UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Prediction of Isoform Functions and Interactions with ncRNAs via Deep Learning

Permalink

https://escholarship.org/uc/item/5rx8w6nf

Author

Shaw, Dipan Lal

Publication Date

2020

Copyright Information

This work is made available under the terms of a Creative Commons Attribution-ShareAlike License, availalbe at https://creativecommons.org/licenses/by-sa/4.0/

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Prediction of Isoform Functions and Interactions with ncRNAs via Deep Learning

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Dipan Lal Shaw

December 2020

Dissertation Committee:

Dr. Tao Jiang, Chairperson
Dr. Stefano Lonardi
Dr. Sika Zheng
Dr. Zhijia Zhao

Copyright by
Dipan Lal Shaw
2020

The Dissertation of Dipan Lal Shaw is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to acknowledge everyone who played a role in my academic accomplishments.

I am grateful to my advisor, Dr. Tao Jiang, for his constant guidance and continuous support in my Ph.D. research. Special thanks to my committee members, Dr. Stefano Lonardi, Dr. Sika Zheng and Dr. Zhijia Zhao, for their encouragement and insightful comments. In addition, my sincere thanks also goes to Dr. Wenbo Ma and Dr. Minzhu Xie, who provided me precious support to conduct my research.

I am thankful to my school and college teachers, Mr. Ripon Sarker, Mr. Chan Mia and Mr. Nihar Roy, who inspired me a lot in my way here.

I thank my friends and colleagues for your unwavering support.

Finally, I like to thank my parents, who supported me with love and understanding. Without you, I could never have reached this current level of success.

To my parents for all the support.

ABSTRACT OF THE DISSERTATION

Prediction of Isoform Functions and Interactions with ncRNAs via Deep Learning

by

Dipan Lal Shaw

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, December 2020
Dr. Tao Jiang, Chairperson

Alternative splicing generates multiple isoforms from a single gene. This process increases the diversity of gene functions as well as interactions with non-coding RNAs. Although gene functions and interactions have been studied extensively, little is known about isoform functions and their interactions with non-coding RNAs. In this thesis, we first study the isoform function prediction problem. We propose a novel deep learning method, DeepIsoFun, that combines multiple instance learning with domain adaptation. The latter technique helps to transfer the knowledge of gene functions to the prediction of isoform functions and provides additional labeled training data. Our model is trained on a deep neural network architecture so that it can adapt to different expression distributions associated with different gene ontology terms. Next, we approach the problem of predicting interactions between long non-coding RNAs (lncRNAs) and protein isoforms. We propose a novel method, DeepLPI, that combines heterogeneous data using a hybrid framework by integrating a deep neural network and a conditional random field. To overcome the lack of known interactions between lncRNAs and protein isoforms, we adopt a multiple instance

learning approach again. Finally, we propose a new deep learning method, RESmim, to predict interactions between microRNAs and isoforms. It adapts our previous framework but uses a residual neural network to explore various levels of the feature space. We test our methods on different organisms including human, mouse, arabidopsis, and fruit-fly. They perform significantly better than the existing methods in both cross-validation and de novo prediction experiments. The experimental results demonstrate that our methods are effective in identifying the diverse functions and interactions of isoforms.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The central dogma of molecular biology states the flow of genetic information from genes to proteins via transcription and translation. Between transcription and translation, there is a mechanism called alternative splicing. Due to alternative splicing, multiple isoforms are generated from a single gene. This process increases the diversity of isoform functions as well as interactions with non-coding RNAs. Studies in (5; 6) reveal that more than 95% of human multi-exon genes undergo alternative splicing. Though the changes in the sequences of the isoforms of the same gene are very small, they may have a systematic impact on cell functions and interactions (7). Although gene functions and interactions have been studied extensively, little is known about isoform functions and their interactions with non-coding RNAs. It has been widely reported that isoforms from the same gene sometimes have distinct or even opposite functions (8; 9; 10). The expression of different isoforms may lead to different diseases including, cardiovascular diseases, neuro-degenerative diseases, immune and infectious diseases, metabolic conditions, etc (11). Understanding isoform functions and interactions can help to diagnosis this diseases.

1.1 Isoform function prediction

The experimental determination of isoform functions is expensive and time consuming. Hence, many computational methods were developed to speed up and guide the process. However, most of these methods predict isoform functions at the gene level (12; 13; 14; 15; 16) and do not consider the fact that isoforms are the actual function carriers. In particular, the UniProt Gene Ontology (GO) database has been widely used as a standard reference for gene function annotation (17; 18). It is organized as a directed acyclic graph (DAG) where the nodes represent functional terms (referred to as GO terms) and edges indicate how a term is subdivided into more detailed functional concepts. The DAG is comprised of three main branches, *i.e.*, Biological Process (BP), Molecular Function (MF) and Cellular Component (CC) (17), representing three distinct classes of functional concepts. The functions of a gene are then represented by mapping the gene to all relevant terms in GO. In contrast, very little systematic study has been done about the specific functions of isoforms and there is no central database that provides annotated isoform functions.

In chapter 2, we propose a novel deep learning method, DeepIsoFun, that combines multiple instance learning with domain adaptation. The latter technique helps to transfer the knowledge of gene functions to the prediction of isoform functions and provides additional labeled training data. Our model is trained on a deep neural network architecture so that it can adapt to different expression distributions associated with different gene ontology terms. We evaluated the performance of DeepIsoFun on three expression datasets of human and mouse collected from SRA studies at different times. On each dataset, DeepIsoFun per-

formed significantly better than the existing methods. In terms of area under the receiver operating characteristics curve (or AUC), our method acquired at least 26% improvement and in terms of area under the precision-recall curve (or AUPRC), it acquired at least 10% improvement over the state-of-the-art methods. In addition, we also study the divergence of the functions predicted by our method for isoforms from the same gene and the overall correlation between expression similarity and the similarity of predicted functions.

1.2 Interactions between lncRNAs and protein isoforms

Long non-coding RNAs (lncRNAs) regulate diverse biological processes via their interactions with proteins. The experimental methods to identify these interactions are expensive and time-consuming, and many computational methods have been proposed to predict the interactions. Most of these computational methods are based on the known lncRNA-protein interactions to predict new interactions by using the information of sequences, secondary structures and expression profiles of lncRNAs and proteins (19; 20; 21; 22). Cross-validations showed these methods achieved good prediction performance. However, they neglected the fact that different isoforms of the same gene may interact differently with the same lncRNAs. There is still much room to improve the prediction performance.

In chapter 3, we propose a novel method, DeepLPI, to predict the interactions between lncRNAs and protein isoforms. Our method uses sequence and structure data to extract intrinsic features like functional motifs, and expression data to extract topological features of co-expression relationships. To combine these different data, we adopt a hybrid framework by integrating a deep neural network (DNN) and a conditional random field

(CRF) (23). To overcome the lack of known interactions between lncRNAs and protein isoforms, we apply a multiple instance learning (MIL) approach (24; 25; 26). Our method iteratively trains DNN and CRF to predict interactions between lncRNAs and protein isoforms as well as the interactions between lncRNAs and proteins. DeepLPI improves the prediction performance by 4.9% in terms of AUC and 8.8% in terms of AUPRC over the state-of-the-art method. To proof the efficiency of our model, we conduct analyses on the effects of model components, the impact of structural components using the saliency map, the divergence of interactions and the impact of large interaction datasets. Finally, we validate our method using a series of tests including, the correlation similarity test, the performance on mouse interactions given that the lncRNAs are conserved, and sample case studies on recently discovered lncRNA-protein interactions.

1.3 Interactions between miRNAs and isoforms

Micro RNAs (miRNAs) are small RNA transcripts of 22 nucleotides that regulate biological processes by binding to the isoforms or messenger RNAs (mRNAs) (27). They bind to complementary sequences in isoform to create cleavages or translational repression sites. As a result, the target isoforms are prevented from producing functional peptides and proteins (28).

The mechanism of miRNA-isoform interactions can be divided into two groups, canonical interactions and non-canonical interactions (29; 30). In canonical interactions the seed sequence of a miRNA, which is defined as the first 2-8 nucleotides starting at the $5'$ end and counting toward the $3'$ end, bind to the complementary sequence of the isoforms.

In non-canonical interactions miRNAs can bind outside the seed sequence (29; 30).

Finding miRNA-isoform interactions using experimental methods is expensive and time-consuming. Hence, we need to rely on computational methods to find the interactions on a large scale. In literature, a variety of miRNA target prediction tools are proposed, but their performance is far from being desirable primarily due to the lack of labeled training data. Also, most of the tools are based on assumption that the seed sequence of a miRNA is the only imporatant feature as this interact directly with isoform. These tools perform very well to find canonical interactions, but shows worst performance in finding non-canonical interaction. However, more recent studies have indicated that we should consider the entire sequence of miRNAs and isoforms to find interactions between them (30; 31).

In chapter 4, we present RESmim, a novel miRNA-isoform interaction prediction tool that works with full genomic sequence, structure and expression data of miRNAs and isoforms. To handle longer sequence of isoforms we used residual neural network, that can capture long range dependencies in a sequence. RESmim goes through two stages. In the first stage, we trained a deep neural network (DNN) that used multimodal deep learning (MDL) (23) technique to extracts features from sequence and structure data of miRNAs and isoforms. The MDL fused these extracted features and measures the initial interaction scores between miRNAs and isoforms. In the second stage, two conditional random field (CRF) are designed to exploit the co-expression relationship between miRNAs and co-expression relationship between isoforms. The two CRFs are assigned final prediction scores of interactions between miRNAs and isoforms based on initial interaction scores while trying to keep highly co-expressed miRNAs and highly co-expressed isoforms attaining the

same labels. To overcome the lack of interaction training labels of miRNAs and isoforms, we proposed an iterative semi-supervised training algorithm based on multiple instance learning (MIL) framework similar to (25; 26). In MIL, initially, we assign positive interaction labels for all miRNA and isoform pairs, given that the miRNA interact with that isoform and assign negative interaction labels for all other miRNA and isoform pairs those do not interact with each other. In each iteration, the DNN and CRFs upgrade the initial interaction scores and assign the same labels to co-expressed miRNAs and co-expressed isoforms until it reaches saturation. Under this environment, isoforms of the same protein can interact differently with the same miRNA. But since we are integrating more labeled data in terms of the miRNAs and isoforms interactions, it leads to better prediction results.

1.4 Publications

This dissertation contains material that has appeared in two publications. The DeepIsoFun (32) paper (Chapter 2) is published in the Bioinformatics journal (2019). The DeepLPI (33) paper (Chapter 3) is published in the BMC Bioinformatics journal (2020). The details of these publications are:

- D. Shaw, H. Chen, and T. Jiang. DeepIsoFun: a deep domain adaptation approach to predict isoform functions. Bioinformatics 35(15), pp. 2535-2644, 2019.
- D. Shaw, H. Chen, X. Minzhu, and T. Jiang. DeepLPI: a multimodal deep learning method for predicting the interactions between lncRNAs and protein isoforms. BMC Bioinformatics, 2020, accepted.

Chapter 2

DeepIsoFun: A deep domain adaptation approach to predict isoform functions

2.1 Introduction

In eukaryotes, the mechanism of alternative splicing produces multiple isoforms from the same gene. Studies in (5; 6) reveal that more than 95% of human multi-exon genes undergo alternative splicing. Though the changes in the sequences of the isoforms of the same gene are very small, they may have a systematic impact on cell functions and regulation (7). It has been widely reported that isoforms from the same gene sometimes have distinct or even opposite functions (8; 9; 10). For example, among the two isoforms, *l-KlCpo* and *s-KlCpo*, of gene *KlHEM13* that use different transcription start sites, only *s-KlCpo* is involved in the growth of *K. lactis* Δ*hem13* mutants (34). There is also evidence

that alternative splicing plays an important role in the evolutionary process (35). For example, the absence of exon 9 in one of the isoforms of gene *PTBP1* expressed in the brains of mammals amplifies the evolutionary difference between mammals and the other vertebrates (35). Many studies have found that alternative splicing is critical in human health and diseases. For example, to escape from cell death in tumorigenesis, gene *BCL2L1* produces two isoforms with opposite functions, where *BCL-XS* is pro-apoptosis but *BCL-XL* is anti-apoptosis (36). Similarly, gene *CASP3* has two isoforms, with *CASP3-L* being pro-apoptosis and *CASP3-S* anti-apoptosis (37). An isoform of gene *TNR6* that skips exon 6 may initiate cell death (38). Among the two isoforms of gene *PKM*, *PKM1* and *PKM2* that skip exons 9 and 10 respectively, only *PKM2* is widely expressed in cancer cells (39). Besides these examples, the results in (8; 9; 40; 41) offer more interesting stories of isoforms with dissimilar functions and hence motivate the study of specific functions of isoforms.

There is rich literature concerning the prediction of gene functions (12; 13; 14; 15; 16). In particular, the UniProt Gene Ontology (GO) database has been widely used as a standard reference for gene function annotation (17; 18). It is organized as a directed acyclic graph (DAG) where the nodes represent functional terms (referred to as GO terms) and edges indicate how a term is subdivided into more detailed functional concepts. The DAG is comprised of three main branches, *i.e.*, Biological Process (BP), Molecular Function (MF) and Cellular Component (CC) (17), representing three distinct classes of functional concepts. The functions of a gene are then represented by mapping the gene to all relevant terms in GO. In contrast, very little systematic study has been done about the specific functions of isoforms and there is no central database that provides annotated isoform

functions. Recently, several machine learning approaches were proposed to predict isoform functions from GO and RNA-Seq expression data (3; 4; 42; 43; 44). In other words, these methods attempt to distribute the annotated functions of a gene to its isoforms based on their expression profiles. Since labeled training data were generally unavailable, (3) (see also (44)), (4) and (43) solved the problem by using a semi-supervised learning technique called multiple instance learning (MIL). However, the experimental performance of their methods was quite poor. For example, on their respective datasets, the best areas under the receiver operating characteristics curve (or AUCs) achieved by the methods were only 0.681, 0.671 and 0.677, respectively. We believe that a primary cause of the poor performance was due to the lack of labeled training data.

In this chapter, we propose a novel method, DeepIsoFun, for predicting isoform functions from GO and RNA-Seq expression data. It directly addresses the challenge from the lack of labeled training data by combining MIL with the domain adaptation (DA) technique. The two techniques are somewhat complementary to each other since while MIL takes advantage of the gene-isoform relationship, DA helps to transfer the existing knowledge of gene functions to the prediction of isoform functions. More precisely, we consider each gene as a bag and each isoform as an instance in the context of MIL where the labels (*i.e.*, functions) of the instances in each bag are given as a set (45). The goal of MIL is to assign the labels (*i.e.*, functions) of each bag (*i.e.*, gene) to its instances (*i.e.*, isoforms) with the constraint that each label is assigned to at least one instance in the bag and no instance is assigned a label that does not belong to its bag (46; 47). To apply the DA technique, we take advantage of the fact that genes actually have expression data and

thus can be considered as instances in another domain (*i.e.*, the gene domain). In other words, a gene can be regarded as both an instance in the gene domain as well as a bag in the isoform domain. Since gene functions are known in GO, the DA technique can be used to transfer knowledge (*i.e.*, the relationship between expression and function) from the gene domain (called the source domain) into the isoform domain (called the target domain) (48; 49; 50; 51). Hence, the gene domain helps provide the much needed labeled training data.

The model of DeepIsoFun consists of three classifiers. The first attempts to correctly label the functions of each gene. The second attempts to correctly label the functions of each isoform (via bags). The third tries to make sure that instances from the source and target domains are indistinguishable so knowledge can be transferred. To implement the model, we use a neural network (NN) auto-encoder to extract features from expression data that are both domain-invariant and discriminative for functional prediction, inspired by the work in (52; 48). The three classifiers are also implemented as parallel NNs and connected to the auto-encoder NN to form a deep feed-forward network. The NNs involve mostly standard hidden layers and loss functions and can be trained for each GO term sequentially using a standard back-propagation algorithm based on stochastic gradient descent, but we also incorporated the gradient reversal layer to facilitate the DA method as introduced in (48) and take advantage of the hierarchical structure of GO in training. In particular, we traverse GO starting at the leaf nodes and make sure that the model is trained for all child nodes before it is trained for a parent node so the training for the parent node can benefit from earlier trainings. This also helps maintain the prediction consistency throughout GO.

To evaluate the performance of DeepIsoFun, we use three RNA-Seq expression datasets of human and mouse collected from the NCBI Reference Sequence Archive (SRA) at different times. The first is a new (also the largest) dataset that we extracted from the SRA recently. The other two were studied in (3; 4). To measure the prediction accuracy, we use both AUC and area under the precision recall curve (AUPRC) against specific baselines (measured at the gene level, as done in (3; 4; 43)). Our experimental results consist of two parts. In the first part, we analyze various properties of DeepIsoFun such as the effect of domain adaptation on its performance, impact of the frequency of a GO term in genes on its performance, difference in performance across the three main branches of GO, divergence of the functions predicted for the isoforms of a gene, and correlation between the similarity of expression profiles and the similarity of predicted functions. In the second part, we compare the performance of DeepIsoFun with the methods introduced in (3; 4; 43; 44), mi-SVM, iMILP and WLRM, based on support vector machines (SVMs), label propagation and weighted logistic regression, respectively. On our new dataset, DeepIsoFun outperformed these mi-SVM, iMILP and WLRM methods by 31%, 64% and 23% (against baseline 0.5) in AUC, respectively. In terms of AUPRC, DeepIsoFun outperformed them by 59%, 11% and 63%, respectively, against baseline 0.1. Similar improvements on the other two datasets were also observed. We believe that besides the deep learning framework, the DA technique also played an important role in these significant improvements.

The rest of the chapter is organized as follows. In the Method section, we describe the proposed method and its NN implementation in more detail. The section of Experimental evaluation shows how to determine the key parameters in the NN, the construction

of experimental datasets and all computational results on these datasets. Some possible future work is briefly outlined in the Discussion section.

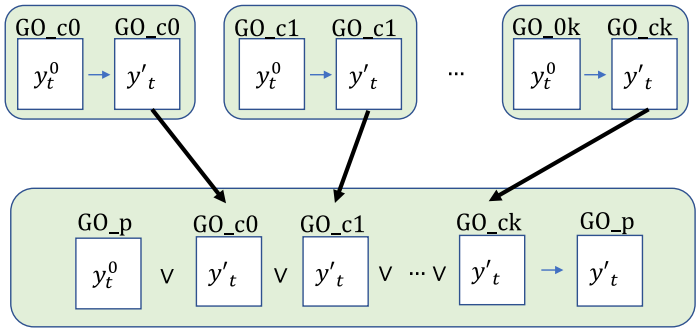


Figure 2.1: A schematic illustration of how the training for child nodes may help the training for parent nodes. The GO terms are sorted in topological order and the NN model is trained for each GO term separately in the reverse order. At a parent node p, the predicated class labels of each isoform for its children are simply added to the initial isoform class labels during its training process.

2.2 Method

In this section, we detail our proposed method, DeepIsoFun, for predicting isoform functions from GO and RNA-Seq data. As outlined above, our learning framework consists of two domains, the gene domain (denoted as $y_d = 0$) and the isoform domain (denoted as $y_d = 1$), where y_d represents a *domain class label*. In the isoform domain, the isoforms of each gene form a bag in the context of MIL. The gene domain will be considered as the source domain and the isoform domain as the target domain in the context of DA. Suppose that there are n genes and m isoforms. Hence, the isoforms are divided into n bags in the isoform domain. Suppose that the expression profiles consists of r experiments.

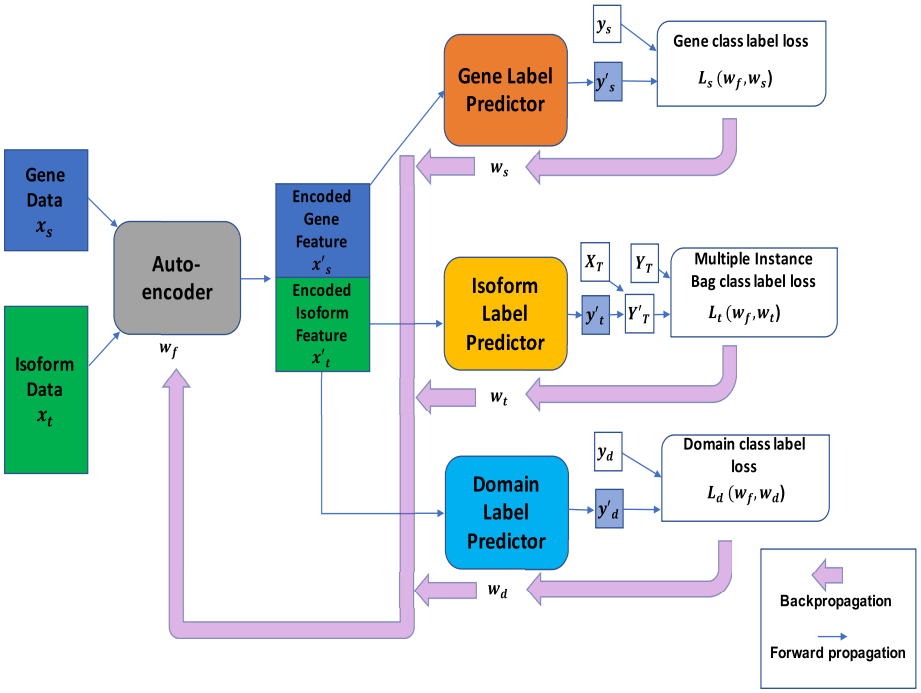


Figure 2.2: The proposed NN architecture. It includes an auto-encoder (gray), a gene class label predictor (orange), an isoform class label predictor (yellow), and a domain label predictor (blue). These four modules jointly form a feed-forward NN, where the auto-encoder consists of two hidden layers and other three components consist of one hidden layer each. The NN is trained using a standard cross-validation so that the auto-encoder extracts features from the input expression profiles to minimize the loss in gene class label prediction loss, minimize the loss in bag class label prediction and maximize the loss in domain classification so that knowledge can be transferred from the gene domain to the isoform domain. In the figure, the rectangle boxes represent the input data and extracted features. Particularly, x_s is the input gene expression data, x_t is the input isoform expression data, x'_s the encoded gene feature data, and x'_t the encoded isoform feature data. Each variable y'_s, y'_t and y'_d represents a predicted gene class label vector, a predicted isoform class label vector and a predicted domain class label vector, respectively. The notation y_s represents the true gene class label vector that is used to calculate L_s, *i.e.*, gene class label loss. The notation X_T represents the membership of isoforms in the bags and $Y_T = y_s$ is the true bag class label vector that is used to calculate L_t, *i.e.*, bag class label loss via a multiple instance loss procedure. The notation y_d is the true domain class label vector that is used for calculating L_d, *i.e.*, domain class label loss. Forward arrows represent forward propagation and backward arrows show how losses are backpropagated to allow for the adjustment of the weights w_f, w_s, w_t, and w_d used in the auto-encoder, gene class label predictor, isoform class label predictor, and domain label predictor, respectively.

Given a GO term, the data in the gene (or source) domain is denoted as a pair (x_s, y_s), where x_s is an $n \times r$ feature matrix representing the expression profiles of all n genes over the r experiments and y_s is an n-dimensional binary vector (called *gene class labels*) indicating whether each gene has the functional term or not. Similarly, the data in the isoform (or target) domain is denoted as a pair (x_t, y_t), where x_t is an $m \times r$ feature matrix representing the expression profile of all isoforms and y_t is an m-dimensional binary vector (called *isoform class labels*) indicating whether each isoform has the functional term or not. The data for each bag of the isoform domain is also denoted as a pair (X_T, Y_T), where X_T is a binary matrix representing the membership of isoforms in each bag (or gene) and Y_T is an n-dimensional binary vector (called *bag class labels*) indicating whether the isoforms in each bag collectively have the functional term or not. Observe that $Y_T = y_s$.

As mentioned above, our method combines the MIL and DA techniques and uses three classifiers to classify isoforms with respect to each GO term. It is implemented on a deep NN architecture with four modules: an auto-encoder, a gene function predictor in the gene domain, an isoform function predictor in the isoform domain, and a domain label predictor, as illustrated in Figure 2.2. The input gene and isoform expression features (x_s and x_t) are mapped by the auto-encoder to obtain an encoded feature matrix x_f. We denote the training weights used in this mapping as w_f.

Our goal for the auto-encoder is to generate new feature vectors that will reduce the loss of predicted gene class label, reduce the loss of predicted bag class label and at the same time, increase the loss of predicted domain class label. This will hopefully force the auto-encoder to generate domain-invariant features and hence realize the transfer of

knowledge from the gene domain into the isoform domain. The new (encoded) feature vectors in the matrix x_f are then partitioned into encoded gene feature vectors x'_s and encoded isoform feature vectors x'_t. Each former vector is mapped by the gene class label predictor to predict a label y'_s in the gene domain and we denote the weights in this mapping as w_s. Each latter vector is mapped by the isoform class label predictor to predict a label y'_t in the target domain and we denote the weights in this mapping as w_t. See Figure 2.2 for the detailed NN architecture.

We train the NN by following a five-fold cross-validation procedure in the isoform domain and use the annotated GO terms of genes to evaluate its performance, similar to (3; 4; 44). The data is partitioned by genes instead of isoforms to avoid potential data leak, as done in (44). Note that since isoforms from homologous genes of the human genome (*i.e.*, paralogs) do not generally share similar expression profiles (4), it is unlikely for them to cause data leak in expression-based prediction of isoforms as demonstrated in (3). All data from the gene domain is always applied to enable DA, but the single-isoform genes in the isoform domain are left out of training to avoid overfitting. Before the training is started, the variable $y_t[i]$ for isoform i is initialized as follows:

$$y_t^0[i] = \begin{cases} 1, \text{ if } x_t \in X_T[i,j] = 1 \wedge Y_T[j] = 1) \\ 0, \text{ if } x_t \in X_T[i,j] = 1 \wedge Y_T[j] = 0) \end{cases} \tag{2.1}$$

The model is then trained for each GO term separately. To take advantage of the hierarchical structure of GO, we traverse GO starting from the leaf nodes and train the model on a parent node only after all its children have been considered. This allows the training for a parent node to benefit from the knowledge learned from its children, as sketched schematically in

Figure 2.1.

The weights w_f, w_d, w_s, w_t are determined during training to minimize the following objective function:

$$\begin{aligned} L(w_f, w_s, w_t, w_d) &= \sum_{i=1,\ldots,n, y_d[i]=0} {L^i}_s(w_f, w_s) \\ +\lambda_1 \sum_{i=1,\ldots,m, y_d[i]=1} {L^i}_t(w_f, w_t) &- \lambda_2 \sum_{i=1,\ldots,n+m} {L^i}_d(w_f, w_d) \end{aligned} \tag{2.2}$$

where ${L^i}_s$ denotes the loss in gene class label prediction at the ith gene, ${L^i}_t$ the loss in bag class label prediction at the ith bag and ${L^i}_d$ the loss in domain class label prediction at the ith gene or isoform (see Figure 2.2). More precisely, for a fixed gene (or bag or isoform/gene) i, these loss functions are:

$$\begin{aligned} {L^i}_s(w_f, w_s) &= -\{y_s[i] \log y'_s[i] + (1 - y_s[i]) \log(1 - y'_s[i])\} \\ {L^i}_t(w_f, w_t) &= -\{Y_T[i] \log Y'_T[i] + (1 - Y_T[i]) \log(1 - Y'_T[i])\} \\ {L^i}_d(w_f, w_d) &= -\{y_d[i] \log y'_d[i] + (1 - y_d[i]) \log(1 - y'_d[i])\} \end{aligned} \tag{2.3}$$

We now show how the loss function $L_t[i]$ is derived. Given the predicted class labels of the isoforms in bag i, we can estimate the class label of the bag using the method proposed in (53) for dealing with multiple instance loss as shown in equation 2.4. Clearly, if at least one instance of the bag is positive, the bag will be predicted as positive; otherwise, it will be considered as negative.

$$Y'_T[i] = 1 - \prod_{j \in \text{bag } i} (1 - y'_t[j]) \tag{2.4}$$

$$y'_t[j] = \frac{1}{1 + e^{-w_t \cdot x'_t[j]}} \tag{2.5}$$

$$x'_t[j] = \frac{1}{1 + e^{-w_f(t) \cdot x_t[j]}} \tag{2.6}$$

Here, i denotes a bag and j an isoform. The predicted isoform class label $y'_t[j]$ for isoform j is calculated by the sigmoid function in equation 2.5. The encoded feature vector of isoform j, $x'_t[j]$, is calculated by another sigmoid function given in equation 2.6. The weights $w_f(t)$ represent the part of w_f derived from the isoform data. The other part of w_f, denoted as $w_f(s)$, represents the weights derived from the gene data. Similar sigmoid functions are used to derive the values of y'_s and y'_d used in equation 2.3.

