SEMI-SUPERVISED LEARNING FOR BIOLOGICAL SEQUENCE CLASSIFICATION

by

ANA STANESCU

B.A., Romanian-American University, Romania, 2006

M.S., James Madison University, U.S.A, 2008

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2015

Abstract

Successful advances in biochemical technologies have led to inexpensive, time-efficient production of massive volumes of data, DNA and protein sequences. As a result, numerous computational methods for genome annotation have emerged, including machine learning and statistical analysis approaches that practically and efficiently analyze and interpret data. Traditional machine learning approaches to genome annotation typically rely on large amounts of labeled data in order to build quality classifiers. The process of labeling data can be expensive and time consuming, as it requires domain knowledge and expert involvement. Semi-supervised learning approaches that can make use of unlabeled data, in addition to small amounts of labeled data, can help reduce the costs associated with labeling. In this context, we focus on semi-supervised learning approaches for biological sequence classification.

Although an attractive concept, semi-supervised learning does not invariably work as intended. Since the assumptions made by learning algorithms cannot be easily verified without considerable domain knowledge or data exploration, semi-supervised learning is not always "safe" to use. Advantageous utilization of the unlabeled data is problem dependent, and more research is needed to identify algorithms that can be used to increase the effectiveness of semi-supervised learning, in general, and for bioinformatics problems, in particular. At a high level, we aim to identify semi-supervised algorithms and data representations that can be used to learn effective classifiers for genome annotation tasks such as cassette exon identification, splice site identification, and protein localization.

In addition, one specific challenge that we address is the "data imbalance" problem, which is prevalent in many domains, including bioinformatics. The data imbalance phenomenon arises when one of the classes to be predicted is underrepresented in the data because instances belonging to that class are rare (noteworthy cases) or difficult to obtain. Ironically, minority classes are typically the most important to learn, because they may be associated with special cases, as in the case of splice site prediction. We propose two main techniques to deal with the data imbalance problem, namely a technique based on “dynamic balancing” (augmenting the originally labeled data only with positive instances during the semi-supervised iterations of the algorithms) and another technique based on ensemble approaches. The results show that with limited amounts of labeled data, semi-supervised approaches can successfully leverage the unlabeled data, thereby surpassing their completely supervised counterparts.

A type of semi-supervised learning, known as “transductive” learning aims to classify the unlabeled data without generalizing to new, previously not encountered instances. Theoretically, this aspect makes transductive learning particularly suitable for the task of genome annotation, in which an entirely sequenced genome is typically available, sometimes accompanied by limited annotation. We study and evaluate various transductive approaches (such as transductive support vector machines and graph based approaches) and sequence representations for the problems of cassette exon identification. The results obtained demonstrate the effectiveness of transductive algorithms in sequence annotation tasks.

SEMI-SUPERVISED LEARNING FOR BIOLOGICAL SEQUENCE CLASSIFICATION

by

ANA STANESCU

B.A., Romanian-American University, Romania, 2006

M.S., James Madison University, U.S.A, 2008

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2015

Approved by:

Major Professor
Doina Caragea

Copyright

Abstract

Successful advances in biochemical technologies have led to inexpensive, time-efficient production of massive volumes of data, DNA and protein sequences. As a result, numerous computational methods for genome annotation have emerged, including machine learning and statistical analysis approaches that practically and efficiently analyze and interpret data. Traditional machine learning approaches to genome annotation typically rely on large amounts of labeled data in order to build quality classifiers. The process of labeling data can be expensive and time consuming, as it requires domain knowledge and expert involvement. Semi-supervised learning approaches that can make use of unlabeled data, in addition to small amounts of labeled data, can help reduce the costs associated with labeling. In this context, we focus on semi-supervised learning approaches for biological sequence classification.

Although an attractive concept, semi-supervised learning does not invariably work as intended. Since the assumptions made by learning algorithms cannot be easily verified without considerable domain knowledge or data exploration, semi-supervised learning is not always "safe" to use. Advantageous utilization of the unlabeled data is problem dependent, and more research is needed to identify algorithms that can be used to increase the effectiveness of semi-supervised learning, in general, and for bioinformatics problems, in particular. At a high level, we aim to identify semi-supervised algorithms and data representations that can be used to learn effective classifiers for genome annotation tasks such as cassette exon identification, splice site identification, and protein localization.

In addition, one specific challenge that we address is the "data imbalance" problem, which is prevalent in many domains, including bioinformatics. The data imbalance phenomenon arises when one of the classes to be predicted is underrepresented in the data because instances belonging to that class are rare (noteworthy cases) or difficult to obtain. Ironically, minority classes are typically the most important to learn, because they may be associated with special cases, as in the case of splice site prediction. We propose two main techniques to deal with the data imbalance problem, namely a technique based on "dynamic balancing" (augmenting the originally labeled data only with positive instances during the semi-supervised iterations of the algorithms) and another technique based on ensemble approaches. The results show that with limited amounts of labeled data, semi-supervised approaches can successfully leverage the unlabeled data, thereby surpassing their completely supervised counterparts.

A type of semi-supervised learning, known as "transductive" learning aims to classify the unlabeled data without generalizing to new, previously not encountered instances. Theoretically, this aspect makes transductive learning particularly suitable for the task of genome annotation, in which an entirely sequenced genome is typically available, sometimes accompanied by limited annotation. We study and evaluate various transductive approaches (such as transductive support vector machines and graph based approaches) and sequence representations for the problems of cassette exon identification. The results obtained demonstrate the effectiveness of transductive algorithms in sequence annotation tasks.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my deepest gratitude and most sincere appreciation to my advisor, Dr. Doina Caragea, for introducing me to the wonderful world of research and for being the best source of knowledge and inspiration during my graduate experience. This dissertation would not have been possible without her wise guidance and constant support. Dr. Caragea has been a great mentor and hopefully our collaboration will continue beyond my time in her lab.

Special thanks go to Dr. Susan J. Brown, for playing a very important role in my career at Kansas State University. I feel privileged for the experience of working closely with Dr. Brown, on many occasions.

I am very thankful to my committee members, Dr. Torben Amtoft and Dr. Dan Andresen, for their valuable participation in this whole process and for their guidance throughout my entire graduate studies at Kansas State University.

Warm thanks go to the former and current students of the MLB club, especially to my collaborators Karthik Tangirala, Swapnil Nagar, and Srilaxmi Cheeti.

It has been an honor and a great experience to be part of the Computing and Information Sciences Department at Kansas State University. Special thanks for their contribution to my education go to Dr. Masaaki Mizuno, Dr. Robby, and Dr. Dave Schmidt.

Dedication

To my wonderful parents.

Chapter 1

Introduction

1.1 Motivation

The developments of biotechnology during the last decades have resulted in powerful high throughput sequencing instruments that can produce biological data (both DNA and derived protein sequences) rapidly and inexpensively. Classical wet-lab annotation methods can no longer handle the sheer volume and complexity of this exponentially accumulating data. The "post-genomic" era, in which efforts are shifted towards data analysis, requires computational methods to assist with the annotation tasks.

Traditionally, supervised machine learning has been successfully used for classification or prediction problems in the field of bioinformatics. Supervised methods, however, require large amounts of labeled data for training in order to produce valuable classifiers, but in many cases obtaining a sufficiently large number of labels is infeasible. Unlabeled instances are more accessible and usually they are available in much larger quantities than the labeled instances. Therefore, semi-supervised learning, in which the classifiers trained on limited amounts of labeled data can be improved by exploiting the large amounts of unlabeled data, can provide cost-effective alternatives and is preferable to supervised learning.

1.2 Overview of Machine Learning Approaches Used

Traditional machine learning algorithms rely on the availability of large amounts of accurate labeled data [Mitchell, 1997] to find a function that fits the training (labeled) data and also generalizes to new data. In supervised learning (SL), as can be seen in Figure 1.1, the training phase produces a model that can be used to classify test (unlabeled) instances. Supervised machine learning produces dependable models when there are large amounts of labeled data available for training, but in reality, labeled data is usually scarce while large volumes of unlabeled data are readily available. For genetics, obtaining labeled data is an expensive process that requires expert involvement and time. From here comes the necessity to use semi-supervised learning, which is a learning paradigm at the intersection between supervised and unsupervised learning.

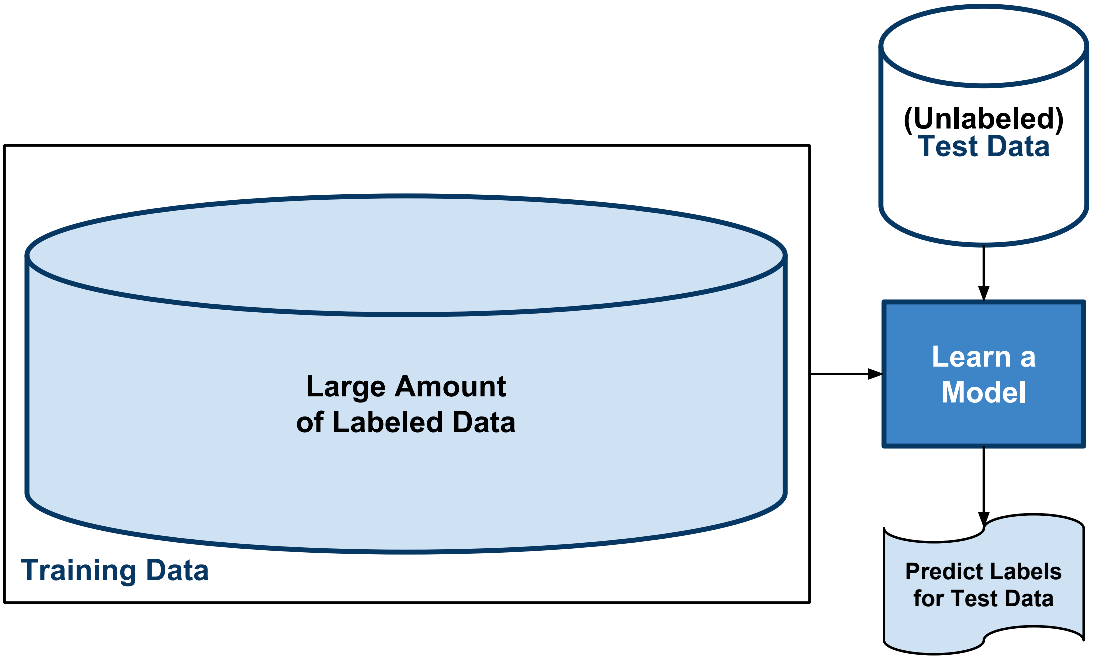


Figure 1.1: **Supervised Learning**: In supervised learning, a model is learned from a large amount of labeled data. The model can the be used to classify test instances not encountered before.

In this research, the focus is on semi-supervised learning, more precisely, semi-supervised classification, and a special case of semi-supervised learning, known as transductive learning.

Semi-supervised learning (SSL) [Chapelle et al., 2006] is a framework in machine learning that provides a comparatively cheap alternative to having experts manually label more data. The aim of semi-supervised learning is to utilize both the small amount of available labeled data and the abundant unlabeled data together, in order to give the maximum generalization ability on a given prediction task. Using unlabeled data together with labeled data sometimes gives better results than using the labeled data alone. The concept of improving supervised classifiers by leveraging unlabeled data is very appealing, yet it does not always work as intended. In practice, it is very common for a classifier to be degraded by the unlabeled data [Li and Zhou, 2011; Catal and Diri, 2009]. Deciding whether or not to use the unlabeled data is a problematic task [Singh et al., 2009], and the focus of ongoing research [Wang and Chen, 2013].

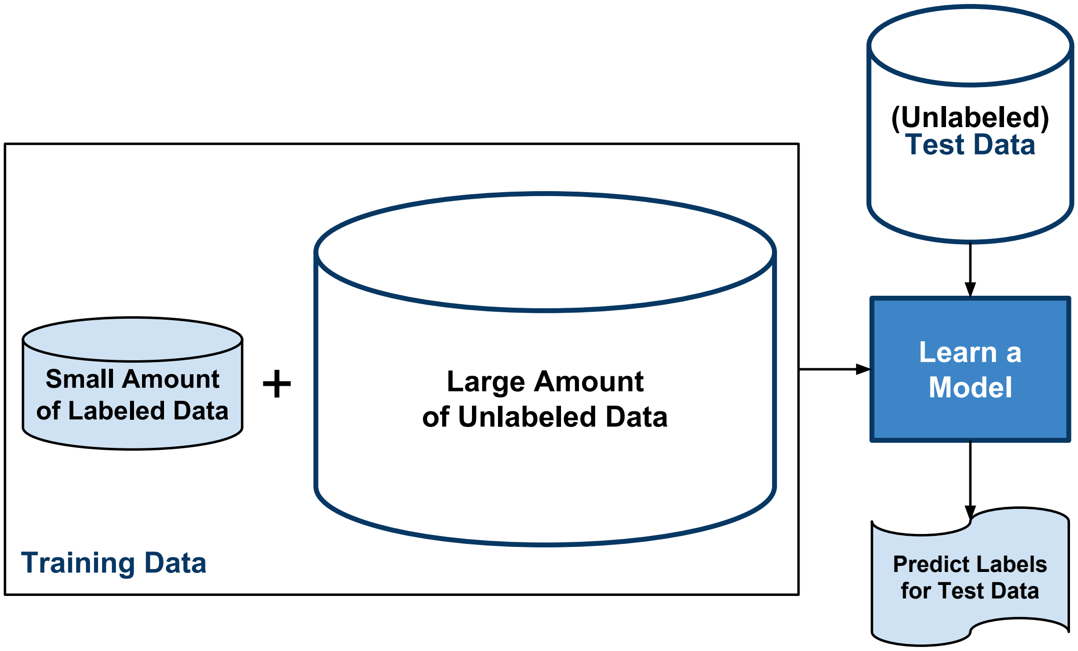


Figure 1.2: **Semi-supervised Learning**: In the semi-supervised learning paradigm, a model is learned from limited labeled data and much larger quantities of unlabeled data. The model can then be used to classify test instances not encountered before.

In a classic semi-supervised learning framework, the algorithm learns a model from the labeled data and aims to improve that model by leveraging the latent knowledge presumably available in the unlabeled data. At the end of the learning process, new unseen (test) instances are classified, as shown in Figure 1.2. This type of learning, where a model is produced and can be used to classify newly encountered instances, is also known as inductive learning, or simply induction.

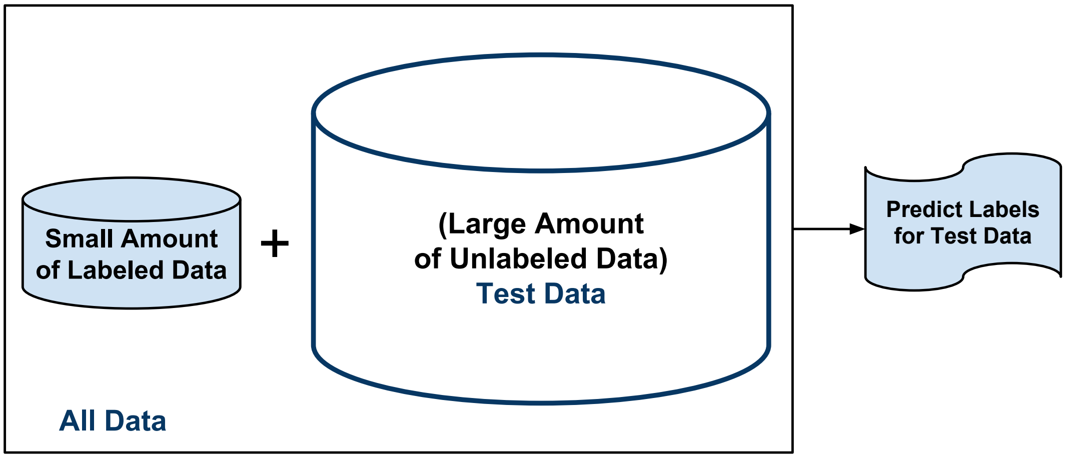

Small Amount of Labeled Data
+
(Large Amount of Unlabeled Data)
Test Data
All Data
Predict Labels for Test Data

Figure 1.3: **Transductive Learning**: In transductive learning, the aim is to find the labels of the unlabeled data, which can be seen as test data.

As opposed to inductive learning, in which a model is produced, transductive learning (TL) aims only to classify the unlabeled data that is already accessible. The algorithm receives the instances to be classified in advance, as shown in Figure 1.3. In such a scenario, where the instances to be classified are known *a priori*, it is desirable to discriminate between the classes in the available data, rather than solving a harder problem, which would be to learn a mode that can be used to discriminate between classes in future unseen data.

1.3 Biological Problems Addressed

The "central dogma of molecular biology", as elaborated by Francis Crick [Crick et al., 1970], explains the flow of genetic information from DNA to protein. It states that DNA duplicates (replication), makes RNA (transcription), which is next transported outside the nucleus into the cytoplasm, where the cellular machinery translates it into proteins (synthesis). Proteins are made of amino-acids. In genetics, gene expression is the most fundamental level at which the genotype gives rise to the phenotype. The genetic code stored in DNA is "interpreted" by gene expression, and the properties of the expression give rise to the organism's phenotype. Two main biological problems are used as applications of the work in this dissertation:

- **Prediction of alternative splicing events**

  Throughout the DNA strands, the are expressed regions (also known as exons) and intervening regions (also known as introns). Alternative splicing, schematically shown in Figure 1.4, is a phenomenon that occurs naturally, during gene regulation, in multicellular organisms and it is a main contributor to isoform diversity.

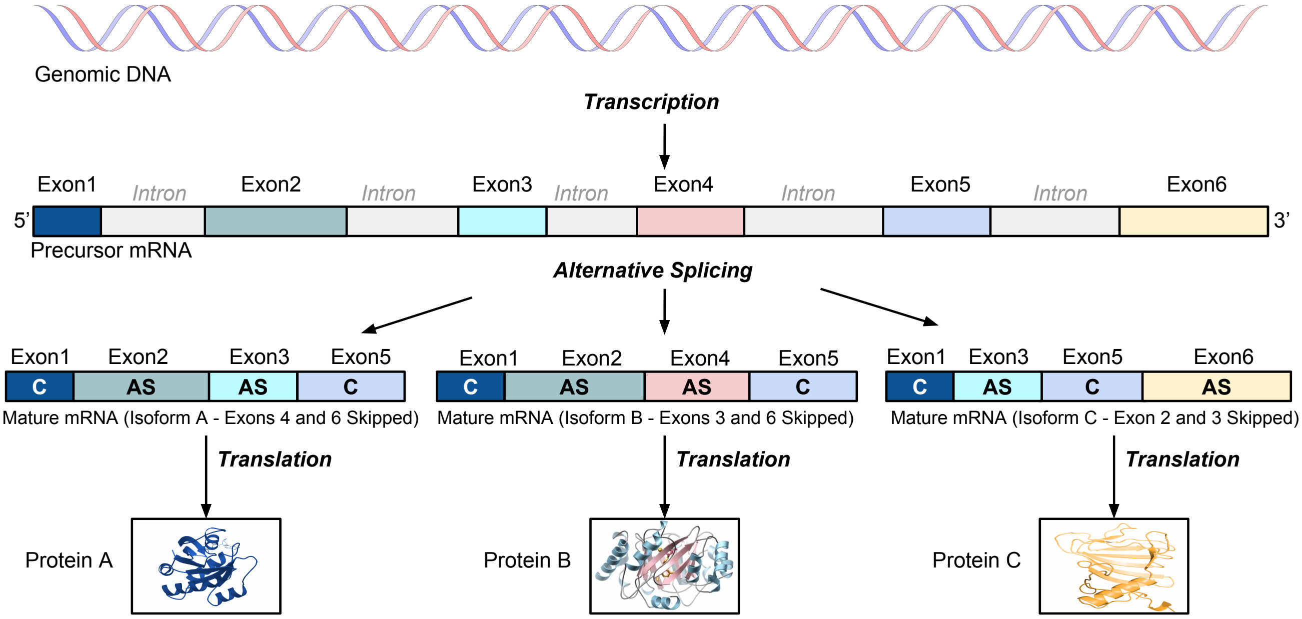


Figure 1.4: **Alternative Splicing** is a regulated process responsible for the formation of multiple proteins from a single gene. Exons 2, 3, 4, and 6 are alternatively spliced, whereas Exons 1 and 5 are constitutive, as they appear in all isoforms.

There are five major types of alternative splicing events: exon skipping, intron retention, alternative donor site, alternative acceptor site, and mutually exclusive exons. In our work, we focus on the first type, exon skipping, and formulate the prediction task as a binary classification problem with the objective of discriminating between skipped exons, also referred to as *alternatively spliced* or *cassette* exons, and *constitutive* exons. We conduct experiments on a dataset from the organism *C. elegans*.

- **Prediction of splice sites**

  Splice sites are conserved nucleotide dimers that indicate the boundaries between exons and introns. As shown in Figure 1.5, they can be donor splice sites, signaled by "GT" and situated at the 5' end of the intron, or acceptor splice sites, indicated by "AG" and situated at the 3' end of the intron.

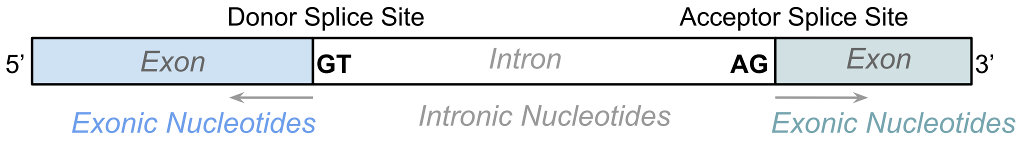


Figure 1.5: **Splice Site Prediction** refers to the identification of "AG" and "GT" dimers as true acceptor splice site and true donor splice site, respectively.

  Accurate gene identification relies heavily on correct determination of splice sites. The major difficulty comes from the fact that such dimers occur very frequently throughout the entire genome and their simple presence is not enough to declare a splice site. Splice site prediction is a problem for which the natural positive (true splice site) to negative (decoy site) ratio is very high. This challenge is known in the machine learning field as the "data imbalance" problem, and requires specially designed approaches, because the existence of a major unevenness between the prior class probabilities leads to impartial learning thereby severely altering the performance of classifiers which otherwise produce acceptable results. Our experiments are conducted on five relatively large and highly imbalanced splice site datasets, from five organisms: *C. elegans, C. remanei, P. pacificus, D. melanogaster*, and *A. thaliana*.

To summarize, the two main obstacles in the way of successfully analyzing and interpreting biological data using computational methods are (1) labeled data insufficiency and (2) the data imbalance problem. While there are some studies on protein classification, DNA-level classification using semi-supervised and transductive approaches has not been studied much.

1.4 Outline and Contributions

The major published contributions of this dissertation include:

- Chapter 2: *Predicting Alternatively Spliced Exons Using Semi-supervised Learning* [Stanescu et al., 2015] (In press.) is an extension of an earlier work, namely *Semi-supervised Learning of Alternatively Spliced Exons using Expectation Maximization Type Approaches* [Stanescu and Caragea, 2012].

  In this chapter, we conducted an ample comparison of iterative semi-supervised learning algorithms applied to the DNA prediction problem of distinguishing between alternatively spliced and constitutive exons. The algorithms are Expectation Maximization, Self-training and Co-training using biologically relevant motifs and length features derived from the sequences.

- Chapter 3: *An Empirical Study of Self-training and Data Balancing Techniques for Splice Site Prediction* [Stanescu and Caragea, 2015b] (Under review) is an extension of *Semi-supervised Self-training Approaches for Imbalanced Splice Site Datasets* [Stanescu and Caragea, 2014a].

  In this chapter, we performed an analysis of self-training classifiers using Naïve Bayes, on five large and highly imbalanced splice site datasets, and have utilized balancing techniques to address the uneven class distributions.

- Chapter 4: *An Empirical Study of Ensemble-based Semi-supervised Learning Approaches for Imbalanced Splice Site Datasets* [Stanescu and Caragea, 2015a] (In press.) is extending the work *Ensemble-based semi-supervised learning approaches for imbalanced splice site datasets* [Stanescu and Caragea, 2014b].

In this chapter, we proposed and studied several ensemble-based variants of two popular semi-supervised learning algorithms, self-training and co-training, and tested their performance on the task of predicting splice sites. We adapted the ensembles to address the highly imbalanced datasets of our case study, and we used various approaches to augment the labeled data during the semi-supervised iterations.

- Chapter 5: *Predicting cassette exons using transductive learning approaches.* [Stanescu and Caragea, 2015c].

  In this chapter, we studied the applicability of three popular transductive techniques and their compatibility with various kernels to the binary DNA classification problem of cassette exon identification.

Chapter 2

Predicting Alternatively Spliced Exons Using Semi-supervised Learning

2.1 Introduction

Supervised machine learning (also known as inductive learning) allows complex tasks to be solved using a mathematical model inferred from prior knowledge (*i.e.*, training data). When used for classification tasks, a classification function, $y = f(x)$, is learned based on training instances, and this function is further used to predict the classes of new, unseen instances. In supervised learning SL, training instances represent object-label pairs of the form $(x, f(x))$. Supervised learning has been applied to a variety of domains, including bioinformatics, among others [Jiang et al., 2013; Rider et al., 2014; Erdoğdu et al., 2013; Yu et al., 2013; Wang and Wu, 2006; Chen, 2008]. The power of supervised machine learning techniques depends on the quality and quantity of labeled data. In general, the more labeled data is available, the better the classifiers learned. However, the process of obtaining labeled instances is slow and/or expensive. On the other hand, unlabeled instances are more accessible. Semi-supervised learning (SSL) algorithms build classifiers using both labeled and unlabeled data. SSL has received a lot of interest because it can improve upon a prediction model learned only from labeled data by incorporating the unlabeled data in the training process. Models trained in the semi-supervised framework are highly desirable when there is substantial unlabeled data available, while labeled instances are scarce and costly. SSL algorithms have shown great potential in many areas including text classification [Nigam et al., 2000; Blum and Mitchell, 1998; Niu et al., 2005; Collins and Singer, 1999; Gupta and Ratinov, 2008; Dai et al., 2007], sentiment categorization [Goldberg and Zhu, 2006], natural language processing [Collins and Singer, 1999] and image classification [Rosenberg et al., 2005].

Similar to these areas, the biological domain presents us with large amounts of unlabeled data that are produced relatively fast and inexpensively. Thanks to advances in the Next Generation Sequencing (NGS) technologies, which have led to the production of unparalleled amounts of genomics data, the interest has gradually shifted towards data interpretation [Baldi and Brunak, 2001], for which automated systems are in high demand. Numerous bioinformatics tasks can be formalized as classification problems, for example the task of recognizing splice sites or alternative splicing events; or the task of predicting protein functions.

The identification of alternative splicing events, in particular, is essential for genome annotation. Alternative splicing is a natural phenomenon, first observed towards the late 1970's [Chow et al., 1977; Berget et al., 1977], that contributes to protein diversity [Schmucker et al., 2000]. Genes contain regions that are expressed, called *exons*, which alternate with intragenic regions, called *introns*. Generally, introns are removed (or spliced out) from the gene sequence and exons are retained (or transcribed) when creating mature RNA (mRNA) from DNA. However, sometimes a gene can produce several transcripts (or splice variants), due to alternative splicing events. For example, an exon can be present in a transcript, but skipped in another transcript, in which case we say that the exon is alternatively spliced (exon skipping event). The alternative gene transcripts are called isoforms. Similarly, an intron can be spliced out in a transcript but retained in another transcript, an alternative splicing event called intron retention. Other manifestations of alternative splicing include: alternative 3' and 5' splice-site selection, mutually exclusive exons [Keren et al., 2010]. Identification of alternative splicing events can be addressed by conducting wet-lab experiments. However, as lab work can be very tedious, computational methods based on Expressed Sequence Tags (EST) and, more recently, RNA-Seq to genome alignments have emerged [Bonizzoni et al., 2005; Nagaraj et al., 2007; Lu et al., 2009]. Supervised machine learning approaches have also been used in the context of alternative splicing, including the prediction of alternatively spliced exons [Rätsch et al., 2005], a binary classification problem, where the two classes are given by alternatively spliced or constitutive exons (*i.e.,* exons that are always present in the transcript). Specialized kernels that model similarities between sequences have been used with Support Vector Machines to predict alternative splicing [Dror et al., 2005; Ben-Hur et al., 2008; Xia et al., 2010].

Supervised learning requires labeled training data, and does not benefit from the large amounts of unlabeled data available in biological sciences. When applicable, semi-supervised learning techniques provide more attractive, cost-effective solutions for bioinformatics, including genome annotation problems such as alternative splicing prediction. While semi-supervised learning has been used for protein classification [Weston et al., 2005; 2006; Kall et al., 2007; Craig and Liao, 2007], it has not been studied much for DNA sequence classification, such as alternative splicing prediction. To address this limitation, in prior work, we studied how EM-type approaches [Stanescu and Caragea, 2012] and Co-training approaches [Tangirala and Caragea, 2011] behave on the problem of predicting alternatively spliced exons. Our previous results motivated us to perform an ample comparison of SSL algorithms on this problem. Specifically, we compare three semi-supervised learning algorithms: Expectation Maximization, Self-training and Co-training. As opposed to prior work, in the current work, we do extensive parameter tuning. Furthermore, we also experiment with Random Forests as a base classifier, in addition to Naïve Bayes and Support Vector Machines algorithms, which we used in prior work. The results of our study show that semi-supervised approaches can lead to good classification performance, as compared to supervised learning from a small amount of data. These results can be used as evidence for the usefulness of semi-supervised learning approaches for genome annotation tasks and could potentially open the avenue for more extensive research in this direction.

The rest of the paper is organized as follows: we provide a review of related work in Section 2.2 and present the approaches used in Section 2.3. Then, in Section 2.4, we describe our experimental design. We discuss the results in Section 2.5, and conclude our study and propose future research directions in Section 2.6.

2.2 Related Work

In this section, we review previous work related to and alternative splicing.

Semi-supervised Learning Algorithms

Expectation Maximization: The EM approach originates from statistics and was formalized as a probabilistic algorithm for maximum likelihood estimation by Dempster et al. [1977]. EM allows the learning of a model in the presence of missing data, through iterative parameter estimation. Its applicability to learning probability distributions and capability of utilizing sufficiently large amounts of unlabeled data in order to build and improve upon a supervised model makes EM a very powerful technique, which has gained a lot of popularity. In recent years, a semi-supervised approach using EM and Naïve Bayes with Probabilistic Labels was proposed by Nigam et al. [2000], in the context of text classification. EM has been shown to be very useful in semi-supervised frameworks for other tasks, including audio categorization tasks [Moreno and Agarwal, 2003], and image retrieval tasks [Dong and Bhanu, 2003]. In biological and medical domains, EM has been used for various problems such as modeling data for creating protein profiles [Nesvizhskii et al., 2003], finding motifs within protein sequences [Lawrence and Reilly, 1990], classifying protiens based on phylogenetic profiles [Craig and Liao, 2007], image reconstruction through clustering [Dong and Bhanu, 2003], etc.

Self-training: Self-training (ST) is another technique that iteratively improves upon an initial supervised classifier. At each iteration, the current classifier is used to label the unlabeled data and the instances whose new labels are assigned with highest confidence will be added to the labeled pool for the next iteration. Since it was first introduced by Yarowsky [1995] for a text disambiguation problem, Self-training has been successfully applied to other problems involving natural language processing [Collins and Singer, 1999], object detection [Rosenberg et al., 2005], and bioinformatics [Kundu et al., 2013]. Other modifications and variations have been proposed, either at the base classifier level [Guo et al., 2012], or for the process of iteratively augmenting the labeled pool [Korecki et al., 2008].

Co-training: As originally described by Blum and Mitchell [1998], Co-training (CoT) is a two-view iterative learning technique, which uses two independent and sufficient feature representations (or views) of the same data, to learn two different classifiers. At each iteration, the training data of one classifier is augmented with the best predictions that the other classifier makes on the unlabeled data. The final classifiers are used together to predict labels for new data. Blum and Mitchell [1998] used Co-training to solve the problem of identifying course pages among other academic web pages. Nigam and Rayid [2000] compared the performance of Co-training with that of supervised learning algorithms, as well as EM and Self-training semi-supervised learning algorithms. Their results showed that Co-training and EM outperformed supervised learning algorithms, irrespective of the independence between the two views used in Co-training. Co-training outperformed EM and Self-training, even in some cases when the two views used were not independent of each other. The authors suggest that Co-training might be more robust to the assumptions made by the base classifier used, although not as robust to its own assumptions (*i.e.,* view sufficiency and independence). Kiritchenko and Matwin [2011] worked on the problem of classifying emails using Co-training and compared the performance obtained with different base classifiers. Specifically, they used Naïve Bayes and Support Vector Machines as base classifiers and found that Co-training with SVM outperformed Co-training with Naïve Bayes.

Semi-supervised Learning in Bioinformatics

The nature of biological data, large amounts of labeled data and small amounts of unlabeled data, has led many researchers to address biological prediction problems using SSL approaches. For example, Weston et al. [2005] classified protein domains into SCP super-families (SCP stands for Structural Classification of Proteins), both in an SSL and a transductive setting. The authors employed cluster kernels (bagged mismatch and neighborhood mismatch kernels) to make use of unlabeled data along with labeled data in learning an SVM classifier.

