NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

POSTGRADUATE PROGRAM
"INFORMATION TECHNOLOGIES IN MEDICINE AND BIOLOGY"

MASTER THESIS

Splice Site Prediction Using Transfer Learning

Simos I. Kazantzidis

Supervisors: **Stavros Perantonis,** Research Director, NCSR Demokritos
Elias Manolakos, Assoc. Professor, Department of Informatics and Telecommunications, University of Athens
Anastasia Krithara, Associate researcher, NCSR Demokritos

ATHENS

MARCH 2016

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
"ΤΕΧΝΟΛΟΓΙΕΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗΝ ΙΑΤΡΙΚΗ ΚΑΙ ΤΗ ΒΙΟΛΟΓΙΑ"

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πρόβλεψη Θέσεων Ματίσματος με τη χρήση Μεταφοράς Μάθησης

Σίμος Η. Καζαντζίδης

Επιβλέποντες: **Σταύρος Περαντώνης,** Διευθυντής Ερευνών, ΕΚΕΦΕ Δημόκριτος
Ηλίας Μανωλάκος, Αν. Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ
Αναστασία Κριθαρά, Συνεργαζόμενη ερευνήτρια, ΕΚΕΦΕ Δημόκριτος

ΑΘΗΝΑ

ΜΑΡΤΙΟΣ 2016

MASTER THESIS

Splice Site Prediction Using Transfer Learning

Simos I. Kazantzidis
R.N.: PIV0127

SUPERVISORS: **Stavros Perantonis,** Research Director, NCSR Demokritos
Elias Manolakos, Assoc. Professor, Department of Informatics and Telecommunications, University of Athens
Anastasia Krithara, Associate researcher, NCSR Demokritos

EXAMINATION COMMITTEE:
Stavros Perantonis, Research Director, NCSR Demokritos
Elias Manolakos, Assoc. Professor, Department of Informatics and Telecommunications, University of Athens
Anastasia Krithara, Associate researcher, NCSR Demokritos

March 2016

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πρόβλεψη Θέσεων Ματίσματος με τη χρήση Μεταφοράς Μάθησης

Σίμος Η. Καζαντζίδης
Α.Μ.: ΠΙΒ0127

ΕΠΙΒΛΕΠΩΝΤΕΣ: **Σταύρος Περαντώνης,** Διευθυντής Ερευνών, ΕΚΕΦΕ Δημόκριτος
Ηλίας Μανωλάκος, Αν. Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ
Αναστασία Κριθαρά, Συνεργαζόμενη ερευνήτρια, ΕΚΕΦΕ Δημόκριτος

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Σταύρος Περαντώνης,** Διευθυντής Ερευνών, ΕΚΕΦΕ Δημόκριτος
Ηλίας Μανωλάκος, Αν. Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ
Αναστασία Κριθαρά, Συνεργαζόμενη ερευνήτρια, ΕΚΕΦΕ Δημόκριτος

Μάρτιος 2016

ABSTRACT

One of the open problems in the field of bioinformatics, is the automatic gene prediction (nucleotide sequence that encodes proteins). More specifically, researchers are trying to predict those positions that correspond to the beginning and the end of genes within a genome. These positions are known as splice sites. Several machine learning techniques have been used for the specific problem. Nevertheless, the acquisition of annotated data, necessary to implement supervised learning techniques, is a significant challenge, as the cost is very large. One of the approaches for addressing this problem is the transferring of knowledge (transfer learning approach). The aim of this work is the study of the representation of genes in order to take into account the sequence of nucleotides within a genome and the role of this representation in transfer learning methods.

SUBJECT AREA: Splice Site Prediction, Computational Biology

KEYWORDS: transfer learning, splice site, machine learning, n-gram graphs

ΠΕΡΙΛΗΨΗ

Ένα από τα ανοιχτά προβλήματα της βιοπληροφορικής, είναι η αυτόματη πρόβλεψη γονιδίων (αλληλουχία νουκλεοτιδίων που κωδικοποιεί πρωτεΐνες). Πιο συγκεκριμένα, οι ερευνητές προσπαθούν να προβλέψουν τις θέσεις που αντιστοιχούν στην αρχή και το τέλος των γονιδίων σε ένα γονιδίωμα. Οι θέσεις αυτές είναι γνωστές ως σήματα ματίσματος (splice sites). Διάφορες τεχνικές της μηχανικής μάθησης έχουν χρησιμοποιηθεί για το συγκεκριμένο πρόβλημα. Παρόλα αυτά, η απόκτηση των επισημειωμένων δεδομένων που είναι αναγκαία για να εφαρμοστούν οι τεχνικές επιβλεπόμενης μάθησης, αποτελεί μια σημαντική πρόκληση, καθώς το κόστος είναι πολύ μεγάλο. Μία από τις προσεγγίσεις για την αντιμετώπιση αυτού του προβλήματος είναι η μεταφορά μάθησης (transfer learning). Στόχος της παρούσας εργασίας είναι η μελέτη της αναπαράστασης των γονιδίων, ώστε να λαμβάνεται υπόψιν η αλληλουχία των νουκλεοτιδίων σε ένα γονιδίωμα, και ο ρόλος της αναπαράστασης αυτής σε μεθόδους μεταφοράς μάθησης.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Πρόβλεψη Θέσεων Ματίσματος, Υπολογιστική Βιολογία

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: μεταφορά μάθησης, θέσεις ματίσματος, μηχανική μάθηση, γράφοι ν-γραμμάτων

To my family.

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Σταύρο Περαντώνη, που δέχτηκε να συνεργαστούμε για τη διεκπεραίωση της διπλωματικής μου εργασίας. Επίσης, θα ήθελα να ευχαριστήσω τον υπεύθυνό μου κ. Γεώργιο Παλιούρα και την επιβλέπουσα κα. Αναστασία Κριθαρά, για την εποικοδομητική συνεργασία που είχαμε σε επίπεδο διπλωματικής αλλά και για τον χρόνο που μου αφιέρωσαν καθ' όλη τη διάρκεια της συγγραφής.

TABLE OF CONTENTS

TABLE OF IMAGES

LIST OF TABLES

PROLOGUE

As part of the preparation of the master thesis of the postgraduate program “**Information Technologies in Medicine and Biology**”, with direction the Bioinformatics field, in collaboration with the National Research Center of Natural Sciences "Demokritos", an attempt was made to develop techniques to predict splice sites using the transfer learning.

1. INTRODUCTION

The field of Computational Biology and biomedical research offer a variety of applications in **big data analysis**, where the role of machine learning is more than necessary by allowing the modeling of basic mechanisms [3]. Despite the huge success of Data Mining technologies and those of Machine Learning in the fields of classification, regression and clustering, many methods achieve good results under the assumption that the training and test data are issued on the same space and with the same distribution [6].

Making a brief historical overview, one can see that already since 1980 computational biotechnology has contributed in locating exons i.e. gene-coding regions. Many machine learning techniques and approaches have been used in order to find and predict donor and acceptor splice sites as well as the protein's secondary structure [15].

The procedure of splice site prediction is fundamental in the field of gene-finding. More specifically, splice sites are locations on DNA at the boundaries of exons and introns. The more accurately a splice site can be located, the more reliable it becomes to locate the genes on DNA, thus, accurate splice site detectors are significant components of state-of-the art gene finders [16].

Modeling and using biological mechanisms, such as the splice site mechanism, means simultaneously the use of complex models, requiring an equivalent suitable sized training set, which is often not available especially in the biomedical domain due to time and expense [13]. Another issue is the appropriate use of decoys (negative training examples), which can definitely change the detectors performance [4].

Transfer Learning, allows the domains, tasks, and distributions used in training and test data to differ by applying knowledge learned previously. In this way, the transfer learning approach represents a preferred method in cases where data from the same organism and even between different species are limited and no annotated data can be obtained.

In this work, we focus on the problem of splice sites recognition. By now it is fairly well understood and there exist experimentally confirmed labels for a broad range of organisms [13]. Since the content of the protein plays an important role in relation to the splicing prediction, it is a critical task for the identification of genes and their functionality [2]. It is worth noting that different kinds of features are required to get high accuracy and precision in splice site prediction [17].

In order to overcome the lack of training data (annotated data) and avoid overfitting, we apply a transfer learning approach to transfer knowledge from sequences of specific species, namely Homo Sapiens, Rerio, Melanogaster, Elegans and Thaliana, in which splice sites are known. Features are being extracted with the help of the n-gram graph representation and various classification methods are used.

The main contributions of this work are:

- The use of N-gram graphs representation: by representing each DNA sequence as an n-gram graph, we can use the N-gram graphs similarity in order to obtain the first two features of the proposed approach.
- The use biological information: There are a few motifs of great importance in order to discover with high possibility a splice site. Thus, using such biological information combined with the above features can help us achieve higher prediction accuracy.

- Combining the two features categories above, we managed to achieve high performance quickly and with low computational cost, as the proposed features space is small, current approaches use a large number of features (thousands), and as a result, they need too much time to produce results. For instance, our largest dataset (40.000 instances) required less than a day in order to produce results.
- Two transfer learning approaches were proposed, which are based on similarity functions. The main idea is to find the most similar instances between training and test datasets.
- A target data transformation approach: We transformed the initial target data with the help of the mean values of the similar data. The distributions of each feature of the target domain approximated those of the source domain.
- The Multiple source domain approach: We considered the case in which the source domain has more than one species and in both approaches, and we incorporated information from the phylogenetic analysis between species.

2. BACKGROUND

2.1 Splice Sites

The exact identification of genes in eukaryotic genomes depends largely on the ability to accurately determine the splicing sites which are segments of DNA that separate the exons and introns within a gene. Apart from determining the structure of the gene, splice sites also determine the amino acid composition of the proteins encoded by the genes.

The procedure of splice site prediction is fundamental in the field of gene-finding if one considers that the transcription of an eukaryotic DNA sequence into messenger RNA occurs only after enzymes splice away noncoding regions (introns) and leave only coding regions (exons) [17].

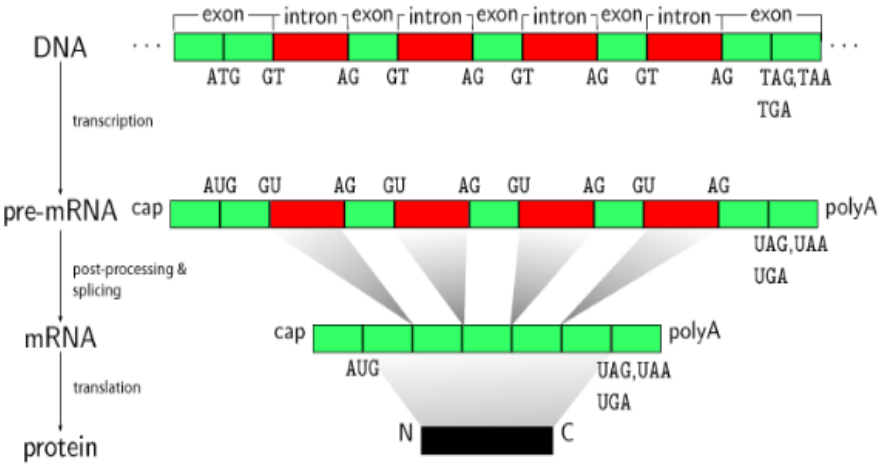

Image 2.1: Basic steps of protein synthesis [1].

