BIOINFORMATICS

Vol. 00 no. 00 2010
Pages 1–9

Semi-Supervised Multi-Task Learning for Predicting Interactions between HIV-1 and Human Proteins

Yanjun Qi [1]*, Oznur Tastan [2], Jaime G. Carbonell [2], Judith Klein-Seetharaman [2] and Jason Weston [3]

[1]NEC Labs America, 4 Independence Way, Princeton, NJ 08540 USA
[2]School of Computer Science, Carnegie Mellon University, PA 15213 USA
[3]Google Research NY, 75 Ninth Avenue, New York, NY 10011 USA

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: Protein-protein interactions (PPIs) are critical for virtually every biological function. Recently, researchers suggested to use supervised learning for the task of classifying pairs of proteins as interacting or not. However, its performance is largely restricted by the availability of truly interacting proteins (*labeled*). Meanwhile there exist a considerable amount of protein pairs where an association appears between two partners, but not enough experimental evidence to support it as a direct interaction (*partially labeled*).

Results: We propose a semi-supervised multi-task framework for predicting PPIs from not only *labeled*, but also *partially labeled* reference sets. The basic idea is to perform multi-task learning on a supervised classification task and a semi-supervised task. The supervised classifier trains a multi-layer perceptron network for PPI predictions from *labeled* examples. The semi-supervised auxiliary task shares network layers of the supervised classifier and trains with *partially labeled* examples. Semi-supervision could be utilized in multiple ways. We tried three approaches in this paper, (1) classification (to distinguish partial positives with negatives); (2) ranking (to rate partial positive more likely than negatives); (3) embedding (to uncover data clusters with two sub-approaches). We applied this framework to identify the set of interacting pairs between HIV-1 and human proteins. Our method improved upon the state-of-the-art method for this task indicating the benefits of semi-supervised multi-task learning using auxiliary information.

Availability: Datasets&Code @ www.cs.cmu.edu/~qyj/HIVsemi

Contact: qyj@cs.cmu.edu

1 INTRODUCTION

Comprehensively identifying protein interactions is essential for understanding the molecular mechanisms underlying biological functions. Because of their importance in development and disease, protein-protein interactions (PPIs) have been the subject of intense research in recent years, both *computationally* and *experimentally*.

Experimental techniques for deciphering protein-protein interactions have been reviewed in (Shoemaker and Panchenko, 2007a). Traditionally, PPIs have been studied individually through the use of genetic, biochemical and biophysical experimental techniques (also termed *small-scale* methods). The related experiments are painfully time-consuming in general (months for detecting just one PPI). In recent years, *large-scale* biological PPI experiments have been introduced to directly detect hundreds or thousands of protein interactions at a time. The two-hybrid (Y2H) screens (Ito *et al.*, 2001; Uetz *et al.*, 2000; Rual *et al.*, 2005; Stelzl *et al.*, 2005) and complex purification detection techniques using mass spectrometry (Gavin *et al.*, 2002, 2006; Ho *et al.*, 2002) are the two most popular approaches successfully applied on a large scale. However the resulting data sets are often incomplete and exhibit high false positive and false negative rates (von Mering *et al.*, 2002; Yu *et al.*, 2008).

Computational methods have been successfully applied to predict protein interactions (reviewed in (Shoemaker and Panchenko, 2007b)). Taking into account that indirect sources may contain partial evidence about protein interations, several approaches derived their prediction on a particular type of information, such as overrepresented domain pairs in interacting proteins (Deng *et al.*, 2002; Wang *et al.*, 2007). An alternative attractive approach is to integrate various indirect or direct evidences from multiple information sources in a statistical learning framework. A classifier is trained to distinguish between positive examples of truly interacting protein pairs and negative examples of non-interacting pairs. *Various methods* have been explored in this framework, including naive Bayes classifier by Jansen et a. (Jansen *et al.*, 2003) , decision tree from Zhang et al. (Zhang *et al.*, 2004), kernel based methods from (Yamanishi *et al.*, 2004; Ben-Hur and Noble, 2005), random forest based method (Qi *et al.*, 2005), logistic regression (Lin *et al.*, 2004), and the strategies of summing likelihood ratio scores (Rhodes *et al.*, 2005; Scott and Barton, 2007; Lee *et al.*, 2004). Most of these studies have been carried out in yeast or human. They aim to predict PPIs within a single organism (termed *'intra-species PPI prediction'*). Recently, using computational methods to predict PPIs between organisms (*'inter-species prediction'*) has raised great interest as well, especially between host and pathogens. Tastan et al. (Tastan *et al.*, 2009) extended the supervised learning framework to predict PPIs between HIV-1 and human proteins. A random forest-based classifier was used to integrate multiple biological information sources and achieved the state-of-art performance for this task. Besides, Davis et al. (Davis *et al.*, 2007) studied ten host-pathogen protein-protein interactions using structural information. Later, Evans et al. (Evans *et al.*, 2009) searched for

*to whom correspondence should be addressed

host protein motifs along virus protein sequences to obtain a list of host proteins highly enriched with those targeted by HIV-1 proteins.

While the supervised framework was shown to improve the quality of current PPI data, its applicability is still limited. Supervised PPI detection requires a large number of labeled training examples to learn accurately. Except several well studied organisms like yeast or human, most inter or intra-species PPI prediction tasks do not have a large number of reliable PPIs available. This limitation largely restricts the prediction ability of computational PPI algorithms.

Besides the small number of reliable PPIs, in general, it is relatively easy to get a certain amount of protein pairs that are highly likely to be interacting pairs (but are not reliably annotated). For instance, in the task of predicting PPIs between HIV-1 and human proteins, NIAID (Fu *et al.*, 2008) database retrieved protein pairs between HIV-1 protein and human protein from the scientific literature (details in Section 2). The extracted pairs are not reliable PPIs, but are very likely to have interaction relationships. From a learning perspective, these pairs are weakly labeled positive examples. When searching PPIs in interesting "intra-species" or "inter-species" cases, adding partially labeled PPI pairs would be an interesting and important research direction to improve computational predictions.

In this paper we present a multi-task learning framework to make use of partial labeled examples together with labeled PPI pairs. A semi-supervised task is plugged into a multiple layer perceptron network architecture as an *auxiliary task*. We train supervised PPI classification and the semi-supervised auxiliary task using the same network architecture *simultaneously*. We applied our method to predict the set of interacting proteins between HIV-1 and human proteins by information integration of multiple biological sources. Our method improved upon the previous approach applied for this task. The results indicate that with the proposed semi-supervised multi-task approach, auxiliary information (partial labels) is able to improve the generalization ability of predicting interaction pairs between HIV-1 and human proteins.

