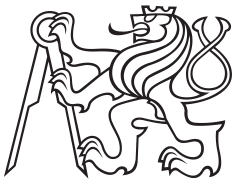
Master's Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Explainable and Transferable Fungal Intron Models

Barbora Mašková

Supervisor: doc. Ing. Jiří Kléma, Ph.D.
Field of study: Medical Electronics and Bioinformatics
Subfield: Bioinformatics
January 2024

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Mašková Barbora** Personal ID number: **483652**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Medical Electronics and Bioinformatics**

Specialisation: **Bioinformatics**

II. Master’s thesis details

Master’s thesis title in English:

Explainable and transferable fungal intron models

Master’s thesis title in Czech:

Vysvětlitelné a přenositelné modely intronů hub

Guidelines:

1. Get acquainted with the existing neural network based fungal intron removal models.
2. Learn about deep neural network explanation methods.
3. Carry out a literature search for methods ad 1 and 2.
4. Explain the existing neural fungal intron models.
a. Find out how far they match theoretical intron models.
b. Find out how far they differ for different taxonomical classes.
5. Optimize intron detection in metagenomes.
a. The goal is to detect as many fungal species as possible with application as few specific models as possible.

Bibliography / sources:

Eraslan, G., Avsec, Ž., Gagneur, J., & Theis, F. J. (2019). Deep learning: new computational modelling techniques for genomics. Nature Reviews Genetics, 20(7), 389-403.
Jeyakumar, J. V., Noor, J., Cheng, Y. H., Garcia, L., & Srivastava, M. (2020). How can i explain this to you? an empirical study of deep neural network explanation methods. Advances in Neural Information Processing Systems, 33, 4211-4222.
Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable AI: A review of machine learning interpretability methods. Entropy, 23(1), 18.

Name and workplace of master’s thesis supervisor:

doc. Ing. Jiří Kléma, Ph.D. Intelligent Data Analysis FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **08.08.2023** Deadline for master's thesis submission: _____

Assignment valid until: **16.02.2025**

doc. Ing. Jiří Kléma, Ph.D. Supervisor’s signature	Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature

Acknowledgements

I would like to thank my supervisor, doc. Ing. Jiřı Kléma, Ph.D., for his patience and all his support and guidance. I would also like to thank Ing. Anh Vu Le for providing me with his code for creating sequence logos. In addition, I would like to thank my family, friends and colleagues who supported me during the difficult times.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of the university thesis.

In Prague, ...

Signature: .....

Abstract

The objective of this thesis is to analyse the recurrent convolutional neural network that has been previously proposed for detecting splice sites in the fungal kingdom and to provide an explanation by employing sequence logos. The main goal is to minimise the number of models required to obtain complete coverage, taking into account time complexity and computational capacity.

Initially, a total of 19 pairs of models were generated, with one model representing the donor and another model representing the acceptor. These pairings were formed based on the taxonomical classification.

Subsequently, two pipelines were created: one for augmentation and another for the purpose of transfer learning. Upon establishing the optimal parameter settings, a thorough evaluation of the models was conducted, revealing the transfer model to be the superior choice.

While comparing the models to determine the optimal number required, an issue with transfer learning occurred. The issue lay in the fact that the application of transfer learning resulted in an improvement in the F1 score of the particular model in question, but led to a decrease in the scores of the other models. Therefore, different kinds of models were chosen. As a result, 9 models have been selected that meet the criteria for classifying the entire fungal world.

In order to justify this decision and explain the selected models, an explanatory technique known as Sequence logos was chosen. Based on these logos, it was concluded that the models were selected appropriately.

Keywords: Fungi, RCNN, Transfer Learning, Sequence Logo, Augmentation, Splice Site Recognition, Explanation methods, Motifs

Supervisor: doc. Ing. Jiří Kléma, Ph.D.

Abstrakt

Cílem této práce je analyzovat implementovanou rekurentní konvoluční neuronovou síť navrženou pro detekci intronů v metagenomech hub a poskytnout vysvětlení za pomoci sekvenčních log. Hlavním cílem je najít minimální počet modelů potřebných k dosažení úplného pokrytí variability druhové rozmanitosti v říši hub s přihlédnutím k časové složitosti a výpočetní náročnosti.

Zpočátku bylo vytvořeno celkem 19 párů modelů - jeden pro donor a jeden pro akceptor. Tyto dvojice byly vytvořeny na základě taxonomické klasifikace.

Následně byly vytvořeny dvě metody učení modelů: jedna za pomoci augmentace dat a druhá za použití transfer learning. Po nalezení optimálních hodnot parametrů těchto metod bylo provedeno důkladné porovnání jednotlivých modelů mezi sebou. Z tohoto porovnání vyšel model natrénovaný pomocí transfer learningu jako nejlepší volba.

Při porovnání modelů k určení nejmenšího počtu potřebných modelů ke klasifikaci říše hub, byl nalezena limitace použití transfer learning. Model doučený pomocí transfer learning sice lépe hodnotil doučenou část, avšak u zbytku jeho schopnost hodnotit klesla. Toto vedlo k výběru modelů naučených jinou metodou. Celkem bylo ze všech vybráno 9 modelů, které s velkou přesností klasifikují celou říši hub.

Pro podpoření tohoto rozhodnutí a vysvětlení vybraných modelů byla vybrána technika zvaná Sekvenční loga. Na základě těchto log a jejich porovnání se dospělo k závěru, že modely byly vybrány vhodně.

Klíčová slova: Houby, RCNN, Transfer Learning, Sekvenční loga, Augmentace dat, Detekce intronů, Metody vysvětlování, Motivy

Překlad názvu: Vysvětlitelné a přenositelné modely intronů hub

Contents

Figures

Tables

Chapter 1

Introduction

Fungi have been around for a very long time, with fossils dating back all the way to 460 to 455 million years ago. However, molecular data suggest that fungi may have originated over a billion years ago, much earlier than previously thought based on the fossil record. While the exact number of fungal species on Earth is unknown, it is known that at least 99,000 species have been identified, and about 1,200 new species are discovered annually. It's estimated that there are likely around 1.5 million fungal species in total [6].

For understanding and better, faster classification of the unknown, there is an effort to develop methods that detect introns with the use of Neural Networks or Machine Learning.

This thesis builds upon previous works dedicated to developing models for automated fungal intron detection. Denis Baručić worked on support vector machine intron detection models in his thesis [2]. Models of the same type were used in the study by Le et al. [26]. The paper confirmed that intron detection significantly improves the annotation of environmental metagenomes, particularly by increasing the proportion of newly predicted genes. Martin Indra addressed the same task with neural network detection models [20]. These models proved to enhance the process of intron detection both in terms of effectiveness and efficiency.

The existing solution offers only two general models for splice sites, which are afterwards merged into potential introns. These introns are then filtered out using a simple algorithm that resolves overlaps based on a scoring system.

This technique achieves favourable outcomes, however exclusively for species belonging to a well-represented phylum, as the overall models are incapable of learning the intricate characteristics of certain minor phyla or classes.

Therefore, this thesis aims to conduct a comprehensive examination of the taxonomy of the fungal kingdom, exploring the potential for phylum or even class models. The primary objective of this thesis is to examine the previously proposed neural network models and attempt to explain them. The additional goal is to conduct a comparative analysis of the models, identify the areas of similarity between them, and reduce the amount that is used in intron detection in regard to effectiveness, time complexity and computational needs.

1.1 Text Structure

This Chapter 1 outlines the fundamental basis of the thesis by introducing the theme and providing an overview of its objectives. In Chapter 2, the attention turns to context, carefully examining the theoretical aspects of splice sites and introns in biology. It also explores the intricacies of Neural Networks (NN) while also describing the NN used in this thesis, and clarifies topics such as augmentation, transfer learning, and sequence logos, as well as looks into the current research on these topics. Chapter 3 focuses on the data, offering valuable information about the specific sorts of data utilised, their formats, and an in-depth examination of taxonomy. Chapter 4 explains the statistical tools used to assess the generated models.

In Chapter 5, a systematic method is followed to explain the process of creating models. The process begins with the development of generic models for acceptor and donor, then advances to phylum models, and finally concludes with the creation of class-specific models. Chapter 6 moves to model training approaches, with a focus on augmentation and transfer learning, and explores their practical implementation. Chapter 7 provides complete descriptions of experiments, including parameter tuning for augmentation and transfer learning, as well as a thorough comparison of different models.

Chapter 8 centres around model description, specifically logo sequences, and comparing them across models. Chapter 9 provides a comprehensive examination and analysis of the experimental data. It thoroughly evaluates all the models and makes a recommendation for the required number of models. Additionally, it investigates the sequence logos. Chapter 10 serves as a definitive conclusion, summarising the entire thesis and offering insights

into possible potential research directions.

Chapter 2

Background

2.1 Fungi

The fungi emerged as a unique group of single-celled eukaryotes during the Precambrian period. Recent estimates of the origins of the fungal kingdom, determined by the examination of the molecular clock, vary between 760 million years ago and 1.06 billion years ago [33].

Based on the phylogenetic relationships of fungi, as indicated by the analysis of 18S ribosomal DNA, the classification of fungi has been revised by [19]. They have placed fungi under three kingdoms: Protozoa, Chromista, and Fungi, all falling under the domain of Eukarya.

This thesis will exclusively concentrate on the Kingdom Fungi, commonly known as "true fungi". The members of the kingdom possess the following characteristics: (1) they obtain nutrition through absorption; (2) their somatic phase consists of either single cells or filaments; (3) their cell walls contain chitin and beta-glucans; (4) they have only whiplash flagella; (5) their mitochondria have flattened cristae; and (6) they have peoxisomes and Golgi bodies present [10].

The most recent taxonomy categorises the known true fungi into nine primary lineages: Opisthosporidia, Chytridiomycota, Neocallimastigomycota, Blastocladiomycota, Zoopagomycota, Mucoromycota, Glomeromycota,

Ascomycota, and Basidiomycota (Figure 2.1) [34]. Besides the previously mentioned phyla, there exists a group of animal parasites known as microsporidians, comprising around 1000 species [51]. The incorporation of microsporidians into the Fungi kingdom is further substantiated by further data derived from genome structure and phylogenetic investigations [4]. They are classified as a separate phylum known as Microspora.

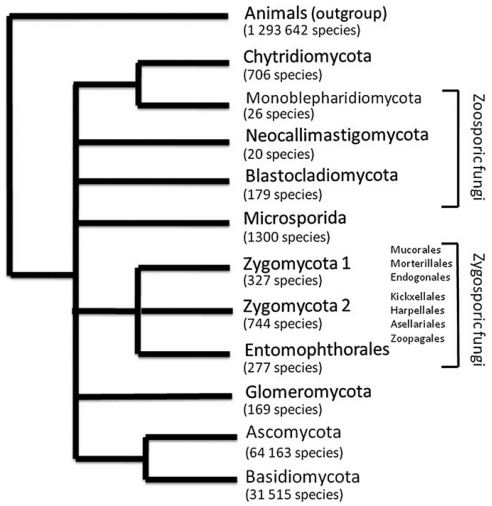


Figure 2.1: Fungal phyla and approximate number of species in each group [46]

Fungal cells share many of the same organelles as other eukaryotic cells. Fungal nuclei are typically small, less than 2 µm in diameter, and can compress and stretch to fit through septal pores and into developing spores. Fungi usually have between 6 and 21 chromosomes that code for anywhere from 6,000 to almost 18,000 genes. The size of fungal genomes varies widely, ranging from 8.5 megabase pairs to just over 400 Mb in filamentous fungi. Compared to other eukaryotic organisms, fungal genomes are relatively small, averaging approximately 1% the size of mammalian genomes and only 1.3 times the size of the largest known bacterial genome [6].

2.2 Pre-mRNA splicing

Genes are made up of DNA, and in some cases, RNA in the case of certain viruses. DNA is a long chain-like molecule composed of small building blocks called monomers. There are four types of monomers in DNA: adenine (A), guanine (G), thymine (T), and cytosine (C). DNA molecules are usually double-stranded, with two complementary strands. In this structure, adenine in one strand always pairs with thymine in the other strand, and guanine in

one strand always pairs with cytosine in the other strand. This strict pairing rule between A and T, and G and C, forms the basis for gene replication [47].

Eukaryotes possess several nuclear genes that encode proteins, which are often interrupted by one or more introns. These introns must be accurately removed from the initial gene transcript before the RNA is transferred to the cytoplasm for translation [24].

The spliceosome, consisting of five snRNPs (small nuclear ribonucleoproteins) named U1, U2, U4, U5, and U6, eliminates introns from the pre-mRNA. Each snRNP is comprised of a tiny RNA molecule that is associated with proteins. Accurate identification of introns is necessary for proper splicing [40].

There are three signals that are involved in guiding splicing: the 5' splice site (5'ss) located at the beginning of the intron, the polypyrimdine tract/3' splice site (PPT-3'ss) lying at the end of the intron, and a branch site (BS) situated before the PPT-3'ss [17] around 18–40 nucleotides upstream [7].

The splicing reaction consists of two sequential phases. The first step involves the breaking down of the 5'–3' phosphate linkage, which connects the 5' exon to the first nucleotide of the intron. The second step involves the breakage of the linkage between the last nucleotide of the intron and the 3' exon, specifically at the 3' splice site. This generates a circular molecule featuring a tail, known as the intron lariat, and a branched structure at the branch point. The unbound 3' hydroxyl on the 5' exon is utilised to initiate an attack and break the bond between the last nucleotide of the intron and the 3' exon, liberating the intron in the form of a lariat. The debranching enzyme simplifies this process into a linear form, which is thus likely to be quickly broken down by exonucleases. The earliest stages entail the identification of the 5' splice site by the U1 snRNA and the attachment of U2 snRNA to the branch-point region, facilitated by SR-proteins. The U4 and U6 small nuclear ribonucleic acids (snRNAs) form a spliceosome by pairing together as a duplex, which brings the 5' splice site and branch point into close proximity. The two catalytic steps are referred to as transesterification reactions. The process of assembling and disassembling the spliceosome involves several ATPases, most of which are RNA helicases. These proteins utilise the energy from ATP hydrolysis to catalyse structural rearrangements in the spliceosome [1]. This whole process is depicted in Figure 2.2.

