletely annotated proteins. We downloaded the GO file (http://geneontology.org/page/download-ontology) (archived date: 2015-10-22), which describes hierarchical relationships between GO terms using a DAG. These GO terms are divided into three branches, describing molecular functions (MF), cellular component (CC), and biological process (BP) functions of proteins. We also downloaded the Gene Ontology Annotation (GOA) (archived date: 2014-10-27) files (http://geneontology.org/page/download-annotations) of *Saccharomyces cerevisiae*, *Homo sapiens*, *Mus musculus,* and *Drosophila*. We preprocessed the GO file to exclude the GO terms tagged "obsolete." To avoid circular prediction, we processed the GOA file to exclude the annotations with evidence code "IEA" (inferred from Electronic Annotation). The missing annotations of a protein often correspond to the descendants of the terms currently annotated to the protein. So the terms corresponding to these missing annotations are located at deeper level than their ancestor terms, and these terms characterize more specific biological functions of proteins than their ancestors. These specific terms are usually annotated to no more than 30 proteins; they are regarded as sparse functions. On the other hand, root terms, GO:0008150 for BP, GO:0003674 for MF, and GO:0005575 for CC, are annotated to majority of proteins; the prediction on these terms is not interesting, so we removed these three root terms. We kept the terms annotated at least one protein in the GOA file for experiments. The statistics of preprocessed GO annotations of proteins in these four model species are listed in Table 1.

We also downloaded recently archived GOA files (date: 2015-10-12) of these four species to validate the performance of DRBM and processed these GOA files in a similar way. We use the data archived in 2014 to train DRBM and then use the data archived in 2015 for validation.

In order to comparatively evaluate the performance of DRBM, we compare it with SVD [15], NtN [16], dRW [18], and AE [29]. SVD, NtN, and dRW are shallow machine learning algorithms. AE and DRBM are deep machine learning methods. DRBM is set with a learning rate of 0.01 for 25 iterations [29]. *L2* regularization is used on all weights, which are initialized randomly from the uniform distribution between 0 and 1. We set the hidden unit function as sigmoid and the number of hidden units as half of visible units and the number of the second hidden layer as half of the first

Table 1: Statistics of experimental datasets. The data in the third column (N) is the number of proteins annotated with at least 1 term for a particular subontology. C is the number of involved GO terms; Avg ± Std is the average number of annotations of a protein and its standard deviation.

Dataset		N	C	Avg ± Std
Homo sapiens	BP	11628	12514	60.24 ± 60.83
	CC	12523	1574	20.17 ± 12.28
	MF	11628	3724	10.97 ± 8.81
Mus musculus	BP	10990	13500	56.26 ± 61.08
	CC	10549	1592	15.73 ± 10.25
	MF	9906	3775	9.59 ± 7.30
Saccharomyces cerevisiae	BP	4671	4909	44.13 ± 31.41
	CC	4128	970	20.67 ± 10.30
	MF	4291	2203	9.60 ± 6.60
Drosophila	BP	6188	6645	48.53 ± 48.97
	CC	4851	1097	15.10 ± 10.27
	MF	4489	2255	9.05 ± 5.75

hidden layer and so on. The number of hidden layers is 5. In the following experiments, to prevent overfitting, we used weight-decay and dropout. Weight-decay adds an extra term to the normal gradient. This extra term is the derivative of a function that penalizes large weights. We used the simplest $L2$ penalty function. As well as that, dropout is a regularization technique for reducing overfitting in neural networks by preventing complex coadaptations on training data [39].

The accuracy of protein function prediction can be evaluated by different evaluation metrics, and the performance of different prediction models is affected by the adopted evaluation metrics. To do a fair and comprehensive comparison, we used four evaluation metrics, *MacroAvgF*1, *AvgROC*, *RankingLoss*, and *Fmax*. These evaluation metrics measure the performance of protein function prediction from different aspects. The first three metrics have been applied to evaluate the results of multilabel learning [40]. *AvgROC* and *Fmax* are recommended metrics for evaluating protein function prediction [6, 41]. *MacroAvgF*1 gets the F1-Score of each term and then takes the average of F1-score across all the terms. *AvgAUC* firstly calculates the area under receiver operating curve of each term and then takes the average value of these areas as whole to measure the performance. *Fmax* [6] is the overall maximum harmonic mean of recall and precision across all possible thresholds on the predicted protein function association matrix. *RankingLoss* computes the average fraction of wrongly predicted annotations ranking ahead of ground-truth annotations of proteins. To be consistent with other evaluation metrics, we use 1 − *RankLoss* instead of *RankingLoss*. Namely, the higher the value of these metrics is, the better the performance is. The formal definition of these metrics can be found in [6, 22, 40]. Since these metrics capture different aspects of a function prediction method, it is difficult for an approach to consistently outperform the others across all the evaluation metrics.