The rest of the article is as follows. In Section 2 we describe the task of predicting PPIs between HIV-1 and human proteins and the available interacting data set in more details. In Section 3 we describe the semi-supervised multi-task learning framework. Section 4 presents the experimental results and Section 5 concludes.

2 TARGET PROBLEM

HIV-1 causes acquired immune deficiency syndrome (AIDS). Since the first epidemic 25 years ago, it remains a serious threat to public health (Trkola, Oct). Both HIV-1 transmission and infection are complex processes, where much remains to be elucidated. HIV-1 encodes only a handful of proteins; however it subverts the cellular machinery for its benefit. Virus-host protein-protein interactions are key in deciphering virus strategies; understanding of which should lead to designing novel ways to get HIV-1 under control.

2.1 Information Integration with Multiple Data Sources

Recently, we made the attempt to predict the global set of interactions between HIV-1 and human host cellular proteins in Tastan et al. (2008) (Tastan *et al.*, 2009). The task is to predict whether a given human to HIV-1 protein pair interacts or not. Thus it was formulated as a binary classification problem, where each protein pair belongs to either the "interaction" or "non-interaction" class. A random forest classifier was applied on a rich feature set including:

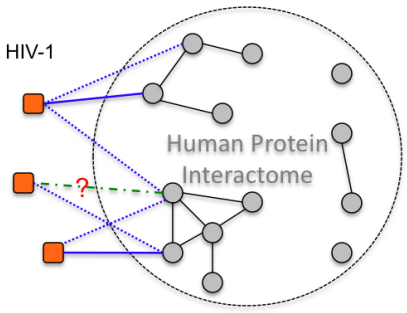


Fig. 1. Target problem: Predicting protein interactions between HIV-1 (organge squre) and human (grey circle). There exist weakly labeled interaction pairs from NIAID (dashed blue edges) and labeled interaction pairs from experts' annotation (solid blue edges). We aim to predict whether a given human to HIV-1 protein pair (dashed green) interacts or not.

- co-occurrence of motifs and their interaction domains;
- gene expression profile features reflecting human gene expression patterns across HIV-1 infected vs. uninfected samples;
- similarity in terms of cellular location, molecular function and biological processes;
- pairwise sequence similarity and similarity of HIV-1 protein sequence to human proteinâĽ™s known human binding partners;
- posttranslational modification similarity to neighbor, which captures if the HIV-1 protein shares any modification with at least one of the binding partners of the human protein;
- similarity of tissue distributions;
- topological properties of the human protein in human protein interaction network;
- similarity of HIV-1 protein to human protein's known binding human partners.
- HIV-1 protein type

All data sources and how they were converted into features representing pair of HIV-1 to human proteins have been described previously in Tastan et al. (2008) (Tastan *et al.*, 2009) and in the supplementary website.

2.2 Partial Positive Labels from NIAID

The gold standard positive set we used in (Tastan *et al.*, 2009) were collected from NIAID (Fu *et al.*, 2008) database where interactions between HIV-1 and cellular proteins reported in the scientific literature were retrieved by this database. It includes **2620** protein pairs involving 1406 human proteins and 17 HIV-1 proteins (15 HIV-1 proteins plus precursors of the envelope (env gp160) and gag (gag pr55)). Each interaction in the database is associated with keywords extracted from scientific literature reporting the interaction. Some of these keywords are strong such as "interacts with" and "binds" (we named this set as "GroupI" containing **955** protein pairs). Which some others are rather weak indicators of a direct interaction such as "upregulates" (this set of pairs were named as "GroupII" which includes **1665** protein pairs). Our previous work (Tastan *et al.*, 2009)) used those "GroupI" interactions (Tastan *et al.*, 2009)) (associated with strong

Table 1. Basic Statistics for the Prediction of Interactions between HIV-1 and Human Proteins by Information Integration

Features	Positive PPIs	Partial Positive	Remaining Pairs	HIV-1 Protein	Human Protein
18	158	2119	352338*	17	20873

This also excludes 226 pairs experts labeled as "unsure or indirect"

keywords, [1])) as training positive examples for binary interaction predictions. However there exist not enough evidence supporting the reliabilities of interactions in NIAID database. Recently researchers (Cusick *et al.*, 2008) found out that the literature-curated protein interaction experiments can be error-prone and possibly of lower quality than commonly assumed.

2.3 Positive Labels from Experts' Annotations

To increase the data quality, we consulted 16 HIV-1 experts about the validity of the interac- tions reported in the NIAID database (15 of the experts are professors well known in the HIV-1 field and the last expert is a PHD student, who had extensively worked on HIV1 for five years. More details of experts' annotation process in paper (Tastan *et al.*, 2010)). HIV-1 experts are sent lists of interacting pairs along with the interaction keywords and the links to the articles reporting the interactions in NIAID. Experts are asked to annotate each pair with the "interact" label if they believe the reported pair is a true direct interaction. If, on the other hand, either they do not believe two proteins interact, annotating it with the label "not interacting", or they think the interaction might be indirect or they are unsure about the label, labeling it as "unsure or indirect". For each HIV-1 protein, the rules to select potential interation partners sent to experts are different. If for a certain HIV-1 protein, the total number of interactions reported in NIAID is not many, we sent all of the interactions reported in the database. In other cases, only the subset of interactions associated with keywords "binds" or "interacts with" are sent (the longer the list is, the slower and more reluctant the experts' responses were). In this way 361 interacting pairs are annotated. Most of the interactions (256/361) were annotated by a single expert and the rest received labels from multiple experts. In cases where there is a disagreement of expert opinions on the labels, majority voting strategy was used to decide which label would be assigned. Finally this resulted in **158** protein pairs which HIV-1 experts attributed as direct interactions between HIV-1 and human proteins.

Thus, this set serves as our positive "gold standard" set. The rest of the NIAID dataset are treated as "partial positives" examples since not enough evidence is yet accumulated for them to be considered as direct interactions but they are likely candidates.

In summary, this binary classification task contains **158** positive example and **2119** partial positive (552 "groupI" and 1567 "groupII") PPI pairs (after removing those pairs labeled as 'not interact' and 'interact' from the experts). Each HIV-1 human protein pair is represented with 18 features. Related statistics of data sets used for this task are listed in Table 1.

