Semi-Supervised Learning of Alternative Splicing Events Using Co-Training

Karthik Tangirala
CIS Department
Kansas State University
Manhattan, Kansas
Email: karthikt@ksu.edu

Doina Caragea
CIS Department
Kansas State University
Manhattan, Kansas
Email: dcaragea@ksu.edu

***Abstract*—Alternative splicing is a phenomenon that gives rise to multiple mRNA transcripts from a single gene. It is believed that a large number of genes undergoes alternative splicing. Predicting alternative splicing events is a problem of great interest to biologists, as it can help them to understand transcript diversity. Supervised machine learning approaches can be used to predict alternative splicing events at genome level. However, supervised approaches require large amounts of labeled data to learn accurate classifiers. While large amounts of genomic data are produced by the new sequencing technologies, labeling these data can be costly and time consuming. Therefore, semi-supervised learning approaches that can make use of large amounts of unlabeled data, in addition to small amounts of labeled data are highly desirable. In this work, we study the usefulness of a semi-supervised learning approach, co-training, for classifying exons as alternatively spliced or constitutive. The co-training algorithm makes use of two views of the data to iteratively learn two classifiers that can inform each other, at each step, with their best predictions on the unlabeled data. We consider three sets of features for constructing views for the problem of predicting alternatively spliced exons: lengths of the exon of interest and its flanking introns, exonic splicing enhancers and intronic regulatory sequences. Naive Bayes and Support Vector Machine (SVM) algorithms are used as based classifiers in our study. Experimental results show that the usage of the unlabeled data can result in better classifiers as compared to those obtained from the small amount of labeled data alone.**

***Keywords*-alternative splicing, semi-supervised learning, co-training.**

I. Introduction

Machine Learning [1] is a branch of artificial intelligence focused on the design and development of algorithms for learning models from data. In a supervised learning scenario, a model is learned from labeled data and used to predict new unseen data. Predicting a new problem can be a difficult task even for humans. However, humans use past experiences to gain knowledge on a particular problem and, then, they predict it. Similarly, computers need existing known data or examples to capture "past experiences." For supervised learning, the examples are pairs of objects and their corresponding labels, generally referred to as training examples. Using the training examples, a machine learning algorithm infers a classification or a regression function, a.k.a. model, which is further used to predict new unlabeled examples.

Supervised machine learning has been used in a variety of domains including text classification, natural language processing, social networking, and bioinformatics, among others. However, the success of supervised learning depends on the availability of large amounts of labeled data. For many application domains, the amount of labeled data is very limited, while large amounts of unlabeled data are easily available. This motivates the need for semi-supervised learning algorithms [2], which can make use of large amounts of unlabeled data, together with small amounts of labeled data, to learn models that can accurately predict new, unseen data. Examples of semi-supervised algorithms include co-training [3] and EM [4]. Semi-supervised learning algorithms have received a lot of attention in the last few years. They have been successfully used in several application domains including text classification [4], [5], natural language processing [6], and sentiment categorization [7].

However, semi-supervised learning algorithms have not been much studied in the bioinformatics domain, with a few notable exceptions [8], [9], [10]. Given the recent advances in next generation sequencing technologies, large amounts of genomics sequence data are produced. Labeling these data can be very expensive. Semi-supervised learning algorithms that can make use of unlabeled data are greatly needed in bioinformatics. To address this need, the work proposed in this paper is focused on the *study of semi-supervised algorithms in the context of bioinformatics classification problems*. Specifically, we consider the problem of predicting *alternative splicing events*.

Genes undergo transcription and translation in the process of protein synthesis. Splicing is a stage in between transcription and translation, in which the coding regions are separated from the non-coding regions to form mRNA. Alternative splicing is a mechanism by which multiple mRNA transcripts are generated from a single gene [11], and can, thus, be seen as an important mean for increasing proteome diversity. Several types of alternative splicing events are known (e.g., exon skipping and intron retention). In this work, we will focus on predicting alternatively spliced exons using co-training based semi-supervised approaches. Co-training requires that data is represented according to two views, sufficient for classification and independent of

each other given the class [3], to iteratively learn classifiers that inform each other with their best predictions on the unlabeled data. We will use intronic and exonic splicing motifs and length features to represent data instances in two different views, and use co-training to learn to discriminate between alternative spliced exons and constitutive exons.

The rest of this paper is organized as follows: Section II contains a brief discussion of the related work. Section III describes in detail the design and implementation of the co-training algorithm. In Section IV, we describe the data used in our study and the feature set used for representing the data. Section V provides the set of experiments performed on the data using co-training algorithm. Section VI presents the results obtained for our experiments. Section VII summarizes the conclusions drawn from this work. Several ideas for future work are presented in Section VIII.