Xu et al. [2009] used the CoForest approach [Li and Zhou, 2007], an ensemble of decision tree classifiers, which makes use of knowledge from both labeled and unlabeled data, to predict protein localization. A 2-gram representation is used to encode protein sequences into feature vectors. Experimental results show that the CoForest approach outperforms several baselines including decision trees, AdaBoost and state-of-the-art SVM classifiers that make use of labeled data only. Pacharawongsakda and Theeramunkong [2013] also addressed the problem of predicting protein localization in the SSL framework. Specifically, the authors proposed iFLAST-CORE, which combines Singular Valued Decomposition (a dimensionality reduction technique) with Co-training (SSL approach). Experimental results suggest that iFLAST-CORE improves the performance as compared to supervised learning or Co-training only.

Kim and Choi [2011] proposed a hybrid generative/discriminative model that makes use of unlabeled sequences along with the available labeled sequences in the process of discovering motifs. Li et al. [2003] proposed a co-updating approach to classify gene function, and their approach makes use of heterogeneous sources of data. Specifically, two classifiers, that benefit each other in a co-training-type fashion, are learned from microarray expression data (vectors of log ratio expression of genes) and phylogenetic profile data.

Teng and Tan [2012] proposed the semi-supervised and shifted bicluster identification algorithm (SS-CoSBI) to identify multiple cell-type-specific histone modification states associated with human enhancers. SS-CoSBI incorporates co-occurrence frequencies of histone modifications (labeled source of information) as probabilistic priors to adjust the similarity measure in biclustering (unsupervised).

Semi-supervised learning has been used also for problems related to gene regulatory networks [Cerulo et al., 2010; Ernst et al., 2008; You et al., 2010], protein-protein interaction networks [Loc, 2012; Nguyen and Ho, 2012; Lei and Aidong, 2010; Jiang and McQuay, 2012; Qi et al., 2010], applications to microarray data [Yip et al., 2009], assembly problems [Xu et al., 2012], etc.

Prediction of Alternative Splicing Events

Originally, the problem of identifying alternatively spliced exons in genomic sequences has been addressed by conducting wet-lab experiments. However, approaches that align Expressed Sequence Tags (EST) and transcripts to genome became prevalent as the sequencing technologies advanced [Nagaraj et al., 2007; Wu and Watanabe, 2005; Kim et al., 2005]. Most recently, alternative splicing events are identified by aligning short RNA-seq reads to a genome. As an alternative to alignment-based methods, machine learning approaches have also been used extensively to predict alternative splicing.

RNA-seq data based approaches

Recent advancements in sequencing technologies facilitated the use of RNA-seq data in the process of identifying alternative splicing events. Sacomoto et al. [2012] and Sacomoto et al. [2013] designed methods based on *de Bruijn* graphs (DBG) obtained from RNA-seq reads to identify alternative splicing events. Zhou et al. [2012] used a combination of alignment and transcript reconstruction tools to identify novel splicing events in human genome. Kroll et al. [2012] used a set of regular expressions along with genome mapping, to identify complex alternative splicing events also in human genome. Pervouchine et al.

[2013] estimated alternative splicing events from RNA-seq data using a metric based on the number of reads that align to the exonic sequence as well as the inclusion and exclusion of exon adjacency.

Machine learning approaches

Rätsch et al. [2005] used SVM to predict alternative splicing events in *C. elegans* in a supervised learning scenario. The authors used a weighted degree kernel along with length-based features to learn a classifier. Dror et al. [2005] predicted alternative splicing events in humans. The authors used conserved information between human and mouse, upstream and downstream intronic sequence motifs, and length-based features in the learning process. Yeo et al. [2005] used a regularized least-square classifier on top of sequence-based features to identify alternative splicing events in human and mouse. Xia et al. [2010] used sequence dependent features (based on GC content, exonic splicing enhancers, intronic regulatory splicing motifs, etc.) to predict alternative splicing events. Eichner et al. [2011] proposed a two stage supervised learning approach to identify alternative splicing events (specifically, exon skipping and intron retention). In the first step, the authors use SVM classifiers to discriminate between exons and introns, while in the second step, they discriminate between alternatively spliced and constitutive events. Chen [2008] addressed the problem of predicting skipped exons using Random Forests (RF) with position-specific conservation scores as features. LeGault and Dewey [2013] proposed an EM-type algorithm for estimating the maximum a posteriori parameters of Probabilistic Splice Graphs (PSG), given RNA-seq data, an alternative transcript quantification task. Probabilistic Splice Graphs represent all isoforms of a gene along with the structural relationships among them.

To the best of our knowledge, with the exception of our own prior work [Stanescu and Caragea, 2012; Tangirala and Caragea, 2011], there is no study that uses SSL for alternative splicing prediction. Furthermore, we are not aware of any study that compares EM, Self-training and Co-training on a DNA sequence classification problem. We focus precisely on this comparison.

2.3 Semi-supervised Learning Approaches Used in Our Study

In this section, we describe the SSL algorithms studied in the context of alternative splicing prediction, and the base classifiers used with the SSL algorithms.

Expectation Maximization (EM)

One of the approaches that we study is the Expectation Maximization, an iterative technique for maximum likelihood estimation. The usage of EM in a semi-supervised framework assumes that a base classifier is first trained on the originally labeled data. Next, the classifier is used to estimate and assign probabilistic class labels for the unlabeled instances. The classifier is trained again using all the instances, originally labeled along with newly labeled. This process is repeated for a fixed number of iterations or until convergence, *i.e.*, until the labels assigned to the unlabeled instances don't change from one iteration to the next. For text classification, Naïve Bayes has been successfully used as a base classifier [Nigam et al., 2000]. Note that we use the multinomial model to capture the frequency of a feature (*e.g.*, word or motif), rather than simply its presence or absence, which would require a multi-variate Bernoulli event model [McCallum and Nigam, 1998].

EM with Weighted Instances (EMW)

One variant of the standard EM approach can be obtained by assigning different weights to the labeled and unlabeled instances during training. This variant was originally proposed by Nigam et al. [2000], and used with NBM. Specifically, this variant introduces a new weighting factor, designed to control the weight of each newly classified unlabeled instance by adjusting (generally, decreasing) the influence of the unlabeled data over the model parameters, and granting more authority to the labeled instances. Thus, for NBM, in addition to the probabilistically-weighted class labels, the unlabeled instances are also given lower weight, which means they contribute less to the final model as compared to the labeled instances. The weighting scheme handles unlabeled instances with more caution, as sometimes they can add noise to the model and ultimately decrease the performance.

Self-training (ST)

Another popular SSL algorithm is Self-training, also known as self-teaching or bootstrapping. It was introduced by Yarowsky [1995], who successfully used it for a natural language processing problem. Self-training can be built around any base-classifier, which is first trained on the labeled data. Then, the unlabeled instances are classified using the resulting model. The newly labeled instances are subsequently used to self-train in the next iteration, by integrating them in the labeled set and re-training. UnlikeEM, where all predictions are used to update the parameters of the model, Self-training only uses the best predictions, and disregards the instances that are labeled with low confidence. An important requirement is to maintain the ratio of positive to negative instances in the labeled training set when adding newly labeled instances.

Co-training (CoT)

Co-training is another iterative algorithm designed by Blum and Mitchell [1998]. It is similar to self-training. However, unlike Self-training, which is a single-view learning algorithm, Co-training uses two views of the data (two different sets of features) to train two supervised classifiers. At each iteration, the classifiers are updated using the newly classified unlabeled data, by adding the most confidently labeled instances to the other classifier's training labeled set. The intuition is to allow classifiers to complement each other's learning and benefit from each other. To get optimal results from Co-training, the views must satisfy the properties of sufficiency (each view should be sufficient to train quality classifiers on its own) and independence (the views are independent of each other given the classes). Similarly to ST, the number of instances added to the training labeled set must not skew the prior class distribution.

The SSL algorithms that we focus on in this study can be seen as wrapper methods, and they all require a base classifier. We use three base classifiers, belonging to three different types of machine learning approaches: generative (Naïve Bayes Multinomial), discriminative

(Support Vector Machines) and ensembles of decision trees (Random Forests). First, we chose Naïve Bayes Multinomial (NBM) as biological sequence classification (in particular, alternative splicing prediction) can be seen as equivalent to text classification. For text, words are generally used to represent the data; for biological sequences we use motifs to represent the data. Given that NBM has been successfully used for text data, we hypothesize it might work well also for biological data. Second, prior work by Dror et al. [2005] has shown that the Support Vector Machines (SVM) algorithm results in good performance when used to identify alternative splicing events in a supervised learning scenario. We hypothesize it might also give good performance as a base classifier in an SSL scenario. Thirdly, ensemble classifiers such as Random Forests (RF) are very popular as they are known to surpass the predictive performance of single learners. Thus, we want to explore the performance of an ensemble classifier as a base classifier in SSL algorithms.

2.4 Experimental Setup

We study the applicability of semi-supervised learning algorithms (EM-type approaches, Self-training and Co-training) to the DNA binary classification problem of distinguishing alternatively spliced exons from constitutive exons.

2.4.1 Dataset and Feature Representation

We use genomic data from the model organism *C. elegans* in our experiments. The dataset was published by Rätsch et al. [2005] and contains 3,018 nucleotide sequences of exons and flanking introns (i.e., each instance is of the form of intron-exon-intron; the dataset is available at `http://people.kyb.tuebingen.mpg.de/raetsch/RASE.old/`). Out of these 3,018 instances, 487 are labeled as alternatively spliced, meaning that the flanked exon can be skipped in some isoforms. The rest of 2,531 sequences are labeled as constitutively spliced, meaning that the exon is present in all known isoforms. The data was labeled based on alignments between ESTs and genomic DNA.

To represent instances in this dataset as vectors, we used two sets of splicing regulators as features. Splicing regulators are biological motifs known to be responsible for the occurrence of alternative splicing events. They can occur in both exons and introns. The ones that occur in exons are called Exonic Splicing Enhancers (ESE), while those occurring in introns are called Intronic Regulatory Sequences (IRS). We used 45 Exonic Splicing Enhancers hexamers (6-nucleotide long) derived by Xia et al. [2010] for the *C. elegans* dataset. For each sequence, we used a sliding window to find the 45 ESE motifs and to obtain their frequencies within each sequence, thus producing a count representation. The class label was not used in this procedure. The set of Intronic Regulatory Sequences (IRS motifs) that we used was derived by Kabat et al. [2006] using comparative genomics in nematodes, 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 assumed to be informative for alternative splicing. We use a set of 210 splicing regulators (ESE and IRS), and represent each instance as a 210-dimensional vector of feature counts (*i.e.,* for each feature or regulator, we record the count of occurrences of that particular feature in the EST sequence). The alternatively spliced sequences are considered to be positive, and the constitutively spliced sequences are negative.

2.4.2 Evaluation Procedure and Parameter Tuning

An objective evaluation of any predictive model requires the use of the cross-validation technique, to avoid any (un)fortunate selection of instances. Similar to Rätsch et al. [2005], we started with a fully labeled data set and simulated unlabeled data by withholding the real labels during the training process, which allowed us to experiment with different amounts of labeled/unlabeled data. We used 5-fold cross validation to create our training/validation/test datasets (while maintaining the original distribution in each fold). Out of the 5 folds, 3 were used for training, one for validation and one for test. Furthermore, the training was randomly split into labeled and unlabeled. In the experiments where we vary the amount of unlabeled data, labeled data represents 5% of the training folds, and the unlabeled data varies from 15% to 95%. These ratios are consistent with ratios that have been previously used in the literature for other applications. Previous work on textual data has reported that SSL algorithms can result in better performance as compared to their supervised counterparts, when only a small amount of labeled data is available, *e.g.,* [Nigam et al., 2000] when 500 labeled documents and 10,000 unlabeled documents were used (*i.e.*, 5% labeled), the accuracy reached 70%.

In the experiments where we vary the amount of labeled data, unlabeled data represents 70% of the training folds, and the labeled data varies from 5% to 30%. Based on the size of our own dataset, when using 30% of training data as labeled data and 70% as unlabeled, the unlabeled to labeled ratio is less than 3:1. According to the literature [Nigam et al., 2000], and also based on our preliminary experimentation, semi-supervised learning does not show much benefit when the amount of unlabeled data is not significantly larger than the amount of labeled data. Thus, for the experiments where we varied the amount of labeled data, we kept the unlabeled data to 70% in order to satisfy the condition of unlabeled data being "larger" than the labeled data.

Given that our data is skewed – there are approximately five times more *"constitutive"* instances as compared to *"alternatively spliced"* instances – the accuracy of the predictions would not reflect the true quality of the classifiers Therefore, we report the performance in terms of area under the Receiver Operating Characteristic curve (auROC) [Huang and Ling, 2005].

The machine learning algorithms that we used have several parameters that need to be set. To select the optimal values for these parameters, models with different parametric values are evaluated on a subset of the data, *i.e.,* the validation set. A range of values (grid search) is used to build different models. Selecting the model that generalizes best on the validation set, will presumably lead to a good generalization on the test set also. For example, in the case of Self-training and Co-training, we must assign values to the *sample size* (how many instances we are to classify at each iteration) and to the *number of iterations* that the algorithm should execute. We included the following values for the sample size in the grid: 25, 50, 75, 100, 125, 150, 200, 250, 300. As for the number of iterations, we used the following values: 1, 5, 10, 15, 20, 25, 50, 100, 150, 200, 250, 300.

Moreover, SVM and RF have specific parameters. For example, in the case of the SVM classifier, we employ a Gaussian kernel, where we need to tune the *cost error* (values tried: 0.1, 0.25, 0.5, 0.75, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 10) and *gamma* (0.05, 0.1, 0.2, 0.3, 0.5) parameters. For RF, the tunable parameters are *depth* of the tree (values from 0 to 30, in steps of 2), *number of features* to consider in random feature selection (values from 0 to 2 x log(number of features + 1)), and *number of trees* in the forest (values from 10 to 50 in steps of 2). In total, for all 3 base classifiers that we used, considering the supervised and semi-supervised cases, the variation of labeled to unlabeled data and the 5 folds, we trained approximately 328 million individual models.

To assess the behavior of the semi-supervised algorithms, we compare their performance (specifically, auROC values) against baselines in each experiment. These values will give us an indication of how much improvement can be expected from using the unlabeled data in a particular case. First, supervised learning just from the labeled subset will give us a lower bound (**LB**) for performance. This will show how well we can learn from the little amount of labeled data available. Next, we run another supervised version of the algorithms, maintaining the same folds, and assuming that no data in the training set is unlabeled. Recall that we deliberately treat some instances as unlabeled, in order to simulate the semi-supervised environment. If all the unlabeled data instances were in fact labeled, this would represent an upper bound (**UB**). The UB value mainly estimates how good is the set of motifs used and gives an upper limit for how well we can expect to do in the semi-supervised framework. We run two-tailed paired t-tests, as opposed to one-tailed t-tests, to identify statistically significant differences in either direction, on all the semi-supervised algorithms for all the variations of unlabeled data. The test determines if the difference between the SSL algorithm and the lower bound is statistically significant or not [Dietterich, 1998].

2.4.3 Research Questions

Our experiments are designed to address the following research questions:

[1.]*How does the performance of semi-supervised learning algorithms compare with that of the corresponding supervised learning algorithms, when the amount of labeled data is very small?*

Given that the dataset used in our study consists of 3,018 instances, 5-fold cross validation splits the training into 2,414 instances and the test into 604 instances. Out of the 2,414 instances, one split was used in parameter tuning, and from the remaining 3 folds, 5% were randomly picked as labeled (*i.e.*, 90 instances). Since biological sequences are usually harder to annotate, we decided to use no less than 5% labeled data in our study.

[2.]*How do the semi-supervised algorithms studied compare with each other and which one shows best improvements as compared to their supervised counterparts?*

We compare the SSL approaches described in Section 2.3 with each other, to understand what type of approach is more appropriate for the classification problem considered, and which one shows the best improvements as compared to the supervised counterparts learned from labeled data only (whose performance can be seen as a lower bound for the semi-supervised algorithms which make use of both labeled and unlabeled data). We also compare the performance with that of supervised classifiers learned by using all data as labeled data, which is possible, given that all our data is labeled (we simulate unlabeled data by "hiding" the labels of the instances in the unlabeled set). The performance of these supervised classifiers can be seen as an upper bound for the semi-supervised learning.

[3.]*How does the performance of the semi-supervised learning algorithms vary with the* *amount of unlabeled data?*

In general, the performance of semi-supervised learning largely depends on the availability of unlabeled data. The more unlabeled data available, the better the semi-supervised classifiers learned, and consequently a higher improvement over the supervised lower bound classifiers is expected. However, it can happen that larger amounts of unlabeled data will produce more noise, which in turn can degrade the performance.

[4.]*How does the performance of the semi-supervised learning algorithms vary with the amount of labeled data?*

Intuitively, the more labeled data is available, the better the classifier learned and the unlabeled instances can be predicted more accurately. Adding the extra knowledge can sharpen the classifiers' quality and enhance their prediction power on further iterations. As a whole, the performance of semi-supervised learning is expected to increase with the increase of the amount of labeled data.

2.5 Results

We run three sets of experiments, each set is using either NBM, SVM or RF as base classifier and each set is summarized in two tables. For each table, we have two supervised bounds (LB and UB obtained using the base classifier in a supervised mode) and four semi-supervised classifiers (EM, EMW, ST, CoT built on top of the base classifier). The first table shows the results of the experiments where we vary the amount of unlabeled data from 15% to 95% of the training folds, while the amount of labeled data is fixed at 5%. The LB column of this table is constant, and it represents the supervised lower bound of the base classifier when trained using only 5% labeled data. The second smaller table, shows experiments where we vary the amount of labeled data from 5% to 30%, while the amount of unlabeled data is fixed at 70%. The values in bold represent the best performance for each experiment and the italicized values represent the statistically significant variations. We also show the standard deviation of the averaged auROC values.

Table 2.1 shows the SSL experiments when NBM is used as the base classifier and the amount of unlabeled data is varied. EMW, followed by EM, exhibit a consistent increase over the LB as more unlabeled data is used. Furthermore, EMW is the only SSL method that shows statistically significant improvements over the supervised lower bound throughout all of the SSL experiments using NBM, without exception. EM seems to benefit more from larger amounts of unlabeled data (more than 20%). ST and CoT outperform the other methods in terms of auROC values, and CoT shows the highest classification performance over all the SSL algorithms but surprisingly, ST and CoT, do not always have a statistically significant performance improvement over the LB, as shown by the paired t-test.

	LB	EM	EMW	ST	CoT	UB
	(supervised)	(SSL)	(SSL)	(SSL)	(SSL)	(supervised)
15%U	0.816±0.033	0.810±0.034	*0.822±0.033*	0.828±0.040	**0.869±0.039**	0.896±0.030
20%U	0.816±0.033	0.824±0.031	*0.825±0.033*	0.834±0.042	**0.903±0.039**	0.905±0.027
25%U	0.816±0.033	0.830±0.012	*0.832±0.027*	*0.848±0.037*	**0.870±0.029**	0.907±0.027
30%U	0.816±0.033	0.837±0.024	*0.836±0.029*	*0.845±0.024*	**0.872±0.030**	0.913±0.025
35%U	0.816±0.033	0.845±0.028	*0.839±0.030*	***0.846±0.033***	0.845±0.039	0.913±0.024
40%U	0.816±0.033	0.852±0.026	*0.841±0.029*	*0.844±0.038*	**0.866±0.034**	0.918±0.022
45%U	0.816±0.033	0.855±*0.024*	*0.844±0.028*	0.845±0.036	**0.875±0.030**	0.920±0.021
50%U	0.816±0.033	*0.854±0.025*	*0.845±0.029*	*0.862±0.023*	**0.873±0.043**	0.924±0.020
55%U	0.816±0.033	*0.851±0.024*	*0.845±0.028*	*0.850±0.027*	***0.885***±**0.017**	0.925±0.022
60%U	0.816±0.033	0.849±0.016	*0.847±0.025*	0.851±0.025	***0.900±0.024***	0.924±0.021
65%U	0.816±0.033	0.849±0.028	*0.848±0.027*	*0.852±0.025*	***0.873±0.039***	0.924±0.021
70%U	0.816±0.033	0.847±0.030	*0.849±0.026*	*0.859±0.019*	***0.881±0.027***	0.923±0.019
75%U	0.816±0.033	*0.853±0.024*	*0.851±0.026*	0.844±0.030	***0.872±0.036***	0.925±0.020
80%U	0.816±0.033	0.845±0.040	*0.852±0.028*	***0.860±0.023***	0.845±0.047	0.926±0.020
85%U	0.816±0.033	*0.857±0.020*	*0.857±0.025*	*0.859±0.023*	***0.895±0.030***	0.927±0.018
90%U	0.816±0.033	0.851±0.023	*0.855±0.025*	*0.863±0.030*	**0.888±0.028**	0.928±0.018
95%U	0.816±0.033	*0.848±0.022*	*0.855±0.024*	*0.865±0.025*	***0.905±0.019***	0.929±0.018

Table 2.1: **Semi-supervised Learning Results Based on Naïve Bayes When Varying the Amount of Unlabeled Data**: Averaged auROC values for the 5 folds and the standard deviation for experiments using NBM as base classifier, when varying the amount of unlabeled data from 15% to 95%, while maintaining a fixed labeled amount of 5%. LB and UB represent the supervised lower and upper bounds, and the semi-supervised algorithms are Expectation Maximization (EM) and the weighted variant (EMW), Self-training (ST) and Co-training (CoT). Values in bold font represent the best performance and italicized values represent statistically significant variations.

Table 2.2 shows the SSL experiments when NBM is used as the base classifier and the amount of labeled data is varied. We observe an expected trend; the larger the amount of labeled data the base classifiers are trained on, the better the auROC values. Interestingly enough, when the amount of labeled data is increased to 25% and 30%, CoT outperforms the upper bound, suggesting that some of the instances from the unlabeled pool receive a label that may be different from the actual label they have in the original dataset. Thus, CoT, by "correcting" these instances, ultimately reaches a classification accuracy that is higher than the accuracy achieved by the supervised model.

	LB	EM	EMW	ST	CoT	UB
	(supervised)	(SSL)	(SSL)	(SSL)	(SSL)	(supervised)
5%L	0.816±0.033	0.847±0.030	*0.849±0.026*	*0.859±0.019*	***0.881±0.027***	0.923±0.019
10%L	0.852±0.032	0.866±0.027	*0.872±0.023*	*0.875±0.020*	0.903±*0.023*	0.924±0.019
15%L	0.872±0.012	0.874±0.014	*0.883±0.009*	0.882±0.019	***0.913±0.010***	0.925±0.019
20%L	0.887±0.011	0.883±0.014	*0.896±0.011*	0.890±0.011	**0.921±0.017**	0.926±0.017
25%L	0.904±0.015	*0.885±0.015*	*0.907±0.015*	0.905±0.012	**0.930±0.008**	0.927±0.018
30%L	0.912±0.015	*0.894±0.016*	*0.914±0.015*	0.910±0.011	**0.934±0.014**	0.929±0.018

Table 2.2: **Semi-supervised Learning Results Based on Naïve Bayes When Varying the Amount of Labeled Data**: Averaged auROC values for the 5 folds and their standard deviation for experiments using NBM as base classifier, when varying the amount of labeled data from 5% to 30%, while maintaining a fixed amount of unlabeled data of 70%. LB and UB represent the supervised lower and upper bounds, and the semi-supervised algorithms are Expectation Maximization (EM) and the weighted variant (EMW), Self-training (ST) and Co-training (CoT). Values in bold font represent the best performance and italicized values represent statistically significant variations.

Table 2.3 shows the SSL experiments when SVM is used as the base classifier and the amount of unlabeled data is varied. When learning the initial model from 5% labeled data, EM and EMW do not seem to benefit from the unlabeled data, as their corresponding auROC values are smaller than the supervised lower bound. ST and CoT show better performance than EM and EMW in terms of auROC values, but sometimes their classification performance is surpassed by the LB. CoT achieves the leading values overall. The improvement in performance of EM over the LB has been found statistically significant by the paired t-test in the majority of cases, followed by EMW and ST. The gain recorded by CoT over the LB is not statistically significant, although it reaches the highest auROC values among all the SSL techniques.

	LB	EM	EMW	ST	CoT	UB
	(supervised)	(SSL)	(SSL)	(SSL)	(SSL)	(supervised)
15%U	0.815±0.042	*0.733±0.028*	0.792±0.04	0.803±0.041	**0.830±0.044**	0.839±0.026
20%U	0.815±0.042	0.783±0.017	*0.776±0.029*	**0.810±0.046**	0.804±0.049	0.876±0.026
25%U	0.815±0.042	0.740±0.048	*0.749±0.038*	*0.802±0.044*	**0.810±0.043**	0.884±0.026
30%U	0.815±0.042	*0.769±0.042*	*0.737±0.038*	0.808±0.040	**0.874±0.023**	0.880±0.026
35%U	0.815±0.042	*0.736±0.030*	*0.741±0.037*	0.791±0.048	**0.841±0.024**	0.883±0.025
40%U	0.815±0.042	*0.758±0.044*	0.737±0.046	0.800±0.045	**0.845±0.034**	0.892±0.029
45%U	0.815±0.042	0.784±0.027	*0.741±0.044*	0.796±0.045	**0.808±0.072**	0.911±0.023
50%U	0.815±0.042	0.765±0.042	0.743±0.035	0.832±0.015	**0.878±0.037**	0.914±0.020
55%U	0.815±0.042	*0.711±0.037*	0.727±0.033	0.835±0.022	**0.860±0.052**	0.911±0.018
60%U	0.815±0.042	*0.706±0.045*	0.765±0.032	**0.838±0.019**	0.804±0.054	0.926±0.017
65%U	0.815±0.042	*0.786±0.041*	0.779±0.032	0.843±0.025	**0.850±0.028**	0.928±0.016
70%U	0.815±0.042	*0.785±0.042*	0.796±0.034	0.858±0.011	**0.869±0.034**	0.931±0.018
75%U	0.815±0.042	*0.715±0.043*	0.775±0.036	**0.850±0.013**	0.837±0.042	0.935±0.015
80%U	0.815±0.042	*0.775±0.041*	0.767±0.038	0.851±0.020	**0.864±0.031**	0.940±0.013
85%U	0.815±0.042	*0.717±0.045*	*0.756±0.035*	0.810±0.013	**0.874±0.053**	0.944±0.010
90%U	0.815±0.042	*0.780±0.043*	*0.758±0.029*	0.809±0.012	**0.873±0.047**	0.941±0.010
95%U	0.815±0.042	*0.783±0.042*	*0.783±0.040*	0.812±0.008	**0.861±0.147**	0.949±0.009

Table 2.3: **Semi-supervised Learning Results Based on SVM When Varying the Amount of Unlabeled Data**: Averaged auROC values for the 5 folds and their standard deviation for experiments using SVM as base classifier, when varying the amount of unlabeled data from 15% to 95%, while maintaining a fixed labeled amount of 5%. LB and UB represent the supervised lower and upper bounds, and the semi-supervised algorithms are Expectation Maximization (EM) and the weighted variant (EMW), Self-training (ST) and Co-training (CoT). Values in bold font represent the best performance and italicized values represent statistically significant variations.

	LB	EM	EMW	ST	CoT	UB
	(supervised)	(SSL)	(SSL)	(SSL)	(SSL)	(supervised)
5%L	0.815±0.042	*0.785±0.042*	0.796±0.034	0.858±0.011	**0.869±0.034**	0.931±0.018
10%L	0.856±0.027	*0.832±0.030*	0.846±0.027	0.826±0.017	**0.867±0.036**	0.926±0.017
15%L	0.837±0.025	*0.847±0.027*	0.840±0.036	0.788±0.036	***0.911±0.011***	0.936±0.015
20%L	0.844±0.025	0.864±0.024	0.843±0.040	0.816±0.037	***0.915±0.016***	0.935±0.012
25%L	0.855±0.026	0.861±0.017	0.888±0.029	0.814±0.025	**0.922±0.020**	0.939±0.012
30%L	0.901±0.016	0.894±0.024	0.907±0.019	0.854±0.021	**0.934±0.020**	0.949±0.011

Table 2.4: **Semi-supervised Learning Results Based on SVM When Varying the Amount of Labeled Data**Averaged auROC values for the 5 folds and their standard deviation for experiments using SVM as base classifier, when varying the amount of labeled data from 5% to 30%, while maintaining a fixed amount of unlabeled data of 70%. LB and UB represent the supervised lower and upper bounds, and the semi-supervised algorithms are Expectation Maximization (EM) and the weighted variant (EMW), Self-training (ST) and Co-training (CoT). Values in bold font represent the best performance and italicized values represent statistically significant variations.

As it can be seen from Table 2.4, when the amount of labeled data is varied, CoT is the most promising algorithm and it is closely approaching the UB as it learns from more labeled data. ST best leverages the unlabeled information in the first case, when it is trained on 5% labeled data, but in all other cases it is surpassed by the LB. EM and EMW exhibit a steady increase in performance with the increase of labeled data, yet they do not always exceed the LB values.

	LB	EM	EMW	ST	CoT	UB
	(supervised)	(SSL)	(SSL)	(SSL)	(SSL)	(supervised)
15%U	0.867±0.05	0.821±0.042	*0.841±0.042*	0.883±0.022	**0.897±0.053**	0.933±0.020
20%U	0.867±0.05	*0.797±0.035*	0.839±0.034	**0.882±0.019**	0.872±0.041	0.930±0.020
25%U	0.867±0.05	0.817±0.039	*0.793±0.019*	0.896±0.028	**0.899±0.049**	0.937±0.012
30%U	0.867±0.05	0.786±0.028	0.805±0.069	0.903±0.024	**0.904±0.042**	0.939±0.009
35%U	0.867±0.05	0.805±0.049	*0.821±0.030*	**0.893±0.020**	0.883±0.048	0.947±0.012
40%U	0.867±0.05	0.815±0.028	0.825±0.033	**0.901±0.031**	0.891±0.061	0.953±0.017
45%U	0.867±0.05	0.812±0.044	0.822±0.045	0.905±0.026	**0.916±0.053**	0.955±0.016
50%U	0.867±0.05	0.788±0.052	0.811±0.053	**0.911±0.016**	0.902±0.061	0.965±0.009
55%U	0.867±0.05	0.791±0.035	*0.794±0.058*	**0.913±0.030**	0.903±0.054	0.963±0.020
60%U	0.867±0.05	0.802±0.032	*0.805±0.042*	0.909±0.014	**0.926±0.052**	0.959±0.005
65%U	0.867±0.05	0.823±0.036	0.814±0.033	0.910±0.015	**0.924±0.049**	0.967±0.008
70%U	0.867±0.05	*0.801±0.017*	0.752±0.111	0.912±0.014	**0.940±0.034**	0.966±0.007
75%U	0.867±0.05	*0.813±0.039*	*0.781±0.064*	0.911±0.031	**0.933±0.049**	0.964±0.008
80%U	0.867±0.05	*0.795±0.042*	0.785±0.062	0.898±0.032	**0.921±0.059**	0.969±0.008
85%U	0.867±0.05	*0.798±0.039*	*0.827±0.045*	0.911±0.026	**0.940±0.056**	0.969±0.007
90%U	0.867±0.05	0.843±0.012	*0.784±0.069*	0.919±0.021	**0.926±0.047**	0.970±0.008
95%U	0.867±0.05	0.777±0.047	*0.774±0.055*	0.912±0.025	**0.942±0.058**	0.971±0.007

Table 2.5: **Semi-supervised Learning Results Based on Random Forest When Varying the Amount of Unlabeled Data**: Averaged auROC values for the 5 folds and their standard deviation for experiments using RF as base classifier, when varying the amount of unlabeled data from 15% to 95%, while maintaining a fixed labeled amount of 5%. LB and UB represent the supervised lower and upper bounds, and the semi-supervised algorithms are Expectation Maximization (EM) and the weighted variant (EMW), Self-training (ST) and Co-training (CoT). Values in bold font represent the best performance and italicized values represent statistically significant variations.

Table 2.5 shows the SSL experiments when RF is used as the base classifier, while varying the amount of unlabeled data. The unlabeled data has proven to be advantageous for ST, and most useful for CoT, where all the values of ST and CoT are surpassing the LB. Both ST and CoT values are relatively close for the first experiments, when the amount of unlabeled data is increased from 15% to 55% but as more unlabeled data is used during training, from

	LB	EM	EMW	ST	CoT	UB
	(supervised)	(SSL)	(SSL)	(SSL)	(SSL)	(supervised)
5%L	0.867±0.050	*0.801±0.017*	0.752±0.111	0.912±0.014	**0.940±0.034**	0.966±0.007
10%L	0.904±0.020	0.881±0.039	*0.872±0.018*	***0.928±0.017***	0.950±0.017	0.963±0.015
15%L	0.922±0.018	0.906±0.026	0.898±0.019	0.932±0.021	**0.960±0.016**	0.972±0.006
20%L	0.935±0.019	0.920±0.016	0.930±0.019	***0.943±0.019***	0.940±0.014	0.965±0.012
25%L	0.939±0.016	0.919±0.020	0.936±0.022	0.947±0.012	**0.960±0.011**	0.968±0.013
30%L	0.948±0.012	0.929±0.023	0.934±0.015	0.942±0.013	**0.964±0.012**	0.969±0.007

Table 2.6: **Semi-supervised Learning Results Based on Random Forest When Varying the Amount of Labeled Data**: Averaged auROC values for the 5 folds and their standard deviation for experiments using RF as base classifier, when varying the amount of labeled data from 5% to 30%, while maintaining a fixed amount of unlabeled data of 70%. LB and UB represent the supervised lower and upper bounds, and the semi-supervised algorithms are Expectation Maximization (EM) and the weighted variant (EMW), Self-training (ST) and Co-training (CoT). Values in bold font represent the best performance and italicized values represent statistically significant variations.