As shown in the figure above, the basic steps of protein synthesis are:

1. Transcription
2. Post-processing
3. Translation

In biology, transcription is the first stage of gene expression and describes the process by which an RNA molecule is created using a DNA strand as a template. The term **transcription** is used because the genetic information in the language of DNA, is transcribed in the language of RNA, by using uracil base instead of thymine.

Pre-mRNA is being transformed into mRNA. More precisely, in an eukaryotic gene, the sequence of mRNA consists of non-coding regions, known as **introns**. Genes start with an exon and may then be interrupted by an intron. This order continues alternately until it ends in an exon. In order to obtain mature mRNA the process of splicing is required, in which introns are removed. In this way, two different splice sites arise: the exon-intron boundary, known as the donor site or 5' site (of the intron) and the intron-exon

boundary, that is the acceptor or 3' site. The genome has many consensus sequences. Thus, by choosing a window close to the splice site and taking k-mers[1] one can get the most frequently occurring nucleotide. Having aligned all the sequences, one can notice which nucleotide is appearing more frequently in each position.

As already mentioned, two types of splice sites must be identified: the donor and the acceptor. The donor's splice site indicates where an exon ends and where an intron starts, while the acceptor's splice site indicates the ending of an intron and an exon starting. Almost most of donor sites are a GT dimer and most acceptor sites are an AG dimer. The fact that these dimers are not necessarily splice sites, complicates their detection [2].

Dimers often occur at non splice site positions. In human DNA, GT dimers can be found about 400 million times overall in both strands. For this reason, the discrimination between true donor sites and decoy positions has to be faced [4]. As one can note, splice site prediction is a difficult problem. The AG and GT dimers cannot be used as features due to their frequent appearance in non splice site sequences. Even the use of positional probabilities was rather a fairly poor approach [17].

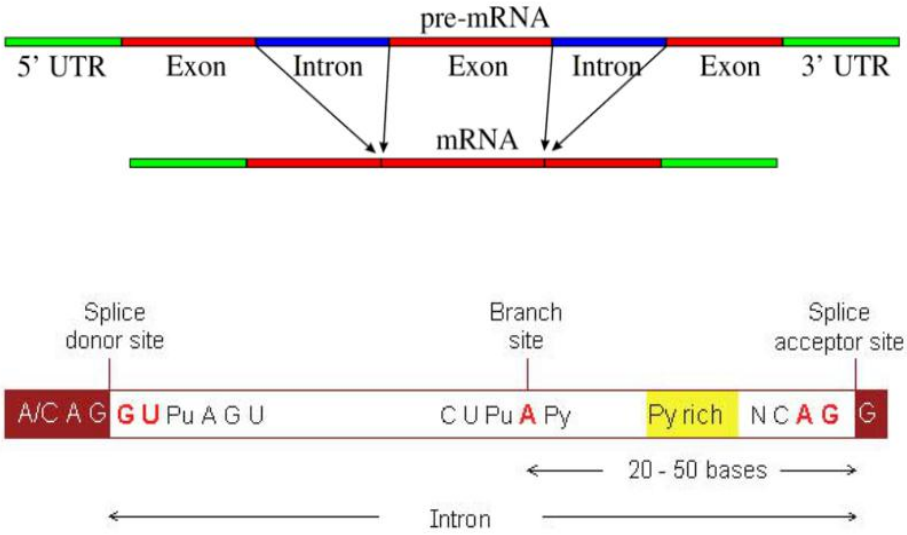

Image 2.2: Splice Site Positions.

There are cases in which splicing occurrences introduce premature termination codons triggering transcript degradation. From the disease-causing nucleotide substitutions that are listed in the Human Gene Mutation Database, a 14% is being thought to cause pre-mRNA splicing disruption, while only 2% of missense mutations disrupt pre-mRNA splicing [8]. Such sequences that disrupt the splicing procedure do not only cause disease but also influence its severity. Identifying mutations that disrupt pre-mRNA splicing becomes gradually an important field in the therapeutic treatment [8].

[1] In computational genomics, k-mers are all the possible subsequences of length k from a read obtained through DNA Sequencing. K-mers are often used in sequence alignments [30].

2.1.1 Splice Site Motifs

Significant scientific work proves that there are a few motifs of great importance in order to discover with high possibility a splice site. Thus, using such biological information has a result to achieve higher prediction accuracy.

Below, are some of the motifs listed [5]:

- Most introns start from the sequence GT. This dimer is a motif for most donor sites and the general motif is “mrgGTrag”.
- Most introns end with the sequence AG. This dimer is a motif for most acceptor sites and the general motif is “yAGr”.
- The branch site is a motif within introns and has the consensus sequence “ynyyrAy” where the position of “A” nucleotide is fully conserved. This motif is being detected 20-50 nucleotides before the acceptor dimer AG.
- The last motif Py (rich-Pyrimidine) is being detected between branch site and acceptor dimer AG. This part of the sequence has high possibility of Cytosine and Thymine appearances.

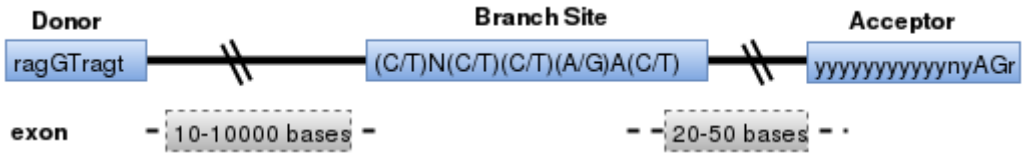

Image 2.3: Pyrimidine-CT, Purine-AG residues, Py-Pyrimidine rich motifs [18].

2.2 Phylogenetic Analysis

Nowadays, Bioinformatics and Computational Biology develop new methods in order to understand various biological processes [9]. One of these studies involve the exploration of evolutionary relationships between living organisms using **phylogenetic analysis**.

Phylogenetic analysis is a method that allows the reporting and evaluation of evolutionary relationships [19].

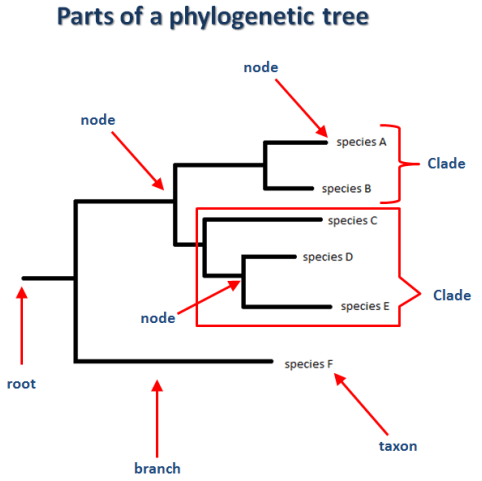

Image 2.4: Phylogenetic tree example [20].

The evolutionary process resulting from the information of phylogenetic analysis typically is displayed by branches and tree diagrams. A simple phylogenetic analysis consists of four stages [19]:

- Alignment - Creation of a data model and exporting phylogenetic dataset.
- Replacement Model Determination - Substitution Model
- Tree Construction
- Tree Evaluation

Each step is crucial for the analysis and should be treated accordingly. For example, trees depend on the alignment they are based on. Therefore, the better the alignment has been made the better the results for the construction of the tree are. When performing a phylogenetic analysis it is enlightening to construct trees based on different modifications of the alignment to see how the proposed alignment affects the resulting tree. Phylogenetic sequence data usually consists of multiple sequence alignments [19]. In the following image the multiple sequence alignment procedure is being presented in the upper part, where three or more biological sequences (protein or nucleic acid) of similar length are being aligned. From the output, homology can be inferred and the evolutionary relationships between the sequences studied [31]. The phylogenetic tree that is being produced is shown below.

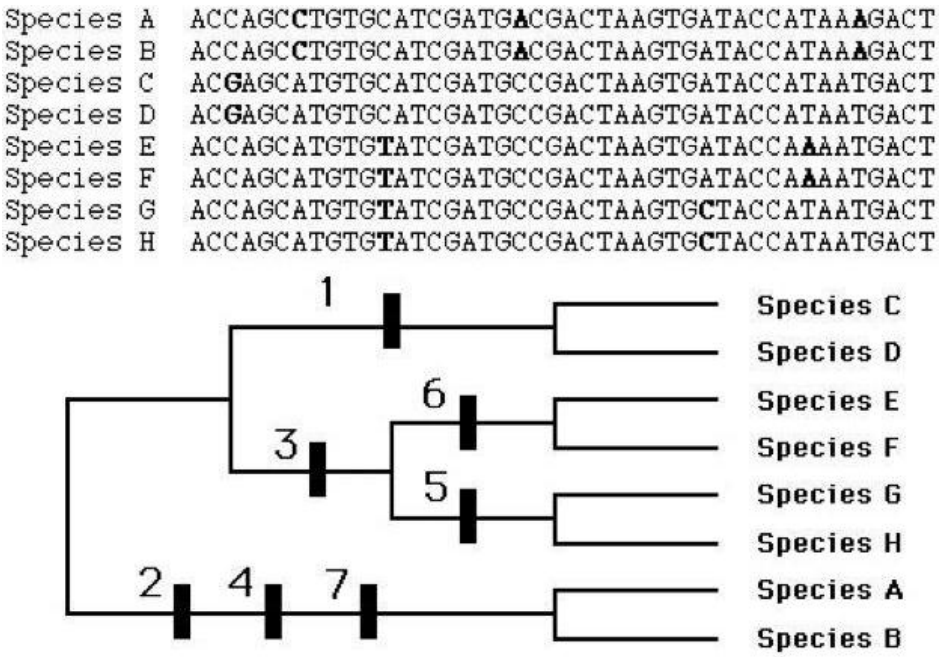

Image 2.5: Phylogenetic Analysis [21].

The use of conserved regions for phylogenetic analysis is also an important issue that one has to consider since incorrect phylogeny can arise due to historically insignificant signals (Axelsen and Palmgren, 1998, Naylor and Brown, 1998) [14].

A typical rule wants closely related organisms to have more common biochemical processes. The difficulty of carrying out biochemical experiments, differentiates from organism to organism, resulting an easier or harder analysis respectively. Biological research aims to understand more such biological complexity. This knowledge can be transferred in turn to other organisms by verifying or refining models.

3. MACHINE LEARNING TECHNIQUES

3.1 Classification Algorithms

In this work, we focused on using different machine learning techniques in order to deal with the splice sites prediction problem. More specifically we used the N-gram graphs representation combined with the biological information of features in order to train a classifier and predict splice sites. At this point two algorithms are being proposed, namely the “Similarity of source and target data algorithm” and the “Target Data Transformation algorithm” concerning the data optimization aiming the classifiers’ better adaptation. Furthermore the CRF probabilistic model has been used in combination with specific patterns and nucleotide motifs. In this way CRF establishes splice site recognition as well.

3.2 Transfer Learning

The field of Machine Learning is a promising field of computer science giving the opportunity to study and construct algorithms with the ability of “learning” and even “predicting” data.

While traditional methods use statistical model strained with previously labeled or unlabeled data assuming the same distribution, the method of Transfer Learning allows diversity in both distributions and domains. It is now possible to use prior knowledge for faster and optimized problem solving [6].

In case of few labeled data **semi-supervised classification** addresses this issue by using a large amount of unlabeled data and a small amount of labeled data. A variety of supervised and semi-supervised cases have been studied in order to deal with imperfect data sets. However most of the times distributions are assumed to be the same for both labeled and unlabeled data, unlike **transfer learning**, where domains and distributions may differ [6].