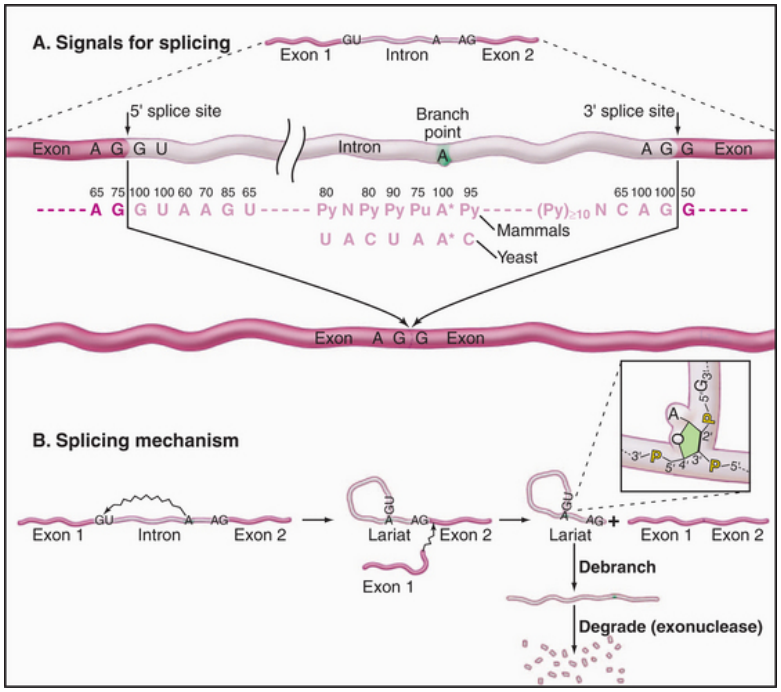


Figure 2.2: Process of pre-mRNA splicing and its signals [1]

2.3 Neural Networks

The neural network is a computational structure that draws inspiration from the intricate workings of the human brain. Just as the brain's network of interconnected neurons enables it to process information, learn, and make decisions, neural networks aim to replicate these capabilities in the realm of artificial intelligence. The equivalents of biological neurons are called units or nodes. Synapses are represented by individual numerical values or weights. These weights are applied to each input signal, multiplying them, before transmitting them to the corresponding cell body [12]. An example of a simple neuron is shown in Figure 2.3.

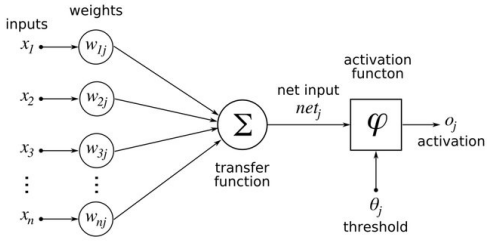


Figure 2.3: Example of simple neuron [21]

The activation of neuron can be mathematically expressed using Formula 2.1.

$$o_j = \phi((\sum x_i \cdot w_{ij}) - b_j) \tag{2.1}$$

The equation describes activation of the j-th neuron, where ϕ: $\mathbb{R} \to \mathbb{R}$ represents the activation function, x_i is an i-th input, $w_{ij} \in \mathbb{R}$ is a weight of the connection from neuron i to neuron j and $b_j \in \mathbb{R}$ is a bias of neuron j.

Most neural networks are organized into groups of units called layers, and in many neural network architectures, these layers are structured in a sequential manner, where each layer's output depends on the preceding. In architectures based on these sequential structures, the primary architectural decisions involve determining the network's depth and the width of each layer. It is worth noting that even a network with just a single hidden layer can effectively learn and fit the training data [14]. An example of a basic neural network structure can be found in Figure 2.4.

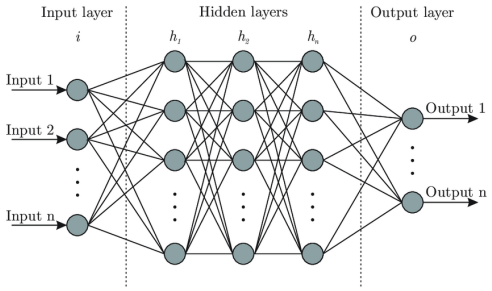


Figure 2.4: Basic Neural Network Structure [44]

2.3.1 Neural Networks and Sequential pattern recognition

At first, Markov chains [38] were used to depict sequential patterns, but they encountered challenges in representing intricate interactions. Deep neural networks have significantly impacted sequential recognition, with the Recurrent Neural Network (RNN) becoming an established model. Long short-term memory (LSTM) is a type of artificial recurrent neural network (RNN) architecture that is commonly employed in the field of deep learning. LSTM, in contrast to conventional feed-forward neural networks (CNN) [28], incorporates feedback connections. It has the capability to process not just individual data points (such as pictures) but also complete sequences of data (such as DNA or RNA) [45].

The merging of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) has been employed to address the limitations of each

network. The Recurrent Convolutional Neural Network (RCNN) merges the recurrent structure of LSTM to capture complex long-term connections and the convolutional operation of CNN to reveal local sequential patterns within hidden states [18].

2.3.2 Overview of used Neural Network

The core framework utilised in this thesis was first introduced by [20]. His thesis provides a thorough examination of the complexities and reasons behind the parameters, methods, and so on. However, this part focuses only on the fundamental elements. The architecture of the model is illustrated in Figure 2.5, which displays a Recurrent Convolutional Neural Network (RCNN) configuration consisting of four convolution layers, Leaky rectified linear activation functions (Leaky ReLU), Bidirectional RNN with long short-term memory, and Dropout layer for preventing overfitting.

Stochastic gradient descent (SGD) emerged as the preferred optimizer during the training phase, showcasing superior performance. The learning rate was set to 0.01 and was dynamically adjusted by multiplying it by a factor of 0.2 after every epoch. The implementation of this adaptive learning rate technique results in improved convergence and fine-tuning of performance.

The optimisation procedure utilised the binary cross-entropy loss function. The batch size was deliberately selected at 16, based on current research that recommends smaller batch sizes due to their demonstrated improvement in results [30].

Figure 2.5: Model Architecture by [20]

2.4 Augmentation

Data augmentation has been an extensively researched field in the domain of machine learning. It refers to a collection of algorithms that generate artificial data based on an existing dataset. The synthetic data usually includes minor alterations in the data that the model's predictions should be unaffected by. Synthetic data can also depict mixtures of remote instances that would be immensely tough to grasp otherwise. Data augmentation is a highly valuable tool for influencing the training of deep neural networks. This is mostly because the transformations are well understood and there are plenty of chances to examine the model's failures. The most typical application of Data Augmentation is to prevent overfitting. Deep Neural Networks, in the absence of augmentation or regularisation, are vulnerable to acquiring incorrect associations and memorising high-frequency patterns that are challenging for humans to identify [43].

Augmentation of DNA data is essential in the field of Deep Neural Networks (DNNs) for genomics and bioinformatics tasks. Data augmentation involves generating new training samples from current data by applying various transformations, thus expanding the dataset. In the domain of DNA data, augmentation techniques are used to provide diverse training instances from genomic sequences. Here are some common methods for augmenting DNA data for DNNs:

- Mutation: a transformation in which single nucleotide mutations are applied at random to a wild-type sequence,
- Translocation: a transformation that chooses a breakpoint in the sequence at random (resulting in two segments) and then switches the order of the two sequence segments,
- Insertion: a transformation in which a random DNA sequence (of unknown length) is randomly inserted into a wild-type sequence.
- Deletion: a transformation in which a random, contiguous section of a wild-type sequence is removed,
- Inversion: a transformation where a random subsequence is replaced by its reverse complement,
- Reverse complement: is a transformation that replaces a random subsequence with its reverse complement,
- Noise Injection: a transformation where Gaussian noise is added to the sequence [27].

Given that this is not a novel technique, several groups (such as Lee et al.[27], Minot et al. [31], Zhang et al. [56], Tyekucheva et al. [49] and Lacan et al. [25]) have already conducted investigations, devised innovative approaches, and created libraries that aid its implementation. One of these groups [27] developed a library called *evoaug*. The image below depicts their explanation of the approach and the augmentation process.

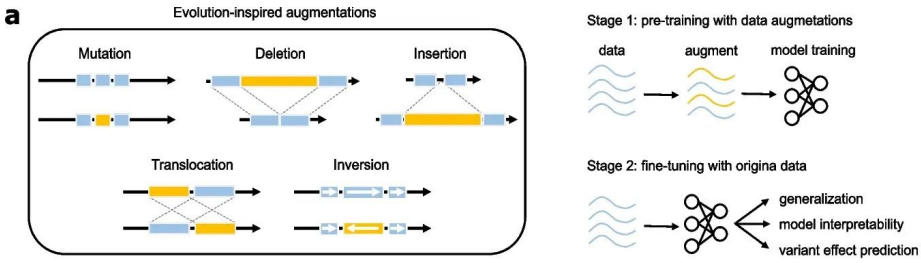


Figure 2.6: Schematic of evolution-inspired data augmentations (left) and the two-stage training curriculum (right) [27].

Because defining the ideal method of augmentation is beyond the scope of this thesis, the Noise injection concept was chosen among all the possible methods from *Evoaug* as it is the most fundamental. Furthermore, based on the results of [27] experimentation, it outperforms the model trained without it.

2.5 Transfer learning

Conventional machine learning techniques have demonstrated efficacy in several applications, although they remain limited in real-world situations. An optimal situation for machine learning involves having an adequate number of labelled training instances that closely match the distribution of the test data. Nevertheless, acquiring an appropriate amount of training data can prove to be costly, laborious, or unattainable. Semi-supervised learning may reduce the requirement for extensive labelled data by leveraging a substantial quantity of unlabelled data to enhance learning precision. Still, the task of gathering unmarked examples can be challenging, rendering conventional models inadequate. Transfer learning shows great promise in tackling this problem [57].

Transfer learning is a technique that enhances the performance of a learner in one domain by utilising knowledge from a closely related domain [52]. According to the article [42], it can mitigate overfitting by excluding undesired noise from the dataset. Transfer learning methods can be categorised into three primary subsets, which are determined by the distinct circumstances between the source and target domains and tasks:

1. *Inductive transfer learning*: the target task varies from the source task, despite the similarity or dissimilarity between the source and target domains. The model requires labelled data from the target domain to be trained,

2. *Transductive transfer learning*: the target and source tasks are identical, however the source and target domains differ. There is a lack of labelled data in the target domain, but there is an excess of labelled data in the source domain,

3. *Unsupervised transfer learning*: the target task is distinct from, yet associated with, the source task. There is a lack of labelled data in both the source and target domains during training [23].

2.5.1 Definition

In the following section, a short definition of domain, task and transfer learning is given.

Domain: The *domain* $\mathcal{D}$ is composed of two elements: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, ..., x_n\} \in \mathcal{X}$.

Task: For a given domain, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a *task* contains two elements: a label space $\mathcal{Y}$ and an objective prediction function $f(\cdot)$ (represented by $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$). The function $f(\cdot)$ is not directly observable but can be acquired through learning from the training data, which consists of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in \mathcal{Y}$. The function $f(\cdot)$ can be utilised for predicting the related label, $f(x)$, of a novel instance x. From a probabilistic perspective, the function $f(x)$ can be represented as $P(y|x)$.

Source and target domains: For simplicity, only the case of one source domain $\mathcal{D}_\mathcal{S}$, and one target domain $\mathcal{D}_\mathcal{T}$ will be considered. To be more precise, the data from the source domain are represented as $\mathcal{D}_\mathcal{S} = \{(x_{S_1}, y_{S_1}), ..., (x_{S_{n_S}}, y_{S_{n_S}})\}$, where $x_{S_i} \in \mathcal{X}_\mathcal{S}$ refers to the data instance and $y_{S_i} \in \mathcal{Y}_\mathcal{S}$ represents the corresponding class label. Similarly, the data from the target domain is expressed as $\mathcal{D}_\mathcal{T} = \{(x_{T_1}, y_{T_1}), ..., (x_{T_{n_T}}, y_{T_{n_T}})\}$, with the input x_{T_i} belonging to $\mathcal{X}_\mathcal{T}$ and the corresponding output $y_{T_i} \in \mathcal{Y}_\mathcal{T}$. In general, $0 \leq n_T \ll n_S$.

Transfer learning: Given a source domain $\mathcal{D}_\mathcal{S}$ and learning task $\mathcal{T}_\mathcal{S}$, a target domain $\mathcal{D}_\mathcal{T}$ and learning task $\mathcal{T}_\mathcal{T}$, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_\mathcal{T}$ using the knowledge in $\mathcal{D}_\mathcal{S}$ and $\mathcal{T}_\mathcal{S}$, where $\mathcal{D}_\mathcal{S}$ is not equal to $\mathcal{D}_\mathcal{T}$, or $\mathcal{T}_\mathcal{S}$ is not equal to $\mathcal{T}_\mathcal{T}$.

The definitions provided above originate from the survey [36].

2.5.2 Transfer learning in splice site recognition

Biological sequence analysis seeks to provide functional annotations to DNA segments, playing a crucial role in the comprehension of a genome. An instance of this is the recognition of splice sites based on the boundaries

between exons and introns, which is a challenging undertaking due to the multitude of potential alternative splicing occurrences [54].