60% to 95%, CoT exhibits the best performance. On the other hand, EM and EMW do not benefit from the unlabeled data, as their performance is constantly exceeded by the LB, in both cases (when the unlabeled data is varied, as well as when the labeled data is varied). When the amount of labeled data is varied in Table 2.6, CoT is still leading the results, followed closely by ST; they both outperform the LB.

We now summarize the results by addressing each of the research questions:

[1.]*How does the performance of semi-supervised learning algorithms compare with that of the corresponding supervised learning algorithms, when the amount of labeled data is very small?*

To answer the first question, we have varied the ratio of labeled to unlabeled data from 5% labeled to 15% unlabeled, to 5% labeled to 95% unlabeled. Overall, our experiments show that having at least 3 times more unlabeled data than labeled data is usually enough to benefit from SSL.

[2.]*How do the semi-supervised algorithms studied compare with each other and which one shows best improvements as compared to their supervised counterparts?*

To answer the second question, we have found that the highest increase over the LB has been obtained by CoT in the case of NBM (around 9%), although the values have not been established as statistically significant by the paired t-test. CoT is closely followed by ST and almost always both of them surpass the LB. The EM-type approaches show statistically significant improvements, especially in the case of NBM, but they are not always advantageous, for instance when they are combined with SVM and RF as base classifiers.

[3.]*How does the performance of the semi-supervised learning algorithms vary with the amount of unlabeled data?*

We have found that all SSL algorithms show most promise when larger amounts of unlabeled data are used, and the auROC values rise by 3-4% in all SSL cases. However, we observe a consistent increase for the EM classifiers, while the performance of ST and CoT can increase and decrease depending on the quality of the data added when moving from one unlabeled data experiment to another. One possible explanation for this is that EM is adding all unlabeled data at each subsequent iteration (although with weights proportional to how confidently that data was classified), while ST and CoT add only the most confident examples. Thus, if some examples are mislabeled with high confidence, they will degrade the overall performance of the classifier.

[4.]*How does the performance of the semi-supervised learning algorithms vary with the amount of labeled data?*

Having more labeled data to learn the initial models from, leads to better final models and improved overall performance.

Overall, the best base classifier is RF, reaching the highest auROC values, for both supervised (up to 0.971) and semi-supervised (up to 0.964) learning. NBM is leveraging the unlabeled data better than RF and SVM, with up to 9% increase over the LB when utilizing 5% labeled and 95% unlabeled data, as opposed to an 8% increase in the case of RF, and 5% in the case of SVM. However, if the available labeled data makes up more than 5% of the training dataset, SVM improves upon the LB by 3%, the highest of all the base classifiers. The best SSL algorithm is CoT, which proves that having two views "informing" each other about high confidence labels can overcome the noise from the unlabeled data. In practice, when such views are unavailable, ST with NBM or RF can take advantage of the unlabeled data available and improve upon a supervised classifier.

2.6 Conclusion and Future Work

In this work, we have carried out a comparison of iterative semi-supervised learning algorithms applied to the DNA prediction problem of distinguishing between alternatively spliced exons and constitutive exons. Our findings confirm that leveraging the unlabeled data during SSL can boost supervised performance. As reported in other studies, Co-training performs the best, followed by Self-training and the EM techniques. Generally, the more unlabeled data is used in training, the better the resulting classifier. This is a promising conclusion which suggests that SSL algorithms can be successfully used for DNA sequence classification. Overall, the best auROC values, corresponding to most accurate classifications have been recorded in the case of RF, for both supervised and semi-supervised paradigms. Given that the Random Forest algorithm is an ensemble type algorithm, it is of interest to study other ensemble approaches in the context of SSL.

Also as part of future research, we would like to study another class of semi-supervised algorithms, namely transductive methods. Transductive learning, as opposed to inductive learning, solves an easier, more specific problem, as its goal is generally to label the unlabeled data (used during training), as opposed to learning a classifier that will be used on future unlabeled, unseen data. For example, we would like to use transductive SVMs and graph-based methods, along with specialized kernels and similarity metrics.

Another direction worth investigating is the use of different feature representations of the DNA sequences to understand how they perform in a semi-supervised scenario. Thus, we are interested in a similar comparison study using intrinsic features that are generated directly from the instances available (*e.g.,* using a sliding window approach), without looking at the class labels, under the assumption that labeled data is scarce.

Chapter 3

An Empirical Study of Self-training and Data Balancing Techniques for Splice Site Prediction

3.1 Introduction

Many domains, such as online social media, e-commerce, scientific literature, medical monitoring and diagnosis, risk management, image recognition, and cyber-security, are constantly generating vast amounts of data. As a result, the current challenges now lie with the analysis and interpretation of the data. The same trend can be observed in biological fields [Baldi and Brunak, 2001]. Cost-effective, high-throughput Next Generation Sequencing technologies have enhanced the production of raw genomic data, which currently occurs at a much faster rate than its annotation. Computational machine learning techniques can help with data annotation, including genomic data, but the effectiveness of machine learning classifiers depends on the training data available, specifically the quality and quantity of the labeled instances, and their class distribution.

Having approximately the same number of instances in each class is critical for producing quality classifiers that can correctly identify all classes. However, in practice, there are frequent situation when one of the categories (classes) to predict is sporadic, usually the class of interest. This is due to the fact that the instances of the minority class are either harder to acquire, or that they are indeed atypical relative to the other class(es). Datasets that exhibit highly imbalanced distributions are common in many applications, for example medical diagnosis, fraud detection, network intrusion, error-prone software modules, image recognition. In such circumstances, when there are significant differences between the class prior rates, standard classifiers that learn well from balanced distributions can sometimes be negatively influenced by the imbalance phenomenon, showing bias towards the majority class. The data imbalance problem has been studied over many decades and domains in the supervised learning framework. We will briefly review data-level and algorithm-level approaches to the data imbalanced problem below; for a comprehensive survey, the reader is directed to [He and Garcia, 2009].

Datasets that are naturally imbalanced across classes can be adjusted before being presented to a learning algorithm by the means of *external*, data-level techniques, such as re-sampling. The easiest way to balance an imbalanced dataset is to use under-sampling, which simply discards the extra instances from the majority class. This technique can decrease the learning computation time, especially when the minority class is exceeded by a few orders of magnitude by the majority class, in which case under-sampling would dispose of a large portion of data. The direct drawback of such an approach is the obvious loss of information that goes with discarding many instances. A careful, more informative selection of the instances to be kept, rather than random sampling, could potentially alleviate some of the knowledge loss [Korecki et al., 2008; Chawla et al., 2002]. Another data re-sampling remedy, contrasting under-sampling, is over-sampling, where the number of instances in the minority class is artificially increased to coincide with the number of instances in the majority class. Longer computational requirements (in terms of both time and memory) and overfitting, due to artificial instance generation, are the main drawbacks of this technique.

Aside from data-level techniques, solutions targeting the algorithm, or *internal* techniques, have also been developed, such as cost-sensitive learning [Ling and Sheng, 2008] and active learning [Li et al., 2012b]. Furthermore, while the concept of classifier ensembles emerged as a way of improving the performance of a single classifier, ensembles have been found to be useful also for skewed distributions. Galar et al. [2012] discuss combining classifiers using bagging, boosting and hybrid-approaches in the supervised framework, to deal with imbalanced datasets.

Similar to other fields, the data imbalance problem has been addressed for many bioinformatics tasks [Wei and Dunbrack Jr, 2013; Batuwita and Palade, 2010; 2012; Lusa and Blagus, 2010; You et al., 2011] in the supervised framework, under the assumption that a sufficiently large amount of labeled data is available. However, in bioinformatics, labeled examples are traditionally obtained via wet-lab experiments, which are expensive and time-consuming methods and necessitate biological know-how. In contrast, unlabeled examples are easily accessible, and on a much larger scale. Such a scenario, in which limited amounts of labeled data and massive amounts of unlabeled data are available, is particularly favorable for automated semi-supervised learning (SSL) algorithms. Exploiting unlabeled data to improve a supervised classifier's performance is an attractive yet challenging task, and an active research topic [Wang and Chen, 2013; Singh et al., 2009]. Semi-supervised learning approaches have been used to address bioinformatics problems, such as disease genes detection [Nguyen and Ho, 2012], prediction of cancer recurrence based on gene expression [Shi and Zhang, 2011], and protein classification [Weston et al., 2005; 2006; Kall et al., 2007; Craig and Liao, 2007]. Unfortunately, it is not unusual for the unlabeled data to deteriorate the performance of a classifier that would otherwise (*i.e.*, in a purely supervised framework) yield a quality prediction model [Li and Zhou, 2011; Catal and Diri, 2009].

Imbalanced distributions further contribute to the difficulty of the problem, and for bioinformatics, there are many problems (*e.g.*, splice site recognition, promoter prediction, protein classification) that suffer from insufficient labeled data as well as disproportionate class rates. This particular niche, semi-supervised learning from imbalanced bioinformatics distributions, has not been much studied, with the exception of a few notable works [Weston et al., 2005; Kundu et al., 2013; Kondratovich et al., 2013] that focus on protein-related problems. For DNA classification, the class imbalance problem has been mostly explored in the supervised framework [García-Pedrajas et al., 2012], rather than the semi-supervised learning framework.

Splice sites are conserved nucleotide dimers found at the interface between exons and introns. They can be donor splice sites, signaled by "GT" and situated at the 5' end of the intron, or acceptor splice sites, indicated by "AG" and situated at the 3' end of the intron. The correct identification of splice sites is an essential task in the genome annotation process. The major difficulty comes from the fact that such dimers occur very frequently throughout the entire genome and their simple presence is not enough to declare a splice site. However, these regulatory regions exhibit certain properties that are easily recognizable by the snRNA-proteins in the pre-mRNA splicing process, which makes them good candidates for classification algorithms that can capture these similarities. The prediction (identification) of splice sites is a problem for which the natural positive (true splice site) to negative (decoy site) ratio is very high, approximately 1% of the "AG" dimers occurring in a genome correspond to splice sites.

In this work, we study the suitability of semi-supervised learning from skewed splice site datasets. More precisely, we are interested in investigating how the positive-to-negative ratio affects semi-supervised learning when external data-level re-balancing techniques are used. Furthermore, as we vary the positive-to-negative and labeled-to-unlabeled ratios, we compare the semi-supervised classifiers with supervised classifiers learned only from the small amounts of labeled data available, to understand when semi-supervised learning is preferable to supervised learning. We use self-training [Yarowsky, 1995] based on Naïve Bayes, as the main SSL approach in our study. The splice site datasets that we use exhibit imbalance degrees of 1-to-99. To examine how the performance is influenced by different levels of class imbalance, we subsample the original datasets in order to obtain lower rates of imbalance, then gradually increase the imbalance level.

The rest of this paper is organized as follows: Section 3.2 describes the approaches studied. We explain how we designed our experiments in Section 3.3: the data used and the feature representation are described in Section 3.3.1, the research questions we are addressing are enumerated in Section 3.3.2, and the evaluation procedure is described in Section 3.3.3. Experimental results and discussions can be found in Section 3.4. In Section 3.5, we contrast our study with other related studies. We summarize our work, draw some conclusions and propose future research directions in Section 3.6.

3.2 Semi-supervised Approaches for Learning from Imbalanced Data

We use self-training based on Naïve Bayes and focus on balancing techniques at the data-level, namely under- and over-sampling, and dynamic-balancing. We chose Naïve Bayes due to its linear complexity and high scalability. Moreover, other algorithms depend on several parameters and their tuning is often critical for obtaining good generalization capability. The results presented in this paper required the individual training of 2,730 models, and extra parameter tuning would have increased this number considerably. Naïve Bayes is purely a frequency estimator based on feature occurrence counts, a concept in harmony with the idea that the surrounding regions of the splice sites share a consensus in statistical patterns.

3.2.1 Self-Training from Imbalanced Data

Yarowsky [1995] introduced self-training in the mid-nineties for a text disambiguation problem. Since then, self-training has produced successful results for other problems in computational linguistics [Collins and Singer, 1999], object detection [Rosenberg et al., 2005], and bioinformatics [Kundu et al., 2013].

As illustrated in Algorithm 1, in self-training, a base learner is first trained on just the labeled data. Next, a randomly chosen sample from the unlabeled pool is labeled using the classifier trained on just the labeled data. From these newly labeled instances, the most confidently classified instances are added to the original labeled set and the classifier is re-trained on the augmented labeled set. This process is iterative, and at each step, a new sample of unlabeled instances is classified with the current classifier, and then used in re-training. One important detail, relevant to the data imbalance problem studied in this work, is that in the classical self-training algorithm, the number of newly labeled instances that are added to the originally labeled training set is chosen such that the positive-to-negative ratio displayed by the labeled data is maintained (*e.g.,* if the class ratio in the initial labeled set is 1-to-5, then 6 examples are extracted from the unlabeled pool and added to the labeled dataset: the topmost confident positive prediction along with the top 5 most confident negative predictions). The iterations continue until a criterion is met, for example until the unlabeled pool is exhausted, or the algorithm reaches a fixed number of iterations. In this study, we refer to this algorithm as *self-training from imbalanced data* (STI) because there is no modification made towards balancing.

Algorithm 1 Self-training Algorithm Yarowsky [1995]

```
Given: a training set comprised of labeled and unlabeled data $D = (D_l, D_u)$, $|D_l| \ll |D_u|$, fixed sample size $S$
Create $U$ by picking $S$ random instances from $D_u$ and update $D_u = D_u - U$
repeat
    Train classifier on current $D_l$
    Classify the instances in $U$ using the classifier
    Select most confident newly labeled instances from $U$ to add to $D_l$, such that the original class distribution is maintained
    Randomly pick instances from $D_u$ to replace the selected most confident labeled instanced in $U$
until $D_u$ is empty
```

3.2.2 Self-Training with Data Balancing

Given the highly skewed datasets of splice sites, where the true acceptors appear in approximately 1% of the total number of sequences, it is important to evaluate some popular re-sampling techniques on this DNA prediction problem in the semi-supervised context. In this paper, we use two *external*, or *data-level* balancing schemes: under-sampling and over-sampling. Both techniques aim to obtain a uniform distribution of instances in each class. In under-sampling (of the majority class), we keep all positive instances and randomly pick a negative number of instances to create a balanced labeled training set to feed to the self-training classifier. We call this variant *Self-Training with Under-sampling* **(STU)**.

In over-sampling (of the minority class), we artificially create positive instances to compensate for the larger number of negative instances. Instead of randomly duplicating positive instances in order to achieve an equal number of examples in each class, we chose to utilize the Synthetic Minority Over-sampling Technique (SMOTE) [Chawla et al., 2002], where instances of interest are generated by interpolation between other positive instances, in the feature space. These new instances are novel, to some extent, and the idea is to avoid overfitting, which can be caused by exact instance replication. We named this variant *Self-Training with Over-sampling* **(STO)**. From the self-training perspective, since the labeled data that the learner is initially trained on is now balanced (via under- or SMOTE over-sampling), only two instances are added into the labeled pool after each iteration for both STU and STO: the top most confident from each class, such that the uniform distribution that was obtained via re-sampling is maintained.

In addition to under- and over-sampling, we also study an alternative way of re-calibrating the classes via the self-training algorithm, specifically we use a dynamic balancing technique. Starting with the original imbalanced labeled dataset, we train a classifier and use it to predict the unlabeled data, similar to the classical approach, STI. The difference is that only the instances that are predicted as positive by the base learner are added to the original labeled set, more precisely, the top-most confidently predicted positive instance is added at each iteration. This approach re-adjusts the original class distribution by utilizing newly labeled instances, instead of artificially generating positive examples like the over-sampling techniques. We named this new variant *Self-Training with Positive* (**STP**). We want to emphasize that unlike the classical self-training algorithm (Algorithm 1) that enforces the constraint on the class distribution to be maintained after each iteration, STP is only adding positive instances, thus compelling a dynamic re-calibrating of the prior.

3.2.3 Supervised Baselines

To evaluate the performance of semi-supervised learning and to observe the effects of the labeled versus unlabeled data, we also train supervised classifiers. We compare each semi-supervised algorithm described in Section 3.2 against its corresponding supervised variant, built using the same re-sampling technique. The supervised counterparts can be seen as lower bounds, and will show how well we can learn with limited labeled data, while the semi-supervised results will give us an indication of how much improvement can be expected from using the unlabeled data in a particular case. Each supervised version of the algorithms is run on the exact same initial labeled set that is presented to the corresponding self-training algorithm.

As STI and STP both start with the originally imbalanced labeled data, we compare them with the supervised lower bound obtained from training the Naïve Bayes classifiers on that same imbalanced labeled data. We name this classifier Lower Bound from Imbalanced data (**LBI**). STU is compared to the supervised Naïve Bayes classifier trained with the under-sampled labeled data, denoted Lower Bound with Under-sampling (**LBU**). Similarity, STO is compared with the supervised Naïve Bayes classifier trained with the SMOTE over-sampled data, denoted Lower Bound with Over-sampling (**LBO**).

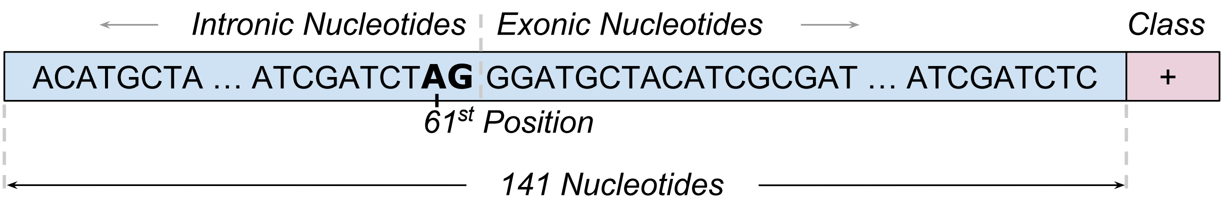
3.3 Experimental Setup

We start this section by describing the data used in our study and the feature representation. As mentioned above, we investigate the behavior of self-training variants in the context of imbalanced data, with application to the binary classification problem of predicting acceptor splice sites in a DNA sequence.

3.3.1 Data and Feature Representation

We use five imbalanced and relatively large datasets of DNA sequences, from five organisms: *C. elegans, C. remanei, P. pacificus, D. melanogaster*, and *A. thaliana*. On average, each data sets contains approximately 160K instances, except for *C. elegans*, which contains roughly 120K instances. Approximately 1% of the instances in each dataset are true acceptor splice sites (our class of interest and the goal of our prediction models), and we denote this category as the positive class. The rest of the instances comprise the decoy sites, or regular *"AG"* dimers that are not acceptor splice sites, and we denote this category as the negative class. All the instances share the same length (141 base pairs) and the *"AG"* dimer is located at the 61st position in the sequence, as illustrated in Figure 3.1. The datasets have been made available by Schweikert et al. [2008], who used them in a domain adaptation context [Schweikert et al., 2008].

Figure 3.1: **Acceptor Splice Site**: Example of a positive instance from the datasets. Each instance is 141 nucleotides long and the "AG" dimer is located at the 61st position in the sequence. The class denotes whether or not the "AG" dimer is a true acceptor splice site (positive) or not (negative).



We use a vectorial representation of the instances, where each sequence is represented as a vector with 141 features corresponding to positions in the sequence, and each feature can take one of the four values $\{A, C, G, T\}$. The value of a feature in a sequence indicates the nucleotide found at that position, corresponding to that feature.

3.3.2 Research Questions

Our experimental design specifically addresses the following research questions:

1. When is semi-supervised learning a good choice on highly imbalanced splice site datasets, and what is the best balancing strategy?

2. How does the algorithms' performance vary with the class distribution?

3. How does the algorithms' performance vary with the labeled-to-unlabeled ratio?

According to the literature [Nigam et al., 2000; Le and Kim, 2014], the performance of semi-supervised classifiers is known to vary with the labeled-to-unlabeled ratio. Moreover, unless the amount of unlabeled data is significantly larger than the amount of labeled data, learning in a semi-supervised framework is not particularly useful, and no major changes are observed between semi-supervised models and their supervised counterparts. To better understand this aspect in the context of imbalanced datasets, we vary the ratio of labeled-to-unlabeled data for all the imbalance degrees. The labeled and unlabeled instances are picked randomly, without replacement, from the original training datasets. As we have a fixed amount of training data for each organism, we refer to the amounts of labeled/unlabeled data in percentages. We vary the labeled data from a very small amount (1% of the training data) to larger amounts (5% and 10%). Consequently, the unlabeled data varies from 99% to 95% and 90%. This allows us to study the variation of the performance with different amounts of labeled data, and also the variation of the performance with the labeled-to-unlabeled ratio, while all available training data is used.

It is also expected for the classification performance to fluctuate with the variation in the class distribution [Estabrooks et al., 2004]. To investigate this behavior more closely, we vary the ratio of positive to negative examples from 1-to-5 to 1-to-99. We obtain different class distributions by randomly discarding the excess instances of the majority class. In order to conduct a fair comparative study, we build the imbalanced datasets incrementally, by starting with the dataset of 1-to-5 imbalance degree, then adding more negative instances to obtain the next dataset with the imbalance degree of 1-to-10, and so forth, until ultimately, the dataset becomes the original set (with imbalance degree of 1-to-99). The 1-to-5 dataset is thus a subset of the 1-to-10 dataset, and both are subsets of the 1-to-15 dataset, etc. The labeled instances are the exact same in all the generated subsets. We want to emphasize that although the actual splice site distribution for our datasets is known (approximately 1% of the *"AG"* occurrences throughout a genome), we vary the class ratios to study the usefulness of semi-supervised approaches with data balancing techniques.

3.3.3 Evaluation Metrics

To measure the predictive ability of our approaches, we compare their performance in terms of the area under the Precision-Recall Curve (auPRC). Since our case study is centered on highly imbalanced datasets, auPRC is a more fitted assessment measure as compared to the area under the Receiver-Operating Curve (auROC) [Davis and Goadrich, 2006]. For our problem, the task is to identify true acceptor splice sites, therefore we report the auPRC values for the minority (positive) class. The auPRC values for the negative class are negligibly similar, differing only in the 3$^{\mathrm{rd}}$ digit for a comparable group of experiments. For each of the five organisms, we use 10-fold cross validation and average the auPRC values across the ten folds. At each round of the cross validation procedure, 10% (or the equivalent of one fold) is set aside for testing, while the remaining 90% (9 remaining folds) of the data is used as training. To simulate semi-supervised conditions, from the training data, we utilize 1%, 5%, and 10%, respectively as labeled data, and the rest, 99%, 95%, and 90% respectively, is treated as unlabeled data, by intentionally ignoring the labels. At last, the labeled data is balanced incrementally to obtain the imbalance ratios used in our study. In our graphs, we report the average over all five organisms of the auPRC values for the positive class (due to the fact that the results were generally consistent).

3.4 Results

We have organized our experiments and their discussion in three main sections, Sections 3.4.1, 3.4.2, 3.4.3. Each of these result sections contains three graphs, one for each ratio of labeled-to-unlabeled data (for varying positive-to-negative ratios). In Section 3.4.1, we present the result from training purely supervised Naïve Bayes classifiers from the original and re-sampled labeled data. In Section 3.4.2, we present the semi-supervised variants. In Section 3.4.3, we compare the best supervised baseline and the top semi-supervised variants in order to understand if the unlabeled data is indeed helping to improve classification. Finally, in Section 3.4.4, we will conclude our results by answering each of the research questions stated in Section 3.3.2.

3.4.1 Supervised Baselines

We have summarized our results of the supervised experiments in Figure 3.2 for 1% labeled data, Figure 3.3 for 5% labeled data, and Figure 3.4 for 10% labeled data. Each graph represents averaged auPRC values over all five organisms, and each group of bars shows the performances of the supervised Naïve Bayes classifier when trained on the original imbalanced labeled data (LBI), on the under-sampled labeled data (LBU), and the over-sampled labeled data (LBO) for one particular imbalance degree.

As can be seen from Figures 3.2, 3.3, and 3.4, the imbalance degree influences the performance of all supervised classifiers. Specifically, the auPRC values decrease with the increase of the class imbalance. For example, in Figure 3.2, the models learned from the

Figure 3.2: **Supervised Learning Results from 1% Labeled Data While Varying the Imbalance Degree**: Averages of the auPRC values for the minority class over all 5 organisms, when learning supervised Naïve Bayes classifiers from **1%** labeled data, while varying the positive-to-negative ratio from 1-to-5 to 1-to-99.

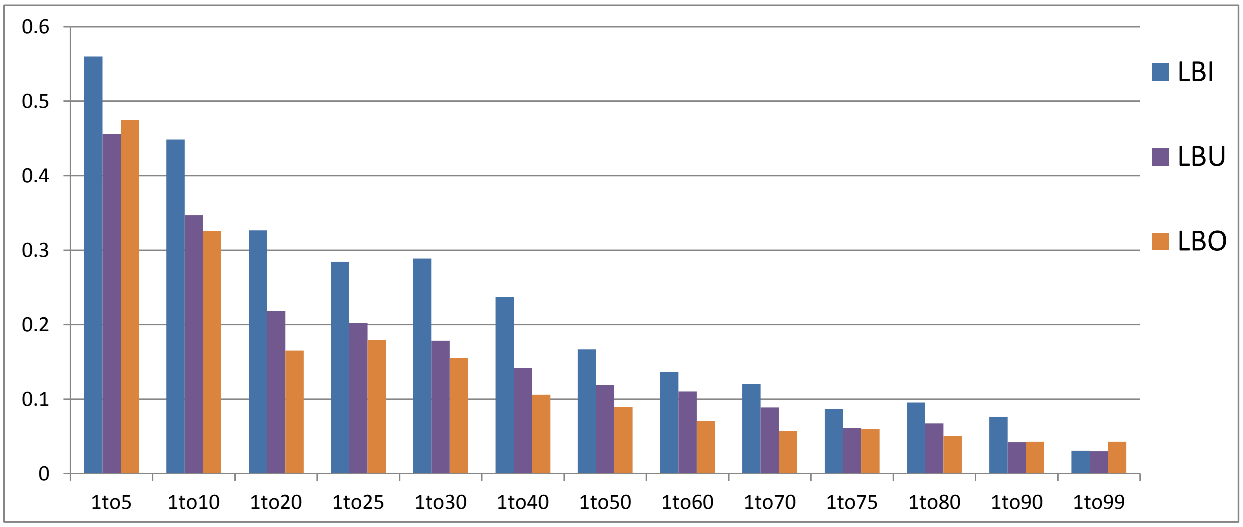
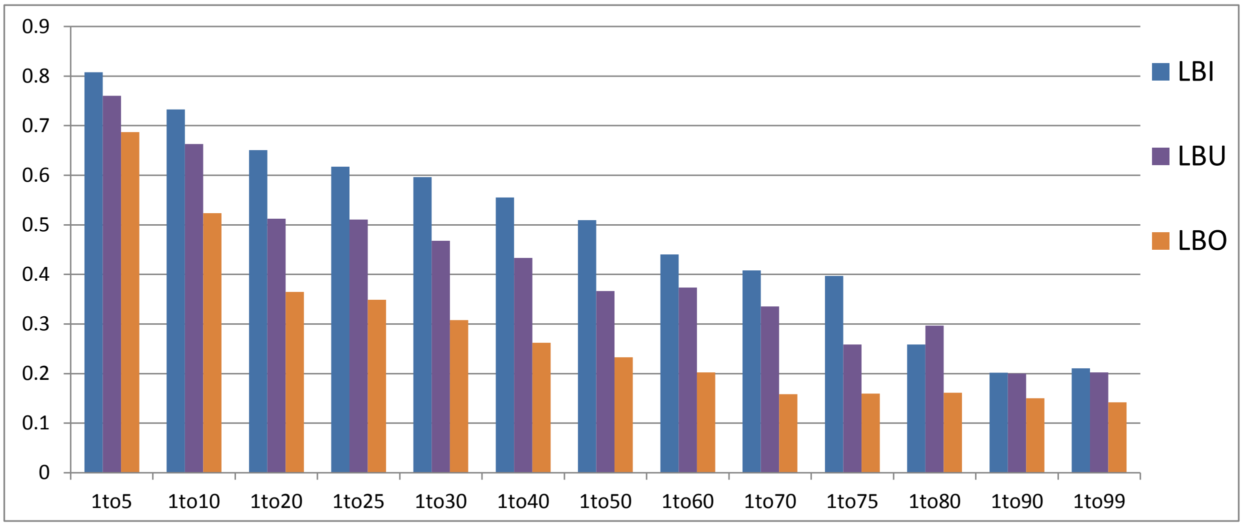


Figure 3.3: **Supervised Learning Results from 1% Labeled Data While Varying the Imbalance Degree**: Averages of the auPRC values for the minority class over all 5 organisms, when learning supervised Naïve Bayes classifiers from **5%** labeled data, while varying the positive-to-negative ratio from 1-to-5 to 1-to-99.



datasets with 1-to-5 imbalance degree reach auPRC values around 0.5, whereas the models obtained from the dataset with 1-to-99 imbalance degree record auPRC values below 0.1. Similar drops in auPRC values can be observed in Figures 3.3 and 3.4. Another foreseen outcome is that the auPRC values increase with the amount of labeled data. The values in Figure 3.2, representing the classifiers trained on very small amounts of labeled data (1%), are lower than the values from Figure 3.3, where the classifiers are trained on more data (5%), which, in turn, are lower than the auPRC values from Figure 3.4, where the classifiers are trained with even more data (10%).

One interesting aspect is that in Figure 3.4, which shows experiments with classifiers trained on 10% labeled data, the differences between performances on different imbalanced distributions are not as big as in Figure 3.2, which shows experiments with classifiers trained on 1% labeled data. This leads to the conclusion that eventually, a sufficient amount of labeled data could allow for the classifiers to converge, and ultimately perform similarly, regardless of the imbalance degree.

Surprisingly, the classifiers trained on the original imbalanced labeled datasets (LBI) are outperforming the other classifiers trained on the re-sampled data. From the re-sampling perspective, under-sampling (LBU) is clearly outperforming over-sampling (LBO). One interesting exception is recorded in the most extreme case of imbalance, 1-to-99, when learning from 1% labeled data, where LBO is surpassing the other classifiers. In such extreme cases, SMOTE over-sampling seems to be a useful technique for generating positive examples.

3.4.2 Semi-supervised Variants

We have summarized our results for the semi-supervised self-training classifiers based on Naïve Bayes in Figure 3.5 for 1% labeled and 99% unlabeled data, Figure 3.6 for 5% labeled and 95% unlabeled data, and Figure 3.7 for 10% labeled and 95% unlabeled data. Again, each graph represents averaged auPRC values over all five organisms, and each group of bars represents all the algorithms' performances for one particular imbalance degree.

Figure 3.4: **Supervised Learning Results from 1% Labeled Data While Varying the Imbalance Degree**: Averages of the auPRC values for the minority class over all 5 organisms, when learning supervised Naïve Bayes classifiers from **10%** labeled data, while varying the positive-to-negative ratio from 1-to-5 to 1-to-99.