II. Related Work on Co-Training Based Semi-Supervised Learning

Co-training is a semi-supervised learning algorithm which utilizes the knowledge of labeled data together with unlabeled data through an iterative transfer of knowledge between two different classifiers. It was first introduced by Blum and Mitchell [3] in the context of web page classification. In their work, data is represented in two views, consisting of the words occurring on a page and the words occurring in the hyperlinks corresponding to the page, respectively. Co-training makes use of a small amount of labeled data and a large amount on unlabeled data, represented in two views, to learn two naive Bayes classifiers that can be used together to accurately predict new unseen data. Experimental results show that co-training has a smaller error rate as compared to the supervised implementation that makes use only of the labeled data.

Nigam and Ghani [12] study the effectiveness of co-training by applying it to several text data sets, such as WebKB course dataset [3], and two subsets of the 20 Newsgroup datasets. The authors compare co-training to another semi-supervised learning approach, specifically the EM algorithm [4], which finds locally maximum parameter estimates without splitting the features into views. Their results show that, generally, co-training gives better results than EM, especially when the two views use are independent and redundant, but also when the two views may not correspond to independent sets of features (both algorithms perform better than the supervised naive Bayes algorithm that uses only the labeled data). The authors suggest that semi-supervised algorithms that explicitly use an independent and sufficient feature split perform better than those that use all features together. Furthermore, they suggest that co-training is more robust to its assumptions than the EM.

Du et al. [13] further study the independence and sufficiency assumptions that co-training makes, and conclude, once again, that co-training works very well if the assumptions are satisfied. Furthermore, they propose an empirical approach for verifying that the assumptions are satisfied for two given views, and also several methods for splitting a single view into two views that satisfy the assumptions. However, their proposed verification and splitting methods are shown to be reliable only when large amounts of labeled data are available, which is not the case when using co-training, thus, making the proposed methods impractical.

Kiritchenko and Matwin [14] evaluate the performance of co-training when various base classifiers such as Naive Bayes and SVM are used. Their results show that, for the problem of classifying emails, co-training with SVM as a base classifier outperforms co-training with Naive Bayes multinomial as a base classifier. The results also show that, for unbalanced data problems, balancing the training set has as effect an increase in the performance of co-training.

In the context of bioinformatics, Xu et al. [15] use a CoForest semi-supervised learning approach to predict protein localization. The CoForest approach, which was first proposed in [16], is similar to the co-training approach, except that it builds a set (forest) of classifiers (trees), which can collectively be used to find the most confident predictions among the unlabeled examples. Experimental results show that the CoForest approach outperforms several baselines including decision tree, AdaBoost and state-of-the-art SVM classifiers that make use of labeled data only.

As opposed to the previous work described above, in this paper we use co-training for another bioinformatics problem, specifically alternative splicing prediction. We will next review the co-training algorithm that we implemented.

III. Co-Training based Semi-Supervised Learning

As mentioned above, the main idea of co-training is to represent data using two views, where each view is defined by a subset of features that can be used to describe the data. The views are assumed to be independent and sufficient. Two different classifiers are learned from the two views and their predictions on the unlabeled data are used to iteratively improve each other, as explained below. The process is graphically depicted in Figure 1.

We denote by $V1$ and $V2$ two views of the available data. Let $L1$ and $L2$ denote the labeled data represented in the views $V1$ and $V2$, respectively. Similarly, unlabeled data in the two views is denoted by $U1$ and $U2$, respectively. Using the known labeled data $L1$ and $L2$, two different classifiers $C1$ and $C2$ are learned. The two classifiers are used to predict the unlabeled data $U1$ and $U2$. As a result, all the unlabeled instances are classified with a certain confidence. In the following step, the most confident predictions from $U1$ are added to the training set of $C2$. Similarly, the most confident predictions from $U2$ are added to the training set of $C1$. Instances added to the training set are deleted from the unlabeled instance set. After adding the best predictions

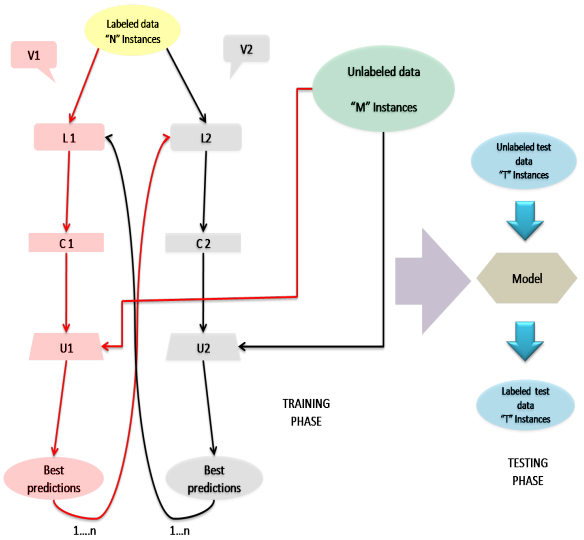


Figure 1. Brief overview of the co-training algorithm. Classifiers C1 and C2 can be any base classifiers, e.g. Naive Bayes and SVM classifiers.