For splice site recognition, there was a proposal by Giannoulis et al. [13] to employ unsupervised transfer learning. Transfer learning is employed to address the issue of inadequately annotated genomes by using knowledge from the well-annotated genome of another organism (source domain) and applying it to our poorly annotated genome (target domain). The proposed approach involves utilising an adapted variant of the K-means algorithm with a representation technique known as n-gram graphs. An alternative method for employing transfer learning is outlined in [32], which involves the utilisation of pre-trained models. Pre-trained models are neural networks that have been previously used in other tasks, allowing them to gain information that can be transferred and applied to new target data [55]. The study conducted by Schweikert et al. [41] examined the efficacy of several transfer learning methods (Figure 2.7) in addressing the mRNA splice site prediction problem.

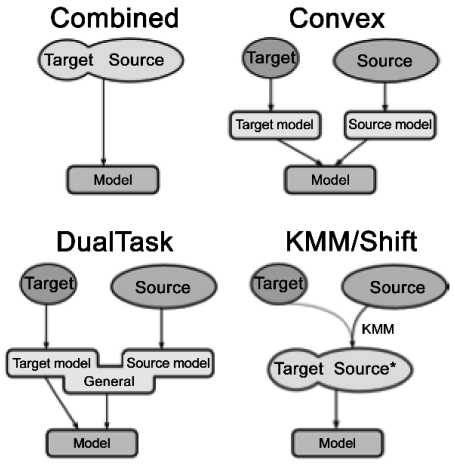


Figure 2.7: Four domain adaptation models (adopted from an invited talk by Gunnar Rätsch, Invited Talk at NIPS Transfer Learning Workshop, December 2009, Whistler, B.C. [41])

This thesis draws inspiration from the transfer learning approach proposed by [32]. The initial training of the base model was conducted using a dataset that is characterised by its substantial size. Subsequently, this model was employed to transfer its knowledge and create a new model using the dataset of phylum or class that exhibits sparse representation.

2.6 Explanation methods

Explanation methods have emerged as a crucial area of research in the field of machine learning, particularly in relation to complex models like deep neural networks (DNNs) due to their black-box nature [29]. They provide an explanation as an image, text, or other visual aid that accompanies a prediction to offer intuition into the underlying reasons for the model output (Figure 2.8). Approaches span contrasting styles that focus on different model elements, e.g., the training dataset or the learned feature representations. Model-transparent approaches highlight which particular input features triggered key activations within a model's weights. Model-agnostic methods such as treating the model as a black box and attempting to approximate the relationship between the input sample and the output prediction. Finally, example-based methods offer instances from the training dataset in an attempt to capture the relationship between a given test input and the underlying training data that contributed to the model's decision [22].

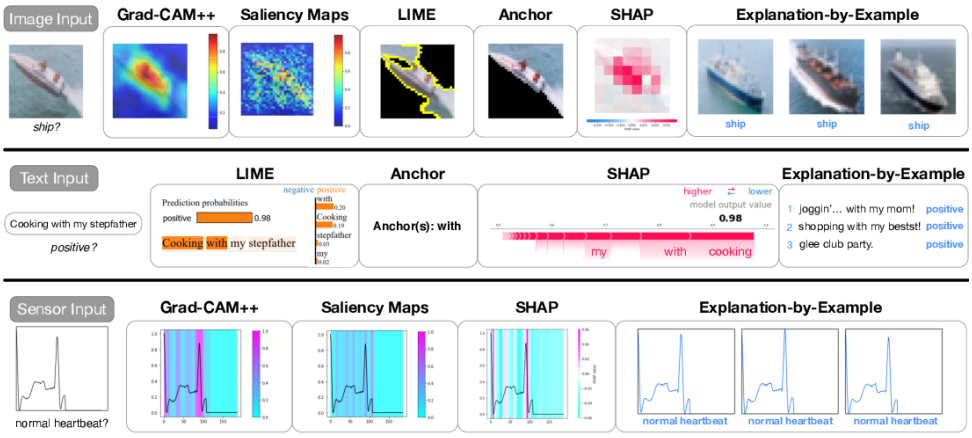


Figure 2.8: Illustration of analysed techniques for interpreting picture, text, and ECG input [22].

Deep models are frequently employed for scientific exploration, but their lack of transparency restricts their effectiveness. Consequently, there is a growing need for models that can be comprehended by people. This requirement is not exclusive to the field of bioinformatics, and there exist novel approaches to enhance the interpretability of machine learning models. Explainable artificial intelligence (XAI) has emerged as a discipline that has made conceptual progress which has the potential to be beneficial for bioinformatics applications, specifically in the realm of scientific exploration [37].

During the process of analysing biological sequences, such as DNA or protein sequences, a technique known as sequence logos came into being and has since gained widespread application.

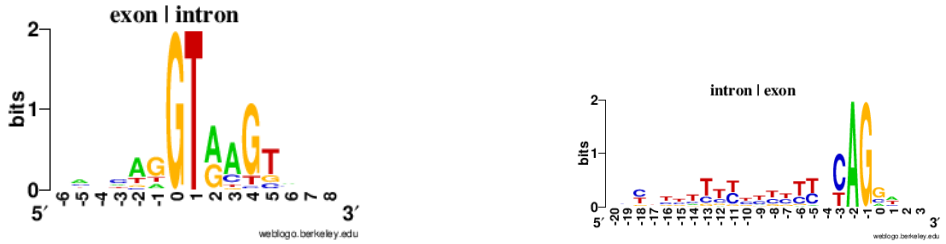
2.6.1 Sequence logo

Sequence logos are powerful graphical representations used to depict sequence motifs and patterns in biological sequences, such as DNA or protein sequences. They provide a visually intuitive way to analyse and interpret the information content and consensus sequences of aligned sequences [39].

Each logo is made up of stacks of symbols, with one stack representing each position in the sequence, respectively. The overall height of the stack is proportional to the amount of information that is included at that point, whereas the height of symbols inside the stack reflects the relative frequency of each amino acid or nucleic acid that is present at that position. The description of a binding site, for instance, that is provided by a sequence logo is typically more detailed and accurate than the description that would be provided by a consensus sequence [3].

In response to the widespread use of sequence logos, a great number of research was conducted, and a great number of libraries, packages, or web-tools were developed. Among the many examples that can be found on the internet, Crooks and al. [9] have developed a generator known as WebLogo; Tareen and al. [48] have created a Python library called Logomaker; Omar Wagih [50] contributed with an R package named as ggseqlogo; and Workman and al. [53] have built a web-based tool labelled as enoLOGOS.

Sequence logos can be used to graphically showcase the results of models, in our case, the splice sites, just as is shown in Figure 2.9.



(a) : Exon-Intron (Donor) Sites

(b) : Intron-Exon (Acceptor) Sites

Figure 2.9: Logos showing a small sample of Human intron-exon splice boundaries [8]

Chapter 3

Decription of Data

Source data were acquired from one of the predecessors. The data consists of FASTA files with the DNA sequence scaffolds of individual organisms and GFF files with DNA feature annotations [20].

3.1 File Formats

3.1.1 FASTA

The FASTA format is a text-based format used to represent nucleotide or peptide sequences. It employs single-letter codes to represent base pairs or amino acids. A FASTA-formatted sequence commences with a brief description on a single line, followed by subsequent lines containing the actual sequence data. The description line is differentiated from the sequence data by a ">" sign in the first column. It is advisable to ensure that the length of each line of text does not exceed 80 characters [16].

What follows is an example sequence in the FASTA format:

```
>scaffold_1
TTGGATAGGCGCCATAGCCCTCCATTGTGGGTGTTAGAACAAGGGCAATTCCTGCCACCTATACTGGCTA
```

```
GAGGCCTGAGTGGCCCAGATTAGCTTAGTATGATTACATAATGCTCCCTATAACACTGGCTGAGAAACAA
TAAGTTTCCTCAGCGATTCGTCTCTATCATTTGGGGATAATAGAATTGACTCGATCTCACTATTGCTAAT
```

3.1.2 GFF

The General Feature Format (GFF) is a text file with tab-delimited fields that contain information on all possible features that might be associated with a nucleic acid or protein sequence. This format can accommodate a wide range of elements, including CDS, microRNAs, binding domains, ORFs, and more. Regrettably, other iterations of the initial GFF format have emerged, resulting in a lack of compatibility among them. The most recent approved format, GFF3, has endeavoured to rectify some deficiencies that were absent in prior iterations [11].

An example of a GFF file:

```
scaffold_1 JGI exon 774 1123 . + . name "fgenesh1_kg.1_#_1_#_Locus4417v1rpkm26.65"; transcriptId 416145
scaffold_1 JGI CDS 1088 1123 . + 0 name "fgenesh1_kg.1_#_1_#_Locus4417v1rpkm26.65"; proteinId 416053; exonNumber 1
scaffold_1 JGI start_codon 1088 1090 . + 0 name "fgenesh1_kg.1_#_1_#_Locus4417v1rpkm26.65"
scaffold_1 JGI exon 1190 1606 . + . name "fgenesh1_kg.1_#_1_#_Locus4417v1rpkm26.65"; transcriptId 416145
scaffold_1 JGI CDS 1190 1606 . + 0 name "fgenesh1_kg.1_#_1_#_Locus4417v1rpkm26.65"; proteinId 416053; exonNumber 2
```

3.2 Taxonomy

The dataset is made up of 862 organisms, which are classified into 8 major phyla. Significantly, the phylum Ascomycota stands out as the most prevalent, making up over 50% of the overall sample. To obtain a comprehensive analysis of the number of organisms in each phylum, please consult Table 3.1.

Phylum	Count
Ascomycota	479
Basidiomycota	295
Blastocladiomycota	4
Cryptomycota	1
Chytridiomycota	21
Microsporidia	8
Mucoromycota	38
Zoopagomycota	16

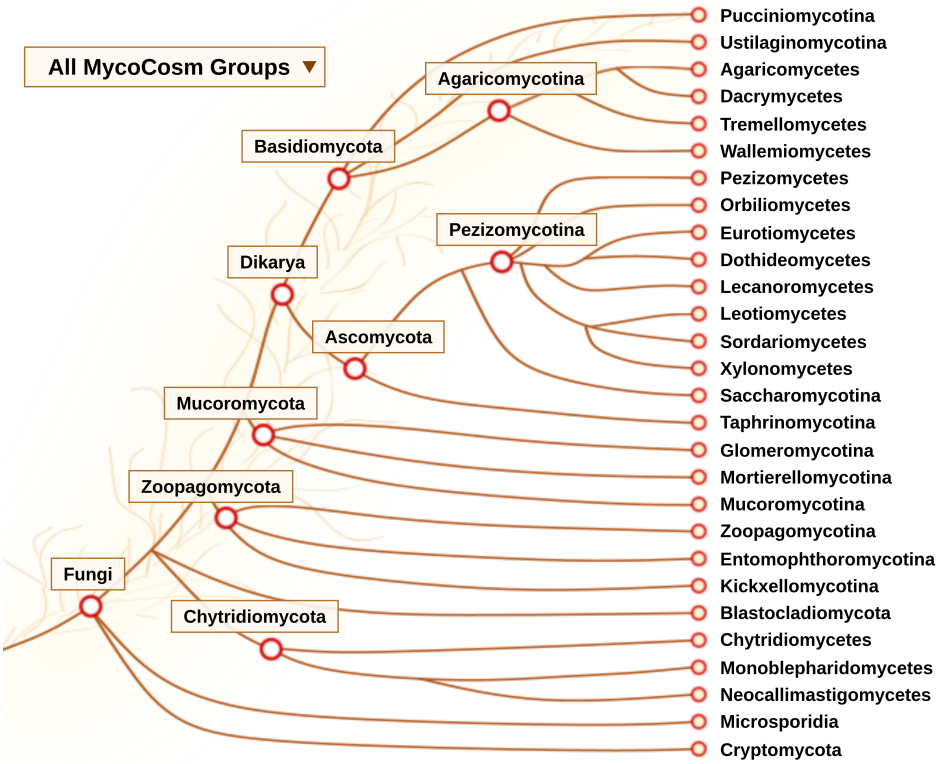
Table 3.1: Representation of phyla in the data

Taxonomy Rank	Taxonomic Count
Phylum	8
Class	46
Order	124
Family	307
Genus	566
Species	862

Table 3.2: Taxonomic distribution at various hierarchical ranks with corresponding counts of taxa.

Table 3.2 is a summary of the taxonomic variety in the data, displaying the number of detected species at various hierarchical levels. This breakdown is a useful resource for comprehending how organisms are distributed across different taxonomic levels. It serves as a basis for exploring the biological characteristics of the set of data and assists in making informed decisions when creating models.

The complete taxonomy of fungi, depicted as a phylogenetic tree, is available in the image below 3.1.



1

Figure 3.1: Classification of Fungi [15]

Chapter 4

Statistics

The evaluation process utilised several statistical measures, which include True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP), to assess the performance of the models. By employing the aforementioned parameters, crucial metrics such as True Positive Rate (TPR) 4.2 and Positive Predictive Value (PPV) 4.1, which are also referred to as recall and precision, respectively, were calculated.

$$PPV = \frac{TP}{TP + FP} \tag{4.1}$$

$$TPR = \frac{TP}{TP + FN} \tag{4.2}$$

These metrics offer valuable information about the models' capacity to accurately detect positive instances and the accuracy of positive identifications.