[1] Small difference exists between the post-processing here and in (Tastan *et al.*, 2009)

The feature set used in our previous work (Tastan *et al.*, 2009) contains totally 35 attributes for each potential HIV-1 to Human protein pair. Among them, 17 items represent which one (assuming i) of the 17 HIV-1 proteins this pair involves with (with the i dimension set to 1 and all the other 16 dimensions set to zero). As mentioned above, since the creation process of positive labels is correlated non-randomly with the type of HIV-1 proteins, we have to remove these 17 features, and use the remainging 18 features to describe each HIV-1 human protein pair.

3 METHOD

A d-dimensional ($d = 18$) feature vector x was constructed for every protein pair (between a HIV-1 protein and a human protein). Each entry in the feature vector summarizes one biological evidence (asking, for example, "Is this HIV-1 protein similar to the human protein's neighbors in terms of sequence?" or "Does this HIV-1 protein include a certain motif that is highly likely to interact with one of the domains in the human protein?" (See Section 2.1). The target variable $y \in \{\pm 1\}$ represents whether this pair interacts (1) or not (-1). Thus, the problem of predicting protein interactions is handled as a binary classification task. Feature vector for the i-th HIV-1 Human protein pair is denoted as x_i, and whether it interacts or not with y_i.

Considering the small number of positive labels (**158**) and a larger set of partial labels (**2119**), we propose to design semi-supervised multi-task learning (SML) strategies for making use of both sets, to achieve better prediction performance.

3.1 Multi-Layer Perceptron Network for Supervised PPI Prediction

Given a set of labeled examples $(x_1, ...x_L)$ and corresponding labels $(y_1, ...y_L)$, our goal is to learn a supervised classifier (e.g. choose a discriminant function) $f(x)$, such that

$$f(x_i) > 0 \quad \text{if } y_i = 1$$
$$f(x_i) < 0 \quad \text{if } y_i = -1.$$

The supervised classifier we chose is a multi-layer perceptron (MLP) network with M layers of hidden units that give a 1-dimensional output:

$$f(x) = \sum_j w_j^O \, h_j^M(x) + b^O, \tag{1}$$

where w^O is the weight vector for the output layer. The m^{th} hidden layer is defined as

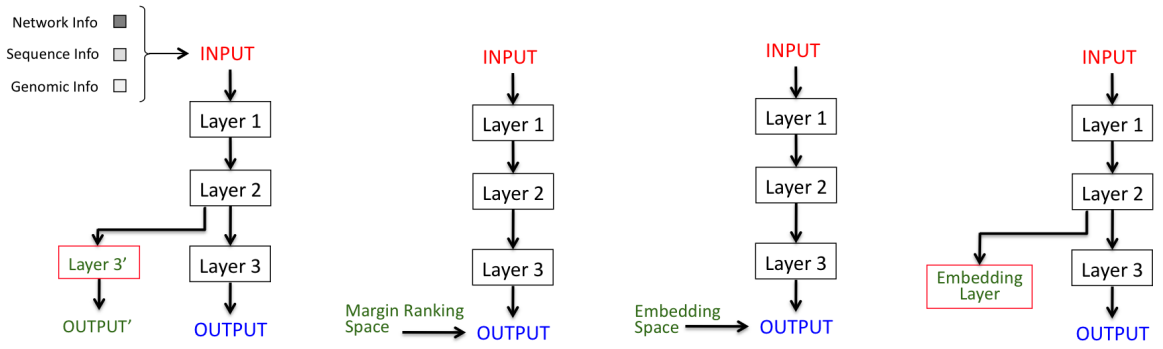
$$h_i^m(x) = S\left(\sum_j w_j^{m,i} \, h_j^{m-1}(x) + b^{m,i}\right), \; m > 1 \tag{2}$$

$$h_i^1(x) = S\left(\sum_{j=1}^{d} w_j^{1,i} \, x_j + b^{1,i}\right) \tag{3}$$

and S is a non-linear squashing function like "tanh". A standard fully connected multiple-layer perceptron network is utilized in this paper.

To train this supervised classifier, we employ the Hinge loss (on labeled examples):

$$\sum_{i=1}^{L} \ell(f(x_i), y_i) = \sum_{i=1}^{L} \max(0, 1 - y_i f(x_i)). \tag{4}$$



(a) With classification (b) With ranking (c) With embedding output (d) With embedding layer

Fig. 2. To do multi-task learning with the supervised PPI classification, the semi-supervised task has four possible ways to extend the network structure of multi-layer perceptron: (a) training another classifier to distinguish partial positive and negative examples; (b) training a ranker to sort partial positive and negative data; (c) training an embedding on the output of the supervised classifier; (d) adding a separate embedding layer for semi-supervision evidence.

3.2 Multi-Task Learning with Semi-Supervised Auxiliary Task

According to the available labels, we could formalize our problem as two tasks: (1) supervised classification with positive (from experts) and negative labels; (2) use a certain strategy for partial positive labels to improve the supervised classification. One natural way to combine two objectives using MLP network structure is through multi-task learning.

Multitask learning is the procedure of learning several tasks at the same time with the aim of mutual benefits. A good overview of multi-task learning, especially focusing on neural networks, can be found in (Caruana, 1997). The idea of sharing information learnt across sub-tasks seems a more economical use of data, where presumably all tasks are learnt *jointly*. A typical example is a MLP network where the first layers will be shared to all tasks and typically to learn levels of feature processing that are useful to all tasks.

For our problem, the second task aims to make use of weak positive labels and is auxiliary to the main classification. There are many ways to handle this **auxiliary task** using MLP networks. In the following, we propose four possibilities. We call them "semi-supervised auxiliary task" because in this task, there are no true labels, just weak labels with diverse levels of confidence (e.g. various keywords associated in NIAID database). Typical semi-supervised learning refers to the use of both labeled and unlabeled data during training. For our task, though not the typical semi-supervised setting, we view it as a similar setup (see Section 3.8 for related work) and three of the proposed auxiliary tasks could be naturally extended to unlabeled data with side information.

In summary, we perform multi-task learning on supervised classification and semi-supervised auxiliary task, which learns two tasks jointly with optimizing the following loss function:

$$\sum_{i=1}^{L} \ell(f(x_i), y_i) + \lambda \; Loss(AuxiliaryTask) \tag{5}$$

Using different network structure and/or distinct loss function, we propose four auxiliary tasks (in Figure 2).