to the training set of the opposite view classifier, two new classifiers are learned. The process is repeated for a certain number of iterations or until all the unlabeled data is utilized by the algorithm. The last two classifiers learned can be used together to predict new test data. That is done by adding up the class probabilities produced by the two classifiers and selecting the most probable class.

Adding the best predictions to the original training data has as effect on the improvement in the classifiers at each iteration. Intuitively, instances which are best predicted by one classifier in one view, may not be well predicted by the classifier in the other view. Transferring information between classifiers can increase the performance of both classifiers.

We explore the use of this particular algorithm on biological data and study its ability to outperform supervised algorithms trained on small amounts of labeled data. One reason for using this particular algorithm is because of its capability to make use of information from two independent views of data. For our problem, the data available can be represented using various combinations of features.

IV. Data and Feature Sets Used

To study the capability of co-training to learn to predict alternatively spliced exons from small amounts of labeled data, we use the alternatively spliced exon dataset from [17], where each instance consists of an internal exon and its flanking introns. More precisely, for each instance, we used a ± 200 window on each side of the acceptor and donor sites (acceptors mark the end of introns, while donors mark the beginning of exons). As a result, instances can be seen as sequences of 802 nucleotides labeled with the class to which they belong. Class "spliced" defines exons that are alternative spliced (present in some transcripts, but not in all), while class "constit" defines exons that are not alternatively spliced (always present in transcripts). The dataset contains a total of 3018 instances, 487 in the "spliced" class and 2531 in the "constit" class.

Many machine learning algorithms assume data to be represented in the form of feature vectors, which means that we need to construct features for our sequence instances. Supervised learning of alternative splicing has shown that the lengths of an exon and its flanking introns can be used as predictive features. In addition to length features, splicing regulators are highly responsible for the occurrence of alternative splicing events, and have been used as predictive features. These splicing regulators can occur in both exons and introns. Enhancing splicing regulators that occur in exons are called ESE motifs. Intronic regulatory sequences which frequently occur in the intronic regions of genes are called IRS motifs. Therefore, in this work, we will represent instances using length and motif features.

A. Length Features

The lengths of spliced exons and their flanking introns are generally different from the lengths of the constitutive exons and their flanking introns [18]. As a consequence, length information can be used to discriminate between the two classes of exons. A feature representation which captures the length information, by considering 30 logarithmically spaced bins was defined in [17], and is used in this work.

B. Exonic Splicing Enhancers (ESE motifs)

In this work, we used a set of 45 ESE motifs derived in [19]. The ESE motifs represent hexamers (i.e., sequences of length 6 base pairs), which occur more frequently in exons than in introns. We used a sliding window to obtain the counts of the 45 ESE motifs in a particular sequence. This gives rise to a vector of length 45 per instance.

C. Intronic Regulatory Sequences (IRS motifs)

We used the set of IRS motifs that was previously used in [19]. These motifs have been derived based on the observation that intronic sequences that are relevant for alternative splicing are highly conserved among closely related species. To form the set of IRS motifs, we combined both the upstream and downstream motifs and removed the duplicate motifs. This resulted in a total of 165 IRS motifs which are assumed to be informative for alternative splicing prediction. As in the case of the ESE motifs, we considered a count representation for the IRS motifs in every sequence. This gives rise to a vector of length 165 per instance.

Thus, our final set of features consists of three subsets: a subset corresponding to length features (real numbers), a subset corresponding to ESE motifs (counts) and a subset corresponding to IRS motifs (counts). We construct several view combinations for co-training, as shown in Table I.

Table I
COMBINATIONS OF VIEWS USED WITH CO-TRAINING

Combination #	View 1	View 2
1	IRS	LENGTH
2	ESE	LENGTH
3	IRS + ESE	LENGTH
4	ESE	IRS

V. EXPERIMENTAL SETUP

The experiments performed in this work are meant to answer the following research questions:

- Can we use co-training to predict alternative splicing?
- How does the performance vary with the amounts of labeled and unlabeled data?
- What views of features result in better performance?
- What base classifiers give the best results for the alternative splicing problem?