4.2. Experimental Results. Based on the experimental protocols introduced above, we conduct experiments to investigate the performance of DRBM on protein function prediction. In Table 2, we report the experimental results on proteins of *Homo sapiens* annotated with BP, CC, and MF terms, respectively. The results on *Mus musculus*, *Saccharomyces cerevisiae*, and *Drosophila* are provided in Tables 3–5. In these tables, the best results are in *boldface*.

From these tables, we can see that DRBM achieves better results than NtN, dRW, SVD, and AE in most cases. We further analyze the differences between DRBM and these comparing methods by Wilcoxon signed rank test [42, 43], we find that DRBM performs significantly better than NtN, dRW, and SVD on the first three metrics (where p values are all smaller than 0.004), and it also gets better performance than deep AE across these four metrics (p value smaller than 0.001). dRW often obtains larger Fmax than DRBM; the possible reason is that dRW utilizes threshold to filter out some predictions and thus increases the true positive rate.

dRW applies downward random walks with restart on the GO directed acyclic graph to predict protein function; dRW takes into count the hierarchical structure relationship between GO terms and achieves better results than NtN and SVD. This observation confirms that the hierarchical relationship between terms plays important roles in protein function prediction. Although dRW utilizes the hierarchical structure relationship between terms, it is still a shallow machine learning method and it does not capture the deep associations between proteins and GO terms as DRBM does, so it is often outperformed by DRBM.

The results of NtN and SVD are always lower than those of AE and DRBM. The possible reason is that singular value decomposition on sparse matrix is not suitable for this kind of protein function prediction problem, in which there are complex hierarchical relationships between GO terms. NtN uses the ontology hierarchy to adjust the weights of protein function associations, but it does not get better results than SVD. The reason is that NtN gives large weights to specific annotations but small weights to shallow annotations. From the true path rule, ancestor terms are generally annotated to more proteins than their descendant terms. For this reason, NtN is often outperformed by SVD and say nothing of AE

Table 2: Experimental results on *Homo sapiens*.

		MacroAvg$F1$	AvgROC	1 − RankLoss	Fmax
BP	NtN	0.0107	0.7498	0.6920	0.1712
	dRW	0.6902	0.9044	0.8737	**0.9301**
	SVD	0.7313	0.9053	0.9349	0.9206
	AE	0.5341	0.9049	0.8495	0.5617
	DRBM	**0.8378**	**0.9109**	**0.9883**	0.9217
CC	NtN	0.0036	0.6569	0.6641	0.1063
	dRW	0.6806	0.8999	0.9186	**0.9516**
	SVD	0.7139	0.8942	0.9592	0.9157
	AE	**0.8081**	0.8932	0.9629	0.8819
	DRBM	0.7982	**0.9192**	**0.9955**	0.9437
MF	NtN	0.3891	0.7767	0.8450	0.0121
	dRW	0.7909	**0.9130**	0.9208	**0.9529**
	SVD	0.8022	0.8022	0.9526	0.9480
	AE	0.7683	0.9047	0.8186	0.5604
	DRBM	**0.8517**	0.9085	**0.9898**	0.9470

Table 3: Experimental results on *Mus musculus*.