3.3 Auxiliary Task I: Classification

Figure 3(a) illustrates the first strategy to use partial lables. This is the classical way of multi-tasking in the MLP framework. Our auxiliary task shares the first m layers of the original MLP, but have a new output layer:

$$g(x) = \sum_j w_j^{AUX,i} \; h_j^m(x) + b^{AUX,i} \tag{6}$$

This network is trained to *distinguish* partial positive examples from negative examples (e.g. classification), simultaneously as we train the original network on *labeled* data. Assuming a set of partially labeled examples $(x_{L+1}, ...x_{L+U})$. In this auxiliary task, they are assigned with corresponding pseudo labels $(y'_{1+U}, ...y'_{L+U})$. We train this pseudo classficiation with hinge loss as well, which means,

$$Loss(AuxiliaryTask) = \sum_{j=L+1}^{L+U} \max(0, 1 - y'_j g(x_j)). \tag{7}$$

3.4 Auxiliary Task II: Ranking

Illustrated in Figure 3(b), this time we use the same network architecture for both two tasks. The auxiliary information we know for the second task is "partial labeled PPI pairs are more likely to be true than negative pairs". This could be formalized as a "ranking" task using MLP: to rank "weak positive examples" highly than "negative examples" if ordering them by the output $f(x)$ from the MLP. Naturally the above assumption comes to minimize a ranking-type margin cost:

$$\sum_{p \in P} \sum_{n \in N} \max\left(0, \; 1 - f(x_p) + f(x_n)\right) \; , \tag{8}$$

where P means the index of partial positives and N represents the set of negative examples. The training is handled with stochastic gradient descent which samples the cost *online* w.r.t. (p, n).

3.5 Auxiliary Task III: Embedding on Output

One key assumption used by many semi-supervised algorithms is the structure assumption, which assumes that points within the same structure (such as a cluster or a manifold) are likely to have the same label (Chapelle *et al.*, 2006). In this task, we explore the partial labeled examples to uncover the hidden structure within $P(x)$ aiming to improve predictions of $P(y|x)$.

This could be pursued through the embedding technique called **Siamese Networks**, proposed by neural networks researchers, see

e.g. (Bromley *et al.*, 1993; Weston *et al.*, 2008). The proposed model contains a network with two identical copies of the same function, with the same weights, fed into a "distance"-measuring layer. Given two examples x_i and x_j, we can feed each of them into the two identical networks, and use the last "distance" layer to compute whether the two examples are similar or not. If we know in advance whether they are similar or not, this pairwise "labeling" can be used to train the network for learning an embedding.

A margin-based loss following (Weston *et al.*, 2008) is chosen for the "Siamese Network" embedding:

$$L(f_i, f_j, W_{ij}) = \begin{cases} ||f_i - f_j||_1 & \text{if } W_{ij} = 1, \\ \max(0, m - ||f_i - f_j||_1) & \text{if } W_{ij} = 0 \end{cases} \quad (9)$$

This loss function encourages similar examples (where $W_{ij} = 1$) to be close, and dissimilar ones to have a distance of at least m from each other. The matrix W of weights W_{ij} specifies the similarity or dissimilarity between examples x_i and x_j. This matrix is supplied in advance and serves as the "pairwise labeling" for the embedding loss function.

Matrix W specifies which examples are neighbors to each other (hence have value $W_{ij} = 1$), and which are not ($W_{ij} = 0$). For our data, since we have a reasonable amount of *partially labeled* example, we build part of the binary matrix W for the "Siamese Network" using this prior knowledge. Three strategies or their combinations are considered in our experiments:

1. Neighboring pairs with both examples from the *partially labeled* set.
2. Neighboring pairs with one *partially labeled* example and the other from the positively *labeled* set.
3. Non-neighboring pairs with one *partially labeled* example and the other a negatively *labeled* example.

Combinations of these components correspond to different choices of the matrix W.

All the example pairs fed to the embedding training belong to the above three cases. Our motivation is that even though examples of partial positive PPI sets have not enough evidence to be considered as direct interactions, they are highly likely candidates. Thus in the embedding space, these pairs should be similar to each other, and dissimilar to pairs including a negative example.

During the training procedure, the model is updated by these example pairs. These either "push" similar examples together or "pull" dissimilar examples apart from each other. This known structure in our data is exactly what we want to preserve.

It is very natural to multi-task auxiliary embedding with our main supervised classification task. The "Siamese Network" embedding made use of a neural network with two identical copies and an extra "distance measuring" layer. We can just use our supervised classfier MLP as the base network for embedding, which equals to add the "Siamese Network" embedding as a regularizer on our main classifier MLP. Then the question is: where or which layer to add ? Weston et al. (2008) (Weston *et al.*, 2008) proposed three possible ways and we used two of them as our auxiliary prediction task, as shown in Figure 2(c) and Figure 2(d).

In Figure 2(c), the embedding is on the output which means we add a semi-supervised regularizer to the supervised loss of the entire network's output (1):

$$\sum_{i=1}^{L} \ell(f(x_i), y_i) + \lambda \sum_{i,j=1}^{L+U} L(f(x_i), f(x_j), W_{ij}) \quad (10)$$

Where we have denoted labeled training examples (protein-protein pairs) for the supervised task as $x_i, i = 1, \ldots, L$ and partially labeled examples used in the embedding task as $x_i, i = L+1, \ldots, L+U$.

Here, we try to classify labeled examples, whilst simultaneously the embedding tries to push the classification score of partial positive examples close to the scores of positive examples, and apart from those scores of negative labeled examples.

3.6 Auxiliary Task IV: Embedding on Layer

Figure 2(d) shows another way to add embedding as the auxiliary task. We create an auxiliary network which shares the first m layers of the original MLP but has a new set of weights:

$$g_i(x) = \sum_j w_j^{AUX,i} \, h_j^m(x) + b^{AUX,i} \quad (11)$$

This network is then trained to *embed* our partially labeled examples simultaneously as we train the original network on *labeled* data. That is, we optimize the objective:

$$\sum_{i=1}^{L} \ell(f(x_i), y_i) + \lambda \sum_{i,j=1}^{P+U} L(g(x_i), g(x_j), W_{ij}). \quad (12)$$

Essentially, what we are doing is pushing the partial labels close to each other or to the positive set (depending on the training pairs for embedding, in some (hidden) feature space. At the same time, we try to classify the labeled examples accurately, based on their known labels.