As mentioned above, we would like to seek the values of w_f, w_s, w_t, w_d to achieve a saddle point of equation 2.2 such that

$$\begin{aligned} \hat{w}_f, \hat{w}_s, \hat{w}_t &= arg\ min_{w_f, w_s, w_t} L(w_f, w_s, w_t, \hat{w}_d) \\ \hat{w}_d &= arg\ max_{w_d} L(\hat{w}_f, \hat{w}_s, \hat{w}_t, w_d) \end{aligned} \tag{2.7}$$

At the saddle point, the weights w_d of the domain label predictor maximize the loss in domain classification while the weights w_s and w_t of the class label predictors minimize the loss in functional prediction in both domains. The feature mapping weights w_f help minimize the class label prediction loss while maximizing the domain classification loss. A saddle point of equation 2.7 can be found as a stationary point by using the following stochastic updates as suggested in (48):

$$w_f \leftarrow w_f - \alpha(\frac{\partial L_s}{\partial w_f(s)} + \lambda_1 \frac{\partial L_t}{\partial w_f(t)} - \lambda_2 \frac{\partial L_d}{\partial w_f})$$

$$w_s \leftarrow w_s - \alpha(\frac{\partial L_s}{\partial w_s})$$

$$w_t \leftarrow w_t - \alpha(\frac{\partial L_t}{\partial w_t})$$

$$w_d \leftarrow w_d - \alpha(\frac{\partial L_d}{\partial w_d})$$

where the parameters λ_1 and λ_2 control the relative contributions of the predictors during learning and α denotes the learning rate in this process.

2.3 Experimental evaluation

In this section, we describe in detail how to choose the key parameters in the NN model, how the test data is collected, and how the computational experiments are performed as well as what are their results.

2.3.1 The deep NN parameters

DeepIsoFun has been implemented in Caffe (54). In our NN architecture, the auto-encoder consists of two fully-connected layers to extract common features of the gene and isoform domains. The first fully-connected layer consists of 600 neurons and the second fully-connected layer consists of 200 neurons. The number of hidden layers and size of each layer (*i.e.*, number of neurons in the layer) were optimized by a standard grid search method (55; 56). The gene class label predictor and isoform class label predictor modules are both output layers, and hence have only a single output neuron each. The domain label predictor module uses a fully connected layer with 300 neurons and an output layer

with a domain output neuron. We used a standard stochastic gradient descent optimization method to minimize the training error as represented by the loss function given in equation 2.2 that involves two parameters λ_1 and λ_2. Both parameters were tuned experimentally by following suggestions in the literature (56; 57). In particular, the parameter λ_2 weighting the contribution from domain label prediction was set by using the following formula:

$$\lambda_2 = \frac{2}{1 + e^{-10p}} - 1$$

By adjusting $p \in [0, 1]$, we gradually tuned λ_2 so that noise from the domain label predictor is minimized at early training stages. The isoform domain data was partitioned in the five-fold cross-validation procedure to produce the training and test data. The batch size used in stochastic training of the NN model was 200. In other words, 200 source samples (genes) and 200 target samples (isoforms) are merged to create a batch. At the initial training stage, the leaning rate was set as $\alpha = 0.001$. As training progresses, we update the learning rate by using the standard step decay procedure (58) implemented in Caffe. We also checked if the learning was diverging (*e.g.*, very large loss values were observed), and dropped the initial learning rate by a factor 10 until convergence has been achieved.

2.3.2 Collection of datasets

Manually reviewed mRNA isoform sequences and gene sequences of human were collected from the NCBI RefSeq (59). To collect the expression profiles of these isoforms, we took an initial set of 4643 RNA-Seq experiments from the NCBI SRA database (60), and selected datasets with 50 million to 100 million reads. These experiments represented different physiological and cell conditions but were not involved in population studies. Such

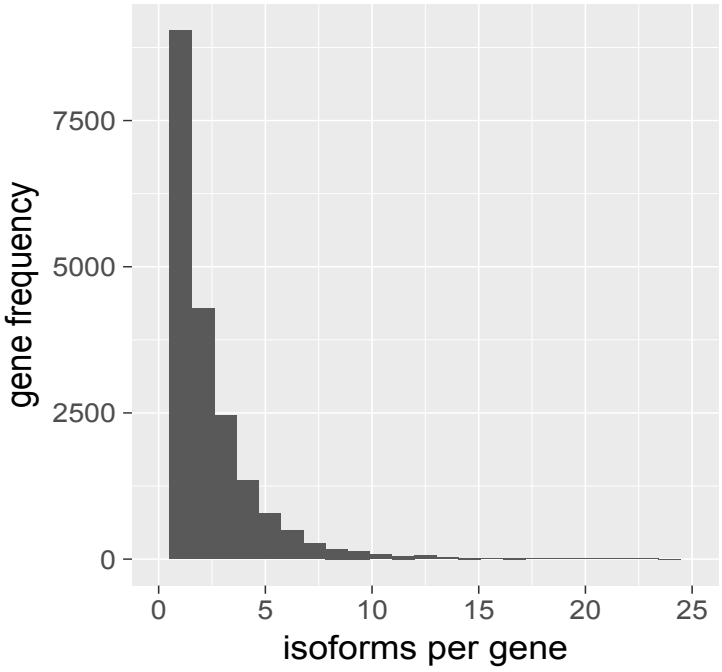


Figure 2.3: The distribution of isoforms over genes. Out of the 19532 genes in RefSeq, 9039 have only one isoform (called single isoform genes or SIGs) and 10313 have more than one isoform (called multiple isoform genes or MIGs).

a diverse set of expression data may reflect many complex characteristics of the isoforms. The tool Kallisto (2) with Sleuth (61) was used to generate isoform expression data measured in TPMs (Transcripts Per Million). The expression level of a gene in a dataset was estimated by summing up the expression levels of all its isoforms. Experiments with the pseudo-alignment ratio less than 0.7 were discarded to ensure data quality. We also filtered out poorly covered genes and their corresponding isoforms in these experiments. Finally, the expression data of 19532 genes and 47393 isoforms from 1735 RNA-Seq experiments formed our first dataset (simply called Dataset#1). Out of these genes, 9039 have only one isoform and are called single-isoform genes (SIGs) and 10313 have more than one isoform and are called multiple-isoform genes (MIGs). The distribution of isoforms over genes is shown in Figure 2.3. UniProt genes were mapped to RefSeq genes by using the UniProt ID mapping file. The UniProt GO database was used to annotate the functions of each RefSeq gene, where GO functions inferred from electronic annotation (IEA) evidence code were discarded as done in (4). In other words, only manually curated functions were used for the final annotation. The number of genes associated with a GO term is referred to as the GO term size. Intuitively, GO terms with small sizes are computationally difficult to learn since its data is highly skewed (*i.e.*, mostly negative). In particular, it was assumed in (3) that a GO term with size less than 5 might be very specific to certain genes and thus not very useful in the cross-validation training procedure. We hence did not consider such infrequent GO terms in our experiments. The basic version of GO was used to generate the parent-child relationship between GO terms (17). Out of all 44612 GO terms, 14563 appear in human annotations. After the above filtration, 4272 GO terms were kept for our experimental

evaluation work. In addition to Dataset#1, we also used the datasets with their respective GO annotations introduced in (3) and (4) (called Dataset#2 and Dataset#3, resp.) to ensure our comparison results are unbiased, where Dataset#2 was generated from 116 SRA mouse studies consisting of 365 experiments and Dataset#3 was generated from 29 SRA human studies consisting of 455 experiments with the requirement that each study had more than 6 experiments.

2.3.3 Experimental results

Since isoform functions are generally unavailable, we evaluated the performance of DeepIsoFun using gene level functional annotations by considering SIGs and MIGs either together or separately, as done similarly in (3; 4). Because each SIG contains only one isoform, its functional annotation can be used to directly validate the predicted functions of the involved isoform. For a MIG, we can only check if the set of the predicted functions of its isoforms is consistent with its annotated GO terms (3). We also estimated the functional divergence achieved by the isoforms of the same gene by calculating the semantic dissimilarity for each of the three main branches of GO (*i.e.*, CC, BP and MF). The tool GOssTo (62) was used to perform this estimation because it was able to take into account the hierarchical structure of GO. Moreover, we analyzed how the DA technique really helped the performance of our method, how the size of a GO term impacted the performance and the correlation between expression similarity and predicted function similarity for isoforms. Finally, we compared our method with the methods in (3; 4; 43; 44) in terms of AUC and AUPRC against specific baselines by focusing on a small set of GO terms (*i.e.*, GO Slim with 117 terms) that have been widely used in the literature (17). Here, a baseline rep-

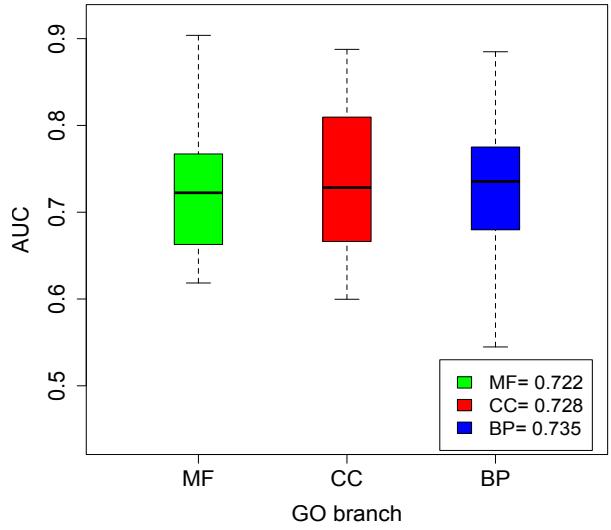
resents the performance of a random (untrained) classifier (63). While the baseline in an AUC estimation is always 0.5 (64; 65), the baseline in an AUPRC estimation depends on data imbalance and equals the proportion of positive instances (63). The latter measure is known to be more suitable for imbalanced data. Note that for highly imbalanced data (like ours), AUPRC values are often quite low (66; 63). However, we may still use them to compare the relative performance of different methods on various datasets, taking into account actual baselines.

Performance on the three main branches of GO

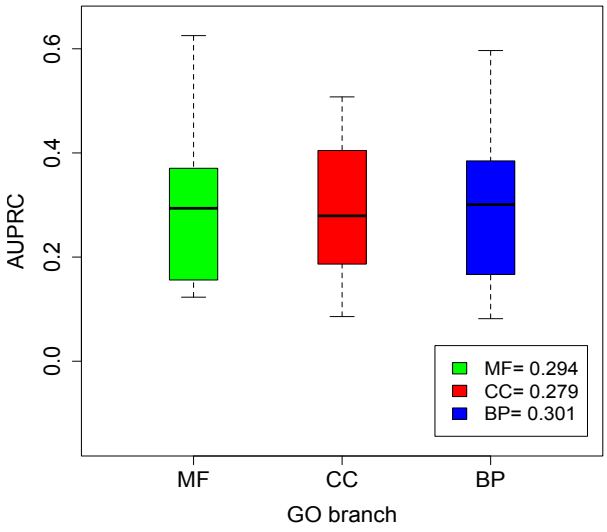
Since the three main branches carry very different meanings in gene functions and are often treated separately in the literature, we compared the performance of DeepIsoFun on them. Out of the 4272 GO terms, 699 belong to CC, 2178 BP and 1395 MF. The distributions of GO term sizes on the three branches are similar. The average AUC values on BP, CC and MF are 0.735, 0.728 and 0.722, respectively, (see Figure 2a) and the average AUPRC values are 0.301, 0.279 and 0.294, respectively, (see Figure 2b). This robust performance of DeepIsoFun on the three main branches of GO shows that the terms on the branches probably follow similar distributions (as already observed on the distributions of their sizes).

Impact of the size of a GO term on performance

Some GO terms are very specific to certain genes while the others are more general. To test how the size (or popularity) of a GO term would impact the performance of



(a)



(b)

Figure 2.4: Comparison of performance on the three main branches of GO. (a) The average AUC values on the three branches. (b) The average AUPRC values on the three branches.

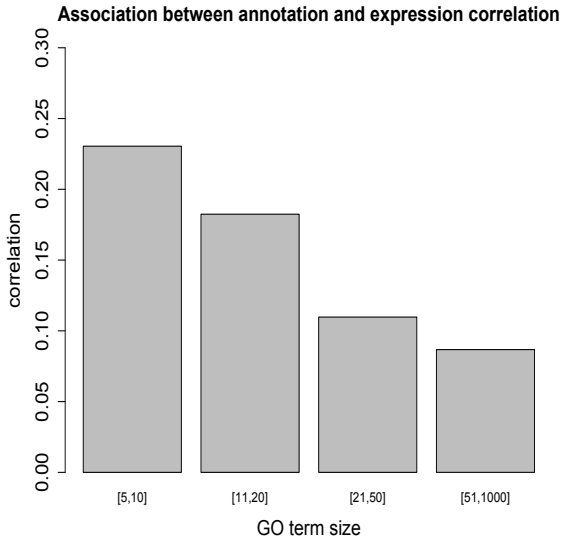


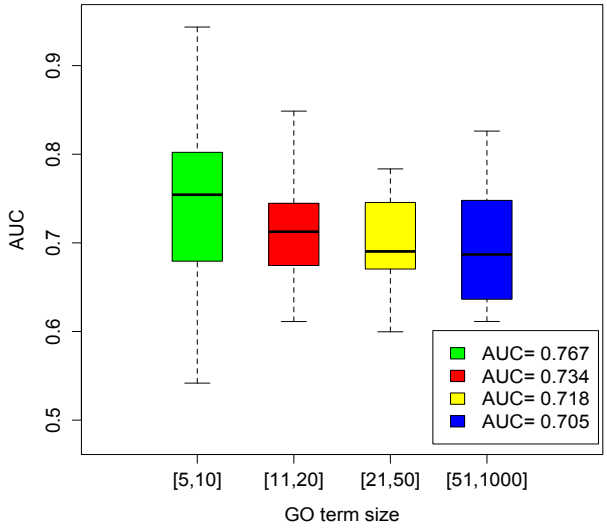
Figure 2.5: Correlation between expression similarity and functional similarity with respect to different GO term sizes. The GO terms were again divided into four groups based on sizes. For each group, all genes that have been annotated with at least one term in the group were collected. For these genes, we generated two matrices to represent their (pairwise) similarity in terms of expression profiles or GO functions. Then, using a standard tool (cor.test) in the R Stats package, we estimated the Pearson correlation between these two similarity matrices. Clearly, the stronger the correlation, the more likely we are able to predict GO functions based on expression. As shown in the histogram, the correlation decreases as the GO term size increases although the correlation is weak in all groups. This trend might indicate that there is more annotation noise in GO terms with larger sizes, which in turn might help explain why the performance of DeepIsoDun decreases with respect to GO terms with larger sizes.

DeepIsoFun, we divided the GO terms into four groups based on size. The four groups consist of GO terms of sizes in ranges [5-10], [11-20], [21-50], and [51-1000], respectively. The performance of DeepIsoFun on these groups is given in Figure 3a. DeepIsoFun performed better on GO terms with smaller sizes in general. This pattern seems to contradict intuition, but it is consistent with the findings in (4) and can perhaps be explained by the large the amount of (annotation) noise in large size GO terms. To confirm this, we further analyzed the correlation between expression similarity and functional similarity with respect to GO

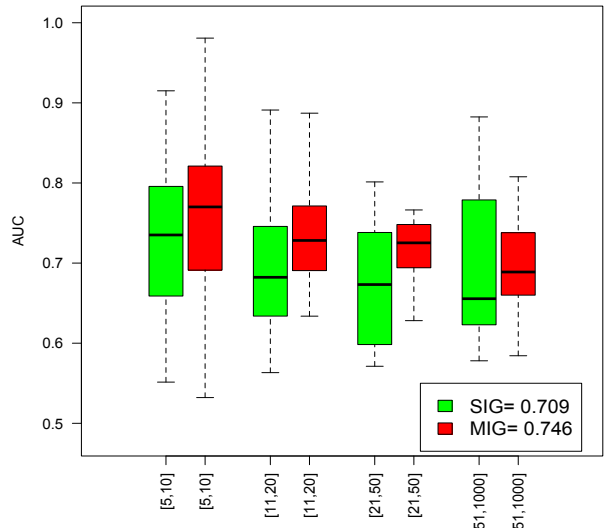
terms in each of the four groups. The results in Figure 2.5 suggest that the correlation decreases as the GO term size increases. The weak correlations shown in the figure also partially explain why the AUC and AUPRC values obtained in Figure 2 are not very high.

Performance on MIGs vs SIGs

In the previous section, we considered the performance of DeepIsoFun on all genes, including both SIGs and MIGs. Since our ultimate goal is to dissect functions of different functions of the same gene, we would like to compare the performance on MIGs with that on SIGs in this subsection. As shown in Figure 3b, the performance increases as term size decreases. Moreover, the performance on MIGs achieved in these groups are consistently better than the performance on SIGs. More precisely, the performance on MIGs was 14%, 23%, 27%, and 19% better (against the baseline 0.5) than that on SIGs in the four groups, respectively. Hence, DeepIsoFun was more effective in predicting functions for genes with multiple isoforms than genes with a single isoform, probably because of the functional diversity usually acquired by the former. Another plausible cause is that, since most (95%) human genes are expected to be MIGs, many SIGs could represent poorly annotated genes that have large numbers of undiscovered isoforms. Therefore, we also analyzed the performance of DeepIsoFun on MIGs with a certain number of isoforms. As shown in Figure 2.7, the AUC performance of DeepIsoFun increases (slightly) as more isoforms are found in a MIG.



(a)



(b)

Figure 2.6: Comparison of AUCs achieved in four groups of GO terms from CC with different sizes. The four groups contain terms with sizes in ranges [5-10], [11-20], [21-75], and [76-1000], respectively. (a) The average AUC values achieved by the terms in the four groups are 0.767, 0.734, 0.718, and 0.705, respectively. The plot shows that generally as the size of a GO term increases, its achieved AUC actually decreases. (b) DeepIsoFun consistently performed better on MIGs over SIGs. The average AUC values on MIGs achieved in the four groups are 0.79, 0.748, 0.745, and 0.709, respectively. the average AUC values on SIGs achieved by in four groups are 0.755, 0.702, 0.693, and 0.675, respectively.

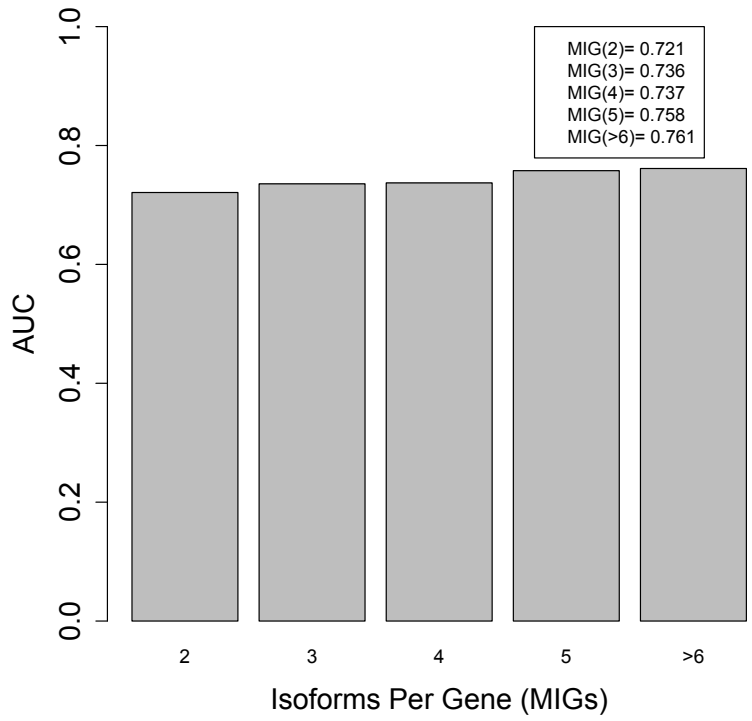
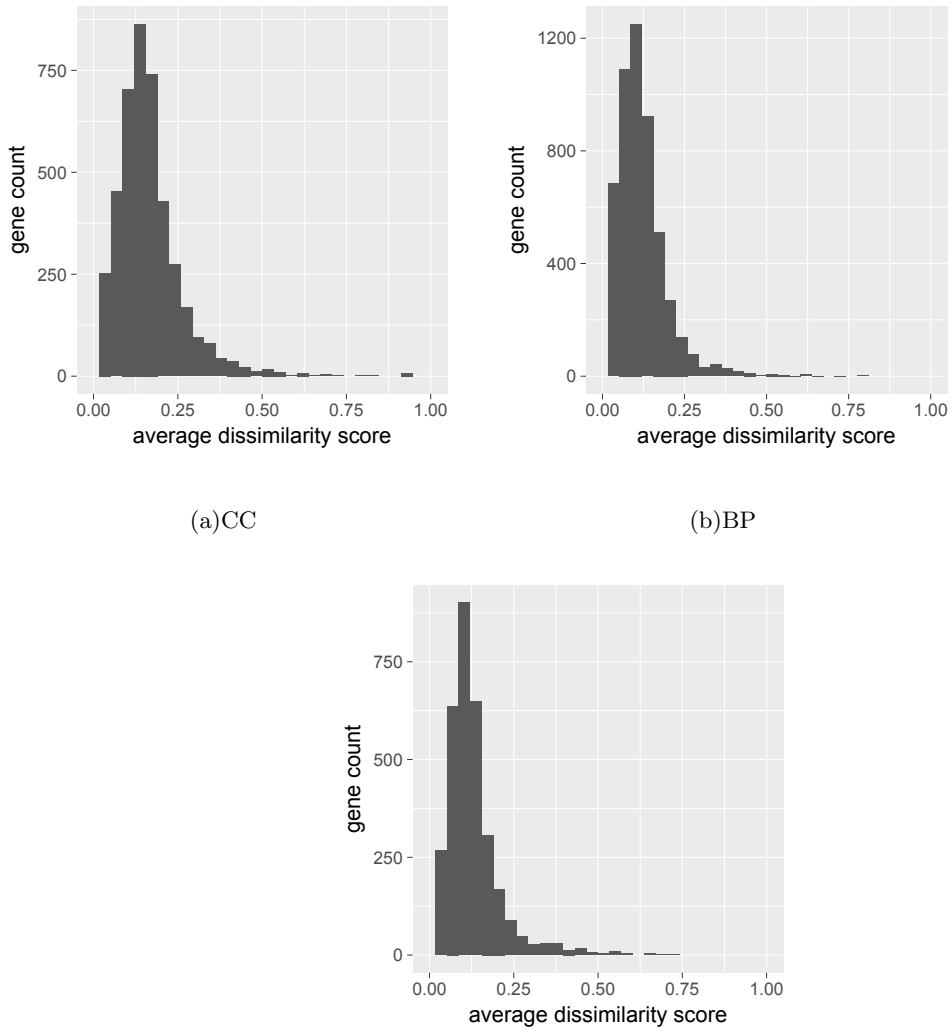


Figure 2.7: Performance of DeepIsoFun on MIGs with different numbers of isoforms. We divided MIGs into five groups: MIGs with 2 isoforms, 3 isoforms, 4 isoforms, 5 isoforms, or more than 5 isoforms. The average AUC performace of DeepIsoFun is shown in the histogram. It increases slightly as the number of isoforms per gene increases.



(a)CC

(b)BP

(c)MF

Figure 2.8: Functional dissimilarity distributions on the three main branches of GO.

Dissimilarity among the predicted functions of isoforms

Since our ultimate goal is to dissect the functions of isoforms, we estimate the functional divergence of the isoforms of the same gene. For each GO term, the gene-wise method simGIC of GOssTo (62; 67) was used to calculate the semantic similarity score

in the range of [0,1] for each gene based on the predicted functions of its isoforms. The dissimilarity score was simply defined as one minus the similarity score (4). Again, the three main branches of GO (*i.e.*, CC, BP and MF) were considered separately. Out of the 10313 MIGs, 4310 genes appear in CC, 5224 appear in BP and 3217 appear in MF (a gene may contain functions from multiple branches). For each branch, the functional divergence of a gene is calculated as the average dissimilarity scores over all terms on the branch. Figure 2.8 shows the distribution of functional dissimilarity scores among the isoforms of each gene. As observed in the literature (4; 68), many genes exhibited low average dissimilarity scores. More precisely, about 24% (1033) of the genes that appear in CC showed average dissimilarity scores less than 0.1. For BP and MF, this percentage rose to 46% (2405 genes) and 39% (1280 genes). On the other hand, about 7%, 3% and 4% of the genes have average dissimilarity scores greater than 0.3 on the three branches, respectively. These results are consistent with the fact that the isoforms of the same genes have very similar sequences, which lead them to perform mostly similar functions, but some isoforms may still have very different functions due to large changes in promoters and/or composition of coding exons.

Effectiveness of domain adaptation

A main novelty in DeepIsoFun is the use of DA (domain adaptation) to create labeled training data and transfer knowledge from the gene domain to the isoform domain. To test the effectiveness of DA in the experiments, we compared DeepIsoFun with a version without DA where the third part of the objective function in equation 2.2 is disabled. Compared with the average AUC of 0.695 achieved by the restricted DeepIsoFun without DA, DeepIsoFun with DA performed 18% better against the baseline 0.5 as shown in Figure

2.9. We then further compared the two versions on the four GO term groups based on term sizes and found that the DA technique always made a significant difference. More specifically, it helped DeepIsoFun to achieve 19%, 19%, 17%, and 20% better AUC (against the baseline 0.5) in the four groups, respectively.

