Multi-view prediction of protein function

Artem Sokolov
Department of Computer Science
Colorado State University
Fort Collins, CO 80523
sokolov@cs.colostate.edu

Asa Ben-Hur
Department of Computer Science
Colorado State University
Fort Collins, CO 80523
asa@cs.colostate.edu

ABSTRACT

The problem of predicting protein function using Gene Ontology terms is a hierarchical classification problem. There are a variety of genomic data that are relevant to a protein's function: its sequence, its interactions with other proteins, expression of its gene, etc. Some of these sources (interactions and expression) are species-specific, while protein sequence is comparable across species, which complicates the task of integrating labeled data from a target species with labeled data from other species. We address this problem using the methodology of structured output learning, present a framework based on multi-view learning that is naturally suited for combining both types of data, and demonstrate its effectiveness in making predictions for proteins in *S. cerevisiae* and *M. musculus*. The code for our framework is available at `http://strut.sourceforge.net`.

Keywords

Function prediction, multi-view learning, kernel methods

1. INTRODUCTION

The cost and effort involved in determining the function of a protein *in vivo* has led to the development of many computational methods aimed at predicting protein function from a variety of biological data, such as protein sequence, protein-protein interactions, and gene expression [21]. The Gene Ontology (GO) is the standard ontology used for specifying the function of a protein. It comprises a set of terms that belong to three separate hierarchies: molecular function, biological process and cellular component, where terms deeper in a hierarchy describe function in a more specific way. The GO consortium maintains GO terms and the hierarchies in which they are embedded [11]. In machine learning terms, prediction of GO terms can be formulated as a hierarchical multi-label classification problem [4].

Sequence or structural similarity has been the basis for protein function prediction for a long time [16], and is typically employed as a nearest-neighbor method—*transfer of annotation* from proteins with known functions with the help of alignment tools such as BLAST [1]. This approach works well when a clear signal of similarity exists, but is not able to effectively handle today's variety of noisy high-throughput biological data such as gene-expression, protein protein interactions and other genomic data that is informative of protein function [21]. This has led to the development of machine learning approaches that typically address the problem as a collection of binary classification problems: whether a protein should be associated with a given GO term (see e.g., [19]).

Several recent publications have presented methods for addressing GO term prediction as a hierarchical classification problem. Most of them train classifiers for individual GO terms and then reconcile their predictions with the hierarchical constraints via Bayesian networks or logistic regression [4, 20]. Mostafavi and Morris proposed extensions of their GeneMANIA method that directly predict a hierarchical labeling of a protein [18]. In parallel, we presented the GOstruct method, which directly predicts the full set of GO terms associated with a protein using structural SVMs [25]. *This method exhibits state-of-the-art performance in a comparison with several methods tested in the Mousefunc benchmark.* A similar approach was applied to prediction of enzyme function by Astikainen *et al.* [2].

The availability of a large variety of genomic data relevant to the task of protein function prediction has led to the development of a variety of methods for integrating those disparate data sources. Approaches include kernel methods [15, 20] or label propagation on a network whose nodes are proteins and edges indicate similarity according to some data source [9, 19, 27]. *All these methods perform data integration in a given species, and are not able to take into account the labels of annotated proteins in other species.* The challenge in doing this integration is that examples are heterogeneous—examples representing proteins in the given species have features that capture diverse data: gene expression, protein-protein interactions, and sequence similarity. Most of this data, except for sequence, is species-specific: protein interactions are probed experimentally in a given species, and the expression of a given gene measured in one set of experiments is difficult to compare meaningfully to expression measured in another species, under possibly different conditions.

In this paper, we explore extensions to the GOstruct framework that allow combining both species-specific features and cross-species features computed from sequence, to predict GO annotations in a given species. The proposed extensions

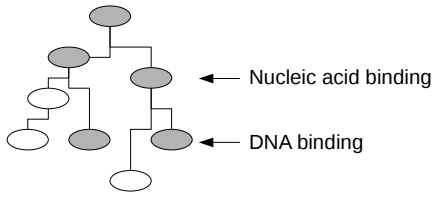

Figure 1: A schematic representation of the hierarchical label space for GO term annotation. Given that a protein is associated with a particular node in the GO hierarchy (e.g. DNA binding), it is also associated with all its ancestors in the hierarchy (including the direct parent of *DNA binding* which is *Nucleic acid binding*). The collection of GO terms associated with a protein (shaded in the figure) correspond to the nonzero entries in the vector representing the annotations.

are based on the idea of multi-view learning, which is an approach for dealing with multiple independent feature sets and unlabeled data. In multi-view learning, the input-space features are separated into two or more groups ("views") and a separate model is trained for each view with the goal of maximizing the accuracy on the labeled data and minimizing view disagreement on the unlabeled data [5]. The application of this technique to structured output spaces is fairly recent and several algorithms exist that either minimize the disagreement explicitly [10, 17] or use a more heuristic co-training approach where each view suggests labels for its peers [6].