When rare minority classes are involved in creating experimental datasets, it is common practice to retain all available samples from this class while reducing the number of samples from the majority class. While this approach may be justifiable during training, it becomes problematic during testing because it disregards the true imbalance ratio that the classifier will encounter when deployed. Therefore, relying on precision computed directly on the test set in this scenario can be misleading [5]. Given these factors, the Precision computed directly on the test set may be deceptive, requiring an adaptation. The incorporation of priors (p_{test} and p_{real}) facilitates the achievement of a more accurate representation. The modified equation 4.3 is shown below.

$$PPV_{adjusted} = \frac{\frac{p_{real}}{p_{test}} \cdot TP}{\frac{p_{real}}{p_{test}} \cdot TP + \frac{1-p_{real}}{1-p_{test}} \cdot FP} \tag{4.3}$$

For the purpose of conducting a thorough model comparison, the F-score was calculated as a complete metric to evaluate accuracy. The F-score is a balanced evaluation metric that takes into account both false positives and false negatives and is calculated based on the precision and recall of the model. The equation 4.4 can be seen below.

$$F = \frac{2}{TPR^{-1} + PPV_{adjusted}^{-1}} \tag{4.4}$$

Chapter 5

Model creation

This chapter focuses on the creation of models, specifically examining the complex process of model development. It highlights the progression from general donor and acceptor models to more specific phylum models and eventually concludes with the development of taxonomy mixture models.

5.1 Donor and acceptor models

In order to improve understanding of the codes presented by [20], two distinct models were deliberately created, with each model designed for analysing donor and acceptor splice sites. During the training and evaluation stages, it became evident that producing a single, all-encompassing model required a significant amount of time. In order to improve the model's efficiency, a comprehensive evaluation of its performance within specific phyla was conducted. The results, as shown in Tables 5.1 and 5.2, highlight the difficulties of using a universal model, with F-scores for each phylum consistently below the 0.85 threshold.

Throughout the training and evaluation phase, a few significant obstacles became apparent. An error occurred when creating the dataset for the phylum Cryptomycota. Later on, when assessing this phylum, the decision to include it was reconsidered, as constructing a complete dataset was deemed unfeasible due to the scarcity of organisms available for this phylum.

Another noteworthy obstacle arose during the examination of the phylum Microsporidia. While the datasets were generated without any issues, the models faced challenges during their evaluation. At that point, the precise cause remained unknown. During the course of the investigation, a significant observation arose - the data related to Microsporidia primarily consisted of files for false splice sites, whereas true ones were only present in small quantities.

Based on these discoveries, it became clear that depending only on two broad models resulted in less than ideal outcomes due to their inherent generic nature. As a result, this understanding led to a change in strategy, recognising the requirement for a more refined and specialised model setup to attain superior results.

	ASCOMYCOTA	BASIDIOMYCOTA	BLASTOCLADIOMYCOTA	CHYTRIDIOMYCOTA	MUCOROMYCOTA	ZOOPAGOMYCOTA
TN	92.6	91.2	93.5	85	92.9	91.6
TP	90.6	88.1	92.1	87.2	90.6	80.6
FN	9.4	11.9	7.9	12.8	9.4	19.4
FP	7.4	8.8	6.5	15	7.1	8.4
TPR	0.906	0.881	0.921	0.872	0.906	0.806
PPV	0.92	0.91	0.93	0.85	0.93	0.91
adjusted PPV	0.75	0.71	0.78	0.59	0.76	0.71
F	0.82	0.79	0.84	0.71	0.83	0.75

Table 5.1: Evatuation of the whole donor model on individual phyla

	ASCOMYCOTA	BASIDIOMYCOTA	BLASTOCLADIOMYCOTA	CHYTRIDIOMYCOTA	MUCOROMYCOTA	ZOOPAGOMYCOTA
TN	92.6	91.2	93.5	85	82.5	92.7
TP	90.6	88.1	92.1	87.2	96.2	85.2
FN	9.4	11.9	7.9	12.8	3.8	14.8
FP	7.4	8.8	6.5	15	17.5	7.3
TPR	0.906	0.881	0.921	0.872	0.962	0.852
PPV	0.92	0.91	0.93	0.85	0.85	0.92
adjusted PPV	0.75	0.71	0.78	0.59	0.58	0.74
F	0.82	0.79	0.84	0.71	0.72	0.79

Table 5.2: Evatuation of the whole acceptor model on individual phyla

5.2 Phyla models

When the shortcomings of the generic models were brought to light, it became abundantly clear that in order to get better outcomes, it was necessary to employ models that were both more precise and more specialised. It was ultimately decided that the advancement of the research strategy would be in constructing and training models tailored to specific phyla. The individual phyla are listed below.

- Ascomycota
- Basidiomycota
- Blastocladiomycota

- Chytridiomycota
- Mucoromycota
- Zoopagomycota

For each of the indicated phyla, two separate models for donor and acceptor splice sites were created. Following that, a comprehensive assessment was carried out on every phylum, which involved self-assessment and evaluation of other phyla, and the results are reported in detail in Tables 5.3 and 5.4. It is important to mention that, although there were difficulties in training models for the phylum Microsporidia, the datasets were included in the evaluation process in order to understand how other models are able to work with them.

Although the F-scores of these models occasionally exceeded the 0.85 barrier, none of them reached a value higher than 0.90. One noteworthy finding was that there was no one superior model that could be applied to all groups of organisms. Each model showed different levels of success, and even when analysing a model within its group, it was difficult to determine its optimality.

These findings led to a fundamental change in thinking, motivating the investigation of models specifically designed for each class and utilising the transfer learning approach to overcome the 0.90 threshold. To address the difficulties presented by datasets with a restricted number of species, a mechanism for augmentation was implemented to improve the performance and generalisation abilities of the model.

5.3 Class models

Before implementing class-specific models, it was considered essential to conduct a thorough examination of the distribution of organisms within each class. This strategic decision was made based on the knowledge obtained from the difficulties faced with the phylum Cryptomycota, where the creation of a dataset proved unattainable due to it consisting only of one organism. Figure 5.1 visually illustrates the distribution patterns of species across classes.

Phyla	Class	number of	sub_phyla
Basidiomycota	Agaricomycetes	219	----------
Basidiomycota	Agaricostilbomycetes	3	Pucciniomycotina
Basidiomycota	Atractiellomycetes	1	Pucciniomycotina
Zoopagomycota	Basidiobolomycetes	1	Entomophthoromycotina
Blastocladiomycota	Blastocladiomycetes	3	x
Chytridiomycota	Chytridiomycetes	13	----------
Basidiomycota	Classiculomycetes	1	Pucciniomycotina
Ascomycota	Coniocybomycetes	1	Pezizomycotina
Basidiomycota	Cystobasidiomycetes	3	Pucciniomycotina
Basidiomycota	Dacrymycetes	4	Agaricomycotina
Zoopagomycota	Dimargaritomycetes	1	Kickxellomycotina
Ascomycota	Dothideomycetes	111	----------
Zoopagomycota	Entomophthoromycetes	2	Entomophthoromycotina
Ascomycota	Eurotiomycetes	135	----------
Basidiomycota	Exobasidiomycetes	11	Ustilaginomycotina
Basidiomycota	Geminibasidiomycetes	1	Wallemiomycotina
Mucoromycota	Glomeromycetes	1	Glomeromycotina
Zoopagomycota	Kickxellomycetes	6	Kickxellomycotina
Ascomycota	Lecanoromycetes	4	Pezizomycotina
Ascomycota	Leotiomycetes	31	----------
Basidiomycota	Malasseziomycetes	1	Ustilaginomycotina
Basidiomycota	Microbotryomycetes	11	Pucciniomycotina
Basidiomycota	Mixiomycetes	1	Pucciniomycotina
Basidiomycota	Moniliellomycetes	1	Ustilaginomycotina
Chytridiomycota	Monoblepharidomycetes	2	x
Mucoromycota	Mortierellomycetes	3	Mortierellomycotina
Mucoromycota	Mucoromycetes	31	----------
Chytridiomycota	Neocallimastigomycetes	4	x
Ascomycota	Neolectomycetes	1	Taphrinomycotina
Ascomycota	Orbiliomycetes	2	Pezizomycotina
Ascomycota	Pezizomycetes	18	----------
Blastocladiomycota	Physodermatomycetes	1	x
Ascomycota	Pneumocystidomycetes	1	Taphrinomycotina
Basidiomycota	Pucciniomycetes	6	Pucciniomycotina
Ascomycota	Saccharomycetes	40	
Ascomycota	Schizosaccharomycetes	4	Taphrinomycotina
Ascomycota	Sordariomycetes	120	----------
Basidiomycota	Spiculogloeomycetes	1	Pucciniomycotina
Ascomycota	Taphrinomycetes	3	Taphrinomycotina
Basidiomycota	Tremellomycetes	15	----------
Mucoromycota	Umbelopsidomycetes	3	Mucoromycotina
Basidiomycota	Ustilaginomycetes	8	Ustilaginomycotina
Basidiomycota	Wallemiomycetes	2	Wallemiomycotina
Ascomycota	Xylonomycetes	2	Pezizomycotina
Zoopagomycota	Zoopagomycetes	5	Zoopagomycotina
Microsporidia	No class	8	

Figure 5.1: Distribution of organisms in classes

Upon further examination of the distribution, it was determined that many classes consisted solely of one organism. In such cases, a more detailed approach was taken by examining the sub-phylum to which the individual organism belongs. Figure 5.1 provides an additional visual representation of this decision-making process. Nevertheless, when confronted with difficulties in constructing models for sub-phyla due to conflicts with classes from the same sub-phyla that are prevalent in organisms, an alternative approach was utilised.

Models were created in these situations by combining classes according to their parent phylum, thereby preventing unnecessary duplications.

The thorough approach led to the development of 19 separate models that together cover all organisms (except Cryptomycota and Microsporidia, as mentioned before). The following list presents these models.

- **Phyla models**
  - Blastocladiomycota
  - Zoopagomycota
- **Subphyla models**
  - Pucciniomycotina
  - Ustilaginomycotina
  - Taphrinomycotina
- **Class models**
  - Agaricomycetes
  - Chytridiomycetes
  - Dothideomycetes
  - Eurotiomycetes
  - Leotiomycetes
  - Mucoromycetes
  - Pezizomycetes
  - Sordariomycetes
  - Tremellomycetes
  - Saccharomycetes
- **Merged class models**
  - Monoblepharidomycetes + Neocallimastigomycetes
  - Dacrymycetes + Geminibasidiomycetes + Wallemiomycetes
  - Glomeromycetes + Mortierellomycetes + Umbelopsidomycetes
  - Coniocybomycetes + Orbiliomycetes + Xylonomycetes + Lecanoromycetes

After creating datasets specifically designed for each model, a lengthy procedure of training and evaluation followed. This phase involved a series of carefully planned experiments to determine the effectiveness of augmentation and transfer learning in attaining the main goal: determining the fewest number of models needed to cover the entire fungal domain. The specific information and results of these experimental efforts are explained in the next chapter, Chapter 7.

Table 5.3: Evatuation of the donor models on individual phyla

TPR							
Model | evaluace na	ASCOMYCOTA	BASIDIOMYCOTA	BLASTOCLADIOMYCOTA	CHYTRIDIOMYCOTA	MICROSPORIDIA	MUCOROMYCOTA	ZOOPAGOMYCOTA
ASCOMYCOTA	0.898	0.893	0.89	0.88	0.962	0.916	0.817
BASIDIOMYCOTA	0.885	0.918	0.908	0.88	0.962	0.884	0.804
BLASTOCLADIOMYCOTA	0.807	0.805	0.908	0.73	0.849	0.714	0.79
CHYTRIDIOMYCOTA	0.855	0.856	0.889	0.855	0.943	0.91	0.832
MUCOROMYCOTA	0.761	0.707	0.753	0.83	0.925	0.919	0.815
ZOOPAGOMYCOTA	0.847	0.824	0.835	0.85	0.962	0.93	0.857
PPV							
Model | evaluace na	ASCOMYCOTA	BASIDIOMYCOTA	BLASTOCLADIOMYCOTA	CHYTRIDIOMYCOTA	MICROSPORIDIA	MUCOROMYCOTA	ZOOPAGOMYCOTA
ASCOMYCOTA	0.94	0.94	0.96	0.84	0.89	0.91	0.90
BASIDIOMYCOTA	0.94	0.94	0.96	0.88	0.91	0.91	0.89
BLASTOCLADIOMYCOTA	0.84	0.83	0.86	0.80	0.79	0.80	0.80
CHYTRIDIOMYCOTA	0.91	0.91	0.91	0.90	0.88	0.92	0.87
MUCOROMYCOTA	0.95	0.95	0.97	0.90	0.89	0.95	0.94
ZOOPAGOMYCOTA	0.86	0.86	0.88	0.86	0.80	0.88	0.84
Adjusted PPV							
Model | evaluace na	ASCOMYCOTA	BASIDIOMYCOTA	BLASTOCLADIOMYCOTA	CHYTRIDIOMYCOTA	MICROSPORIDIA	MUCOROMYCOTA	ZOOPAGOMYCOTA
ASCOMYCOTA	0.79	0.81	0.87	0.56	0.68	0.72	0.70
BASIDIOMYCOTA	0.79	0.80	0.84	0.64	0.72	0.72	0.67
BLASTOCLADIOMYCOTA	0.57	0.55	0.60	0.50	0.48	0.50	0.51
CHYTRIDIOMYCOTA	0.71	0.71	0.72	0.70	0.64	0.74	0.62
MUCOROMYCOTA	0.83	0.82	0.87	0.69	0.67	0.82	0.81
ZOOPAGOMYCOTA	0.61	0.61	0.64	0.61	0.50	0.65	0.56
F							
Model | evaluace na	ASCOMYCOTA	BASIDIOMYCOTA	BLASTOCLADIOMYCOTA	CHYTRIDIOMYCOTA	MICROSPORIDIA	MUCOROMYCOTA	ZOOPAGOMYCOTA
ASCOMYCOTA	0.84	0.85	0.88	0.68	0.80	0.80	0.75
BASIDIOMYCOTA	0.83	0.85	0.87	0.74	0.82	0.80	0.73
BLASTOCLADIOMYCOTA	0.67	0.65	0.73	0.59	0.62	0.59	0.62
CHYTRIDIOMYCOTA	0.77	0.78	0.79	0.77	0.76	0.81	0.71
MUCOROMYCOTA	0.79	0.76	0.81	0.75	0.78	0.87	0.81
ZOOPAGOMYCOTA	0.71	0.70	0.73	0.71	0.65	0.77	0.68