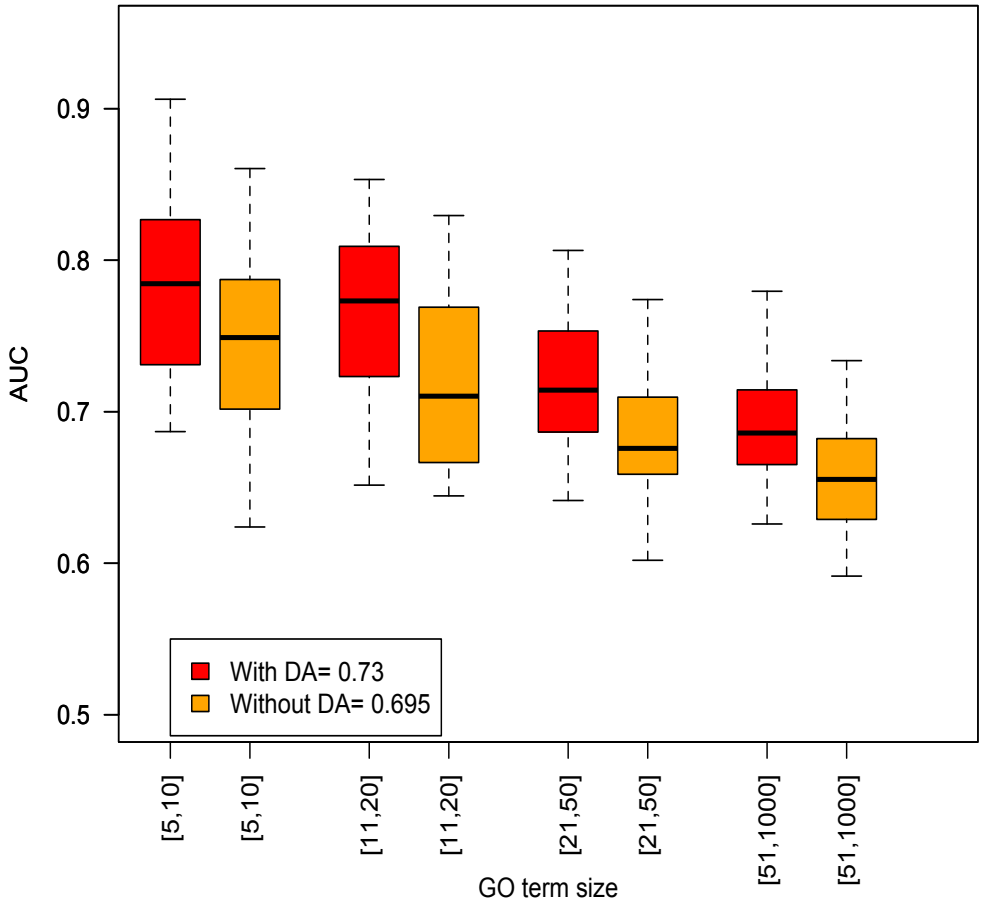
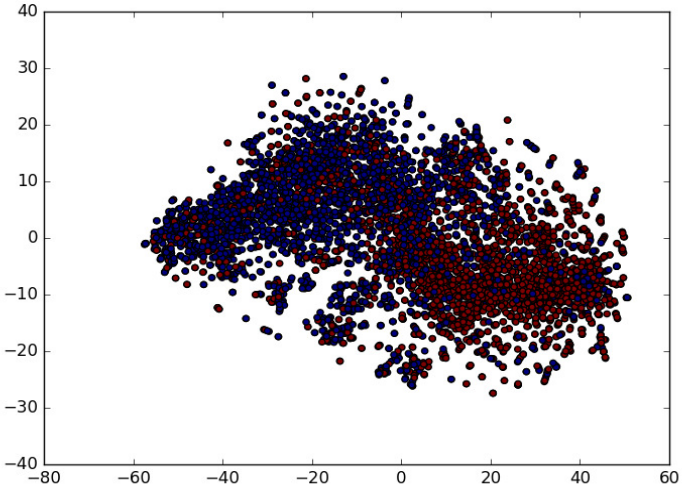
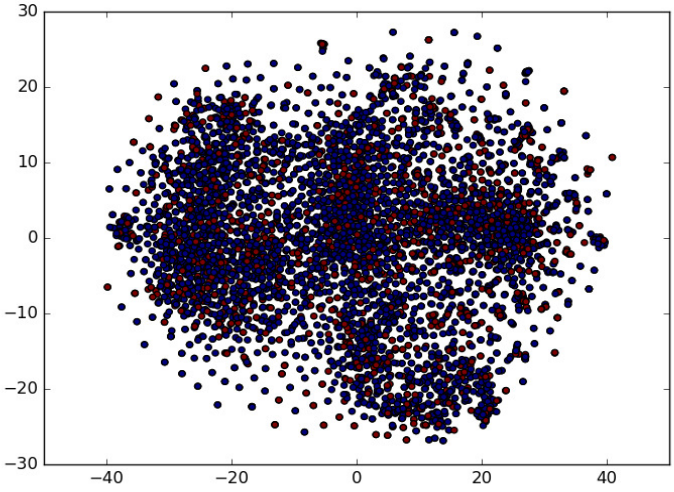


Figure 2.9: Comparison of AUCs achieved in the four groups of GO terms by DeepIsoFun with and without the DA technique. The average AUC value achieved by DeepIsoFun with DA in all four groups is 0.730 and the corresponding AUC achieved by DeepIsoFun without DA is 0.695. The benefit of DA is also clearly shown in the comparison over individual groups.



(a)Without DA



(b)With DA

Figure 2.10: Effectiveness of DA in mixing the two domains. Red dots represent samples from the source domain after feature extraction and blue dots represent samples from the target domain. The tool t-SNE (1) was used to perform dimensionality reduction and visualization. The DA technique clearly makes the two distributions harder to distinguish.

We also tested if the DA technique was actually able to mix the two domains (so knowledge can be transferred). The plots in Figure 2.10 made by using t-SNE (1) show clearly that the extracted features from the two domains became indistinguishable with the help of DA. This makes it possible to transfer knowledge (*i.e.*, the relationship between expression profiles and functions) from the gene domain to the isoform domain and is a key to the improved performance of DeepIsoFun.

Correlation between expression similarity and the similarity of predicted functions

Given the difficulty in testing the performance of DeepIsoFun directly due to the lack of isoform function benchmark, we tested how the predicted isoform functions are correlated with their expression profiles. After all, this was the original hypothesis behind the design of DeepIsoFun. We performed a hierarchical clustering of the isoforms based on the expression data and Euclidean distance by using a standard tool (hclust) in the R Stats package. Eight clusters were defined from the clustering tree using the same tool. Then, the average distance between the expression profiles of the isoforms within each cluster was calculated and normalized to the range of [0,1]. The same thing was done to estimate the average distance between the predicted GO terms of the isoforms within each cluster. The distributions of the average distances over the clusters are shown in Figure 2.11. Clearly, isoforms with similar expression profiles resulted in similar predicted functions.

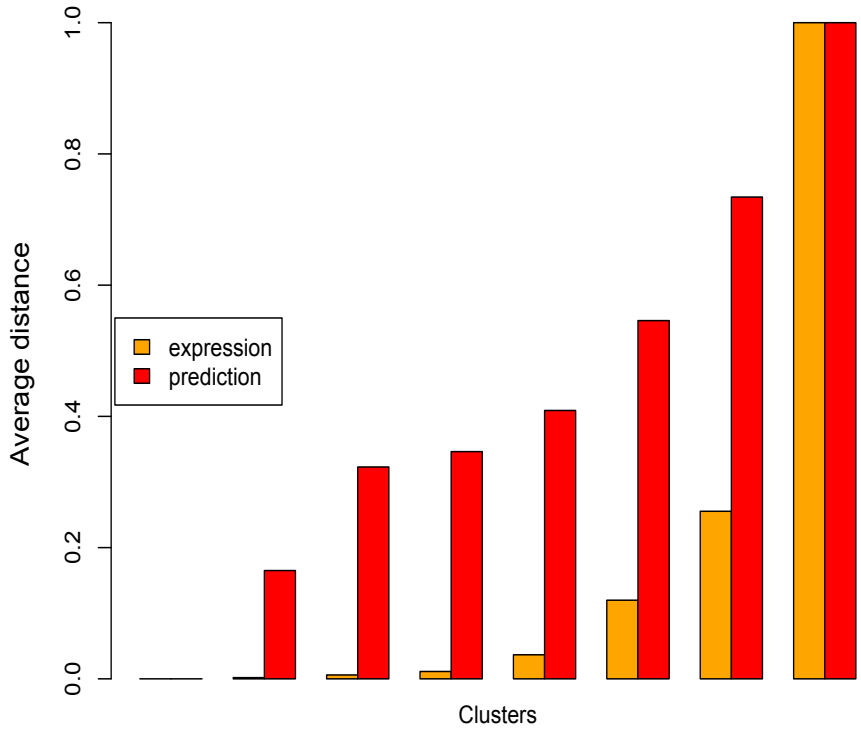


Figure 2.11: Comparison between the distribution of expression similarity and the distribution of similarity concerning predicted functions. The plot shows a clear positive correlation between the two distributions.

Comparison with the existing methods

We compared the performance of DeepIsoFun with three existing methods, iterative Multiple Instance Label Propagation (iMILP) (4), Multiple Instance SVM (mi-SVM) (3; 44) and the Weighted Logistic Regression Method (WLRM) (43). Here, iMILP is the iterative version of MILP where a feature selection wrapper method is run over MILP to achieve better performance (4). For completeness, we will also included MILP in the com-

parison. Note that WLRM was compared in (43) with two recent methods for solving MIL, namely miFV (69) and miVLAD (70), and found to perform better in the prediction of isoform functions. In addition to Dataset#1 analyzed above, we also considered the two expression datasets introduced in (3; 4), Dataset#2 and Dataset#3, respectively. Since mi-SVM and WLRM follow a 2-class classification framework but MILP/iMILP adapt a 3-class classification framework, different benchmarks were used to create functional labels for training and testing in (3; 4; 43; 44). In a 2-class classification framework, an isoform is classified as either positive or negative with respect to each GO term, while in a 3-class classification framework, an isoform is classified as either positive, negative or unknown. Hence, we present the comparison between DeepIsoFun and MILP/iMILP in Table 2.1 and the comparison among DeepIsoFun, mi-SVM and WLRM in Table 2.2. Note that the values in the two tables are not directly comparable. On all three datasets, DeepIsoFun performed significantly better than the other methods. The average AUC values achieved by DeepIsoFun on all three datasets are 0.742, 0.734 and 0.720 with respect to the first benchmark (Table 2.1), and 0.735, 0.729 and 0.704 with respect to the second benchmark (Table 2.1). The corresponding values of AUPRC are 0.368 (baseline 0.1), 0.270 (baseline 0.08) and 0.331 (baseline 0.11) with respect to the first benchmark, and 0.292 (baseline 0.1), 0.246 (baseline 0.08) and 0.234 (baseline 0.11) with respect to the second benchmark. Note that although the AUPRC values are lower, they still represent quite decent performance when compared to the baseline values. The best performance achieved on Dataset#1 is perhaps due to the quality of data (since it was collected most recently and processed with updated tools) and its diversity across different tissue conditions. On this dataset, compared to iMILP, MILP,

mi-SVM and WLRM, the AUC of DeepIsoFun increased 64%, 102%, 31%, and 23% against the baseline 0.5, respectively. Similarly, on Dataset#2 (or Dataset#3), the improvements are 73%, 216%, 37%, and 45% (or 26%, 450%, 43%, and 24%) against the baseline, respectively. Since our labeled data was imbalanced, we also compared the performance in AUPRC and observed similar improvements. On Dataset#1 (Dataset#2 and Dataset#3), DeepIsoFun performed 59% (29% and 41%, resp.) better than mi-SVM in AUPRC against respective baselines, 11% (23% and 10%, resp.) better than iMILP, 57% (32% and 20%, resp.) better than MILP, and 63% (62% and 85%, resp.) better than WLRM. We think that these significant improvements in performance over the existing methods on several human and mouse datasets demonstrate the success of the DA technique as well as the power of deep learning.

Table 2.1: Comparison between DeepIsoFun and MILP/iMILP on different expression datasets in terms of AUC and AUPRC values. Dataset#1 was generated from 1735 RNA-Seq experiments by using Kallisto (2). Dataset#2 and Dataset#3 were obtained from (3) and (4), respectively. The benchmark positive and negative instances of each GO term used in testing were defined by following the procedure in (4). The unlabeled instances were ignored in testing. Both Dataset#1 and Dataset#2 were divided based on read length to create different "study groups". There are 24, 24 and 29 study groups in Dataset#1, Dataset#2 and Dataset#3, respectively. On the average, each study group consists of 71, 16 and 17 SRA experiments in Dataset#1, Dataset#2 and Dataset#3, respectively. As done in (4), a selection algorithm was employed by iMILP to choose a subset of study groups on each dataset optimize its performance.

	AUC			AUPRC		
Method / Dataset	DeepIsoFun	MILP	iMILP	DeepIsoFun	MILP	iMILP
Dataset#1	0.742	0.620	0.648	0.368	0.271	0.342
Dataset#2	0.734	0.574	0.635	0.270	0.224	0.235
Dataset#3	0.720	0.540	0.674	0.331	0.294	0.311

Table 2.2: Comparison among DeepIsoFun, mi-SVM and WLRM on different expression datasets in terms of AUC and AUPRC values. The benchmark positive and negative instances of each GO term used in testing were defined by following the procedure in (3).

	AUC			AUPRC		
Method / Dataset	DeepIsoFun	mi-SVM	WLRM	DeepIsoFun	mi-SVM	WLRM
Dataset#1	0.735	0.679	0.691	0.292	0.221	0.218
Dataset#2	0.729	0.667	0.658	0.246	0.209	0.182
Dataset#3	0.704	0.643	0.664	0.234	0.198	0.177

Table 2.3: Average computation time (in minute) per GO term. DeepIsoFun takes 12.09 minutes to process one GO term on Dataset1 on the average (on a standard compute server node), which is 1.42 times faster than mi-SVM, 0.37 times faster than MILP, 2.49 times faster than iMILP, and 0.21 times faster than WLRM.

	DeepIsoFun	mi-SVM	MILP	iMILP	WLRM
Time	12.09	29.26	16.48	42.17	14.59

Some comparisons of the methods in terms of divergence of predicted isoform functions and time efficiency are given in Figure 2.12 and Table 2.3 . We also compared the performance of the methods on two additional datasets concerning *Arabidopsis thaliana* and *Drosophila melanogaster* (*i.e.*, fruit fly) and summarize the comparison results in Table 2.4 and Table 2.5. As the tables show, DeepIsoFun consistently performed better than the other methods in both AUC and AUPRC. The performance of the methods on all five datasets with respect to different GO term sizes is given in Tables S4 and S5.

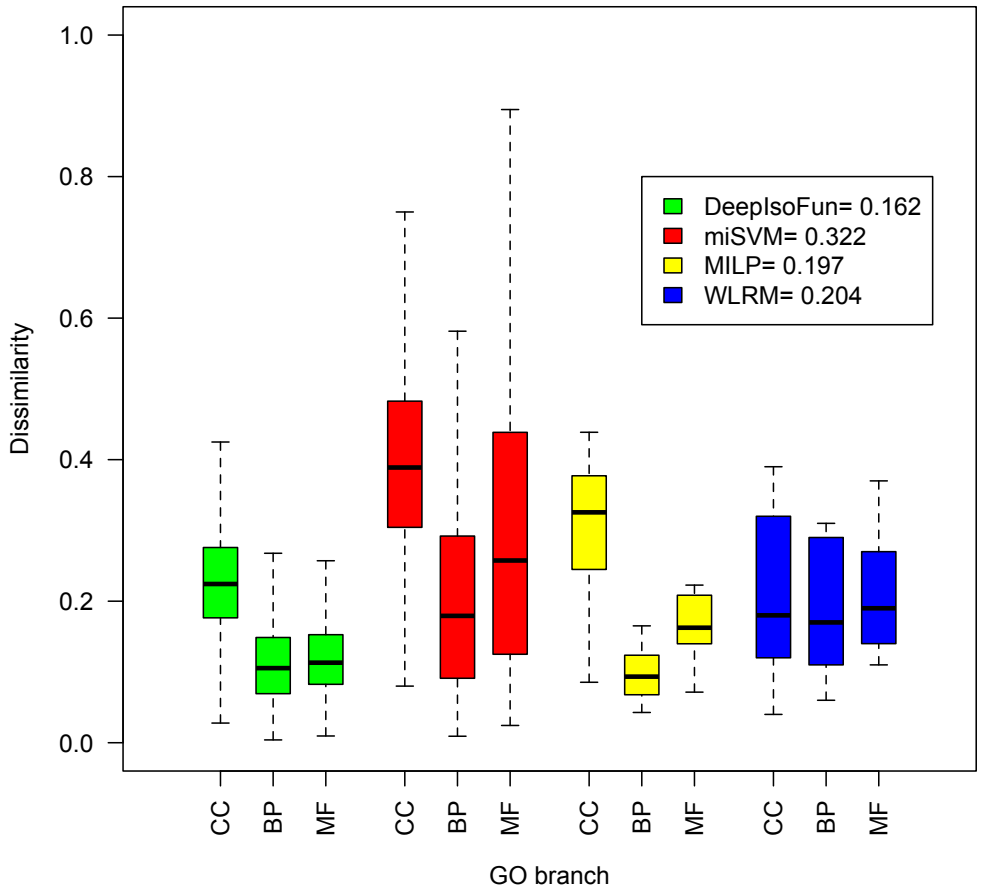


Figure 2.12: Functional dissimilarity distributions on the three main branches of GO achieved by DeepIsoFun, mi-SVM, MILP, and WLRM. The average dissimilarity scores achieved by DeepIsoFun, mi-SVM, MILP, and WLRM are respectively 0.162, 0.322, 0.197 and 0.204. Interestingly, the first three methods all reported the highest divergence on the branch CC.

Validation of some predicted isoform functions

As mentioned before, there has been little systematic study on isoform functions in the literature, and not many specific experimentally-verified functions of isoforms have been reported. Some of the reported functions concern differential regulatory behaviors of

Table 2.4: To check if DeepIsoFun consistently outperforms the other methods on data from more organisms, we tested them on two more expression datasets concerning *Arabidopsis thaliana* and *Drosophila melanogaster* (*i.e.*, fruit fly), respectively named as Dataset#4 and Dataset#5. The data generation procedure is similar as described in Section 3.2. Dataset#4 contains the expression profiles of 24315 genes and 31811 isoforms derived from 13 SRA arabidopsis studies consisting of 101 experiments and Dataset#5 contains the expression profiles of 13022 genes and 28419 isoforms derived from from 11 SRA fruit fly studies consisting of 128 experiments, with the requirement that each study contains at least 6 experiments. The transcript annotations for these two organism were collected from TAIR (https://www.arabidopsis.org/) and FlyBase (http://flybase.org/). The results in the table show that DeepIsoFun consistently performs better than MILP and iMILP. Specifically, in terms of AUC, DeepIsoFun is 97.2% and 58.1% better than MILP and iMILP on Dataset#4 (45.6% and 29.4% better on Dataset#5), respectively, against the baseline 0.5. In terms of AUPRC, DeepIsoFun performs 38.7% and 15.2% better than MILP and iMILP on Dataset#4 (33.6% and 16.1% better on Dataset#5), respectively, against the baseline 0.1.

	AUC			AUPRC		
Method / Dataset	DeepIsoFun	MILP	iMILP	DeepIsoFun	MILP	iMILP
Dataset#4	0.674	0.588	0.610	0.229	0.193	0.212
Dataset#5	0.698	0.636	0.653	0.259	0.219	0.237

Table 2.5: Comparison of DeepIsoFun, mi-SVM and WLRM on Dataset#4 and Dataset#5. Again, DeepIsoFun consistently outperforms the other two methods. In terms of AUC, DeepIsoFun is 30.6% and 22.7% better than mi-SVM and WLRM, respectively, on Dataset#4 (12.2% and 15.7% better on Dataset#5). In terms of AUPRC, DeepIsoFun is 31.1% and 40.6% better than mi-SVM and WLRM, respectively, on Dataset#4 (25.9% and 18.1% better on Dataset#5).

	AUC			AUPRC		
Method / Dataset	DeepIsoFun	mi-SVM	WLRM	DeepIsoFun	mi-SVM	WLRM
Dataset#4	0.662	0.624	0.632	0.197	0.174	0.169
Dataset#5	0.684	0.664	0.659	0.231	0.204	0.211

isoforms in important processes such as the 'regulation of apoptosis process' (GO:0042981). Apoptosis refers to programmed cell death. This GO term has two children with opposite functions, *i.e.*, the 'positive regulation of apoptosis process' or pro-apoptosis (GO:0043065)

Table 2.6: Performance of DeepIsoFun in predicting the apoptosis regulatory, pro-apoptosis and anti-apoptosis functions. The 18 genes that have multiple isoforms and are annotated with both pro-apoptosis and anti-apoptosis functions are listed in the first two columns. Here, the ID of a gene is extracted from the NCBI database. The numbers of isoforms of the genes are shown in the third column. DeepIsoFun was trained on Dataset#1 using a standard five-fold cross validation procedure with GO Slim as the benchmark. The prediction results concerning the three functions are shown in the next three columns, where an "Y" means that the concerned function is predicted for at least one of the isoforms of the concerned gene. The last column shows for each gene, if some isoforms of the gene are predicted to be pro-apoptosis but not anti-apoptosis while some other isoforms of the gene are predicted to be anti-apoptosis but not pro-apoptosis.

Gene name	**Gene ID**	**Isoform count**	**Regulation of apoptosis**	**Pro-apoptosis**	**Anti-apoptosis**	**Differentiated**
FAS	355	4	Y	Y	Y	Y
BARD1	580	5	Y	Y	Y	N
BMP4	652	9	Y	Y	Y	Y
DDX3X	1654	3	Y	N	N	N
DNAJA1	3301	2	Y	Y	N	N
IL6	3569	2	Y	Y	Y	N
MAPK8	5599	17	Y	Y	Y	Y
PSEN2	5664	2	N	N	N	N
RPS27A	6233	3	Y	N	Y	Y
SNCA	6622	4	Y	Y	N	N
YP53	7157	15	Y	Y	Y	Y
YEAF6	7189	2	Y	Y	Y	N
UBA52	7311	8	Y	Y	Y	N
UBB	7314	6	Y	Y	Y	Y
HMGA2	8091	5	Y	N	Y	N
SQSYM1	8878	3	Y	N	Y	N
ZNF268	10795	9	Y	Y	Y	Y
YNFAIP8	25816	6	Y	Y	Y	Y

Table 2.7: Performance of iMILP in predicting the apoptosis regulatory, pro-apoptosis and anti-apoptosis functions. Here, iMILP was trained on Dataset#1 using a standard five-fold cross validation procedure with GO Slim as the benchmark.

Gene name	**Gene ID**	**Isoform count**	**Regulation of apoptosis**	**Pro-apoptosis**	**Anti-apoptosis**	**Differentiated**
FAS	355	4	N	N	N	N
BARD1	580	5	Y	Y	N	N
BMP4	652	9	Y	Y	Y	N
DDX3X	1654	3	Y	N	Y	N
DNAJA1	3301	2	N	N	N	N
IL6	3569	2	Y	N	Y	N
MAPK8	5599	17	Y	Y	Y	N
PSEN2	5664	2	Y	Y	N	N
RPS27A	6233	3	Y	Y	Y	Y
SNCA	6622	4	Y	Y	N	N
YP53	7157	15	Y	Y	Y	Y
YEAF6	7189	2	N	N	N	N
UBA52	7311	8	Y	N	Y	N
UBB	7314	6	Y	Y	Y	Y
HMGA2	8091	5	Y	N	Y	N
SQSYM1	8878	3	N	N	N	N
ZNF268	10795	9	Y	Y	Y	Y
YNFAIP8	25816	6	Y	Y	Y	Y

Table 2.8: Performance of mi-SVM in predicting the apoptosis regulatory, pro-apoptosis and anti-apoptosis functions. Here, mi-SVM was trained on Dataset#1 using a standard five-fold cross validation procedure with GO Slim as the benchmark.

Gene name	**Gene ID**	**Isoform count**	**Regulation of apoptosis**	**Pro-apoptosis**	**Anti-apoptosis**	**Differentiated**
FAS	355	4	Y	N	Y	N
BARD1	580	5	Y	Y	Y	Y
BMP4	652	9	Y	Y	Y	N
DDX3X	1654	3	Y	N	Y	N
DNAJA1	3301	2	N	N	N	N
IL6	3569	2	Y	Y	N	N
MAPK8	5599	17	Y	Y	Y	Y
PSEN2	5664	2	N	N	N	N
RPS27A	6233	3	Y	Y	Y	Y
SNCA	6622	4	Y	Y	N	N
YP53	7157	15	Y	Y	Y	Y
YEAF6	7189	2	N	N	N	N
UBA52	7311	8	Y	Y	Y	N
UBB	7314	6	Y	Y	Y	N
HMGA2	8091	5	Y	Y	N	N
SQSYM1	8878	3	Y	N	Y	N
ZNF268	10795	9	Y	Y	Y	N
YNFAIP8	25816	6	Y	Y	N	N

Table 2.9: Performance of WLRM in predicting the apoptosis regulatory, pro-apoptosis and anti-apoptosis functions. Here, WLRM was trained on Dataset#1 using a standard five-fold cross validation procedure with GO Slim as the benchmark.

Gene name	**Gene ID**	**Isoform count**	**Regulation of apoptosis**	**Pro-apoptosis**	**Anti-apoptosis**	**Differentiated**
FAS	355	4	Y	Y	Y	N
BARD1	580	5	Y	Y	Y	N
BMP4	652	9	Y	Y	Y	Y
DDX3X	1654	3	N	N	N	N
DNAJA1	3301	2	Y	Y	N	N
IL6	3569	2	Y	N	Y	N
MAPK8	5599	17	Y	Y	Y	N
PSEN2	5664	2	N	N	N	N
RPS27A	6233	3	N	N	N	N
SNCA	6622	4	Y	Y	Y	N
YP53	7157	15	Y	Y	Y	Y
YEAF6	7189	2	N	N	N	N
UBA52	7311	8	Y	Y	Y	N
UBB	7314	6	Y	N	Y	Y
HMGA2	8091	5	Y	Y	Y	N
SQSYM1	8878	3	Y	Y	Y	N
ZNF268	10795	9	Y	N	Y	N
YNFAIP8	25816	6	Y	Y	Y	N

and the 'negative regulation of apoptosis process' or anti-apoptosis (GO:0043066). For MIGs with both pro-apoptosis and anti-apoptosis functions, it would be interesting to know if it has some isoforms that are pro-apoptosis but not anti-apoptosis and some other isoforms that are anti-apoptosis but not pro-apoptosis. In other words, we would like to know if the pro- and anti-apoptosis functions of the gene are *differentiated* among its isoforms. To investigate such MIGs, we searched for all genes that have multiple isoforms and are annotated with both pro-apoptosis and anti-apoptosis functions. Totally, 18 such genes were found (see Table 2.6 S6 in the Supplementary Materials). The number of isoforms in each of these genes ranges from 2 to 17. Tables 2.6, 2.7, 2.8, 2.9 show the performance of DeepIsoFun, iMILP, mi-SVM, and WLRM, respectively, in predicting the apoptosis regulatory, pro-apoptosis and anti-apoptosis functions, measured at the gene level. DeepIsoFun was able to predict the apoptosis regulatory function for the isoforms of 17 out of the 18 genes (94.4% recall), the pro-apoptosis function for the isoforms of 13 genes (72.2% recall) and the anti-apoptosis function for the isoforms of 14 genes (77.7% recall). In contrast, iMILP achieved recalls 77.7%, 55.6% and 61.1%, mi-SVM achieved recalls 83.3%, 66.7% and 61.1% and WLRM achieved recalls 77.7%, 61.1% and 72.2% in predicting the three functions, respectively. Futhermore, the tables show that DeepIsoFun was able to differentiate the pro- and anti-apoptosis functions among isoforms for 8 of the 18 genes while iMILP, mi-SVM and WLRM were only able to do it for 5, 4 and 3 genes, respectively. Although we do not know exactly how many of these genes have differentiated pro- and anti-apoptosis functions among their isoforms, it is perhaps reasonable to conjecture that most of these genes do possess this property.