We performed several sets of experiments, as shown in Table II. The classifiers used are: Naive Bayes (NB), Naive Bayes Multinomial (NBM) and Support Vector Machines (SVM), depending on the type of data in the corresponding views. For SVM, we used a Gaussian kernel with default parameters and a C value of 0.5 (C is an SVM parameter that can be seen as a penalty for errors). In our experiments, we also varied the number of iterations performed for co-training from 25 to 150, and the number of examples that are predicted at each iteration (called sample size) from 25 to 250. We found that a value of 150 worked the best overall for the number of iterations, while a value of 225 worked the best for the sample size.

Table II
EXPERIMENTS PERFORMED USING CO-TRAINING ALGORITHM.

Exp#	Combinatio# (Table I)	Alg. for View 1	Alg. for View 2
Exp 1	1, 2, 3	NBM	NB
Exp 2	1, 2, 3	NBM	SVM
Exp 3	1, 2, 3	SVM	SVM
Exp 4	4	NBM	NBM
Exp 5	4	SVM	SVM

We evaluated our co-training implementation using 5-fold cross validation. The original dataset is split into training and testing subsets, by using 4 out of 5 folds for training and the 5-th fold for testing. The training data is further divided into "unlabeled" and "labeled" data according to various percentage splits (from the total training data). Specifically,

- The amount of labeled data varies from 0.05% to 0.3%;
- The amount of unlabeled data varies from 0.15% to 0.95%.

Thus, in our experiments, we simulate unlabeled data by "ignoring" the labels of the data in the "unlabeled" data split. This allows us to compute an *upper bound* for the co-training algorithm, by training a supervised classifier on all the available training data (i.e., using the labels of the examples in the "unlabeled" data split) and testing it on the test data. To evaluate the usefulness of co-training, we also compute a *lower bound* by training a supervised classifier on the "labeled" data only (i.e., excluding the data in the "unlabeled" data split) and testing it on the test data.

VI. RESULTS

The results of the experiments designed to answer the questions in Section V are presented in what follows.

A. Study on Views and Base Classifiers

The goal of the first set of experiments is to identify the best views to be used with co-training and the best base classifiers corresponding to those views. Table III shows the results for all the experiments listed in Table II, when 0.05% of the data is used as labeled data and 0.95% is used as unlabeled data. The results reported for these experiments are the best AUC values obtained for a particular combination of views and base classifiers.

By analyzing the results in Table III, we can make the following observations:

- The best co-training performance is obtained when using the views IRS vs LENGTH, and SVM as a base classifier for both views. Specifically, the AUC value is 0.916 in this case. The next best value is 0.907 and is obtained for the views IRS+ESE vs LENGTH, also with SVM as a base classifier for both views. This shows that on our dataset, co-training gives the best results when the views used correspond to motifs and length features, respectively, and SVM is used as a base classifier. Intuitively, these views should be independent given the class variable. Also, the upper bound results show that each view in itself is predictive of the class, given enough data. Thus, co-training assumptions seem to be satisfied for these combinations of views and base classifiers, and the results support this claim.
- When the amount of labeled data is small, it is interesting to note that the combination of views IRS vs LENGTH gives better results than the combination of views IRS+ESE vs LENGTH. Generally, we would expect that adding more information (ESE) would increase the classifier performance. Instead, a decrease in the AUC value is observed. This result leads us to believe that co-training with SVM as a base classifier cannot handle well ESE features, when small amounts of labeled data are available. However, supervised SVM classifiers can make use of the ESE features, as the upper bound obtained using all the data as labeled data is better for IRS+ESE+LENGTH than it is for IRS+LENGTH.
- The following observations suggest that ESE motifs alone are not predictive when used with SVM. First, we can see that the combination of views ESE vs LENGTH

Table III
AUC VALUES FOR CO-TRAINING APPLIED ON VARIOUS COMBINATIONS OF VIEWS (FIRST COLUMN) AND BASE CLASSIFIERS (SECOND COLUMN). COLUMN 3 SHOWS THE LOWER BOUND FOR VIEW 1 (LBV1), WHILE COLUMN 4 SHOWS THE UPPER BOUND FOR VIEW 1 (UBV1). SIMILARLY, THE LOWER BOUND FOR VIEW 2 (LBV2) AND UPPER BOUND FOR VIEW 2 (UBV2) ARE SHOWN IN COLUMNS 5 AND 6, RESPECTIVELY. COLUMNS 7 AND 8 SHOW THE LOWER BOUND FOR COMBINED VIEW (LB) AND UPPER BOUND FOR COMBINED VIEW (UB). THE CO-TRAINING RESULTS (CT) ARE SHOWN IN THE LAST COLUMN. THE BEST CO-TRAINING RESULTS FOR EACH COMBINATION OF VIEWS AND BASE CLASSIFIERS ARE HIGHLIGHTED.