		MacroAvg$F1$	AvgROC	1 − RankLoss	Fmax
BP	NtN	0.0154	0.6950	0.7055	0.1542
	dRW	0.5666	0.8155	0.8296	**0.9049**
	SVD	0.6169	0.8220	0.9130	0.8914
	AE	0.4573	0.8139	0.8219	0.5340
	DRBM	**0.7221**	**0.8476**	**0.9841**	0.8962
CC	NtN	0.0055	0.6244	0.6436	0.1062
	dRW	0.4913	0.8001	0.7857	**0.8694**
	SVD	0.5415	0.7847	0.8856	0.8539
	AE	0.6548	0.7933	0.9139	**0.8694**
	DRBM	**0.6676**	**0.8412**	**0.9813**	0.8644
MF	NtN	0.7338	0.9135	0.9401	0.0111
	dRW	0.8742	0.9493	0.9474	**0.9693**
	SVD	0.7408	0.9466	0.9703	0.9188
	AE	0.9035	0.9461	0.9724	0.7044
	DRBM	**0.9133**	**0.9492**	**0.9906**	0.9652

Table 4: Experimental results on *Saccharomyces cerevisiae*.

		MacroAvg$F1$	AvgROC	1 − RankLoss	Fmax
BP	NtN	0.0072	0.7026	0.7027	0.1172
	dRW	0.8042	**0.9268**	0.9337	**0.9649**
	SVD	0.7794	0.9199	0.9659	0.9440
	AE	0.6990	0.9179	0.9252	0.5032
	DRBM	**0.8524**	0.9256	**0.9905**	0.9555
CC	NtN	0.0072	0.7026	0.7027	0.1172
	dRW	0.8112	0.9264	0.9612	**0.9771**
	SVD	0.7408	0.9274	0.9767	0.9198
	AE	0.8595	0.9262	0.9851	**0.9771**
	DRBM	**0.8722**	**0.9278**	**0.9948**	0.9744
MF	NtN	0.7338	0.9135	0.9401	0.0111
	dRW	0.8742	**0.9493**	0.9474	**0.9693**
	SVD	0.7408	0.9466	0.9703	0.9188
	AE	0.9035	0.9461	0.9724	0.7044
	DRBM	**0.9133**	0.9492	**0.9906**	0.9652

Table 5: Experimental results on *Drosophila*.

		MacroAvg$F1$	AvgROC	1 − RankLoss	Fmax
BP	NtN	**0.7724**	0.8450	0.8958	0.9416
	dRW	0.6875	0.8525	0.9011	**0.9455**
	SVD	0.6852	0.8516	0.9479	0.9371
	AE	0.5882	0.8486	0.9049	0.5772
	DRBM	0.7699	**0.8601**	**0.9877**	0.9382
CC	NtN	0.0101	0.6475	0.7808	0.1957
	dRW	0.6599	0.8425	0.9210	**0.9553**
	SVD	0.6446	0.8222	0.9585	0.9156
	AE	0.7331	0.8251	0.9678	**0.9553**
	DRBM	**0.7438**	**0.8558**	**0.9922**	0.9448
MF	NtN	0.5071	0.7640	0.9065	0.0700
	dRW	0.7346	**0.8206**	0.9309	**0.9610**
	SVD	0.7131	0.8125	0.9631	0.9549
	AE	0.7558	0.8133	0.9639	0.6429
	DRBM	**0.7719**	0.8187	**0.9895**	0.9499

Table 6: Runtime cost (seconds) on *Homo sapiens* and *Mus musculus* in BP subontology.

	NtN	dRW	SVD	AE	DRBM
Homo sapiens	30180	27660	1200	15840	6180
Mus musculus	24180	28020	1260	33780	7500

and DRBM. Both AE and DRBM are deep machine learning techniques, but DRBM frequently performs better than AE. That is because the generalization ability of AE is not as well as that of DRBM, and AE is easy to fall into local optimal. In summary, these results and comparisons demonstrate that DRBM can capture deep associations between proteins and GO terms, and thus it achieves better performance than other related methods across different evaluation measures. DRBM is an effective alternative approach for protein function prediction.

4.3. Runtime Analysis. Here, we study runtime (include training phase and test phase) cost of these comparing methods on *Homo sapiens* and *Mus musculus* in GO BP subontology, since this subontology includes much more annotations and GO terms. The experimental platform is Windows Server 2008, Intel Xeon E7-4820, 64 GB RAM. The recorded runtime for these comparing methods is reported in Table 6.