2.4 Discussion

Although DeepIsoFun achieved significant improvement over the existing methods in isoform function prediction, its performance as measured by AUC and AUPRC in our experiments still remained less than desirable. The prediction of isoform functions is challenging not only because of the lack of labeled training data (*i.e.*, specific functions are known for very few isoforms) and noisy GO annotation, but also because the data is very imbalanced. That is, most GO terms are only associated with a small number of genes and hence the negative examples are far more than the positive examples. This makes the situation especially bad when the performance is measured in AUPRC since the number of false positive examples tends to be high and thus the precision tends to be low. We dealt with the problem by leaving out infrequent GO terms that are associated with fewer than five genes, although such terms often represent specific functions and could be biologically the most relevant. On the other hand, most functions of genes are yet to be discovered. Hence, three-class classification was proposed in (4) as a way to address the data imbalance issue. However, such an approach often leads to conservative predictions and may fail to predict many isoform specific functions. We plan to study machine learning (including unsupervised learning) techniques that can help produce meaningful predictions for infrequent GO terms.

Another challenge we faced was the heterogeneity of the expression data. While a large dataset covering many tissues and conditions (such as Dataset#1) provides rich information about isoform functions, it also contains a lot of noise that makes the extraction of informative features difficult. (4) solved this problem by using an elaborate search pro-

cedure to identify the best subset of RNA-Seq experiments in the input data. However, the search consumes a lot of time, especially when the number of input RNA-Seq experiments is large. We plan to apply DeepIsoFun to tissue-specific data to see how its performance will be affected as well as if some tissue-specific isoform functions can be discovered.

Chapter 3

DeepLPI: a multimodal deep learning method for predicting the interactions between lncRNAs and protein isoforms

3.1 Introduction

Long non-coding RNAs (lncRNAs) are RNA transcripts of more than 200 nucleotides that are not translated to proteins. Previous research (71; 72) has demonstrated that lncRNAs participate energetically in almost the whole process of cells. However, the functions of most lncRNAs are unknown. To understand the function of an lncRNA, it is necessary to identify what other biological molecules it is able to interact with, especially

proteins (73; 74). By interacting with proteins, lncRNAs could regulate the expression of genes, influence nuclear architecture and modulate the activity of proteins (75). Therefore, identifying lncRNA-protein interactions is an important approach to understand the potential functions of lncRNAs.

Current methods to identify lncRNA-protein interactions are based on biological experiments and computational models. With the rapid development of molecular biology techniques, large-scale experimental approaches such as PAR-CLIP (76), RNAcompete (77), HITS-CLIP (78), and RIP-Chip (79) have been developed to detect RNA-protein binding and have been used to find lncRNA-protein interactions. However, these experimental approaches are expensive and time-consuming (73). Based on the known lncRNA-protein interactions, many computational methods have been introduced for mining novel lncRNA-protein interactions. According to Zhang *et al.* (73), the computational methods could be grouped into two broad categories, machine learning-based methods and network-based methods. The machine learning-based methods build binary classifiers to predict lncRNA-protein pairs as interactive or non-interactive. These methods trained their classifiers using sequence, structure and physicochemical features of lncRNAs and proteins. For example, RPISeq (80) utilized the sequence information of RNAs and proteins to train a random forest classifier and a support vector machine classifier. Bellucci *et al.* trained catRAPID (81) using the physicochemical properties and secondary structure propensities of 592 protein-RNA pairs to predict novel RNA-protein interactions. Wang *et al.* (82) built a protein-RNA interaction prediction model using the naive Bayes classifier based only on sequence information. LncPro (83) used Fisher's linear discriminant approach to compute a matrix based

on lncRNA and protein sequence information, and used the matrix to score the interactions between an lncRNA-protein pair. Based on the sequence and secondary structural information of RNAs and proteins, RPI-Pred (84) trained a support vector machine. RpiCOOL (85) trained a random forest classifier using sequence motifs and repeat patterns. LPI-BLS (86) used sequence information of known lncRNA-protein pairs to learn multiple BLS (broad learning system) classifiers and integrated the classifiers with a logistical regression model. Recently, IPMiner (19), RPI-SAN (20), RPITER (21) and lncADeep (22) employed deep learning techniques to build lncRNA-protein interaction prediction models based on sequence and/or structural information.

Note that, there are several recently developed methods for predicting general ncRNA-protein interactions based on machine learning (87; 88; 89; 90), but they do not consider lncRNAs specifically and are hence less relevant to our work.

The network-based methods integrate heterozygous information associated with lncRNAs and proteins into a network (73) and utilize the topological relationship of lncRNAs and proteins to predict lncRNA-protein interactions. Li *et al.* proposed LPIHN (91) that integrated an lncRNA-lncRNA similarity network, an lncRNA-protein interaction network and a protein-protein interaction (PPI) network into a heterogeneous network, and used a random walk with restart technique on the heterogeneous network to infer lncRNA-protein interactions. Ge *et al.* developed a different network approach LPBNI (92) using an lncRNA-protein bipartite network inference method. Based on a heterogenous network similar to LPIHN, Xiao *et al.* proposed PLPIHS (93) using HeteSim scores (94) to infer lncRNA-protein interactions, and Hu *et al.* introduced an eigenvalue transformation-based

semi-supervised link prediction method LPI-ETSLP (95). Zhang *et al.* designed a linear neighborhood propagation method LPLNP (96). Zhao *et al.* utilized both random walk and neighborhood regularized logistic matrix factorization and proposed IRWNRLPI (97). Deng *et al.* proposed PLIPCOM (98), which combined diffusion and HeteSim features of heterogeneous lncRNA-protein networks and applied a gradient tree boosting algorithm to predict interactions. More recently, (99) combined multiple similarities and multiple features related to lncRNAs and proteins into a feature projection ensemble learning frame. Zhao *et al.* proposed a semi-supervised learning method LPI-BNPRA (100). Shen *et al.* proposed LPI-KTASLP (101), which used multivariate information about lncRNAs and proteins to conduct a semi-supervised link prediction. Xie *et al.* (102) constructed a network integrating the information about lncRNA expressions, protein-protein interactions and known lncRNA-protein interactions, and adopted a bipartite network recommendation method to predict lncRNA-protein interactions.

Though a lot of computational methods for predicting lncRNA-protein interactions have been introduced, many challenges still remain. First, in the above studies, the machine-learning based methods only focused on the intrinsic features of lncRNAs and proteins and the network based methods mostly focused on the topological features of associated biological networks of lncRNAs and proteins (73). An integration of all these features might lead to a better prediction. Second, all methods proposed above neglected the fact that a gene may encode multiple protein isoforms and different isoforms of the same gene may interact differently with the same lncRNA, which could inevitably impact their prediction performance.

In this chapter, we attempt to address these issues and propose a novel method, named DeepLPI (multimodal **Deep** learning method for predicting **L**ncRNA-**P**rotein **I**soform interactions). DeepLPI uses sequence, structure and expression data of lncRNAs and protein isoforms. Instead of using the canonical proteins of each gene, DeepLPI considers all protein isoforms, which could help to detect lncRNA-protein interactions more accurately. DeepLPI extracts intrinsic features such as functional motifs from the sequence and structure data, and obtains network topological features from the expression data. Note that, DeepLPI uses mRNA expression data to extract network topological features instead of PPI data as done in the existing methods because most of the available PPI data do not provide the details about isoforms. Moreover, it is possible to build an isoform-isoform interaction network based on mRNA expression data (103).

DeepLPI consists of two parts. In the first part, we train a deep neural network (DNN) that uses the multimodal deep learning (MDL) (23) technique to extract features from the sequence and structure data of lncRNAs and protein isoforms. The MDL fuses these extracted features and measures the initial interaction scores between lncRNAs and protein isoforms. In the second part, a conditional random field (CRF) is designed to exploit the co-expression relationship among lncRNAs and the co-expression relationship among protein isoforms. The CRF assigns final interaction scores between lncRNAs and protein isoforms based on initial interaction scores while trying to keep highly co-expressed lncRNAs and highly co-expressed protein isoforms attaining similar interaction patterns. To overcome the lack of interaction training labels for lncRNAs and protein isoforms, we propose an iterative semi-supervised training algorithm based on the multiple instance

learning (MIL) framework similar to (24; 25; 26). In MIL, for each positive lncRNA and protein interaction pair (r, p), we initially assign positive interaction labels to all pairs (r, i) for each isoform i of p and negative interaction labels to all other pairs of lncRNAs and protein isoforms. In each iteration, the DNN and CRF update the initial interaction scores using co-expressed lncRNAs and co-expressed isoforms until convergence is reached. In this setting, the isoforms of the a protein/gene can interact differently with the same lncRNA. This flexibility and the integration of both intrinsic and network topological features may potentially lead to a better prediction.

To evaluate the performance of DeepLPI, we first measure its prediction performance using protein (*i.e.*, gene) level interactions with lncRNAs provided in the NPInter v3.0 database. We make sure at least a half of our negative interaction examples contain lncRNAs and proteins that are present in the positive interactions (but do not interact with each other). The rest of the negative interactions contain lncRNAs or proteins that are not present in the positive interactions. This helps overcome the overfitting issue. DeepLPI achieved an average AUC (area under receiver operating characteristic curve) of 0.866 and AUPRC (area under the precision-recall curve) of 0.703 on the human interaction dataset. We also compare our method with both machine-learning based methods and network based methods for predicting lncRNA-protein interactions surveyed above on the same dataset. Based on availability and ease of use, 11 methods were chosen for the comparison. The experimental results demonstrate that our method significantly outperformed the others. We further evaluate the effect of various components of our model (*i.e.*, the so-called ablation study), which essentially indicates the effectiveness of each source of data (isoforms, struc-

tures, sequences, and expressions) incorporated and how these data are effectively captured by their corresponding components of the model. We also analyze the divergence of isoform interactions, *i.e.*, how isoforms from the same protein may interact differently with lncRNAs. Finally, we validate our method via a series of tests including the correlation similarity test, prediction of mouse lncRNA-protein interactions using the model trained on human data (since lncRNAs are conserved), and case studies of recently discovered lncRNA-protein interactions in the literature.

3.2 Methods

3.2.1 Datasets

The ground truth interactions between lncRNAs and proteins were downloaded from the NPInter v3.0 database (104). This is the most enriched database that integrates experimentally verified functional interactions. We kept only the interacting pairs labeled with 'Homo sapiens'. Though the data of NPInter has kept on growing, the number of involved lncRNAs and proteins is still very small at present. In the current version, there are 10031 interactions between 1817 lncRNAs and 151 proteins. These interactions are considered as positive interactions. To train a neural network model, we also need to sample negative interactions that represent pairs of lncRNAs and proteins that do not interact with each other. As the population of negative interactions count is large, complete random sampling of it may contain few lncRNAs and proteins that present in positive interactions, which might lead to overfitting (105). To reduce overfitting, we make sure that at least a half of the negative lncRNA-protein interaction pairs contain lncRNAs and proteins that

appear in positive interaction pairs (but do not interact with each other). The rest of the negative interaction pairs consist of randomly chosen lncRNAs and proteins that do not appear in positive interaction pairs.

The lncRNA sequences and the protein isoform sequences of human genome were downloaded from GENCODE (106) and ENSEMBL (107), respectively. The sequences were then used to predict their secondary structures. To predict the secondary structure of an lncRNA, we used RNAShapes (108). The output of RNAShapes was converted to a structure sequence using the EDeN tool (http://github.com/fabriziocosta/EDeN) as in (109). An lncRNA structure typically consists of six structural components: stem (S), multiloop (M), hairpin loop (H), internal loop (I), dangling end (T), and dangling start (F). To predict the secondary structure from a protein isoform sequence, we used SPIDER2 (110). SPIDER2 uses a deep neural network to predict a 3-state protein secondary structure whose structural components consist of helix (H), strand (E) and coil (C).

The third type of data that we collected are mRNA and lncRNA expression data. The mRNA expression data are obtained from the literature (32), and the lncRNA expression data were downloaded from the Co-LncRNA database (111). The mRNA and lncRNA expression data are based on high-throughput human RNA sequencing experiments of 334 studies (1,735 samples) and 241 studies (6,560 samples), respectively. We used the expression data to build co-expression networks. To ensure network quality, we only considered RNA sequencing studies with at least ten samples. Finally, 42 mRNA sequencing studies and 54 lncRNA sequencing studies were kept with a total of 1134 samples and 1429 samples, respectively. Note that an mRNA transcript uniquely corresponds to a protein

isoform. In the following, an isoform means either an mRNA transcript or protein isoform. Since different databases use different identifier naming conventions to record protein isoforms, mRNA and lncRNA, ID conversion tools from (112; 113; 111) were used to identify the same moleculars from different data sources and perform the mapping between protein isoforms and mRNAs. Finally, we filtered the data and kept the isoforms and lncRNAs that appear in both the sequence data and the expression data.

Data representation.

An lncRNA is a character sequence composed of 4 unique ribose nucleotides: cytosine (C), adenine (A), guanine (G), and uracil (U). A protein isoform is a sequence consisting of 20 unique amino acid codes. We generate hexamers and trimers from an lncRNA sequence and a protein isoform sequence, respectively. An lncRNA of length n is represented as $n-5$ consecutive hexamers of ribose nucleotides, and a protein isoform of length n is represented as $n-2$ consecutive trimers of amino acids. A hexamer of nucleotides is encoded as an integer from 0 to 4^6-1, and a trimer of amino acids is encoded as an integer from 0 to 20^3-1. As in (114), to help our deep learning model to learn the intrinsic properties of the sequences efficiently, the integer encoding sequences of lncRNAs and proteins are further encoded using a standard dense embedding technique (115). A dense embedding maps an integer index of the vocabulary to a dense vector of floats, which is achieved by an embedding layer of our deep learning network using the training data. The embedding layer aims to obtain meaningful dense vectors, which could be utilized to calculate correlations between sequences and are used as the input features of lncRNA and protein isoforms. We used a 64-dimensional dense vector to encode a hexamer of nucleotides (or a trimer of amino

acids).

Different from the sequence data, the structure of an lncRNA or a protein is often not unique, since multiple structures could be predicted for a single sequence by RNAShapes and SPIDER2. To keep more predicted structural information of an lncRNA of length n, a $6 \times n$ matrix as shown in Figure 3.1 is used to encode multiple predicted structures, where the six rows represent six different structural component types and the value at the ith row and jth column is the sum of probabilities of the predicted structures with the jth nucleotide of the lncRNA being of the ith structural component type. Similar to the lncRNA structure representation, a $3 \times n$ probability matrix as shown in Figure 3.2 is used to represent multiple predicted structures of SPIDER2 for a protein with n amino acids.

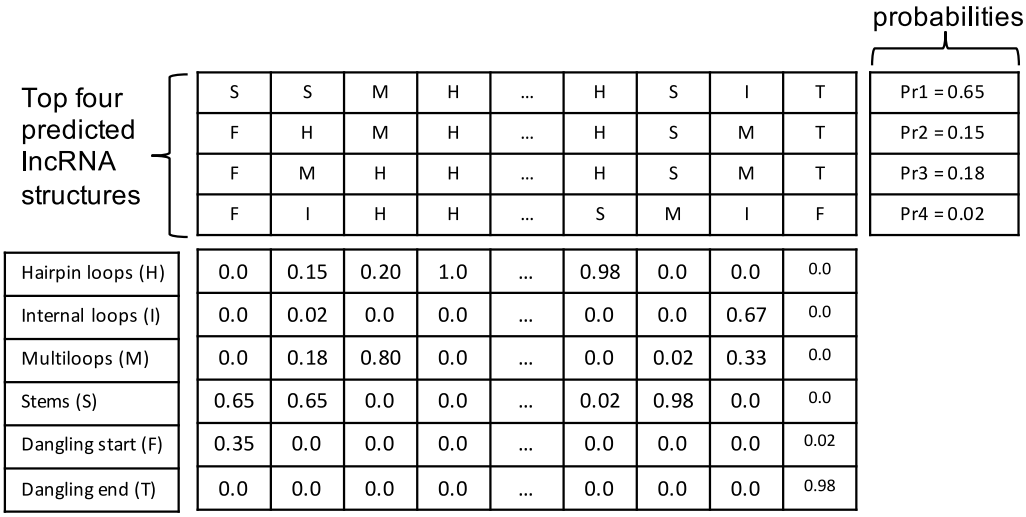


Figure 3.1: The representation of multiple predicted structures of an lncRNA. Four predicted structures are merged into a single matrix based on their probabilities.

Helix (H)	0.903	0.981	0.735	0.691	...	0.440	0.506	0.730	0.809
Strand (E)	0.004	0.001	0.006	0.004	...	0.387	0.334	0.213	0.184
Coil (C)	0.088	0.017	0.251	0.279	...	0.180	0.198	0.052	0.019

Figure 3.2: The representation of multiple predicted structures of a protein. SPIDER2 predicts the probability of each candidate structure, which is summed into a matrix according to structural component types and amino acid positions.

3.2.2 Model architecture and training

DeepLPI predicts the interactions between lncRNAs and protein isoforms by integrating the information of sequence, structure and expression data into a unified predictive model. It consists of two learning submodels. The first is a multimodal deep learning neural network (MDLNN) model and the second a conditional random field (CRF) model. The MDLNN model extracts and fuses the (intrinsic) features from the sequence and structure data of lncRNAs and protein isoforms, and calculates the initial scores of the interactions between lncRNAs and protein isoforms. The CRF model makes a final prediction based on both the initial interaction scores and the expression data of mRNAs (corresponding to protein isoforms) and lncRNAs. To overcome the lack of ground truth interactions between lncRNAs and proteins, we develop a semi-supervised algorithm following (24; 25; 26) to train the MDLNN and CRF models together iteratively. Figure 4.1 shows a schematic illustration of DeepLPI. More details of the method are described in the following subsections.

Extracting sequence and structure features using multimodal deep learning neural network.

To learn intrinsic features related to lncRNA-protein isoform interactions from the sequence and structure data, we construct a multimodal deep learning neural network (MDLNN). We use convolutional layers to extract local features and long short-term memory layer (LSTM) layers to extract short-range and long-range dependencies. At first, MDLNN uses a standard dense embedding technique (115) to map the sequences of lncRNAs and protein isoforms into a 64-dimensional vector space, which is implemented by using embedding layers (denoted as embed($\cdot$)) of Keras (116). After a training process, the embedding layers are able to learn appropriate mappings such that the mapped dense vectors could capture similarities between the sequences. Then, the dense vector matrices representing the sequences and the matrices encoding the predicted structures of lncRNAs and protein isoforms pass through one-dimensional convolutional layers with 4 convolutional filters (denoted as conv($\cdot$)) to obtain the local features of the sequence and structure data. After that, max pooling (denoted as pool($\cdot$)) layers are used to downsample the output of the convolutional layers to reduce the learning time of the subsequent layers. Based on downsampled features, LSTM layers (denoted as lstm($\cdot$)) are used to learn the features that represent the short-range and long-range intrinsic properties of the sequences and structures as in (117; 118). These features extracted from lncRNA sequences, lncRNA structures, protein isoform sequences and structures are merged together as the input of an LSTM layer followed by a dense layer. The LSTM layer and dense layer (denoted as dense($\cdot$)) are intended to learn the interaction patterns between lncRNAs and protein isoforms. Finally,

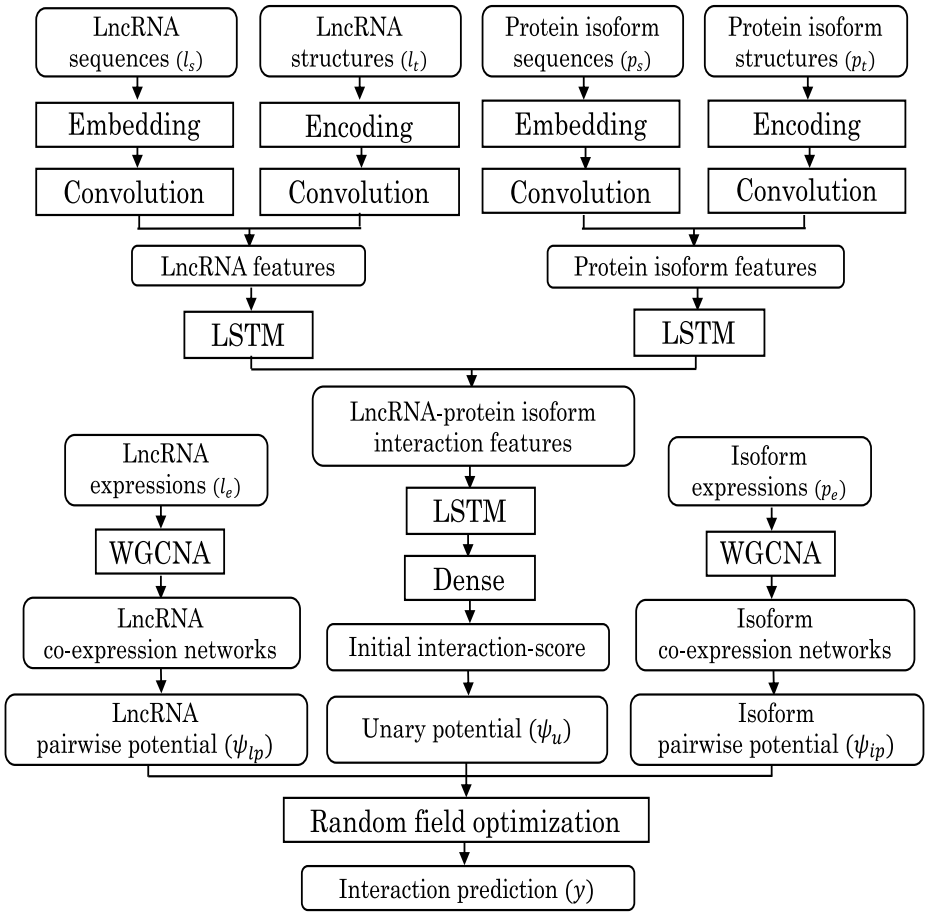


Figure 3.3: A flowchart of DeepLPI. It begins with a multimodal deep learning neural network (MDLNN) that uses embedding layers, convolutional layers, LSTM layers and other layers of Keras to extract features from the sequence and structure data of lncRNAs and protein isoforms, and calculate initial interaction scores. Weighted correlation network analysis (WGCNA) is used to construct co-expression networks from expression data of lncRNAs and protein isoforms. Based on the pairwise potentials and unary potentials inferred from the co-expression relationship and the initial interaction scores, respectively, a conditional random field (CRF) optimization is used to predict the interactions between lncRNAs and protein isoforms. The whole model is trained using an iterative semi-supervised learning algorithm based on multiple instance learning (MIL).

the output of the dense layer is fed into a logistic regression layer (denoted as logit$(\cdot)$) to compute an initial interaction score. Given an lncRNA sequence l_s, a protein isoform sequence p_s, and the predicted structures l_t and p_t of the lncRNA sequence and protein

isoform sequence, respectively, the initial interaction score (IIS) is calculated as follows:

$$
\begin{aligned}
&\text{IIS}(l_s, p_s, l_t, p_t) = \\
&\text{logit}(\text{dense}(\text{lstm}(\text{merge}(f_l(f_{l_s}(l_s), f_{l_t}(l_t)), f_p(f_{p_s}(p_s), f_{p_t}(p_t))))))\\
&f_p(f_{p_s}(p_s), f_{p_t}(p_t)) = \text{lstm}(\text{merge}(f_{p_s}(p_s), f_{p_t}(p_t)))\\
&f_l(f_{l_s}(l_s), f_{l_t}(l_t)) = \text{lstm}(\text{merge}(f_{l_s}(l_s), f_{l_t}(l_t)))\\
&f_{l_s}(l_s) = \text{pool}(\text{conv}(\text{embed}(l_s)))\\
&f_{p_s}(p_s) = \text{pool}(\text{conv}((p_s)))\\
&f_{l_t}(l_t) = \text{pool}(\text{conv}(\text{embed}(l_t)))\\
&f_{p_t}(p_t) = \text{pool}(\text{conv}((p_t)))
\end{aligned}
\tag{3.1}
$$

Incorporating co-expression relationships using a CRF.

Based on the experimental evidence that we have found in the literature, co-expressed isoforms and co-expressed lncRNAs often exhibit similar interactions (119). To incorporate the co-expression relationships between the isoforms and between the lncRNAs, we use a weighted correlation network analysis (WGCNA) method (120) to construct a co-expression network for the isoforms and one for the lncRNAs separately. In the lncRNA (or protein isoform) co-expression network, the vertices are the lncRNAs (or isoforms, respectively). The edge between vertices i and j has weight $w_{ij} = s_{ij}^{\beta}$, where s_{ij} is the absolute value of the Pearson correlation coefficient (PCC) between the expression profiles of the corresponding lncRNAs (or isoforms) and β is the soft thresholding parameter ($\beta = 6$ in our experiments as suggested by (121) for unsiged networks). Based on the pairwise poten-

tials inferred from the co-expression relationships and the unary potentials inferred from the initial interaction scores output by the MDLNN, DeepLPI next uses a conditional random field (CRF) optimization to predict the interactions between lncRNAs and protein isoforms. Note that our CRF optiimzation framework is very similar to the framework introduced in (24) for inferring isoform functions. Since many details are different, we will still include a full description of it below for completeness.

For the ith lncRNA-protein isoform pair, denote the lncRNA sequence as l_{s_i}, the protein isoform sequence as p_{s_i}, the lncRNA structure as l_{t_i}, the protein isoform structure as p_{t_i}, the lncRNA expression profile as l_{e_i}, the protein isoform expression profile as p_{e_i}, and the binary label indicating whether there is an interaction between the lncRNA and the protein isoform as y_i. The CRF optimization model aims to obtain the labels y for each lncRNA-protein isoform pair by minimizing the Gibbs energy function below:

$$\begin{aligned} E(y|l_s, p_s, l_t, p_t, l_e, p_e) = \theta_1 \sum_i \psi_u(y_i|l_{s_i}, p_{s_i}, l_{t_i}, p_{t_i}) + \\ \theta_2 \sum_{i<j} \psi_{ip}(y_i, y_j|p_{e_i}, p_{e_j}) + \theta_3 \sum_{i<j} \psi_{lp}(y_i, y_j|l_{e_i}, l_{e_j}) \end{aligned} \tag{3.2}$$

Here, the Gibbs energy is a weighted summation of unary potentials ψ_u, isoform pairwise potentials ψ_{ip} and lncRNA pairwise potentials ψ_{lp}. The unary potentials ψ_u are calculated from the the initial interaction scores as $\psi_u(0|l_{s_i}, p_{s_i}, l_{t_i}, p_{t_i}) = \text{IIS}(l_s, p_s, l_t, p_t)$ and $\psi_u(1|l_{s_i}, p_{s_i}, l_{t_i}, p_{t_i}) = 1 - \text{IIS}(l_s, p_s, l_t, p_t)$. For the ith and jth lncRNA-protein isoform pairs, their pairwise potential is defined as follows:

$$\begin{aligned} \psi_{ip}(y_i, y_j|p_{e_i}, p_{e_j}) = \mu_p(y_i, y_j) \sum_q w_q(p_{e_i}, p_{e_j}) \\ \psi_{lp}(y_i, y_j|l_{e_i}, l_{e_j}) = \mu_p(y_i, y_j) \sum_r w_r(l_{e_i}, l_{e_j}) \end{aligned} \tag{3.3}$$

where $w_q(p_{e_i}, p_{e_j})$ is the weight of the edge between isoforms i and j in the q-th isoform co-expression network and $w_r(p_{e_i}, p_{e_j})$ is the weight of the edge between lncRNAs i

and j in the r-th lncRNA co-expression network. $\mu_p(y_i, y_j)$ is a label compatibility function whose value is 1 if $y_i \neq y_j$ or 0 otherwise. It is used to penalize highly co-expressed isoforms and highly co-expressed lncRNAs assigned with different interaction labels. The weights θ_1, θ_2 and θ_3 are used to control the relative importance of ψ_u, ψ_{ip} and ψ_{lp} in the Gibbs energy. They will be discussed in the next subsection.