Table 5.4: Evatuation of the acceptor models on individual phyla

TPR							
Model | evaluace na	ASCOMYCOTA	BASIDIOMYCOTA	BLASTOCLADIOMYCOTA	CHYTRIDIOMYCOTA	MICROSPORIDIA	MUCOROMYCOTA	ZOOPAGOMYCOTA
ASCOMYCOTA	0.895	0.796	0.879	0.86	0.75	0.947	0.765
BASIDIOMYCOTA	0.894	0.892	0.924	0.87	0.712	0.957	0.851
BLASTOCLADIOMYCOTA	0.493	0.688	0.541	0.778	0.538	0.869	0.304
CHYTRIDIOMYCOTA	0.753	0.828	0.929	0.79	0.308	0.927	0.836
MUCOROMYCOTA	0.653	0.649	0.8	0.805	0.635	0.941	0.506
ZOOPAGOMYCOTA	0.677	0.762	0.811	0.798	0.788	0.938	0.688
PPV							
Model | evaluace na	ASCOMYCOTA	BASIDIOMYCOTA	BLASTOCLADIOMYCOTA	CHYTRIDIOMYCOTA	MICROSPORIDIA	MUCOROMYCOTA	ZOOPAGOMYCOTA
ASCOMYCOTA	0.94	0.92	0.95	0.87	0.80	0.83	0.93
BASIDIOMYCOTA	0.92	0.90	0.93	0.84	0.84	0.82	0.91
BLASTOCLADIOMYCOTA	0.77	0.79	0.86	0.71	0.76	0.71	0.82
CHYTRIDIOMYCOTA	0.84	0.84	0.83	0.83	0.73	0.85	0.82
MUCOROMYCOTA	0.94	0.94	0.97	0.89	0.79	0.88	0.94
ZOOPAGOMYCOTA	0.82	0.83	0.85	0.82	0.80	0.83	0.81
Adjusted PPV							
Model | evaluace na	ASCOMYCOTA	BASIDIOMYCOTA	BLASTOCLADIOMYCOTA	CHYTRIDIOMYCOTA	MICROSPORIDIA	MUCOROMYCOTA	ZOOPAGOMYCOTA
ASCOMYCOTA	0.80	0.75	0.84	0.62	0.49	0.54	0.77
BASIDIOMYCOTA	0.73	0.70	0.77	0.56	0.57	0.53	0.72
BLASTOCLADIOMYCOTA	0.46	0.49	0.61	0.38	0.44	0.38	0.52
CHYTRIDIOMYCOTA	0.57	0.56	0.56	0.56	0.40	0.59	0.54
MUCOROMYCOTA	0.80	0.78	0.88	0.67	0.48	0.65	0.79
ZOOPAGOMYCOTA	0.54	0.55	0.58	0.53	0.51	0.55	0.51
F							
Model | evaluace na	ASCOMYCOTA	BASIDIOMYCOTA	BLASTOCLADIOMYCOTA	CHYTRIDIOMYCOTA	MICROSPORIDIA	MUCOROMYCOTA	ZOOPAGOMYCOTA
ASCOMYCOTA	0.84	0.77	0.86	0.72	0.60	0.69	0.77
BASIDIOMYCOTA	0.80	0.78	0.84	0.68	0.63	0.69	0.78
BLASTOCLADIOMYCOTA	0.48	0.57	0.58	0.51	0.48	0.53	0.38
CHYTRIDIOMYCOTA	0.65	0.67	0.70	0.65	0.35	0.72	0.66
MUCOROMYCOTA	0.72	0.71	0.84	0.73	0.55	0.77	0.62
ZOOPAGOMYCOTA	0.60	0.64	0.68	0.64	0.62	0.69	0.59

Part I

Model training

Chapter 6

Methods

6.1 Augmentation

6.1.1 Implementation

The most crucial step of implementation was integrating the augmentation into an already prepared pipeline that was created by Bc. Martin Indra [20]. For an efficient training process, it was critical to ensure precise handling of data generated from preprocessing and generate the dataset file in the correct format. All of these needs were taken into account, and the new pipeline is represented in 6.1 Part **A**.

This pipeline thoroughly details the whole process that leads to the development of the final graph. It includes a full explanation of the inputs and outputs for each block, with a notable focus on the augmentation block. However, it is essential to consider that the specific data augmentations used may establish a bias in the structure of motif grammars. Thus, the DNN is fine-tuned on the original, unaltered data in the second step to enhance these features and lead the function towards the observed biology, removing any bias produced by the data augmentations [27].

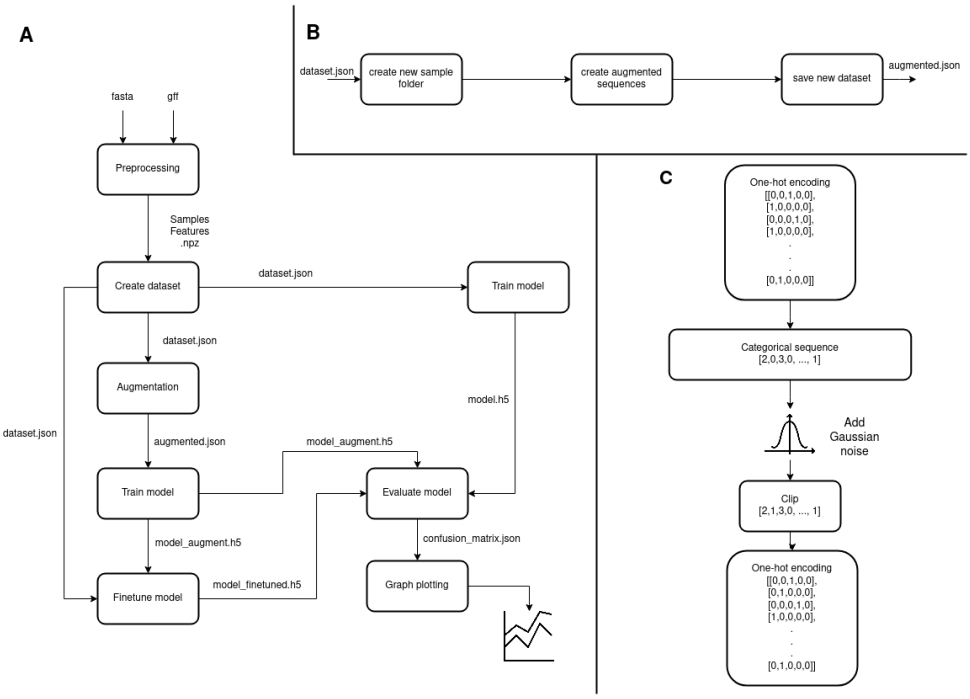


Figure 6.1: Schematic of the process of augmentation. **A** The graph depicts the pipeline of the entire augmentation process, from raw data in Fasta and GFF files to building datasets, training and testing models, and finally displaying the performance of the models by comparing F1 scores. **B** Zoom in on the procedure under the "Augmentation" block. **C** A comprehensive description of how the augmented data was obtained and how Gaussian noise is added to the sequences.

Part **B** of Figure 6.1 depicts the internal processes of this block. It is worth noting that it includes more than just the addition of random noise. Given the data access methods used by Bc. Martin Indra's programmes, there was a need to recreate the way the original samples are stored and apply it to the augmented ones as well. Regardless, the main result of this phase is the generation of a new JSON file that describes the new dataset.

Due to the preprocessing sequences being encoded as one-hot, adding noise to them directly proved unfeasible. It became necessary to convert these sequences into a format compatible with the NumPy library's function for creating random Gaussian noise. As a result, it was decided to first turn the sequence into a categorical variable, allowing for the noise addition. Before returning to one-hot encoding, a clipping step was added to make sure that the output values were within the range corresponding to the number of bases, namely 5 for A, C, G, T, and N for random. The entire process is illustrated in Figure 6.1 part **C**, accompanied by an illustrative example.

6.2 Transfer learning

6.2.1 Implementation

The provided methodology, as shown in Figure 6.2, divides the transfer learning process into two separate phases, each of which contributes to the improvement of model adaptability for certain tasks. Two datasets are derived after data preprocessing to assure data quality and consistency: one for pre-training and the other for fine-tuning. During the pre-training phase, a model is trained on the initial dataset, providing a comprehensive understanding of general characteristics and representations. The model goes through a transfer learning block with the second dataset at the same time, adapting its expertise to the job at hand. Using the second dataset, an independently trained model is built in parallel. The comparative examination of the transfer-learned model and the independently trained model provides insight into their effectiveness. A graph is created to provide a more visual representation, illustrating the performance metrics derived from the evaluation. This graphical depiction serves as a visual aid, allowing a clearer understanding of the framework's ability to improve model performance when compared to traditional training approaches.

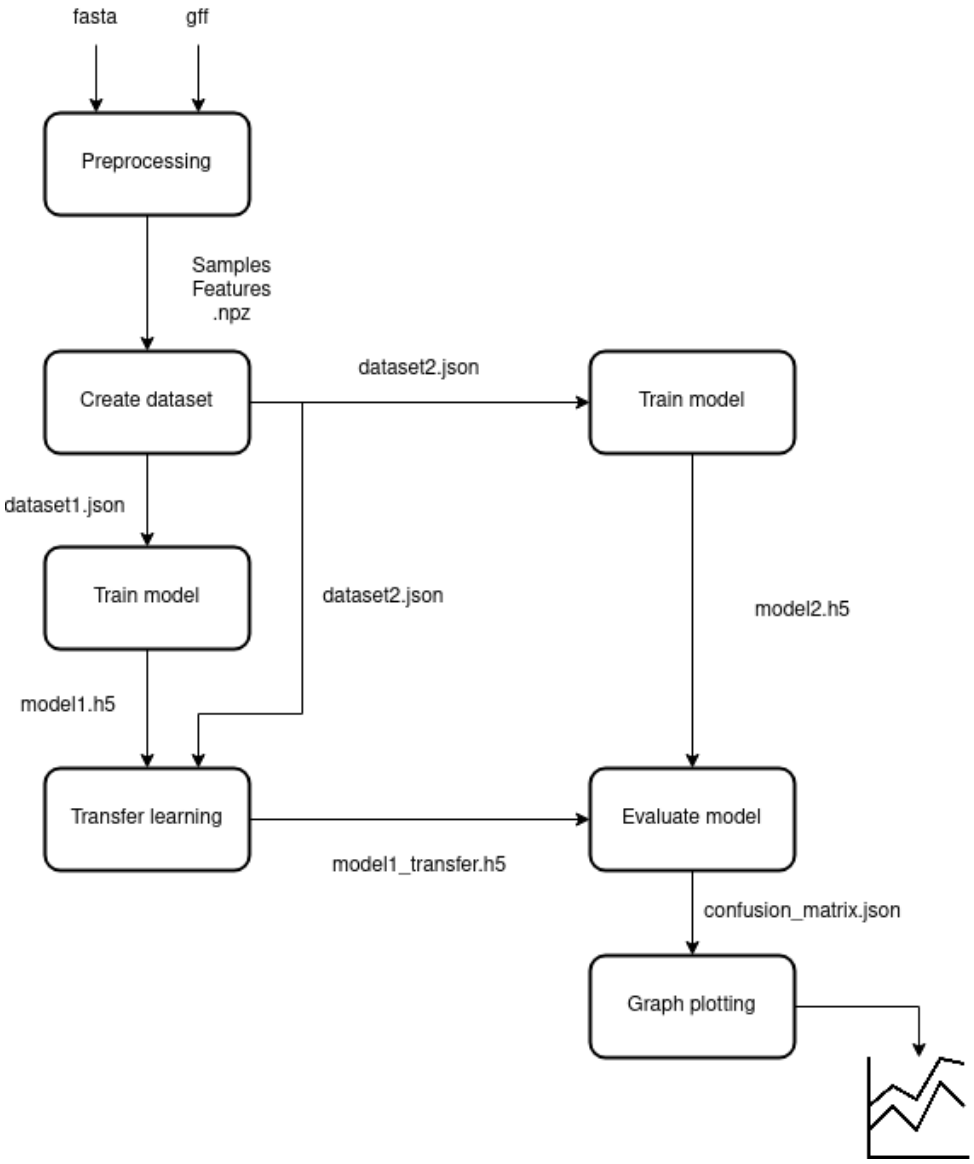


Figure 6.2: Schematic of the process of transfer learning.

Chapter 7

Experiments

Each and every research endeavour needs the provision of evidence, and it is precisely in this context that experiments present themselves. This chapter provides a thorough description of these experiments, including augmentation, transfer learning, and a comparative examination of normal, augmented, and transfer models.

Parameter adjustment becomes crucial while implementing the augmentation approach. An elaborate investigation, specifically using the Blastocladiomycota phylum, explores the ideal configurations for the mean, standard deviation, and proportion of augmented data. The augmentation experiment's performance is analysed and presented in three visual graphs, providing valuable insights.

The transfer learning experiment involves a thorough examination of the most effective parameter configurations, including the learning rate, the number of frozen layers, and batch sizes. The experiment demonstrates that frozen layers have a negligible effect, leading to a strategic change in focus towards investigating the interactions between learning rates and batch sizes.

The last experiment's objective is to conduct a comparative examination of normal, augmented, and transfer models using datasets with increasing organism counts. The emphasis lies in identifying the circumstances in which each type of model demonstrates exceptional performance, particularly in scenarios involving a limited number of species. Due to the impact of evolutionary proximity between the dataset and the base model during transfer learning four distinct settings were created.

7.1 Augmentation

7.1.1 Parameter tuning

The requirement for parameter tuning arose as a critical requirement as a result of the implementation of the augmentation strategy. An experiment was created to determine the best setting of the parameters. Due to time complexity, just one phylum was picked.