Multi-view learning has been applied to natural-language processing [6], document categorization [10, 17] and signal processing [7]. However, all these applications maintain an implicit assumption that every example can be represented in every view. In this paper we break away from this assumption by treating all *cross-species* features, such as sequence similarity, as one view and all *species-specific* features, such as protein-protein interactions, as another. We explore co-training [5, 6] and transductive learning [28] as the two approaches to assigning labels to unlabeled data. We demonstrate that our multi-view framework, that combines all available sources of data, outperforms all single-view formulations in situations that simulate annotation of a newly-sequenced genome.

2. METHODS

We formulate the problem of GO term prediction as follows. Given a protein $\mathbf{x}$, we are interested in inferring its function $\mathbf{y}$ given by a vector of binary variables $(y_1, y_2, ..., y_m)$, where $y_i \in \{0, 1\}$ denotes whether a protein is annotated with the i^{th} GO term. A valid annotation $\mathbf{y}$ is one in which whenever a protein is annotated with a given GO term, it is also annotated with all of the term's ancestors in the GO hierarchy. Figure 1 shows an example of a valid label. We work with a set of n_l labeled training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n_l}$ where the inputs $\mathbf{x}_i$ belong to the space of proteins $\mathcal{X}$ and the labels $\mathbf{y}_i$ belong to the space of GO annotations, $\mathcal{Y}$.

We first describe the structured-output multi-view framework as it applies to labeled data only and then show how unlabeled data is incorporated. We consider two feature maps, $\phi^{(c)}(\mathbf{x})$ and $\phi^{(s)}(\mathbf{x})$. The first one is defined for all proteins and comprises a set of *cross-species features* that characterize the sequence of a protein. The second map, $\phi^{(s)}(\mathbf{x})$ is defined for proteins from a particular species only and comprises a set of *species-specific features* such as protein-protein interactions and gene expression data. Our goal is to leverage information from both feature maps to make predictions about the function of proteins in the target species. We note that each view will contain a different number of labeled examples, but in the interest of keeping the notation simple, we use a single variable n_l, with the understanding that its value will vary from one view to another.

In its basic formulation, a structured-output method learns a compatibility function $f(\mathbf{x}, \mathbf{y})$ between inputs and outputs; it infers a label as the label most compatible with a given input [26]:

$$\hat{\mathbf{y}} = h(\mathbf{x}) = \arg\max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}). \quad (1)$$

In what follows we make the assumption that the compatibility function is linear in a feature space defined by a feature map $\psi(\mathbf{x}, \mathbf{y})$ of both inputs and outputs:

$$f(\mathbf{x}, \mathbf{y}) = \mathbf{w}^T \psi(\mathbf{x}, \mathbf{y}). \quad (2)$$

In our multi-view setting we use two compatibility functions: $f^{(c)}$, which handles the cross-species view, and $f^{(s)}$, which handles the species-specific view. The inference is then performed according to

$$\hat{\mathbf{y}} = h(\mathbf{x}) = \arg\max_{\mathbf{y}} \left(f^{(c)}(\mathbf{x}, \mathbf{y}) + f^{(s)}(\mathbf{x}, \mathbf{y}) \right). \quad (3)$$

Each compatibility function, $f^{(c)}$ and $f^{(s)}$, is associated with its own feature map, which we denote by $\psi^{(c)}$ and $\psi^{(s)}$, respectively. We note that both feature maps $\psi^{(c)}$ and $\psi^{(s)}$ are functions of both inputs and outputs as is standard in structured-output methods. In our experiments we consider all pair-wise products of input-space features $\phi^{(c)}$ with the binary output variables y_i to generate $\psi^{(c)}$, and similarly for $\psi^{(s)}$. To avoid computing these feature maps directly, we work in the dual and make use of kernels. Given the input-space and output-space kernels

$$\begin{aligned} K_{\mathcal{X}}^{(c)}(\mathbf{x}_1, \mathbf{x}_2) &= \phi^{(c)}(\mathbf{x}_1)^T \phi^{(c)}(\mathbf{x}_2) \\ K_{\mathcal{X}}^{(s)}(\mathbf{x}_1, \mathbf{x}_2) &= \phi^{(s)}(\mathbf{x}_1)^T \phi^{(s)}(\mathbf{x}_2) \\ K_{\mathcal{Y}}(\mathbf{y}_1, \mathbf{y}_2) &= \mathbf{y}_1^T \mathbf{y}_2 - 1, \end{aligned}$$

we compute the joint kernel values for the cross-species view as

$$K^{(c)}\left((\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2)\right) = K_{\mathcal{X}}^{(c)}(\mathbf{x}_1, \mathbf{x}_2) K_{\mathcal{Y}}(\mathbf{y}_1, \mathbf{y}_2) \quad (4)$$

and, similarly, as

$$K^{(s)}\left((\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2)\right) = K_{\mathcal{X}}^{(s)}(\mathbf{x}_1, \mathbf{x}_2) K_{\mathcal{Y}}(\mathbf{y}_1, \mathbf{y}_2) \quad (5)$$

for the species-specific view. Our intuition is that two input-output example pairs have high similarity if they are similar in both input and output spaces.