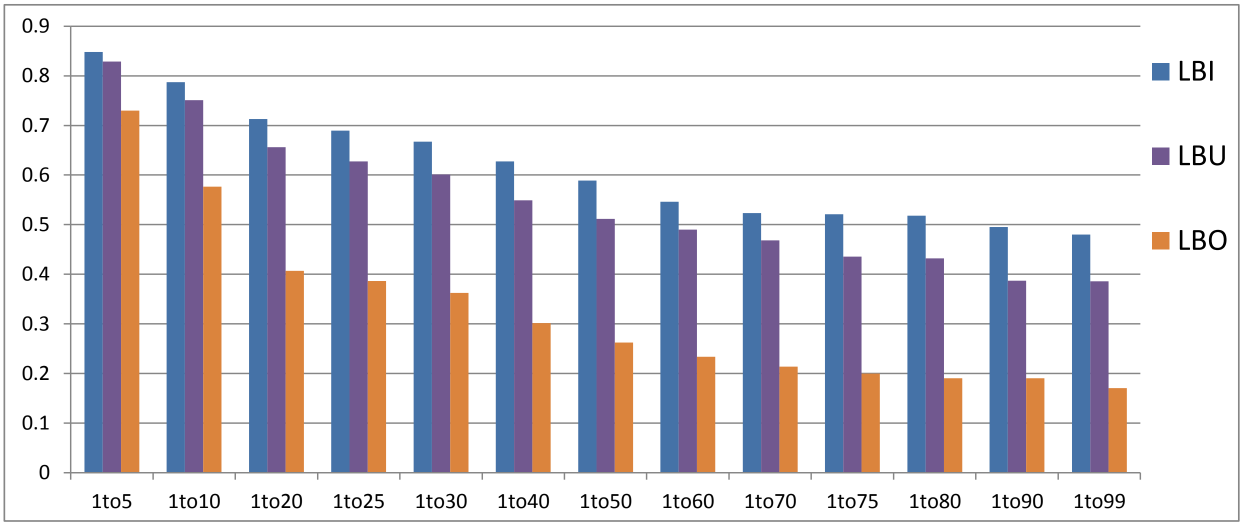
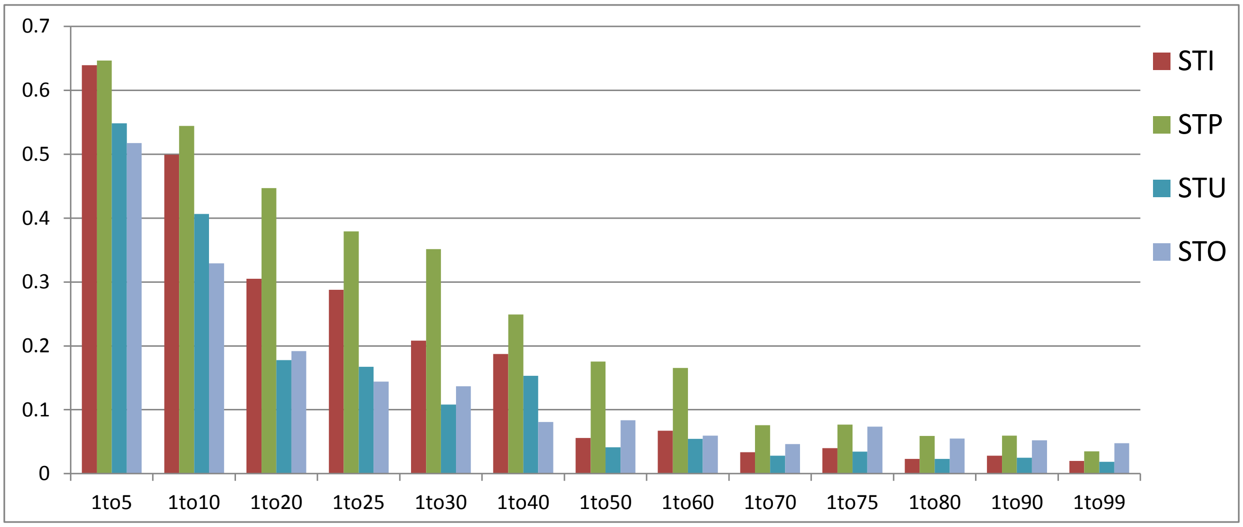
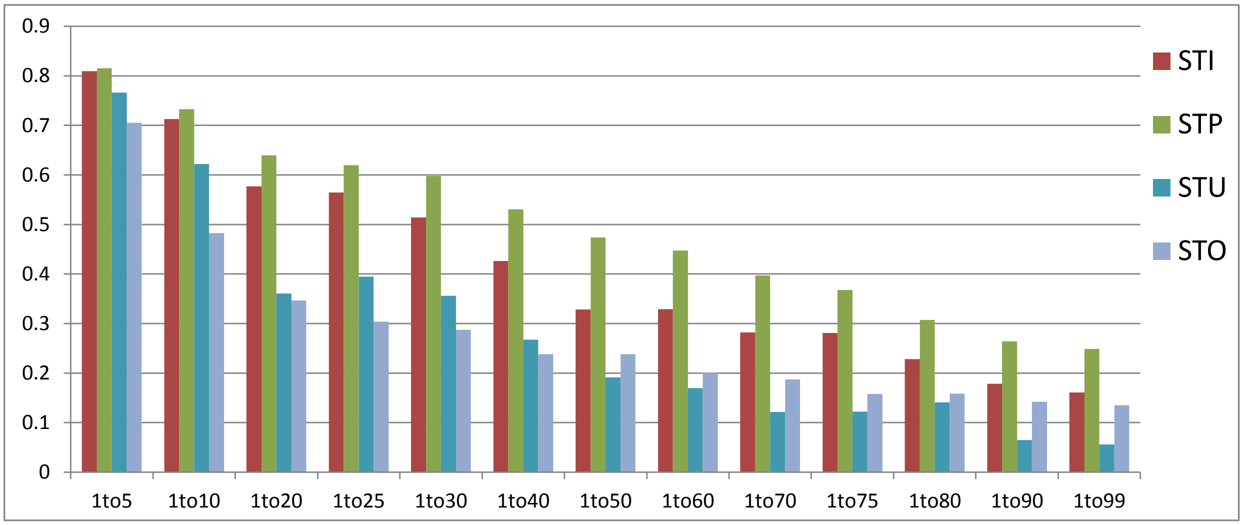


Figure 3.5: **Semi-supervised Learning Results from 10% Labeled Data While Varying the Imbalance Degree**: Averages of the auPRC values for the minority class over all 5 organisms, when learning self-training classifiers based on Naïve Bayes from training data consisting of **1%** labeled **99%** unlabeled, while varying the positive to negative ratio from 1-to-5 to 1-to-99.



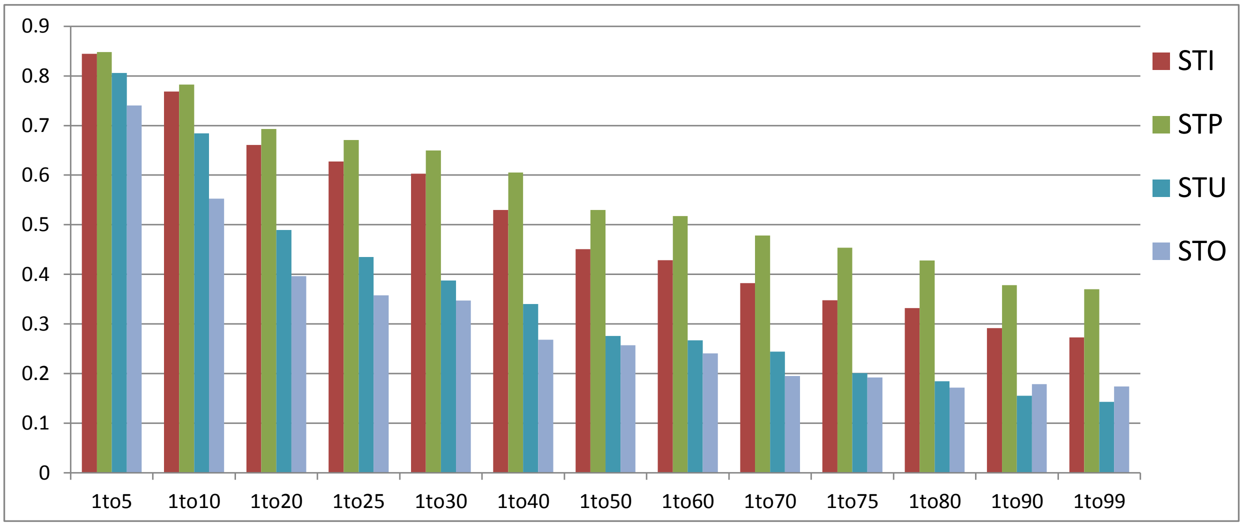
As can be seen from Figures 3.5, 3.6, and 3.7, STP is the best SSL algorithm in almost all imbalanced cases. The only exception was recorded in Figure 3.5, for the case where 1% of the training data is labeled and the imbalance degree is maximum (1-to-99), in which case STO is slightly better. This shows that gradually balancing the labeled data during the semi-supervised iteration (by adding only positive instances to the labeled dataset) is a useful technique for addressing imbalanced distributions. The classical approach, STI, is the second best for degrees of imbalance of up to 1-to-40, whereas for the more imbalanced cases, the over-sampling technique, STO, is approaching STP and surpasses its performance in the 1-to-99 case.

Figure 3.6: **Semi-supervised Learning Results from 5% Labeled Data While Varying the Imbalance Degree**: Averages of the auPRC values for the minority class over all 5 organisms, when learning self-training classifiers based on Naïve Bayes from training data consisting of **5%** labeled **95%** unlabeled, while varying the positive to negative ratio from 1-to-5 to 1-to-99.



In Figure 3.6, when the ratio of labeled to unlabeled data is 5% to 95%, STP is consistently and considerably outperforming all the other algorithms, followed again by STI. For distributions ranging from 1-to-5 to 1-to-50, STU is learning better than STO. Similar trends have been reported for supervised learning by Lusa and Blagus [2010], who found that when the class imbalance is not too severe, under-sampling is working better than over-sampling. For the more extreme cases, STO is a better learner than STU. Similar to the experiments using 1% labeled data (Figure 3.5), the generation of synthetic samples via SMOTE is a good practice to use with self-training approaches for highly skewed datasets.

Figure 3.7: **Semi-supervised Learning Results from 10% Labeled Data While Varying the Imbalance Degree**: Averages of the auPRC values for the minority class over all 5 organisms, when learning self-training classifiers based on Naïve Bayes from training data consisting of **10%** labeled **90%** unlabeled, while varying the positive to negative ratio from 1-to-5 to 1-to-99.



Finally, in Figure 3.7, we can see that when the ratio of labeled-to-unlabeled data is 10% to 90%, the trends are similar to the trends in the graphs where the ratio of labeled-to-unlabeled data is 5% to 95%, mainly the STP approach is consistently and considerably outperforming all the other algorithms, followed yet again by STI. One possible explanation for the fact that STP and STI are leading these charts could be that the initial labeled set they use in self-training is imbalanced, meaning it has more instances than the under-sampled set, and, hence, more information, and also less noise than the over-sampled set, where artificial instances may perturb the base classifier. This observation is also valid for the supervised cases shown in Section 3.4.1, where both under-sampling and over-sampling were surpassed by LBI, the algorithm trained on the original class distribution. However, in the case of semi-supervised learning, dynamically balancing in subsequent iterations of the self-training algorithm (by adding only positive instances, *i.e.*, STP), is better than no balancing (STI). One difference between the trends in Figure 3.6 (5% labeled and 95% unlabeled) and the trends in Figure 3.7 (10% labeled and 90% unlabeled) is that STO is surpassing STU starting with imbalance degrees of 1-to-50 as opposed to 1-to-90. One possible reason is that the more labeled data is added, the less the algorithms benefit from the over-sampling technique, which is most probably introducing noise.

It has been reported that under-sampling is more suitable for semi-supervised learning on imbalanced datasets than over-sampling [Li et al., 2011]. We have observed the same trend when the imbalance degree is relatively low (1-to-5 to 1-to-25) and the amount of labeled data is relatively large (10%), characteristics that mirror the characteristics of the data from [Li et al., 2011]. However, for highly imbalanced datasets and smaller amounts of labeled data (1% and 5%), we have found SMOTE over-sampling to be a better approach.

3.4.3 Supervised versus Semi-supervised Approaches

In this section, for easier visualization, we are presenting the best semi-supervised variants, STP and STI (as revealed by the graphs in Section 3.4.2) in comparison with the best supervised baseline, LBI (as revealed by the graphs in Section 3.4.1).

In Figure 3.8, when the amount of labeled data comprises 1% of the training set, STI outperforms all supervised algorithms in all cases of up to 1-to-60 imbalance degrees. For the more imbalanced cases, from 1-to-70 to 1-to-99, there are no consistent patterns observable, but it seems that supervised learning in general outperforms semi-supervised algorithms. A possible explanation is that the models learned from very limited amounts of highly skewed data are weak, thus mislabeling the originally unlabeled instances and deteriorating the classification.

In Figure 3.9, the labeled data available is now 5%, and STI is obviously useful when the imbalance degrees are more extreme (from 1-to-80 to 1-to-99), whereas for milder degrees of imbalance, STI is comparable with the supervised lower bound LBI. One possible expla-

Figure 3.8: **Semi-supervised vs. Supervised Performance when Learning from 1% Labeled Data While Varying the Imbalance Degree**: Averages of the auPRC values for the minority class over all 5 organisms, when learning classifiers based on Naïve Bayes from training data consisting of **1%** labeled **99%** unlabeled, while varying the positive to negative ratio from 1-to-5 to 1-to-99. This graph shows the best supervised baseline, LBI, and the top two most accurate semi-supervised algorithms, STP and STI.

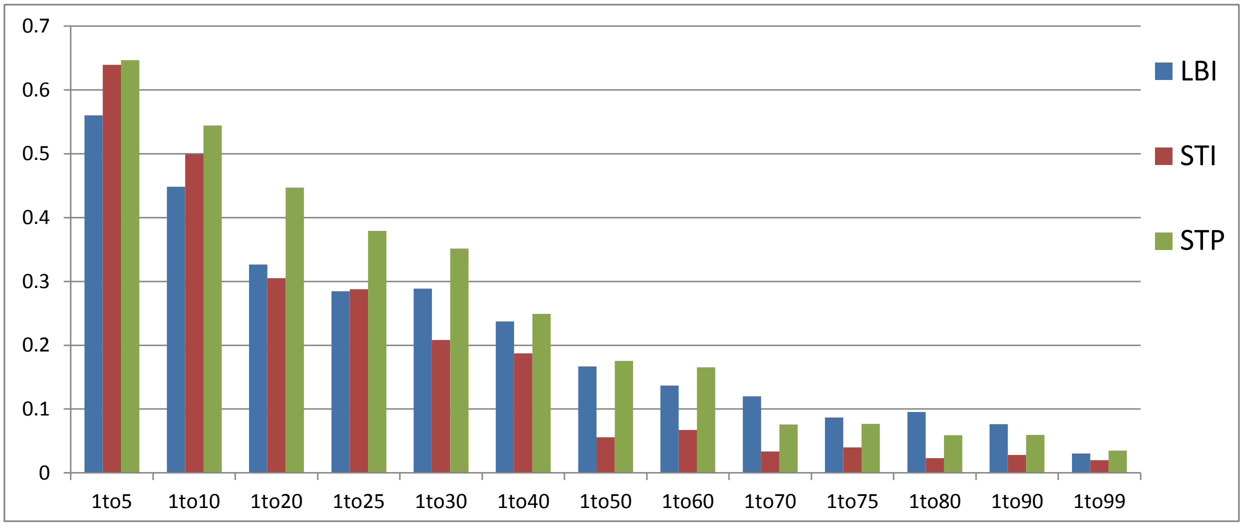
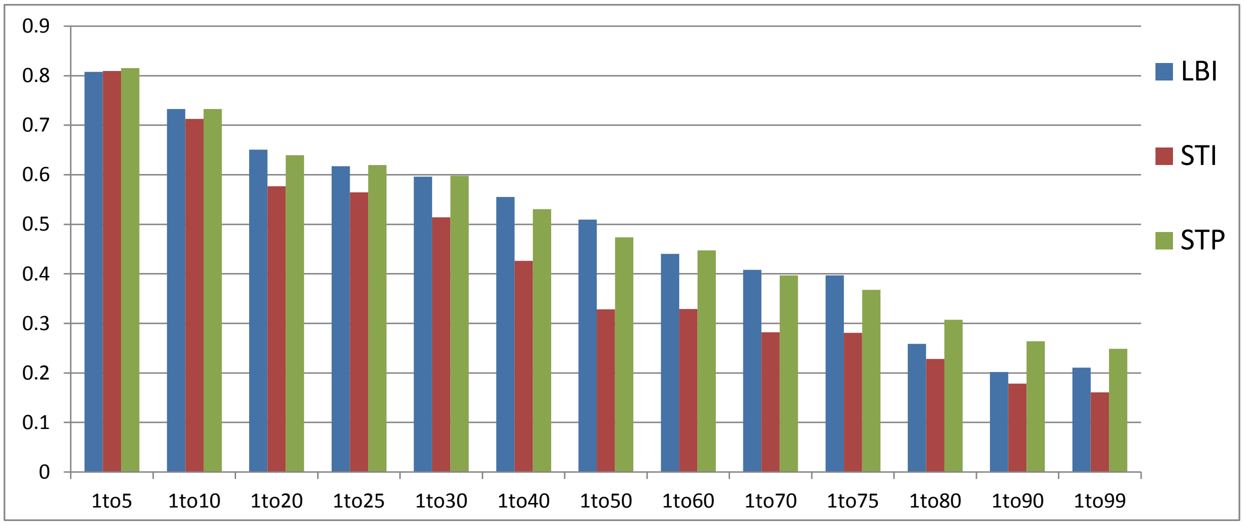


Figure 3.9: **Semi-supervised vs. Supervised Performance when Learning from 5% Labeled Data While Varying the Imbalance Degree**: Averages of the auPRC values for the minority class over all 5 organisms, when learning classifiers based on Naïve Bayes from training data consisting of **5%** labeled **95%** unlabeled, while varying the positive to negative ratio from 1-to-5 to 1-to-99. This graph shows the best supervised baseline, LBI, and the top two most accurate semi-supervised algorithms, STP and STI.



nation is that with more labeled data, the original classifiers learned from labeled data are better, and thus some unlabeled instances are correctly labeled as positive. However, for smaller imbalance degrees, those instances don't contribute significantly to the performance, given that LBI and STI are comparable. As opposed to that, for higher imbalance degrees, the correctly labeled instances can make a difference, and help the semi-supervised STI classifier surpass the supervised LBI classifier. The classical self-training approach, STI, is always falling below the supervised baseline, although the difference between them is small for lower imbalance degrees and higher for higher imbalance degrees. That might mean that some negative instances that are added to the labeled datasets may be mislabeled, especially in the case of the higher imbalance degrees, and thus they are deteriorating the performance as compared to the supervised counterpart.

When the amount of labeled data makes up a tenth of the total training data available for learning, the supervised baseline is outperforming the semi-supervised algorithms, as can be observed from Figure 3.10. One reason could be that 10% labeled data is enough to train a competitive supervised classifier, and semi-supervised learning should be used as a solution for the cases where the labeled data is truly limited. Similar results for the YATSI-based semi-supervised algorithm using Naïve Bayes were reported in [Catal and Diri, 2009] on the problem of software fault-detection.

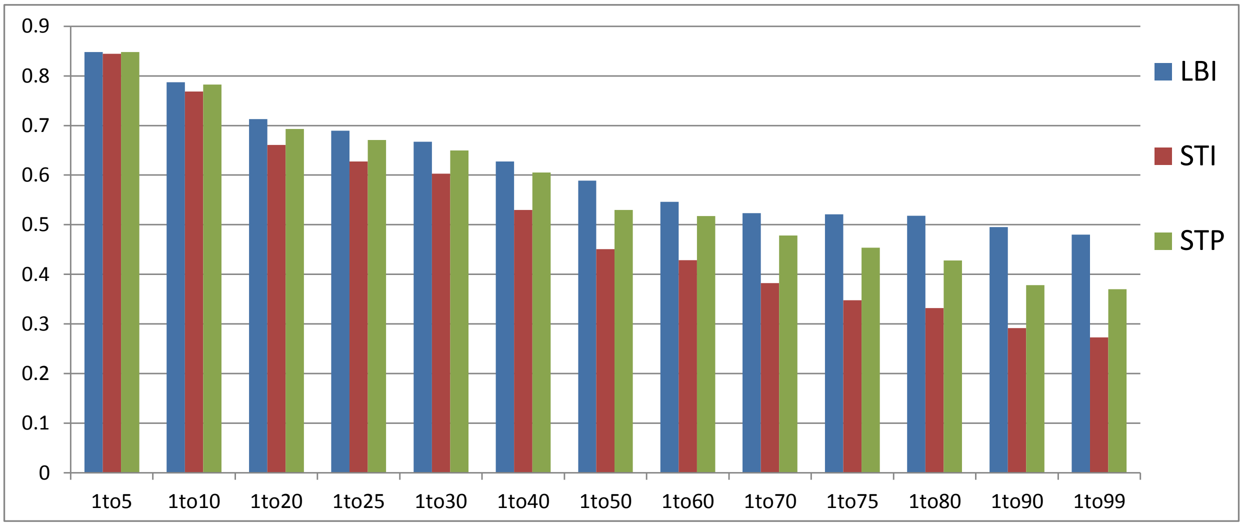
3.4.4 Addressing the Research Questions

In this subsection, we use our experimental results to answer each of the research questions that motivated this work and summarize some general trends suggested by our study.

1. *When is semi-supervised learning a good choice on highly imbalanced splice site datasets, and what is the best strategy (combination of base-classifier and re-sampling technique)?*

Our empirical results suggests that semi-supervised learning, especially self-training with

Figure 3.10: **Semi-supervised vs. Supervised Performance when Learning from 10% Labeled Data While Varying the Imbalance Degree**: Averages of the auPRC values for the minority class over all 5 organisms, when learning classifiers based on Naïve Bayes from training data consisting of **10%** labeled **90%** unlabeled, while varying the positive to negative ratio from 1-to-5 to 1-to-99. This graph shows the best supervised baseline, LBI, and the top two most accurate semi-supervised algorithms, STP and STI.



dynamic balancing with positive data only (STP), is a better choice than supervised learning, when the amount of labeled instances represents a small percentage of the total data available for training (1% in our case). As more labeled data becomes available (and comprises 5% of the training data in our experiments), STI is beneficial for datasets with very high imbalance degrees. However, for larger amounts of labeled data (10% in our case), the classical supervised baseline may become preferable.

2. *How does the algorithms' performance vary with the class distribution?*

As expected, the class distribution ratio influences all the algorithms, supervised and semi-supervised alike. Overall, better values for auPRC are obtained when the class ratio is smaller and the values decrease for the more highly imbalanced cases. Although the datasets increase in size with the increase in the imbalance degree (there are more instances belonging to the majority class), the prediction problem becomes more difficult as the positive class is more and more underrepresented and the learning is biased towards the majority class. The dynamic balancing of STP, followed closely by the classical STI, are more beneficial than any other re-sampling technique (under-sampling and over-sampling). For STP and STI, we run two-tailed paired t-tests, as opposed to one-tailed t-tests, to identify statistically significant differences in either direction, on the semi-supervised algorithms for all the variations of unlabeled data, for each organism. The test determines if the difference between the SSL algorithm and the lower bound is statistically significant or not [Dietterich, 1998]. The STP and STI results were found statistically significant by the paired t-test, with only a few (8) scattered exceptions for STP in some organisms (out of a total of 195 experiments).

3. *How does the algorithms' performance vary with the labeled-to-unlabeled ratio?*

Our results show that the more labeled data is used in training (1% vs 5% vs 10%), the higher the increase in auPRC values, for all classifiers. This is an expected trend, as more labeled data improves the initial models, which can then classify more accurately the unlabeled instances and subsequently identify more appropriate examples to add back to the labeled set for further re-training.

3.5 Related Work

The large volumes of genomic data ask for machine learning and statistical methods to assist the complex process of genome annotation. Supervised machine learning approaches for bioinformatics problems have been widely used [Liu et al., 2012; Chen, 2008; Wang and Wu, 2006; Yu et al., 2013; Erdoğdu et al., 2013; Jiang et al., 2013; Rider et al., 2014; Huang, 2013]. The problem of identifying splice sites using machine learning techniques has also been addressed, mostly by supervised methods [Baten et al., 2006; 2007; Sonnenburg et al., 2007; Castelo and Guigó, 2004; Batuwita and Palade, 2012]. For example, in [Li et al., 2012a], the authors present a state-of-the-art method using SVM and an RBF kernel for human splice site detection. In [Baten et al., 2006], the authors use a combination between a Markov Model of order 1 and SVM with polynomial kernel, for the NN269 dataset, with imbalance degrees of approximately 1-to-4.2 in the case of acceptor sites, and 1-to-3.71 in the case of donor sites. In [Baten et al., 2007], a Markov Model approach is used on a human splice site dataset with imbalance degrees of 1-to-96 for acceptors and 1-to-116 for donors. The goal of our work was not to obtain the best possible results for splice site classification, which has already been successfully addressed by Sonnenburg et al. [2007] using SVM and specialized kernels, but rather to explore semi-supervised learning as a possible solution for splice site prediction, and to study the effects of imbalanced distributions on semi-supervised learning algorithms.

Batuwita and Palade applied SVM [Batuwita and Palade, 2010] with re-sampling methods on four imbalanced biological datasets and the Pageblocks dataset from the UCI, with up to 8K instances and no more than 1-to-50 imbalance degree. They proposed to first identify the most informative negative instances, and then randomly over-sample the positive instances in order to reach the same number of negative instances selected. They suggested that the points located close to the class boundary are the most informative and used the separating hyperplane found by SVM trained on the original imbalanced dataset. Two of their insights regarding over-sampling are: (1) over-sampling could result in better classification than under-sampling, by increasing sensitivity while reducing the specificity at much milder rates compared to under-sampling, and they propose a more efficient over-sampling technique, especially tailored for SVM; (2) over-sampling unavoidably increases the size of the training set which makes SVM perform exponentially slower. This was one of the major reasons why we chose Naïve Bayes as the classification algorithm for our study.

The same authors also proposed a novel measure for evaluating the learning of supervised classifiers on imbalanced bioinformatics datasets, namely the "adjusted geometric-mean" [Batuwita and Palade, 2012]. In this work, the authors conducted experiments on ten DNA (including splice sites) and protein datasets, with imbalance degrees of up to 1-to-14 and dataset sizes with up to 10K instances. In our study, we use much larger datasets (160K) with imbalance degrees of 1-to-99. Large-margin based classifiers, such as SVM, would be impractical, due to their large number of parameters that need tuning and longer computational times as compared to Naïve Bayes. Similar to Batuwita and Palade [2012], we also used under- and SMOTE over-sampling.

Wei and Dunbrack Jr [2013] explored the effects that balancing both the training and test datasets have on the SVM algorithm. The authors studied the problem of classifying human missense mutations as deleterious or neutral; they systematically varied the proportion of deleterious to neutral mutations in the training set, to conclude that balancing the training dataset is producing more accurate SVM classifiers in terms of several accuracy measures, while the class unevenness in the test data is irrelevant. Their study is particularly useful for problems where the prior distribution of the test data is unknown.

Similar results have been reported by Lusa and Blagus [2010], who found that balancing the class prior in the training set is a good choice, especially if the instances are represented in a high-dimensional space. Their focus is on high-dimensionality datasets and study the behavior of various algorithms on binary classification problems on a simulated set and a genuine bioinformatics set from a breast cancer gene expression microarray study, publicly available. They explore under- and over-sampling, as well as the "multiple down-sizing" technique, which is basically an ensemble of sub-classifiers trained on balanced subsets, and the final prediction is obtained via majority voting of the sub-classifiers. We considered this approach to be more algorithmically-oriented and plan to devote a separate study concerned with building ensemble of classifiers to deal with imbalanced data in the semi-supervised framework whereas in this work, we mainly focus on straightforward data specific re-balancing techniques.

Semi-supervised learning has mostly been studied on protein classification [Weston et al., 2005; 2006; Kall et al., 2007; Craig and Liao, 2007; Xu et al., 2009] and efforts on semi-supervised learning from imbalanced distributions have focused on protein datasets with relatively low imbalance degrees. For example, Kondratovich et al. [2013] address the problem of molecule activity prediction and they experiment with ten relatively small (3,000 instances) molecule activity datasets with imbalance degrees no larger than to 1-to-40. They use Transductive Support Vector Machines (TSVM), which is a subtype of semi-supervised learning. The classical TSVM algorithm, without any re-sampling, is performing quite well, and is found to be somewhat insensitive to the imbalance degree. Our results are similar for self-training with imbalanced data (STI), the classical self-training approach, which also does not specifically address the imbalance distribution. In our results, STI was the second best variant that we experimented with, after STP. It would be interesting to apply TSVM on the splice site DNA datasets and observe how the performance changes with higher imbalance degrees.

Semi-supervised learning from imbalanced datasets has been explored in other domains. Catal and Diri [2009] proposed the immune-based YATSI (Yet Another Two Stage Idea) [Driessens et al., 2006] algorithm to predict faulty software, which is a semi-supervised meta-algorithm that can be wrapped around any supervised (base) classifier. In YATSI, at each iteration, the decision of which instances to add to the labeled set is based on an ensemble of predictions calculated from the k-nearest neighbors, in terms of Euclidean distance. The weights of an unlabeled instance's neighbors are summed per class and the class with the largest weight is assigned as the label of that instance. Their experimental setup was based on four datasets where the class of interest made up from 7% to 21% of the data, and they varied the amount of labeled data from 5% to 10% and 20%.

In [Drummond et al., 2003], the authors investigated the C4.5 algorithm's compatibility with under- and over-sampling in terms of cost curves on four UCI datasets. They found that under-sampling is more sensitive to class distributions than over-sampling, referencing other studies that uncovered the same patterns with respect to decision trees. In this study, we also found that under-sampling is outperforming over-sampling, in particular the SMOTE technique, in terms of auPRC values.

Chen [2008] studied imbalanced datasets in the supervised and semi-supervised frameworks for the 2008 U.C. San Diego Data Mining competition, using re-sampling techniques like SMOTE, over-sampling (by duplicating each minority instance an equal number of times) and random under-sampling in combination with Decision Trees, Naïve Bayes, and Neural Networks. For the supervised task, the dataset contains 40K instances and has a 1-to-10 imbalance ratio. For Naïve Bayes, neither of the re-sampling techniques produced significantly different results. For Random Forests, under-sampling and SMOTE improved the results by 8%, while over-sampling by duplication gave similar results to the classifier built on the imbalanced set. For Neural Networks, all three techniques significantly improved the accuracy. For the semi-supervised task, the dataset consists of roughly 68.5K instances, and only 60 of them are labeled as positive, while the rest are unlabeled; the test set consists of approximately 11.4K instances. In such a context, when solely positive examples are labeled, the problem is known as PU-learning, *i.e.*, learning from only Positive and Unlabeled data. The strategy from [Chen, 2008] was to first identify negative examples, utilizing a technique, called Spy technique, and then train a Naïve Bayes supervised classifier, which, as expected, produced better results than treating all the unlabeled data as negative examples.

3.6 Conclusions and Future Work

In this study, we have performed an analysis of self-training classifiers using Naïve Bayes, on five large and highly imbalanced DNA datasets, and have utilized balancing techniques to address the uneven class distributions. Empirical evidence shows that when the labeled data represents a very small percentage of the total number of training instances (in our case, 1%), while the remaining instances are unlabeled, semi-supervised learning algorithms are a better choice than purely supervised classification algorithms. Our results also reveal that for the given problem of acceptor splice site detection, if more than 10% of the total training instances is labeled, the user will benefit more from training supervised algorithms. As our results suggest, the use of semi-supervised learning under the difficult conditions of skewed class priors and very limited amounts of labeled data, this study could potentially open doors for more extensive research targeting DNA classification in semi-supervised scenarios, which fit well with the current data availability constraints in bioinformatics.

In future studies, we hope that experimenting with different DNA datasets will reveal additional insights into the DNA semi-supervised classification problem. Utilizing other algorithms, such as co- and multi-training, which make use of multiple independent views of the data, could potentially increase the classification ability.

Chapter 4

An Empirical Study of Ensemble-based Semi-supervised Learning Approaches for Imbalanced Splice Site Datasets

4.1 Background

Advances in biochemical technologies over the past decades have given rise to Next Generation Sequencing platforms that quickly produce genomic data at much lower costs than ever before. Such overwhelmingly large volumes of sequenced DNA remain difficult to annotate. As a result, numerous computational methods for genome annotation have emerged, including machine learning and statistical analysis approaches that practically and efficiently analyze and interpret data. Supervised machine learning algorithms typically perform well when large amounts of labeled data are available. In bioinformatics and many other data-rich disciplines, the process of labeling instances is costly; however, unlabeled instances are inexpensive and readily available. For a scenario in which the amount of labeled data is relatively small and the amount of unlabeled data is substantially larger, semi-supervised learning represents a cost-effective alternative to manual labeling.

Because semi-supervised learning algorithms use both labeled and unlabeled instances in the training process, they can produce classifiers that achieve better performance than completely supervised learning algorithms that have only a small amount of labeled data available for training [Wang et al., 2003; Kasabov and Pang, 2003; Stanescu et al., 2015]. The principle behind semi-supervised learning is that intrinsic knowledge within unlabeled data can be leveraged in order to strengthen the prediction capability of a supervised model that only uses labeled instances, thereby providing a potential advantage for semi-supervised learning. Model parameters learned by a supervised classifier from a small amount of labeled data may be steered towards a more realistic distribution (which more closely resembles the distribution of the test data) by the unlabeled data.