By searching for an assignment $\hat{y}$ of labels minimizing the Gibbs energy $E(\hat{y}|l_s, p_s, l_t, p_t, l_e, p_e)$, we attempt to find appropriate labels for lncRNA-protein isoform pairs with low unary energies such that highly co-expressed isoforms would have the same interaction patterns with highly co-expressed lncRNAs. Since computing an exact solution to the Gibbs energy minimization problem is challenging, we apply an efficient approximation algorithm called the mean-field approximation as in (122) to obtain an approximate solution, sketched below.

It is easy to see that minimizing the Gibbs energy is equal to maximizing the following probability:

$$P(y|l_s, p_s, l_t, p_t, l_e, p_e) = \frac{1}{Z} exp(-E(y|l_s, p_s, l_t, p_t, l_e, p_e)) \tag{3.4}$$

where $Z = \sum_y exp(-E(y|l_s, p_s, l_t, p_t, l_e, p_e))$ is a normalization constant. Let $Q(y|l_s, p_s, l_t, p_t, l_e, p_e)$ be the product of independent marginal probabilities, *i.e.*,

$$Q(y|l_s, p_s, l_t, p_t, l_e, p_e) = \prod_i Q_i(y_i|l_{s_i}, p_{s_i}, l_{t_i}, p_{t_i}, l_{e_i}, p_{e_i}) \tag{3.5}$$

Instead of computing the exact distribution of $P(y|l_s, p_s, l_t, p_t, l_e, p_e)$, we use $Q(y|l_s, p_s, l_t, p_t, l_e, p_e)$ with the minimum KL-divergence $\mathbf{D}(Q||P)$ to approximate P, and adopt the following it-

erative update equation to obtain a Q with the minimum KL-divergence:

$$
\begin{aligned}
&Q_i(y_i|l_{s_i}, p_{s_i}, l_{t_i}, p_{t_i}, l_{e_i}, p_{e_i}) = \\
&\frac{1}{Z_i}\ exp\{-\theta_1\psi_u(y_i|l_{s_i}, p_{s_i}, l_{t_i}, p_{t_i}) \\
&- \theta_2 \sum_{i\neq j}\sum_{q} w_q(p_{e_i}, p_{e_j})Q_j(1-y_i|l_{s_j}, p_{s_j}, l_{t_j}, p_{t_j}, l_{e_j}, p_{e_j}) \\
&- \theta_3 \sum_{i\neq j}\sum_{r} w_r(l_{e_i}, l_{e_j})Q_j(1-y_i|l_{s_j}, p_{s_j}, l_{t_j}, p_{t_j}, l_{e_j}, p_{e_j})\}
\end{aligned}
\tag{3.6}
$$

Here, we initialize Q_i with the unary potential and update it iteratively according to Equation 3.6 until convergence, when the final output of our model is obtained.

Training the model with the MIL framework.

Because the ground truth lncRNA-protein isoform interactions are generally unavailable, conventional supervised training algorithms cannot be directly applied to our model. Similar to (24) again, here we adopt a semi-supervised training algorithm under the MIL framework as in (25; 26). In this MIL framework, for each lncRNA, a protein/gene is treated as a bag, the isoforms of a protein/gene are treated as the instances in the bag, and only the ground truth of the bag (*i.e.*, the true lncRNA-protein interaction label) is assumed. We further require that a positive bag should contain at least one positive instance and a negative bag should contain no positive instances. DeepLPI first initializes all instances of positive bags with positive labels, and the other instances with negative labels. Then, the model parameters are optimized with the initial labels in the following standard supervised learning manner.

Given a batch of training instances $(l_s, p_s, l_t, p_t, l_e, p_e, \hat{y})$, the loss functions in terms of the MDLNN parameters w and in terms of the CRF parameters θ are defined as the

following negative log likelihoods, respectively.

$$\begin{aligned}\ell_{\text{MDLNN}}(w : l_s,p_s,l_t,p_t,\hat{y}) = -\sum_i (\hat{y_i}\log(\text{IIS}(l_{s_i},p_{s_i},l_{t_i},p_{t_i})) \\ +(1-\hat{y_i})\log(1-\text{IIS}(l_{s_i},p_{s_i},l_{t_i},p_{t_i})))\end{aligned} \tag{3.7}$$

$$\begin{aligned}\ell_{\text{CRF}}(\theta : l_s,p_s,l_t,p_t,l_e,p_e,\hat{y}) = \\ -\log P(\hat{y}|l_s,p_s,l_t,p_t,l_e,p_e) + \sum_i \frac{{\theta_i}^2}{2\sigma^2}\end{aligned} \tag{3.8}$$

In Equation 3.8, the parameter σ is used to regularize the importance of the co-expression networks in the model optimization and set as 0.1 in our following experiments. We use the Nadam optimization algorithm to update the MDLNN parameters w so ℓ_{MDLNN} could be minimized. To minimize ℓ_{CRF}, we use the L-BFGS-B algorithm as in (24) to update CRF parameters θ.

We perform inference for every instance in the positive bags after each update of the parameters of the model, using the model with the updated parameters. Here, the label of an instance is updated according to the inference: $\hat{y_i} = \text{argmax}_{y_i} P_i(y_i)$. We also adopt the following constraint: for each positive bag, if all its instances are assigned with negative labels, we force the instance with the largest positive prediction score $P_i(1)$ in the bag as positive. The steps of updating parameters and labels are repeated alternately until convergence.

3.3 Results and validation

In this section, we first compare the performance of DeepLPI with some state-of-the-art methods and analyze the effectiveness of our method in terms of each type of data we used and each component of the model. Next, we validate the prediction results

of DeepLPI using correlation analyses, a mouse dataset as well as some newly discovered human lncRNA-protein interactions in the literature.

3.3.1 Prediction of lncRNA-protein interactions

We first compare the performance of DeepLPI with both of machine-learning based methods and network based methods. Then, we evaluate the effectiveness of each component of our model (*i.e.*, the ablation study). We also study the divergence of lncRNAs interacting with different isoforms of the same protein/gene, and compare the structural components of lncRNAs and protein isoforms in both interactive and non-interactive pairs. Finally, we test DeepLPI on some smaller and older lncRNA-protein interaction datasets and observe how its performance could be impacted by the size of training data.

Prediction performance comparison between DeepLPI and the existing methods.

Since there is no benchmark data for lncRNA-protein isoform interactions, we could only evaluate the performance of DeepLPI based on the benchmark data of (human) lncRNA-protein interactions downloaded from the NPInter v3.0 (104) database. We compare DeepLPI with some state-of-the-art methods including machine-learning based and network based methods.

The popular machine-learning based methods are catRAPID (81), RPISeq (80), lncPro (83), RPI-Pred (84), rpiCOOL (85), IPMiner (19), RPI-SAN (20), lncADeep (22), RPITER (21) and LPI-BLS (86). Among these methods, some (lncPro and rpiCOOL)

Table 3.1: Comparison of prediction performance on lncRNA-protein interactions on the NPInter v3.0 human dataset.

Broad category	Methods	AUC	AUPRC
Machine-learning based methods	RPISeq (RF)	0.708	0.486
	RPISeq (SVM)	0.701	0.473
	lncPro	0.723	0.588
	rpiCOOL	0.721	0.503
	IPMiner	0.714	0.569
	lncADeep	0.825	0.646
	RPITER	0.827	0.664
	LPI-BLS	0.782	0.575
	DeepLPI	0.866	0.703
Network based methods	LPIHN	0.776	0.421
	LPBNI	0.786	0.559
	PLPIHS	0.672	0.483
	LPLNP	0.801	0.566
	PLIPCOM	0.821	0.609
	SFPEL-LPI	0.823	0.599

provide stand-alone programs, some (catRAPID, RPISeq and RPI-Pred) provide web-based services, some (IPMiner, lncADeep, RPITER and LPI-BLS) are re-trainable with available source codes, while the others are unavailable. Predicting lncRNA-protein interactions on a large scale using web-based services of catRAPID and RPI-Pred is time-consuming and often fails in the case of long input sequences. The publicly available network based methods are LPIHN (91), LPBNI (92), PLPIHS (123), LPLNP (96), PLIPCOM (98), and SFPEL-LPI (99). Therefore, we compare our method with seven machine-learning based methods lncPro, rpiCOOL, IPMiner, lncADeep, RPITER, LPI-BLS and RPISeq, and six network based methods LPIHN, LPBNI, PLPIHS, LPLNP, PLIPCOM, and SFPEL-LPI. Default parameters of these methods are used as recommended by their authors.

Table 4.1 shows the average test results in 10 runs of five-fold cross validations on the NPInter v3.0 human dataset. The AUC values of RPISeq (RF), RPISeq (SVM),

lncPro, rpiCOOL, IPMiner, lncADeep, RPITER, LPI-BLS and DeepLPI are 0.708, 0.701, 0.723, 0.721, 0.714, 0.825, 0.827, 0.782 and 0.866, respectively, and their AUPRC values are 0.486, 0.473, 0.588, 0.503, 0.569, 0.646, 0.664, 0.575 and 0.685, respectively. DeepLPI outperformed these machine-learning based methods by 22.3%, 23.5%, 19.7%, 20.1%, 21.2%, 4.9%, 4.7%, and 10.7% in terms of AUC and by 44.6%, 48.6%, 19.6%, 39.8%, 23.6%, 8.8%, 5.9%, and 22.2% in terms of AUPRC, respectively. The AUC values of LPIHN, LPBNI, PLPIHS, LPLNP, PLIPCOM and SFPEL-LPI are 0.776, 0.786, 0.672, 0.801, 0.821 and 0.823, respectively, and the AUPRC values are 0.421, 0.559, 0.483, 0.566, 0.609 and 0.599, respectively. DeepLPI also outperformed these network based methods by 11.6%, 10.2%, 28.9%, 8.1%, 5.5% and 5.2% in terms of AUC and 67.0%, 25.8%, 45.5%, 24.2%, 15.4% and 17.4% in terms of AUPRC scores, respectively. Since these results show that DeepLPI, lncADeep and RPITER performed better than the others, we will only compare these three methods in the following experiments.

To test if our sampling method for generating negative interactions is helpful in reducing overfitting, we repeat the above experiment with all negative interactions sampled randomly and compare DeepLPI with two of the best-performing existing methods, lncADeep and RPITER. The AUC values of DeepLPI, lncADeep and RPITER are 0.923, 0.905 and 0.894, respectively, and their AUPRC values are 0.776, 0.753 and 0.747, respectively. While all AUC and AUPRC values of the three methods have increased significantly, DeepLPI consistently performs better than the other two.

We also evaluate the performance of the methods using the leave-one-out cross-validation (LOOCV) experiment, although it is computationally more expensive. In this

experiment, the AUC values of DeepLPI, lncADeep and RPITER are 0.855, 0.801 and 0.811, respectively, and their AUPRC values are 0.694, 0.638 and 0.649, respectively. Compared to those in the five-fold cross-validation experiment, the AUC and AUPRC values of all methods decreased a little, which might be due to variance in the data as discussed in (124).

In order to test if homologous protein sequences might have an impact on the performance of DeepLPI and potentially cause data leak and/or model overfitting, we search for homologous proteins in our benchmark dataset based on EggNog (125). It turns out that only 5% of the proteins are homologous (to other sequences). We repeat the above five-fold cross-validation experiment for DeepLPI by keeping all interactions involving homologous proteins in the same fold. The AUC and AUPRC values decrease only slightly from 0.866 to 0.861 and from 0.703 to 0.699. This suggests that data leak or model overfitting were unlikely or very limited in our experiment.

Analyzing the effects of model components.

In order to assess the contribution of the biological features considered in our model as well as its major computational components, we conduct an ablation study by removing various features/components from the model and evaluate how such a change would affect the performance of the model. More specifically, we test how the model is affected when the MIL learning with protein isoforms is replaced by conventional learning with proteins, when the CRF component along with the expression data are removed, and when the sequence or structure data are removed.

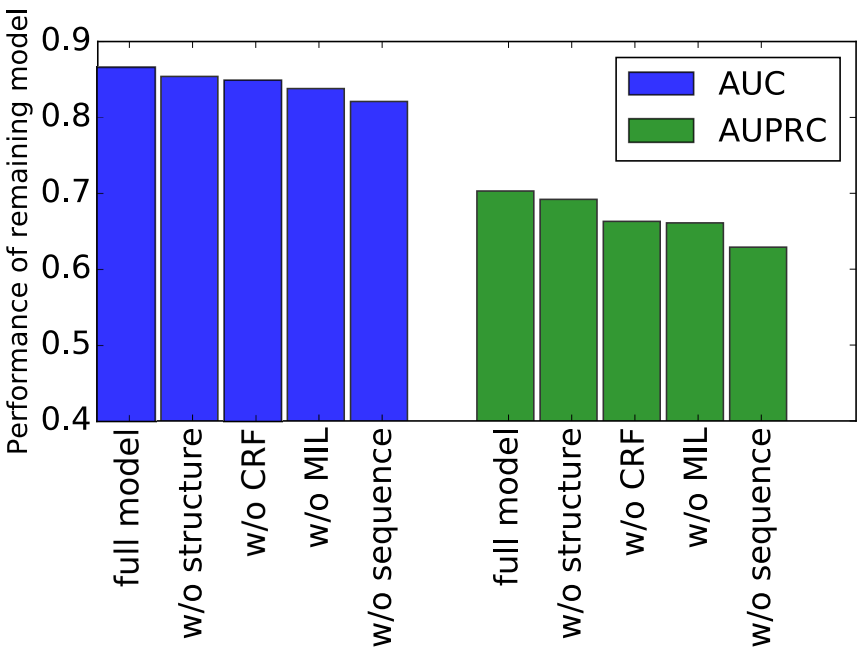


Figure 3.4: The effect on performance of removing various components from the model. The average AUC and AUPRC values of DeepLPI, DeepLPI without structure data, DeepLPI without CRF, DeepLPI without using MIL/isoforms, and DeepLPI without sequence data are shown in the figure.

Figure 4.4 shows that the average AUC of DeepLPI dropped 1.4%, 2.0%, 3.3%, and 5.5% without the structure data, without the CRF component for incorporating expression data, without the MIL learning framework for incorporating isoforms, and without the sequence data, respectively. Without these components or data, the performance in term of AUPRC shows a similar declining trend with the percentage decreases being 1.6%, 6.0%, 6.3% and 11.6%, respectively. In particular, when we consider proteins instead of protein isoforms in the model, its AUC dropped from 0.866 to 0.842, which demonstrates the significance of using isoform data. The results also suggest that the CRF component was effective in improving the prediction performance via the integration of expression data. Among all types of input data, the sequences are clearly the most important for the model.

Although the usage of structure data did not boost the performance of the model significantly, it allows us to observe interesting enrichment of structural components in interactive lncRNAs, as discussed in the next subsection.

Structural components at important positions in interactive and non-interactive pairs.

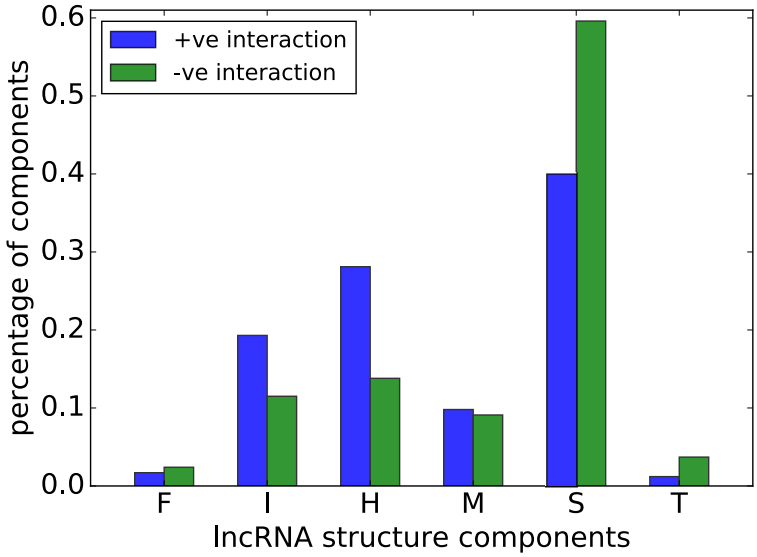
It would be interesting to study how the structural components of lncRNAs and protein isoforms are distributed in interactive pairs, especially around their interacting sites, and what structural components may contribute more to the interactions than the others. For each lncRNA-protein isoform pair, we use saliency maps (126) to compute importance weights at each position in both sequences. These weights indicate how a position might impact the prediction outcome by our model (*i.e.*, interactive or non-interactive). The lncRNA and protein structural components at heavily weighted (*i.e.*, important) positions of interactive and non-interactive pairs are profiled and shown in Figure (a). For each structural component, the average occurrence frequency across all instances is calculated. We can see that at important lncRNA positions, hairpin loops (H) occur much more often in interactive pairs than in non-interactive pairs. The same appears to be true for inner loops (I). On the other hand, stems (S) occur much often at the important lncRNA positions of interactive pairs than at the important lncRNA positions of non-interactive pairs. These suggest that open/unpaired lncRNA positions perhaps play more important roles in their interactions with proteins, and is consistent with several studies in the literature (127; 74).

Similar to lncRNA structural components, we also profile protein isoform structural components in Figure (b). However, we are unable to observe a significant difference

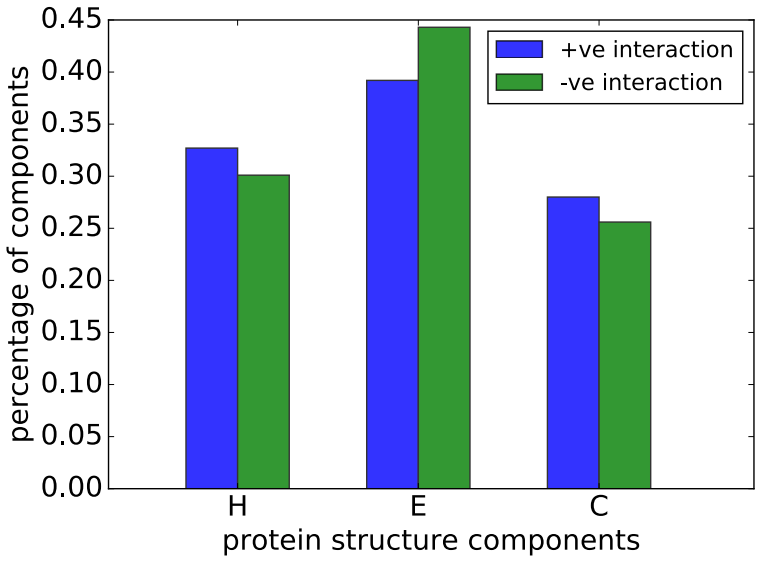
between the distributions of the structural components at important protein isoform positions of interactive and non-interactive pairs. We suspect that a more detailed protein structure representation might help reveal some difference, but was unable to pursue it given the time complexity involved in obtaining such representations with high quality.

Divergence of lncRNAs interacting with the isoforms of the same protein.

Our ultimate goal is to find lncRNA interactions at the isoform level. Hence, it would be useful to analyze how lncRNAs interact divergently with the isoforms of the same protein. We first estimate the similarity of predicted lncRNA interactions for each pair of isoforms in terms of the semantic similarity score using GOssTo (128). As in (24; 32), the semantic dissimilarity score between two isoforms is then defined as one minus their similarity score. We consider only proteins with multiple isoforms (MIPs) and collected all interactions between lncRNAs and the isoforms of the MIPs as predicted by DeepLPI trained on the the NPInter v3.0 dataset. For each MIP, the interaction divergence of its isoforms was calculated by averaging the semantic dissimilarity scores of all pairs of its isoforms. Among these MIPs, 71.6% (1,548 out of 2,163) were estimated to have divergent isoform interactions (*i.e.*, with semantic dissimilarity scores greater than 0). The dissimilarity score distributions for MIPs that have divergent isoform interactions are shown in Figure 3.6 where the mean score value is 0.302.



((a))



((b))

Figure 3.5: (a) Distribution of lncRNA structural components. (b) Distribution of protein structural components.

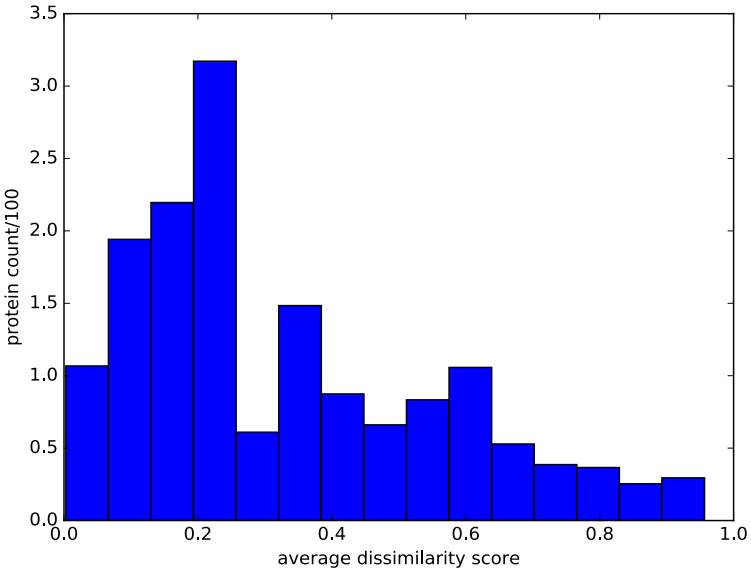


Figure 3.6: Distributions of semantic dissimilarity scores of MIPs. For each MIP, the semantic dissimilarity score indicates the divergence of the lncRNAs interacting with its different isoforms. The score range of [0, 1] is equally divided into 15 bins. For each bin, we count how many MIPs have semantic dissimilarity scores in this range.

Table 3.2: Performance of DeepLPI, lncADeep and RPITER when datasets from RPI369, RPI1807, RPI2241, and NPInter v2.0 are used for training and the NPInter v3.0 dataset is used for testing. Here, #int represents the number of positive lncRNA-protein interactions contained in a training dataset. As the training data increases, the performance DeepLPI, lncADeep and RPITER improves as expected, but the rate of improvement for DeepLPI is higher than the other methods.

Train \ Test		NPInter v3.0					
Name	#int	DeepLPI		lncADeep		RPITER	
		AUC	AUPRC	AUC	AUPRC	AUC	AUPRC
RPI369	369	0.563	0.202	0.549	0.195	0.551	0.197
RPI1807	1807	0.597	0.271	0.574	0.260	0.580	0.263
RPI2241	2241	0.626	0.328	0.597	0.302	0.609	0.321
NPInter v2.0	6204	0.733	0.513	0.681	0.461	0.693	0.474

The impact of training data size.

We have collected several (older and smaller) ncRNA-protein interaction datasets including RPI369, RPI1807, RPI2241, and NPInter v2.0. We would like to test how the

DeepLPI, lncADeep and RPITER methods perform when these different datasets are used for training and the comparatively newer dataset NPInter v3.0 is used for testing. Since the datasets overlap with each other quite a bit, we make sure that the test interactions do not contain any of the training interactions to prevent a possible data leak. The prediction results are shown in Table 3.2. The results suggests that the sample size of the training data has a significant effect on the prediction performance of DeepLPI, lncADeep and RPITER. When more training samples are available, these models achieve a better prediction performance, as expected. However, the rate of improvement with respect to the number of interactions is much higher for DeepLPI than for the other two methods.

3.3.2 Validation of predicted lncRNA-protein isoform interactions

To validate the prediction lncRNA-protein isoform interaction results of DeepLPI, we analyze the correlations between isoform sequence similarity, lncRNA sequence similarity, as well as their structure similarity and expression similarity. Moreover, we evaluate the prediction performance of DeepLPI (trained on the NPInter v3.0 human interaction data) using a mouse lncRNA-protein interaction dataset and some new human lncRNA-protein interactions from the recent literature that were not included in the NPInter v3.0 database as the test data.

Correlation analyses.

Our basic assumption is that similar lncRNAs tend to interact with similar protein isoforms. To check if our predicted interactions accord to the assumption, we conducted a series analyses of correlation between the similarity of lncRNAs and the similarity of their

interactive protein isoforms.

From the lncRNA-protein isoform interactions predicted by DeepLPI, we grouped 1,534 involved lncRNAs into 50 clusters according to a hierarchical clustering based on a generalized Levenshtein (edit) distance. For each group, we calculated a (sequence, structure or expression) similarity score for each pair of lncRNAs in the group and the average score of the group. We also calculated a similarity score for each pair of protein isoforms that have interaction with the lncRNAs in the group and the average score of all such pairs of protein isoforms. The similarity score between two lncRNA (or protein isoform) sequences is defined as the global alignment score normalized by the alignment length. All similarity scores were normalized to the range of [0, 1]. At last, Pearson's correlation coefficient (PCC) was used to measure the pairwise correlation between lncRNA sequence similarity and protein isoform sequence similarity (Fig. (a)). Similarly, we calculated the PCC between lncRNA expression similarity and protein isoform expression similarity (Fig. (b)), and the PCC between lncRNA structure similarity and protein isoform structure similarity (Fig. (c)). The PCC between lncRNAs sequence similarity and lncRNA expression similarity (Fig. (d)) and the PCC between protein isoform sequence similarity and isoform expression similarity (Fig. (e)) are also included as a useful reference.

Clearly, positive correlations are found in all above analyzes. The strong correlations in Fig. (a) - Fig. (c) conform that similar lncRNAs tend to interact with similar protein isoforms. An interesting observation is that our correlation analysis results are highly consistent with the experimental results in subsection 4.3.1. For example, the strongest correlation between the sequence similarities of lncRNAs and protein isoforms is consistent

with the most significant drop in the prediction performance when the sequence data was removed. The moderate correlation coefficients in Fig. (d) and Fig. (e) suggest that the sequence and expression data contain complementary features and thus might explain why their combination helped improve the performance of our model.

Table 3.3: Prediction of interactions involving mouse lncRNA Gas5.

lncRNA	Protein	Machine-learning based methods							Network based methods				
		lncPro	IPMiner	lncADeep	rpiCool	RPITER	LPI-BLS	DeepLPI	LPIHN	LPBNI	LPLNP	PLIPCOM	SFPEL-LPI
Gas5	Q92900	o	o	o	x	o	x	o	o	o	o	o	o
Gas5	Q92833	o	o	o	x	o	x	o	o	o	o	o	o
Gas5	Q8BIF2	x	x	o	x	o	x	o	x	x	x	x	o
Gas5	Q16630	x	x	x	x	x	x	o	x	x	x	x	o
Gas5	Q9NR56	x	x	x	x	x	x	x	x	x	x	x	o
Gas5	Q8R003	o	x	o	x	o	o	o	x	x	x	o	x
Gas5	P38432	x	x	o	x	o	o	o	x	o	o	o	x
Gas5	P84103	x	o	o	x	o	o	o	x	o	o	o	x
Gas5	Q08170	x	x	o	x	o	o	o	o	x	x	o	x
Gas5	Q15910	x	o	o	x	o	o	o	x	x	x	x	x
Gas5	H0YB86	x	o	x	o	x	o	o	o	o	o	x	o
Gas5	P35637	x	x	x	o	o	x	o	x	o	x	o	o
Gas5	Q921F2	x	x	o	o	x	x	o	x	x	o	x	o
Recall		.231	.385	.692	.231	.615	.462	.923	.385	.462	.462	.538	.615

Note: The predicted and unpredicted interactions are represented as circles and crosses in the table.

Performance on an independent interaction dataset of mouse.

To further validate the effectiveness of DeepLPI in lncRNA-protein interaction prediction, we test DeepLPI and the other existing methods on a dataset independent from the training data. More specifically, we trained all models with the human lncRNA-protein interactions from the NPinter v3.0 database and tested the models on 3580 mouse lncRNA-protein interactions in the same database. Although there is a high genetic similarity between mouse and human (and hence the conservation of lncRNAs), the performance of all models dropped. The AUC of DeepLPI decreased from 0.866 (human) to 0.753 (mouse), but it was still the best since the highest AUC of the other models on the mouse test data

was 0.68. An obvious reason for the performance drops might be because lncRNAs do not show the same pattern of evolutionary conservation as protein-coding genes (129).