To measure performance of structured-output methods, it is not enough to simply determine if the inferred label matches the true label, a measure otherwise known as the 0-1 loss. We must be able to differentiate between slight and gross misclassifications, because a prediction that differs from the true annotation by a single node deep in the

hierarchy is significantly better than a prediction made in an entirely different region. To capture this notion of accuracy we use the loss function

$$\Delta(\mathbf{y}_1, \mathbf{y}_2) = 1 - 2\frac{K_{\mathcal{Y}}(\mathbf{y}_1, \mathbf{y}_2)}{(K_{\mathcal{Y}}(\mathbf{y}_1, \mathbf{y}_1) + K_{\mathcal{Y}}(\mathbf{y}_2, \mathbf{y}_2))}, \quad (6)$$

which is essentially the F_1 measure expressed using kernels [25].

In addition to the kernel loss, we propose a method for computing ROC curves for our structured output methodology. This requires the definition of a confidence measure at the level of individual GO terms; the compatibility function provides a confidence measure at the level of a set of GO terms To allow us to compute ROC curves we define a confidence measure for predictions of GO term i as:

$$c_i(\mathbf{x}) = \max_{\mathbf{y} \in \mathcal{Y}_i^+} f(\mathbf{x}, \mathbf{y}) - \max_{\mathbf{y} \in \mathcal{Y}_i^-} f(\mathbf{x}, \mathbf{y}), \quad (7)$$

where $\mathcal{Y}_i^+ = \{\mathbf{y} \in \mathcal{Y} | y_i = 1\}$ is a subset of all labels that satisfy the hierarchical constraints and have the i^{th} variable set to 1. The subset $\mathcal{Y}_i^-$ is defined in a similar fashion, except with the i^{th} variable being set to 0. The values of $c_i(\mathbf{x})$ computed on test examples can be directly used to compute an ROC curve for GO term i.

When working with labeled data only, each view is trained independently of the other using the structured SVM formulation [26]:

$$\min_{\mathbf{w}, \xi} \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{C_l}{n}\sum_{i=1}^{n} \xi_i \quad (8)$$

$$\text{s.t.} \quad \xi_i \geq 0, \quad i = 1, \ldots, n_l,$$
$$\mathbf{w}^T\psi(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{w}^T\psi(\mathbf{x}_i, \mathbf{y}) \geq \Delta(\mathbf{y}, \mathbf{y}_i) - \xi_i,$$
$$i = 1, \ldots, n_l, \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i$$

where $\psi(\mathbf{x}_i, \mathbf{y}_i)$ is the feature map of the corresponding view. The margin violations are measured by the variables ξ_i, known as *slacks*. The above formulation is known as the margin rescaling version of the structured SVM; it allows higher compatibility function values for candidate labels that are closer to the truth. The parameter C_l controls the trade-off between margin magnitude and the amount of margin violations.

In addition to the multi-view method outlined above, we investigate an approach we call the *chain* classifier. In this approach, the predictions made by the cross-species classifier are incorporated into the species-specific feature map by adding a feature for each GO term. In other words, $\arg\max_{\mathbf{y}} f^{(c)}(\mathbf{x}_i, \mathbf{y})$ becomes a set of features in $\phi^{(s)}(\mathbf{x}_i)$. The inference made by the species-specific classifier is then reported as the overall prediction. Inference for both views is performed according to Equation (1).

2.1 Unlabeled Examples

In addition to the labeled data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n_l}$, we are also given unlabeled data $\{(\mathbf{x}_i)\}_{i=n_l+1}^{n_l+n_u}$. The objective of multi-view learning now becomes two-fold: maximize the accuracy on the labeled data and minimize the disagreement between views on unlabeled data [6]. Figure 2 presents the graphical overview of the approach. We require that all unlabeled examples span both views to make disagreement minimization possible in the absence of labels.

When dealing with labeled data, we aim to maximize the margin between the true label $\mathbf{y}_i$ and all other candidates. A similar principle holds for the unlabeled data. Given an unlabeled example, we would like to maximize the margin in compatibility between some label $\mathbf{z}_i$ and all other labels. Formally, for each view we would like to optimize

$$\min_{\mathbf{w}, \xi} \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{C_l}{n_l}\sum_{i=1}^{n_l} \xi_i + \frac{C_u}{n_u}\sum_{i=n_1+1}^{n_l+n_u} \xi_i \quad (9)$$

$$\text{s.t.} \quad \xi_i \geq 0, \quad i = 1, \ldots, n_l + n_u.$$
$$\mathbf{w}^T\psi(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{w}^T\psi(\mathbf{x}_i, \mathbf{y}) \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i$$
$$i = 1, \ldots, n, \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i$$
$$\exists \mathbf{z}_i \quad \mathbf{w}^T\psi(\mathbf{x}_i, \mathbf{z}_i) - \mathbf{w}^T\psi(\mathbf{x}_i, \mathbf{y}) \geq \Delta(\mathbf{z}_i, \mathbf{y}) - \xi_i$$
$$i = n_l + 1, \ldots, n_l + n_u, \mathbf{y} \in \mathcal{Y} \setminus \mathbf{z}_i.$$