Unfortunately, unlabeled data can also drive the model parameters away from the true distribution if misclassification errors reinforce themselves. Thus, in practice, semi-supervised learning does not always work as intended [Chawla and Karakoulas, 2005; Nigam and Rayid, 2000; Zhou and Li, 2010]. Moreover, under incorrect assumptions, *e.g.,* regarding the relationship between marginal and conditional distributions of data, semi-supervised learning models risk to perform worse than their supervised counterparts. Given that for many prediction problems the assumptions made by learning algorithms cannot be easily verified without considerable domain knowledge [Ben-David et al., 2008] or data exploration, semi-supervised learning is not always "safe" to use. Advantageous utilization of the unlabeled data is problem-dependent, and more research is needed to identify algorithms that can be used to increase the effectiveness of semi-supervised learning [Li and Zhou, 2015; Le and Kim, 2014], in general, and for bioinformatics problems, in particular. At a high level, we aim to identify semi-supervised algorithms that can be used to learn effective classifiers for genome annotation tasks.

In this context, a specific challenge that we address is the “data imbalance” problem, which is prevalent in many domains, including bioinformatics. The data imbalance phenomenon arises when one of the classes to be predicted is underrepresented in the data because instances belonging to that class are rare (noteworthy cases) or hard to obtain. Ironically, minority classes are typically the most important to learn, because they may be associated with special cases. In general, anomaly or novelty detection problems exhibit highly imbalanced distributions. Specific applications outside the bioinformatics area include credit card fraud, cyber intrusions, medical diagnosis, face recognition, defect detection in error-prone software modules, *etc.* As established in the literature (*e.g.*, [Chawla et al., 2004]), the existence of a major unevenness between the prior class probabilities leads to impartial learning. As a result, classifiers that produce good classification results under normal circumstances (*i.e.*, in the presence of balanced or mildly imbalanced distributions) can be seriously compromised when faced with skewed distributions, as classifiers become strongly biased towards the majority class. In bioinformatics, problems such as promoter recognition, splice site detection, and protein classification are especially difficult because these problems naturally exhibit highly imbalanced distributions.

Re-sampling datasets in order to reach balanced distributions is a common practice that sometimes improves classification performance, as the model encounters an equal number of instances from each class, thereby producing a more appropriate discriminative function as opposed to a function obtained from skewed distributions. However, it is not well understood what is the most appropriate balancing method. Context-dependent conclusions are usually driven by empirical observations concerning both the classifier used and the imbalance degree. The most straightforward method is under-sampling, in which instances that belong to the majority class are eliminated until a balanced distribution is reached. As a consequence, information is lost, which is obviously not desirable, given the value of labeled instances, yet this is a good way to speed up the computation. Moreover, studies have shown the effectiveness of under-sampling [Li et al., 2011] despite its obvious limitations. Over-sampling is another popular re-sampling method in which instances of the minority class are generated artificially to counterbalance majority instances. These synthetic instances can potentially improve the classifier, as it gains access to more labeled data. The trade-off between longer computation times associated with larger datasets and better classification performance is usually worthwhile. However, with oversampling, classifiers are prone to overfitting, due to duplicate instances.

An algorithmic approach to handle imbalanced data distributions is based on ensembles of classifiers. Limited amounts of labeled data naturally lead to "weaker" classifiers, but ensembles of "weak" classifiers tend to surpass the performance of any single constituent classifier. Moreover, ensembles typically improve the prediction accuracy obtained from a single classifier by a factor that validates the effort and cost associated with learning multiple models. Intuitively, "bagging" several classifiers leads to better overfitting control, since averaging the high variability of individual classifiers also averages the classifiers' overfitting. The first effective model ensemble surfaced in the mid 1990s [Breiman, 1996], under the name "bootstrap aggregating" (bagging), which is a meta-algorithm that performs model averaging over models trained on multiple subsets, *i.e.*, bootstrap replicates of the training set. The predictions of the models are combined by voting (in the case of classification) or averaging (in the case of regression) in order to output a single final verdict that reflects the ensemble decision. Originally applied to decision trees, bagging can be used with any classification or regression model and it is especially effective in conjunction with utilization of unstable nonlinear models (*i.e.*, a small change in the training set can cause a significant change in the model's learned parameters). Ensembles of classifiers that utilize bagging, boosting, and hybrid-approaches for imbalanced datasets in the supervised framework were reviewed by Galar et al. [2012].

For a comprehensive survey of data re-sampling and algorithmic approaches to the imbalanced data problem in the supervised learning framework, the reader is referred to [He and Garcia, 2009]. As opposed to supervised learning, fewer efforts have been aimed at the data imbalance problem in the semi-supervised learning framework, with some notable exceptions. In particular, in a previous study [Stanescu and Caragea, 2014a], we experimented with data re-sampling and algorithmic solutions and observed that dynamically balancing the classifiers during the semi-supervised iterations of the algorithm is a useful solution that works better than under- and SMOTE (Synthetic Minority Over-sampling Technique) over-sampling for splice site prediction in the context of single semi-supervised classifiers. We also found that ensembles usually tend to perform better than re-sampling techniques, except for extreme cases when the imbalance degree is 1-to-99, in which case oversampling performs slightly better than the ensemble-based approach. In a subsequent study [Stanescu and Caragea, 2014b], we empirically evaluated ensembles of self-training semi-supervised classifiers and found that maintaining diversity during the process of semi-supervised learning is an important requirement for the ensemble. In the current study, we experiment with both self-training and co-training, utilizing a different feature representation than the one we used in [Stanescu and Caragea, 2014b], to accommodate co-training, which requires two views (representations) of the data.

Similar to our prior work, the current study is performed on the problem of predicting splice sites, a challenging, but important task in genome annotation [Lomsadze et al., 2014]. Splice sites are located at the boundaries between exons and introns. At the 3' end of an intron, the "AG" dimer denotes an acceptor splice site; at the 5' end of the intron, the "GT" dimer denotes a donor splice site. Other non-consensus splice sites exist, but they are not considered in this work. We formulate the task of predicting acceptor splice sites as a binary classification problem in which the positive class represents true acceptor splice sites and the negative class is comprised by decoy "AG" sites. We use five relatively large datasets from five organisms. The distribution of the data (ratio of the size of the minority class to majority class) is very skewed - approximately 1% of "AG" dimers are actually acceptor splice sites.

Among others, Sonnenburg et al. [2007] previously addressed the splice site prediction problem, in the supervised framework, using Support Vector Machines (SVM) and specialized kernels. As opposed to prior work, in this work, our goal is to investigate ensemble-based semi-supervised learning as a potential solution for splice site prediction and to study the effects of imbalanced distributions on semi-supervised algorithms when labeled data is sparse. Given the large datasets of our case study and the numerous models that needed to be trained to simulate different imbalanced degrees for different ensemble variants, we chose Naïve Bayes as the base classifier in co-training and self-training, because of its computation speed and to avoid tuning hyper-parameters (that many other classifiers require in order to perform well). Although, theoretically, the *i.i.d.* assumption (that the observed features are identically and independently distributed) does not hold for many problems (including for the problem studied in this work) generative models such as Naïve Bayes can show superior performance to discriminative models such as SVM, especially when small amounts of labeled data are available [Druck et al., 2007; Stanescu et al., 2015].

The rest of this paper is organized as follows. We continue with a review of related work in the next section, where we also contrast our study with other similar studies. In **Methods**, we describe our approaches, namely the semi-supervised learning ensembles based on self-training and co-training. Section **Data** is dedicated to describing the datasets and the feature representation used with our classifiers. The experimental setting is described in **Experimental setup**, starting with the research questions that motivated the study and continuing with details of the evaluation procedure. We discuss the performance of our approaches in **Results**, and finally, in Section **Conclusion**, we conclude the study and suggest directions for future work.

4.2 Related work

Genome annotation is an ample task that requires machine learning and statistical methods to assist experimental approaches, especially given the large amount of genomic data being generated at unprecedented rates. Supervised machine learning approaches have been widely used in bioinformatics for many tasks, including splice site prediction [Sonnenburg et al., 2007; Baten et al., 2006; 2007; Castelo and Guigó, 2004; Batuwita and Palade, 2012]. For example, human splice site detection was explored in [Li et al., 2012a] using SVM classifiers with a Gaussian kernel, and in [Baten et al., 2006] using a combination of Markov Models and SVM classifiers with polynomial kernels. The work in [Baten et al., 2007] proposed a Markov Model approach for splice site detection in a human dataset with imbalance degrees of 1-to-96 for acceptors and 1-to-116 for donors.

Semi-supervised learning has generally been used in bioinformatics to solve protein classification problems [Weston et al., 2005; 2006; Kall et al., 2007; Craig and Liao, 2007; Xu et al., 2009; Wu et al., 2015], with a few notable exceptions focused on DNA classification [Kasabov and Pang, 2003; Stanescu et al., 2015]. A small number of studies [Kondratovich et al., 2013; Weston et al., 2005; Kundu et al., 2013] have explored the data imbalance problem in the semi-supervised context and proposed effective solutions, but the imbalance degrees were moderate. For example, in [Kondratovich et al., 2013], the authors addressed the problem of molecule activity prediction and experimented with transductive SVM classifiers on datasets with relatively small sizes (3K instances), exhibiting imbalance degrees no higher than 1-to-40.

As opposed to that, we focus on datasets with higher degrees of imbalance (up to 1-to-99) and study the behavior of semi-supervised learning algorithms when the available labeled data is less than 1% of the total amount of training data. In general, such a small amount of labeled data is expected to lead to weak classifiers, but an ensemble of classifiers could help overcome this shortcoming to some extent. Galar et al. [2012] showed that, in supervised frameworks, ensembles perform better than single learners trained on re-sampled data. citeLusa:2010 found that balancing the class prior in the training set via ”multiple down-sizing”, in other words, training an ensemble of subclassifiers on balanced subsets, is particularly useful for high-dimensional representations. They showed this using a simulated set and a genuine, publicly available dataset from a breast cancer gene expression microarray study. Another study by Li et al. [2011] also concluded that an ensemble of co-training classifiers is suitable for imbalanced datasets.

Our objective in this study was to adapt existing semi-supervised learning ensembles to datasets with high degrees of imbalance. Towards this goal, we used the approach from [Li et al., 2011] as inspiration for two of the methods presented in this work. In [Li et al., 2011], the authors proposed that, as the co-training sub-classifiers iterate, the balanced labeled subsets are augmented with the same instances, specifically, the most confidently labeled positive instances and the most confidently labeled negative instances. In our previous work on the problem of splice site prediction [Stanescu and Caragea, 2014b], we found that adding different instances to each self-training subsets leads to improved prediction because diversity is maintained. However, it was not clear what is the best way to manipulate the original distribution to ensure the largest diversity among ensemble members. Motivated by the results of our dynamic balancing technique, where only positive instances are added to the training set during the self-training iterations [Stanescu and Caragea, 2014a], and also by our preliminary results on ensemble approaches based on self-training classifiers [Stanescu and Caragea, 2014b], in the current study, we further analyze various combinations of ensembles and dynamic balancing, with focus on how the augmentation of labeled data should be managed during the semi-supervised iterations. We also experiment with co-training, in addition to self-training, and investigate how ensembles of self-training and co-training Naïve Bayes classifiers behave in the semi-supervised framework when dealing with various imbalance ratios.

A study from Wei and Dunbrack [Wei and Dunbrack Jr, 2013] that explored the effects of various distributions on supervised learning was centered around classification of human missense mutations as deleterious or neutral. By systematically varying the ratio of deleterious to neutral mutations in the training set, the authors concluded that balancing the training dataset improves the performance of SVM as evaluated by several accuracy measures, even when the distribution of the data is just mildly imbalanced. The study in [Wei and Dunbrack Jr, 2013] was performed under the assumption that the real distribution of deleterious versus neutral mutations is unknown. In the datasets used in our work [Schweikert et al., 2008], the proportion of true splice sites was assumed to be approximately 1% of the total number of occurrences of the “AG” dimer throughout the genome, and thus this was the highest imbalance degree that we experimented with (*i.e.*, 1-to-99). However, we varied the ratio of splice site to non-splice site “AG”s from 1-to-5 to 1-to-99, to perform a systematic study of the performance obtained using ensemble-based semi-supervised approaches as a function of the imbalance ratio.

4.3 Methods

This section describes the algorithms studied. As we focus on ensemble-based semi-supervised learning from imbalanced class distributions, specifically ensembles of self-training and co-training classifiers, we will first provide background on self-training and co-training, and also on ensemble learning. Then, we will describe the supervised ensemble approach used as a baseline in our evaluation, and finally, our proposed self-training and co-training ensemble variants.

4.3.1 Self-training

Self-training, also known as self-teaching or bootstrapping, is an iterative meta-algorithm, that can be wrapped around any base classifier. Yarowsky [Yarowsky, 1995] originally introduced self-training and applied it to a natural language processing problem, namely word-sense disambiguation. The first step in self-training is to build a classifier using the labeled data. Then, the labeled dataset is augmented with the most confidently predicted instances from the unlabeled pool, and the model is rebuilt. The process is repeated until a criterion is met, *e.g.*, until the unlabeled dataset has been fully classified or a fixed number of iterations has been reached. In our work, we classify a sub-sample of unlabeled data at each iteration (as opposed to all unlabeled data) in order to increase computation speed. The most confidently classified instances are assigned the predicted class and used to retrain the model. The remaining instances, classified with less confidence, are discarded. The algorithm iterates until the unlabeled dataset has been exhaustively sampled.

4.3.2 Co-training

citeBlum:1998 introduced co-training, also an iterative meta-algorithm, to solve the problem of identifying course pages among other academic web pages. Similar to self-training, co-training is applicable to any base classifier. Unlike self-training, which is a single view algorithm, co-training requires two independent and sufficient views (a.k.a., feature representations) of the same data in order to learn two classifiers. At each iteration, both classifiers label the unlabeled instances and the labeled training data of one classifier is augmented with the most confidently labeled instances predicted by the other classifier. Similar to self-training, in our work we classify only a sub-sample of unlabeled data at each iteration. Instances from the sub-sample classified with small confidence are discarded. The algorithm iterates until the unlabeled dataset has been exhaustively sampled.

4.3.3 Ensembles

Ensemble learning exploits the idea that combinations of weak learners can lead to better performance. Moreover, it is known that diversity among subclassifiers is an important constraint for the success of ensemble learning [38, 39]. However, learning Naïve Bayes classifiers from bootstrap replicates will not always lead to sufficiently "diverse" models, especially for problems with highly imbalanced distributions. In order to ensure sufficient variance between the original training data subsets of our highly imbalanced datasets, we used a technique initially recommended by Liu et al. [2009], who proposed training each subclassifier of the ensemble on a balanced subset of the data, providing subclassifiers with the opportunity to learn each class equally, while the ensemble continues to reflect the original class distribution. An implementation of this technique by Li et al. [2011] proved to be successful for the problem of sentiment classification, and was used as inspiration in our work.

4.3.4 Supervised Lower Bound

Generally, supervised models trained only on the available labeled data are used as baselines for semi-supervised algorithms. Thus, the hypothesis that unlabeled data helps is verified against supervised models that entirely ignore unlabeled instances. Because our focus is on ensemble methods and ensembles of classifiers typically outperform single classifiers, the lower bound for our approaches is an ensemble of supervised classifiers. Specifically, we train ensembles of Naïve Bayes classifiers using re-sampled balanced subsets and use their averaged predictions to classify the test instances. This approach is referred to as the Lower Bound Ensemble (LBE).

4.3.5 Ensembles inspired by the original approach: CTEO and STEO

In [Li et al., 2011], co-training classifiers were augmented with the topmost confidently labeled positive and negative instances, as found by classifiers trained on balanced labeled subsets. The authors set the number of iterations at 50, and classified all unlabeled instances at each iteration. Moreover, the two views of the co-training classifiers were created at each iteration, using ”dynamic subspace generation” (random feature splitting into two views), in order to ensure diverse subclassifiers.

However, this exact approach did not produce satisfactory results in our case, so we modified the algorithm from [Li et al., 2011] in order to better accommodate our problem. We named the resulting approach Co-Training Ensemble inspired by the Original approach

(CTEO). We also experimented with a variant where co-training was replaced with self-training, and named this variant Self-Training Ensemble inspired by the Original approach (STEO). The pseudocode for both CTEO and STEO variants is illustrated in Algorithm 2. As can be seen, Steps 7-9 are described for co-training (first line) and self-training (second line, in italic font), separately.

The first modification we made to the original ensemble-based approach, for both self-training and co-training variants, is that we kept the features fixed, *i.e.*, used "static" instead of "dynamic subspace generation." For co-training, we used a nucleotide/position representation as one view, and a 3-nucleotide/position representation as the second view, under the assumption that each view is sufficient to make accurate predictions, and the views are (possibly) independent given the class.

The second modification we made is that we did not classify all unlabeled instances at each iteration; instead, we classified only a fixed subsample of the unlabeled data, as proposed in the classical co-training algorithm [Blum and Mitchell, 1998]. This alteration speeds up the computation process. The last modification that we made is that once a subsample was labeled and the top most confidently labeled instances were selected to augment the originally labeled dataset, we simply discard the rest of the subsample, thereby differing from the classical co-training approach [Blum and Mitchell, 1998] and from the original co-training ensemble approach [Li et al., 2011]. This change also leads to faster computation times and, based on our experimentation, reduces the risk of adding mistakenly labeled instances to the labeled set in subsequent iterations. Furthermore, the last two adjustments lead to a fixed number of semi-supervised iterations, *i.e.*, as the algorithm ends when the unlabeled data pool is exhausted. We use a subsample size that is dependent on the dataset size, and selected such that the algorithm iterates approximately the same number of times (50) for each set of experiments, for a certain imbalance degree. After the iterations terminate, the ensemble is used to classify the test set by averaging the predictions of the constituent subclassifiers.

An important observation regarding Step 9 in Algorithm 1 is that, in the case of co-training, when the two classifiers based on $view_1$ and $view_2$, respectively, make their predictions, an instance is added to the pseudolabeled set P only if (1) no conflict exists between the classifiers, *i.e.,* both classifiers agree on the label, and (2) one classifier predicts the label with high confidence, while the other predicts the same label with low confidence. These conditions ensure that the two views inform each other of their best predictions, thereby enhancing each other's learning.

Algorithm 2 Ensembles inspired by the original approach [Li et al., 2011] - CTEO/STEO

```
Given: a training set comprised of labeled and unlabeled data $D = (D_l, D_u)$, $|D_l| \ll |D_u|$
Create $U$ by picking $S$ random instances from $D_u$ and update $D_u = D_u$ - $U$, $S$ = sample size
Generate $N$ balanced subsets from $D_l : D_{l1}, \ldots, D_{ln}$
repeat
    Initialize $P = \emptyset$
    for $i = 1$ to $N$ do
        CT: Train subclassifiers $C_{i1}$ on $view_1$ and $C_{i2}$ on $view_2$ of balanced subset $D_{li}$
        *ST: Train subclassifier $C_i$ on combined views of balanced subset $D_{li}$*
        CT: Classify instances in $U$ using the classifiers $C_{i1}$ and $C_{i2}$
        *ST: Classify instances in $U$ using subclassifier $C_i$*
        CT: Use $C_{i1}$ and $C_{i2}$ to select 2 positive and 2 negative instances and add them to $P$
        *ST: Use $C_i$ to select 2 positive and 2 negative instances, and add them to $P$*
    end for
    Augment each balanced subset with the instances from $P$
    Discard remaining unused instances from $U$
    Create a new unlabeled sample $U$ and update $D_u = D_u$ - $U$
until $U$ is empty (*i.e.*, the unlabeled data is exhausted)
```

As mentioned above, STEO differs from the co-training based ensemble, CTEO, at Steps 7-9 in Algorithm 1: instead of using two subclassifiers trained on two different views, only one classifier is built using all features ($view_1$ and $view_2$ combined), and then this classifier is used to select the best two positive predictions and the best two negative predictions. Because each subclassifier in CTEO contributes one positive and one negative instance, after one iteration, the set P of pseudo-labeled instances contains 2N positive instances and 2N negative instances. Therefore, in STEO, we add the top two positives and top two negatives as predicted by the same subclassifier C_i in order to maintain an augmentation rate identical to the augmentation rate in CTEO. After the semi-supervised iterations terminate, the ensemble is used to predict the labels of the test set. The predictions of every subclassifier in the ensemble on a test instance are combined via averaging, and the resulting probabilities represent the final class distribution of the instance.

4.3.6 Ensembles using dynamic balancing with positive: STEP and CTEP

The following two approaches use the dynamic balancing technique proposed in [Stanescu and Caragea, 2014a], found to be successful for the classical self-training algorithm when the dataset exhibits imbalanced distributions. The dynamic balancing occurs during the semi-supervised iterations of the algorithm and uses only the instances that the classifier (or subclassifiers in the ensemble) predicted as positive to augment the originally labeled set. In the ensemble context, subclassifiers are used to select the most confidently predicted positive instances. These variants are named Co-Training Ensemble with Positive (CTEP) and Self-Training Ensemble with Positive (STEP), and illustrated in Algorithm 3. As before, the co-training and self-training variants differ at Steps 7-9. For CTEP, during Step 9, the instance classified as positive with topmost confidence in one view and low confidence in the second view is added to P, and vice-versa. For STEP, the two most confidently labeled positive instances are added to P, such that the augmentation rate is identical to that from CTEP.

Algorithm 3 Ensembles using dynamic balancing with positive - STEP/CTEP

```
Given: a training set comprised of labeled and unlabeled data $D = (D_l, D_u)$, $|D_l| \ll |D_u|$

Create $U$ by picking $S$ random instances from $D_u$ and update $D_u = D_u$ - $U$, $S$ = sample size
Generate $N$ balanced subsets from $D_l$ : $D_{l1}, \ldots, D_{ln}$
repeat
    Initialize $P = \emptyset$
    for $i = 1$ to $N$ do
        CT: Train subclassifiers $C_{i1}$ on $view_1$ and $C_{i2}$ on $view_2$ of balanced subset $D_{li}$
        ST: Train subclassifier $C_i$ on combined views of balanced subset $D_{li}$
        CT: Classify instances in $U$ using subclassifiers $C_{i1}$ and $C_{i2}$
        ST: Classify instances in $U$ using subclassifier $C_i$
        CT: Use $C_{i1}$ and $C_{i2}$ to select 2 positive instances and add them to $P$
        ST: Use $C_i$ to select 2 positive instances and add them to $P$
    end for
    Augment each balanced subset with the instances from $P$
    Discard remaining unused instances from $U$
    Create a new unlabeled sample $U$ and update $D_u = D_u$ - $U$
until $U$ is empty (i.e., the unlabeled data is exhausted)
```

4.3.7 Ensembles that distribute the newly labeled instances: CTEOD and STEOD

Our next semi-supervised ensemble variants are based on CTEO and STEO, respectively, and distribute the most confidently labeled instances among the classifiers in the ensemble. They are referred to as Co-Training Ensemble Original Distributed (CTEOD) and Self-Training Ensemble Original Distributed (STEOD), and shown in Algorithm 4. In CTEOD and STEOD, as opposed to CTEO and STEO, instances are distributed such that each balanced subset receives two unique instances, one positive and one negative, from each view, instead of adding all instances from P to every balanced subset. The idea that motivated this change was that different instance distributions would ensure a certain level of diversity for the constituent classifiers of the ensemble. In Algorithm 4, the co-training and self-training variants differ at Steps 6-8. As can be seen, the main difference compared to CTEO and STEO is at Step 9, where classifier C_{i1} trained on $view_1$ is augmented with the top positive and top negative instances as predicted by classifier C_{i2} trained on $view_2$, and vice-versa. Therefore, each balanced subset is augmented with two positive instances and two negative instances, and the ensemble better conserves its initial diversity.

Algorithm 4 Ensembles that distribute newly labeled instances - CTEOD/STEOD

```
Given: a training set comprised of labeled and unlabeled data $D = (D_l, D_u)$, $|D_l| \ll |D_u|$
Create $U$ by picking $S$ random instances from $D_u$ and update $D_u = D_u$ - $U$, $S$ = sample size
Generate $N$ balanced subsets from $D_l$ : $D_{l1}$, ..., $D_{ln}$
repeat
    for $i = 1$ to $N$ do
        CT: Train subclassifiers $C_{i1}$ on $view_1$ and $C_{i2}$ on $view_2$ of balanced subset $D_{li}$
        ST: Train subclassifier $C_i$ on combined views of balanced subset $D_{li}$
        CT: Classify instances in $U$ using subclassifiers $C_{i1}$ and $C_{i2}$
        ST: Classify instances in $U$ using subclassifier $C_i$
        CT: Use $C_{i1}$ and $C_{i2}$ to select 2 positive instances and 2 negative instances
        ST: Use $C_i$ to select 2 positive instances and 2 negative instances
        Augment current balanced subset, $D_{li}$, with selected positive and negative instances
    end for
    Discard remaining unused instances from $U$
    Create a new unlabeled sample $U$ and update $D_u = D_u$ - $U$
until $U$ is empty (i.e., the unlabeled data is exhausted)
```

4.3.8 Ensembles that distribute only positive instances - CTEPD and STEPD

Our last semi-supervised ensemble variants are based on CTEP and STEP. We again use the dynamic balancing technique from [15] that adds only positive instances in the semi-supervised iterations. In addition, instances are distributed among the balanced labeled subsets, such that diversity is maintained and the subclassifiers are trained on diverse enough instance subsets, thus increasing the diversity of the constituent ensemble classifiers. The resulting variants are named Co-Training Ensemble with Positive Distributed (CTEPD) and Self-Training Ensemble with Positive Distributed (STEPD),and shown in Algorithm 5.

The co-training and self-training variants differ at Steps 6-8. Overall, at each iteration, 2N unique positive instances augment the ensemble in which N is the imbalance degree since two instances originated from each co-training subclassifier. More specifically, each of the N subclassifier receives two positive instances, different from the instances received by the other subclassifiers.

Algorithm 5 Ensembles that distribute only positive instances - CTEPD/STEPD

```
1:  Given: a training set comprised of labeled and unlabeled data $D = (D_l, D_u)$, $|D_l| \ll |D_u|$

2:  Create $U$ by picking $S$ random instances from $D_u$ and update $D_u = D_u$ - $U$, $S$ = sample size
3:  Generate $N$ balanced subsets from $D_l$ : $D_{l1}, \ldots, D_{ln}$
4:  repeat
5:      for $i = 1$ to $N$ do
6:          CT: Train subclassifiers $C_{i1}$ on $view_1$ and $C_{i2}$ on $view_2$ of balanced subset $D_{li}$
            ST: Train subclassifier $C_i$ on combined views of balanced subset $D_{li}$
7:          CT: Classify instances in $U$ using subclassifiers $C_{i1}$ and $C_{i2}$
            ST: Classify instances in $U$ using subclassifier $C_i$
8:          CT: Use $C_{i1}$ and $C_{i2}$ to select 2 positive instances and add them to $P$
            ST: Use $C_i$ to select 2 positive instances and add them to $P$
9:          Augment the current balanced subset with positive and negative instances
10:     end for
11:     Discard remaining unused instances from $U$
12:     Create a new unlabeled sample $U$ and update $D_u = D_u$ - $U$
13: until $U$ is empty (i.e., the unlabeled data is exhausted)
```

4.4 Data and feature representation

For our empirical evaluation, we used five imbalanced and relatively large datasets, originally published in [Schweikert et al., 2008] and used for a domain adaptation study. The datasets belong to five organisms, *C. elegans*, which contains approximately 120K instances, and *C. remanei, P. pacificus, D. melanogaster*, and *A. thaliana*, which contain approximately 160K instances each. In each of these datasets, the true acceptor splice sites represent 1% of the total number of instances, hence the datasets exhibit a 1-to-99 imbalance ratio. The class label of each instance is either positive to indicate a true acceptor splice site, or negative to indicate a decoy splice site.

In our previous work [Stanescu and Caragea, 2014a;b], we used 141-dimensional feature vectors to represent instances, $x = (x_1, x_2, ..., x_N) \in \mathbb{R}^N$ $(N = 141)$. Each dimension corresponds to a position in the original sequences, and takes as values one of the four nucleotides $\{A, C, G, T\}$, as shown in Figure 4.1. Specifically, feature x_i indicates the nucleotide found at the corresponding position i. In the current work, because the co-training algorithm requires two views of the data, we use the nucleotide/position representation as the first view and the 3-nucleotide/position representation from [Herndon and Caragea, 2014] as the second view. As the name suggests, 3-nucleotides are sequences of length 3 (also referred to as 3-mers or "codons"). Intuitively, 3-nucleotides can capture more context information, as compared to single nucleotides. The 3-nucleotide/position representation, thus, captures additional correlations between nucleotides, while maintaining a low number of features (specifically, 139 features for our sequences which have length 141), thereby making the two views comparable. Given that nucleotide/position and 3-nucleotide/position features have shown to be effective in a domain adaptation scenario [Herndon and Caragea, 2014], we hypothesize that semi-supervised learning could also benefit from these feature representations. For self-training, we used the two views together and trained the classifiers on the complete set of features.

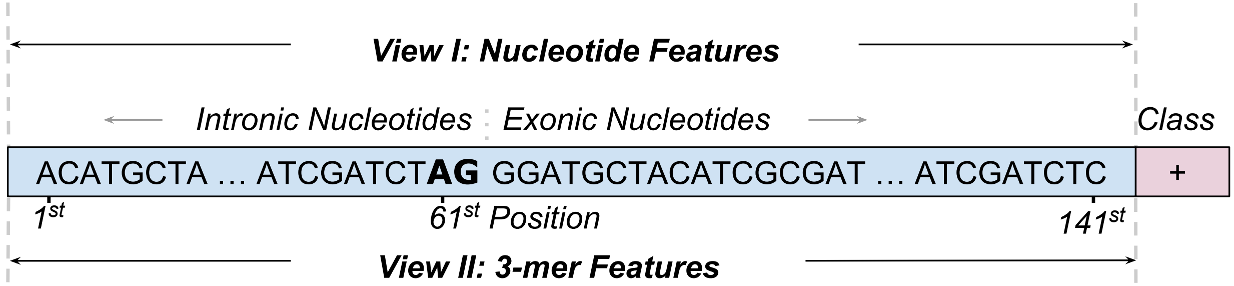


Figure 4.1: **Co-training Views of Acceptor Splice Site**: Each instance is a 141-nt window around the splice site, with the "AG" dimer starting at position 61. The sequence is used to generate two views for co-training: one based on nucleotides and another one based on 3-mers.

4.5 Experimental setup

4.5.1 Research questions

The experiments were designed to answer the following research questions:

1. Which ensembles are more affected by imbalanced distributions, supervised ensembles or semi-supervised ensembles?

2. How does the performance of the approaches vary with the imbalance degree?

3. What is the best strategy for utilizing newly labeled instances when using ensembles of semi-supervised classifiers trained on highly imbalanced data?

The five datasets used in this study were labeled, and therefore we were able to create, via re-sampling, various data subsets with various imbalance degrees (from 1-to-5 to the original 1-to-99), in order to observe the algorithms' performance with respect to the imbalance degree. For example, in the original *D. melanogaster* dataset, with the imbalance degree of 1-to-99, there are $159,748$ instances, 1,598 positives and 158,150 negatives. In order to create the dataset for each experiment, we kept the positive instances and re-sampled at random N number of negative instances to obtain a new dataset with an imbalance degree of 1-to-N. For example, in the 1-to-5 experimental dataset for *D. melanogaster*, there are 9,588 instances, 1,598 positives and 7,990 negatives. The rest of the datasets, corresponding to higher imbalance degrees, were built incrementally so that the dataset with the imbalance degree of 1-to-10 contains all the instances from the 1-to-5 dataset, and also contains additional negative instances to reach the desired imbalance.

As can be seen, for each experiment, the number of instances varies, and in the semi-supervised iterations, we used a sample size proportional to the dataset size, such that the experiments iterate roughly the same number of times.

Because classifiers are highly susceptible to data variation and prone to sampling bias, we evaluated the models using 10-fold cross validation in which nine folds were used to train the model and the tenth fold was used for testing. Data comprising the nine training folds is further divided into labeled and unlabeled. We randomly pick labeled instances such that the ratio of positive to negative is maintained and the total number of instances represents no more than 1%.

4.5.2 Evaluation

Because of the highly skewed distributions of the datasets, in order to objectively measure the predictive ability of our approaches, we compared their performance in terms of the area under the Precision-Recall Curve (auPRC), which is a more appropriate assessment measure than the area under the Receiver-Operating Curve (auROC) [Davis and Goadrich, 2006; Jeni et al., 2013]. In order to evaluate the results, we averaged auPRC values for the minority (positive) class across the ten folds for each organism. While the trends are generally maintained for individual organisms, we report averages of auPRC values over the five organisms, for easier interpretation. We performed two-tailed paired t-tests, as opposed to one-tailed t-tests, to identify statistically significant differences in either direction, on all semi-supervised algorithms for all variations of imbalance degrees. The test determines if the difference between a semi-supervised ensemble algorithm and its corresponding supervised ensemble baseline (seen as a lower bound) is statistically significant [Dietterich, 1998].