3.7 Semi-Supervised Multi-Task Learning (SML)

The overall goal is that the auxiliary task is to improve accuracy on the supervised task by uncovering the hidden structure in the original data. All tasks, including classification, ranking and embedding (two kinds of embeddings tried here), are trained by stochastic gradient descent. The training cooperation between the main task and the auxiliary task could be summaried as looping over two tasks:

1. Select the next task.
2. Select a random training example for this task.
3. Update the MLP network parameters for this task by taking a gradient step with respect to this example.
4. Go to 1.

To give a concrete example, the pseudocode of multitasking with "embedding layer" case is given in Algorithm 1.

3.8 Related Work

Traditional supervised classifiers use only labeled data (feature/label pairs) to train. Semi-supervised learning refers to the use of both labeled and unlabeled data during training and has become one of the most natural forms of training, since unlabeled data and/or certain side information is normally abundant. Many semi-supervised learning algorithms exist, including Self-training (Scudder, 1965), co-training ((Blum and Mitchell, 1998)), Transductive SVMs ((Joachims, 1999;

Algorithm 1 Multi-Tasking with Embedding on Layer

Input: labeled data (x_i, y_i), $i = 1, \ldots, L$, partially labeled data x_i, $i = L+1, \ldots, L+U$, set of functions $f(\cdot)$, and embedding function $g(\cdot)$, see Figure 2(d) and equations (11), (12).

repeat

Pick a random *labeled* example (x_i, y_i).
Make a gradient step to optimize $\ell(f(x_i), y_i)$.

Pick a random *partially labeled* example x_p.
Pick a random example x_q, where $W_{pq} = 1$.
Make a gradient step for $\lambda L(g(x_p), g(x_q), 1)$.

Pick a random *partially labeled* example x_m.
Pick a random example x_n, where $W_{mn} = 0$.
Make a gradient step for $\lambda L(g(x_m), g(x_n), 0)$.

until stopping criteria is met.

Collobert *et al.*, 2006)), graph-based regularization ((Zhu *et al.*, 2003)), entropy regularization (Grandvalet and Bengio, 2005) and EM with generative mixture models (Nigam *et al.*, 2000), see (Chapelle *et al.*, 2006) for a review. Except self-training and co-training, most of these semi-supervised methods have scalability problems for realistic tasks.

Moreover, some methods use auxiliary tasks on large unlabeled corpora for training sequence models (i.e. through multi-task learning). Ando and Zhang (Ando and Zhang, 2005) proposed a method based on defining multiple tasks using unlabeled data that are multi-tasked with the task of interest, which they showed to perform very well on natural language processing tasks. Similarly, the language model strategy proposed in (Collobert and Weston, 2008) is another type of auxiliary task. Methods using auxiliary information can often find good solutions, however the selection of auxiliary problem requires significant insights. Our SML methods belong to this semi-supervised category.

In the field of computational protein-protein interaction predictions, there were attemps to add semi-supervision in as well. Yip et al. (Yip and Gerstein, 2009) proposed two semi-supervised learning methods to improve the so called "local models", by augmenting the limited number of gold-standard training instances with carefully chosen and highly confident pseudo examples. "Local models" were introduced by Bleakley et al. (Bleakley *et al.*, 2007) that uses a local model to allow for flexible modeling of subgroups of interactions. A local model is built for each protein, using the known interactions and non-interactions of this protein as the positive and negative examples. The resulting classification rule predicts edges associated with a single protein. Thus, each pair of proteins receives two predictions, each from the local model of either protein. The accuracy of computational techniques proposed for PPI network reconstruction is consistently limited by the small number of high-confidence examples. Specifically, for the local model approach, the uneven distribution of positive examples across the potential interaction space, with some objects having many known interactions and others few, makes it hard to predict new interaction partners for those proteins having very few known interactions reliably.

Two semi-supervised strategies were proposed by Yip et al. (Yip and Gerstein, 2009) to improve "local model": (1) The first method adds highly confident predictions from one local model as the pseudo examples of another. (2) The second strategy takes the most similar and most dissimilar proteins of each protein based on a feature evidence as the pseudo examples to add into training. Both strategies are simiar to the "self-training" (Scudder, 1965) idea proposed in machine learning community. "Self-training" utilizes large sets of unlabeled examples and try to improve over supervised methods by iteratively adding self-labeled *examples* predicted by the current model. This can give improvements to a model, but care must be taken as the predictions are prone to noise.

4 RESULTS

4.1 Experimental Setting

When training the classification model, non-interacting examples are required. However it is almost impossible to show two proteins do not interact, a real set of non-interacting protein pairs does not exist. A commonly applied strategy is to randomly select protein pairs from all possible protein pairs as the negative set, excluding those interacting ones. Here we exclude all those pairs that are in NIAID database. For interacting pairs between HIV-1 and human proteins, it is estimated that roughly only 1 in about 100 possible pairs actually interacts (Tastan *et al.*, 2009). This is an extreme unbalanced ratio between positive and negative sets. We use this ratio to build the negative set which includes ~16,000 random negative pairs.

The positive pairs in our setting include only those PPIs pairs confirmed by the HIV-1 experts as "interacting" (**158** pairs). The partial positive pairs (**2119** left pairs of NIAID) function as auxillary information in the training phase only.

To measure the predictive power of SML for identifying protein interactions between HIV-1 and Human, we compared four variants of SML with three other popular classifiers:

- RF: Random Forest;
- SVM: Support Vector Machine;
- MLP: Multi-Layer Perceptron Neural Net;

The four SML strategies are named as:

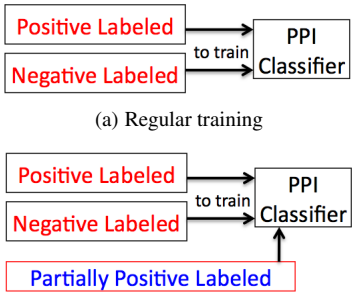
- SMLC: SML with auxiliary classification task;
- SMLR: SML with auxiliary ranking task;
- SMEO: SML with embedding on output space;
- SMEL: SML with embedding on layer;

The four SML methods and the MLP model are implemented using Torch 5 package (Weston *et al.*, 2008). We used standard toolkits for the other methods. Specifically, The LIBSVM toolkit was used for SVM (Chang and Lin, 2001). Random Forest was from the Berkeley RF package (Breiman, 2001).

4.2 Baselines to Compare

Essential our task formulation is still within the framework of supervised classification of protein pairs through information integration. The choice of above three baseline methods is because:

- (1) Under the supervised classification framework, RF and SVM were shown to give the best performance for yeast PPI prediction (Qi *et al.*, 2006; Lin *et al.*, 2004).
- (2) RF was shown to give the state-of-art performance for the HIV-human PPI prediction task (partial labels used for training in that case (Tastan *et al.*, 2009)).