We pursue two approaches that approximate a solution to this problem. The first approach follows the co-training algorithm, proposed by Brefeld *et al.* [6]. Each view suggests its most compatible label to be used as the "true" label $\mathbf{z}_i$ for the other view. The other view then updates its model based on the proposed label and makes its own suggestion to the first view. The process is repeated until consensus or until some number of iterations. The second approach is a generalization of the transductive structured SVM [28] to multi-view learning. The label $\mathbf{z}$ is simply inferred using the current model as $\mathbf{z}_i = \arg\max_{\mathbf{y}} \left(f^{(c)}(\mathbf{x}_i, \mathbf{y}) + f^{(s)}(\mathbf{x}_i, \mathbf{y})\right)$ [28].

2.2 Training and Inference

We have to address several issues associated with training of the proposed SVMs. One issue is that the size of the output space $\mathcal{Y}$ is exponential in the number of GO terms, which leads to an unmanageable number of constraints in Eqns (8) or (9). To deal with this complexity, we choose to focus only on those labels that appear in our dataset, arguing that we have the best chance to learn from combinations of GO terms that are biologically relevant [25].

Additionally, training follows the working set approach [6, 26], where a set of active constraints is maintained, and is grown incrementally by adding the most violated constraint at every iteration. The outer loop of the algorithm iterates over the training examples, both labeled and unlabeled. The inner loop that performs the model update is presented as Algorithm 1. This algorithm addresses training of all the SVM formulations considered here. The inner loop signals whether a new constraint has been added to the working set of a particular training example, and the outer loop terminates when no new constraints have been added after a full pass through the training data. Algorithms based around a working set are guaranteed to converge in a polynomial number of steps [6, 26]. In most of our experiments, the number of iterations did not exceed 50.

Algorithm 1 adds a new constraint to the working set only if it is violated by a larger amount than the current largest violation. We maintain a separate working set and a separate set of dual variables $\alpha_{i\mathbf{y}}$ for each view. We optimize the dual objective using a projection method by first finding the optimal solution in an unconstrained space and then projecting it to satisfy the constraints. The number of α variables associated with the working set of a single training

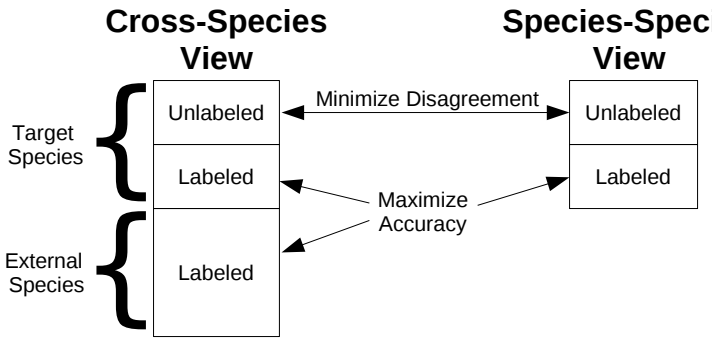

Figure 2: The multi-view approach. Data is separated into two views: a cross-species view that contains features computed from sequence, and a species-specific view that contains features computed from PPI data in the target species (*S. cerevisiae* or *M. Musculus*). The objective is to maximize the accuracy on the labeled data and minimize the disagreement on the unlabeled data.

example is usually fairly small (less than 50), and the projection method converges to the optimal solution faster than SMO-like algorithms [22], which are commonly employed for this task.

The only place in Algorithm 1 where the two views interact is during the inference of the label $\mathbf{z}$ for unlabeled examples. As mentioned above, we explore two ways of inferring the label $\mathbf{z}$. First is the transductive approach, which simply infers the most compatible label using the current model [28]. The second approach is the co-training algorithm proposed by Brefeld *et. al* [6]. There are several deviations from that algorithm, however.

Brefeld *et. al*'s algorithm cross-assigns the labels suggested by each view as "truth" for the peer view. In our experience, after the weights are updated, each view will correctly infer the label suggested to it by the peer view, but those labels are still in disagreement. So, the labels get cross-assigned again and the algorithm continues to alternate between the two states of label assignment, neither of which yields consensus. To get around this problem, we replace cross-assignment with a one-way assignment where the label suggested by the first view is given to the second view and, after the second view updates its weights, we verify that the new inference matches the suggestion. If it doesn't, then the second view suggests its label to the first view and the update is performed analogously.

Another deviation from the original algorithm is in the number of constraints added at every iteration. Brefeld *et. al* proposed to keep adding constraints for a particular training example until all constraints outside of the working set are violated by no more than the constraints in the working set [6]. Instead, we choose to add constraints until a consensus between the two views is reached. Once the consensus label $\mathbf{z}$ is obtained, at most one additional constraint is added by Algorithm 1. Further constraints are not included until the example is revisited again. Our intuition is two-fold: focusing entirely on a single unlabeled example before moving on to the next one is likely to skew the model towards the examples considered earlier; and adding a single violated constraint per iteration is more consistent with how we treat labeled examples and inferences from the transductive SVM, which allows for a cleaner comparison.