4.6 Results and discussion

Our experimental results are compiled in Table 4.1. The first column represents the imbalance degree of the experiment, which is varied from 1-to-5 to 1-to-99, by randomly discarding negative (majority) instances. The second column, LBE, shows the results of the supervised lower bound, which is also an ensemble, consisting of supervised classifiers. LBE is used as the baseline against which to compare the semi-supervised approaches. From the third column onwards, each method is presented for co-training and self-training. The results are discussed by addressing the research questions. Values marked with bold font represent performances of the semi-supervised experiments that outperform the supervised lower bound. The starred (*) values denote experiments whose variation in comparison to the lower bound was found to be statistically significant by the paired t-test in all five organisms. The values marked with a plus (†) indicate experiments that the paired t-test found to be statistically significant in four out of five organisms. The values marked with a diamond (✧) indicate experiments that the paired t-test found to be statistically significant in three out of five organisms.

1. *Which ensembles are more affected by imbalanced distributions, supervised ensembles or semi-supervised ensembles?*

   The supervised baseline remains somewhat constant irrespective of the imbalance degree, showing that additional labeled data can help alleviate problems caused by extreme cases of imbalance. Note that experiments with milder degrees of imbalance contain less instances than experiments with higher degrees of imbalance, given the way we constructed our datasets. When the imbalance degree is the highest, 1-to-99, we used the entire dataset. Compared to supervised learning, semi-supervised learning ensembles show a slow decrease in performance as the imbalance degrees become more prominent, most probably due to the fact that additional unlabeled data is more difficult to label correctly.

2. *How does the performance of the approaches vary with the imbalance degree?*

   As can be seen from the table, for lower degrees of imbalance (1-to-5 to 1-to-40), semi-supervised ensembles are considerably surpassing the supervised baselines. As the experiments become increasingly difficult (the imbalance degree becomes more prominent), some semi-supervised ensembles deteriorate as a result of unlabeled data being incorrectly classified with high confidence, and they are surpassed by the supervised baselines.

   In the original study [Li et al., 2011] that inspired our CTEO and STEO variants, the ensemble approach was used to predict the sentiment polarity of Amazon reviews with imbalance degrees ranging between 1-to-5 and 1-to-8, and proved to be superior to supervised baselines. Our variants, CTEO and STEO, also produced good results for experiments with relatively low imbalance degrees, 1-to-5 and 1-to-10. From 1-to-20 onwards, however, the CTEO and STEO semi-supervised ensembles performed worse than their supervised baselines, but, surprisingly, the self-training ensembles more effectively utilized the unlabeled data as compared to the co-training ensembles. For approaches that employ the "dynamic balancing" technique [Stanescu and Caragea, 2014a] in which only positive instances are used, the ensemble based on co-training CTEP leveraged the unlabeled data and surpassed the supervised counterpart for experiments with up to 1-to-60 imbalance degree, after which point no discernible difference was observed between CTEP and the baseline. The ensemble based on self-training, STEP, is more sensitive and was deteriorated by the unlabeled data beginning with Experiment 1-to-10. The "pseudo" positive instances could have been misclassified, thereby misleading the classifiers, which all use the same newly labeled positive instances. In general, the ensembles that do not distribute the instances among their subclassifiers, deteriorate and fall below the baseline for moderate and high degrees of imbalance. Variants of the algorithms where instances are distributed tend to outperform the other approaches. When both positive and negative instances are used to augment the labeled data, CTEOD and STEOD outperformed the not-distributed versions CTEO and STEO. The self-training based approach STEOD still falls below the supervised baseline for experiments over 1-to-50, but the co-training based approach CTEOD is surpassing the baseline for all experiments. The variants CTEPD and STEPD, which add only positive instances and distribute them, surpassed the baseline for all experiments. No significant difference in performance between CTEOD and CTEPD was observed, but STEPD outperformed STEOD and surpassed the baseline in all experiments. Thus, the "dynamic" balancing approach proved to be more useful for the self-training based ensemble.

3. *What is the best strategy for utilizing newly labeled instances when using ensembles of semi-supervised classifiers trained on highly imbalanced data?*
   One important observation that can be made based on our results is that the distribution of the newly labeled instances among subclassifiers in order to ensure subclassifier diversity is a useful approach for semi-supervised ensembles. Variants that distribute the newly labeled instances (either positive and negative for CTEOD and STEOD, or solely positive for CTEPD and STEPD) achieved overall better performance than the classifiers that receive all the newly labeled instances (CTEO, STEO, CTEP, and STEP). Therefore, the conclusion is that diversity in this case is more useful than the addition of substantially more“pseudo” (newly) labeled instances during the semi-supervised iterations.

Our results for the paired t-test showed no particular consistency, specifically some experiments and results were statistically significant and others were not.

4.7 Conclusions

In this work, we proposed and studied several ensemble-based variants of two popular semi-supervised learning algorithms, self-training and co-training, and tested their performance on the task of predicting splice sites. The task was formulated as a binary classification problem and the models' performance was tested on five large acceptor splice site datasets from five organisms. We adapted the ensembles to address the highly imbalanced datasets of our case study, and we used various approaches to augment the labeled data during the semi-supervised iterations. Our results showed that one important constraint of any ensemble (based on self-training or co-training) is to maintain diversity of the ensemble's subclassifiers, by augmenting the labeled subsets of subclassifiers with unique newly labeled instances. Maintaining the ensemble diversity by adding less but unique instances to each

Table 4.1: **Results from Semi-supervised Ensemble-based Approaches**: The values represent averages of auPRC values for the positive class over the five organisms when the class imbalance degree varies from 1-to-5 to 1-to-99 and the amount of labeled instances represents less than 1% of the training data. LBE is the ensemble-based supervised lower bound. CTEO and STEO are the co-training-based and self-training-based ensembles inspired by the original approach in [Li et al., 2011]. CTEP and STEP are the co-training and self-training based ensembles that use the "dynamic balancing" approach introduced in [Stanescu and Caragea, 2014a], in which only positive instances are used in semi-supervised iterations to augment the originally labeled training data. CTEOD and STEOD add positive and negative instances but distribute them among all subclassifiers, such that the balance and diversity of each subclassifier's labeled subset is maintained. CTEPD and STEPD use "dynamic balancing" but also distribute instances among all subclassifiers. The bold font denotes the semi-supervised experiments that outperform the lower bound. The starred (*) values denote experiments whose variation in comparison to the lower bound was found to be statistically significant by the paired t-test in all five organisms. The values marked with a plus (†) indicate experiments that the paired t-test found to be statistically significant in four out of five organisms. The values marked with a diamond (✧) indicate experiments that the paired t-test found to be statistically significant in three out of five organisms.

Imbal. Degree	**LBE**	**CTEO**	**STEO**	**CTEP**	**STEP**	**CTEOD**	**STEOD**	**CTEPD**	**STEPD**
1-to-5	0.452	**0.526**✧	**0.567***	**0.647***	**0.479**✧	**0.692***	**0.652***	**0.644†**	**0.612**✧
1-to-10	0.434	**0.462**	**0.455†**	**0.557†**	0.343†	**0.584***	**0.573†**	**0.584†**	**0.573†**
1-to-20	0.437	0.434	**0.440**✧	**0.522†**	0.292✧	**0.515**✧	**0.529†**	**0.523**✧	**0.526***
1-to-25	0.437	0.384✧	0.423✧	**0.497**✧	0.245*	**0.507**✧	**0.465**✧	**0.510**✧	**0.507†**
1-to-30	0.430	0.336*	0.408✧	**0.484**✧	0.239*	**0.509†**	**0.470**✧	**0.503**✧	**0.514***
1-to-40	0.443	0.404†	0.409	**0.492**✧	0.222†	**0.503**✧	**0.468**	**0.504**✧	**0.497†**
1-to-50	0.450	0.372†	0.409✧	**0.491**	0.236*	**0.508**✧	**0.451**	**0.504**	**0.486**
1-to-60	0.471	0.388†	0.398	**0.472**	0.195†	**0.496**	0.423	**0.494**✧	**0.474**
1-to-70	0.450	0.392†	0.411	**0.462**	0.207†	**0.474**✧	0.444	**0.480**✧	**0.478**
1-to-75	0.454	0.388	0.399✧	**0.460**✧	0.249†	**0.483**✧	0.435	**0.483**	**0.471**
1-to-80	0.449	0.353†	0.386†	0.436	0.204*	**0.457**	0.421✧	**0.460**✧	**0.465†**
1-to-90	0.453	0.359†	0.410	0.449	0.242	**0.470**	0.423	**0.473†**	**0.456**
1-to-99	0.446	0.376	0.389✧	0.440†	0.226†	**0.464**	0.414	**0.459**	**0.457**

subclassifier is a better approach than adding the same (larger sets of) instances to all subclassifiers.

In order to address highly skewed distributions, we found that dynamically balancing of ensembles by utilizing only positive instances during semi-supervised iterations to augment the labeled data and distributing them among constituent subclassifiers is a useful technique that benefits both types of ensembles, but especially the self-training-based approaches. For co-training-based approaches, whether instances from both classes are added (CTEOD) or just positives (CTEPD), the performance variations are negligible. Both approaches CTEPD and CTEOD surpass the other semi-supervised ensembles studied.

In general, our results show that ensembles based on self-training are surpassed by the ensembles based on co-training, a trend that has been reported many times in the literature for single classifiers, *e.g.*, in the prediction of alternatively spliced exons [Stanescu et al., 2015], or text classification [Nigam and Rayid, 2000].

As part of future work, we consider exploring other base learners (*e.g.*, large margin classifiers) for self-training and co-training algorithms. Given that aggregated stacking produced the best results for protein function prediction and genetic interactions prediction in [Whalen and Pandey, 2013], it would be interesting to explore meta-learning and ensemble selection for the splice site prediction problem. Transductive approaches demonstrated great potential for protein classification from imbalanced datasets [Kondratovich et al., 2013], and SVM has previously been shown to successfully identify splice sites [Sonnenburg et al., 2007]. Therefore, the behavior of SVM in a transductive context is of interest in relation to splice site prediction.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

A.S. and D.C. designed the study. A.S. carried out the computational aspect of the analysis. All authors participated in the writing of the manuscript; all authors read and approved the final manuscript.

Acknowledgements

The computing for this project was performed on the Beocat Research Cluster at Kansas State University, which is funded, in part, by grants MRI-1126709, CC-NIE-1341026, MRI-1429316, CC-IIE-1440548.

Declarations

The authors gratefully acknowledge financial support for the publication of this article from

the Department of Computing and Information Sciences at Kansas State University.

Chapter 5

Predicting Cassette Exons Using Transductive Learning Approaches

5.1 Introduction

Supervised machine learning produces dependable classifiers when large amounts of labeled data are available for training. Because of expensive generation, however, labeled data is usually scarce. Unlabeled data is easier to obtain as a result of advancement in high throughput Next Generation Sequencing (NGS) technologies. This scenario, in which limited amounts of labeled data along with considerably larger amounts of unlabeled data are available, suggests the use of semi-supervised learning (SSL), which is a learning paradigm at the intersection of supervised and unsupervised learning. SSL requires a small amount of labeled data and larger amounts of unlabeled data in order to build classification tools that perform better than models trained only on labeled data. Improving supervised classifiers by leveraging unlabeled data is a very appealing concept, although it does not always work as intended: in practice, the unlabeled data can degrade a classifier [Catal and Diri, 2009; Li and Zhou, 2011]. Understanding whether or not unlabeled data will enhance a supervised learning classifier for a particular problem is still the focus of ongoing research [Singh et al., 2009; Wang and Chen, 2013].

In a classic semi-supervised environment, a learner has access to labeled and unlabeled examples during the training phase, and the classifier must produce a classifier that can be used to predict the class of future data points not previously encountered. A subtype of SSL, called *transductive* learning, aims to classify unlabeled data without generalizing to other new, unseen examples. The goal of transduction is not to produce an *inductive* model (as in supervised and SSL), but to predict the labels of the unlabeled data to which the algorithm has access during the training phase. This may be an advantage for the algorithm, and transduction is sometimes viewed as an "easier" case of semi-supervised learning.

Theoretically, transduction is particularly suitable for genome annotation, in which a newly sequenced genome, ready to be annotated, is typically available up front, along with limited annotation. Vapnik introduced a popular large-margin transductive approach, known as Transductive Support Vector Machines (TSVM) [Vapnik and Vapnik, 1998]. TSVM has primarily been used for protein-related problems in bioinformatics [Shin et al., 2009; Kondratovich et al., 2013; Kuang et al., 2005; Pang and Kasabov, 2004], with a notable exception for promoter recognition [Kasabov and Pang, 2003].

One of the most popular graph-based transductive algorithms is Label Propagation (LP), proposed by Zhu and Ghahramani [2002], in which available labels are propagated across a graph, thereby resembling the Markov random-walk algorithm. LP was originally tested on the problem of recognizing handwritten digits, but it has also produced successful results on problems related to natural language processing (*e.g.*, word sense disambiguation). LP is one of the first methods to gain rapid popularity, and it remains in use as a baseline for derivations of graph-based algorithmic approaches.

A more recent transductive algorithm is the Adsorption algorithm, a graph-based approach first introduced by Baluja *et al.* Baluja et al. [2008] in the context of YouTube video recommendation. As a variation of "Adsorption", Talukdar and Crammer Talukdar and Crammer [2009] proposed the "Modified Adsorption" algorithm (MAD) and used it for sentiment classification on Twitter data. Several other problems have been addressed using MAD Kirchhoff and Alexandrescu [2011]; Liu and Kirchhoff [2013], but only a limited amount of work has been conducted on biology-related classification problems, with the exception of De Baets [2014], who applied MAD to a gene prioritization problem. We believe that MAD's suitability for bioinformatics comes form the fact that it is scalable to accommodate the large amounts of data available in biology-related fields, and can also handle multiclass problems. The goal of this study is to increase understand of the strengths and limitations of the three popular transductive learning algorithms (TSVM, LP, and MAD) for DNA sequence classification, with concrete applications to the problem of predicting a type of alternative splicing, specifically cassette exons.

Alternative splicing, a naturally-occurring phenomenon first observed in the late 1970s, increases proteome complexity in eukaryotes. Alternative splicing occurs after transcription. There several types of alternative splicing events, but in this work we focus on alternatively spliced exons, also called "cassette" or "skipped" exons. As illustrated in Figure 5.1, when transcribing DNA into mRNA, some exons, called "constitutive" exons, are always transcribed, while the "cassette" exons can be skipped in some isoforms.

The identification of alternative splicing events, in particular, "cassette" exons, is an essential step in the task of genome annotation and can be addressed by conducting wet-lab experiments. However, such experiments are time-consuming and require expert involvement, and unfortunately computational methods based on Expressed Sequence Tags (EST) and full length cDNA are still expensive because constructing them is difficult. Recently, RNA-Seq to genome alignments have emerged [Bonizzoni et al., 2005; Lu et al., 2009], but are not accurate enough (*e.g.*, Cufflinks only detects 44% of true alternative splicing events, as shown in a recent study [Deng and Zhu, 2014]).

Supervised machine learning approaches have also been implemented for the problem of predicting alternative splicing events, including the prediction of cassette exons. In [Rätsch et al., 2005], the task is formulated as a binary classification problem, where the two classes

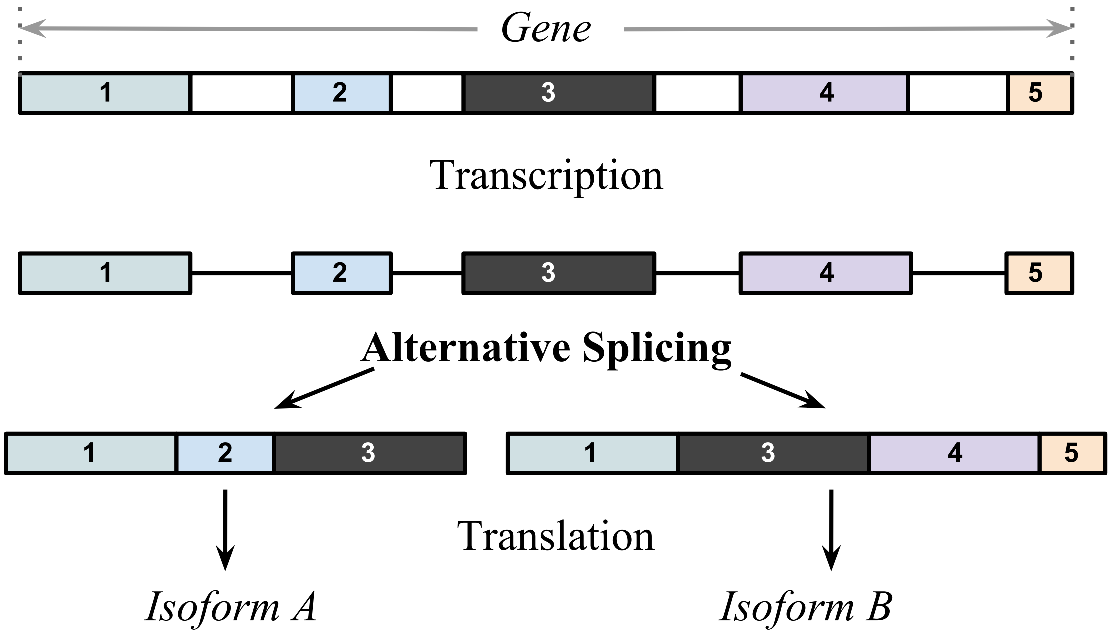


Figure 5.1: **Cassette versus Constitutive Exons**: Exons 1 and 3 are "constitutive" since they appear in all isoforms, while exons 2, 4, and 5 are "cassette" exons, because they are excluded from some isoforms.

are given by "cassette" (alternatively spliced) exons and "constitutive" exons (*i.e.*, exons that are always transcribed). In [Dror et al., 2005], the focus is on predicting alternative splicing events in humans. The authors used conserved information between human and mouse, upstream and downstream intronic sequence motifs, and length-based features in the learning process. Specialized biological kernels that model similarities between sequences have been used with SVM to predict alternative splicing [Dror et al., 2005; Ben-Hur et al., 2008].

To the best of our knowledge, no study has compared transductive algorithms on a DNA sequence classification problem; therefore our research focuses precisely on this comparison. The contributions of this paper are threefold: (1) We study and compare three transductive algorithms based on two paradigms (large-margin and graph-based) in order to evaluate the algorithms' suitability for DNA sequence classification. More specifically, we use TSVM, LP, and MAD to predict cassette exons in *Caenorhabditis elegans*; (2) We experiment with various data representations and kernels to determine which of them exhibits stronger compatibility with transductive methods. We utilize an additive kernel comprised of the spectrum kernel or weighted degree kernel with shifts on the actual sequence, along with a linear kernel on sequence length features; (3) We study the effects of the amount of labeled data on the performance of the transductive algorithms considered.

The rest of the paper is organized as follows. In Section 5.2, we review relevant and related works and present the context of our study, and explain the need for this research. We present the algorithms in Section 5.3, and data and similarity measures used are described in Section 5.4. We enumerate research questions that we want to address and outline the experimental setting in Section 5.5. The results are presented and discussed in Section 5.6. Finally, we present our conclusions in Section 5.7, where we also enumerate several directions we are interested in pursuing as future work.

5.2 Related Work

Transductive learning has been applied to a wide range of domains, including text classification, sentiment analysis, movie and video recommendation, natural language processing, image and phonetic processing, and prediction or diagnosis of various events in medical fields. In bioinformatics, transductive approaches have been successfully used primarily for protein-related problems.

Shin *et al.* Shin et al. [2009] proposed a method for combining multiple graphs obtained from several independent and complementary sources of information. The resulting combined graph was used with spectral clustering to determine functional classes of yeast proteins, a multiclass prediction problem. Weston *et al.* Weston et al. [2005] classified protein domains into SCP super families (SCP stands for Structural Classification of Proteins). The authors employed cluster kernels (bagged mismatch and neighborhood mismatch kernels) to utilize unlabeled data and labeled data. Kondratovich *et al.* Kondratovich et al. [2013] utilized TSVM for the problem of molecule activity prediction.

Comparative studies of transductive algorithms have been conducted for sentiment classification, including a recent study by Yong *et al.* Ren et al. [2014] at the document level, for underresourced languages. The authors compared MAD and LP and ran experiments on datasets from three domains (hotels, notebooks, and books). The datasets consist of approximately 4,000 reviews, out of which a balanced subset of 300 comprised labeled instances, manually annotated in terms of sentiment polarity (150 positive and 150 negative reviews). Yong *et al.* Ren et al. [2014] also decreased the amount of labeled data (from 300 instances to 20 instances) in order to assess the algorithms' behavior with various amounts of labeled data. Results showed that MAD outperformed LP. We conduct a similar study, but we compare TSVM, LP, and MAD on a biological (DNA) classification problem.

For DNA classification, purely SSL approaches, such as Expectation Maximization, Self-training, and Co-training, have been studied for the problem of predicting alternatively spliced exons Stanescu et al. [2015] and acceptor splice sites Stanescu and Caragea [2014a;b]. However, the collection of studies on purely transductive approaches is not as rich; here we mention a notable exception from Kasabov *et al.* Kasabov and Pang [2003], who used TSVM on the problem of promoter recognition in a multispecies dataset.

Because transductive learning algorithms rely on similarities, biological kernels are also relevant to our work. Specialized biological kernels have been proven to enhance classification capabilities of supervised large-margin classifiers, for protein related problems. For example, Kuang *et al.* Kuang et al. [2005] used SVM with profile-based string kernels from PSI-BLAST profiling for the problems of protein classification and detecting remote homology of proteins, in a supervised classification setting. Rangwala and Karypis Rangwala and Karypis [2005] designed two classes of kernels, window-based and alignment-based, for SVM to be used for the problem of detecting remote homologs and identifying folds, respectively.

For supervised DNA sequence classification, Rätsch *et al.*Rätsch et al. [2005] created a biological string kernel, called the weighted degree kernel with shifts, and used this kernel with SVM. We also employ this kernel in our study but in a transductive framework.

5.3 Transductive Approaches Studied

In this section we describe the types of methods compared in our study, with a focus on transductive learning and determining which algorithm produces the best results. Many popular transductive algorithms have different assumptions, but in this study we will focus on one margin-based algorithm in this study, namely TSVM (Section 5.3.1) and two graph-based algorithms, LP (Section 5.3.2) and MAD (Section 5.3.3). Other transductive approaches such as Learning with Local and Global Consistency Zhou et al. [2004] and Label Matrix Normalization Li et al. [2013], did not produce satisfactory results on our data, and were therefore excluded from this paper.

5.3.1 Transductive Support Vector Machines (TSVM)

The TSVM algorithm Vapnik and Vapnik [1998] is an extension to the classical SVM algorithm. The "low density separation" assumption states that points residing in the same cluster share the same label and that the decision boundary should reside in a low density region, known as a large margin. This separating hyperplane maximizes the margin while minimizing the training error, as a penalty term for misclassification must be introduced for the non-linearly separable cases. Because TSVM optimization is an intractable problem, Joachims Joachims [1999] proposed a solution resembling the classical "self-training" approach because it uses the completely supervised SVM built on the labeled data, and then "switches" labels of the unlabeled (test) data in order to optimize the objective function while consistently classifying the originally labeled examples. In other words, the new boundary must be consistent with the labeled data. In this paper, we use SVMLight Joachims [1999] implementation of TSVM that was designed to accommodate problems with datasets of no more than a few thousand examples.

Similar to SVM, TSVM can benefit from the "kernel trick", in which the traditional dot product that appears in the original SVM optimization problem is replaced by a nonlinear kernel function, which provides an alternative to measuring the similarity between two instances. Instead of utilizing the dot product of the instances' vector representation, the kernel models different notions of similarity that are more appropriate for the problem studied. This kernelized version that transforms the representation of instances to a higher dimensional space allows customized solutions to calculate similarities between instances. We experiment with various sequence representations and similarity kernels, as explained in Section 5.4. The same representations and similarity kernels are used to build similarity (affinity) matrices for the graph-based approach.

5.3.2 Label Propagation (LP)

In graph-based methods, all available data, including labeled instances $\{(x_1, y_1), ..., (x_l, y_l)\}$ and unlabeled (or test) instances $\{(x_{l+1}, y_{l+1}), ..., (x_u, y_u)\}$ where usually $l \ll u$, are represented as nodes in an undirected graph. Formally, the graph is defined as $G = \{V, E, W\}$, where V represents the set of nodes (vertices), $E = V \times V$ is the set of edges that represents every pair of nodes, and W is the set of weights associated with the edges. Weights on the edges reflect the similarities between the connected nodes. The "smoothness" assumption of graph-based methods states that because nodes connected by a strong edge are very similar, the nodes are more likely to share the same label. LP Zhu and Ghahramani [2002] is a transductive algorithm that spreads labels of the originally labeled nodes throughout the graph in order to classify unlabeled nodes, which receive a class distribution in the form of "soft" labels (probabilities). The elements of the vector Y_v maintain the node's v prior class distribution, and are different from zero if the node is labeled, and null if the node is unlabeled. The second vector $\widehat{Y_v}$ is initialized to zero and its dimensions get assigned values for each class, as inferred by the algorithm. The smoothness assumption can be mathematically formulated as the optimization problem from Equation (5.1), where labels $\widehat{y}_i$ and $\widehat{y}_j$ of nodes v_i and v_j, respectively, should be similar for a large W_{ij} in order to minimize the function, while ensuring that the original labels are maintained.

$$\min \sum_{i,j} W_{ij}(\widehat{y}_i - \widehat{y}_j)^2, \; s.t.\ \widehat{Y}_l = Y_l. \tag{5.1}$$

The function from Equation (5.1) can be solved iteratively, using Algorithm 6, which utilizes the nodes' label distribution given in the form of a matrix $Y = (l+u) \times C$, where l represents the number of labeled examples, u is the number of unlabeled examples, and C is the number of classes. Next, a probabilistic transition matrix T is computed such that the probability of jumping from node i to node j is

$$T_{ij} = \frac{w_{ij}}{\sum_{k=1}^{l+u} w_{kj}} \tag{5.2}$$

After the initialization of $\widehat{Y_v}$ with class labels for $\widehat{Y_l}$ and arbitrary values for $\widehat{Y_u}$, actual propagation occurs (line 4 in Algorithm 6). The algorithm continues with re-setting of the initial labels (line 5 in Algorithm 6) in order to reinforce the labels of the originally labeled training data. This operation is referred to as "clamping" of the labels. The iterations are then repeated until convergence (*i.e.*, until the propagation is complete and the labels do not vary much between iterations).

Algorithm 6 Label Propagation (LP)

```
Require: Similarity Graph $G = \{V, E, W\}$, Label Matrix $Y_v$
1: Compute $T = D^{-1}W$, where $D$ is diagonal degree matrix
2: Initialize $\widehat{Y_v} = Y_v$
3: repeat
4:     $\widehat{Y_v} = T\widehat{Y_v}$
5:     $\widehat{Y_v} = Y_v, (v \in V_l)$ "Clamp" the original labels
6: until $\widehat{Y_v}$ converges
7: return $\widehat{Y_v}$, the estimated probability distribution over the labels of vertex $v$
```

5.3.3 Modified Adsorption (MAD)

The original Adsorption Baluja et al. [2008] algorithm resembles the concepts of LP Zhu and Ghahramani [2002] and also Zhu et al. [2003]. MAD Talukdar and Crammer [2009] can be considered a "random walk"-type approach that propagates labels throughout the graph in a more controlled manner, by the means of three probabilities: (1) injection probability, p_v^{inj}, which returns the initial Y_v label distribution of a node; (2) continuation probability, p_v^{cont}, that continues to propagate the label from v onto the next node v' with probability proportional with the similarity between the two nodes, given by:

$$Pr[v'|v] = \frac{W_{v'v}}{\sum_{u:(u,v)\in E} W_{v'v}} \tag{5.3}$$

and (3) termination (or abandonment) probability, p_v^{term} that terminates the propagation process for a node. The condition is that $p_v^{inj}+p_v^{cont}+p_v^{term} = 1$.

LP and MAD differ from each other in (1) that MAD does not reinforce the initial class distribution carried by the training labeled data, thereby presumably dealing with potential noise in the original label data and (2) that MAD can express uncertainty regarding classification through the means of a dummy label that acts as an extra "class" initialized to zero in the beginning and later assigned the default abandonment probability when/if the label propagation is abandoned at a given training phase (iteration).

The actual class distribution of every node v is stored in Y_v, which is a $(C+1)$-dimensional row vector enhanced to hold the extra dummy variable ν. C is the number of classes. Similar to the notation from LP, the predicted (inferred) class distribution of every node is stored in $\widehat{Y}_v$. MAD also utilizes a $(C+1)$-dimensional row vector $\mathbf{r}$ whose elements are set to zero, except for the extra element holding the dummy label, which is set to 1 ($\mathbf{r}_l = 0$ for $l \neq \nu$, $\mathbf{r}_\nu = 1$).

MAD, an extensions to the original Adsorption algorithm, has a well-defined optimization function (Equation 5.4) that can be solved iteratively in matrix form using the Jacobi method (Algorithm 7). The first term of the cost function captures the constraint that the inferred labels should not significantly differ from the original labels. The second term ensures the "smoothness" assumption and the third term is a regularizer that discourages uncertainty. The importance of each term is controlled by three hyperparameters, μ_1, μ_2, and μ_3.

$$\begin{aligned} \min \sum_v [\mu_1 \sum_k p_v^{inj} (Y_{vk} - \widehat{Y_{vk}})^2 + \\ \mu_2 \sum_v \sum_j p_v^{cont} w_{vj} (\widehat{Y_{vk}} - \widehat{Y_{jk}})^2 + \\ \mu_3 \sum_k p_v^{term} (\widehat{Y_{vk}} - R_{vk})^2] \end{aligned} \tag{5.4}$$

Algorithm 7 Modified Adsorption (MAD)

```
Require: Similarity Graph $G = \{V, E, W\}$, Label Matrix $Y_v$, Probabilities $p_v^{inj}$, $p_v^{cont}$, $p_v^{term}$,
    $\forall v \in V$
Initialize $\widehat{Y_v} = Y_v$
repeat
    $D_v = \frac{\sum_u W_{uv} \widehat{Y_v}}{\sum u W_{uv}}$
    for $v \in V$ do
        $\widehat{Y_v} = p_v^{inj} \times Y_v + p_v^{cont} \times D_v + p_v^{term} \times \mathbf{r}$
    end for
until $\widehat{Y_v}$ converges
return $\widehat{Y_v}$, the estimated probability distribution over the labels of vertex $v$
```

In this work, we use the Junto implementation of LP and MAD, from *https://github.com/parthatalukdar/junto* and we maintain the default parameters. All three transductive approaches explored in this work require a similarity measurement, in the form of a kernel function, for each pair of instances, such as in the case of TSVM, or they require a similarity measurement in the form of a similarity matrix, as in the case of the graph-based MAD and LP algorithms.

5.4 Data Representation and Similarity Measures

In our experiments, we use genomic data from the model organism *Caenorhabditis elegans* in our experiments. The dataset was published by Rätsch et al. [2005] and it is publicly available at `http://people.kyb.tuebingen.mpg.de/raetsch/RASE.old/`. The dataset contains 3,018 nucleotide sequences of exons and adjacent introns, *i.e.*, each instance is in the form *left intron–exon–right intron*, as illustrated in Figure 5.2. Out of these 3,018 instances, 487 are labeled as alternatively spliced, meaning that the flanked exon is a cassette exon that can be skipped in some isoforms. The remaining 2,531 sequences are labeled as constitutive, meaning that the exon is present in all known isoforms. The data was labeled based on alignments between ESTs and genomic DNA.

Given the intron-exon-intron sequence, two types of features are readily available: (1) content-based features obtained directly from the DNA sequence, and (2) length-based numeric features obtained from the lengths of the exons and their flanking introns. Accordingly, two types of similarity scores can be captured by string kernels and numeric kernels, respectively. Because kernels are additive, these two scores can be added, to more accurately reflect the overall similarity between two instances. In our study, we experiment with three different ways for capturing content-based similarity at the sequence-level using string kernels, as described below. For lengths, we always use a linear kernel that computes the dot (inner) product between numeric features. Along with the dataset, Rätsch et al. [2005] also made available length features associated with the instances.

Length features are obtained directly from [Rätsch et al., 2005] in which lengths of each upstream intron, exon and downstream intron (of every sequence in the set) were used to generate 30-dimensional logarithmically spaced vectors for a total of 90 features per instance, corresponding to the three lengths. The set of length features also includes 3-dimensional vectors that characterize the frame of the stop codon, resulting in 15 additional features for a total of 105 length features (LG) per instance. Labels of the instances were not used during the feature generation process. The following sections describe how we used the

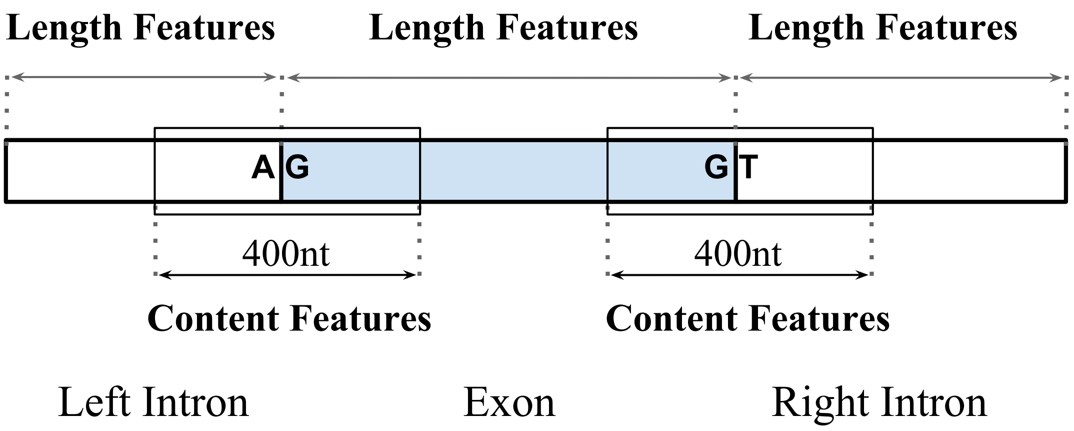


Figure 5.2: **Intron-Exon-Intron Sequence Example**: Example of an instance from the dataset in the form of intron-exon-intron. Content-based features are generated from 400-nt windows around the splice sites, while length features are obtained from the lengths of the exons and flanking introns.

string kernels to capture DNA-level similarities.