The rationale behind choosing phylum *Blastocladiomycota* is due to its relatively small dataset, which consists of only four organisms. The essential factors under consideration were the standard deviation, mean, and the proportion of augmented data in the dataset. For example, the augmentation process involved experimenting with different percentages, such as 25%, signifying the proportion of augmented data added to the original dataset, which were tested. The precise values for these parameters were crucial in establishing the augmentation strategy and, as a result, the overall performance of the model.

The exact values of these parameters tested were:

- Mean = [0.0, 0.25, 0.5, 1.0]
- Standard Deviation = [0.1, 0.2, 0.3, 0.5, 1.0]
- Percentage = [25.0, 50.0, 75.0, 100.0]

Three specific graphs were created to completely examine the performance of each parameter arrangement. The first graph provides a visual depiction of all parameter combinations, providing a full overview. The second graph focuses on highlighting the best-performing models from each percentage group, providing insights into the top performers in various augmentation scenarios. Finally, a boxplot was used to depict the distribution of the data, offering a brief summary of the experimental results. The graphical representation of these results is accessible in Figures for donor model 7.1, 7.2a, and 7.2b and for acceptor model 7.3, 7.4a and 7.4b.

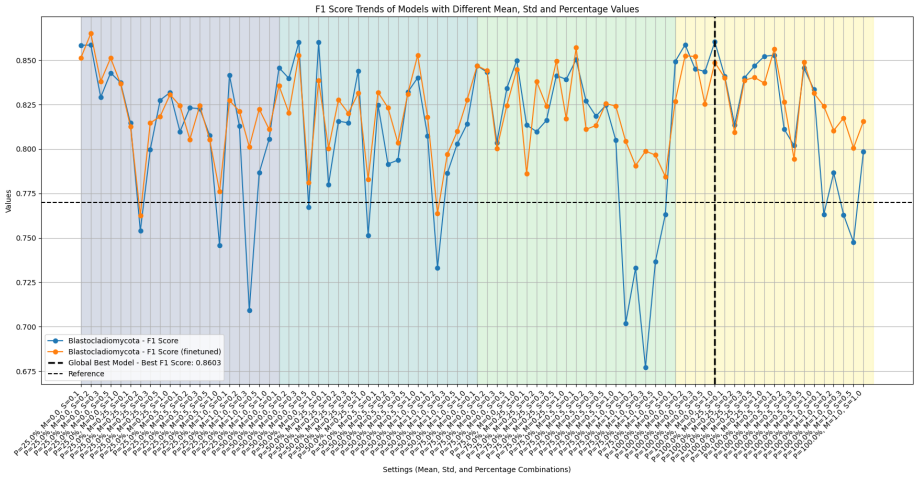
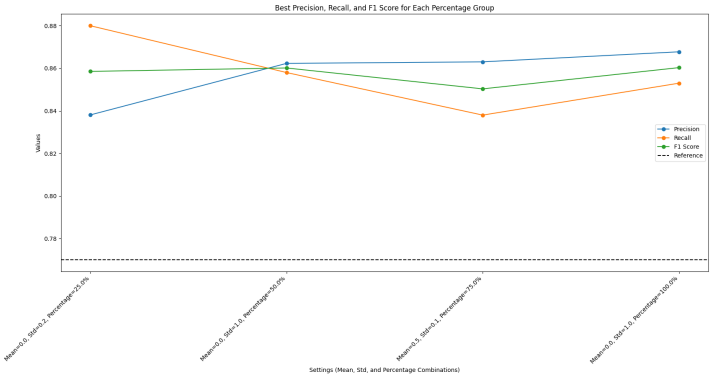
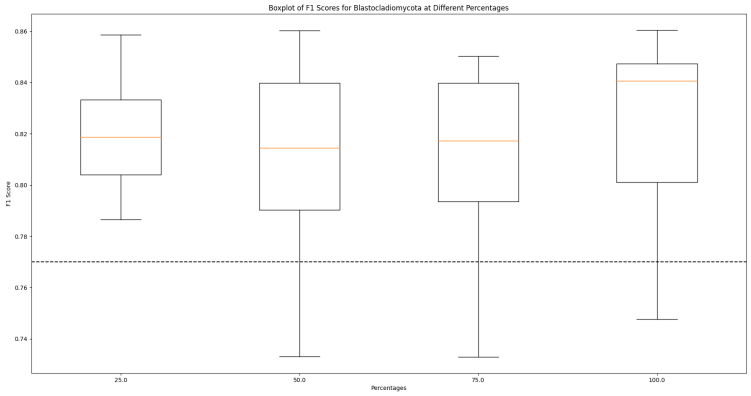


Figure 7.1: F1 Score Trends of Donor Models with Different Mean, Std and Percentage Values



(a) : Best Precision, Recall, and F1 Score for Each Percentage Group



(b) : Boxplot of F1 Scores at Different Percentages

Figure 7.2: Augmented donor model review

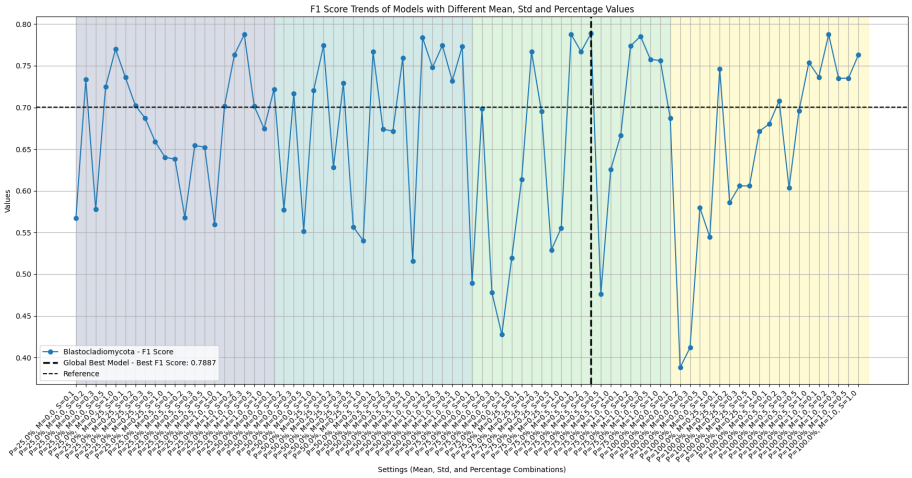
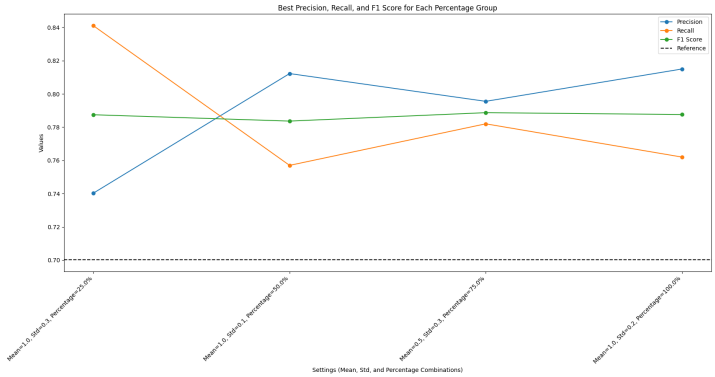
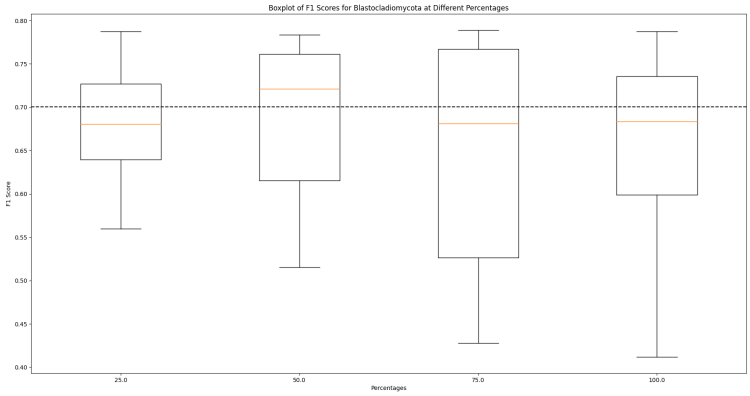


Figure 7.3: F1 Score Trends of Acceptor Models with Different Mean, Std and Percentage Values



(a) : Best Precision, Recall, and F1 Score for Each Percentage Group



(b) : Boxplot of F1 Scores at Different Percentages

Figure 7.4: Augmented acceptor model review

7.1.2 Results

The efficacy of this strategy has been demonstrated in my research on augmenting DNA sequences, with superior outcomes for the model when trained using augmented data. Despite the obvious improvements, determining the ideal set of parameters remains a difficult task. Experimentation found that the F1 scores associated with different parameter combinations are surprisingly similar as can be seen in 7.1 or 7.3. The results analysis, as depicted in 7.2a and 7.4a, reveals an interesting observation regarding the comparison of the best-performing models from each percentage group. Contrary to expectations, there is no obvious dominance or significant difference in performance outcomes across the examined groups. This finding emphasises the difficulty in determining the most effective parameter setting.

While a single best parameter choice remains elusive in our investigation, a careful examination of box plots 7.2b and 7.4b reveals intriguing trends in the data distribution. Notably, the 25% group in the donor model consistently has the smallest interquartile range, shortest whisker lengths, and outliers that closely hug the box. This pattern indicates a degree of coherence and stability within this subset, showing a possible association between the chosen parameters and the observed model performance features. Based on the aforementioned findings, the setting of 'mean': 0.0, 'std': 0.2', and 'percentage': 25.0 was selected for the training of augmented donor models. On the other hand, upon examining the outcomes of the acceptor model, it was not immediately clear what the optimal parameter configuration was. Although all percentage groups in 7.4a have at least one model that outperforms the normal one, 7.4b reveals an intriguing observation. Each percentage group has a significantly wide interquartile range, surpassing even the F1 score of the reference normal model. As a result, selecting the best-performing model became challenging. Ultimately, a setting of 'mean': 1.0, 'std': 0.1', and 'percentage': 50.0 was selected. This particular percentage group was the only one where the mean exceeded the reference value.

7.2 Transfer learning

7.2.1 Tuning hyperparameters

A comprehensive experiment is carried out to investigate the optimal setting for hyperparameters of transfer learning. The mentioned hyperparameters are:

- Learning rate = [0.001, 0.0001, 2e-05, 5e-05]
- Number of frozen layers = [0, ..., 10]
- Batch size = [1, 2, 4, 8, 12, 16]

Because of time-based complexity constraints, batch size tweaking was skipped over at first in this investigation. Given the extensive range of hyperparameters investigated, including the examination of frozen layers (ranging from 0 to 10) and learning rates (with values of '0.001', '0.0001', '2e-05', '5e-05'), a balance between the depth of exploration and the computational resources available was critical. To maximise efficiency, two batch sizes — 16 and 4 — were chosen to capture a reasonable spectrum of batch size effects while avoiding an exhaustive search. The results are shown in Figures 7.5 and 7.6.

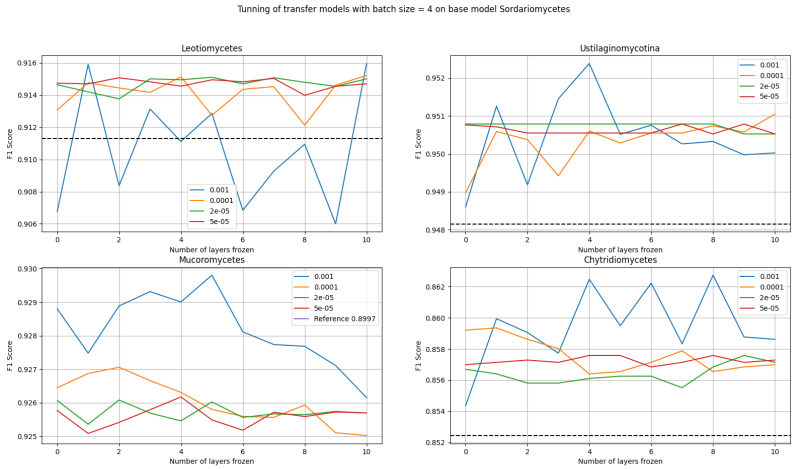


Figure 7.5: Results of tuning hyperparameters with batch size = 4 on base donor model Sordariomycetes.

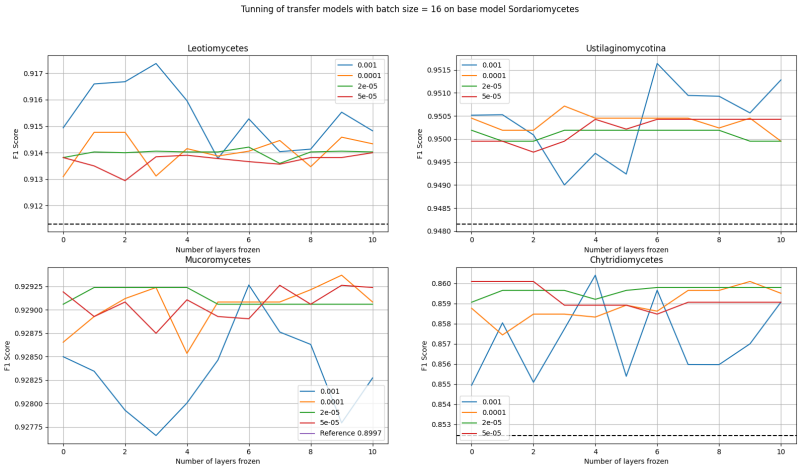


Figure 7.6: Results of tuning hyperparameters with batch size = 16 on base donor model Sordariomycetes.

After the findings were examined, it became clear that the variation in the number of frozen layers had no significant impact on model performance. As a result, a strategic choice was made to change the scope of the investigation. The following experiment was then carried out with a focus on the relationship between learning rates and batch sizes while ignoring layer freezing. Figure 7.7 depicts the outcome.