The final implementation issue is the order in which the training examples are traversed. We alternate between a full pass through the labeled data and a full pass through the unlabeled data. Interspersing unlabeled data in such a way prevents overfitting of the model to the labeled data and "guides" the model towards a state that better captures the general structure of the data. Note that it's not viable to completely randomize the order of example traversal, since there's a different number of labeled examples in each view.

3. EXPERIMENTAL SETUP

In each of our experiments we make predictions in a target species using data from that species and data from other species, which we call the external species. As target species we use *S. cerevisiae* and *M. musculus*. As external species for yeast we use *D. melanogaster* and *S. pombe*, and *H. sapiens* is used as an external species for mouse. We choose external species that are reasonably close to the target species and have a significant number of experimentally derived GO annotations.

GO Annotations.

We downloaded GO annotations from the Gene Ontology website (`http://www.geneontology.org`). Every GO annotation is accompanied by an evidence code that designates how it was obtained. We excluded all annotations that were obtained by computational predictions, as the inclusion of these annotations introduces prediction bias [23]. We limited our analysis to the following evidence codes: IDA, TAS, IMP, IGI, IPI, IEP, NAS, TC.

Sequence Data.

We used features based on protein sequence to construct the cross-species view. Protein sequences for all species were retrieved from the UniProt database (`http://uniprot.org`). In the cases where a gene has multiple splice forms, the longest one was used. Sequence features were extracted as follows.

BLAST hits.

We represented a protein in terms of its BLAST scores

Algorithm 1 Model update for a single example $\mathbf{x}_i$, for which a separate working set is maintained. The algorithm finds the most violated constraint using label $\mathbf{z}$, which is taken to be $\mathbf{y}_i$ for the labeled examples and inferred otherwise. If the new constraint is violated by a larger amount than the constraints already in the working set, it is added to the working set and the dual objective variables are updated using a projection algorithm.

Input: Training example $\mathbf{x}_i$, precision ϵ.
Output: Whether a new constraint has been added.
Define the current working set $\mathcal{W}_i = \{\mathbf{y} | \alpha_{i\mathbf{y}} \neq 0\}$.
if $\mathbf{x}_i$ is labeled **then**
  Define $\mathbf{z} = \mathbf{y}_i$.
else if using co-training **then**
  repeat
    Alternate between each view suggesting $\mathbf{z}$ [6]
  until Consensus is reached or r_{max} iterations.
else if using transduction **then**
  Define $\mathbf{z} = \arg\max_{\mathbf{y}} \left(f^{(c)}(\mathbf{x}_i, \mathbf{y}) + f^{(s)}(\mathbf{x}_i, \mathbf{y}) \right)$
end if
If $\mathbf{z}$ changed since the last iteration, clear the working set.
for each view $v = \{c, s\}$ **do**
  Find the largest margin violation and the associated slacks:
  $\bar{\mathbf{y}} \leftarrow \arg\max_{\mathbf{y} \in \mathcal{Y} \setminus \mathbf{z}} f^{(v)}(\mathbf{x}_i, \mathbf{y})$
  $\xi_i \leftarrow \max^{+}_{\mathbf{y} \in \mathcal{W}_i} (\Delta(\mathbf{z}, \mathbf{y}) - f^{(v)}(\mathbf{x}_i, \mathbf{z}) + f^{(v)}(\mathbf{x}_i, \mathbf{y})$
  $\bar{\xi} \leftarrow \max \left\{ 0, (\Delta(\mathbf{z}, \bar{\mathbf{y}}) - f^{(v)}(\mathbf{x}_i, \mathbf{z}) + f^{(v)}(\mathbf{x}_i, \bar{\mathbf{y}})) \right\}$
  if $\bar{\xi} > \xi_i + \epsilon$ **then**
    Add the constraint to the working set: $\mathcal{W}_i \leftarrow \mathcal{W}_i \cup \{\bar{\mathbf{y}}\}$
    Optimize the dual objective over the working set $\mathcal{W}_i$ keeping $\alpha_{j\mathbf{y}}$ fixed for $j \neq i$.
  end if
end for

against a database of annotated proteins [1]. This representation is known as the empirical kernel map [24]. We performed all-vs-all BLAST and the output was post-processed by excluding all hits with e-values above 50.0. The remaining e-values were divided by 50.0 to normalize them. Any values below 1e-10 after normalization were brought up to 1e-10. We then use the negative log of the resulting values as features.

Localization signals.

We computed features that capture protein localization signals using the WolfPsort program [12].

Transmembrane protein predictions.

For each protein we obtained predictions of the number of transmembrane domains using the TMHMM program [14], and an indicator variable was associated with each number of transmembrane domains.

K-mer composition of N and C termini.

The N and C termini ends of a protein contain signals that are important for protein localization, binding and other protein functions [3]. We computed features that represent the 3-mer composition of 10 amino acid segments in the N and C termini of each protein.