Features#	Classifers	LBV1	UBV1	LBV2	UBV2	LB	UB	CT
ESE vs LENGTH	NBM + NB	0.709	0.771	0.803	0.826	-	-	**0.826**
ESE vs LENGTH	NBM + SVM	0.709	0.771	0.795	0.835	-	-	0.747
ESE vs LENGTH	SVM + SVM	0.681	0.772	0.795	0.835	0.821	0.862	0.732
IRS vs LENGTH	NBM + NB	0.806	0.907	0.803	0.826	-	-	0.848
IRS vs LENGTH	NBM + SVM	0.806	0.907	0.795	0.835	-	-	0.901
IRS vs LENGTH	SVM + SVM	0.795	0.895	0.795	0.835	0.858	0.907	**0.916**
ESE+IRS vs LENGTH	NBM + NB	0.848	0.93	0.803	0.826	-	-	0.874
ESE+IRS vs LENGTH	NBM + SVM	0.848	0.93	0.795	0.835	-	-	0.892
ESE+IRS vs LENGTH	SVM + SVM	0.794	0.921	0.795	0.835	0.825	0.916	**0.907**
ESE vs IRS	NBM + NBM	0.709	0.771	0.806	0.907	0.848	0.93	**0.889**
ESE vs IRS	SVM + SVM	0.681	0.772	0.795	0.895	0.794	0.921	0.862

gives the best result when naive Bayes classifiers (NBM and NB, respectively) are used as base classifiers. In this case, the co-training result is similar or better than the upper bounds on the corresponding views. In the other cases, when SVM is used on one or both views, the co-training result is worst than the lower bounds. Furthermore, the combination of views ESE vs IRS, with NBM as base classifier for both views, gives better results than the experiment where SVM is used as base classifier. Therefore, we can conclude that naive Bayes can better capture the information in the ESE motifs than the SVM classifier.

- Another interesting fact to notice is that the combinations of views IRS vs LENGTH and IRS+ESE vs LENGTH, with SVM as base classifier on both views, result in AUC values which are greater than all upper bounds (i.e., UBV1, UBV2 and UB). One possible explanation for this behavior can be that co-training might be able to avoid some noise in the data (mislabeled instances). Co-training learns two classifiers from two different views. If the same instance is classified with high confidence by both classifiers, but the classification labels are different (in other words, the two predictions are conflicting), then that instance is skipped as opposed to being added to the training set of both classifiers. It can be that the instance with conflicting labels is mislabeled in the original training set. Thus, co-training can ignore possible mislabeling of the data, while supervised learning will use the mislabeled data and can result in a biased classifier. Another possible explanation can be that, the ensemble learning is more beneficial than learning directly from the combined views. In other words, learning separately from two different views and interchanging the best knowledge can be better than learning directly from the combined features.
- We also observe that the best result for the IRS vs ESE combination of views is obtained when NBM is applied on both views. SVM gives worst results in this case. As discussed above, one possible explanation for this is that SVM does not capture well the information in the ESE features. Furthermore, both ESE and IRS motifs are represented using counts and the NBM is known to work very well for count representations (e.g., the count representation can give better results when used with NBM than when used with SVM for text classification problems). As opposed to this, when LENGTH features are used, the results are generally better for SVM.
- At last, the results show that using the same base classifier for both views gives better results than when different classifiers are used for the two views (e.g. NBM and SVM). In other words, classifiers are able to help each other better if they are similar in nature and capture the same type of information.

As a general conclusion of the results, we can claim that IRS vs LENGTH and IRS+ESE vs LENGTH combinations of views, used with SVM, give the best results for our dataset. When NBM is used as a base classifier, the combination IRS vs ESE gives the best results. Therefore, in the next subsections, we will focus on these combinations.

B. Varying the Amount of Labeled Data

Focusing on the three combinations of views and classifiers mentioned above, in this section we evaluate the performance of co-training when varying the amount of labeled data, while the amount of unlabeled data is fixed. We first present the results of the experiments performed with NBM as base classifier for both views (IRS vs ESE), followed by results performed with SVM as base classifier for both views IRS vs LENGTH and IRS+ESE vs LENGTH.

1) NBM results: In this set of experiments, we vary the amount of labeled data from 5% to 30%, while keeping

the amount of unlabeled data fixed to 70%. The results are shown in Figure 2 (a). As can be seen, increasing the amount of labeled data results in better performance for both co-training and supervised learning from labeled data only (lower bound). However, the difference between the co-training and the lower bound decreases as the amount of labeled data increases, and the lower bound becomes greater than the co-training as the amount of labeled data reaches 30% (although neither of them gets very close to the upper bound). This result shows that co-training works well when the amount of labeled data is small.