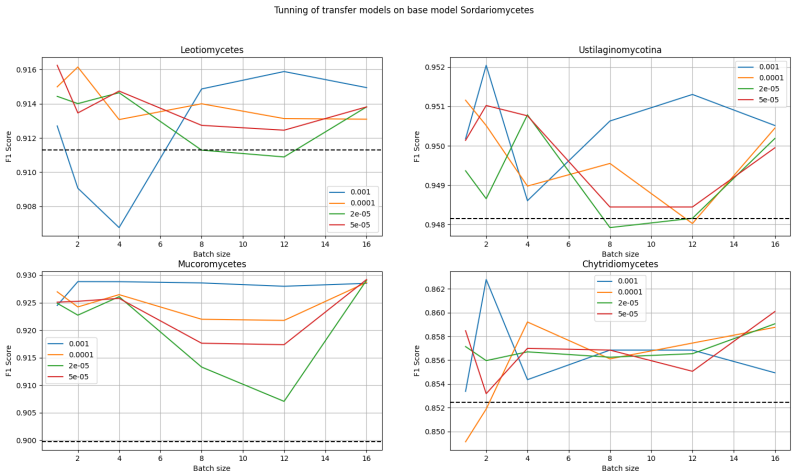


Figure 7.7: Results of tuning learning rate and batch size on the base donor model Sordariomycetes.

A parallel study was conducted to develop acceptor models based on the insights gained from training donor models, in order to achieve a thorough

understanding and optimisation. Based on the information obtained from the donor model studies, the acceptor models were primarily concerned with the important factors of batch size and learning rate. The results for acceptor models can be seen below in Figure 7.8.

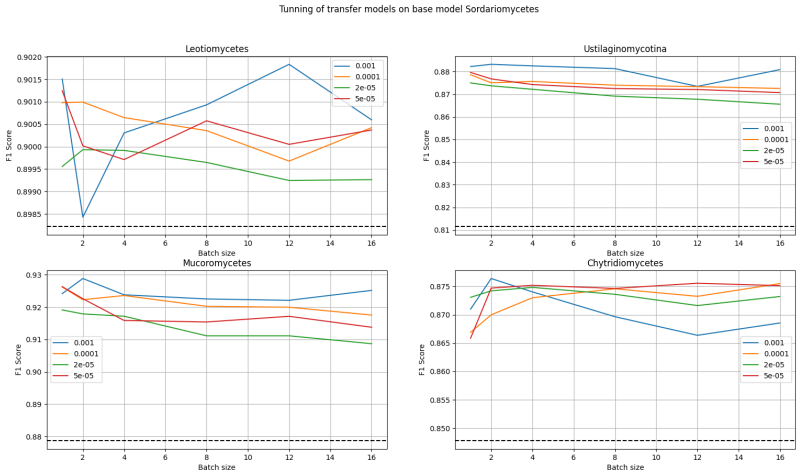


Figure 7.8: Results of tuning learning rate and batch size on base acceptor model Sordariomycetes.

7.2.2 Results

Following the discovery that the number of frozen layers had no effect on the F1 score, attention was focused on a controlled experiment involving the fine-tuning of the learning rate and batch size parameters. During the research process, it became abundantly evident that the results were not optimal, which eventually resulted in an in-depth analysis to determine the most effective parameter configuration. As such, the optimal parameters 'learning_rate': 5e-05, 'batch_size': 16 were successfully identified for all four classes in the case of the donor model, since they yielded the highest average F1 score. Interestingly, the results of the acceptor model were easier to interpret due to the absence of significant fluctuations. Throughout all the groups, a setting of 'learning_rate': 0.001, 'batch_size': 2 was chosen. Nevertheless, the observed variations in parameter settings have prompted consideration of the possible benefits of customising parameters for each class, considering the lack of significant differences.

7.3 Model comparison

One of the primary goals was to conduct a comparative analysis of different models, namely normal, augmented, and transfer models, with the aim of determining their respective levels of usability. The experiment aimed to evaluate the impact of augmentation and transfer learning on performance across different datasets, taking into account the substantial variation in the number of organisms in each dataset. This variety spanned from datasets with fewer than a dozen organisms to those with larger amounts of data. In order to thoroughly assess the performance of the model within this range, it was essential to conduct a detailed assessment.

The main objective of the experiment was to identify situations in which the traditional model was adequate compared to those where augmented or transfer learning models were more beneficial. This in-depth evaluation not only enhanced the understanding of the efficacy of each model but also yielded vital insights about the versatility of augmentation and transfer learning methods, as influenced by the unique characteristics of the dataset.

Four separate settings were constructed as a consequence of this discovery.

1. Dataset of class Eurotiomycetes (phyla: Ascomycota, number of organisms: 135) transfer on the base model of class Sordariomycetes (phyla: Ascomycota, number of organisms: 120)
2. Dataset of class Ustilaginomycotina (phyla: Basidiomycota, number of organisms: 21) transfer on the base model of class Agaricomycetes (phyla: Basidiomycota, number of organisms: 223)
3. Dataset of class Eurotiomycetes (phyla: Ascomycota, number of organisms: 135) transfer on the base model of class Agaricomycetes (phyla: Basidiomycota, number of organisms: 223)
4. Dataset of class Mucoromycetes (phyla: Mucomycota, number of organisms: 31) transfer on the base model of class Leotiomycetes (phyla: Ascomycota, number of organisms: 32)

The findings of this analysis for donor and acceptor models are visually presented in Figure 7.9 and 7.10 below. The graphs below illustrate the performance of the normal model in blue, the augmentation model in orange, and the transfer models in green. Additionally, there is a line of reference.

This reference provides an evaluation of the dataset carried by the base model. Due to limited time, the third setting was disregarded while researching acceptor models.

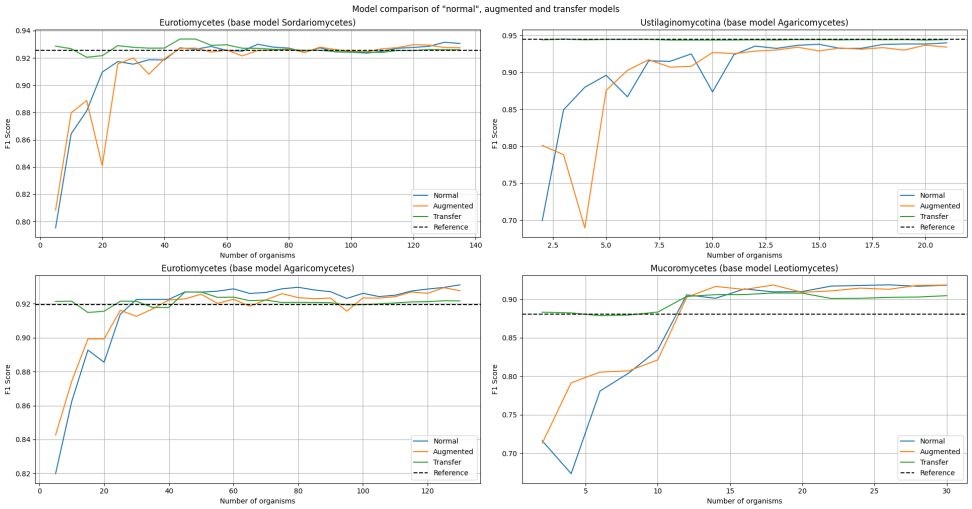


Figure 7.9: Comparison of normal, augmented and transfer donor models. The setting of the augmented model: mean 0.0, std 0.2, percentage 25.0 and of transfer model: lr=5e-05, batch size = 16 frozen layers = 0

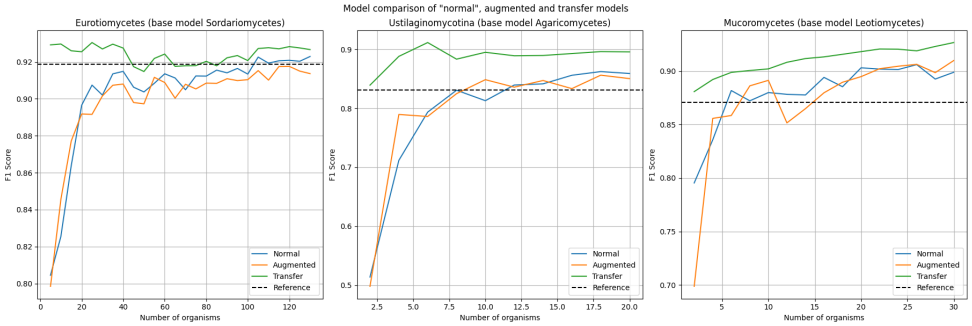


Figure 7.10: Comparison of normal, augmented and transfer acceptor models. The setting of the augmented model: mean 1.0, std 0.1, percentage 50.0 and of transfer model: lr=0.001, batch size = 2 frozen layers = 0

7.3.1 Results

After carefully analysing the results, it becomes evident that there is a strong correlation that highlights the crucial significance of the number of organisms in the dataset in determining the effectiveness of the models in identifying splice sites. Unsurprisingly, the normal model had difficulties attaining similar results as the base model, regardless of modifications in the base model's

dimensions or dataset qualities. This disparity implies a possible constraint in the normal model's capacity to adjust to various dataset settings.

In contrast, the augmentation model demonstrated favourable results, especially when handling vast datasets. The performance of the augmentation model exceeded that of the base model, particularly when the dataset included a significant number of creatures. Nevertheless, a significant limitation became apparent while working with smaller datasets. This was likely a result of the configuration of the augmentation process, as well as the constraint on the amount of data that may be augmented without compromising the model's performance.

The transfer model, which showed the best overall results, displayed an impressive ability to withstand concerns about the size of the dataset. It immediately outperformed the base model. Nevertheless, a subtle difficulty arose about the model's evaluation proximity to the dataset. The complexity of this matter became evident when analysing the case of the Eurotiomycetes class, which was trained using the Sordariomycetes class model. Both classes belong to the same phylum. The evaluation proximity issue proposes that the performance of the transfer model is highly dependent on the relationship between the model and the used dataset.

These findings clearly indicate that transfer learning is the most effective approach for encompassing the entire fungal kingdom with the fewest number of models.

Part II

Model explanation

Chapter 8

Logo sequences

Sequence logos, when incorporated into CNN explanations, function as interpretable visualisations that shed light on the decision-making mechanisms of the network. The deliberate choice of using sequence logos for comprehending splice site models is based on the fact that these visualisations not only offer clarity, but also effectively capture and depict complex patterns within the data. Their efficacy derives from their capacity to condense complex information into visually comprehensible formats, making them more readable even for individuals who are unfamiliar with the intricacies of CNNs.

8.1 Logo creation

The sequence logo creation procedure greatly benefited from the innovative code written by a doctorate candidate Anh Vu Le. The code developed by Le presented a highly sophisticated framework; yet, the adaptation of this framework to the dataset that was created by the code from Indra [20] remained unknown. In regard to the intricate nature of the task and the limited amount of time available, a practical choice was made to utilise the preprocessing, dataset creation and logo visualisation offered by Le.

8.1.1 Preproccesing of data and dataset creation

FASTA files were generated for each fungus by extracting intron and exon sequences from Assembly and GFF files. In addition to the FASTAs, CSV files were generated containing the locations of introns and exons. The fungus directories are composed of splice site window sequences, which include both donor and acceptor FASTAs. These sequences are 402 nucleotides in length and contain AG and GT di-nucleotides in the middle. There are separate versions for the false donor and true acceptor windows.

The ipynb notebook simplifies the process of generating training and validation datasets by enabling users to define specific parameters, such as the phyla and species to be included, the number of samples per species, and the ratio between positive and negative samples. This procedure utilises the previously stated pre-generated splice site windows.

8.1.2 The process of sequence logo creation

The creation of logos requires a methodical process that can be divided into four essential steps. Initially, it was crucial to choose a single organism for each model, taking into account that the evaluation focused solely on individual organisms rather than entire classes, sub-phyla, or phyla. The selected organisms and their taxonomic classification include:

- Tralac1: Agaricomycetes
- Blabri1: Blastocladiomycota
- Synfus1: Zoopagomycota
- Obemuc1: Chytridiomycetes
- Disac1: Dothideomycetes
- Claim1: Eurotiomycetes
- Bissp1: Leotiomycetes
- Pilano1: Mucoromycetes
- Sarco1: Pezizomycetes
- Colgr1: Sordariomycetes

- Kocim1: Tremellomycetes
- Picpa1: Saccharomycetes
- Spoli1: Pucciniomycotina
- Jamsp1: Ustilaginomycotina
- Saico1: Taphrinomycotina
- Anasp1: Monoblepharidomycetes + Neocallimastigomycetes
- Calco1: Dacrymycetes + Geminibasidiomycetes + Wallemiomycetes
- Gloin1: Glomeromycetes + Mortierellomycetes + Umbelopsidomycetes
- Symko1: Coniocybomycetes + Orbiliomycetes + Xylonomycetes + Lecanoromycetes

The next two steps required utilising a donor model to identify acceptor dimers and vice versa to ensure compatibility with introns. Attributions were acquired through these procedures. The last stage was displaying the performance of these models on introns, using the previously obtained attributions, by sequence logos. The Deeplift library's viz_sequence function was utilised to generate visualisations of the randomly selected introns. Furthermore, after visualising introns as logos, high-scoring regions (motifs) were also extracted and the position weight matrix (PWM) was computed for later comparison of logos.