5.4.1 Weighted Degree Kernel with Shifts (WDS)

The similarity between two DNA strings using the Weighted Degree kernel with Shifts (**WDS**) [Rätsch et al., 2005] is given by the count of co-occurrences of exact k-mers at correspondent (exact or shifted) positions in the sequences, where $k \in \{1..degree\}$, and whose weights are controlled by β coefficients, with β dependent on the size of k. In order to utilize WDS, the DNA sequences must have equal lengths. Because most splicing regulatory information is typically aggregated in the proximity of splice sites, the WDS is applied on 400-nucleotide windows centered around the acceptor and donor splice sites in regions upstream and downstream of the exon. The more sequence overlap that exists close to the splice site, the higher the score captured by the WDS. Leveraging the additive property of kernels, the two score values that correspond to donor and acceptor sites are then added; the combined kernel reflects the overall sequence similarity. For more details regarding WDS, the reader is referred to [Rätsch et al., 2005].

5.4.2 K-Spectrum Kernel (k-SK)

The k-Spectrum Kernel (k-SK) is a linear kernel introduced in [Leslie et al., 2002] for strings; we combine it with the linear kernel for length features. The Spectrum Kernel, designed for protein classification using SVM, is similar in nature to the feature vector representation of sequences because it describes the content of a sequence, in terms of substring frequencies. However, it is ignorant to the order or position of such occurrences. In order to calculate the pairwise similarity of two instances (DNA strings), the k-SK uses all subsequences of a fixed length k that occur throughout the instance. If subsequences co-occur frequently throughout two DNA strings, their dot (inner) product under the kernel will be large. The intuition is that the more subsequences two DNA strings have in common, the more likely they are to be similar and share the same biological functions. Biological signals are relatively short, usually 6-14 nucleotides long. We use the Spectrum Kernel with length $k = 6$ denoted **6SK** because a majority of the biological motifs described next are 6-nucleotides long. Other studies of exonic splicing regulators have also focused on hexamers [Fairbrother et al., 2002; Wang et al., 2009].

5.4.3 *Motif*-Spectrum Kernel (MSK)

WDS and SK can be used if there is no prior knowledge about biologically significant motifs (that have an influence on the problem of interest) because WDS and SK use all possible occurrences of subsequences of variable length (in the case of WDS) or fixed length (in the case of SK) to compute similarities. In order to better understand how well "unbiased" kernels capture sequence similarity in a transductive framework, we use the Spectrum Kernel in a slightly different manner. Instead of using all occurrences of k-length subsequences, we use only a selected subset of motifs recognized to have biological significance, and we omit the rest of the subsequences. In other words, we only account for biological motifs, known as splicing regulators, established to work as signals responsible for the occurrence of alternative splicing and potentially result in good classification performance. We denote this kernel as *Motif*-Spectrum Kernel, **MKS**.

Biologically relevant signals, such as splicing regulators, can occur in exons and introns. The ones that occur in exons are called Exonic Splicing Enhancers (ESE), while those occurring in introns are called Intronic Regulatory Sequences (IRS). We use 45 ESE hexamers (6-nucleotide long) derived by Xia et al. [2010] for the *Caenorhabditis elegans* dataset. The set of IRS motifs Kabat et al. [2006] was obtained using comparative genomics in nematodes based on the observation that intronic sequences that are relevant for alternative splicing are highly conserved among closely related species. In order to form the set of IRS motifs, we combined the upstream and downstream motifs and removed duplicate motifs, resulting in a total of 165 IRS motifs assumed to be informative for alternative splicing. The class label was not used in any of these procedures, and repetitive regions were not specifically addressed. A total of 205 biological motifs with variable lengths were present. Their usefulness in a purely semi-supervised framework was reported in [Stanescu et al., 2015], and we anticipate that its quality will also aid transduction.

5.5 Experimental Setup

In this work, we investigate the performance of transductive algorithms TSVM, LP, and MAD on the binary classification problem of predicting cassette exons. Our experimental setup is designed to address the following research questions:

1. What is the most effective transductive algorithm for the problem of identifying cassette exons based on DNA sequences?

2. How does the performance of the transductive algorithms vary with the amount of labeled data?

3. What is the most useful sequence representation and similarity measure (or kernel) when classifying instances transductively?

5.5.1 Evaluation

We used 5-fold cross-validation to avoid sampling bias and to be consistent with [Rätsch et al., 2005]. Furthermore, in order to use the tuned parameters of the Weighted Degree kernel with Shifts (WDS), we utilized identical splits from the supervised study conducted on the same dataset as [Rätsch et al., 2005]. In order to simulate a transductive environment, we deliberately hide some of the labels at random.

In general, the effect of the labeled data on the classification ability, in semi-supervised and transductive frameworks, is far more significant than the effect that the same amount of unlabeled data would have [Joachims, 1999]. In order for the unlabeled instances to have an observable impact, they must significantly outnumber the labeled instances. Therefore, we limit the amount of labeled data to 20% of the total dataset, and the test (unlabeled) instances represent the remaining 80%. In order to observe variation in the algorithms' performance, we also decrease the labeled data from 20% (approximately 600 instances per fold, on average) to 5% (approximately 150 instances per fold, on average), by discarding some instances at random, while the test dataset remains the same 80% (approximately 2,415 instances per fold, on average).

Because our dataset is relatively imbalanced (with approximately 5 times more *"constitutive"* instances compared to *"cassette"* instances) – the accuracy of the predictions would not reflect the quality of the classifiers [Provost et al., 1998]. Therefore, we report the performance in terms of area under the Receiver Operating Characteristic curve (auROC) [Huang and Ling, 2005], averaged over 5 folds, and the afferent variance.

5.6 Results

We present our results in Table 5.1. The auROC values emphasized in bold font represent the best values obtained by an algorithm for a given amount of labeled data. The colored cells highlight values of the best result overall for a given amount of labeled data. In the first column, the percentages refer to the amount of labeled data used for training the algorithms. The three groups of experiments represent the performances of TSVM, LP, and MAD algorithms using each of the three data representations (and corresponding kernels): (1) the Weighted Degree kernel with Shifts (WDS) for the DNA sequence along with the Linear Kernel (LK) for the Length Features (LG), (2) the 6-Spectrum Kernel (6SK) capturing 6-mers along with the Linear Kernel (LK) for the Length Features (LG), and (3) the M-Spectrum Kernel (MSK) for the biologically relevant motifs and the Linear Kernel (LK) for the Length Features (LG).

	TSVM			LP			MAD		
	WDS+LK	6SK+LK	MSK+LK	WDS+LK	6SK+LK	MSK+LK	WDS+LK	6SK+LK	MSK+LK
	DNA+LG	6MERS+LG	Motifs+LG	DNA+LG	6MERS+LG	Motifs+LG	DNA+LG	6MERS+LG	Motifs+LG
5%	0.777±4.9E-4	0.614±3.9E-4	**0.903± 1.9E-4**	**0.800± 6.3E-4**	0.615± 4.6E-4	0.534±93.3E-6	**0.828±39.0E-4**	0.621± 4.9E-4	0.742±2.0E-4
10%	0.811±3.6E-4	0.652±4.5E-4	**0.916±59.6E-6**	**0.810±71.4E-6**	0.698± 6.3E-4	0.565±14.3E-6	**0.828± 7.9E-4**	0.729± 3.3E-4	0.781±4.3E-4
15%	0.838±3.7E-4	0.616±7.0E-4	**0.887± 1.3E-4**	0.801± 3.5E-4	**0.815±15.1E-4**	0.596±37.2E-6	0.830±37.9E-4	**0.873± 1.6E-4**	0.830±3.5E-4
20%	0.858±4.5E-4	0.700±3.4E-4	**0.926±94.8E-6**	0.814±14.1E-6	**0.864±70.9E-6**	0.612± 1.7E-4	0.888± 4.5E-4	**0.942±32.2E-6**	0.835±3.5E-4

Table 5.1: **Transductive Results for Cassette Exon Identification**: Averages of auROC values over the 5 folds and the corresponding variance, while varying the amount of labeled data from 5% to 20%, and maintaining a fixed test set of 80%. The algorithms are Transductive Support Vector Machines (**TSVM**), Label Propagation (**LP**), and Modified Adsorption (**MAD**). The first similarity measure used is the Weighted Degree kernel with Shifts (**WDS**) for the DNA sequence along with the Linear Kernel (**LK**) for the Length Features (LG). The second similarity measure is the 6-Spectrum Kernel (**6SK**) capturing 6-mers along with the Linear Kernel (**LK**) for the Length Features (LG). The third similarity measure is the M-Spectrum Kernel (**MSK**) for the exonic splicing enhancers and intronic regulatory sequences (Motifs) along with the Linear Kernel (**LK**) for the Length Features (LG). The values emphasized in bold font represent the best performance recorded by an algorithm for a given amount of labeled data, and the colored cells highlight the values of the best results overall, for a given amount of labeled data.

We discuss the results by answering the research questions.

1) *What is the most effective transductive algorithm for the problem of identifying cassette exons based on DNA sequences?* Empirical results of our study are encouraging, showing that from limited amounts of labeled data, the performance of transductive classifiers reaches high auROC values (from 0.903 to 0.942 for various amounts of labeled data). These values are comparable to the ones from our previous study of purely semi-supervised algorithms for this problem [Stanescu et al., 2015], however, a direct comparison is not possible since

the unlabeled and test sets differ in semi-supervised learning from transductive, where the unlabeled data is the actual test data to predict. Overall, TSVM performs better than MAD and LP, especially when trained on smaller amounts of labeled data (5% to 15%). However, MAD more advantageously utilizes the 20% labeled instances.

2) *How does the performance of the transductive algorithms vary with the amount of labeled data?* As expected, the amount of labeled data is a deciding factor for training quality classifiers, and auROC values for all algorithms generally increase with the increase in the amount of labeled data. The trends from our study are consistent with the trends reported on the task of sentiment classification [Ren et al., 2014].

For the 6-mers representation, MAD and LP recorded more rapid increases in performance from increasingly larger amounts of labeled data. The classification performance improved from 0.621 auROC in the case of 5% labeled data to 0.942 auROC in the case of 20% labeled for MAD, and from 0.615 auROC to 0.864 auROC in the case of LP. TSVM is not as sensitive to the amount of labeled data, and variations are not as abrupt as for graph-based approaches. However, for 6-mers and motifs, TSVM records a counterintuitive decrease in performance at 15% labeled data, most likely due to an erroneously found hyperplane, unrepresentative of the whole labeled data, also suggested by slightly higher variance. This is understandable since TSVM relies on support vectors found in the low density region, as opposed to graph-based methods that utilize a diffusion approach to propagate labels.

3) *What is the most useful sequence representation and similarity measure (or kernel) when classifying instances transductively?* WDS is particularly suitable for MAD and LP when learning from limited amounts of labeled data and somewhat useful for TSVM when additional labeled data is available. The 6SK is most appropriate for MAD, which, compared to all three algorithms, seems to be least susceptible to noise, indicated by the fact that when using 6-mers, which probably contain more noisy features than the other representations, MAD achieves better results than TSVM and LP. MSK (biological motifs) along with the length features are the most helpful for TSVM, possibly because TSVM is able to locate

a more accurate hyperplane in the space rendered by informative features (*i.e.*, biological motifs established as relevant to alternative splicing) since they are fewer than the 6-mers, which render data to a much higher dimensional space, thereby increasing the difficulty in identifying a good separation.

For 6-mers, TSVM records its worst performance as it is unable to find a correct separating hyperplane in the space generated by these features, possibly due to an unnecessarily high dimensionality (20 times higher than the motifs; 4.2K 6-mers vs 210 motifs). Because MAD has more more features available in the 6-mers set, a greater amount of common information could be propagated among the instances. However, if some of the information in the 6-mers set is noisy, the labeling becomes erroneous, since strong edges could connect positive instances to negative instances. This can potentially occur for small amounts of labeled data (*e.g.*, 5% and 10%). However, for relatively larger amounts of labeled data, (*e.g.*, 15% and 20%), the 6-mers can propagate the labels more accurately. For LP, the best performance is recorded for 6-mers, when the algorithm is presented with relatively larger amounts of labeled data (15% and 20%).

As opposed to TSVM, MAD records unsatisfactory results from MSK (the motif representation), possibly due to the fact that there are only 210 motifs available, and they don't capture overall sequence similarity as well as the set of all 6-mers used by the 6KS, or the various-length matches captured within close proximity of the splice sites by the WDS. Furthermore, a smaller set of motifs could lead to higher-degree nodes which are discouraged in MAD, hence the correct label is not propagated along the connected nodes. For LP, the motif representation is the least compatible.

5.7 Conclusions

In this study, we investigate the applicability of transductive approaches to DNA sequence classification. The case study of our work is the problem of discriminating between cassette (or alternatively spliced) and constitutive exons. Experimental results suggest that transductive learning is a useful approach for addressing DNA sequence classification tasks, but we should note that it may be possible to observe different trends for different problems.

We found that biologically relevant features are better exploited by the discriminative nature of the TSVM algorithm, which is able to find a good separation boundary in the space defined by biological motifs. However, when such features are unavailable, the k-Spectrum Kernel is more appropriate for graph-based approaches if a reasonable amount of labeled data is available. Although the best classification performance came mostly from TSVM, this is not a feasible solution when managing massive amounts of data, comprised of more than a few thousand instances. However, MAD is particularly suitable for "big data" and could solve problems posed by larger datasets. Similar to previously reported results Ren et al. [2014], MAD outperformed LP on all cases.

In future work, we plan to address other DNA sequence classification problems and evaluate graph-based algorithms on more ample datasets (with hundred thousands instances). Furthermore, choosing the appropriate similarity measure strongly influences the effectiveness of graph-based approaches and kernel-based algorithms. An investigation of other kernels and their compatibility with DNA transductive classification would also be interesting and beneficial.

Acknowledgment

The computing for this project was performed on the Beocat Research Cluster at KSU, which is funded in part by NSF grants ACI-1341026, CNS-1126709, CNS-1006860, and EPS-0919443.

Chapter 6

Conclusions and Future Work

Semi-supervised and transductive learning algorithms constitute an efficient and less expensive alternative to accumulating extra labels thereby improving classification performance. In this dissertation, we have explored the applicability and usefulness of semi-supervised and transductive learning methods for bioinformatics problems. Our experiments show that such algorithms can take advantage of the unlabeled data and ultimately achieve better accuracy than purely supervised algorithms. For the problem of cassette exon identification, the performances achieved by semi-supervised and transductive learning are as high as 0.964 auROC (as opposed to its supervised counterpart of 0.867) and 0.942, respectively.

In general, the performance of all algorithms (semi-supervised, transductive, as well as the supervised baselines) improves with more labeled data. Iterative wrapper methods, such as Expectation Maximization, Self-training, and Co-training have proven to be surpassing the predictive capabilities of their supervised counterparts when the amount of unlabeled data is at least four times as large as the amount of labeled data.

For better visualization, we have summarized the results of our experiments from Chapter 2 for cassette exon identification from Table 2.1 and Table 2.2 in the following two graphs, Figure 6.1 and Figure 6.2.

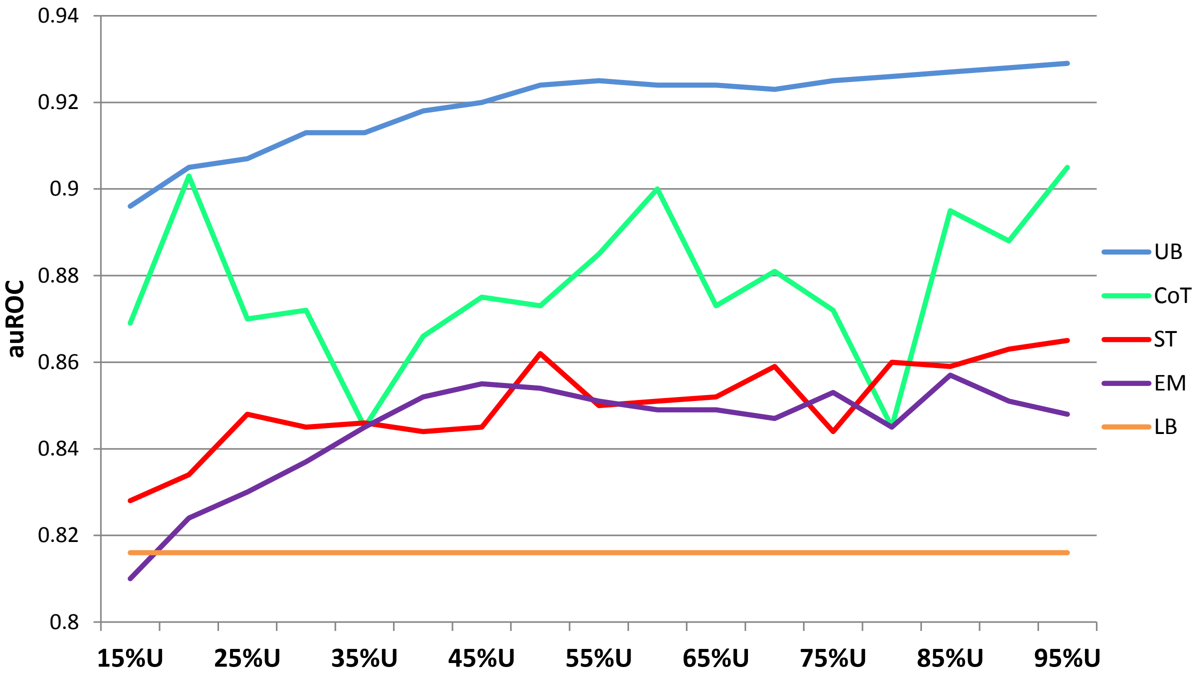


Figure 6.1: **Trends of SSL algorithms when trained from increasingly large amounts of unlabeled data** Performance (auROC) variation with increasing amounts of unlabeled data of three iterative semi-supervised algorithms, Expectation Maximization (EM), Self-training (ST), Co-training (CoT) with Naïve Bayes Multinomial (NBM) as base classifier while the labeled data remains fixed at 5% from the training set

As previously reported in the literature, Co-training is outperforming self-training, probably due to the two views "informing" each other about the best predictions. However, Co-training is also more unstable than Self-training and Expectation Maximization, with high variation in the performance when learning from increasingly large amounts of unlabeled data. One possible explanation is that during the semi-supervised iterations, instances incorrectly classified with high confidence are perturbing the classifiers, and errors reinforce themselves leading to a decrease in performance.

From the experiments of Chapter 2, another apparent trend that is consistent with previously reported results is that Random Forest, because it is an ensemble, surpasses SVM and Naïve Bayes Multinomial, reaching the highest auROC values, for both supervised (up to 0.971) and semi-supervised (up to 0.964) learning.

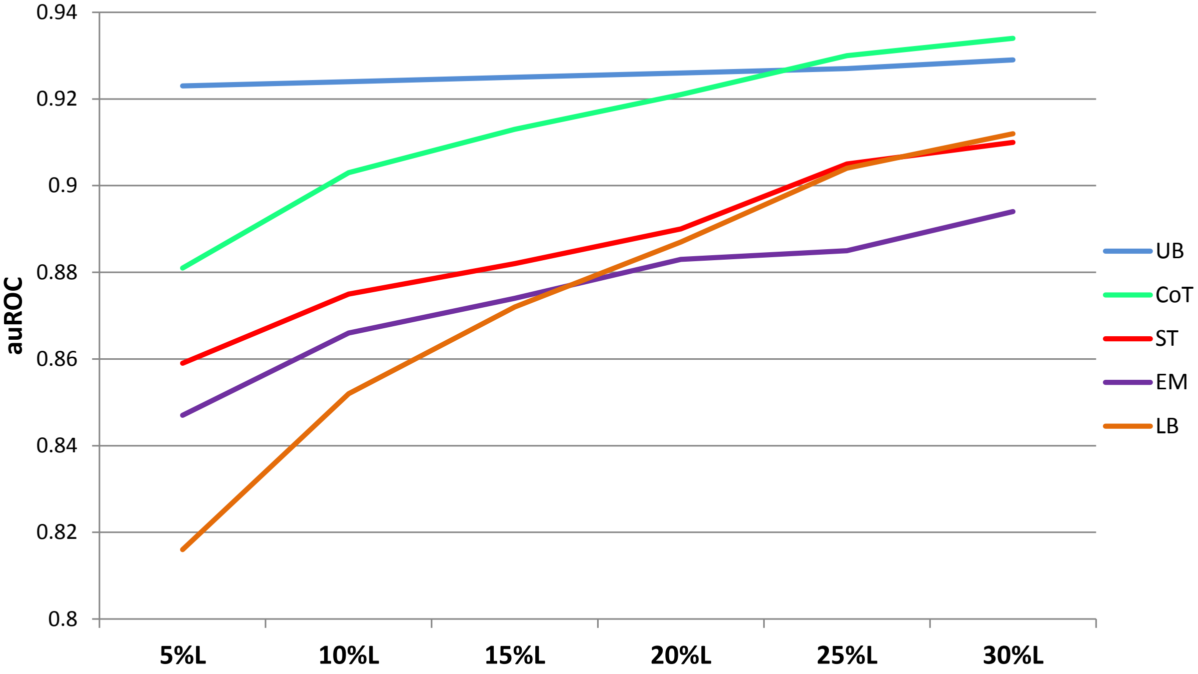


Figure 6.2: **Trends of SSL algorithms when trained from increasingly large amounts of labeled data** Performance (auROC) variation with increasing amounts of labeled data of three iterative semi-supervised algorithms, Expectation Maximization (EM), Self-training (ST), Co-training (CoT) with Naïve Bayes Multinomial (NBM) as base classifier while the unlabeled data remains fixed at 70% from the training set

We present some of the most interesting trends for two of the transductive learning algorithms from Chapter 5 in Figure 6.3. TSVM is particularly compatible with the representation from biological motifs, being able to find a good separation hyperplane with the help of relatively few, but highly predictive features. Conversely, MAD is more resilient to the noise present in the 6-mers set, whose dimensionality is more than two orders of magnitude higher than the dimensionality of the motifs.

TSVM learns very poorly from the 6-mers set; it is possibly steered in the wrong direction by the high dimensionality of the 6-mers. MAD find the set of biological motifs too small for a correct propagation of the labels throughout the graph. The variable-length positional match representation (denoted "_dna" in Figure 6.3) captured by the Weighted Degree Kernel with Shifts (WDS) represents a good alternative when biological motifs are not available, especially when the labeled data represents less than 15% of the total amount of available training instances.

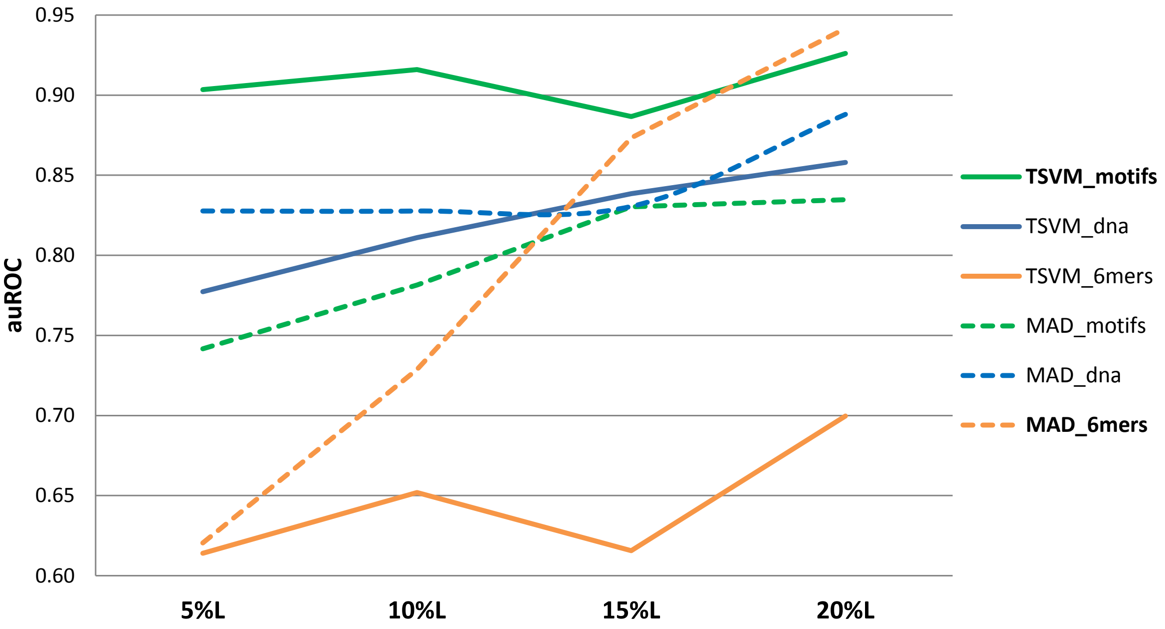


Figure 6.3: **Trends of transductive learning from increasing amounts of labeled data** Performance (auROC) variation of TSVM and MAD with DNA-based features (_dna), 6-mers, or biological motifs.

Splice site prediction, the second problem studied in this work, faces the data imbalance problem, a challenge prevalent in bioinformatics. Specifically designed for the semi-supervised paradigm, we have introduced a novel "dynamic balancing" (Section 3.2.2) technique that can improve traditional iterative semi-supervised classifier when the data exhibits highly skewed class distributions. The method requires that the originally limited set of labeled instances be augmented only with newly classified positives. This approach ensures that the already insufficient labeled information is not wasted (as in the case of under-sampling, where discarding negative instances results in information loss) and it also potentially helps overfitting (as in the case of over-sampling, where instance replication may cause bias towards the labeled set yielding in poor generalization).

Some interesting trends are presented in Figure 6.4 and Figure 6.5 and correspond to the results from Section 3.4.2. STP is consistently outperforming the other balancing methods as well as the original baseline. As the imbalance gets more and more prominent, the performance of all algorithms decreases, showing that a good classification function is more difficult to obtain since the learning algorithms are more biased towards the majority class. Also, as the problem gets more difficult, *i.e.*, the imbalance degree is higher than 1-to-50, over-sampling is outperforming under-sampling when learning from 5% labeled data. When learning from 10% labeled data, over-sampling takes over at imbalance degrees of 1-to-90 and higher, which leads to the conclusion that if enough labeled data is available, under-sampling should be preferred, since it decreases computation times.

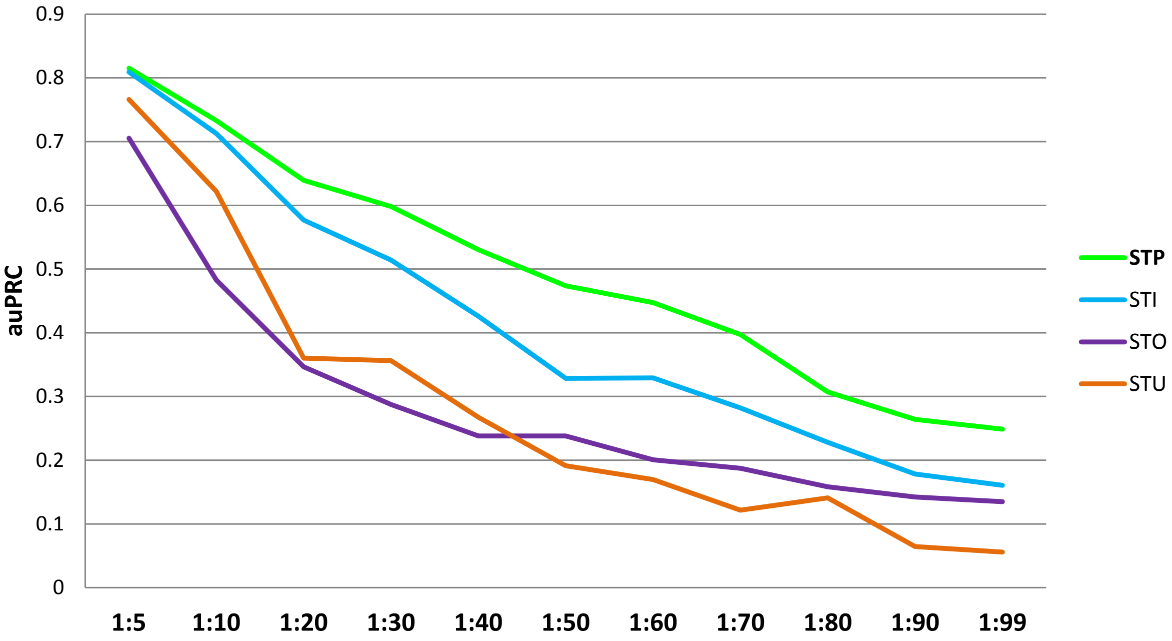


Figure 6.4: **Trends of SSL and data balancing methods for various imbalance degrees when learning from 5% labeled data** Performance (auPRC)variation of self-training based on Naïve Bayes from increasingly high imbalance degrees: Self-training Imbalanced (STI), Self-Training with Positives (STP), Self-training from Under-sampled (STU) and Self-training from Over-sampled (STO).

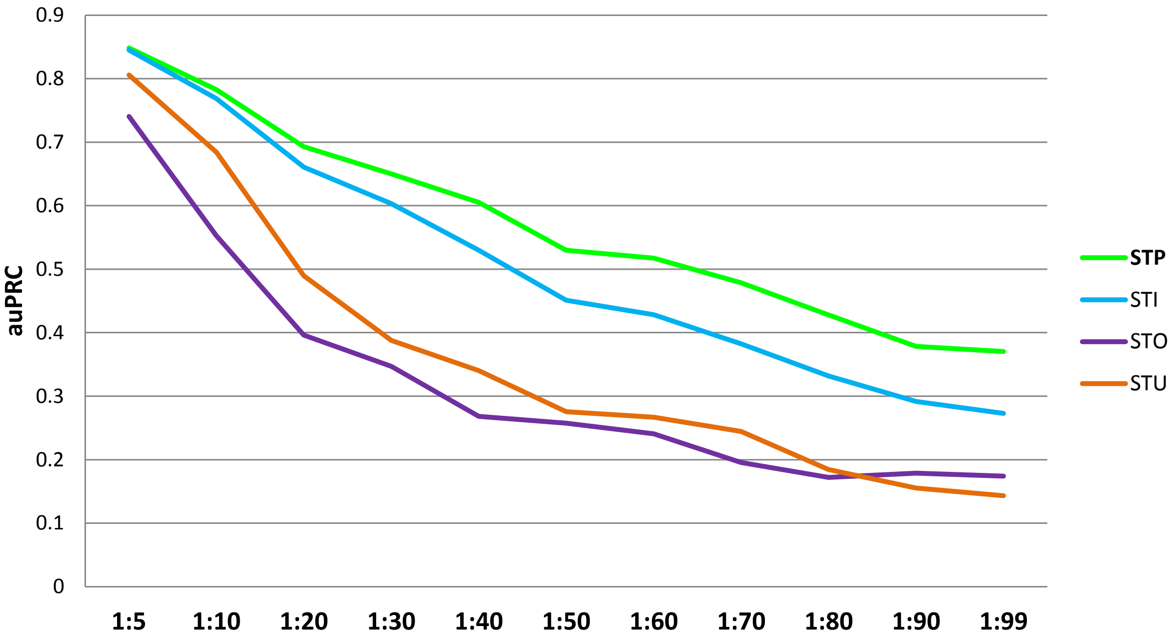


Figure 6.5: **Trends of SSL and data balancing methods for various imbalance degrees when learning from 10% labeled data** Performance (auPRC) variation of self-training based on Naïve Bayes from increasingly high imbalance degrees: Self-training Imbalanced (STI), Self-Training with Positives (STP), Self-training from Under-sampled (STU) and Self-training from Over-sampled (STO).

As opposed to single semi-supervised classifiers (Chapter 3), the ensemble-based approaches achieve satisfactory performance from as little as 1% labeled data for the problem of splice site prediction. We present some interesting trends from our empirical analysis of Chapter 4 in Figure 6.6. The performance of the supervised ensemble baseline is not dropping with increasing data imbalance as the semi-supervised ensembles (or the single semi-supervised classifiers). The "dynamic balancing" has proved useful for ensemble classifiers as well, and in addition, distributing the instances in order to maintain the diversity of the constituent sub-classifiers is also a useful approach. Similarly to our findings from Chapter 2, the ensembles based on Co-training (CoT) are slightly outperforming the Self-training (ST) ensembles.