Since 1995, transfer learning research appears under various names such as learning to learn, life-long learning, knowledge transfer, inductive transfer, multitask learning, knowledge consolidation, context-sensitive learning, knowledge-based inductive bias and others. Between them, **multitask learning framework** learns multiple different tasks simultaneously [6].

On the other hand, Multitask Learning learns both, the source and target tasks simultaneously, while Transfer Learning tries boosting target’s domain’s performance by using the source domain data. In this way, weights of the loss functions are the same for both, source and target data, unlike to the approach of Transfer Learning, where loss functions may differ [6].

Daily, we face a variety of transfer learning situations. For example, we learn recognizing and distinguishing fruits, or playing an organ in order to learn another afterwards. Transfer learning is based in applying such previously learned knowledge in order to solve problems [6].

3.2.1 Transfer Learning Approaches

Three are the main issues in transfer learning one has to deal with [6]:

- **what to transfer:** what part of knowledge can be transferred
- **how to transfer:** algorithms needed in order to transfer knowledge
- **when to transfer:** in which situations transferring should be done

There are three basic approaches of Transfer Learning [3]:

- **Inductive Transfer Learning**

The target task is different from the source task requiring some labeled data inducing an objective predictive model. According to the labeled and unlabeled data in the source domain, two categories are further distinguished [3]:

- There are labeled data available in the source domain. Note that in this case, inductive transfer learning is similar to the multitask learning [6].
- No labeled data are available in the source domain. Note that in this case, inductive transfer learning is similar to the self-taught learning, where the label spaces between the source and target domains may differ, suggesting that the side information of the source domain cannot be used directly [6].

Following is the definition of Inductive Transfer Learning:

"Given a source domain D_S and a learning task T_S, a target domain D_T and a learning task T_T, inductive transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $T_S \neq T_T$" [6].

- **Transductive Transfer Learning**

Although the meaning of "transductive" in the traditional machine learning approach, refers to the situation where all test data are required to be seen at training time, and that the learned model cannot be reused for future data, our work will sink to the definition based on the report of [6], were "transductive learning" is described the situation where the tasks must be the same and all target domain data are unlabeled[6].

Following is the definition of Transductive Transfer Learning:

"Given a source domain D_S and a corresponding learning task T_S, a target domain D_T and a corresponding learning task T_T, transductive transfer learning aims to improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$ and $T_S = T_T$. In addition, some unlabeled target-domain data must be available at training time" [6].

In other words, the source and target domain have the same tasks and the predictive function can be adjusted properly into the target domain in order to predict unlabeled target-domain data [6].

- **Unsupervised Transfer Learning**

  No labeled data are available in the source and target domains [3]. Data representations (or similarity and kernel matrices) need to be produced that will be evaluated on supervised learning tasks.

Following is the definition of Unsupervised Transfer Learning:

"Given a source domain D_S with a learning task T_S, a target domain D_T and a corresponding learning task T_T, unsupervised transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $T_S \neq T_T$ and Y_S and Y_T are not observable" [6].

4. STATE OF THE ART TOOLS AND ALGORITHMS

4.1 Splice Site Prediction Tools

This chapter presents related work in the field of splice site prediction. In the following table some tools are presented:

Table 4.1: Splice Site Prediction Tools.

Program	Organism	Method
GeneSplicer	Arabidopsis, human	HMM + MDD
NETPLANTGENE	Arabidopsis	NN
NETGENE	Human, C.elegans, Arabidopsis	NN + HMM
SPLICEVIEW	Eukaryotes	Score with consensus
NNSPLICE	Drosophila, humanorother	NN
SPLICE PREDICTOR	Arabidopsis,maize	linear models
BCM-SPL	Human, Drosophila, C.elegans	Linear

4.2 Splice Site Prediction Algorithms

In order to predict splice sites, various classification based methods have been used. The basic idea using such classification is to use a splice site sequence as a feature vector. The classifier uses the feature vectors of training samples as input in order to train model. The classifier can now predict the splice sites [5].

The main steps in splice sites classification are:

- **Feature extraction**

  Proper input representation plays an important role for the classifier. In this step, orthogonal encoding is used because of its simple processing and effectiveness as we already mentioned. Orthogonal encoding is often used to encode DNA sequences by features vectors. Each nucleotide is being represented with four binary bits, from which only one has the value 1 to represent one of the possible explicit values. This simple mapping procedure allows better classification results. In case of more fuzzy input data, Salekdeh et al. proposed a schema by which only four encoding patterns are used [5].

- **Classification**

  A set of labeled training data is used to train a classifier, resulting a classifier that separates the categories of sequence samples. For this purpose, different classification techniques are being used, including artificial neural networks (ANN), support vector machines (SVM) etc. with the latter providing very good results in splice site detection due to its high accuracies [5].

Various methods have been used for splice site recognition. Kernel-based and feature based methods are some of them, with the first having achieved really good performances in many species [17]. Other splice site detectors proposed linear SVMs on binary features, which achieved better results than previous Markov models [4]. Other methods have used multilayer neural networks with Markovian probabilities as inputs. More specifically, three Markov models have been trained on three segments of the input sequence, the upstream, signal and downstream segments. Although the results were satisfactory for small datasets, the slow training of the neural networks for imbalanced number of true and decoy examples, forced the authors to downsample the number of negatives for training [4]. Finally, a Bayesian Network based method, models statistical dependencies between nucleotide positions [4].

An important process in the classification of splice sites is the feature extraction. Two basic and well known methods are:

- Probabilistic models and
- Encoding schema

Probabilistic models are available to model local sequence behavior. On the other hand, in the **Encoding schema** such as the orthogonal encoding, nucleotides in sequence are viewed as unordered categorical values. Although orthogonal encoding is a method that is widely used because of its effectiveness, its accuracy can be influenced in case of ignoring the orders of nucleotides and codon usage. Therefore, a more effective feature extraction method is needed in order to improve the accuracy, transforming splice site sequence to a feature vector [5].

Feature-based methods aim to identify features that can be distinguished. The Feature Generation Algorithm (FGA) is such a method for splice site prediction having achieved good results [17].

There are cases in which splicing occurrences introduce premature termination codons triggering transcript degradation. From the disease-causing nucleotide substitutions that are listed in the Human Gene Mutation Database, a 14% is being thought to cause pre-mRNA splicing disruption, while only 2% of missense mutations disrupt pre-mRNA splicing [8]. Such sequences that disrupt the splicing procedure do not only cause disease but also influence its severity. Identifying mutations that disrupt pre-mRNA splicing becomes gradually an important field in the therapeutic treatment [8].

Some results from similar work are being represented [1][2][3]. We chose the most stable and higher performed algorithms.

Below the compared algorithms are being analyzed:

- **SVMS,T :** The idea of dual task is to have simultaneous optimization of both models and similarity between the solution enforced. In case of little target data availability, the training on source data performs much better than training on the target data, otherwise training on target data easily outperforms training the source data [1].

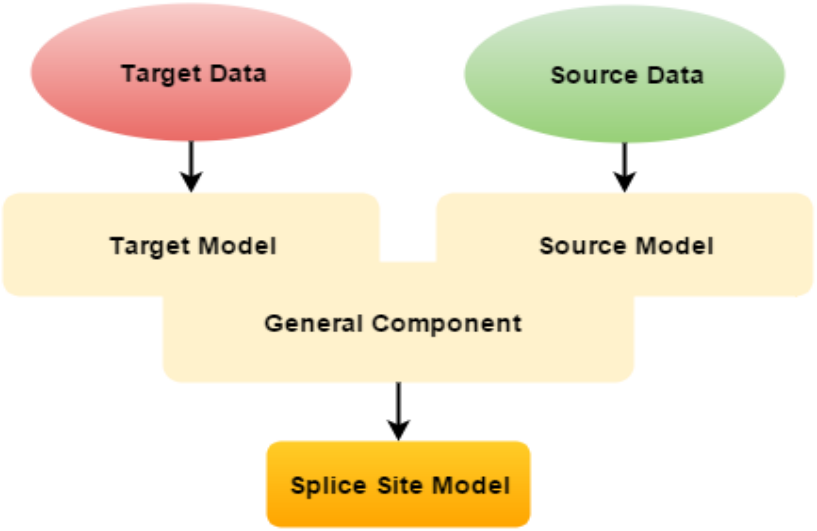

Figure 4.1: Diagram for Dual task.

- **NBT and A1:** Both are baseline naïve Bayes classifiers trained on target labeled and source data and they are both probabilistic models as well. The first one is a Naïve Bayes Tree classifier while the second one is based on improving the multinomial Naïve Bayes classifier, in which low weights are assigned to the target data [2].

- **AFMS:** The idea of All Features Majority Strategy (AFMS), is to use majority voting between the four representations of each sequence. In case of a tie the highest sum of absolute values is selected. It is observed that knowledge obtained from the source domain is better to be used only for the initialization of the centroids and not during the iterations. AFMS is a stable algorithm in many cases, without extreme oscillations. The classification potential of this strategy seems to achieve high performances in most organisms [3].

5. PROPOSED APPROACH

5.1 Introduction

In this work, we propose a new method for the problem of splice sites recognition. The method combines the use of biological features with different representations such as the N-gram graphs. Different classifiers are studied in order to choose the most appropriate for our problem setting. The CRF (Conditional Random Field) algorithm, which is also widely used in the field of gene prediction, is being studied as well.

The choice of the N-gram graph representation is based on the fact that it provides accurate results in machine learning problems [3][7]. With the use of the biological features' information, these rates increase further. In our research, due to unbalanced data, we have chosen to study the F-measure statistic, in order to get more representative results.

With the help of the N-gram graphs, features can be extracted. Using each DNA sequence and with the help of N-gram graphs' similarity we obtain our first two features, the splice sites and non-splice sites respectively. This procedure is done both in the target and in the source domain as well.

At this point it should be noted that in order to avoid overfitting we used 70% of the training set. Also in multiple source domains, where the source domain has more than one species, two solutions were tested:

- 1st solution: We took into account the merging of the source domain's mean graphs.
- 2nd solution: A second approach could be applied by choosing the closest source domain in accordance with the target domain. This decision could be taken based on a phylogenetic analysis. We followed the first approach since we achieved better results.

Our algorithm is trained for one species while we try to adapt the classifier to make splice site prediction for a different one. The feature vector we use is the same in both source and target domains. Comparing the distributions of the features of the source and target domains, if we could manage to customize them in an appropriate manner in order to behave similarly, then we would relegate the issue in a Machine Learning problem and solve it as usual [23].

In this study, we examine two algorithms. The first one provides the most similar sequences to the classifier, while the second transforms the test data in order to bridge the gap with the training data.

5.2 Algorithms for transfer learning optimization

In this section we propose two algorithms which will try to optimize the data aiming the classifiers' better adaptation.

5.2.1 First Algorithm-Similarity of source and target data

The basic idea of the first approach concerns the merging of instances from the source domain that are more similar to those of the target domain.

Having the source data we will distinguish them in splice sites (label -1) and non splice sites (label 1). Respectively for the target domain, using K Means algorithm [3] we split the data into two clusters and use the SVM classifier, which characterizes the cluster with the bigger amount of non splice sites sequences as a “non splice sites cluster”. Respectively, the amount of splice sites sequences is being considered for the splice sites cluster. At this point, source and target data have been characterized. We obtain the most similar (negative from source with negative from target and positive from source with positive from target) between them with the use of the cosine distance. The data produced are added to the training data. With the new training set we train the SVM classifier and learn a model in order to be able to classify.