To further investigate the prediction performance of DeepLPI on interactions between proteins and lncRNAs conserved between human and mouse such as Gas5, Rmst, Neat1 and Meg3 (129; 130), we selected 39 interactions involving conserved lncRNAs from the 3580 mouse interactions. Of the 39 interactions, 89.7% have been correctly predicted by DeepLPI. In particular, since Gas5 is an extensively studied mouse lncRNA that plays an important role in modulating self-renewal (131), we show the interaction prediction results concerning mouse Gas5 in Table 3.3. The table demonstrates again that DeepLPI achieved the highest prediction accuracy.

Table 3.4: The source of 12 recently reported lncRNA-protein interactions in the literature.

lncRNA	**Protein**	**Reference**
PANDAR	PTBP1	Found in paper (132)
lnc-SH2D7.1	FBXL22	Found in paper (133)
lnc-SH2D7.1	LPIN2	Found in paper (133)
lnc-DCAF811.1	PEBP1	Found in paper (133)
lnc-DCAF811.1	DNAJB12	Found in paper (133)
lnc-NIT1	POR	Found in paper (133)
AC011498.1	CEBPA	Found in paper (134)
CRNED	CEBPA	Found in paper (134)
LINC00504	CEBPA	Found in paper (134)
AC011498.1	NPM1	Found in paper (134)
LL22NC03-N64E9.1	KLF2	Found in paper (135)
LL22NC03-N64E9.1	EZH2	Found in paper (135)

A case study on new interactions.

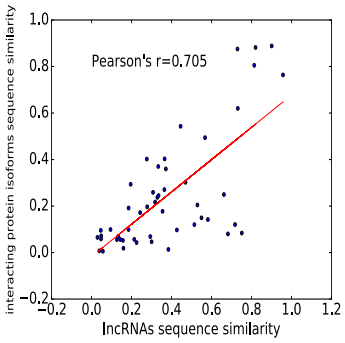
We further validate our model using some new lncRNA-protein interactions from the recent literature that were not included in the NPinter v3.0 database. After a careful

Table 3.5: Prediction results concerning 12 new lncRNA-protein interactions from recent literature.

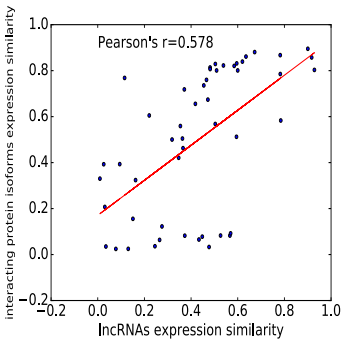
lncRNA	Protein	Machine-learning based methods							Network based methods				
		lncPro	IPMiner	lncADeep	rpiCool	RPITER	LPI-BLS	DeepLPI	LPIHN	LPBNI	LPLNP	PLIPCOM	SFPEL-LPI
PANDAR	PTBP1	x	o	o	o	o	o	o	x	x	x	x	x
lnc-SH2D7.1	FBXL22	x	o	x	x	o	x	o	o	x	x	x	x
lnc-SH2D7.1	LPIN2	o	x	o	o	x	o	o	o	o	o	o	o
lnc-DCAF811.1	PEBP1	o	x	x	x	o	o	o	o	o	o	o	o
lnc-DCAF811.1	DNAJB12	o	x	x	x	o	x	o	x	o	o	x	o
lnc-NIT1	POR	o	x	o	x	x	x	o	x	o	x	x	o
AC011498.1	CEBPA	x	x	x	x	o	x	x	x	x	x	x	x
CRNED	CEBPA	x	o	o	x	x	o	x	x	x	x	x	x
LINC00504	CEBPA	x	o	o	o	o	o	o	x	x	x	x	o
AC011498.1	NPM1	x	x	o	o	x	x	o	x	x	x	o	x
LL22NC03-N64E9.1	KLF2	x	x	x	x	o	x	x	x	x	o	o	x
LL22NC03-N64E9.1	EZH2	x	o	o	x	o	o	o	x	o	x	o	o
Recall		.333	.417	.583	.333	.666	.500	.750	.250	.417	.333	.417	.500

Note: The predicted and unpredicted interactions are represented as circles and crosses in the table.

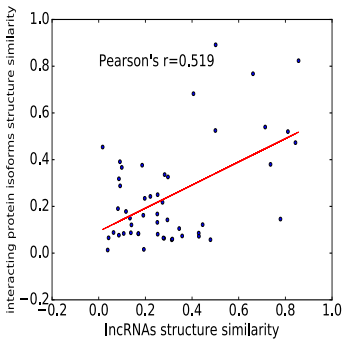
literature search, we found 12 new lncRNA-protein interactions (132; 133; 134; 135). The details of these interactions are provided in Table 3.4. The prediction results concerning these new lncRNA-protein interactions by the methods are illustrated in Table 3.5. The results show that DeepLPI was able to find out novel interactions often missed by the other methods.



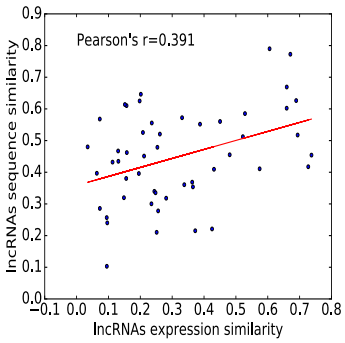
((a))



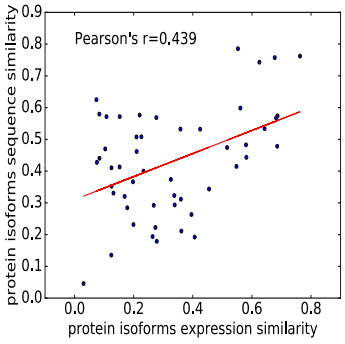
((b))



((c))



((d))



((e))

Figure 3.7: Correlation analysis.(a) Correlation between lncRNA sequence and protein isoform sequence similarities. (b) Correlation between lncRNA expression and protein isoform expression similarities (c) Correlation between lncRNAs structure and protein isoform structure similarities. (d) Correlation between lncRNAs sequence and lncRNAs expression similarities. (e) Correlation between protein isoforms sequence and protein isoforms expression similarities.

3.4 Discussion

The knowledge of interactions between lncRNAs and protein isoforms could help understand the functions of lncRNAs. In this chapter, we proposed a machine-learning based method, DeepLPI, to predict interactions between lncRNAs and protein isoforms. DeepLPI uses a multimodal deep learning neural network to extract intrinsic features from the sequence and structure data of lncRNAs and protein isoforms and a conditional random field to extract network topological features from their expression data. We designed a multiple instance learning iterative algorithm to train the prediction model using an availablelncRNA-protein interaction dataset, and performed extensive experiments to show that DeepLPI achieves a significantly better accuracy in predicting lncRNA-protein interactions compared with the state-of-the-art methods. The multimodal learning feature of DeepLPI allows it to integrate more types of data besides sequences, structures and expression profiles. With minor modifications, DeepLPI could be adapted to predict miRNA-protein interactions, as well as more complex interactions such as lncRNA-miRNA-protein interactions.

Our divergence analysis shows that many isoforms of the same gene interact with different lncRNAs. Hence, it would be of practical importance to study the interactions between lncRNAs and protein isoforms (as opposed to proteins or genes). However, as far as we know, DeepLPI is the first attempt to predict lncRNA-protein isoform interactions, and its performance is still far from being desirable. It might be possible to improve the performance of DeepLPI by using better (*e.g.*, tissue-specific) expression data, more detailed protein secondary structure representations and high quality isoform-isoform interaction

network data. We plan to investigate these directions in the near future.

3.5 Availability of data and materials

DeepLPI is implemented in Python and freely available to the public on https://github.com/dls03/DeepLPI

Chapter 4

RESmim: A deep learning method for predicting the interactions between miRNAs and isoforms

4.1 Introduction

Micro RNAs (miRNAs) are small RNA transcripts of 22 nucleotides that regulate biological processes by binding to the isoforms or messenger RNAs (mRNAs) (27). They bind to complementary sequences in isoform to create cleavages or translational repression sites. As a result, the target isoforms are prevented from producing functional peptides and proteins (28).

The mechanism of miRNA-isoform interactions can be divided into two groups, canonical interactions and non-canonical interactions (29; 30). In canonical interactions the

seed sequence of a miRNA, which is defined as the first 2-8 nucleotides starting at the $5'$ end and counting toward the $3'$ end, bind to the complementary sequence of the isoforms. In non-canonical interactions miRNAs can bind outside the seed sequence (29; 30).

Finding miRNA-isoform interactions using experimental methods is expensive and time-consuming. Hence, we need to rely on computational methods to find the interactions on a large scale. In literature, a variety of miRNA target prediction tools are proposed, but their performance is far from being desirable primarily due to the lack of labeled training data. Also, most of the tools are based on assumption that the seed sequence of a miRNA is the only imporatant feature as this interact directly with isoform. These tools perform very well to find canonical interactions, but shows worst performance in finding non-canonical interaction. However, more recent studies have indicated that we should consider the entire sequence of miRNAs and isoforms to find interactions between them (30; 31).

In literature, the computational methods can be grouped into two broad categories, rule-based methods and machine-learning based methods (136; 137). The rule-based methods derives the classifier based on important observations from biological experiments, such as thermodynamic stability of the miRNA-isoform duplex, evolutionary conservation etc. TargetScan (138), miRanda (139), PITA (140), etc are popular rule-based methods. The machine-learning based methods build a binary classifier between known interaction of miRNA-isoform pairs named as positive interactions and non-interaction pairs named as negative interactions. These methods rely on the sequence, structure and physicochemical properties of miRNAs and isoforms. The machine-learning based methods include MirTarget (141), PicTar (142), miTarget (143), etc. However, both of the rule-based and the

machine-learning based methods liaise heavily on hand-crafted features. Such hand-crafted features are time-consuming to generate and often incomplete.

Recent progress in processing power and rise of deep-learning methods shows us how we can dispatch hand-crafted features and harness multi-dimensional features from the raw data directly (144). In recent years, deep-learning based tools are proposed to the miRNAs-isoforms intearctions prediction problem and beat the performance of existing tools. MiRTDL (145) used convolutional neural networks to analyze matrices of miRNA seed. DeepTarget (146), relied on recurrent neural networks to process miRNAs and truncated segments of isoforms. DeepMirTar (147) and miRAW (137) only rely on $3'$UTR segments of isoforms.

Many challenges remain to be solved in this area. First, in the above studies, only partial sequence of miRNAs or isoforms has been considered. Second, each method has certain limitations on how they integrated sequence, structure or expression data. Unorganized integration of these data sometimes leads to the wrong prediction, which impact the performance. Another problem in the machine-learning based methods is that the same amount of negative interactions (non-interactive miRNA-isoform pair) as positive interactions (interactive miRNA-isoform pair) are sampled out from the pairs of miRNAs and isoforms. Since the number of negative interaction pool is very high, the sampling of negative interactions needs to be specific. Otherwise, the method can leads to the over-fitting issue, where it cannot predict new interactions accurately. Last but not least, all the methods suffer from a lack of ground truth interactions between miRNAs and isoforms which cause the inverse impact on performance.

In this chapter, we present RESmim, a novel miRNA-isoform interaction prediction tool that works with full genomic sequence, structure and expression data of miRNAs and isoforms. To handle longer sequence of isoforms we used residual neural network, that can capture long range dependencies in a sequence. RESmim goes through two stages. In the first stage, we trained a deep neural network (DNN) that used multimodal deep learning (MDL) (23) technique to extracts features from sequence and structure data of miRNAs and isoforms. The MDL fused these extracted features and measures the initial interaction scores between miRNAs and isoforms. In the second stage, two conditional random field (CRF) are designed to exploit the co-expression relationship between miRNAs and co-expression relationship between isoforms. The two CRFs are assigned final prediction scores of interactions between miRNAs and isoforms based on initial interaction scores while trying to keep highly co-expressed miRNAs and highly co-expressed isoforms attaining the same labels. To overcome the lack of interaction training labels of miRNAs and isoforms, we proposed an iterative semi-supervised training algorithm based on multiple instance learning (MIL) framework similar to (25; 26). In MIL, initially, we assign positive interaction labels for all miRNA and isoform pairs, given that the miRNA interact with that isoform and assign negative interaction labels for all other miRNA and isoform pairs those do not interact with each other. In each iteration, the DNN and CRFs upgrade the initial interaction scores and assign the same labels to co-expressed miRNAs and co-expressed isoforms until it reaches saturation. Under this environment, isoforms of the same protein can interact differently with the same miRNA. But since we are integrating more labeled data in terms of the miRNAs and isoforms interactions, it leads to better prediction results.

To evaluate the performance of RESmim, we first measure its prediction performance using gene level interactions with miRNAs. We make sure that negative interactions contain miRNAs and isoforms that also present in positive interactions. This helps us to overcome the outfitting issues. RESmim achieves an average AUC of 0.831 and AUPRC of 0.795 on the Human interaction dataset. Based on the availability of implementation and convenience, we compare our method with deep-learning based methods . In total, 4 methods are chosen for comparison. The results demonstrate that our method outperforms the others. We further analyze the effect of the model component, which demonstrates the effectiveness of each data source (structure, sequence, and expression) we used. We analyze the structural component to show how the model captures important features. We also demonstrate the divergence of isoform interactions which shows how isoforms from the same gene interact differently with miRNAs. Finally, we validate our method using a test case analysis on experimentally discovered miRNA-isoform interactions.

4.2 Materials and Methods

4.2.1 Datasets

A key factor for successful application of any machine-learning classification technique is access to a sufficiently variable and representative dataset that will generalize a trained model to new and unseen data. For the miRNA-isoform interaction prediction problem, this requires a comprehensive dataset of verified positive and negative interactions that encompass both canonical and non-canonical examples. While there are multiple data repositories providing information regarding experimentally validated positive interactions,

there are significantly fewer experimentally verified negative interactions.

The ground truth interactions between miRNAs and genes were downloaded from the Diana TarBase (148) and MirTarBase (149). Diana TarBase database contains experimentally verified 121,090 positive and 2,940 negative interactions. MirTarBase contains only experimentally verified 479,340 positive interactions. Following (137), we merged these two database. While merging we only consider human genome and remove inconsistent entries. This produced a final dataset of 380,512 positive and 796 negative interactions between 2,591 miRNAs and 15,046 genes. To train a neural network model, we also need to find more negative interactions. As the population of negative interactions count is large, complete random sampling of it may contain few miRNAs and genes that present in positive interactions, which might lead to overfitting (105). To reduce overfitting, we make sure that at least a half of the negative miRNA-gene interaction pairs contain miRNAs and genes that appear in positive interaction pairs (but do not interact with each other). The rest of the negative interaction pairs consist of randomly chosen miRNAs and genes that do not appear in positive interaction pairs.

The miRNA sequences and the mRNA isoform sequences of human genome were downloaded from GENCODE (106) and ENSEMBL (107), respectively. The sequences were then used to predict their secondary structures. To predict the secondary structure of miRNA and isoform, we used RNAShapes (108). The output of RNAShapes was converted to a structure sequence using the EDeN tool (http://github.com/fabriziocosta/EDeN) as in (109). The miRNA and mRNA isoform structure typically consists of six structural components: stem (S), multiloop (M), hairpin loop (H), internal loop (I), dangling end (T), and dangling

start (F).

The third type of data that we collected are miRNA and isoform expression data. The miRNA expression data are obtained from the dbDEMC (150), and the isoform expression data were downloaded from the literature (32). The miRNA and isoform expression data are based on high-throughput human RNA sequencing experiments. We used the expression data to build co-expression networks. Since different databases use different identifier naming conventions to record miRNA and isoform, ID conversion tools from (112; 113; 111) were used to identify the same moleculars from different data sources and perform the mapping between isoforms. Finally, we filtered the data and kept the miRNAs and isoform that appear in both the sequence data and the expression data.

4.2.2 Methods

RESmim predicts the interactions between miRNAs and isoforms by integrating the information of sequence, structure and expression data into a unified predictive model. It consists of two learning submodels. The first is a multimodal deep learning neural network (MDLNN) model and the second a conditional random field (CRF) model. The MDLNN model extracts and fuses the (intrinsic) features from the sequence and structure data of miRNAs and isoforms, and calculates the initial scores of the interactions between miRNAs and isoforms. The CRF model makes a final prediction based on both the initial interaction scores and the expression data of isoforms and miRNAs. To overcome the lack of ground truth interactions between miRNAs and isoforms, we develop a semi-supervised algorithm following (24; 25; 26) to train the MDLNN and CRF models together iteratively. Figure 4.1 shows a schematic illustration of RESmim. More details of the method are described in

the following subsections.

Exploring sequence and structure features using multimodal deep learning neural network.

To learn intrinsic features related to miRNA-isoform interactions from the sequence and structure data, we construct a multimodal deep learning neural network (MDLNN). At first, MDLNN uses a standard dense embedding technique (115) to map the sequences of miRNAs and isoforms into a 64-dimensional vector space, which is implemented by using embedding layers of Keras (116). After a training process, the embedding layers are able to learn appropriate mappings such that the mapped dense vectors could capture similarities between the sequences.

Because of the longer length isoform sequence compare to miRNA, we used different convolution layers structure to processes these sequence. Particularly, we use convolutional layers with residual blocks (ConvolutionRB) to extract long-range dependencies in isoforms. Details of the convolutional layers with residual blocks are depicted in Figure 4.2 and Figure 4.3. Compare to that, for miRNAs we used only 2 convolutional filters to obtain the local features of the sequence and structure data.

After that, max pooling (denoted as pool($\cdot$)) layers are used to downsample the output of the convolutional layers to reduce the learning time of the subsequent layers.

These features extracted from miRNA sequences, miRNA structures, isoform sequences and isoform structures are merged together as the input of a convolutional layer

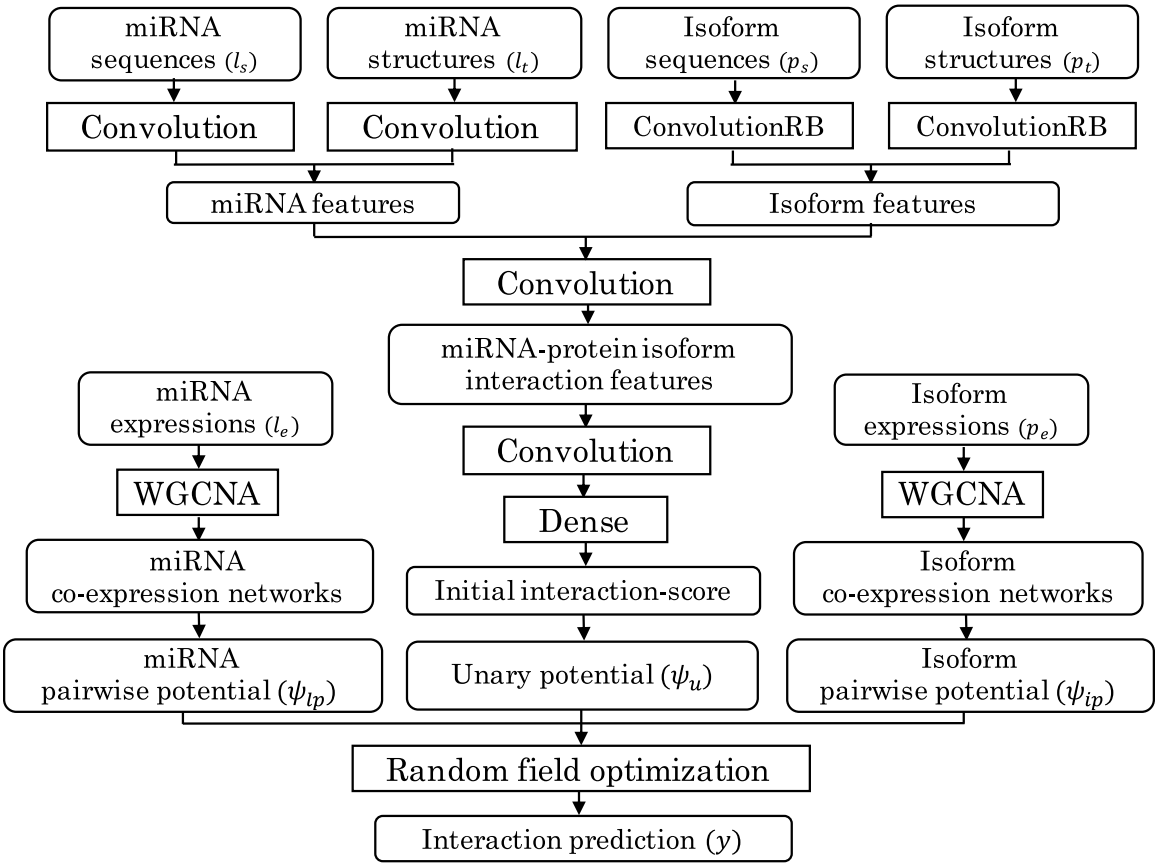


Figure 4.1: A flowchart of RESmim. It begins with a multimodal deep learning neural network (MDLNN) that uses embedding layers, convolutional layers, convolutional layers with residual blocks and other layers of Keras to extract features from the sequence and structure data of miRNAs and isoforms, and calculate initial interaction scores. Weighted correlation network analysis (WGCNA) is used to construct co-expression networks from expression data of miRNAs and isoforms. Based on the pairwise potentials and unary potentials inferred from the co-expression relationship and the initial interaction scores, respectively, a conditional random field (CRF) optimization is used to predict the interactions between miRNAs and isoforms. The whole model is trained using an iterative semi-supervised learning algorithm based on multiple instance learning (MIL).

followed by a dense layer. The convolutional layer and dense layer are intended to learn the interaction patterns between miRNAs and isoforms. Finally, the output of the dense layer is fed into a logistic regression layer to compute an initial interaction score. Given a miRNA sequence l_s, an isoform sequence p_s, and the predicted structures l_t and p_t of the

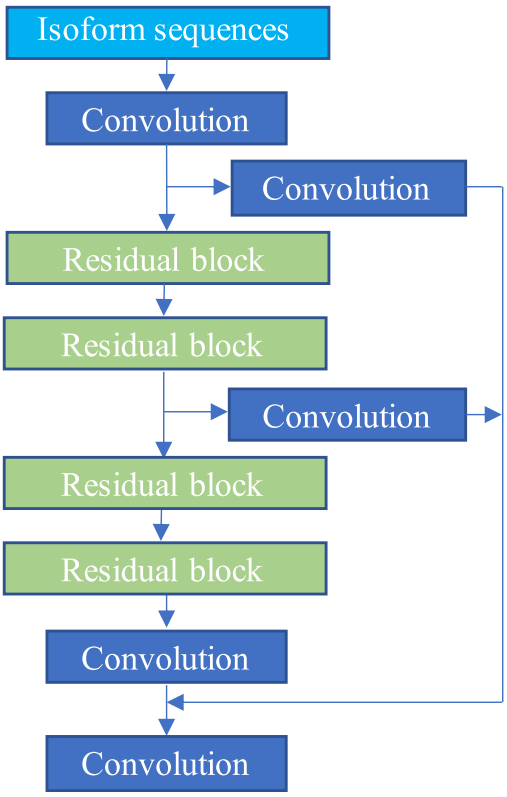


Figure 4.2: A flowchart of the convolutional layers with residual blocks, which consist of 4 stacked residual blocks connecting the input layer to the final convolution layer also known as penultimate layer. These residual blocks are stacked in a way that the output of previous residual block is connected to the input of the next residual block. Further, the output of every second residual block is added to the input of the penultimate layer.

miRNA sequence and isoform sequence, respectively, the initial interaction score (IIS) is calculated following (24).

To capture the co-expression relationships between the isoforms and between the miRNAs, we use a weighted correlation network analysis (WGCNA) (120) to construct a co-expression network. Further we followed (24) to incorporate a conditional random field (CRF) optimization to predict the interactions between miRNAs and isoforms.

Due to the lack of ground truth miRNA-isoform interactions, conventional su-

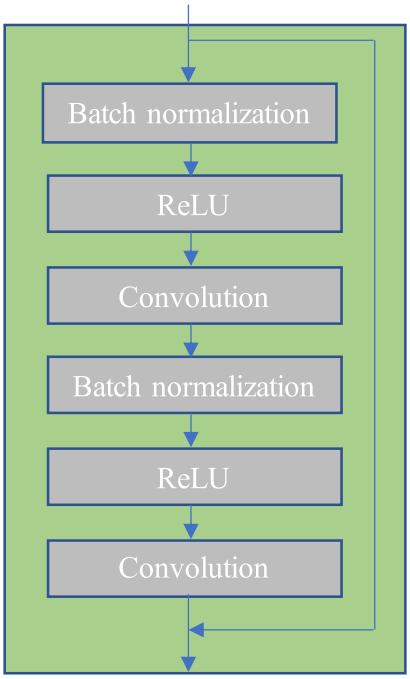


Figure 4.3: A flowchart of the residual block, which consists of batch normalization layers, rectified linear units (ReLU), and convolutional layer.

pervised training algorithms cannot be directly applied to our model. Hence, we adopt a semi-supervised training algorithm under the MIL framework as in (25; 26). In the MIL framework, for each miRNA, a gene is treated as a bag, the isoforms of a gene are treated as the instances in the bag, and only the ground truth of the bag (*i.e.*, the true miRNA-gene interaction label) is assumed. We further require that a positive bag should contain at least one positive instance and a negative bag should contain no positive instances. RESmim first initializes all instances of positive bags with positive labels, and the other instances with negative labels. Then, the model parameters are optimized with the initial labels in the following standard supervised learning manner. We use the Nadam optimization algorithm to update the MDLNN parameters. We use the L-BFGS-B algorithm as in (24) to update CRF parameters. The parameter update step and the label update step are repeated

alternately until convergence.

4.3 Results and Validation

In this section, we first compare the performance of RESmim with some state-of-the-art methods and analyze the effectiveness of our method in terms of each type of data we used and each component of the model. Next, we validate the prediction results of RESmim using experimentally verified test case analyses.

4.3.1 Prediction of miRNA-isoform interactions

We compare the performance of RESmim with machine-learning based methods. Then, we evaluate the effectiveness of each component of our model (*i.e.*, the ablation study). We also study the divergence of miRNAs interacting with different isoforms of the same protein/gene, and compare the structural components of miRNAs and isoforms in both interactive and non-interactive pairs.

Prediction performance comparison between RESmim and the existing methods.

Since there is no benchmark data for miRNA-isoform interactions, we could only evaluate the performance of RESmim based on the benchmark data of (human) miRNA-gene interactions. We compare RESmim with state-of-the-art machine-learning based based methods, e.g., deepTarget (146), MiRTDL (145), DeepMirTar (147), miRAW (137). Table 4.1 shows the average test results in 10 runs of five-fold cross validations on human interaction dataset. The AUC values of deepTarget, MiRTDL, DeepMirTar, miRAW, and

Table 4.1: Comparison of prediction performance of RESmim.

	All interactions		Canonical interactions		Non-canonical interactions	
Methods	AUC	AUPRC	AUC	AUPRC	AUC	AUPRC
deepTarget	0.774	0.721	0.801	0.746	0.707	0.657
MiRTDL	0.802	0.755	0.834	0.786	0.721	0.677
DeepMirTar	0.814	0.761	0.824	0.779	0.791	0.742
miRAW	0.823	0.787	0.827	0.792	0.816	0.769
RESmim	0.831	0.795	0.831	0.787	0.835	0.816

RESmim are 0.774, 0.802, 0.814, 0.823 and 0.831, respectively, and their AUPRC values are 0.721, 0.755, 0.761, 0.787 and 0.795, respectively. RESmim outperformed these machine-learning based methods by 7.4%, 3.6%, 2.1%, and 0.9% in terms of AUC and by 1.03%, 0.5%, 0.4%, and 0.1% in terms of AUPRC, respectively.