Target Species	*S. cerevisiae*			*M. musculus*		
Namespace	MF	BP	CC	MF	BP	CC
# Target	3401	4332	4115	3150	2633	2125
# External	3917	3000	5000	5000	3000	5000
# GO terms	317	946	308	310	1697	240

Table 1: The number of proteins in the target and external species, as well as the number of GO terms considered in each dataset. Namespace designations are as follows: MF - molecular function; BP - biological process; CC - cellular component.

Low complexity regions.

Low-complexity regions in proteins are abundant, have an effect on protein function and are not typically captured by standard sequence comparison methods [8]. Each protein is scanned with a sliding window of size 20, and a low-complexity segment is defined as the window that contains the smallest number of distinct amino acids. We use the amino acid composition of that segment as features.

Interaction Data.

We used *S. cerevisiae* and *M. musculus* protein-protein interaction (PPI) data from STRING 8.3 [13] for species-specific information. A protein is represented by a vector of variables, where component i indicates the STRING evidence score of an interaction between protein i and the given protein.

Dataset.

The data pre-processing steps provided a certain number of target proteins that have features in both views. Five-fold cross-validation is performed on this set of proteins. Additional proteins, with cross-species features only, were obtained from the external species *D. melanogaster*, *S. pombe* and *H. sapiens*. Table 1 provides several statistics about each dataset. In the interest of keeping the run times down, we randomly subsampled the external set down to 5000 proteins for molecular function and cellular component experiments and down to 3000 proteins for biological process experiments.

Kernels.

We used linear kernels in both input and output spaces. All kernels were then normalized according to

$$K(z_1, z_2) = \frac{K(z_1, z_2)}{\sqrt{K(z_1, z_1) K(z_2, z_2))}}.$$

Multiple sets of features were combined via unweighted kernel summation.

Model Selection and Classifier Assessment.

In each target species, classifier performance was estimated using five-fold cross-validation on the proteins that have features in both views; folds were randomly selected such that no two proteins from different folds have more than 50% sequence identity. To select appropriate values for the parameters C_l and C_u, we ran four-fold cross-validation on the training data in each experiment. The values of $\frac{C_l}{n_l} = 1$ and $\frac{C_u}{n_u} = 0.1$ yielded the highest accuracy on the validation

	Kernel Loss					
Target species	*S. cerevisiae*			*M. musculus*		
Namespace	MF	BP	CC	MF	BP	CC
Cross-Species	0.48	0.55	0.32	0.35	0.60	0.32
Species-Specific	0.44	0.35	0.21	0.38	0.55	0.30
Joint	0.34	0.35	0.20	0.32	0.54	0.28
Multi-view	0.34	0.34	0.22	0.30	0.53	0.27
Chain	0.33	0.34	0.20	0.31	0.55	0.27
BNN-Chain	0.33	0.35	0.20	0.32	0.55	0.28
	AUC					
Target species	*S. cerevisiae*			*M. musculus*		
Namespace	MF	BP	CC	MF	BP	CC
Cross-Species	0.87	0.79	0.78	0.89	0.67	0.80
Species-Specific	0.90	0.94	0.94	0.83	0.81	0.84
Joint	0.94	0.94	0.95	0.88	0.80	0.85
Multi-view	0.95	0.94	0.94	0.90	0.79	0.88
Chain	0.94	0.94	0.95	0.90	0.82	0.87
BNN-Chain	0.94	0.94	0.95	0.89	0.82	0.87

Table 2: Classifier performance in predicting GO terms, quantified by mean loss per example (top) and mean AUC per GO term (bottom) when no unlabeled data is used. Lower loss values and higher AUC values are better. The results were obtained via five-fold cross-validation on all proteins from the target species. Multi-view and cross-species SVMs were also provided with the training examples from external species.

	Kernel Loss			
	# Training Samples			
Classifier	2720	1500	1000	500
Cross-Species	0.48	0.49	0.51	0.51
Species-Specific	0.44	0.48	0.52	0.56
Joint	0.34	0.40	0.44	0.52
Multi-view	0.34	0.37	0.40	0.43
Chain	0.33	0.36	0.39	0.45
BNN-Chain	0.33	0.38	0.41	0.48
	AUC			
	# Training Samples			
Classifier	2720	1500	1000	500
Cross-Species	0.87	0.86	0.84	0.84
Species-Specific	0.90	0.87	0.85	0.80
Joint	0.94	0.91	0.89	0.83
Multi-view	0.95	0.94	0.92	0.90
Chain	0.94	0.92	0.91	0.88
BNN-Chain	0.94	0.93	0.91	0.87

Table 3: Classifier performance in predicting molecular function GO terms, quantified by mean loss per example (top) and mean AUC per GO term (bottom) when no unlabeled data is used. Lower loss values and higher AUC values are better. The number of training examples refers to *S. cerevisiae* proteins that are represented in both views. Multi-view and Cross-Species SVMs were provided the additional 3917 proteins that only have BLAST features.

set almost universally.