2) SVM results: Similar to the NBM case, in this set of experiments, we vary the amount of labeled data from 5% to 30%, while keeping the amount of unlabeled data fixed to 70%. However, in this case, SVM is used as base classifier for both views, and we experiment with the IRS vs LENGTH and IRS+ESE vs LENGTH combinations of views. The results for the combination IRS vs LENGTH are shown in Figure 2 (b). Similar to the NBM results, we notice that the performance of co-training increases with the amount of labeled data. However, the results are much better as compared to the lower bound, and, in fact, they are even better than the supervised upper bound. The difference between co-training and the upper bound increases with the amount of labeled data, reinforcing our observation that co-training is able to ignore examples that might be mislabeled (or inconsistent) in the training data. The results for the combination IRS+ESE vs LENGTH are shown in Figure 2 (c) and suggest a similar trend as observed for the IRS vs LENGTH combination. However, the rate at which co-training increases is greater for smaller amounts of labeled data (10% to 15%) as compared to larger amounts of labeled data (25% to 30%).

Based on these observations, we can conclude that co-training is effectively using the unlabeled data and gives better results than the supervised learning that uses only the labeled data. While the performance increases with more labeled data, the benefit over the lower bound gets smaller. Thus, co-training should ideally be used when the amount of labeled data is very small.

C. Varying the Amount of Unlabeled Data

For the combination of features and classifiers described in Section VI-B, we are also studying the variation of the co-training performance with the amount of unlabeled data, for a fixed amount of labeled data. As before, we first present the results of the experiments performed with NBM as base classifier for views IRS vs ESE, followed by results performed with SVM as base classifier for views IRS vs LENGTH and IRS+ESE vs LENGTH, respectively.

1) NBM results: For this set of experiments, we fixed the amount of labeled data to 5% and studied the performance of co-training when we vary the amount of unlabeled data from 15% to 95%. The results for the IRS vs ESE combination of views and NBM as a base classifier are shown in Figure 3 (a). While co-training gives better results than the lower bound (showing that unlabeled data helps), the performance increases with the amount of unlabeled data up to a point; adding unlabeled data beyond that point does not help much.

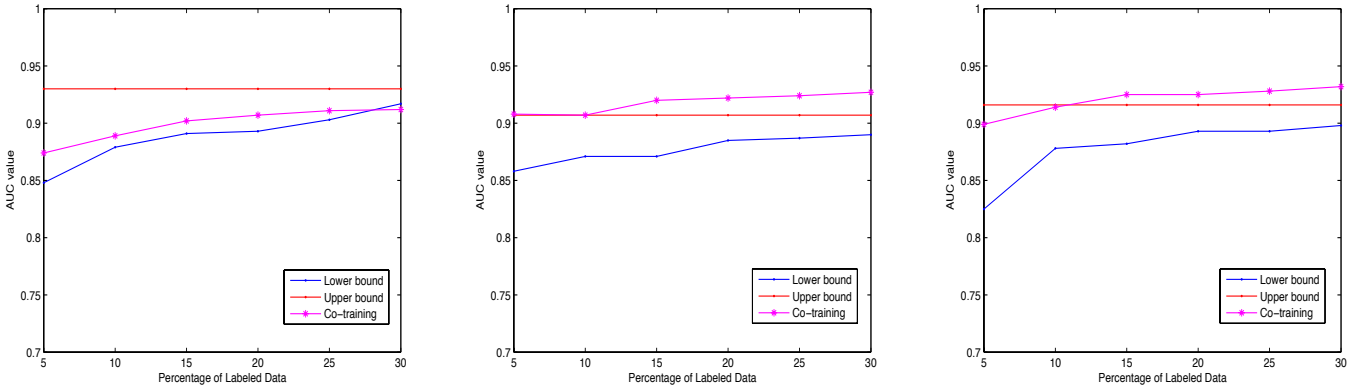
2) SVM results: Similar to the NBM experiments, for this set of experiments, we fixed the amount of labeled data to 5% and studied the performance of co-training when we vary the amount of unlabeled data from 15% to 95%. Here, SVM is used as a base classifier. The results for the IRS vs LENGTH combination of views are presented in Figure 3 (b) and the results for IRS+ESE vs LENGTH are presented in 3 (c). As in the case of NBM, the performance of co-training is better than the lower bound, which shows that the unlabeled data helps. When we vary the amount of unlabeled data from 15%, initially the performance of co-training increases up to a point. However, adding more unlabeled data beyond that point does not result in a consistent increase in the performance of co-training, when the IRS vs LENGTH combination is used. However, for the IRS+ESE vs LENGTH combination of views, no consistent performance increase can be observed across the sequence of percentages from 15% to 95% for unlabeled data. One observation that can be made is that the results of the experiments when SVM is used as a base classifier are better than those obtained with NBM, in the sense that the co-training results are closer to the upper bound for SVM.