(a) Regular training

(b) Adding partial positive into positive

Fig. 3. Two ways to train baseline classifiers for performance comparison. (a) train with positive + negative; (b) train with positive + partial positive (treat as positive) + negative.

- (3). RF method utilizes a collection of decision trees to determine if a protein pair interacts or not. It was shown to be robust against noise and missing feature values, which is instrinct of this classification problem. Thus we want to compare our semi-supervised multi-tasking strategies to RF.
- (4). Our SML models are built on MLP networks. Thus it is worth to compare and investigate how much improvement we could achieve beyong the baseline: MLP network classifier.

Besides, we also evaluate the performance of all three base classifiers when adding those partially labeled pairs in the training. Ideally these partial labels should be weighted differently in the training compared to those experts' labels. But since partial positive pairs are associated with different keywords in NIAID, it is tricky to select the weights. We finally used a simple strategy in evaluation: just add them as training positive examples. Figure 3 summarized two ways we utilized in training baseline classifiers. For the case in Figure 3(b), we name three baselines as:

- RF-P: Random Forest adding partial positives;
- SVM-P: Support Vector Machine adding partial positives;
- MLP-P: Multi-Layer Perceptron Neural Net adding partial positives;

We first discuss the experimental data setting and evaluation strategy used in performance comparison. Next we present evaluation results while comparing SML methods to several popular classifiers for determining protein interaction pairs between HIV-1 and human proteins.

All comparisons were based on five folds cross-validation (CV) with 20 randomly repeated CV runs to obtain average performance scores. The reason that we repeat cross-validation runs is that randomness exists when sampling the negative training set. To conqure this random effect, we pursued multiple CV runs on multiple independently sampled negative sets.

For each classifier, parameter optimization was carried out independently in identical cross-valiation fashion. Each method has distinct sets of parameters to tune. For SMEL, we need to learn the underline MLP network structure (hidden layer, hidden units, etc), the learning rate, choices of embedding pairs, ratio between embedding and classifier during the joint training. SMEO has a similar set of parameters as well. For SMLC, we need to learn the underline MLP network structure, ratio between the main classification and the pseudo classification, and the learning rate. For SMLR, we need to learn the underline MLP network structure, ratio between the main classification and the pseudo ranking, the learning rate and the choices of pairwise ranking pairs. To avoid overfitting, we did not try very deep MLP architecture. Thus either linear (if possible) or adding one hidden layer was tried for MLP architecutre. The best parameters found for the classification auxiliar model is with one hidden layer, 15 hidden units and learning rate 0.005. For the ranking model, the best setup is with linear linear layer with learning rate 0.01. For the SML "embedding output" model, the choice is one hidden layer, five hidden units, learning rate 0.005 and we train embedding with only the pushing apart step. For the SML "embedding on layer", the best setup involves one hidden layer, 8 hidden units, learning 0.005 and the embedding layer also has size 8.

4.3 Evaluation Metrics

When evaluating the performance of a classifier on an imbalanced test set such as is the case here, computing accuracy is not useful because a high true-negative (TN) rate can easily be obtained by chance. Therefore, we evaluated the quality of our predictive model using two metrics which ignore the success on the TN rate: the receiver operating curve (ROC) and precision vs. recall curve (Flach, 2004).

In Precision vs. Recall curves, precision refers to the fraction of interacting pairs predicted by the classifier that are truly interacting (âŁœtrue positivesâŁž). Recall measures how many of the known pairs of interacting proteins have been identified by the learning model. The Precision vs. Recall curve is then plotted for different cutoffs on the predicted score. Since it is hard to compare curves, Mean Average Precision (MAP) score is used to summarize the precision recall curve and is the mean of the Average Precision scores across recall levels. Among evaluation measures, MAP has been shown to have especially good discrimination and stability. We also report the Precision Recall breakpoint (PRB) score, the value of which means where precision is equal to recall.

Receiver Operator Characteristic (ROC) curves plot the true positive rate against the false positive rate for different cut-off values of the predicted score. The area under the ROC curve (AUC) is commonly used as a summary measure of diagnostic accuracy. AUC is interpreted as the probability that a randomly selected "event" will be regarded with greater suspicion (in terms of its continuous measurement) than a randomly selected "non-event". For our prediction task where positive class and negative class are extremely unbalanced, we are predominantly concerned with the detection performance of our models under conditions where the false positive rate is low. Considering the true positive examples (around 32 examples) in each cross-validation run's testing, we use 50 as a cut-off, i.e. R50 is a partial AUC score that measures the area under the ROC curve until reaching 50 negative predictions (in this case if all true positives are successfully detected, this achieves a recall 100% and precision 40%).

All these score range between 0 and 1, where values close to 1 indicates more successful predictions.

4.4 Performance

Table 2 lists the average Mean Average Precision (MAP), Precision Recall Breaking Point (PRB), partial AUC R50 and AUC scores from four proposed SML models and three baseline classifiers (each have two cases of training). The scores are averaged from 20 randomly repeated five-folds Cross-Validation runs.

For three baselines, the second type of training (to add partial labels) achieves better performance than the regular training, which is not surprising. MLP model (MAP-P 0.21) makes comparable performance to the state-of-the-art RF (MAP 0.213) model. SVM (MAP 0.156) does not performance so well on this task where we tried the linear and RBF kernels.

All SML models perform better than baseline strategies (on all metrics, with MAP 0.229 $\sim$ 0.277). This is expected since our partial positive examples are associated with keywords related to "protein-protein interactions". SML auxiliary tasks tried to capture the intrinsic patterns underlying these weak labels, from either these labels themselves, or their pairwise relationships with other examples. Multi-tasking with MLP improve the performance compared to MLP alone. We could conclude that SML achieves the state-of-art performance on the task of predicting interactions between HIV-1 and human protein.

Among four SML models, the SMLR-"auxiliary ranking" task and the SMEO-"embedding on output" task seem to capture the patterns of partial labels better compared the other two, e.g. SMLC and SMEL. We think this observation makes sense since essentially the only reliable assumption we could derive from weak positive labels is "partial positives are more likely to be interecting than negative random pairs". The ranking auxiliar task - SMLR performed training relying on this assumption exactly, which achieved the best R50 (0.310, 0.1 increase on RF) and the best AUC (0.919) scores. Under the best parameter setup (learned by CV), the "embedding output"-SMEO task is similar to SMLR where it tried to push the network output value of partial positives apart from the output of random negatives. This achieved the best MAP (0.277, about 0.07 increase on RF) and the best PRB (0.326, about 0.04 increase on RF) scores. The other two models could not capture this assumption directly, consequently resulting in less improvement from multi-tasking.