4. RESULTS

4.1 Impact of Cross-Species Information

The first experiment is designed to illustrate the improvement in prediction accuracy we obtain by introducing information from other species in the absence of unlabeled data. The multi-view SVM in this case combines the cross-species and species-specific SVMs that are trained separately; both models are used together for inference, as per Equation (3). The chain classifier first trains a structured SVM on the cross-species view; it then incorporates the predictions made by this SVM into the input-space feature map for the species-specific view. A second SVM, trained on these predictions combined with the PPI data, is then applied to the test set. We consider a variant of the chain classifier that uses a BLAST-nearest-neighbor (BNN) approach to perform the cross-species prediction instead of a structured SVM. We refer to this classifier variant as "BNN-Chain". In preliminary experiments the structured SVM provided more accurate predictions than the BNN approach [25], but the BNN approach is more scalable to the large datasets that can be used for the cross-species classifier. As a baseline, we trained a single structured-output SVM, which we call joint-SVM, on target species data only, combining the features from both views. Additionally, we trained two single-view SVMs: one using exclusively cross-species information and one using exclusively species-specific features.

We observe that performance as measured by the kernel loss is correlated with the number of variables in the output space: the best results in Table 2 are observed in the cellular component namespace, which has the smallest number of GO terms being considered (Table 1). The biological process namespace contains about three times as many GO terms and generally leads to higher loss values observed in Table 2. Because the AUC values are computed on a term-by-term basis, they fail to measure the interdependence of GO terms in a prediction and appear much more uniform across the three namespaces.

The results in Table 2 demonstrate the advantage of the multi-view and chain approaches: these classifiers achieve the lowest loss and highest AUC of all the methods, and achieve higher performance than either view by itself. The BNN-chain classifier achieves slightly worse performance than the chain classifier that uses the structured SVM; however, the ability to use this classifier with larger much larger external species datasets makes it a highly viable approach. The species-specific classifier that uses only PPI data performs better in yeast than in mouse, which is attributed to the better characterization of its interaction network. For the cross-species classifier we see the opposite effect, with the mouse classifier exhibiting better accuracy than the yeast classifier. The accuracy of the cross-species view has to do with how well annotated are closely related species (*S. pombe* in the case of *S. cerevisiae*, and *H. sapiens* in the case of *M. musculus*).

To further investigate the interplay between cross-species and species-specific information, we ran additional experiments, reducing the number of target-species proteins while keeping the set of external proteins fixed. We present the results of these experiments on molecular function in *S. cerevisiae* in Table 3. As expected, the cross-species in-

formation becomes more important as the number of training examples in the target species decreases. In particular, the cross-species SVM outperforms both the species-specific and the joint SVMs when the number of *S. cerevisiae* proteins is 500, a scenario that more closely simulates annotating a newly-sequenced genome. We note that, in all cases, the multi-view and chain classifiers outperform all other methods, demonstrating their robustness in combining cross-species and species-specific information. We further observe that as the cross-species features become more relevant, proper utilization of those features becomes important; this is signified by the BNN-based chain classifier degrading in performance faster than the SVM-based chain classifier.

4.2 Impact of Unlabeled Data

The second set of experiments is designed to measure the impact of unlabeled data. We ran five-fold cross-validation using the same test data in every fold as above. The *S. cerevisiae* training data for every experiment was split into labeled and unlabeled examples. Similar to experiments in the previous subsection, we include all labeled proteins that only have feature representation in the cross-species view.

# examples		Loss		AUC	
# Lbld	# Ulbld	CO-tr.	Trans.	CO-tr.	Trans.
500	0	0.43	0.43	0.90	0.90
500	500	0.50	0.62	0.86	0.77
500	1000	0.65	0.66	0.75	0.74
500	1500	0.67	0.77	0.75	0.66
500	2000	0.71	0.77	0.72	0.63
1000	0	0.40	0.40	0.92	0.92
1000	500	0.42	0.44	0.91	0.89
1000	1000	0.43	0.57	0.91	0.81
1000	1500	0.42	0.62	0.91	0.77

Table 4: Mean loss per example for co-training and transductive SVMs computed for various numbers of labeled and unlabeled *S. cerevisiae* training examples. The number of non *cerevisiae* proteins was the same in all cases. The test data used in these experiments was identical to that used in Table 2.

As shown in Table 4 the addition of unlabeled data had negative impact on classifier performance. The co-training approach appears to be affected less severely, but both semi-supervised algorithms fail to learn from unlabeled data. We conjecture that this is due to the sparsity with which the joint input-output space is characterized by the labeled examples.

5. CONCLUSION

We introduced a multi-view framework for protein function prediction that integrates heterogeneous sources of data: species-specific information and features that capture protein similarity across multiple species. In the absence of unlabeled data, we demonstrated its advantage over classifiers that don't have access to cross-species information. We observed degraded performance when incorporating unlabeled data, with less degradation observed with the co-training approach. Our approach offers flexibility with regards to data availability: for newly sequenced genomes where no other high-throughput data is available the cross-species classifier can be applied; in more well studied species the full framework can be used. We are making the code available at `http://strut.sourceforge.net`. In future work we will supplement the species-specific view with additional sources of data.