Based on the above observations, we can conclude that the unlabeled data results in better classifiers as compared to those learned from labeled data only (results better than the lower bound and close or even better than the upper bound). However, adding unlabeled data beyond a certain percentage does not result in significant improvements, as can be seen in Figure 3. To find good combinations of features, in Figure 4 we plot the co-training results for all combinations described in Table I, when SVM is used as a base classifier for both views, and the amount of unlabeled data is varied from 50% to 95%. As can be seen from the figure, the results corresponding to IRS vs LENGTH and IRS+ESE vs LENGTH are comparable, while the results for IRS vs ESE and ESE vs LENGTH are much worst. However, the results of ESE vs LENGTH are the worst, as has also been seen from Table III

VII. Summary and Conclusions

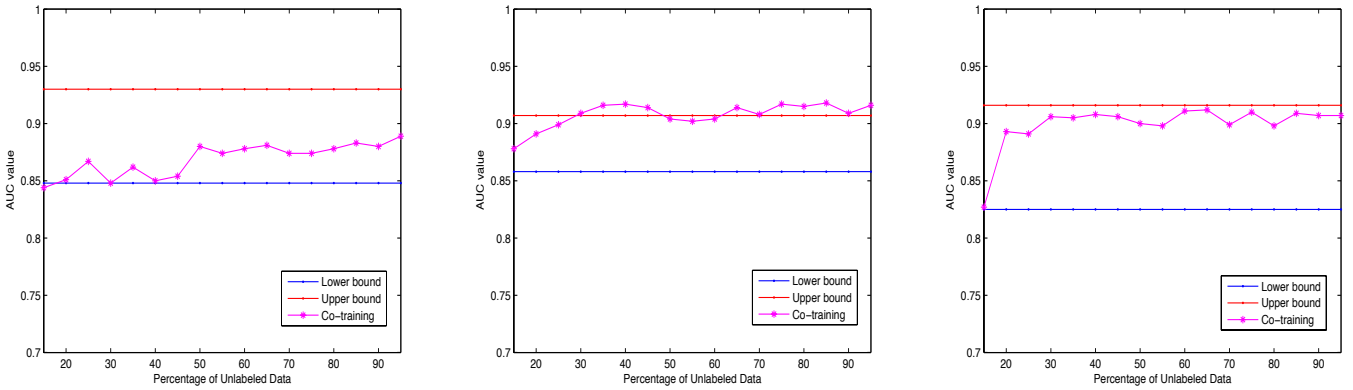
In this paper, we have studied the applicability of co-training to a genomics bioinformatics problem, specifically the problem of predicting alternatively spliced exons, under the assumption that only a small amount of labeled data is available. The following main observations can be drawn:

- Experimental results have shown that co-training performs well on biological data when the sets of features used as views are sufficient and independent (e.g., IRS vs LENGTH and IRS+ESE vs LENGTH).



(a) ESE vs IRS, vary labeled (b) IRS vs LENGTH, vary labeled (c) IRS + ESE vs LENGTH, vary labeled

Figure 2. Co-training results (AUC values) when varying the amount of labeled data from 5% to 30%, while the amount of unlabeled data is fixed to 70%. (a) IRS vs ESE are used as views, and NBM is used as base classifier for both views; (b) IRS vs LENGTH are used as views, and SVM is used as base classifier; (c) IRS + ESE vs LENGTH are used as views and SVM is used as base classifier.



(a) ESE vs IRS, vary unlabeled (b) IRS vs LENGTH, vary unlabeled (c) IRS + ESE vs LENGTH, vary unlabeled

Figure 3. Co-training results (AUC values) when varying the amount of unlabeled data from 15% to 95%, while the amount of labeled data is fixed to 0.05%. (a) IRS vs ESE are used as views, and NBM is used as base classifier for both views; (b) IRS vs LENGTH are used as views, and SVM is used as base classifier; (c) IRS + ESE vs LENGTH are used as views and SVM is used as base classifier.

- As expected, the performance of co-training increases with the amount of labeled data. When we varied the amount of unlabeled data, the performance of co-training increases initially but does not show a consistent pattern for larger percentages of unlabeled data. This suggests that, while the unlabeled data can help, it does not help much beyond a certain point.
- We can also conclude that SVM works best as a base classifier when motifs versus length features are used as views. However, NBM gives the best results for the combination of IRS vs ESE (although the corresponding AUC value is smaller than the best value obtained in SVM).Using different classifiers for the two views was found to be ineffective.

Based on the above observations, co-training is found to be effective in predicting alternative splicing events in genes. Unlabeled data plays a crucial role in improving the performance. This suggests that co-training can also be effective for various other biological problems where we have large amounts of unlabeled data.