4.5 Validation

A final model was trained with all available expert labeled interactions using the best parameter setting we found for SML. Certain randomness exists when sampling the negative training set. Thus we utilized multiple independently sampled negative sets to conqure this random effect and to reduce the potential bias inherent in using a single training set. Through bagging models trained with five randomly sampled negative sets, our final score is obtained through value averaging.

We then ranked all HIV-1 to human protein pairs according to their classification score. The derived ranked order list were thresholded and the top ranked 1500 pairs build our list of predicted PPIs. For this PPI list we check if whether the human protein is reported in the functional siRNA screen (which screen indentified a set of genes to have an effect on HIV-1 infection (Brass *et al.*, 2008)). Also we check the human proteins in our top ranked PPI list, whether they have been detected in virion or not (Ott, 2008). The predicted pairs that involve with the virion related human genes would be of great interest to HIV-1 virologists. Due to the space limitation, the top ranked list and the related discussion are in our supplementary web: www.cs.cmu.edu/~qyj/HIVsemi.

5 CONCLUSIONS

Supervised learning methods have been used for the task of classifying pairs of proteins as interacting or not. However their performance is restricted by the availability of labeled training examples. In many cases, there exist considerable amount of pairs, where an association is proposed in the literature but not enough experimental evidence presented to support it as a direct interaction, for instance in our task of predicting human to HIV-1 protein interactome.

In this paper we design a semi-supervised multi-task learning framework to integrate a larger set of interacting protein-pairs retrieved from literature (partial labels) and a smaller set of interactions annotated by experts. The proposed SML combine a semi-supervised auxiliary task with the supervised PPI classifier. A multi-layer perceptron network is trained for PPI classification on *labeled* examples. Simultaneously we multi-task this network with an auxiliary task which aims to use partial positive labels to improve the supervised classification. Four auxiliary strategies are tried for the task of predicting interactions between HIV-1 and human proteins. Through cross-validations, our method was shown to improve upon the best previous method for this task indicating the benefits of multi-tasking with auxiliary information.

Beyond good performance on our task, the proposed SML framework is a flexible framework for general computational PPI prediction tasks. Besides partial labels, SML models could be easily extended to other species or to incorporate other auxiliary information, such as other kinds of partial labels or side information between unlabeled protein pairs. For instance, the noisy interaction pairs from high throughput experiments in human could be used to build neighbor pairs for training SML model (e.g. embedding on output) very naturally and this has great potentials for PPI predictions in human.

ACKNOWLEDGEMENTS

The authors would like to show sincere thanks to Chris Aiken from Department of Microbiology and Immunology, Vanderbilt University School of Medicine, for his trenmendous help during our expert labeling process.

REFERENCES

Ando, R. K. and Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, **6**, 1817–1853.

Ben-Hur, A. and Noble, W. (2005). Kernel methods for predicting protein-protein interactions. *Bioinformatics (Proceedings of the Intelligent Systems for Molecular Biology Conference)*, **21**, i38–i46.

Bleakley, K., Biau, G., and Vert, J.-P. (2007). Supervised reconstruction of biological networks with local models. *Bioinformatics*, **23**(13), i57–i65.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *COLT' 98*, pages 92–100.

Brass, A. L., Dykxhoorn, D. M., Benita, Y., Yan, N., Engelman, A., Xavier, R. J., Lieberman, J., and Elledge, S. J. (2008). Identification of host proteins required for hiv infection through a functional genomic screen. *Science*, **319**(5865), 921–926.

Breiman, L. (2001). Random forests. *Machine Learning*, **45**, 5–32. article.

Bromley, J., Guyon, I., Lecun, Y., Scklnger, E., and Shah, R. (1993). Signature verification using a siamese time delay neural network. In *In NIPS Proc*.

Caruana, R. (1997). Multitask learning. In *Machine Learning*, pages 41–75.

Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Chapelle, O., Schölkopf, B., and Zien, A., editors (2006). *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. MIT Press.

Collobert, R. and Weston, J. (2008). A unified architecture for nlp: deep neural networks with multitask learning. In *ICML'08*, pages 160–167.

Collobert, R., Sinz, F., Weston, J., and Bottou, L. (2006). Large scale transductive SVMs. *J. Mach. Learn. Res.*, **7**, 1687–1712.

Cusick, M. E., Yu, H., Smolyar, A., Venkatesan, K., Carvunis, A.-R., Simonis, N., Rual, J.-F., Borick, H., Braun, P., Dreze, M., Vandenhaute, J., Galli, M., Yazaki, J.,

Table 2. Performance Comparison (with multiple metric scores). SMLC: SML with classification task; SMLR: SML with ranking task; SMEO: SML with embedding on output; SMEL: SML with embedding on layer; RF: Random Forest; SVM: Support Vector Machine; MLP: Multi-Layer Perceptron Net. RF-P: RF adding partial positive; SVM-P: SVM adding partial positive; MLP-P: MLP adding partial positive.

Method	**R50 mean**	R50 std	**MAP mean**	MAP std	**PRB mean**	PRB std	**AUC mean**	AUC std
SMLC	0.277	0.07	0.263	0.06	0.312	0.06	0.905	0.04
SMLR	**0.310**	0.06	0.268	0.07	0.311	0.05	**0.919**	0.03
SMEO	0.309	0.07	**0.277**	0.06	**0.326**	0.06	0.908	0.03
SMEL	0.278	0.05	0.229	0.05	0.283	0.07	0.903	0.03
RF	0.169	0.03	0.135	0.02	0.180	0.04	0.893	0.02
RF-P	0.220	0.06	0.213	0.05	0.281	0.04	0.896	0.03
SVM	0.136	0.03	0.111	0.03	0.147	0.02	0.653	0.02
SVM-P	0.183	0.01	0.156	0.01	0.213	0.01	0.675	0.01
MLP	0.204	0.04	0.197	0.04	0.257	0.05	0.859	0.03
MLP-P	0.229	0.02	0.210	0.03	0.282	0.05	0.893	0.03

Hill, D. E., Ecker, J. R., Roth, F. P., and Vidal, M. (2008). Literature-curated protein interaction datasets. *Nature Methods*, **6**(1), 39–46.