From this table, we can see that DRBM is faster than these comparing methods, except SVD. NtN and dRW spend a lot of time to compute semantic similarity between GO terms, so they take more time than others. In contrast, SVD directly applies matrix decomposition on the protein function association matrix and the matrix is sparse, so SVD takes fewer time than DRBM. AE employs back propagation neural networks to tune parameters; it costs a large amount of time. DRBM utilizes Contrastive Divergence, which is a fast learning algorithm, to optimize the parameters, so its runtime is fewer than AE. This comparison further confirms DRBM is an efficient and effective alternative solution for protein function prediction.

5. Conclusions

In this paper, we study how to predict additional functional annotations of annotated proteins. We investigate deep restricted Boltzmann machines (DRBM) for this purpose. Our empirical study on the proteins of *Saccharomyces cerevisiae*, *Homo sapiens*, *Mus musculus*, and *Drosophila* shows that DRBM outperforms several competitive related methods, especially shallow machine learning models. This paper will drive more research on using deep machine learning techniques for protein function prediction. As part of our future work, we will integrate other types of proteomic data with DRBM to further boost the prediction performance.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is partially supported by Natural Science Foundation of China (no. 61402378), Natural Science Foundation of CQ CSTC (nos. cstc2014jcyjA40031 and cstc2016jcyjA0351), Science and Technology Development of Jilin Province of China (20150101051JC and 20160520099JH), Science and Technology Foundation of Guizhou (Grant no. QKHJC20161076), the Science and Technology Top-Notch Talents Support Project of Colleges and Universities in Guizhou (Grant no. QJHKY2016065), and Fundamental Research Funds for the Central Universities of China (nos. XDJK2016B009 and 2362015XK07).

References

[1] R. J. Roberts, "Identifying protein functiona call for community action," *PLoS Biology*, vol. 2, no. 3, p. e42, 2004.

[2] G. Pandey, V. Kumar, and M. Steinbach, in *Computational approaches for protein function prediction: a survey*, pp. 6–28, Department of Computer Science and Engineering, University of Minnesota, A survey, 2006.

[3] A. M. Schnoes, D. C. Ream, A. W. Thorman, P. C. Babbitt, and I. Friedberg, "Biases in the experimental annotations of protein function and their effect on our understanding of protein function space," *PLoS Computational Biology*, vol. 9, no. 5, Article ID e1003063, 2013.

[4] P. D. Thomas, V. Wood, C. J. Mungall, S. E. Lewis, and J. A. Blake, "On the use of gene ontology annotations to assess functional similarity among orthologs and paralogs: a short report," *PLoS Computational Biology*, vol. 8, no. 2, Article ID e1002386, 2012.

[5] P. Legrain, R. Aebersold, A. Archakov et al., "The human proteome project: current state and future direction," *Molecular & Cellular Proteomics*, vol. 10, no. 7, article 009993, 2011.

[6] P. Radivojac, W. Clark, T. Oron et al., "A large-scale evaluation of computational protein function prediction," *Nature Methods*, vol. 10, no. 3, pp. 221–227, 2013.

[7] D. Lee, O. Redfern, and C. Orengo, "Predicting protein function from sequence and structure," *Nature Reviews Molecular Cell Biology*, vol. 8, no. 12, pp. 995–1005, 2007.

[8] R. Sharan, I. Ulitsky, and R. Shamir, "Network-based prediction of protein function," *Molecular Systems Biology*, vol. 3, p. 88, 2007.

[9] M. Cao, C. M. Pietras, X. Feng et al., "New directions for diffusion-based network prediction of protein function: Incorporating pathways with confidence," *Bioinformatics*, vol. 30, no. 12, pp. I219–I227, 2014.

[10] N. Cesa-Bianchi, M. Re, and G. Valentini, "Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference," *Machine Learning*, vol. 88, no. 1-2, pp. 209–241, 2012.

[11] G. Yu, C. Domeniconi, H. Rangwala, G. Zhang, and Z. Yu, "Transductive multi-label ensemble classification for protein function prediction," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2012*, pp. 1077–1085, chn, August 2012.

[12] G. Yu, G. Fu, J. Wang, and H. Zhu, "Predicting Protein Function via Semantic Integration of Multiple Networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 2, pp. 220–232, 2016.

[13] M. Ashburner, C. A. Ball, J. A. Blake et al., "Gene ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[14] O. D. King, R. E. Foulger, S. S. Dwight, J. V. White, and F. P. Roth, "Predicting gene function from patterns of annotation," *Genome Research*, vol. 13, no. 5, pp. 896–904, 2003.