In the following diagram, one can see the workflow of the Similarity Algorithm:

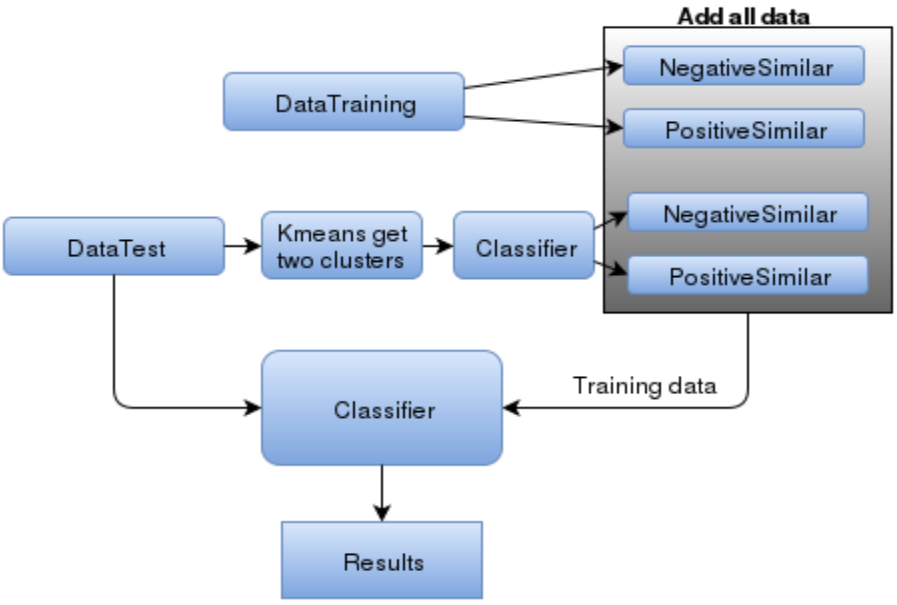

Image 5.1: Similarity Algorithm Workflow.

5.2.2 Second Algorithm-Target Data Transformation

The main idea described in the following algorithm, which is conceived from the paper [3], is as follows:

As in the first algorithm, the most similar source and target data are obtained in the same way. Based on the following equation (1), we transform the initial target data with the help of the mean values of the similar data. For instance, calculating the mean value of a feature, of each similar-source and similar-target data, and defining a static value “a”, the distributions of each feature approximate each other [32].

$$\mathrm{fx} = \mathrm{a} * \mathrm{fx} + (\mathrm{a} - 1) * \mathrm{fx}\left(\frac{\mathrm{meanTraining(feature\_x)}}{\mathrm{meanTest(feature\_x)}}\right) \qquad (1)$$

In the following diagram the workflow of the Data Transformation Algorithm is being presented:

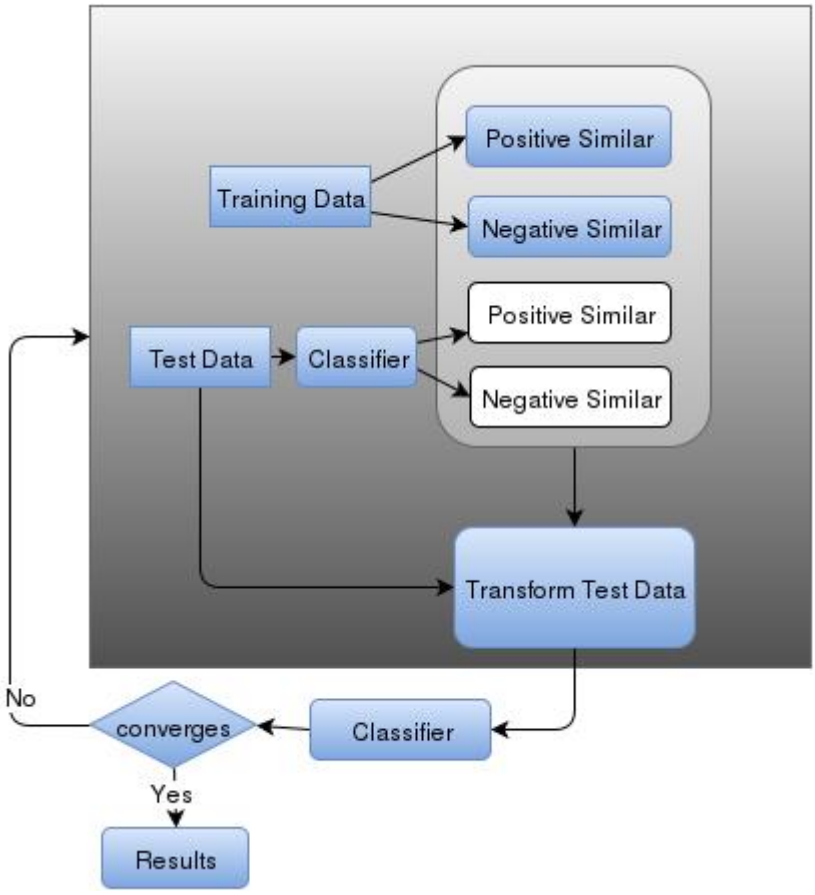

Image 5.2: Data Transformation Algorithm Workflow.

5.3 Sequence Representation

5.3.1 N-Gram Graphs

N-gram graph representation, has been proposed widely in the field of natural language processing. An initial definition that could be given describes n-gram as a possibly ordered set of words that contains n elements [7]. N-gram graph representation methodology manages to capture local and global characteristics of the analyzed sequences [3].

Based on the definition, the N-gram graph (NGG) is a graph $G=<V^G,E^G,L,W>$, where V^G is the set of vertices, E^G is the set of edges, L is a function assigning a label to each vertex and to each edge and W is a function assigning a weight to every edge. The graph has n-grams labeling its vertices $u^G \in V^G$. The edges $e^G \in E^G$ connecting the n-grams indicate proximity of the corresponding vertex n-grams. The weight of the edges can indicate a variety of traits.

It is important to note that in N-gram graphs each vertex is unique. In order to create the n-gram graph from a given sequence, a fixed-width window D_{win} of characters around a given n-gram N_0 is used. All character n-grams within the window are considered to be neighbors of N_0.These neighbors are represented as connected vertices in the text

graph. Each edge e=<a,b> is weighted based on the number of co-occurrences of the neighbors within a window in the sequence [3][7].

The idea of our first approach uses the N-gram graphs in which close subsequences consist of a crucial part of the sequence. Essentially, the N-gram graph is a histogram of symbols' co-occurrences which are captured when found into a maximum distance (window) of each other. Also, it's worth noting that N-gram graphs are deterministic, they offer more information based on the representation of co-occurrences, they provide trade-off between expressiveness and generalization and they can be combined with vector representation of sequences in order to allow machine learning techniques to classify sequences [22].

The n-gram graph, compared to other representation methods, differs in many areas from the typical graphs structure [7].

- In case of a feature vector creation from an N-gram graph, in which the edges are the dimensions of the feature vector, the indirect relation between vertices is lost.
- If one uses the same information in order to construct a vector then there is high complexity in the transformation process
- Using N-gram graph representation in Natural Language Processing, no assumption can be made about the underlying language as a result the representation is made language-neutral and independent of writing orientation.
- The N-gram graph can be used in many applications, such as text representation, gene prediction etc.

5.3.2 N-gram Graphs Algorithm

The algorithm we used for N-gram graph extraction of a sequence has the following main steps [7]:

- Initializing the N-gram graph by setting the parameters.
- For each sequence a graph is being created.
- Merging graphs to a mean graph for both labels.

Having the necessary data information for each species, namely the sequences and their labels (positive or negative), two features are being extracted. The first step is to draw mean graphs for labeled and unlabeled data. Mean graphs are generated from the training set where the labels are known.

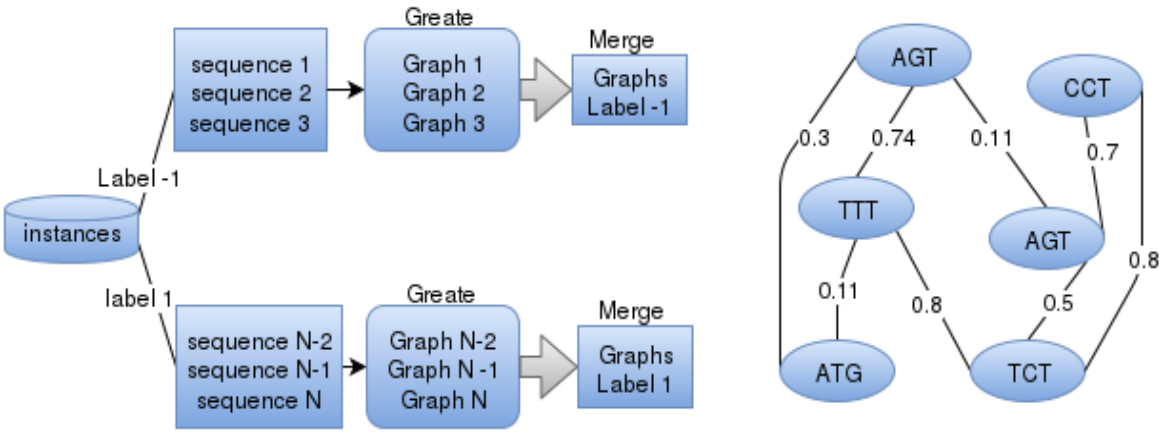

Image 5.3: Mean graph creation.

With the help of the mean N-gram graphs for each species two features can be extracted. The first one concerns the negative-unlabeled data and the second the positive-labeled data. Simultaneously, biological features for each sequence are being created as well.

Having created the features for each species, with the help of a classifier we can perform evaluation and estimate the accuracy and F-measure in order to check the algorithm's success rates. At this point several classifiers have been tested such as the Decision Tree, SVM with RBF kernel, Linear SVM and others.

5.3.3 Biological Features

At the same time, biological features for each sequence are being created. The biological features are the following:

- The nucleotide occurrences' rates (see 6.2 section).
- The sum of the occurrences' rates of the purine and pyrimidine cores, in order to express the probability of more frequent C and T nucleotides' occurrence.
- The branch site Motif "ynyyrAy". This motif is being detected 20-50 nucleotides before the acceptor dimer AG.
- Acceptor Motif "AG". This dimer is a motif for most acceptor sites and the general motif is "yAGr".
- Donor Motif "GT". This dimer is a motif for most donor sites and the general motif is "mrgGTrag".
- The Pairwise Alignment Score (see 6.2 section).

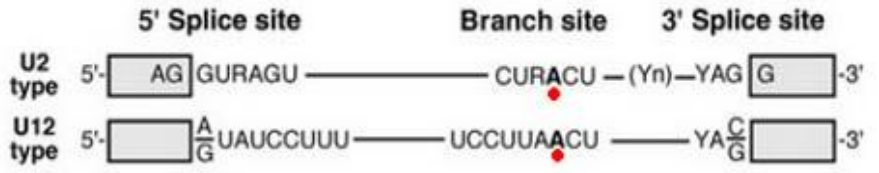

Image 5.4: Splice site Motifs [24].

5.4 N-gram Graph Parameters

N-gram graph has some values that must be initialized such as min, max and distance value. The distance is a window, while min and max values are the limits for the size of the combinations that can be made in this window. Depending on these values, a feature can obtain high resolution efficiency.