Davis, F. P., Barkan, D. T., Eswar, N., McKerrow, J. H., and Sali, A. (2007). Host pathogen protein interactions predicted by comparative modeling. *Protein Sci*, **16**(12), 2585–2596.

Deng, M., Mehta, S., Sun, F., and Chen, T. (2002). Inferring domain-domain interactions from protein-protein interactions. *Genome Res.*, **12**(10), 1540–8. Their method is actually an EM-based MLE.

Evans, P., Dampier, W., Ungar, L., and Tozeren, A. (2009). Prediction of hiv-1 virus-host protein interactions using virus and host sequence motifs. *BMC medical genomics*, **2**(1).

Flach, P. (2004). The many faces of roc analysis in machine learning. *ICML-04 Tutorial*. article.

Fu, W., Sanders-Beer, B. E., Katz, K. S., Maglott, D. R., Pruitt, K. D., and Ptak, R. G. (2008). Human immunodeficiency virus type 1, human protein interaction database at ncbi. *Nucl. Acids Res.*, pages gkn708+.

Gavin, A., Aloy, P., Grandi, P., et al., and Superti-Furga, G. (2006). Proteome survey reveals modularity of the yeast cell machinery. *Nature*, **440**(7084), 631–6.

Gavin, A.-C., Bosche, M., Krause, R., et al., and Superti-Furga, G. (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, **415**(6868), 141–7.

Grandvalet, Y. and Bengio, Y. (2005). Semi-supervised learning by entropy minimization. In *NIPS'05*, pages 529–536.

Ho, Y., Gruhler, A., Heilbut, A., Bader, G. D., Moore, L., Adams, S.-L., et al., and Tyers, M. (2002). Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*, **415**(6868), 180–3.

Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., and Sakaki, Y. (2001). A comprehensive two-hybrid analysis to explore the yeast proteininteractome. *Proc. Natl. Acad. Sci. USA*, **98**(8), 4569–4574.

Jansen, R., Yu, H., Dreenbaum, D., Kluger, Y., and et. al. (2003). A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, **302**, 449–53.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *ICML '99*, pages 200–209.

Lee, I., Date, S., Adai, A., and Marcotte, E. (2004). A probabilistic functional network of yeast genes. *Science*, **306**(5701), 1555–8.

Lin, N., Wu, B., Jansen, R., Gerstein, M., and Zhao, H. (2004). Information assessment on predicting protein-protein interactions. *BMC Bioinformatics*, **5**, 154.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. volume 39, pages 103–134.

Ott, D. E. (2008). Cellular proteins detected in hiv-1. *Rev Med Virol*, **18**(3), 159–175.

Qi, Y., Klein-Seetharaman, J., and Bar-Joseph, Z. (2005). Random forest similarity for protein-protein interaction prediction from multiple sources. *Pacific Symposium on Biocomputing*, **10**, 531–542.

Qi, Y., Bar-Joseph, Z., and Klein-Seetharaman, J. (2006). Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *PROTEINS: Structure, Function, and Bioinformatics.*, **63**(3), 490–500.

Rhodes, D. R., Tomlins, S. A., Varambally, S., Mahavisno, V., Barrette, T., Kalyana-Sundaram, S., Ghosh, D., Pandey, A., and Chinnaiyan, A. M. (2005). Probabilistic model of the human protein-protein interaction network. *Nat Biotechnol.*, **8**, 951–9. article.

Rual, J. F., Venkatesan, K., et al., Roth, F. P., and Vidal, M. (2005). Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, **437**(7062), 1173–8. 1476-4687 (Electronic) Journal Article.

Scott, M. S. and Barton, G. J. (2007). Probabilistic prediction and ranking of human protein-protein interactions. *BMC Bioinformatics*, **8**, 239. 1471-2105 (Electronic) Comparative Study Journal Article Research Support, Non-U.S. Gov't.

Scudder, H. (1965). Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, **11**(3), 363–371.

Shoemaker, B. A. and Panchenko, A. R. (2007a). Deciphering protein-protein interactions. part i. experimental techniques and databases. *PLoS Comput Biol*, **3**(3), e42. 1553-7358 (Electronic) Journal Article Research Support, N.I.H., Intramural Review.

Shoemaker, B. A. and Panchenko, A. R. (2007b). Deciphering protein-protein interactions. part ii. computational methods to predict protein and domain interaction partners. *PLoS Comput Biol*, **3**(4), e43. 1553-7358 (Electronic) Journal Article Research Support, N.I.H., Intramural Review.

Stelzl, U., Worm, U., Lalowski, M., Haenig, C., Brembeck, F., et al., and Wanker, E. (2005). A human protein-protein interaction network: A resource for annotating the proteome. *Cell*, **122**(6), 830–2.

Tastan, O., Qi, Y., Carbonell, J., and Klein-Seetharaman, J. (2009). Prediction of interactions between hiv-1 and human proteins by information integration. In *Pacific Symposium on Biocomputing (PSB)*, volume 14.

Tastan, O., Carbonell, J. G., and Klein-Seetharaman, J. (2010). Refining literature curated protein-protein interactions through community opinio. *Submitted*.

Trkola, A. (2004 Oct). HIV-host interactions: vital to the virus and key to its inhibition. *Curr Opin Microbiol*, **7**(5), 555–559.

Uetz, P., Giot, L., Cagney, G., Mansfield, T. A., et al., and Rothberg, J. M. (2000). A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, **403**, 623–627.

von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., and Bork, P. (2002). Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, **417**(6887), 399–403.

Wang, H., Segal, E., Ben-Hur, A., Li, Q., Vidal, M., and Koller, D. (2007). InSite: A computational method for identifying protein-protein interaction binding sites on a proteome-wide scale. *Genome Biology*, **8**(9), R192.1–R192.18.

Weston, J., Ratle, F., and Collobert, R. (2008). Deep learning via semi-supervised embedding. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1168–1175, New York, NY, USA. ACM.

Yamanishi, Y., Vert, J., and Kanehisa, M. (2004). Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, **20**, 363–370.

Yip, K. Y. and Gerstein, M. (2009). Training set expansion: an approach to improving the reconstruction of biological networks from limited and uneven reliable interactions. *Bioinformatics*, **25**(2), 243–250.

Yu, H., Braun, P., et al., and Vidal, M. (2008). High-quality binary protein interaction map of the yeast interactome network. *Science*, **322**(5898), 104–110.

Zhang, L., Wong, S., King, O., and Roth, F. (2004). Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC Bioinformatics*, **5**, 38.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *ICML'03*, pages 912–919.