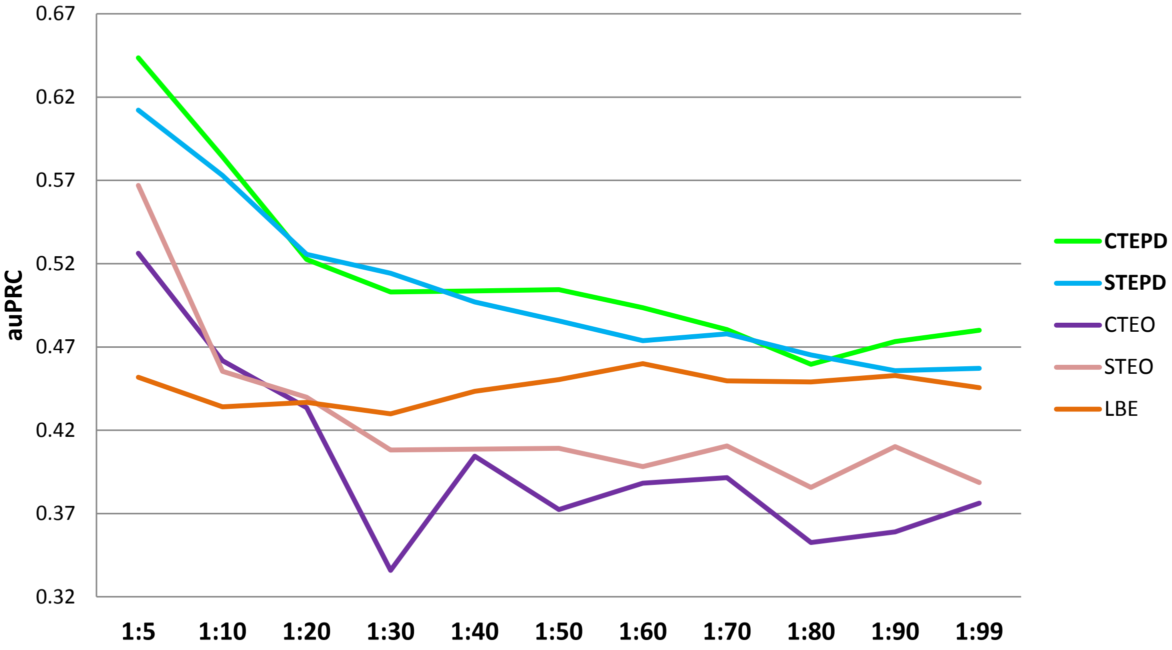


Figure 6.6: **Trends of ensemble-based SSL for various imbalance degrees when learning from 1% labeled data** Performance (auPRC) variation of ensemble-based semi-supervised classifiers from increasingly high imbalance degrees. CTEPD and STEPD are our proposed ensemble variants built from positive distributed pseudo-labeled data obtained during the semi-supervised iterations. The approaches inspired from the original approach are denoted STEO and CTEO, and the ensemble based supervised lower bound is LBE.

We propose several future research directions for the two problems explored in our work, cassette exon identification and acceptor splice site prediction.

Transductive learning, especially the graph-based Modified Adsorption algorithm, has not been used much for bioinformatics problems. Given that it is a robust algorithm (with a well defined optimization function) designed for "big data", its applicability to other problems is worth studying. A potential study could focus on the comparison of transductive learning approaches and unsupervised feature generation, when biologically relevant features are not easily available, to explore various representations and their compatibility with these algorithms.

Graph-based methods, such as Modified Adsorption and Label Propagation, applied to splice site prediction can provide new insights into the behavior of transductive algorithms in the presence of highly imbalanced distributions, as well as the algorithms' compatibility with various representations of DNA sequences.

Ensemble-based semi-supervised learning might benefit from different base-classifiers, or techniques such as stacking or ensemble-selection.

Glossary

6SK 6-Mer Spectrum Kernel. 102, 103

auPRC Area under Precision Recall curve. 42, 43, 45, 54, 58, 79

auROC Area under Receiver Operating Characteristic curve. 21, 22, 25, 26, 30, 31, 42, 79, 101–103, 106, 107

CoT Co-training. 13, 18, 24–26, 28–30, 111

CTEO Co-training ensemble inspired by the original approach. 71–73, 80–82

CTEOD Co-training ensemble inspired by the original approach and distributed. 74, 81–84

CTEP Co-training ensemble with positive. 73, 75, 81, 82

CTEPD Co-training ensemble with positive distributed. 75, 81, 82, 84

DNA Deoxyribonucleic acid. 1, 4, 5, 7, 10, 11, 16, 19, 20, 31, 35, 38, 40, 56, 57, 59, 60, 66, 88, 89, 91, 97–100, 102, 104, 105, 113

EM Expectation Maximization. 11–13, 16–18, 24–26, 28–31

EMW Expectation Maximization with weighted instances. 24–26, 28, 29

ESE Exonic Splicing Enhancers. 20, 100

EST Expressed sequence tags. 11, 15, 20

IRS Intronic Regulatory Sequences. 20, 100

LB Lower Bound. 24–26, 28–30

LBI Lower bound from imbalanced. 43, 45, 48, 49, 52

LBO Lower bound from over-sampling data. 43, 45

LBU Lower bound from under-sampling. 43, 45

LG Length features. 97, 102

LK Linear Kernel. 102

LP Label Propagation. x, 87–89, 91–93, 95, 96, 100, 102–105

MAD Modified Adsorption. x, 87–89, 91, 92, 95, 96, 100, 102–105, 108

MSK Motif Spectrum Kernel. xi, 99, 102–104

NBM Naïve Bayes Multinomial. 17, 19, 24, 25, 29–31

RF Random Forest. 16, 19, 24, 28, 30, 31

RNA Ribonucleic acid. 4, 16

SK Spectrum Kernel. x, 99

SL Supervised learning. 2, 9

SMOTE Synthetic Minority Over-sampling Technique. 38, 39, 45, 48, 49, 56, 58

SSL Semi-supervised learning. 3, 10, 11, 14, 16–19, 21, 23, 25, 26, 28–31, 86, 91

ST Self-training. 13, 24–26, 28–31, 111

STEO Self-training ensemble inspired by the original approach. 71–74, 80–82

STEOD Self-training ensemble inspired by the original approach and distributed. 74, 81, 82

STEP Self-training ensemble with positive. 73, 75, 81, 82

STEPD Self-training ensemble with positive distributed. 75, 81, 82

STI Self-training from imbalanced data. 37–39, 47–49, 52, 54, 57

STO Self-training from imbalanced data. 38, 39

STP Self-training with positive instances. 39, 47–50, 109

STU Self-training from under-sampling. 38, 39

SVM Support Vector Machines. 14, 16, 19, 22, 24, 26, 30, 31, 55, 56, 65, 66, 68, 84, 92, 99, 107

TL Transductive Learning. 4

TSVM Transductive Support Vector Machines. x, 57, 87–92, 96, 100, 102–105, 108

UB Upper Bound. 24, 28

WDS Weighted Degree Kernel with Shifts. x, 98, 99, 101–104, 108

Bibliography

Baldi, P. and Brunak, S. (2001). *Bioinformatics: the machine learning approach*. MIT press.

Baluja, S., Seth, R., Sivakumar, D., Jing, Y., Yagnik, J., Kumar, S., Ravichandran, D., and Aly, M. (2008). Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceedings of the Seventeenth International Conference on World Wide Web*, pages 895–904. ACM.

Baten, A., Halgamuge, S., Chang, B., and Wickramarachchi, N. (2007). Biological sequence data preprocessing for classification: A case study in splice site identification. In Liu, D., Fei, S., Hou, Z., Zhang, H., and Sun, C., editors, *Advances in Neural Networks – ISNN 2007*, volume 4492 of *Lecture Notes in Computer Science*, pages 1221–1230. Springer Berlin Heidelberg, Berlin Heidelberg.

Baten, A. K., Chang, B. C., Halgamuge, S. K., and Li, J. (2006). Splice site identification using probabilistic parameters and SVM classification. *BMC bioinformatics*, 7(Suppl 5):S15.

Batuwita, R. and Palade, V. (2010). Efficient resampling methods for training support vector machines with imbalanced datasets. In *IJCNN*, pages 1–8.

Batuwita, R. and Palade, V. (2012). Adjusted geometric-mean: a novel performance measure for imbalanced bioinformatics datasets learning. *Journal of bioinformatics and computational biology*, 10(04).

Ben-David, S., Lu, T., and Pál, D. (2008). Does unlabeled data provably help? worst-case analysis of the sample complexity of semi-supervised learning. In *COLT*, pages 33–44.

Ben-Hur, A., Ong, C. S., Sonnenburg, S., Scholkopf, B., and Rätsch, G. (2008). Support vector machines and kernels for computational biology. *PLoS computational biology*, 4(10):e1000173.

Berget, S. M., Moore, C., and Sharp, P. A. (1977). Spliced segments at the 5'terminus of adenovirus 2 late mRNA. *Proceedings of the National Academy of Sciences*, 74(8):3171–3175.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 92–100, New York, NY, USA. ACM.

Bonizzoni, P., Rizzi, R., and Pesole, G. (2005). ASPIC: a novel method to predict the exon-intron structure of a gene that is optimally compatible to a set of transcript sequences. *BMC Bioinformatics*, 6(1):1–16.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Castelo, R. and Guigó, R. (2004). Splice site identification by idlBNs. *Bioinformatics*, 20(suppl 1):i69–i76.

Catal, C. and Diri, B. (2009). Unlabelled extra data do not always mean extra performance for semi-supervised fault prediction. *Expert Systems*, 26(5):458–471.

Cerulo, L., Elkan, C., and Ceccarelli, M. (2010). Learning gene regulatory networks from only positive and unlabeled data. *BMC Bioinformatics*, 11(1):228+.

Chapelle, O., Schölkopf, B., Zien, A., et al. (2006). *Semi-supervised learning*, volume 2. MIT press Cambridge.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique". *Journal of Artificial Intelligence Research*, 16(1):321–357.

Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, 6(1):1–6.

Chawla, N. V. and Karakoulas, G. I. (2005). Learning from labeled and unlabeled data: An empirical study across techniques and domains. *J. Artif. Intell. Res.(JAIR)*, 23:331–366.

Chen, Y. (2008). Learning classifiers from imbalanced, only positive and unlabeled data sets. project report for uc san diego data mining contest. *Computer Science, Iowa State University, Ames, IA*.

Chow, L., Gelinas, R., Broker, T., and Roberts, R. (1977). An amazing sequence arrangement at the 5' ends of adenovirus 2 messenger RNA. *Cell*, 12(1):1–8.

Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110.

Craig, R. A. and Liao, L. (2007). Transductive learning with em algorithm to classify proteins based on phylogenetic profiles. *Int. J. Data Mining and Bioinformatics*, 1(4):337–351.

Crick, F. et al. (1970). Central dogma of molecular biology. *Nature*, 227(5258):561–563.

Dai, W., Xue, G., Yang, Q., and Yu, Y. (2007). Transferring Naïve Bayes classifiers for text classification. In *Proceedings of the Twenty-second AAAI Conference on Artificial Intelligence*, pages 540–545.

Davis, J. and Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In *Proceedings of The Twenty Third International Conference on Machine Learning*, pages 233–240. ACM.

De Baets, L. (2014). Identifying novel neuroblastoma oncogenes using machine learning.

Master's thesis, Department of Information Technology, Faculty of Engineering and Architecture, Universiteit Gent.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Deng, N. and Zhu, D. (2014). dsplicetype: A multivariate model for detecting various types of differential splicing events using rna-seq. In *Bioinformatics Research and Applications*, pages 322–333. Springer.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923.

Dong, A. and Bhanu, B. (2003). A new semi-supervised EM algorithm for image retrieval. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–662–II–667 vol.2.

Driessens, K., Reutemann, P., Pfahringer, B., and Leschi, C. (2006). Using weighted nearest neighbor to benefit from unlabeled data. In Ng, W.-K., Kitsuregawa, M., Li, J., and Chang, K., editors, *Advances in Knowledge Discovery and Data Mining*, volume 3918 of *Lecture Notes in Computer Science*, pages 60–69. Springer Berlin Heidelberg, Berlin Heidelberg.

Dror, G., Sorek, R., and Shamir, R. (2005). Accurate identification of alternatively spliced exons using support vector machine. *Bioinformatics*, 21(7):897–901.

Druck, G., Pal, C., McCallum, A., and Zhu, X. (2007). Semi-supervised classification with hybrid generative/discriminative methods. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–289. ACM.

Drummond, C., Holte, R. C., et al. (2003). C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11. Citeseer.

Eichner, J., Zeller, G., Laubinger, S., and Rätsch, G. (2011). Support vector machines-based identification of alternative splicing in *Arabidopsis thaliana* from whole-genome tiling arrays. *BMC bioinformatics*, 12(1):55.

Erdoğdu, U., Tan, M., Alhajj, R., Polat, F., Rokne, J., and Demetrick, D. (2013). Integrating machine learning techniques into robust data enrichment approach and its application to gene expression data. *Int. J. Data Mining and Bioinformatics*, 8(3):247–281.

Ernst, J., Beg, Q. K., Kay, K. A., Balázsi, G., Oltvai, Z. N., and Bar-Joseph, Z. (2008). A semi-supervised method for predicting transcription factor - gene interactions in *escherichiacoli*. *PLoS Comput Biol*, 4(3):e1000044.

Estabrooks, A., Jo, T., and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36.

Fairbrother, W. G., Yeh, R.-F., Sharp, P. A., and Burge, C. B. (2002). Predictive identification of exonic splicing enhancers in human genes. *Science*, 297(5583):1007–1013.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., and Herrera, F. (2012). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(4):463–484.

García-Pedrajas, N., Pérez-Rodríguez, J., García-Pedrajas, M., Ortiz-Boyer, D., and Fyfe, C. (2012). Class imbalance methods for translation initiation site recognition in DNA sequences. *Knowledge-Based Systems*, 25(1):22–34.

Goldberg, A. and Zhu, X. (2006). 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*, TextGraphs-1, pages 45–52.

Guo, Y., Zhang, H., and Spencer, B. (2012). Cost-sensitive self-training. In Kosseim, L. and Inkpen, D., editors, *Advances in Artificial Intelligence*, volume 7310 of *Lecture Notes in Computer Science*, pages 74–84. Springer Berlin Heidelberg.

Gupta, R. and Ratinov, L. (2008). Text categorization with knowledge transfer from heterogeneous data sources. In *Proceedings of the Twenty-third National Conference on Artificial intelligence - Volume 2*, pages 842–847.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Herndon, N. and Caragea, D. (2014). Empirical study of domain adaptation with naïve Bayes on the task of splice site prediction. In *Proceedings of The Fifth International Conference on Bioinformatics Models, Methods and Algorithms*.

Huang, J. (2013). An ensemble learning approach for prediction of phosphorylation sites. *Int. J. Bioinformatics Research and Applications*, 9(3):271–284.

Huang, J. and Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17:299–310.

Jeni, L., Cohn, J., and de la Torre, F. (2013). Facing imbalanced data–recommendations for the use of performance metrics. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 245–251.

Jiang, J. Q. and McQuay, L. J. (2012). Predicting protein function by multi-label correlated semi-supervised learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(4):1059–1069.

Jiang, Q., Wang, G., Jin, S., Li, Y., and Wang, Y. (2013). Predicting human microRNA-disease associations based on Support Vector Machine. *Int. J. Data Mining and Bioinformatics*, 8(3):282–293.

Joachims, T. (1999). Svmlight: Support vector machine. *SVM-Light Support Vector Machine http://svmlight.joachims.org, University of Dortmund*, 19(4).

Kabat, J. L., Barberan-Soler, S., McKenna, P., Clawson, H., Farrer, T., and Zahler, A. M. (2006). Intronic alternative splicing regulators identified by comparative genomics in nematodes. *PLoS computational biology*, 2(7):e86.

Kall, L., Canterbury, J. D., Weston, J., Noble, W. S., and MacCoss, M. J. (2007). Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature Methods*, 4(11):923–925.

Kasabov, N. and Pang, S. (2003). Transductive support vector machines and applications in bioinformatics for promoter recognition. In *Neural Networks and Signal Processing. Proceedings of the 2003 International Conference on*, volume 1, pages 1–6.

Keren, H., Lev-Maor, G., and Ast, G. (2010). Alternative splicing and evolution: diversification, exon definition and function. *Nature reviews.*, 11(5):345–355.

Kim, J. K. and Choi, S. (2011). Probabilistic models for semisupervised discriminative motif discovery in DNA sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(5):1309–1317.

Kim, P., Kim, N., Lee, Y., Kim, B., Shin, Y., and Lee, S. (2005). ECgene: genome annotation for alternative splicing. *Nucleic Acids Research*, 33(suppl 1):D75–D79.

Kirchhoff, K. and Alexandrescu, A. (2011). Phonetic classification using controlled random walks. In *INTERSPEECH*, pages 2389–2392.

Kiritchenko, S. and Matwin, S. (2011). Email classification with co-training. In *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*, pages 301–312. IBM Corp.

Kondratovich, E., Baskin, I. I., and Varnek, A. (2013). Transd. SVM: Promising Approach to Model Small and Unbalanced Datasets. *Molecular Informatics*, 32(3):261–266.

Korecki, J. N., Banfield, R. E., Hall, L. O., Bowyer, K. W., and Kegelmeyer, W. P. (2008). Semi-supervised learning on large complex simulations. In *Proceedings of The Nineteenth International Conference on Pattern Recognition, ICPR 2008*, pages 1–4. IEEE.

Kroll, J. E., Souza, J. E. d., Stransky, B., Souza, G. A. d., and Souza, S. J. d. (2012). Integrating transcriptome and proteome information for the analysis of alternative splicing. In *Proceedings of the 2012 IEEE Second International Conference on Healthcare Informatics, Imaging and Systems Biology*, HISB '12, pages 119–, Washington, DC, USA. IEEE Computer Society.

Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., Freund, Y., and Leslie, C. (2005). Profile-based string kernels for remote homology detection and motif extraction. *Journal of bioinformatics and computational biology*, 3(03):527–550.

Kundu, K., Costa, F., Huber, M., Reth, M., and Backofen, R. (2013). Semi-supervised prediction of SH2-peptide interactions from imbalanced high-throughput data. *PLoS One*, 8(5):e62732.

Lawrence, C. E. and Reilly, A. A. (1990). An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, 7(1):41–51.

Le, T.-B. and Kim, S.-W. (2014). On incrementally using a small portion of strong unlabeled data for semi-supervised learning algorithms. *Pattern Recognition Letters*, 41:53–64.

LeGault, L. H. and Dewey, C. N. (2013). Inference of alternative splicing from RNA-Seq data with probabilistic splice graphs. *Bioinformatics*, 29(18).

Lei, S. and Aidong, Z. (2010). Semi-supervised learning protein complexes from protein interaction networks. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 247–252.

Leslie, C. S., Eskin, E., and Noble, W. S. (2002). The spectrum kernel: A string kernel for SVM protein classif. In *Pacific Symposium on Biocomputing*, volume 7, pages 566–575.

Li, F., Li, G., Yang, N., Xia, F., and Yu, C. (2013). Label matrix normalization for semi-supervised learning from imbalanced data. *New Review of Hypermedia and Multimedia.*

Li, J., Wang, L., Wang, H., Bai, L., and Yuan, Z. (2012a). High-accuracy splice site prediction based on sequence component and position features. *Genetics and Molecular Research*, 11(3):3432–3451.

Li, M. and Zhou, Z.-H. (2007). Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man, and Cybernetics*, 37(6):1088–1098.

Li, S., Ju, S., Zhou, G., and Li, X. (2012b). Active learning for imbalanced sentiment classification. In *Proceedings of The 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 139–148. Association for Computational Linguistics.

Li, S., Wang, Z., Zhou, G., and Lee, S. Y. M. (2011). Semi-supervised learning for imbalanced sentiment classification. In *Proceedings of The Twenty Second International*

Joint Conference on Artificial Intelligence-Volume Volume Three, pages 1826–1831. AAAI Press.

Li, T., Zhu, S., Li, Q., and Ogihara, M. (2003). Gene functional classification by semi-supervised learning from heterogeneous data. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, SAC '03, pages 78–82, New York, NY, USA.

Li, Y.-f. and Zhou, Z.-h. (2011). Towards making unlabeled data never hurt. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 1081–1088.

Li, Y.-F. and Zhou, Z.-H. (2015). Towards making unlabeled data never hurt. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):175–188.

Ling, C. X. and Sheng, V. S. (2008). Cost-sensitive learning and the class imbalance problem. *Encyclopedia of Machine Learning.*

Liu, S., Chen, Y., and Wilkins, D. (2012). Large margin classifiers and random forests for integrated biological prediction. *I. J. Bioinformatics Research and Applications*, 8(1):38–53.

Liu, X.-Y., Wu, J., and Zhou, Z.-H. (2009). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2):539–550.

Liu, Y. and Kirchhoff, K. (2013). Graph-based semi-supervised learning for phone and segment classification. In *INTERSPEECH*, pages 1840–1843.

Loc, T. (2012). Application of three graph laplacian based semi-supervised learning methods to protein function prediction problem. *CoRR*, abs/1211.4289.

Lomsadze, A., Burns, P. D., and Borodovsky, M. (2014). Integration of mapped rna-seq reads into automatic training of eukaryotic gene finding algorithm. *Nucleic acids research*, page gku557.

Lu, H., Lin, L., Sato, S., Xing, Y., and Lee, C. J. (2009). Predicting functional alternative splicing by measuring rna selection pressure from multigenome alignments. *PLoS Computational Biology*, 5:e1000608.

Lusa, L. and Blagus, R. (2010). Class prediction for high-dimensional class-imbalanced data. *BMC bioinformatics*, 11(1):523.

McCallum, A. and Nigam, K. (1998). A comparison of event models for Naïve Bayes text classification. In *In AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48. AAAI Press.

Mitchell, T. (1997). *Machine Learning (Mcgraw-Hill International Edit).* McGraw-Hill Education (ISE Editions), 1st edition.

Moreno, P. J. and Agarwal, S. (2003). An experimental study of semi-supervised EM. Technical report, HP Labs.

Nagaraj, S. H., Gasser, R. B., and Ranganathan, S. (2007). A hitchhiker's guide to expressed sequence tag (EST) analysis. *Briefings in Bioinformatics*, 8(1):6–21.

Nesvizhskii, A. I., Keller, A., Kolker, E., and Aebersold, R. (2003). A statistical model for identifying proteins by tandem mass spectrometry. *Analytical Chemistry*, 75(17):4646–58.

Nguyen, T.-P. and Ho, T.-B. (2012). Detecting disease genes based on semi-supervised learning and protein-protein interaction networks. *Artificial Intelligence in Medicine*, 54(1):63–71.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Mach. Learn.*, 39(2-3):103–134.

Nigam, K. and Rayid, G. (2000). Analyzing the effectiveness and applicability of co-training. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, CIKM '00, pages 86–93.

Niu, Z.-Y., Ji, D.-H., and Tan, C. L. (2005). Word sense disambiguation using label propagation based semi-supervised learning. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05.

Pacharawongsakda, E. and Theeramunkong, T. (2013). Predict subcellular locations of singleplex and multiplex proteins by semi-supervised learning and dimension-reducing general mode of Chou's PseAAC. *NanoBioscience, IEEE Transactions on*, 12(4):311–320.

Pang, S. and Kasabov, N. (2004). Inductive vs transductive inference, global vs local models: SVM, TSVM, and SVMT for gene expression classification problems. In *International Joint Conference on Neural Networks*, volume 2, pages 1197–1202. IEEE.

Pervouchine, D. D., Knowles, D. G., and Guigó, R. (2013). Intron-centric estimation of alternative splicing from RNA-seq data. *Bioinformatics*, 29(2):273–274.

Provost, F. J., Fawcett, T., and Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the 15th International Conference on Machine Learning*, ICML '98. Morgan Kaufmann Publishers Inc.

Qi, Y., Tastan, O., Carbonell, J. G., Klein-Seetharaman, J., and Weston, J. (2010). Semi-supervised multi-task learning for predicting interactions between HIV-1 and human proteins. *Bioinformatics*, 26(18):i645–i652.

Rangwala, H. and Karypis, G. (2005). Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, 21(23):4239–4247.

Rätsch, G., Sonnenburg, S., and Schölkopf, B. (2005). RASE: recognition of alternatively spliced exons in *c. elegans*. In *Proceedings of 13th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, volume 21, pages 369–377.

Ren, Y., Kaji, N., Yoshinaga, N., and Kitsuregawa, M. (2014). Sentiment classification in under-resourced languages using graph-based semi-supervised learning methods. *IEICE Transactions on Information and Systems*, 97(4):790–797.

Rider, A. K., Siwo, G., Emrich, S. J., Ferdig, M. T., and Chawla, N. V. (2014). A supervised learning approach to the ensemble clustering of genes. *Int. J. Data Mining and Bioinformatics*, pages 199–219.

Rosenberg, C., Hebert, M., and Schneiderman, H. (2005). Semi-supervised self-training of object detection models. In *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision WACV/MOTIONS.*, volume 1, pages 29–36. IEEE.

Sacomoto, G., Kielbassa, J., Chikhi, R., Uricaru, R., Antoniou, P., Sagot, M. F., Peterlongo, P., and Lacroix, V. (2012). KISSPLICE: de-novo calling alternative splicing events from RNA-seq data. *BMC Bioinformatics*, 13(Suppl 6):S5+.

Sacomoto, G., Lacroix, V., and Sagot, M.-F. (2013). A polynomial delay algorithm for the enumeration of bubbles with length constraints in directed graphs and its application to the detection of alternative splicing in RNA-seq data. In *Algorithms in Bioinformatics*, pages 99–111. Springer.

Schmucker, D., Clemens, J. C., Shu, H., Worby, C. A., Xiao, J., Muda, M., Dixon, J. E., and Zipursky, S. L. (2000). *Drosophila* Dscam is an axon guidance receptor exhibiting extraordinary molecular diversity. *Cell*, 101(6):671–684.

Schweikert, G., Widmer, C., Schölkopf, B., and Rätsch, G. (2008). An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In *Advances in Neural Information Processing Systems 21*, volume 8, pages 1433–1440.

Shi, M. and Zhang, B. (2011). Semi-supervised learning improves gene expression-based prediction of cancer recurrence. *Bioinformatics*, 27(21):3017–3023.

Shin, H., Tsuda, K., and Schölkopf, B. (2009). Protein functional class prediction with a combined graph. *Expert Systems with Applications*, 36(2):3284–3292.

Singh, A., Nowak, R., and Zhu, X. (2009). Unlabeled data: Now it helps, now it doesn't. In *Advances in Neural Information Processing Systems*, pages 1513–1520.

Sonnenburg, S., Schweikert, G., Philips, P., Behr, J., and Rätsch, G. (2007). Accurate splice site prediction using support vector machines. *BMC Bioinformatics*, 8(Suppl 10):1–16.

Stanescu, A. and Caragea, D. (2012). Semi-supervised learning of alternatively spliced exons using Expectation Maximization type approaches. In *Proceedings of The Third International Conference on Bioinformatics Models, Methods and Algorithms*, pages 240–245.

Stanescu, A. and Caragea, D. (2014a). Semi-supervised self-training approaches for imbalanced splice site datasets. In *Proceedings of The Sixth International Conference on Bioinformatics and Computational Biology, BICoB 2014*, pages 131–136.

Stanescu, A. and Caragea, D. (2014b). Ensemble-based semi-supervised learning approaches for imbalanced splice site datasets. In *Proceedings of the Sixth IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2014*, pages 432–437.

Stanescu, A. and Caragea, D. (2015a). An empirical study of ensemble-based semi-supervised learning approaches for imbalanced splice site datasets. *In press: BMC Systems Biology.*

Stanescu, A. and Caragea, D. (2015b). An empirical study of self-training and data balancing techniques for splice site prediction. *Under review: International Journal of Bioinformatics Research and Applications.*

Stanescu, A. and Caragea, D. (2015c). Predicting cassette exons using transductive learning approaches. In *2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*.

Stanescu, A., Tangirala, K., and Caragea, D. (2015). Predicting alternatively spliced exons using semi-supervised learning. *In press: International Journal of Data Mining and Bioinformatics*.

Talukdar, P. P. and Crammer, K. (2009). New regularized algorithms for transductive learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 442–457. Springer Berlin Heidelberg.

Tangirala, K. and Caragea, D. (2011). Semi-supervised learning of alternatively spliced exons using co-training. In *Procesdings of The 2011 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 243–246.

Teng, L. and Tan, K. (2012). Finding combinatorial histone code by semi-supervised biclustering. *BMC Genomics*, 13(1):1–11.

Vapnik, V. N. and Vapnik, V. (1998). *Statistical learning theory*, volume 2. Wiley New York.

Wang, J. T. L. and Wu, X. (2006). Kernel design for RNA classification using Support Vector Machines. *Int. J. Data Mining and Bioinformatics*, 1(1):57–76.

Wang, L., Chan, K. L., and Zhang, Z. (2003). Bootstrapping svm active learning by incorporating unlabelled images for image retrieval. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–629–I–634 vol.1.

Wang, X., Wang, K., Radovich, M., Wang, Y., Wang, G., Feng, W., Sanford, J. R., and Liu, Y. (2009). Genome-wide prediction of cis-acting rna elements regulating tissue-specific pre-mrna alternative splicing. *BMC genomics*, 10(Suppl 1):S4.

Wang, Y. and Chen, S. (2013). Safety-aware semi-supervised classification. *Neural Networks and Learning Systems, IEEE Transactions on*, 24(11):1763–1772.

Wei, Q. and Dunbrack Jr, R. L. (2013). The role of balanced training and testing data sets for binary classifiers in bioinformatics. *PLoS One*, 8(7):e67863.

Weston, J., Kuang, R., Leslie, C., and Noble, W. S. (2006). Protein ranking by semi-supervised network propagation. *BMC Bioinformatics*, 7(Suppl 1):1–9.

Weston, J., Leslie, C., Ie, E., Zhou, D., Elisseeff, A., and Noble, W. S. (2005). Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247.

Whalen, S. and Pandey, G. (2013). A comparative analysis of ensemble classifiers: case studies in genomics. In *The IEEE 13th International Conference on Data Mining (ICDM*, pages 807–816. IEEE.

Wu, Q., Wang, Z., Li, C., Ye, Y., Li, Y., and Sun, N. (2015). Protein functional properties prediction in sparsely-label ppi networks through regularized non-negative matrix factorization. *BMC systems biology*, 9(Suppl 1):S9.

Wu, T. D. and Watanabe, C. K. (2005). GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics*, 21(9):1859–1875.

Xia, J., Caragea, D., and Brown, S. J. (2010). Prediction of alternatively spliced exons using support vector machines. *I. J. Data Mining and Bioinformatics*, 4(4):411–430.

Xu, B., Wei, X., Deng, L., Guan, J., and Zhou, S. (2012). A semi-supervised boosting svm for predicting hot spots at protein-protein interfaces. *BMC systems biology*, 6(Suppl 2):S6.

Xu, Q., Hu, D. H., Xue, H., Yu, W., and Yang, Q. (2009). Semi-supervised protein subcellular localization. *BMC Bioinformatics*, 10(Suppl 1):S47.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of The Thirty Third Annual Meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics.

Yeo, G. W., Van Nostrand, E., Holste, D., Poggio, T., and Burge, C. B. (2005). Identification and analysis of alternative splicing events conserved in human and mouse. *Proceedings of the National Academy of Sciences of the United States of America*, 102(8):2850–2855.

Yip, K. Y., Cheung, L., Cheung, D. W., Jing, L., and Ng, M. K. (2009). A semi-supervised approach to projected clustering with applications to microarray data. *Int. J. of Data Mining and Bioinformatics*, 3(3):229–259.

You, M., Zhao, R.-W., Li, G.-Z., and Hu, X. (2011). MAPLSC: A novel multi-class classifier for medical diagnosis. *Int. J. Data Mining and Bioinformatics*, 5(4):383–401.

You, Z., Yin, Z., Han, K., Huang, D.-S., and Zhou, X. (2010). A semi-supervised learning approach to predict synthetic genetic interactions by combining functional and topological properties of functional gene network. *BMC Bioinformatics*, 11(1):343.

Yu, Y., Wang, X., Lin, L., Sun, C., and Wang, X. (2013). A supervised approach to detect protein complex by combining biological and topological properties. *Int. J. Data Mining and Bioinformatics*, 8(1):105–121.

Zhou, A., Breese, M., Hao, Y., Edenberg, H., Li, L., Skaar, T., and Liu, Y. (2012). Alt event finder: a tool for extracting alternative splicing events from RNA-seq data. *BMC Genomics*, 13(8):S10+.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. *Advances in neural information processing systems*, 16(16):321–328.

Zhou, Z.-H. and Li, M. (2010). Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24(3):415–439.

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02-107, Carnegie Mellon University.

Zhu, X., Ghahramani, Z., Lafferty, J., et al. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919.