These values were selected experimentally, having in mind that triplets of nucleotides are being used during the DNA translation process (e.g. defining min=3, max=4 and distance=3, N-gram graph will represent the sequences with motifs consisted of three and four nucleotides).

5.5 Phylogenetic Analysis

Phylogenetic analysis helps improve the transfer learning approach by showing the phylogenetic distance between species. In this way one can classify them in order to have better visualization and put a weight on more significant data. For example, those that are present in a same branch, have probably the same genes as well. The closer the species, the more similar the data is, and thus the splice sites will have a corresponding similarity.

Due to unbalanced data, i.e. the number of non splice site instances is greater than the number of splice site instances, and because we are facing a multiple source domain problem we can use weights for each instance helping the classifier to achieve higher resolution efficiency.

In the picture below, our approach in finding splice sites is being represented using phylogenetic analysis, which will provide us with the distance matrix (Distance Matrix) with the phylogenetic distances of our species.

Image 5.5: Phylogenetic Analysis Steps.

.

For the application of weights we propose the following:

Firstly, we put weights in each instance through phylogenetic analysis. This will lead to help the algorithm to give greater weights to instances from the source domain which is phylogenetically close to the target domain. This solution helps the multiple source domain approach.

Having received a conserved region that is in all species (i.e. a protein [14]), we applied the multiple source domain approach and got the distance matrix, which we then converted into rates. We obtained this information in order to give weights to the instances.

Secondly, according to the number of instances that each label has, we put a weight. For example, if we have 1000 instances of which 70% are characterized with label -1 and 30% with label 1 we could define a weight "x" for the labels -1 and a weight "2x" for the label 1.

Finally, we used the cosine distance in order to get the similarity. Afterwards we get for each instance the similarity rate with respect to the data set and that is being used as the instance's weight.

$$Similarity = \cos(\theta) = \frac{A \cdot B}{|A|\,|B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

With very little data, features and instances, the algorithm converges to very good results with respect to those that already exist and in relatively quick time.

5.6 Classifiers

Having the features for each species, several classifiers have been tested such as the Decision Tree, SVM with RBF kernel, Linear SVM and others in order to perform evaluation.

Generally, in order to choose classifiers one has to have the following characteristics in mind [33]:

- Computational cost
- Expected data types of features/labels
- Suitability for certain sizes and dimensions of data sets
- Fast performance
- Good accuracies
- Good error approximation

As one can notice from the "Experiments and Results" section, KNN and SVM classifiers performed best.

5.7 CRF

The algorithm we used has the following steps:

- Create patterns for the CRF [25].
- Use of Wapiti program (See Experiments).

The patterns created have forms such as those shown in the following image [25]:

Feature Type	SubTypes
BASE	$b_{i-5}, b_{i-4}, b_{i-3}, b_{i-2}, b_{i-1}, b_i$
	$b_{i-4}, b_{i-3}, b_{i-2}, b_{i-1}, b_i$
	$b_{i+4}, b_{i+3}, b_{i+2}, b_{i+1}, b_i$
	b_{i-2}, b_{i-1}, b_i
	b_{i+2}, b_{i+1}, b_i
	b_{i-1}, b_i
	b_{i+1}, b_i
	$b_{i-2}, b_{i-1}, b_i, b_{i+1}, b_{i+2}$
	b_{i-1}
	b_{i+1}
	b_i

Image 5.6: CRF patterns [25].

For example, using the SubType "b_i", crf will take incidence rates for each nucleotide, while using the SubType "bi, bi+1" crf will take incidence rates for all dimers and so on. From these patterns, features are being created from the crf program, that are nothing more but nucleotide motifs. Those compose a model which is being used in order to take a decision concerning each nucleotides label.

In our approach, for each sequence, we sum up CRF's decision in order to decide whether the specific sequence is a splice site or not. In other words, if the sum is positive, then the sequence is a splice site.

6. EXPERIMENTS AND RESULTS

6.1 Experimental Setup

From the references [1] one can see that the dataset was taken from the Rätsch lab (http://cbio.mskcc.org/public/raetschlab/user/behr/splicing/), and so did we. The main idea is to recognize splice sites in different species. In most experiments, C.elegans is being used as a source domain and the other species as target domain.

Our dataset, provided by the Rätsch laboratory, consists of sequences of the following species, from which we used only the acceptor splice sites:

- A_thaliana
- C_elegans
- D_melanogaster
- D_rerio
- H_sapiens

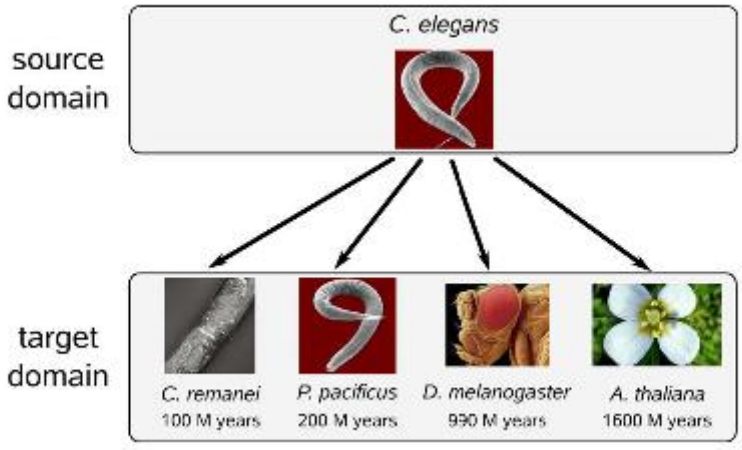

Image 6.1: The distance in years between C. elegans and other species.

The dataset consists of sequences that are made up from 200 nucleotides and only a 1% from the dataset is a splice site (positive instances). All these approaches use a few instances from the target domain in order to train their algorithm and the rest of the sequences are used for evaluation.

Having our preprocessed dataset (See APPENDIX 1), we choose some data for our experiments. For instance, we choose randomly 40.000 sequences from the Fasta file, in which 70% are non-labeled and 30% labeled data, in order to represent real world data.

Evaluation is being performed and the F-measure metrics are being estimated in order to check our algorithms' success rates. Different classifiers have been tested such as the Decision Tree, SVM with RBF kernel, Linear SVM and others. Furthermore, with the Libsvm library we tested linear and polynomial kernel and finally the KNN classifier was studied in which the 3-NN and 21-NN were tested, having used the Manhattan distance.

Concerning the CRF algorithm the Wapiti program was used. Wapiti is a program for training and using discriminative sequence labeling models with various algorithms using an elastic penalty. It currently implements maxent models, maximum entropy Markov models (MEMM) and linear-chain conditional random fields (CRF) models. It can work in different mode depending on the first argument, either training a model, labeling new data, or dumping a model in readable form [28].

In our case, we used Wapiti as a CRF algorithm program. It receives as parameters two files, the training and test set. These files have the following structure: The DNA sequences are represented in a file in which each line represents a nucleotide and the label of the sequence it belongs. Also, sequences are being separated by an empty line.

The patterns are set in a separate file and those will be followed by the CRF algorithm in order to extract results. Having tried various patterns we conclude to those that resulted optimally.

Having CRF's prediction results for each test set, summing up the nucleotides' labels' results from each sequence, a decision is being taken in order to clarify if the label of the particular sequence is negative or positive. This experiment was performed with 16.000 sequences.

6.2 Feature Extraction

The biological features mentioned previously where extracted using a dynamic algorithm. In the case of the nucleotide occurrences' rates, the user can choose the number of nucleotides as done in K-mer as well. Concerning the pairwise alignment score, the Biojava package global alignment is being applied between each sequence and the mean graph's sequence. In this way we get two features, comparing the similarity of the sequence with the string that uniquely characterizes the mean graph with unlabeled data and the string that uniquely characterizes the mean graph with labeled data respectively.

6.3 NGRAM Parameters

Table 6.1: Choosing min, max and distance parameters, with KNN classifier and 6.500 entries.

Species / Ngram-Parameters	H_sapiens	D_rerio	D_melanogaster	C_elegans	A_thaliana
M:8,M:8,D:1	0.86	0.71	0.82	0.75	0.74
M:4,M:6,D:2	**0.82**	**0.80**	**0.83**	**0.81**	**0.81**
M:2,M:4,D:3	0.81	0.79	0.82	0.80	0.80
M:2,M:4,D:2	0.80	0.80	0.83	0.79	0.81
M:3,M:4,D:3	**0.81**	**0.80**	**0.83**	**0.81**	**0.81**
M:4,M:4,D:1	0.81	0.81	0.83	0.80	0.81
M:3,M:3,D:3	**0.82**	**0.80**	**0.84**	**0.81**	**0.80**
M:6,M:6,D:1	0.83	0.80	0.82	0.80	0.81
M:3,M:5,D:3	0.82	0.80	0.83	0.80	0.80
M:3,M:3,D:2	0.81	0.80	0.83	0.80	0.80

In the above table, we chose several values (F-measure) for the parameters of the N-gram graph algorithm. We noticed that we achieve the best results with the bolded

values. We ended up choosing the values: min=3, max= 4 and distance=3. It is also worth noting that each cell of the table represents the average of each target domain (i.e. for each species) that comes up. The data have been presented appropriately in order to show the differences and achieve better visualization.

Table 6.2: Using whole or part of the sequence, with KNN classifier and 6.500 entries.

Species / Ngram-Parameters	H_sapiens	D_rerio	D_melanogaster	C_elegans	A_thaliana
M:4,M:6,D:2 All:	0.82	0.81	0.83	0.81	0.81
Part:	0.80	0.78	0.79	0.78	0.79
M:3,M:4,D:3 All:	**0.81**	**0.81**	**0.84**	**0.81**	**0.81**
Part:	0.81	0.79	0.82	0.79	0.80
M:3,M:3,D:3 All:	0.81	0.80	0.84	0.80	0.81
Part:	0.81	0.79	0.81	0.79	0.79

The above table shows the optimal parameter sets from the experiment mentioned previously and compares the results that are obtained using whether the whole sequence or part of it. This part of the sequence uses 50 nucleotides left of the acceptor site, due to the biological information that is located in this area. We note that the results do not differentiate a lot as we saw before. Nevertheless we are choosing the set min=3, max= 4 and distance= 3, because of the lower computational costs and slightly higher results achievement. Another fact that has been observed was that by using only a 1/4 of the sequence (50 nucleotides) the specific set of parameters gives results similar to those of the entire sequence reducing further the computational costs mentioned above.

But this is not a coincidence at all because according to the approaches of paperwork [29] that use K-mers we notice that best results are being achieved when we use 4-mer and 6-mer. So in our case these values help features' generalization capability and as a result we can export better results in transfer learning as well.

Table 6.3: KNN classifier.

Entries / Species	**1000**	**2500**	**6500**	**16000**
H_sapiens_acc	0.99	0.99	0.99	0.98
D_rerio_acc	0.99	0.99	0.99	0.98
D_melanogaster_acc	0.99	0.99	0.98	0.98
C_elegans_acc	0.97	0.98	0.96	0.95
A_thaliana_acc	0.97	0.98	0.97	0.95

In the table above we have used the source and the target domain of the same species. The columns describe the number of sequences taken into account in each experiment while the lines describe the target domain (species). Finally, we chose the parameters set of min=8, max=8 and distance=1 for the N-gram graph algorithm, achieving best results and low costs simultaneously. We ended up using 6.500 sequences for our experiments.