We further analyse the performance of these methods on canonical and non-canonical interactions. In contrast to all other methods RESmim performs better in non-canonical interactions compare to canonical interactions (see Table 4.1).

Analyzing the effects of model components.

To evaluate the contribution of the key components and biological features used in our model, we perform an ablation study by removing various components/features from the model and measure how the performance of the model would be affected. More specifically, we test how the model is affected when the MIL learning with isoforms is replaced by conventional learning with proteins, when the CRF component along with the expression data are removed, and when the sequence or structure data are removed.

Figure 4.4 shows that the average AUC of RESmim dropped 2.7%, 1.8%, 1.6%,

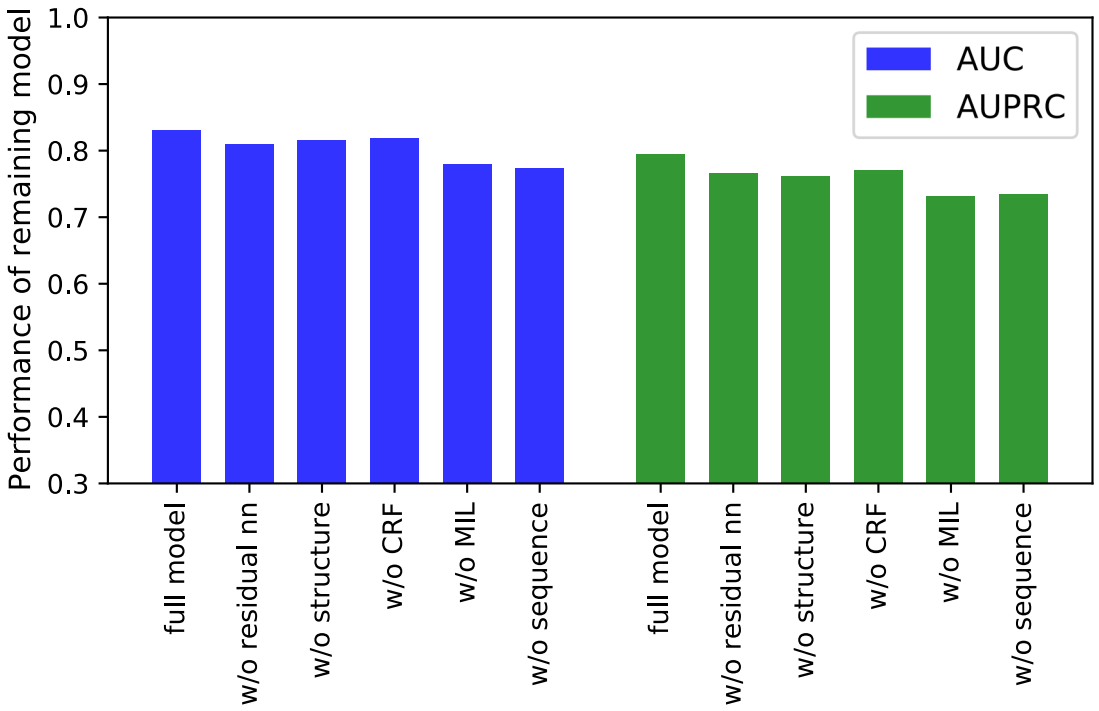


Figure 4.4: The effect on performance of removing various components from the model. The average AUC and AUPRC values of RESmim, RESmim without residual neural network, RESmim without structure data, RESmim without CRF, RESmim without using MIL/isoforms, and RESmim without sequence data are shown in the figure.

6.7%, and 7.5% without the residual neural network, without the structure data, without the CRF component for incorporating expression data, without the MIL learning framework for incorporating isoforms, and without the sequence data, respectively. Without these components or data, the performance in term of AUPRC shows a similar declining trend with the percentage decreases being 3.8%, 4.3%, 3.1%, 8.6% and 8.3%, respectively. The results suggest that the expression data and the sequences data are most effective in improving the prediction performance of the model. Although the usage of structure data did not boost the performance of the model significantly, it allows us to observe interesting enrichment of structural components in interactive miRNAs, as discussed in the next subsection.

Structural components at important positions in interactive and non-interactive pairs.

It would be interesting to study how the structural components of miRNAs and isoforms are distributed in interactive pairs, especially around their interacting sites, and what structural components may contribute more to the interactions than the others. For each miRNA-isoform pair, we use saliency maps (126) to compute importance weights at each position in both sequences. These weights indicate how a position might impact the prediction outcome by our model (*i.e.*, interactive or non-interactive). The miRNA and isoform structural components at heavily weighted (*i.e.*, important) positions of interactive and non-interactive pairs are profiled and shown in Figure 4.5. For each structural component, the average occurrence frequency across all instances is calculated. We can see that at important miRNA and isoform positions, hairpin loops (H) and internal loops (I) occur much more often in interactive pairs than in non-interactive pairs. The ratio of dangling start (F) positions between miRNA and isoform is much higher in the positive interactions compare to the negative interactions. On the other hand, stems (S) occur much often at the non-interactive pairs than the interactive pairs. These suggest that open/unpaired miRNA and isoform positions perhaps play more important roles in their interactions with interactions, and is consistent with several studies in the literature (151; 127; 74).

Divergence of miRNAs interacting with the isoforms of the same gene.

Our ultimate goal is to find miRNA interactions at the isoform level. Hence, it would be useful to analyze how miRNAs interact divergently with the isoforms of the same

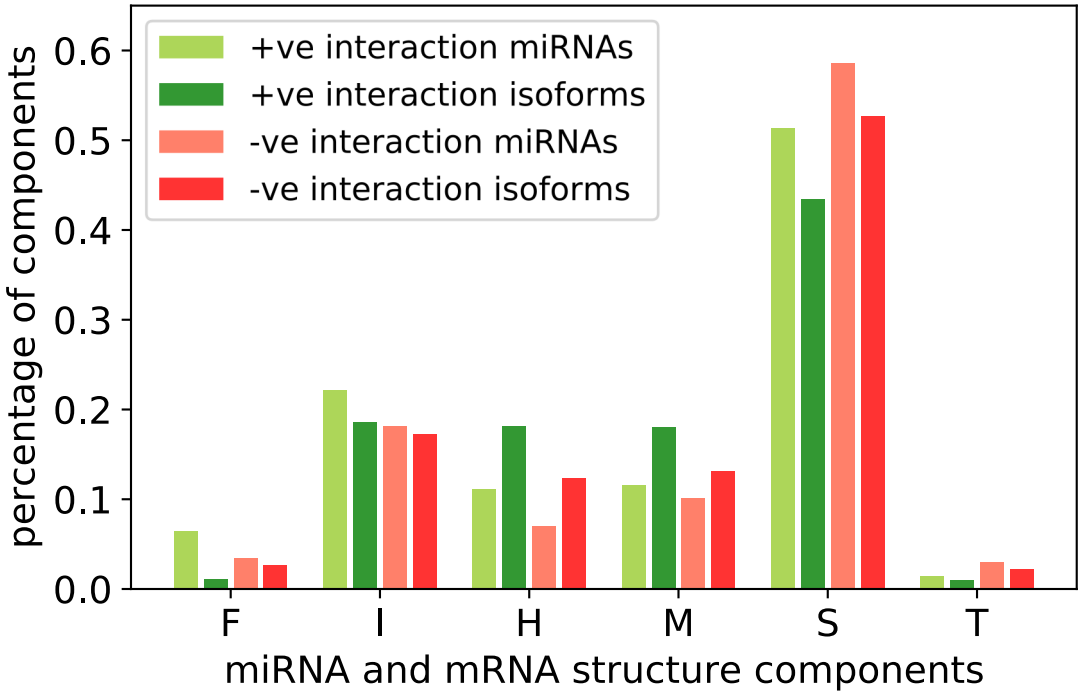


Figure 4.5: Distribution of miRNA and isoform structural components.

gene. We estimated the similarity of predicted miRNA interactions for each pair of isoforms in terms of the semantic similarity score using GOssTo (128). The semantic dissimilarity score of two isoforms is then defined as one minus their similarity score. We considered only multiple isoforms genes (MIGs) and collected all interactions between miRNAs and the isoforms of the MIGs as predicted by RESmim. For each MIG, the interaction divergence of its isoforms was calculated by averaging the semantic dissimilarity scores of all pairs of its isoforms. The dissimilarity score distributions for MIPs that have divergent isoform interactions are shown in Figure 4.6 where the mean score value is 0.256.

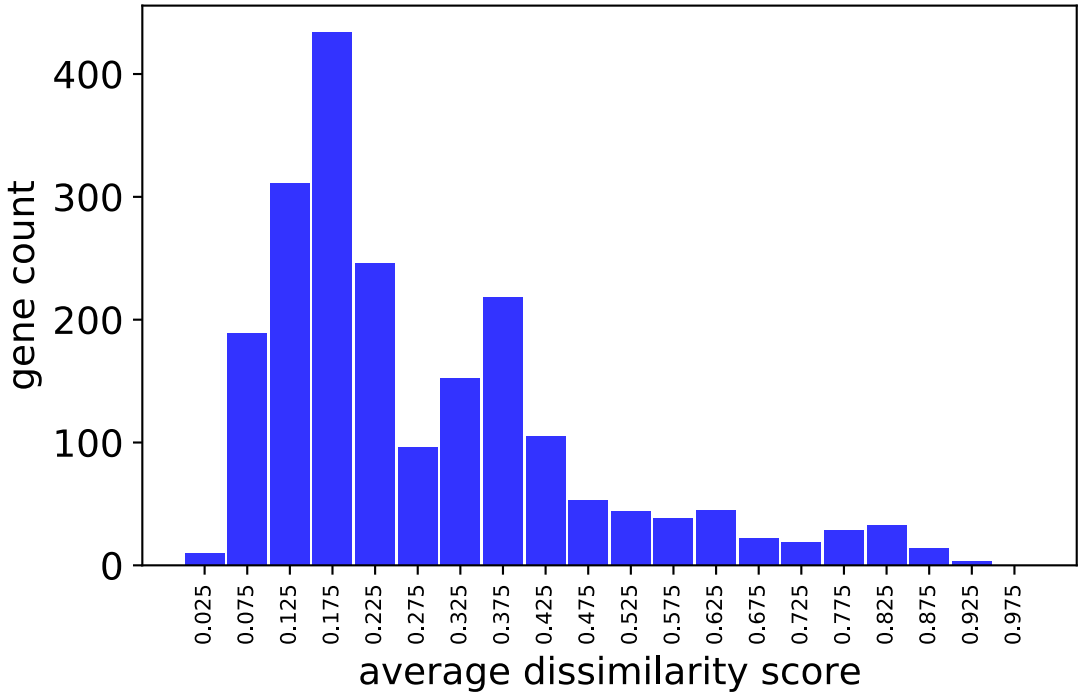


Figure 4.6: Distributions of semantic dissimilarity scores of MIGs. For each MIG, the semantic dissimilarity score indicates the divergence of the miRNAs interacting with its different isoforms. The score range of [0, 1] is equally divided into 20 bins. For each bin, we count how many MIGs have semantic dissimilarity scores in this range.

Case study on interactions.

As mentioned before, there has been little systematic study on miRNAs-isoforms interactions in the literature, and not many specific experimentally-verified functions of isoforms have been reported. After a careful literature search, we found 5 MIGs consist of 18 isoforms, where the isoforms of a gene interact differently with a miRNA. Details of the prediction result given in Table 4.2.

Table 4.2: Comparison of prediction performance of RESmim.

miRNAs	Genes	Isoforms	Varified Interaction	RESmim Prediction
hsa-mir-128b	ARPP-21	NM_016300	1	1
		NM_001025069	0	0
hsa-mir-361	CHM	NM_000390	1	1
		NM_001037312	0	0
hsa-mir-615	HOXC4	NM_014620	1	1
		NM_004503	0	0
hsa-mir-628	CCPG1	NM_020739	1	1
		NM_001033559	0	1
		NM_001033560	0	0
		NM_130810	0	0
hsa-mir-661	PLEC1	NM_000445	1	1
		NM_201378	1	0
		NM_201379	1	1
		NM_201380	1	1
		NM_201381	0	1
		NM_201382	0	1
		NM_201383	0	1
		NM_201384	0	0

4.4 Discussion

Finding the interactions between miRNAs and isoforms should be the ultimate goal in the process of understand how miRNAs works. In this project, we propose a machine-learning based method, RESmim, to predict the interactions between miRNAs and isoforms. The idea here to integrate multimodal data, that not only used intrinsic features from sequence and structure information but also used the topological features from expression data. Eventually, RESmim beats all the existing methods in terms of performance comparison. Analysis and validation results also show different aspects of these findings.

However, the performance of RESmim could be further improved in several aspects. First, the structure data derived for miRNA and isoform is used a prediction tool. Real

structure data is much more complicated and time-consuming to generate. In the future, if we can develop a method that gives a better-predicted structure data, RESmim model can perform better. Second, we cannot directly use the PPI network at isoform level since the databases for this purpose is still under development. We believe, PPI network at isoform level certaintly boost the performance of our model.

The modality learning feature gives the model a scalable property which we belive can be used to integrate different source of data other than sequence and structure. Furthermore we believe that, our model could be used to identify more complex interaction, e.g., lncRNA-miRNA-protein interaction.

Chapter 5

Conclusions

Modern sequencing techniques has led to an unrivaled explosion in the amount of genomic data. These data create new opportunities in the field of bioinformatics. At the same time, due to the significant improvement of computational power deep learning has become one of the most successful machine learning algorithms in recent years. Biological data analysis with deep learning is the key to solve many hard computational problem in the field of bioinformatics.

In this dissertation, we addressed three problems that are related to isoforms. We used deep learning techniques to process isoform level genomic information, e.g., sequences, structures, expression data. Our proposed tools used multiple instance learning to handle to scarcity of labeled training data. In particular, the first method, DeepIsoFun, predict the isoform functions. It combines multiple instance learning with domain adaptation. The latter technique helps to transfer the knowledge of gene functions to the prediction of isoform functions and provides additional labeled training data. In our second study,

we propose DeepLPI, which predict the interactions between long non-coding RNAs and protein isoforms. It combines heterogeneous data using a hybrid framework by integrating a deep neural network and a conditional random field. Finally, we proposed RESmim, to predict interactions between microRNAs and isoforms using a residual deep neural network.

Bibliography

[1] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[2] N. L. Bray, H. Pimentel, P. Melsted, and L. Pachter, "Near-optimal probabilistic rna-seq quantification," *Nature biotechnology*, vol. 34, no. 5, pp. 525–527, 2016.

[3] R. Eksi, H.-D. Li, R. Menon, Y. Wen, G. S. Omenn, M. Kretzler, and Y. Guan, "Systematically differentiating functions for alternatively spliced isoforms through integrating rna-seq data," *PLoS computational biology*, vol. 9, no. 11, p. e1003314, 2013.

[4] W. Li, S. Kang, C.-C. Liu, S. Zhang, Y. Shi, Y. Liu, and X. J. Zhou, "High-resolution functional annotation of human transcriptome: predicting isoform functions by a novel multiple instance-based label propagation method," *Nucleic acids research*, vol. 42, no. 6, pp. e39–e39, 2013.

[5] Q. Pan, O. Shai, L. J. Lee, B. J. Frey, and B. J. Blencowe, "Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing," *Nature genetics*, vol. 40, no. 12, pp. 1413–1415, 2008.

[6] E. T. Wang, R. Sandberg, S. Luo, I. Khrebtukova, L. Zhang, C. Mayr, S. F. Kingsmore, G. P. Schroth, and C. B. Burge, "Alternative isoform regulation in human tissue transcriptomes," *Nature*, vol. 456, no. 7221, pp. 470–476, 2008.

[7] L. Gallego-Paez, M. Bordone, A. Leote, N. Saraiva-Agostinho, M. Ascensão-Ferreira, and N. Barbosa-Morais, "Alternative splicing: the pledge, the turn, and the prestige," *Human Genetics*, pp. 1–28, 2017.

[8] D. Himeji, T. Horiuchi, H. Tsukamoto, K. Hayashi, T. Watanabe, and M. Harada, "Characterization of caspase-8l: a novel isoform of caspase-8 that behaves as an inhibitor of the caspase cascade," *Blood*, vol. 99, no. 11, pp. 4070–4078, 2002.

[9] E. Melamud and J. Moult, "Stochastic noise in splicing machinery," *Nucleic acids research*, vol. 37, no. 14, pp. 4873–4886, 2009.

[10] K. F. Mittendorf, C. L. Deatherage, M. D. Ohi, and C. R. Sanders, "Tailoring of membrane proteins by alternative splicing of pre-mrna," *Biochemistry*, vol. 51, no. 28, p. 5541, 2012.

[11] J. Tazi, N. Bakkour, and S. Stamm, "Alternative splicing and disease," *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, vol. 1792, no. 1, pp. 14–26, 2009.

[12] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya, "Hierarchical multi-label prediction of gene function," *Bioinformatics*, vol. 22, no. 7, pp. 830–836, 2006.

[13] H. Mi, A. Muruganujan, and P. D. Thomas, "Panther in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees," *Nucleic acids research*, vol. 41, no. D1, pp. D377–D386, 2012.

[14] L. Schietgat, C. Vens, J. Struyf, H. Blockeel, D. Kocev, and S. Džeroski, "Predicting gene function using hierarchical multi-label decision tree ensembles," *BMC bioinformatics*, vol. 11, no. 1, p. 2, 2010.

[15] A. Vinayagam, R. König, J. Moormann, F. Schubert, R. Eils, K.-H. Glatting, and S. Suhai, "Applying support vector machines for gene ontology based gene function prediction," *BMC bioinformatics*, vol. 5, no. 1, p. 116, 2004.

[16] J. Yang, R. Yan, A. Roy, D. Xu, J. Poisson, and Y. Zhang, "The i-tasser suite: protein structure and function prediction," *Nature methods*, vol. 12, no. 1, pp. 7–8, 2015.

[17] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, *et al.*, "Gene ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[18] D. Barrell, E. Dimmer, R. P. Huntley, D. Binns, C. O'donovan, and R. Apweiler, "The goa database in 2009—an integrated gene ontology annotation resource," *Nucleic acids research*, vol. 37, no. suppl_1, pp. D396–D403, 2008.

[19] X. Pan, Y.-X. Fan, J. Yan, and H.-B. Shen, "Ipminer: hidden ncrna-protein interaction sequential pattern mining with stacked autoencoder for accurate computational prediction," *Bmc Genomics*, vol. 17, no. 1, p. 582, 2016.

[20] H.-C. Yi, Z.-H. You, D.-S. Huang, X. Li, T.-H. Jiang, and L.-P. Li, "A deep learning framework for robust and accurate prediction of ncrna-protein interactions using evolutionary information," *Molecular Therapy-Nucleic Acids*, vol. 11, pp. 337–344, 2018.

[21] C. Peng, S. Han, H. Zhang, and Y. Li, "Rpiter: A hierarchical deep learning framework for ncrna–protein interaction prediction," *International journal of molecular sciences*, vol. 20, no. 5, p. 1070, 2019.

[22] C. Yang, L. Yang, M. Zhou, H. Xie, C. Zhang, M. D. Wang, and H. Zhu, "Lncadeep: an ab initio lncrna identification and functional annotation tool based on deep learning," *Bioinformatics*, vol. 34, no. 22, pp. 3825–3834, 2018.

[23] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 689–696, 2011.

[24] H. Chen, D. Shaw, J. Zeng, D. Bu, and T. Jiang, "Diffuse: predicting isoform functions from sequences and expression profiles via deep learning," *Bioinformatics*, vol. 35, no. 14, pp. i284–i294, 2019.

[25] S. Andrews, T. Hofmann, and I. Tsochantaridis, "Multiple instance learning with generalized support vector machines," in *AAAI/IAAI*, pp. 943–944, 2002.

[26] X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu, "Revisiting multiple instance neural networks," *Pattern Recognition*, vol. 74, pp. 15–24, 2018.

[27] D. P. Bartel, "Micrornas: genomics, biogenesis, mechanism, and function," *cell*, vol. 116, no. 2, pp. 281–297, 2004.

[28] M. Thomas, J. Lieberman, and A. Lal, "Desperately seeking microrna targets," *Nature structural & molecular biology*, vol. 17, no. 10, p. 1169, 2010.

[29] H. Seok, J. Ham, E.-S. Jang, and S. W. Chi, "Microrna target recognition: insights from transcriptome-wide non-canonical interactions," *Molecules and cells*, vol. 39, no. 5, p. 375, 2016.

[30] C. J. Stavast and S. J. Erkeland, "The non-canonical aspects of micrornas: Many roads to gene regulation," *Cells*, vol. 8, no. 11, p. 1465, 2019.

[31] N. Cloonan, "Re-thinking mirna-mrna interactions: Intertwining issues confound target discovery," *Bioessays*, vol. 37, no. 4, pp. 379–388, 2015.

[32] D. Shaw, H. Chen, and T. Jiang, "Deepisofun: a deep domain adaptation approach to predict isoform functions," *Bioinformatics*, 2018.

[33] D. Shaw, H. Chen, X. Minzhu, and T. Jiang, "Deeplpi: a multimodal deep learning method for predicting the interactions between lncrnas and protein isoforms," *BMC Bioinformatics*, 2020.

[34] Á. V. Vázquez, M. Blanco, J. Zaborowska, P. Soengas, M. I. González-Siso, M. Becerra, E. Rodríguez-Belmonte, and M. E. Cerdán, "Two proteins with different functions are derived from the klhem13 gene," *Eukaryotic cell*, vol. 10, no. 10, pp. 1331–1339, 2011.

[35] S. Gueroussov, T. Gonatopoulos-Pournatzis, M. Irimia, B. Raj, Z.-Y. Lin, A.-C. Gingras, and B. J. Blencowe, "An alternative splicing event amplifies evolutionary differences between vertebrates," *Science*, vol. 349, no. 6250, pp. 868–873, 2015.

[36] T. Revil, J. Toutant, L. Shkreta, D. Garneau, P. Cloutier, and B. Chabot, "Protein kinase c-dependent control of bcl-x alternative splicing," *Molecular and cellular biology*, vol. 27, no. 24, pp. 8431–8441, 2007.

[37] F. Végran, R. Boidot, C. Oudin, J.-M. Riedinger, F. Bonnetain, and S. Lizard-Nacol, "Overexpression of caspase-3s splice variant in locally advanced breast carcinoma is associated with poor response to neoadjuvant chemotherapy," *Clinical cancer research*, vol. 12, no. 19, pp. 5794–5800, 2006.

[38] P. Bouillet and L. A. O'reilly, "Cd95, bim and t cell homeostasis," *Nature Reviews Immunology*, vol. 9, no. 7, pp. 514–519, 2009.

[39] S. Mazurek, C. B. Boschek, F. Hugo, and E. Eigenbrodt, "Pyruvate kinase type m2 and its role in tumor growth and spreading," in *Seminars in cancer biology*, vol. 15, pp. 300–308, Elsevier, 2005.

[40] J. Oberwinkler, A. Lis, K. M. Giehl, V. Flockerzi, and S. E. Philipp, "Alternative splicing switches the divalent cation selectivity of trpm3 channels," *Journal of Biological Chemistry*, vol. 280, no. 23, pp. 22540–22548, 2005.

[41] J. K. Pickrell, A. A. Pai, Y. Gilad, and J. K. Pritchard, "Noisy splicing drives mrna isoform diversity in human cells," *PLoS genetics*, vol. 6, no. 12, p. e1001236, 2010.

[42] H.-D. Li, G. S. Omenn, and Y. Guan, "Misomine: a genome-scale high-resolution data portal of expression, function and networks at the splice isoform level in the mouse," *Database*, vol. 2015, p. bav045, 2015.

[43] T. Luo, W. Zhang, S. Qiu, Y. Yang, D. Yi, G. Wang, J. Ye, and J. Wang, "Functional annotation of human protein coding isoforms via non-convex multi-instance learning," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 345–354, ACM, 2017.

[44] B. Panwar, R. Menon, R. Eksi, H.-D. Li, G. S. Omenn, and Y. Guan, "Genome-wide functional annotation of human protein-coding splice variants using multiple instance learning," *Journal of proteome research*, vol. 15, no. 6, pp. 1747–1753, 2016.

[45] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1, pp. 31–71, 1997.

[46] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in neural information processing systems*, pp. 577–584, 2003.

[47] J. Wang, L. Cai, and X. Zhao, "Multiple-instance learning via an rbf kernel-based extreme learning machine," *Journal of Intelligent Systems*, vol. 26, no. 1, pp. 185–195, 2017.

[48] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, pp. 1180–1189, 2015.

[49] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning*, pp. 97–105, 2015.

[50] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[51] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[52] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, "Domain-adversarial neural networks," *arXiv preprint arXiv:1412.4446*, 2014.

[53] X. Wang, Z. Zhu, C. Yao, and X. Bai, "Relaxed multiple-instance svm with application to object discovery," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1224–1232, 2015.

[54] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675–678, ACM, 2014.

[55] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems*, pp. 2546–2554, 2011.

[56] J. S. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[57] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, pp. 2951–2959, 2012.

[58] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, pp. 1139–1147, 2013.

[59] K. D. Pruitt, T. Tatusova, and D. R. Maglott, "Ncbi reference sequence (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins," *Nucleic acids research*, vol. 33, no. suppl_1, pp. D501–D504, 2005.

[60] R. Leinonen, H. Sugawara, and M. Shumway, "The sequence read archive," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D19–D21, 2010.

[61] H. J. Pimentel, N. Bray, S. Puente, P. Melsted, and L. Pachter, "Differential analysis of rna-seq incorporating quantification uncertainty. biorxiv," 2016.

[62] H. Caniza, A. E. Romero, S. Heron, H. Yang, A. Devoto, M. Frasca, M. Mesiti, G. Valentini, and A. Paccanaro, "Gossto: a stand-alone application and a web tool for calculating semantic similarities on the gene ontology," *Bioinformatics*, vol. 30, no. 15, pp. 2235–2236, 2014.

[63] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, p. e0118432, 2015.

[64] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[65] C. E. Metz, "Basic principles of roc analysis," in *Seminars in nuclear medicine*, vol. 8, pp. 283–298, Elsevier, 1978.

[66] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, ACM, 2006.

[67] C. Pesquita, D. Faria, H. Bastos, A. Falcao, and F. Couto, "Evaluating go-based semantic similarity measures," in *Proc. 10th Annual Bio-Ontologies Meeting*, vol. 37, p. 38, 2007.

[68] A. Schlicker, F. S. Domingues, J. Rahnenführer, and T. Lengauer, "A new measure for functional similarity of gene products based on gene ontology," *BMC bioinformatics*, vol. 7, no. 1, p. 302, 2006.

[69] X.-S. Wei, J. Wu, and Z.-H. Zhou, "Scalable multi-instance learning," in *Data Mining (ICDM), 2014 IEEE International Conference on*, pp. 1037–1042, IEEE, 2014.

[70] X.-S. Wei, J. Wu, and Z.-H. Zhou, "Scalable algorithms for multi-instance learning," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 4, pp. 975–987, 2017.

[71] J. E. Wilusz, H. Sunwoo, and D. L. Spector, "Long noncoding rnas: functional surprises from the rna world," *Genes Dev*, vol. 23, no. 13, pp. 1494–504, 2009.

[72] A. F. Palazzo and E. S. Lee, "Non-coding rna: what is functional and what is junk?," *Frontiers in genetics*, vol. 6, p. 2, 2015.

[73] H. Zhang, Y. Liang, S. Han, C. Peng, and Y. Li, "Long noncoding rna and protein interactions: From experimental results to computational models based on network methods," *International journal of molecular sciences*, vol. 20, no. 6, p. 1284, 2019.