[15] P. Khatri, B. Done, A. Rao, A. Done, and S. Draghici, "A semantic analysis of the annotations of the human genome," *Bioinformatics*, vol. 21, no. 16, pp. 3416–3421, 2005.

[16] B. Done, P. Khatri, A. Done, and S. Drăghici, "Predicting novel human gene ontology annotations using semantic analysis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 1, pp. 91–99, 2010.

[17] Y. Tao, L. Sam, J. Li, C. Friedman, and Y. A. Lussier, "Information theory applied to the sparse gene ontology annotation network to predict novel gene function," *Bioinformatics*, vol. 23, no. 13, pp. i529–i538, 2007.

[18] G. Yu, H. Zhu, C. Domeniconi, and J. Liu, "Predicting protein function via downward random walks on a gene ontology," *BMC Bioinformatics*, vol. 16, no. 1, article no. 271, 2015.

[19] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[20] H. Tong, C. Faloutsos, and J.-Y. Pan, "Random walk with restart: Fast solutions and applications," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 327–346, 2008.

[21] G. Yu, H. Rangwala, C. Domeniconi, G. Zhang, and Z. Yu, "Protein function prediction with incomplete annotations," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 3, pp. 579–591, 2013.

[22] G. Yu, C. Domeniconi, H. Rangwala, and G. Zhang, "Protein function prediction using dependence maximization," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, vol. 8188 of *Lecture Notes in Computer Science*, pp. 574–589, Springer Berlin Heidelberg.

[23] G. Yu, H. Zhu, and C. Domeniconi, "Predicting protein functions using incomplete hierarchical labels," *BMC Bioinformatics*, vol. 16, no. 1, article no. 1, 2015.

[24] G. Fu, G. Yu, J. Wang, and Z. Zhang, "Novel protein function prediction using a direct hybrid graph," *Science China-Information Science*, vol. 46, no. 4, pp. 461–475, 2016.

[25] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2013.

[26] J. Eickholt and J. Cheng, "Predicting protein residue-residue contacts using deep networks and boosting," *Bioinformatics*, vol. 28, no. 23, pp. 3066–3072, 2012.

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

[28] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *American Association for the Advancement of Science. Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[29] D. Chicco, P. Sadowski, and P. Baldi, "Deep autoencoder neural networks for gene ontology annotation predictions," in *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, ACM BCB 2014*, pp. 533–540, usa, September 2014.

[30] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[31] I. Fasel and J. Berry, "Deep belief networks for real-time extraction of tongue contours from ultrasound during speech," in *Proceedings of the 20th International Conference on Pattern Recognition, ICPR 2010*, pp. 1493–1496, tur, August 2010.

[32] A. Fischer and C. Igel, "An Introduction to Restricted Boltzmann Machines," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, vol. 7441 of *Lecture Notes in Computer Science*, pp. 14–36, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[33] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proceedings of the 24th International Conference on Machine learning (ICML '07)*, vol. 227, pp. 791–798, Corvallis, Oregon, June 2007.

[34] Y. Wang and J. Zeng, "Predicting drug-target interactions using restricted Boltzmann machines," *Bioinformatics*, vol. 29, no. 13, pp. i126–i134, 2013.

[35] X. Li, F. Zhao, and Y. Guo, "Conditional restricted boltzmann machines for multi-label learning with incomplete labels," in *Proceedings of the in Proceedings of 18th International Conference on Artificial Intelligence and Statistics*, pp. 635–643, 2015.

[36] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[37] G. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., vol. 7700 of *Lecture Notes in Computer Science*, pp. 599–619, Springer, Berlin, Germany, 2nd edition, 2012.

[38] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann Machines," in *Proceedings of the In Proceedings of 12th International Conference on Artificial Intelligence and Statistics*, pp. 448–455, 2009.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[40] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[41] Y. Jiang, "An expanded evaluation of protein function prediction methods shows an improvement in accuracy," *Genome Biology*, vol. 17, no. 1–19, pp. 1819–1837, 2016.

[42] L. Wilcoxon, "Individual comparison by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.

[43] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1–30, 2006.