6.4 Choosing Classifier

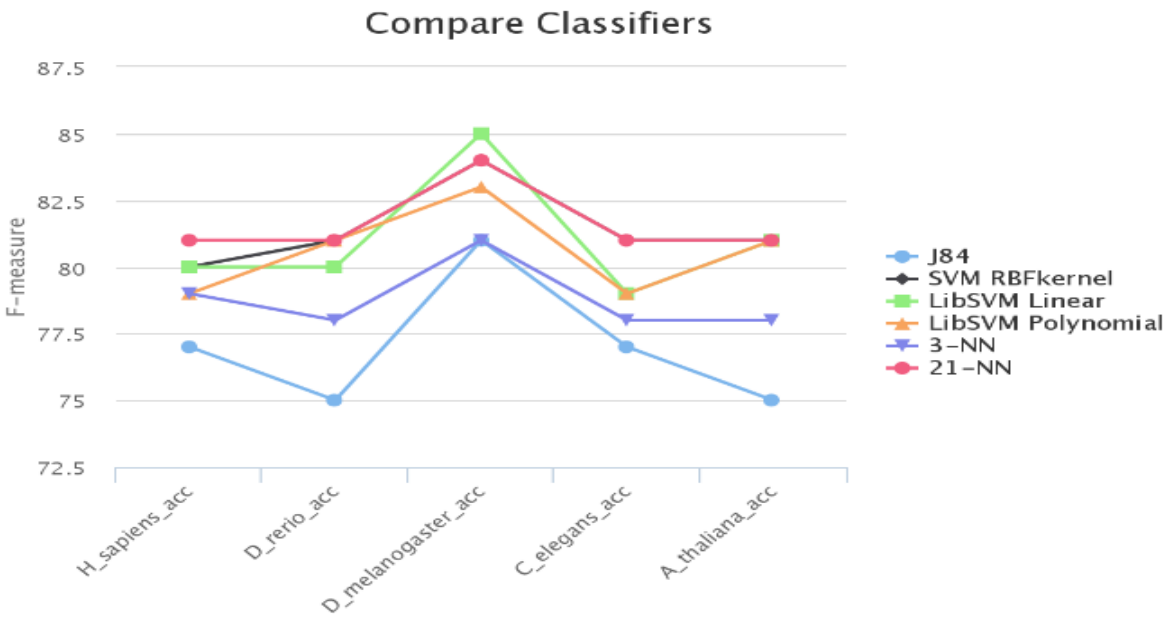

Image 6.2: Comparing Classifiers.

In the above diagram we see the various classifiers that have been used and the averages of the results (i.e. averages of the target domain) for each species. We notice that the best results are being produced using the KNN classifier. The species have been sorted according to their phylogenetic distances. Thus, the more the phylogenetical distance from the species grows, the more the rates decline.

Table 6.4: KNN overall results.

Source / Target	H_sapiens	D_rerio	D_melanogaster	C_elegans	A_thaliana
H_sapiens	0.87	0.82	0.84	0.76	0.76
D_rerio	0.77	0.86	0.83	0.76	0.81
D_melanogaster	0.84	0.83	0.89	0.84	0.80
C_elegans	0.78	0.75	0.82	0.87	0.80
A_thaliana	0.79	0.81	0.80	0.81	0.84

The classifiers that were used and studied are, the decision tree, the SVM classifier which was tested with several kernels such as RBF Kernel and who had the best results. Furthermore, with the Libsvm library we tested linear and polynomial kernel and finally the KNN classifier was studied in which the 3-NN and 21-NN were tested, having used the Manhattan distance.

6.5 Unbalanced Data

Having received a conserved region that is in all species (i.e. a protein [14]), the multiple source domain approach was applied in previous section. Following, the distance matrix is being shown, which was converted into rates.

Table 6.5: Distance Matrix.

Species	H_sapiens	D_rerio	D_melanogaster	C_elegans	A_thaliana
H_sapiens	100.00	44.44	37.66	29.86	17.92
D_rerio	44.44	100.00	33.83	28.51	16.62
D_melanogaster	37.66	33.83	100.00	29.46	13.99
C_elegans	29.86	28.51	29.46	100.00	16.90
A_thaliana	17.92	16.62	13.99	16.90	100.00

6.6 Similarity of Source and Target Data

Having trained the classifier with the new training set, the accuracy of our algorithm is being presented in the following table:

Table 6.6: First Algorithm results.

Source / Target	H_sapiens	D_rerio	D_melanogaster	C_elegans	A_thaliana
H_sapiens	0.84	0.83	0.87	0.82	0.82
D_rerio	0.81	0.84	0.83	0.75	0.80
D_melanogaster	0.81	0.82	0.86	0.84	0.82
C_elegans	0.80	0.72	0.83	0.87	0.78
A_thaliana	0.79	0.82	0.79	0.80	0.83

The algorithm having as main characteristic the similarity and taking advantage of the testing data, achieves stability and improves results. Below we can see an overall picture-diagram with the classifiers.

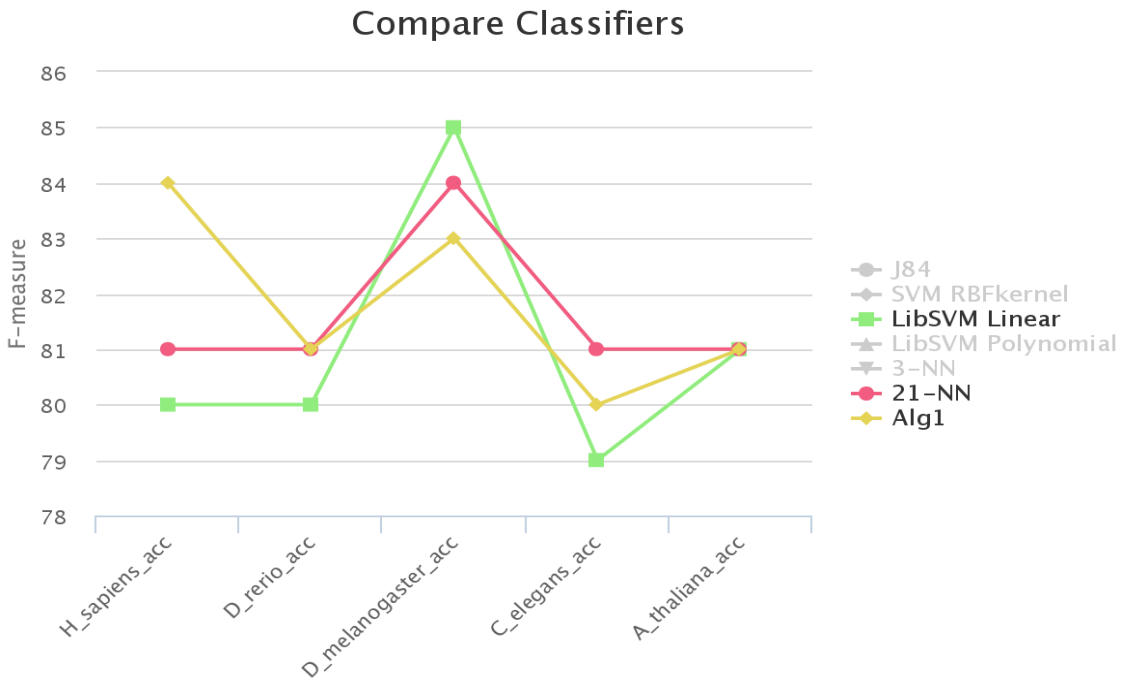

Image 6.3: Best Classifiers Comparison with first algorithm.

6.7 Target Data Transformation

Table 6.7: Second Algorithm results.

Source / Target	H_sapiens	D_rerio	D_melanogaster	C_elegans	A_thaliana
H_sapiens	0.82	0.83	0.85	0.78	0.77
D_rerio	0.79	0.81	0.82	0.72	0.80
D_melanogaster	0.81	0.67	0.86	0.80	0.78
C_elegans	0.81	0.60	0.84	0.87	0.76
A_thaliana	0.81	0.79	0.79	0.78	0.83

Although the algorithm uses as essential characteristic the transformation and the similarity and takes advantage of the testing data we do not notice improved results.

Image 6.4: Best Classifiers Comparison with second algorithm.

6.8 NGRAM (Multiple Source Domain)

The previously mentioned algorithms (Similarity of source and target data algorithm and Target Data Transformation algorithm), were also studied with the Multiple Source Domain approach. In the following table Homo Sapiens Species is notated as “S” or “Sap”, Rerio species is notated as “R” or “Rer”, Melanogaster species is notated as “M” or “Melang”, Elegans species as “E” or “Eleg” and finally Thaliana species is notated as “T” or “Thal”. The horizontal row of the species presents the source domains and the first column presents the target domains. Finally, in case the same species is included in the source and the target domain simultaneously, the task is being interrupted immediately and visualized as zeros. The results are listed below:

Table 6.8: Multiple Source Domain results for the KNN.

Species	M,R	M,T	M,S	S,R	S,T	M,E	R,T	S,E	R,E	T,E
Sap.	0.84	0.83	0.00	0.00	0.00	0.82	0.82	0.00	0.82	0.77
Rer.	0.00	0.84	0.82	0.00	0.80	0.82	0.00	0.81	0.00	0.80
Melang.	0.00	0.00	0.00	0.85	0.83	0.00	0.84	0.85	0.85	0.83
Eleg.	0.83	0.83	0.83	0.80	0.81	0.00	0.79	0.00	0.00	0.00
Thal.	0.81	0.00	0.81	0.81	0.00	0.81	0.00	0.81	0.82	0.00

Table 6.9: Multiple Source Domain results for the first algorithm.

Species	M,R	M,T	M,S	S,R	S,T	M,E	R,T	S,E	R,E	T,E
Sap.	0.86	0.86	0.00	0.00	0.00	0.85	0.85	0.00	0.84	0.77
Rer.	0.00	0.81	0.82	0.00	0.81	0.81	0.00	0.81	0.00	0.79
Melang.	0.00	0.00	0.00	0.81	0.82	0.00	0.85	0.82	0.84	0.85
Eleg.	0.83	0.84	0.83	0.78	0.80	0.00	0.81	0.00	0.00	0.00
Thal.	0.80	0.00	0.79	0.81	0.00	0.81	0.00	0.82	0.82	0.00

Table 6.10: Multiple Source Domain results for the second algorithm.

Species	M,R	M,T	M,S	S,R	S,T	M,E	R,T	S,E	R,E	T,E
Sap.	0.83	0.83	0.00	0.00	0.00	0.83	0.83	0.00	0.83	0.83
Rer.	0.00	0.80	0.81	0.00	0.76	0.80	0.00	0.78	0.00	0.79
Melang.	0.00	0.00	0.00	0.78	0.78	0.00	0.84	0.81	0.84	0.82
Eleg.	0.81	0.83	0.83	0.75	0.79	0.00	0.81	0.00	0.00	0.00
Thal.	0.77	0.00	0.79	0.76	0.00	0.80	0.00	0.80	0.81	0.00

Comparing the averages, we notice that the results remain constant. Also, when we have two species on the source domain, target domain's value is very close to the corresponding values that would have resulted if we had each source domain separately. For example if we had as source domain the Melanogaster and Thaliana species and as target domain the Homo Sapiens species then the target value would range between Melanogaster/Sapiens and Thaliana/Sapiens values.