[74] A. R. Gawronski, M. Uhl, Y. Zhang, Y.-Y. Lin, Y. S. Niknafs, V. R. Ramnarine, R. Malik, F. Feng, A. M. Chinnaiyan, C. C. Collins, *et al.*, "Mechrna: prediction of lncrna mechanisms from rna–rna and rna–protein interactions," *Bioinformatics*, vol. 34, no. 18, pp. 3101–3110, 2018.

[75] F. Kopp and J. T. Mendell, "Functional classification and experimental dissection of long noncoding rnas," *Cell*, vol. 172, no. 3, pp. 393–407, 2018.

[76] M. Hafner, M. Landthaler, L. Burger, M. Khorshid, J. Hausser, P. Berninger, A. Rothballer, M. Ascano, Jr, A.-C. Jungkamp, M. Munschauer, A. Ulrich, G. S. Wardle,

S. Dewell, M. Zavolan, and T. Tuschl, "Transcriptome-wide identification of rna-binding protein and microrna target sites by par-clip," *Cell*, vol. 141, pp. 129–41, Apr 2010.

[77] D. Ray, H. Kazan, E. T. Chan, L. Peña Castillo, S. Chaudhry, S. Talukder, B. J. Blencowe, Q. Morris, and T. R. Hughes, "Rapid and systematic analysis of the rna recognition specificities of rna-binding proteins," *Nat Biotechnol*, vol. 27, pp. 667–70, Jul 2009.

[78] D. D. Licatalosi, A. Mele, J. J. Fak, J. Ule, M. Kayikci, S. W. Chi, T. A. Clark, A. C. Schweitzer, J. E. Blume, X. Wang, J. C. Darnell, and R. B. Darnell, "Hits-clip yields genome-wide insights into brain alternative rna processing," *Nature*, vol. 456, pp. 464–9, Nov 2008.

[79] J. D. Keene, J. M. Komisarow, and M. B. Friedersdorf, "Rip-chip: the isolation and identification of mrnas, microrna and protein components of ribonucleoprotein complexes from cell extracts," *Nat Protoc*, vol. 1, no. 1, pp. 302–7, 2006.

[80] U. K. Muppirala, V. G. Honavar, and D. Dobbs, "Predicting rna-protein interactions using only sequence information," *BMC bioinformatics*, vol. 12, no. 1, p. 489, 2011.

[81] M. Bellucci, F. Agostini, M. Masin, and G. G. Tartaglia, "Predicting protein associations with long noncoding rnas," *Nature methods*, vol. 8, no. 6, p. 444, 2011.

[82] Y. Wang, X. Chen, Z.-P. Liu, Q. Huang, Y. Wang, D. Xu, X.-S. Zhang, R. Chen, and L. Chen, "De novo prediction of rna–protein interactions from sequence information," *Molecular BioSystems*, vol. 9, no. 1, pp. 133–142, 2013.

[83] Q. Lu, S. Ren, M. Lu, Y. Zhang, D. Zhu, X. Zhang, and T. Li, "Computational prediction of associations between long non-coding rnas and proteins," *BMC genomics*, vol. 14, no. 1, p. 651, 2013.

[84] V. Suresh, L. Liu, D. Adjeroh, and X. Zhou, "Rpi-pred: predicting ncrna-protein interaction using sequence and structural information," *Nucleic acids research*, vol. 43, no. 3, pp. 1370–1379, 2015.

[85] M. Akbaripour-Elahabad, J. Zahiri, R. Rafeh, M. Eslami, and M. Azari, "rpicool: A tool for in silico rna–protein interaction detection using random forest," *Journal of theoretical biology*, vol. 402, pp. 1–8, 2016.

[86] X.-N. Fan and S.-W. Zhang, "Lpi-bls: Predicting lncrna–protein interactions with a broad learning system-based stacked ensemble classifier," *Neurocomputing*, 2019.

[87] H.-C. Yi, Z.-H. You, M.-N. Wang, Z.-H. Guo, Y.-B. Wang, and J.-R. Zhou, "Rpi-se: a stacking ensemble learning framework for ncrna-protein interactions prediction using sequence information," *BMC bioinformatics*, vol. 21, no. 1, pp. 1–10, 2020.

[88] L. Wang, X. Yan, M.-L. Liu, K.-J. Song, X.-F. Sun, and W.-W. Pan, "Prediction of rna-protein interactions by combining deep convolutional neural network with feature selection ensemble method," *Journal of theoretical biology*, vol. 461, pp. 230–238, 2019.

[89] Z.-H. Zhan, L.-N. Jia, Y. Zhou, L.-P. Li, and H.-C. Yi, "Bgfe: a deep learning model for ncrna-protein interaction predictions based on improved sequence information," *International journal of molecular sciences*, vol. 20, no. 4, p. 978, 2019.

[90] S. Cheng, L. Zhang, J. Tan, W. Gong, C. Li, and X. Zhang, "Dm-rpis: Predicting ncrna-protein interactions using stacked ensembling strategy," *Computational biology and chemistry*, vol. 83, p. 107088, 2019.

[91] A. Li, M. Ge, Y. Zhang, C. Peng, and M. Wang, "Predicting long noncoding rna and protein interactions using heterogeneous network model," *BioMed research international*, vol. 2015, 2015.

[92] M. Ge, A. Li, and M. Wang, "A bipartite network-based method for prediction of long non-coding rna–protein interactions," *Genomics, proteomics & bioinformatics*, vol. 14, no. 1, pp. 62–71, 2016.

[93] Y. Xiao, J. Zhang, and L. Deng, "Prediction of lncrna-protein interactions using hetesim scores based on heterogeneous networks," *Scientific reports*, vol. 7, no. 1, p. 3664, 2017.

[94] C. Shi, X. Kong, Y. Huang, S. Y. Philip, and B. Wu, "Hetesim: A general framework for relevance measure in heterogeneous networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2479–2492, 2014.

[95] H. Hu, C. Zhu, H. Ai, L. Zhang, J. Zhao, Q. Zhao, and H. Liu, "Lpi-etslp: lncrna–protein interaction prediction using eigenvalue transformation-based semi-supervised link prediction," *Molecular BioSystems*, vol. 13, no. 9, pp. 1781–1787, 2017.

[96] W. Zhang, Q. Qu, Y. Zhang, and W. Wang, "The linear neighborhood propagation method for predicting long non-coding rna–protein interactions," *Neurocomputing*, vol. 273, pp. 526–534, 2018.

[97] Q. Zhao, Y. Zhang, H. Hu, G. Ren, W. Zhang, and H. Liu, "Irwnrlpi: integrating random walk and neighborhood regularized logistic matrix factorization for lncrna-protein interaction prediction," *Frontiers in genetics*, vol. 9, p. 239, 2018.

[98] L. Deng, J. Wang, Y. Xiao, Z. Wang, and H. Liu, "Accurate prediction of protein-lncrna interactions by diffusion and hetesim features across heterogeneous network," *BMC bioinformatics*, vol. 19, no. 1, p. 370, 2018.

[99] W. Zhang, X. Yue, G. Tang, W. Wu, F. Huang, and X. Zhang, "Sfpel-lpi: Sequence-based feature projection ensemble learning for predicting lncrna-protein interactions," *PLoS computational biology*, vol. 14, no. 12, p. e1006616, 2018.

[100] Q. Zhao, H. Yu, Z. Ming, H. Hu, G. Ren, and H. Liu, "The bipartite network projection-recommended algorithm for predicting long non-coding rna-protein interactions," *Molecular Therapy-Nucleic Acids*, vol. 13, pp. 464–471, 2018.

[101] C. Shen, Y. Ding, J. Tang, L. Jiang, and F. Guo, "Lpi-ktaslp: Prediction of lncrna-protein interaction by semi-supervised link learning with multivariate information," *IEEE Access*, vol. 7, pp. 13486–13496, 2019.

[102] G. Xie, C. Wu, Y. Sun, Z. Fan, and J. Liu, "Lpi-ibnra: Long non-coding rna-protein interaction prediction based on improved bipartite network recommender algorithm," *Frontiers in genetics*, vol. 10, p. 343, 2019.

[103] Y.-T. Tseng, W. Li, C.-H. Chen, S. Zhang, J. J. Chen, X. J. Zhou, and C.-C. Liu, "Iiidb: a database for isoform-isoform interactions and isoform network modules," *BMC genomics*, vol. 16, no. 2, p. S10, 2015.

[104] Y. Hao, W. Wu, H. Li, J. Yuan, J. Luo, Y. Zhao, and R. Chen, "Npinter v3. 0: an upgraded database of noncoding rna-associated interactions," *Database*, vol. 2016, 2016.

[105] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[106] J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski, B. L. Aken, D. Barrell, A. Zadissa, S. Searle, *et al.*, "Gencode: the reference human genome annotation for the encode project," *Genome research*, vol. 22, no. 9, pp. 1760–1774, 2012.

[107] T. Hubbard, D. Barker, E. Birney, G. Cameron, Y. Chen, L. Clark, T. Cox, J. Cuff, V. Curwen, T. Down, *et al.*, "The ensembl genome database project," *Nucleic acids research*, vol. 30, no. 1, pp. 38–41, 2002.

[108] P. Steffen, B. Voß, M. Rehmsmeier, J. Reeder, and R. Giegerich, "Rnashapes: an integrated rna analysis package based on abstract shapes," *Bioinformatics*, vol. 22, no. 4, pp. 500–503, 2005.

[109] X. Pan, P. Rijnbeek, J. Yan, and H.-B. Shen, "Prediction of rna-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks," *BMC genomics*, vol. 19, no. 1, p. 511, 2018.

[110] Y. Yang, R. Heffernan, K. Paliwal, J. Lyons, A. Dehzangi, A. Sharma, J. Wang, A. Sattar, and Y. Zhou, "Spider2: A package to predict secondary structure, accessible surface area, and main-chain torsional angles by deep neural networks," in *Prediction of Protein Secondary Structure*, pp. 55–63, Springer, 2017.

[111] Z. Zhao, J. Bai, A. Wu, Y. Wang, J. Zhang, Z. Wang, Y. Li, J. Xu, and X. Li, "Co-lncrna: investigating the lncrna combinatorial effects in go annotations and kegg pathways based on human rna-seq data," *Database*, vol. 2015, 2015.

[112] N. A. O'Leary, M. W. Wright, J. R. Brister, S. Ciufo, D. Haddad, R. McVeigh, B. Rajput, B. Robbertse, B. Smith-White, D. Ako-Adjei, *et al.*, "Reference sequence (refseq)

database at ncbi: current status, taxonomic expansion, and functional annotation," *Nucleic acids research*, vol. 44, no. D1, pp. D733–D745, 2015.

[113] S. Fang, L. Zhang, J. Guo, Y. Niu, Y. Wu, H. Li, L. Zhao, X. Li, X. Teng, X. Sun, *et al.*, "Noncodev5: a comprehensive annotation database for long non-coding rnas," *Nucleic acids research*, vol. 46, no. D1, pp. D308–D314, 2017.

[114] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, 2017.

[115] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[116] F. Chollet *et al.*, "Keras." https://keras.io, 2015.

[117] D. Quang and X. Xie, "Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences," *Nucleic acids research*, vol. 44, no. 11, pp. e107–e107, 2016.

[118] D. Quang and X. Xie, "Factornet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data," *Methods*, 2019.

[119] R. Ehsani and F. Drabløs, "Measures of co-expression for improved function prediction of long non-coding rnas," *BMC bioinformatics*, vol. 19, no. 1, p. 533, 2018.

[120] P. Langfelder and S. Horvath, "Wgcna: an r package for weighted correlation network analysis," *BMC bioinformatics*, vol. 9, no. 1, p. 559, 2008.

[121] P. Langfelder and S. Horvath, "Wgcna package faq," 2017.

[122] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in neural information processing systems*, pp. 109–117, 2011.

[123] J. Yang, A. Li, M. Ge, and M. Wang, "Prediction of interactions between lncrna and protein by using relevance search in a heterogeneous lncrna-protein network," in *2015 34th Chinese Control Conference (Ccc)*, pp. 8540–8544, IEEE, 2015.

[124] Q. F. Gronau and E.-J. Wagenmakers, "Limitations of bayesian leave-one-out cross-validation for model selection," *Computational brain & behavior*, vol. 2, no. 1, pp. 1–11, 2019.

[125] J. Huerta-Cepas, D. Szklarczyk, D. Heller, A. Hernández-Plaza, S. K. Forslund, H. Cook, D. R. Mende, I. Letunic, T. Rattei, L. J. Jensen, *et al.*, "eggnog 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses," *Nucleic acids research*, vol. 47, no. D1, pp. D309–D314, 2019.

[126] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[127] Y. Wan, K. Qu, Q. C. Zhang, R. A. Flynn, O. Manor, Z. Ouyang, J. Zhang, R. C. Spitale, M. P. Snyder, E. Segal, *et al.*, "Landscape and variation of rna secondary structure across the human transcriptome," *Nature*, vol. 505, no. 7485, p. 706, 2014.

[128] H. Caniza, A. E. Romero, S. Heron, H. Yang, A. Devoto, M. Frasca, M. Mesiti, G. Valentini, and A. Paccanaro, "Gossto: a stand-alone application and a web tool for calculating semantic similarities on the gene ontology," *Bioinformatics*, vol. 30, no. 15, pp. 2235–2236, 2014.

[129] P. Johnsson, L. Lipovich, D. Grandér, and K. V. Morris, "Evolutionary conservation of long non-coding rnas; sequence, structure, function," *Biochimica et Biophysica Acta (BBA)-General Subjects*, vol. 1840, no. 3, pp. 1063–1071, 2014.

[130] D. Li and M. Q. Yang, "Identification and characterization of conserved lncrnas in human and rat brain," *BMC bioinformatics*, vol. 18, no. 14, p. 489, 2017.

[131] J. Tu, G. Tian, H.-H. Cheung, W. Wei, and T.-l. Lee, "Gas5 is an essential lncrna regulator for self-renewal and pluripotency of mouse embryonic stem cells and induced pluripotent stem cells," *Stem cell research & therapy*, vol. 9, no. 1, p. 71, 2018.

[132] N. Pospiech, H. Cibis, L. Dietrich, F. Müller, T. Bange, and S. Hennig, "Identification of novel pandar protein interaction partners involved in splicing regulation," *Scientific reports*, vol. 8, no. 1, p. 2798, 2018.

[133] M. Zhang, Y. Gu, M. Su, S. Zhang, C. Chen, W. Lv, and Y. Zhang, "Inferring novel lncrna associated with ventricular septal defect by dna methylation interaction network," *BioRxiv*, p. 459677, 2018.

[134] X. Yin, S. Huang, R. Zhu, F. Fan, C. Sun, and Y. Hu, "Identification of long non-coding rna competing interactions and biological pathways associated with prognosis in pediatric and adolescent cytogenetically normal acute myeloid leukemia," *Cancer cell international*, vol. 18, no. 1, p. 122, 2018.

[135] Y. Xing, Z. Zhao, Y. Zhu, L. Zhao, A. Zhu, and D. Piao, "Comprehensive analysis of differential expression profiles of mrnas and lncrnas and identification of a 14-lncrna prognostic signature for patients with colon adenocarcinoma," *Oncology reports*, vol. 39, no. 5, pp. 2365–2375, 2018.

[136] D. Yue, H. Liu, and Y. Huang, "Survey of computational algorithms for microrna target prediction," *Current genomics*, vol. 10, no. 7, pp. 478–492, 2009.

[137] A. Pla, X. Zhong, and S. Rayner, "miraw: A deep learning-based approach to predict microrna targets by analyzing whole microrna transcripts," *PLoS computational biology*, vol. 14, no. 7, p. e1006185, 2018.

[138] B. P. Lewis, C. B. Burge, and D. P. Bartel, "Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microrna targets," *cell*, vol. 120, no. 1, pp. 15–20, 2005.

[139] A. J. Enright, B. John, U. Gaul, T. Tuschl, C. Sander, and D. S. Marks, "Microrna targets in drosophila," *Genome biology*, vol. 5, no. 1, p. R1, 2003.

[140] M. Kertesz, N. Iovino, U. Unnerstall, U. Gaul, and E. Segal, "The role of site accessibility in microrna target recognition," *Nature genetics*, vol. 39, no. 10, pp. 1278–1284, 2007.

[141] X. Wang, "mirdb: a microrna target prediction and functional annotation database with a wiki interface," *Rna*, vol. 14, no. 6, pp. 1012–1017, 2008.

[142] A. Krek, D. Grün, M. N. Poy, R. Wolf, L. Rosenberg, E. J. Epstein, P. MacMenamin, I. Da Piedade, K. C. Gunsalus, M. Stoffel, *et al.*, "Combinatorial microrna target predictions," *Nature genetics*, vol. 37, no. 5, pp. 495–500, 2005.

[143] S.-K. Kim, J.-W. Nam, J.-K. Rhee, W.-J. Lee, and B.-T. Zhang, "mitarget: microrna target gene prediction using a support vector machine," *BMC bioinformatics*, vol. 7, no. 1, pp. 1–12, 2006.

[144] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[145] S. Cheng, M. Guo, C. Wang, X. Liu, Y. Liu, and X. Wu, "Mirtdl: a deep learning approach for mirna target prediction," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 13, no. 6, pp. 1161–1169, 2015.

[146] B. Lee, J. Baek, S. Park, and S. Yoon, "deeptarget: end-to-end learning framework for microrna target prediction using deep recurrent neural networks," in *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 434–442, 2016.

[147] M. Wen, P. Cong, Z. Zhang, H. Lu, and T. Li, "Deepmirtar: a deep-learning approach for predicting human mirna targets," *Bioinformatics*, vol. 34, no. 22, pp. 3781–3787, 2018.

[148] I. S. Vlachos, M. D. Paraskevopoulou, D. Karagkouni, G. Georgakilas, T. Vergoulis, I. Kanellos, I.-L. Anastasopoulos, S. Maniou, K. Karathanou, D. Kalfakakou, *et al.*, "Diana-tarbase v7. 0: indexing more than half a million experimentally supported mirna: mrna interactions," *Nucleic acids research*, vol. 43, no. D1, pp. D153–D159, 2015.

[149] H.-Y. Huang, Y.-C.-D. Lin, J. Li, K.-Y. Huang, S. Shrestha, H.-C. Hong, Y. Tang, Y.-G. Chen, C.-N. Jin, Y. Yu, *et al.*, "mirtarbase 2020: updates to the experimentally validated microrna–target interaction database," *Nucleic acids research*, vol. 48, no. D1, pp. D148–D154, 2020.

[150] Z. Yang, L. Wu, A. Wang, W. Tang, Y. Zhao, H. Zhao, and A. E. Teschendorff, "dbdemc 2.0: updated database of differentially expressed mirnas in human cancers," *Nucleic acids research*, vol. 45, no. D1, pp. D812–D818, 2017.

[151] A. Belter, D. Gudanis, K. Rolle, M. Piwecka, Z. Gdaniec, M. Z. Naskret-Barciszewska, and J. Barciszewski, "Mature mirnas form secondary structure, which suggests their function beyond risc," *PloS one*, vol. 9, no. 11, p. e113848, 2014.

[152] N. Ludwig, P. Leidinger, K. Becker, C. Backes, T. Fehlmann, C. Pallasch, S. Rheinheimer, B. Meder, C. Stähler, E. Meese, *et al.*, "Distribution of mirna expression across human tissues," *Nucleic acids research*, vol. 44, no. 8, pp. 3865–3877, 2016.

[153] H. Hezroni, R. B.-T. Perry, Z. Meir, G. Housman, Y. Lubelsky, and I. Ulitsky, "A subset of conserved mammalian long non-coding rnas are fossils of ancestral protein-coding genes," *Genome biology*, vol. 18, no. 1, p. 162, 2017.

[154] C. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 1, pp. 10–24, 2017.

[155] X. Zheng, Y. Wang, K. Tian, J. Zhou, J. Guan, L. Luo, and S. Zhou, "Fusing multiple protein-protein similarity networks to effectively predict lncrna-protein interactions," *BMC bioinformatics*, vol. 18, no. 12, p. 420, 2017.

[156] J. Yuan, W. Wu, C. Xie, G. Zhao, Y. Zhao, and R. Chen, "Npinter v2. 0: an updated database of ncrna interactions," *Nucleic acids research*, vol. 42, no. D1, pp. D104–D108, 2013.

[157] C. Sutton, A. McCallum, *et al.*, "An introduction to conditional random fields," *Foundations and Trends® in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.

[158] S. Jalali, D. Bhartiya, M. K. Lalwani, S. Sivasubbu, and V. Scaria, "Systematic transcriptome wide analysis of lncrna-mirna interactions," *PloS one*, vol. 8, no. 2, p. e53823, 2013.

[159] C. Xie, J. Yuan, H. Li, M. Li, G. Zhao, D. Bu, W. Zhu, W. Wu, R. Chen, and Y. Zhao, "Noncodev4: exploring the world of long non-coding rna genes," *Nucleic acids research*, vol. 42, no. D1, pp. D98–D103, 2013.

[160] R. Leinonen, H. Sugawara, M. Shumway, and I. N. S. D. Collaboration, "The sequence read archive," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D19–D21, 2010.

[161] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[162] A. R. Delbridge, S. Grabow, A. Strasser, and D. L. Vaux, "Thirty years of bcl-2: translating cell death discoveries into novel cancer therapies," *Nature reviews Cancer*, vol. 16, no. 2, pp. 99–109, 2016.

[163] P. E. Czabotar, G. Lessene, A. Strasser, and J. M. Adams, "Control of apoptosis by the bcl-2 protein family: implications for physiology and therapy," *Nature reviews Molecular cell biology*, vol. 15, no. 1, pp. 49–63, 2014.

[164] L. Yang, Y. Zhou, Y. Li, J. Zhou, Y. Wu, Y. Cui, G. Yang, and Y. Hong, "Mutations of p53 and kras activate nf-κb to promote chemoresistance and tumorigenesis via dysregulation of cell cycle and suppression of apoptosis in lung cancer cells," *Cancer letters*, vol. 357, no. 2, pp. 520–526, 2015.

[165] G. Ambrosini, C. Adida, and D. C. Altieri, "A novel anti-apoptosis gene, survivin, expressed in cancer and lymphoma," *Nature medicine*, vol. 3, no. 8, pp. 917–921, 1997.

[166] C. Esteban, O. Staeck, S. Baier, Y. Yang, and V. Tresp, "Predicting clinical events by combining static and dynamic information using recurrent neural networks," in *Healthcare Informatics (ICHI), 2016 IEEE International Conference on*, pp. 93–101, IEEE, 2016.

[167] A. Beger, "Precision-recall curves," 2016.

[168] P. Flach and M. Kull, "Precision-recall-gain curves: Pr analysis done right," in *Advances in Neural Information Processing Systems*, pp. 838–846, 2015.

[169] G. W. Beadle and E. L. Tatum, "Genetic control of biochemical reactions in neurospora," *proceedings of the National Academy of Sciences*, vol. 27, no. 11, pp. 499–506, 1941.

[170] O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein, "A bayesian framework for combining heterogeneous data sources for gene function prediction (in saccharomyces cerevisiae)," *Proceedings of the National Academy of Sciences*, vol. 100, no. 14, pp. 8348–8353, 2003.

[171] J. Yang, R. Yan, A. Roy, D. Xu, J. Poisson, and Y. Zhang, "The i-tasser suite: protein structure and function prediction," *Nature methods*, vol. 12, no. 1, pp. 7–8, 2015.

[172] T. Lappalainen, M. Sammeth, M. R. Friedländer, P. AC't Hoen, J. Monlong, M. A. Rivas, M. Gonzalez-Porta, N. Kurbatova, T. Griebel, P. G. Ferreira, *et al.*, "Transcriptome and genome sequencing uncovers functional variation in humans," *Nature*, vol. 501, no. 7468, pp. 506–511, 2013.

[173] R. Edgar, M. Domrachev, and A. E. Lash, "Gene expression omnibus: Ncbi gene expression and hybridization array data repository," *Nucleic acids research*, vol. 30, no. 1, pp. 207–210, 2002.

[174] J. Frühwald, J. C. Londoño, S. Dembla, S. Mannebach, A. Lis, A. Drews, U. Wissenbach, J. Oberwinkler, and S. E. Philipp, "Alternative splicing of a protein domain indispensable for function of transient receptor potential melastatin 3 (trpm3) ion channels," *Journal of Biological Chemistry*, vol. 287, no. 44, pp. 36663–36672, 2012.

[175] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, *et al.*, "Gene ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[176] A. Roberts and L. Pachter, "Streaming fragment assignment for real-time analysis of sequencing experiments," *Nature methods*, vol. 10, no. 1, pp. 71–73, 2013.

[177] E. W. Sayers, T. Barrett, D. A. Benson, E. Bolton, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, S. Federhen, *et al.*, "Database resources of the national center for biotechnology information," *Nucleic acids research*, vol. 40, no. D1, pp. D13–D25, 2012.

[178] Y. Liu and B. Schmidt, "Long read alignment based on maximal exact match seeds," *Bioinformatics*, vol. 28, no. 18, pp. i318–i324, 2012.

[179] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[180] C. Trapnell, L. Pachter, and S. L. Salzberg, "Tophat: discovering splice junctions with rna-seq," *Bioinformatics*, vol. 25, no. 9, pp. 1105–1111, 2009.

[181] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg, "Tophat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions," *Genome biology*, vol. 14, no. 4, p. R36, 2013.

[182] C. Trapnell, A. Roberts, L. Goff, G. Pertea, D. Kim, D. R. Kelley, H. Pimentel, S. L. Salzberg, J. L. Rinn, and L. Pachter, "Differential gene and transcript expression analysis of rna-seq experiments with tophat and cufflinks," *Nature protocols*, vol. 7, no. 3, pp. 562–578, 2012.

[183] Y. Guan, C. L. Myers, D. C. Hess, Z. Barutcuoglu, A. A. Caudy, and O. G. Troyanskaya, "Predicting gene function in a hierarchical context with an ensemble of classifiers," *Genome biology*, vol. 9, no. 1, p. S3, 2008.

[184] D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems," *Machine learning*, vol. 45, no. 2, pp. 171–186, 2001.

[185] D. Quang and X. Xie, "Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences," *Nucleic acids research*, vol. 44, no. 11, pp. e107–e107, 2016.

[186] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning-based sequence model," *Nature methods*, vol. 12, no. 10, pp. 931–934, 2015.

[187] D. Hunt, R. J. Leventer, C. Simons, R. Taft, K. J. Swoboda, M. Gawne-Cain, A. C. Magee, P. D. Turnpenny, D. Baralle, *et al.*, "Whole exome sequencing in family trios reveals de novo mutations in pura as a cause of severe neurodevelopmental delay and learning disability," *Journal of medical genetics*, vol. 51, no. 12, pp. 806–813, 2014.

[188] O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein, "A bayesian framework for combining heterogeneous data sources for gene function prediction (in saccharomyces cerevisiae)," *Proceedings of the National Academy of Sciences*, vol. 100, no. 14, pp. 8348–8353, 2003.

[189] G. O. Consortium *et al.*, "The gene ontology (go) database and informatics resource," *Nucleic acids research*, vol. 32, no. suppl 1, pp. D258–D261, 2004.

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

[191] J. Yang, "Review of multi-instance learning and its applications," *Technical report, School of Computer Science Carnegie Mellon University*, 2005.

[192] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 668–675, 2013.

[193] D. Kotzias, M. Denil, P. Blunsom, and N. de Freitas, "Deep multi-instance transfer learning," *arXiv preprint arXiv:1411.3128*, 2014.

[194] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[195] J. Jiang, "A literature survey on domain adaptation of statistical classifiers," *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, vol. 3, 2008.