6. REFERENCES

[1] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *J. Mol. Biol*, 215(3):403–410, 1990.

[2] K. Astikainen, L. Holm, E. Pitkanen, S. Szedmak, and J. Rousu. Towards structured output prediction of enzyme function. *BMC proceedings*, 2(Suppl 4):S2, 2008.

[3] I. Bahir and M. Linial. Functional grouping based on signatures in protein termini. *Proteins: Structure, Function, and Bioinformatics*, 63(4):996–1004, 2006.

[4] Z. Barutcuoglu, R.E. Schapire, and O.G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830, 2006.

[5] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, page 100. ACM, 1998.

[6] U. Brefeld and T. Scheffer. Semi-supervised learning for structured output variables. In *Proceedings of the 23rd international conference on Machine learning*, pages 145–152. ACM, 2006.

[7] C.M. Christoudias, R. Urtasun, and T. Darrell. Multi-view learning in the presence of view disagreement. In *UAI*, page 5, 2008.

[8] A. Coletta, J.W. Pinney, D.Y.W. Solís, J. Marsh, S.R. Pettifer, and T.K. Attwood. Low-complexity regions within protein sequences have position-dependent roles. *BMC systems biology*, 4(1):43, 2010.

[9] M. Deng, T. Chen, and F. Sun. An integrated probabilistic model for functional prediction of proteins. In *RECOMB*, pages 95–103, 2003.

[10] K. Ganchev, J. Graca, J. Blitzer, and B. Taskar. Multi-view learning over structured and non-identical outputs. In *Proceedings of The 24th Conference on Uncertainty in Artificial Intelligence*. Citeseer, 2008.

[11] Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nat. Genet.*, 25(1):25–9, 2000.

[12] P. Horton, K.J. Park, T. Obayashi, and K. Nakai. Protein subcellular localization prediction with WoLF PSORT. In *Proceedings of the 4th annual Asia Pacific bioinformatics conference APBC06, Taipei, Taiwan*, volume 39, page 48. Citeseer, 2006.

[13] L.J. Jensen, M. Kuhn, M. Stark, S. Chaffron, C. Creevey, J. Muller, T. Doerks, P. Julien, A. Roth, M. Simonovic, et al. STRING 8.a global view on proteins and their functional interactions in 630 organisms. *Nucleic acids research*, 37(suppl 1):D412, 2009.

[14] A. Krogh, B.È. Larsson, G. Von Heijne, and E.L.L. Sonnhammer. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes1. *Journal of molecular biology*, 305(3):567–580, 2001.

[15] H. Lee, Z. Tu, M. Deng, F. Sun, and T. Chen. Diffusion kernel-based logistic regression models for protein function prediction. *OMICS: A Journal of Integrative Biology*, 10(1):40–55, 2006.

[16] Y. Loewenstein, D. Raimondo, O. Redfern, J. Watson, D. Frishman, M. Linial, C. Orengo, J. Thornton, and A. Tramontano. Protein function annotation by homology-based inference. *Genome Biology*, 10(2):207, 2009.

[17] B. Long, P.S. Yu, and Z.M. Zhang. A general model for multiple view unsupervised learning. In *Proceedings of the 8th SIAM International Conference on Data Mining (SDM'08), Atlanta, Georgia, USA*. Citeseer, 2008.

[18] S. Mostafavi and Q. Morris. Using The Gene Ontology Hierarchy when Predicting Gene Function. In *Conference on Uncertainty in Artificial Intelligence*, 2009.

[19] S. Mostafavi, D. Ray, D. Warde-Farley, C. Grouios, and Q. Morris. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, 9(Suppl 1):S4, 2008.

[20] G. Obozinski, G. Lanckriet, C. Grant, M. Jordan, and W. Noble. Consistent probabilistic outputs for protein function prediction. *Genome Biology*, 9(Suppl 1):S6, 2008.

[21] L. Peña-Castillo, M. Tasan, C. Myers, H. Lee, T. Joshi, C. Zhang, Y. Guan, M. Leone, A. Pagnani, W. Kim, et al. A critical assessment of *Mus musculus* gene function prediction using integrated genomic evidence. *Genome Biology*, 9(Suppl 1):S2, 2008.

[22] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Advances in Kernel Methods-Support Vector Learning*, 208, 1999.

[23] M.F. Rogers and A. Ben-Hur. The use of Gene Ontology evidence codes in preventing classifier assessment bias. *Bioinformatics*, 25(9):1173, 2009.

[24] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[25] A. Sokolov and A. Ben-Hur. Hierarchical classification of Gene Ontology terms using the GOstruct method. *Journal of Bioinformatics and Compuational Biology*, 8(2):357–376, 2010.

[26] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(2):1453, 2006.

[27] K. Tsuda, H.J. Shin, and B. Schölkopf. Fast protein classification with multiple networks. In *ECCB*, 2005.

[28] A. Zien, U. Brefeld, and T. Scheffer. Transductive support vector machines for structured variables. In *Proceedings of the 24th international conference on Machine learning*, page 1190. ACM, 2007.