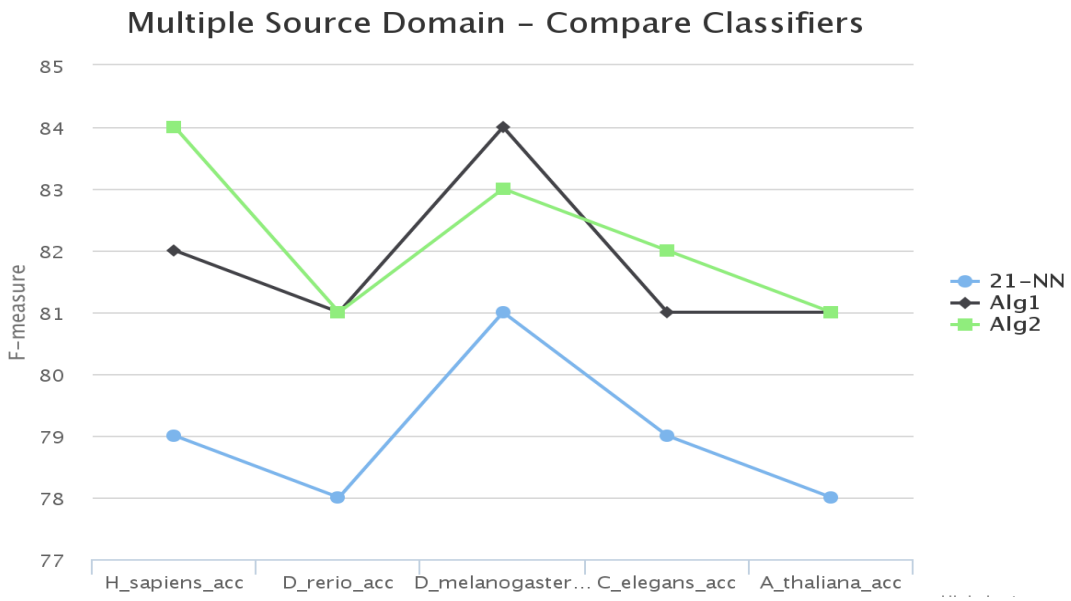

Image 6.5: Comparing Classifiers for the Multiple Source Domain.

6.9 N-gram Graph Comparison with State-of-the-art

Following, the overall results are being presented, compared with the state of the art algorithms mentioned in previous chapter. The performance of the models are evaluated by measuring the accuracy in terms of area under the Receiver Operator Characteristic Curve (auROC). Furthermore C.elegans data were used as training set while all the other species as test set in order to compare our approach with the State of the art approaches:

Table 6.11: D. melanogaster.

Algorithms	**Sequences**			
	2500	**6500**	**16000**	**40000**
SVMS,T	40.80	37.87	52.33	58.17
NBT	13.87	25.00	35.28	45.85
A1	25.83	32.58	39.10	47.49
AFMS a=1	67	-	-	-
CRF	38	46	53	-
Alg. 1	85	85	81	79
Alg. 2	83	78	74	74

Table 6.12: A. thaliana.

Algorithms	Sequences			
	2500	**6500**	**16000**	**40000**
SVMS,T	24.21	27.30	38.46	49.75
NBT	3.10	8.76	28.11	40.92
A1	3.99	13.96	33.62	43.20
AFMS a=1	53	-	-	-
CRF	25	39	51	-
Alg. 1	83	80	78	78
Alg. 2	82	76	69	72

State of the art algorithms are based in probabilistic models and when they use bigger data sets for training in order to achieve better performances, success rates increase with the computational costs simultaneously.

In our approach, we took advantage of both the N-gram graphs and the biological information, in order to extract features keeping the problem's space dimension low at the same time. We notice that despite the dataset's size, our results are fairly close. Furthermore, the time needed in order to execute the biggest experiment did not exceeded a day using a state of the art computer.

Concerning the two algorithms we proposed, the first algorithm's classification potential seems to be greater in most organisms. The results we obtained seems to be comparable with state-of-the-art approaches. Finally, for the organisms with bigger evolutionary distance, it is more difficult to achieve good results, most probably because the secondary structure of the DNA sequence has changed more overtime [3].

7. CONCLUSION

Having started from the basic biological background knowledge concerning the problem of finding splice sites, our work focused on developing transfer learning algorithms using the N-gram graph representation.

Starting with the N-gram graph, we proposed a new method in order to manage finding and recognizing splice sites. More specifically, we combined the use of biological features, with the N-gram graph representation and the study of several classifiers. The choice of the N-gram graph representation was based on the accurate results in machine learning problems it provides. With the use of the biological features' information, these rates increase further.

Apart from the optimal parameter set and the appropriate classifier, experiments were done using whether the whole sequence or part of it (50 nucleotides left of the acceptor site), in which basic biological information is known to be located. In this way, lower computational cost and slightly higher results are being achieved. Another conclusion drawn from our sorted species, concerns the declining rates due to their phylogenetical distance.

Two algorithms were proposed for processing the data received by classifiers. The main idea of the first one was to provide the classifier the most similar sequences while the second algorithm transforms the test data in order to approach the training data.

The basic idea of the first approach concerned the merging of instances from the source domain that are more similar to those of the target domain. The algorithm having as main characteristic the similarity described in the previous section and taking advantage of the testing data, achieved stability and improved results. In the second approach, for each instance of the test set, each feature underwent a small transferring in the training set and a transformation parameter is taken into account. With these changes we tried the target domain approached the source domain and we improved the classifier. The above mentioned algorithms were studied with the Multiple Source Domain approach as well.

CRF's main idea was to use patterns in order to extract a model composed of features (nucleotide motifs) and put them as input into the Wapiti program, which executed the CRF algorithm. Finally we performed evaluation and got our results.

CRF has contributed in the field of gene prediction undeniably and without using the biological information, something that could be set as a future target. Both approaches are very good as machine learning approaches, i.e. making splice site prediction in the same organism. However in our work we dealt more with the N-gram graph algorithm using transfer learning and by adapting biological information the results were satisfactory approaching those of machine learning. In the future researchers could import more biological features [2] in order to increase the accuracy of the algorithm.

All in all, we noticed from our results that our work contributed in the field of splice site recognition in an important manner. We proposed the N-gram graph representation and similarity in order to obtain the first two features of our representation. Simultaneously, biological information was used with the help of a few important motifs. The latter was combined with the N-gram graph features. With the proposed representation, we managed to achieve higher prediction accuracy than the current approaches of the state-of-the-art. In addition, the proposed representation uses a small amount of features, which help us achieve high performances quickly and with low computational cost. We have proposed two transfer learning algorithms based on this representation

and on similarity measures between training and test sets. We proposed an approach for transforming the initial target data with the help of the mean values of the similar data in order to approximate the distributions of each feature of the training set. Finally, the multiple source domain approach helped us considering the case in which the source domain had more than one species and in both approaches we achieved better results. For this setting, we used information from the phylogenetic analysis of species.

With the above results, we achieved better performances using more representative features. Also we reduced the computational costs by using only a small amount of features. Our largest experiment with 40.000 instances required less than a day in order to get results. Finally, in the case of multiple source domain, when we used more than one species as source domain our performances improved significantly[2].

[2] Our work is going to be published in the near future.

APPENDIX 1

➢ Technical Details (Workflow)

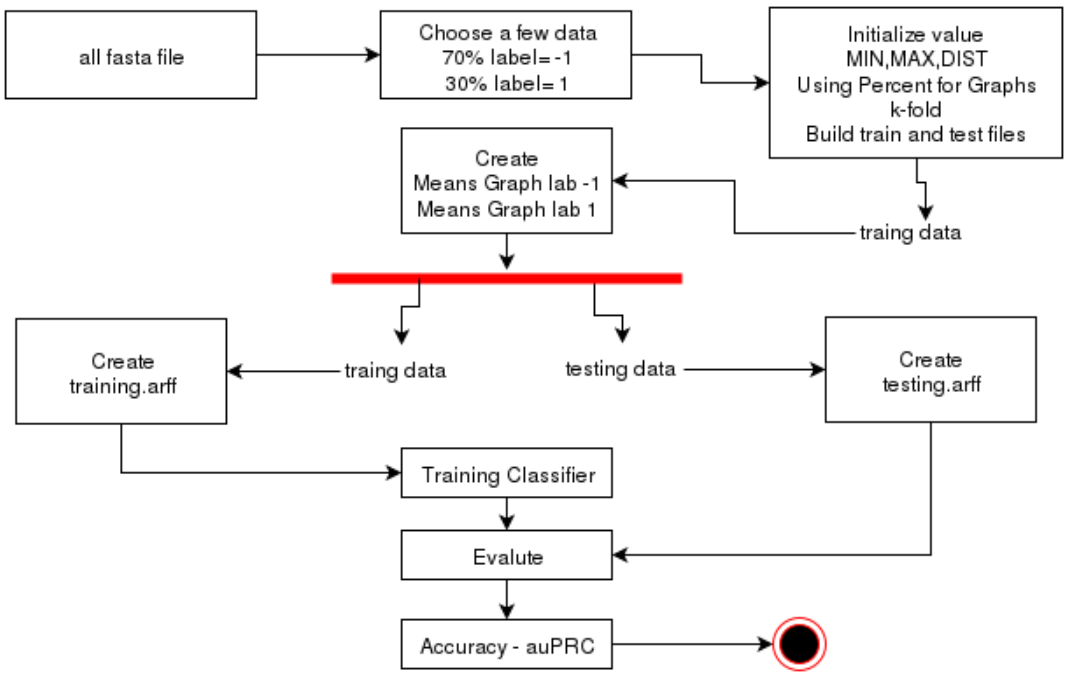

Image A1.1: N-gram graph Workflow.

As we can see from the above workflow, we downloaded the Fasta files for the species from the link (http://cbio.mskcc.org/public/raetschlab/user/behr/splicing/). After the data have been preprocessed, our files have the format we see in the following picture.

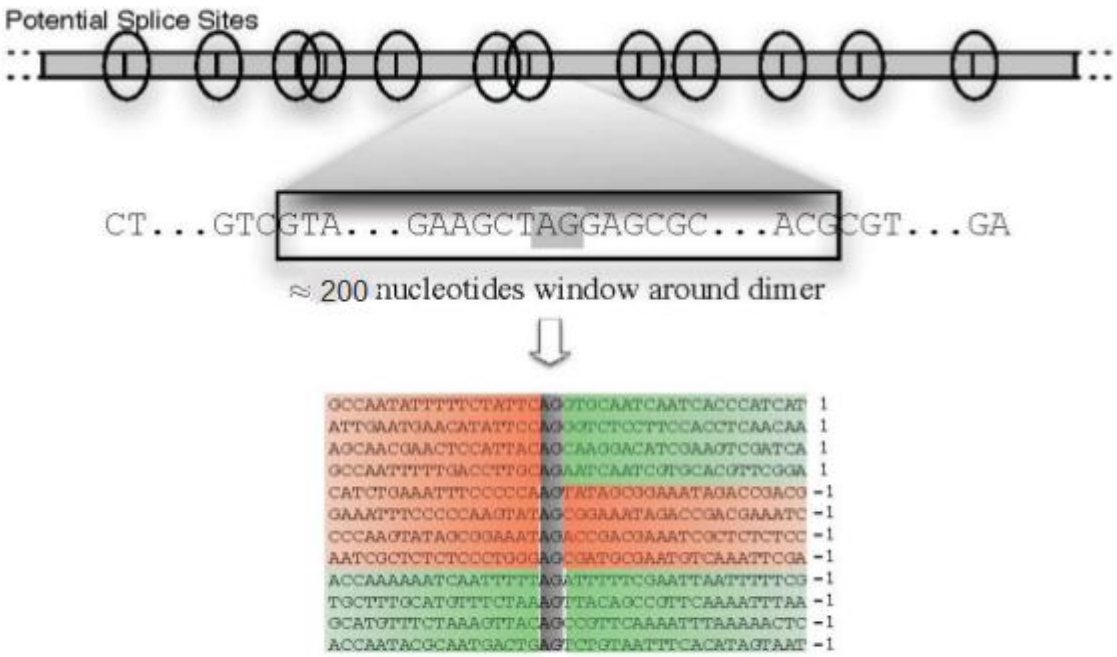

Image A1.2: Data Alignment Representation.