VIII. Future Work

As part of the future work, we would like to test our approach on different datasets from different organisms. We would also like to collect more unlabeled data and run experiments where all the available labeled data is used as training, to see if the results can be improved further. Another idea for future work is to construct a validation scheme which can help identify the best set of features. We would also like to investigate different kernels for SVM

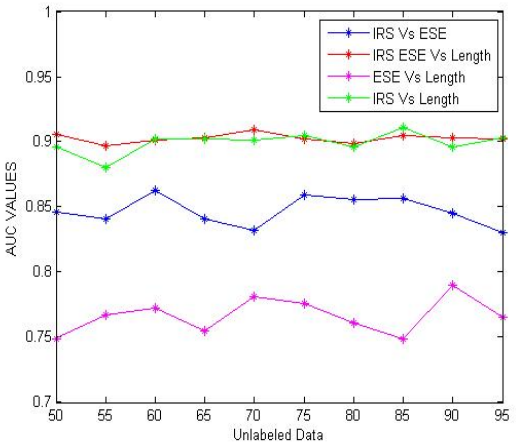


Figure 4. Co-training results for various combination of features, when SVM is used as base classifier and the amount of unlabeled data is varied from 50% to 95% (while the amount of labeled data is fixed to 5%).

(when used in co-training). A Gaussian kernel was used in the current work, but we would like to explore biological kernels such as the weighted degree kernel [17] in the future. At last, it would be interesting to explore an ensemble co-training with multiple views of features.

REFERENCES

[1] M. T. Mitchell, *Machine learning*. McGraw-Hill Companies Inc., international edition, 1997.

[2] X. Zhu, "Semi-supervised learning literature survey," Department of Computer Sciences, University of Wisconsin, Madison, Tech. Rep. 1530, 2005.

[3] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998.

[4] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, pp. 103–134, May 2000.

[5] T. Joachims, "Transductive inference for text classification using support vector machines," in *ICML '99 Proceedings of the Sixteenth International Conference on Machine Learning*. Morgan Kaufmann, 1999, pp. 200–209.

[6] M. Collins and Y. Singer, "Unsupervised models for named entity classification," in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999, pp. 100–110.

[7] A. Goldberg and X. Zhu, "Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization," in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, ser. TextGraphs-1, 2006, pp. 45–52.

[8] L. Kall, J. Canterbury, J. Weston, W. Noble, and M. MacCoss, "Semi-supervised learning for peptide identification from shotgun proteomics datasets," *Nat Meth*, vol. 4, pp. 923–925, 2007.

[9] J. Weston, R. Kuang, C. Leslie, and W. Noble, "Protein ranking by semi-supervised network propagation," *BMC Bioinformatics*, vol. 7, no. Suppl 1, p. S10, 2006.

[10] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. Noble, "Semi-supervised protein classification using cluster kernels," *Bioinformatics*, vol. 21, pp. 3241–3247, 2005.

[11] D. L. Black, "Mechanisms of alternative pre-messenger RNA splicing," *Annual Review of Biochemistry*, vol. 72, pp. 291–336, 2003.

[12] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proceedings of the ninth international conference on Information and knowledge management*, 2000, pp. 86–93.

[13] L. X. Charles, D. Jun, and Z. Zhi-Hua, "When does co-training work in real data?" in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, ser. PAKDD '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 596–603.

[14] S. Kiritchenko and S. Matwin, "Email classification with co-training," in *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*, ser. CASCON '01. IBM Press, 2001.

[15] Q. Xu, D. Hu, H. Xue, W. Yu, and Q. Yang, "Semi-supervised protein subcellular localization," *BMC Bioinformatics*, vol. 10, no. Suppl 1, p. S47, 2009.

[16] M. Li and Z.-H. Zhou, "Improve Computer-Aided Diagnosis With Machine Learning Techniques Using Undiagnosed Samples," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 37, no. 6, pp. 1088–1098, 2007.

[17] G. Ratsch, S. Sonnenburg, and B. Scholkopf, "RASE: recognition of alternatively spliced exons in *c. elegans*," in *Proceedings of Thirteenth Int. Conference on Intelligent Systems for Molecular Biology (ISMB)*, vol. 21, 2005, pp. 369–377.

[18] G. Dror, R. Sorek, and R. Shamir, "Accurate identification of alternatively spliced exons using support vector machine," *Bioinformatics*, vol. 21, pp. 897–901, 2005.

[19] J. Xia, D. Caragea, and S. J. Brown, "Prediction of alternatively spliced exons using support vector machines," *International Journal on Data Mining and Bioinformatics (IJDMB)*, vol. 4, pp. 411–430, 2010.