One can notice that they have been aligned to the pattern of the acceptor (AG) or the donor (GT) respectively. From these files some data are chosen for our experiments as we mentioned in the previous subchapter. Each row in the new file contains the sequence and in the end the label for each sequence is being presented.

The result is a new “arff” file which consists of the features produced from our algorithm (N-gram graph) and the biological features as well. The data from each species are stored in a structure in order to use them. In this structure, we keep for each species the original data, the sequence and the label of the sequence and for each of these sequences features are being extracted and stored (Instances). These attributes are being extracted according to the N-gram graph algorithm and the biological features, which then are being used by the classifier in order to extract the final results.

APPENDIX 2

➢ Choosing Classifier (Workflow)

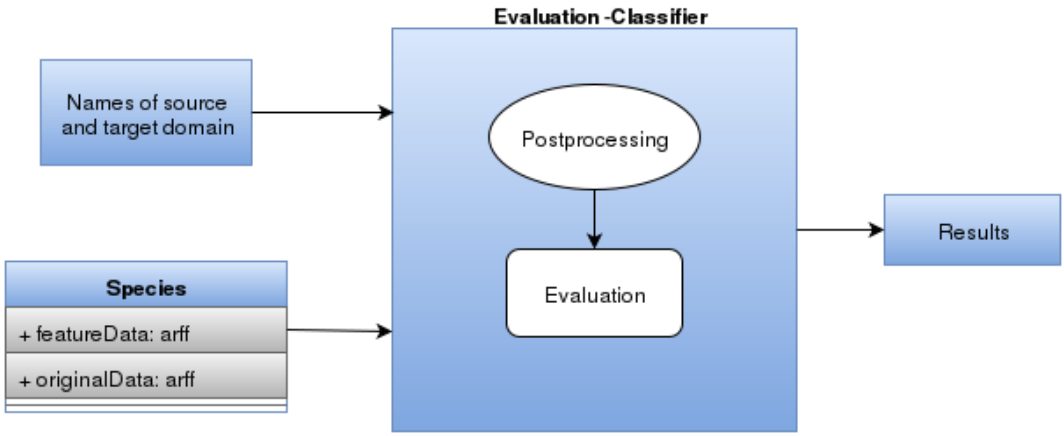

Image A2.1: Evaluation Diagram.

Having created the arff files containing the features for each species, several classifiers have been tested such as the Decision Tree, SVM with RBF kernel, Linear SVM and others in order to perform evaluation.

APPENDIX 3

➢ CRF Algorithm (Workflow)

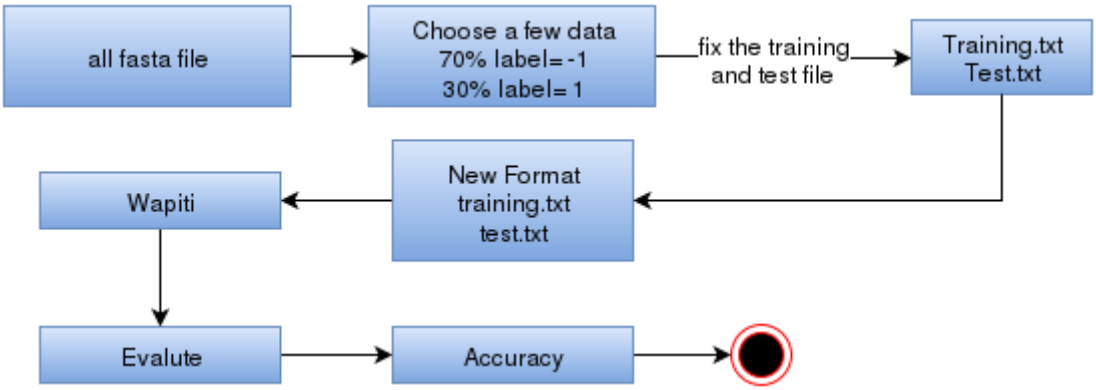

Image A3.1: CRF workflow.

As before in the N-gram graphs, the main idea is to use again a preprocessing process for our data set in order to extract the training set and the test set and put them as input into the Wapiti program, which will execute the CRF algorithm. Finally we will perform evaluation and we will get our results.

The data are aligned to the pattern of the acceptor (AG) or the donor (GT) respectively. From these files we choose some data for our experiments. As in the N-gram graphs, we could choose randomly 40.000 sequences from the Fasta file, in which 70% are no-labeled and 30% labeled data, in order to represent real world data. Each row in the new file contains again the sequence and in the end the label for each sequence is being presented.

TABLE OF ABBREVIATIONS

DNA	Deoxyribonucleic acid
RNA	Ribonucleic acid
mRNA	Messenger Ribonucleic acid
pre-mRNA	Precursor Messenger Ribonucleic acid
HMM	Hidden Markov Models
CRF	Conditional Random Field
SVM	Support Vector Machines
ANN	Artificial Neural Networks
EKPA	National and Kapodistrian University of Athens
FGA	Feature Generation Algorithm
NBT	Naïve Bayes Tree
AFMS	All Features Majority Strategy
NGG	N-gram Graph
RBF	Radial Basis Function
Libsvm	Library Support Vector Machines
KNN	K-Nearest Neighbors
auROC	area under the Receiver Operator Characteristic Curve

REFERENCES

[1] G. Schweikert, C. Widmer, B. Schölkopf, and G. Rätsch, An empirical analysis of domain adaptation algorithm for genomic sequence analysis, Advances in Neural Information Processing Systems, 2008, pp. 1433-1440.

[2] N. Herndon and D. Caragea, Empirical Study of Domain Adaptation Algorithms on the Task of Splice Site Prediction, Biomedical Engineering Systems and Technologies, Springer, vol. 511, Jan. 2016, pp 195-211.

[3] G. Giannoulis, A. Krithara, C. Karatsalos, G. Paliouras, Splice site recognition using transfer learning, Springer, 2014.

[4] S. Sonnenburg, G. Schweikert†, P. Philips, J. Behr and G. Rätsch, Accurate splice site prediction using support vector machines, BMC Bioinformatics, 2007.

[5] D. Wei , W. Zhuang , Q. Jiang and Y. Wei , A new classification method for human gene splice site prediction, Health Information Science, Springer, 2012, pp. 121-130.

[6] S. Pan and Q. Yang, A Survey on Transfer Learning, IEEE Transactions on knowledge and data engineering, no. 10, vol. 22, Oct. 2010, pp. 1345-1359.

[7] G. Giannakopoulos, Automatic Summarization from Multiple Documents, Phd Thesis, Department of Information and Communication Systems Engineering, University of the Aegean, 2009.

[8] M. Mort et al. MutPred Splice: machine learning-based prediction of exonic variants that disrupt splicing, Genome Biology, 2014.

[9] Ν. Αλαχιώτης, «Μελέτη, Σχεδιασμός και Υλοποίηση της Συνάρτησης Φυλογενετικής Πιθανοφάνειας σε Αναδιατασσόμενη Λογική», Διπλωματική Εργασία, Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών, Πολυτεχνείο Κρήτης.

[10] J. Vinson, D. Decaprio, M. Pearson, S.Luoma, and J. Galagan, Comparative Gene Prediction using Conditional Random Fields, Advances in Neural Information Processing Systems 19 (NIPS'06), 2007, pp. 1441-1448.

[11] C. Sutton and A. McCallum, An Introduction to Conditional Random Fields, Foundations and Trends in Machine Learning, no. 4, vol. 4, 2011, pp. 267–373.

[12] M. Doherty, Gene Prediction with Conditional Random Fields, Master Thesis, Department of Electrical Engineering and Computer Science, Broad Inst., June 2007.

[13] C. Widmer and G. Ratsch, Multitask learning in computational biology, JMLR, 2012, pp.207-216.

[14] A. Møller, T. Asp, P. Holm, M. Palmgren, Phylogenetic analysis of P5 P-type ATPases, a eukaryotic lineage of secretory pathway pumps, Molecular Phylogenetics and Evolution, 2007, pp. 619–634.

[15] G. Bari et al., Survey on Nucleotide Encoding Techniques and SVM Kernel Design for Human Splice Site Prediction, ibc, Dec. 2012.

[16] G. Rätsch and S. Sonnenburg, Accurate Splice Site Prediction for Caenorhabditis Elegans, MIT Press series on Computational Molecular Biology, MIT Press, 2004, p. 277-298.

[17] U. Kamath et al., An evolutionary algorithm approach for feature generation from sequence data and its application to DNA splice site prediction, IEEE/ACM Trans Comput. Biol. Bioinform., 2012, pp.1387-1398.

[18] "Chemistry of Life",2007; http://chemistryolife.blogspot.gr/2007/12/splice-site_06.html, [accessed 22/2/16].

[19] F. Brinkman et. al., Phylogenetic Analysis, Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins, 2001, pp. 322-358.

[20] P. Liò and N. Goldman, Models of Molecular Evolution and Phylogeny. 1998, pp. 1233-1244.

[21] Phylogenetics: DNA Protocol, 2013; http://bcrc.bio.umass.edu/intro/content/phylogenetics-dna-protocol, [accessed 22/2/16].

[22] D. Polychronopoulos et. al., Analysis and Classification of Constrained DNA Elements with N-gram Graphs and Genomic Signatures, Springer, vol. 8542, 2014, pp. 220-234.

[23] A. Arnold, R. Nallapati and W. Cohen, A Comparative Study of Methods for Transductive Transfer Learning, IEEE, 2007, pp. 77-82.

[24] "Minor spliceosome", Wikipedia, 2014; https://en.wikipedia.org/wiki/Minor_spliceosome, [accessed 22/2/16].

[25] A. Culotta et al., Gene Prediction with Conditional Random Fields, 2005.

[26] H. Wallach, Conditional Random Fields, 2005; http://www.inference.phy.cam.ac.uk/hmw26/crf/, [accessed 22/2/16].

[27] "What are conditional random fields", 2013; https://prateekvjoshi.com/2013/02/23/what-are-conditional-random-fields/, [accessed 22/2/16].

[28] T. Lavergne, Wapiti, 2008; https://wapiti.limsi.fr/manual.html#description [accessed 22/2/16].

[29] A. Stanescu et al., Predicting alternatively spliced exons using semi-supervised learning, IJDMB, vol. 14, 2016.

[30] “K-Mer”, Wikipedia; https://en.wikipedia.org/wiki/K-mer, [accessed 22/2/16].
[31] “Multiple Sequence Alignment”, EMBL-EBI; http://www.ebi.ac.uk/Tools/msa/, [accessed 22/2/16].
[32] A. Arnold, R. Nallapati and W. Cohen, A Comparative Study of Methods for Transductive Transfer Learning, IEEE, 2007, pp. 77-82.
[33] “Top five classifiers to try first”, Stack Exchange; http://stats.stackexchange.com/questions/7610/top-five-classifiers-to-try-first, [accessed 22/2/16].