8.2 Results

A grand total of 38 logos were systematically created, with each organism providing two logos—one generated from the donor model and another from the acceptor model. These logos collectively offer a thorough understanding of the models' performance in many biological scenarios. The entire collection of logos can be found conveniently in the appendices (refer to Appendices). In order to do a detailed comparison of the logos produced by the acceptor and donor models, the R package *DiffLogo* [35] was utilised. This comparative analysis provides insight into the differences between the two types of logos, presents useful information on the models' different responses and pinpointing areas of divergence. The results of these comparisons are likewise included in the appendices. Upon careful review of the logos, it becomes apparent that certain models have higher performance in comparison to others. Certain models demonstrate exceptional proficiency in identifying dimers, while others struggle and instead identify patterns, particularly emphasising the presence of the base T.

Part III

Discussion and Conclusion

Chapter 9

Discussion

9.1 Experiments

The experimentation involved two primary approaches: augmentation and transfer learning, each yielding unique results and findings. The augmentation strategy involved the integration of augmentation into an already established pipeline. The technique of augmentation, although improving the scores of certain classes, was not universally applicable, requiring further adjustment to the original data to achieve better usability. Parameter tuning experiments were performed to optimise the augmentation technique, with a specific focus on the Blastocladiomycota phylum because of its smaller dataset. The presence of identical F1 scores across several parameter combinations has revealed the difficulty in identifying an ideal set. The analysis of box plots uncovered interesting patterns, guiding the choice of parameters for donor and acceptor models. Based on all the findings, these are the settings that were chosen:

- Donor: 'mean': 0.0, 'std': 0.2', and 'percentage': 25.0
- Acceptor: 'mean': 1.0, 'std': 0.1', and 'percentage': 50.0

The transfer learning strategy adopted a phased approach, which involved splitting the process into two distinct phases: pre-training and fine-tuning. A comprehensive experiment was carried out to optimise hyperparameters,

with particular emphasis on frozen layers, learning rates, and batch sizes. Remarkably, the performance was not much affected by the number of frozen layers. Further tests focused on determining the most effective learning rates and batch sizes. This resulted in the determination of the ideal parameters for donor models as a learning rate of 5e-05 and a batch size of 16 and for acceptor models as a learning rate of 0.001 and a batch size of 2. Nevertheless, the intricacy of parameter configurations necessitated the contemplation of customisation tailored to individual classes as it became apparent that the differences between classes are bigger than was initially thought.

The objective of the model comparison phase was to evaluate the usability of several models, specifically the normal, augmented, and transfer models, using a range of different datasets. Four unique environments were established, which entailed transitioning between different classes with different sizes of datasets. The visualisations in Figures 7.9 and 7.10 revealed complex patterns. The standard model experienced difficulties in adjusting to various dataset configurations, suggesting potential constraints. Augmentation demonstrated efficacy, especially when applied to bigger datasets, but encountered difficulties when applied to smaller datasets. The transfer models regularly achieved superior performance compared to the base model, demonstrating their effectiveness in managing datasets of different sizes. Nevertheless, the evaluation proximity problem highlighted the significance of the correlation between the model and the dataset utilised.

In general, transfer learning has demonstrated higher capabilities in achieving high scores. This implies that it could have a significant influence on the final determination of the number of models required to cover the entirety of the fungal kingdom. The demonstrated efficacy of transfer learning in improving model performance is consistent with the results of the background research, which also highlighted the substantial enhancement of model outcomes through transfer learning.

9.2 Model recommendation

A thorough evaluation was carried out to evaluate the models and choose the most efficient options for both donor and acceptor classifications. The selection process gave priority to models with the highest sum of F1 scores, which is an indication of their balanced performance in terms of precision and recall.

The Eurotiomycetes model showed exceptional performance for the donor, achieving a sum of F1 score of 16.3356 and an average score of 0.8598. On the other hand, the Agaricomycetes model proved to be the optimal choice for the acceptor, with a sum of 15.5203 and an average of 0.8169. The data, clearly displayed in the accompanying table 9.1, highlights the outstanding skill of the chosen models.

Model	**F1 Score (Eurotiomycetes)**	**F1 Score (Agaricomycetes)**
Agaricomycetes	0.9277	0.9164
Blastocladiomycota	0.9279	0.8412
Zoopagomycota	0.8976	0.7381
Chytridiomycetes	0.8618	0.9049
Dothideomycetes	0.9214	0.9048
Eurotiomycetes	0.9303	0.9104
Leotiomycetes	0.9132	0.8780
Mucoromycetes	0.8993	0.8974
Pezizomycetes	0.8660	0.8741
Sordariomycetes	0.9115	0.8844
Tremellomycetes	0.9364	0.8103
Pucciniomycotina	0.9465	0.8865
Ustilaginomycotina	0.9517	0.8308
Taphrinomycotina	0.7770	0.6768
merge_model1	0.8415	0.8348
merge_model2	0.9427	0.9293
merge_model3	0.9251	0.8830
merge_model4	0.9578	0.9193

Table 9.1: F1 Scores for models evaluated by Eurotiomycetes for donor and by Agaricomycetes for acceptor

The current scores largely surpass the scores of phylum models in Tables 5.4 and 5.3. Nevertheless, certain classes exhibit particularly low scores, even descending below 0.8, which can be considered undesirable. This implies that depending solely on two models has been deemed inadequate. In light of this, a comprehensive analysis of the models was undertaken to determine the optimal model for assessing each class. The F1 scores and optimal models for evaluation are shown in Table 9.1 below.

	donor		acceptor	
Models	**F1 score**	**best model**	**F1 score**	**best model**
Dothideomycetes	0,9215	Dothideomycetes	0,9198	Eurotiomycetes
Eurotiomycetes	0,9303	Eurotiomycetes	0,9209	Eurotiomycetes
Leotiomycetes	0,9136	Dothideomycetes	0,9007	Eurotiomycetes
Pezizomycetes	0,866	Eurotiomycetes	0,8943	Sordariomycetes
Sordariomycetes	0,9122	Sordariomycetes	0,9107	Sordariomycetes
Taphrinomycotina	0,8031	Taphrinomycotina	0,839	Dothideomycetes
merge_model4	0,9578	Eurotiomycetes	0,9331	Eurotiomycetes
Agaricomycetes	0,9415	Agaricomycetes	0,9164	Agaricomycetes
Tremellomycetes	0,9482	Dothideomycetes	0,9318	Pucciniomycotina
Pucciniomycotina	0,9466	Agaricomycetes	0,8865	Agaricomycetes
Ustilaginomycotina	0,9517	Eurotiomycetes	0,8618	Ustilaginomycotina
merge_model2	0,9526	Agaricomycetes	0,9293	Agaricomycetes
Blastocladiomycota	0,9295	Agaricomycetes	0,8616	Pucciniomycotina
Chytridiomycetes	0,8696	Agaricomycetes	0,9049	Agaricomycetes
merge_model1	0,8952	Mucoromycetes	0,8804	merge_model1
Mucoromycetes	0,9254	Mucoromycetes	0,9135	Mucoromycetes
merge_model3	0,9251	Eurotiomycetes	0,9027	Eurotiomycetes
Zoopagomycota	0,8976	Eurotiomycetes	0,7468	Sordariomycetes

Table 9.2: Results of an analysis of "best models" for each class/subphylum/phylum

The table clearly shows that all 38 models are not required. Upon initial examination, it becomes apparent that only 6 models need to be thought of as potential for donor: Dothideomycetes, Eurotiomycetes, Sordariomycetes, Taphrinomycotina, Agaricomycetes, and Mucoromycetes. Similarly, 8 models are considered adequate for acceptor: Eurotiomycetes, Sordariomycetes, Dothideomycetes, Agaricomycetes, Pucciniomycotina, Ustilaginomycotina, merge_model1, and Mucoromycetes. Only 14 models have been deemed necessary, which is a substantial decrease from the original 38.

Still, there is an opportunity for improvement, since certain models just determine the best class for one model. By relaxing the criterion for the highest possible score, these models can be substituted with ones that occur several times, improving both efficiency and time complexity. Upon thorough analysis, the subsequent modifications can be executed:

- Donor:
  - Sordariomycetes (0.9122) → Eurotiomycetes (0.9115)
  - Taphrinomycotina (0.8031) → Eurotiomycetes (0.7770)
- Acceptor:
  - Dothideomycetes (0.839) → Eurotiomycetes (0.8311)

- Ustilaginomycotina (0.8618) → Pucciniomycotina (0.8617)
- merge_model1 (0.8804) → Mucoromycetes (0.8417)

As a result of these modifications, the number of necessary models has been reduced to 4 for donor (Dothideomycetes, Agaricomycetes, Eurotiomycetes and Mucoromycetes) and 5 for acceptor (Eurotiomycetes, Agaricomycetes, Mucoromycetes, Pucciniomycotina and Sordariomycetes), making a total of 9 models required to thoroughly assess the entire Fungi kingdom.

The results reveal a noticeable pattern, especially apparent when evaluating models from the Ascomycota and Basidiomycota phyla. Models belonging to the Ascomycota phylum are consistently most accurately assessed by models from the same phylum, and a similar pattern is seen for models within the Basidiomycota phylum. The fact that there is internal compatibility among phyla indicates that there is a certain level of specialisation. This highlights the effectiveness of models when they are trained and evaluated within their taxonomic groups. However, models from phyla with extensive datasets, regardless of their taxonomic classification, generally demonstrate better performance when evaluating models from other phyla.

An attempt was made to address a low score by transferring the dataset of the Zoopagomycetes phylum to the Agaricomycetes acceptor model. The experiment resulted in an enhancement of the Zoopagomycota score from 0.7381 to 0.7436. In addition to this favourable alteration, four additional scores experienced a rise. Nevertheless, the drop was observed in thirteen scores, limiting the applicability of this model to only five unique models. Upon completion of the examination of "best models," it was determined that the Sordariomycetes model emerged as the most optimal choice for the phylum Zoopagomycota. Another attempt at transfer learning was made in order to enhance the score.

However, upon comparing the scores of this model with those of the "best models" in Table 9.3, they are shown to be inconsequential. The highest F1 score is still produced by the "best model". The inefficiency of transfer learning can be attributed to the computationally demanding requirements and the crucial necessity for optimal performance in the computer system. Based on the computational complexity and performance limitations, it was determined that transfer learning may not be a viable method to achieve the desired outcomes within the current limits. Further testing and experimentation would be necessary.

Models	Agaricomycetes	transfer model (A)	"best model"	transfer model (S)
Zoopagomycota	0,7381	0,7436	0,7468	0,7454
Pezizomycetes	0,8741	0,8513	0,8943	0,8365
Sordariomycetes	0,8844	0,8589	0,9107	0,8677

Table 9.3: Comparison of models for phylum Zoopagomycota; usage of transfer learning

9.3 Sequence logos

Since the logos generated from the taxonomy-based evaluation did not provide sufficient information about the models, new logos were designed. However, this time the assessment was done based on the findings in 9.2 where 9 models were found to be optimal for covering the whole kingdom. The revised logos can also be found in the Appendices.

The newly generated sequence logos demonstrate the strength and effectiveness of the created models, visually representing their ability to distinguish patterns. These logos not only visually display the models' competence but also reinforce the selections made in the previous section regarding the choice of the best models. An in-depth analysis of the model's performance, as shown in Figure 9.1, specifically examines a region that makes up an intron. This region is carefully highlighted to emphasise the presence of dimers and the branch point. This magnified depiction impressively demonstrates the models' capacity to precisely recognise and differentiate crucial characteristics within genomic sequences. The logos provide clear and precise information, which enhances understanding and confirms that the selected models are appropriate for splice site prediction.

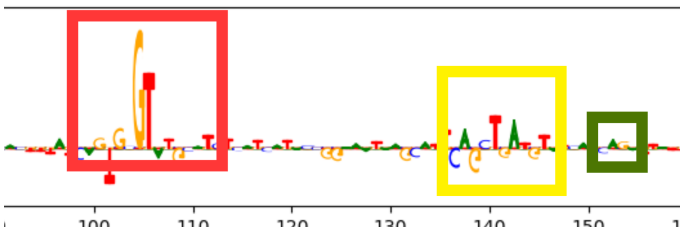


Figure 9.1: Zoom in on one of the logos showcasing the 5' splice site, branch point and 3' splice site

The comparisons conducted by DiffLogo helped clarify the reasoning behind the choice of optimal models. The provided Figure 9.2 presents a convincing visual representation of motif comparisons among organisms belonging to

the phylum Basidiomycota. The noticeable differences identified among the motifs highlight the natural variety in this biological group. The difference in traits between organisms justifies the use of separate models for each organism, taking into account the specific features of their motif compositions.

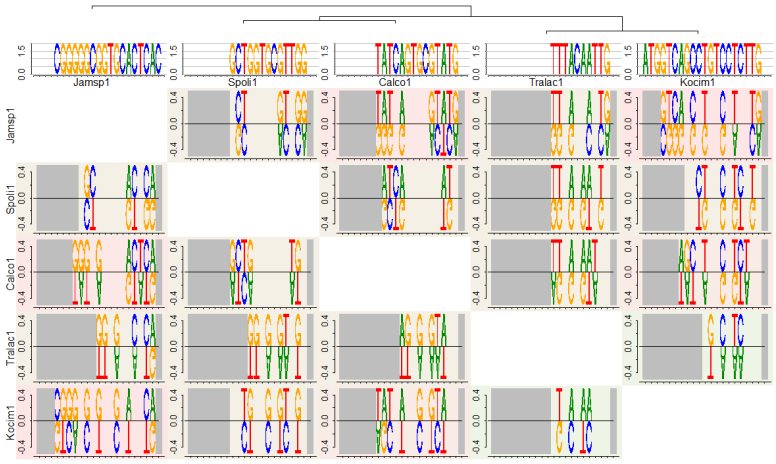


Figure 9.2: Comparison of motifs of interest of models from the Basidiomycota phylum

In contrast, Figure 9.3 presents a comparison of organisms evaluated using the Agaricomycetes model, illustrating a different scenario. The patterns seen in this context demonstrate a striking resemblance, confirming the effectiveness of the Agaricomycetes model.

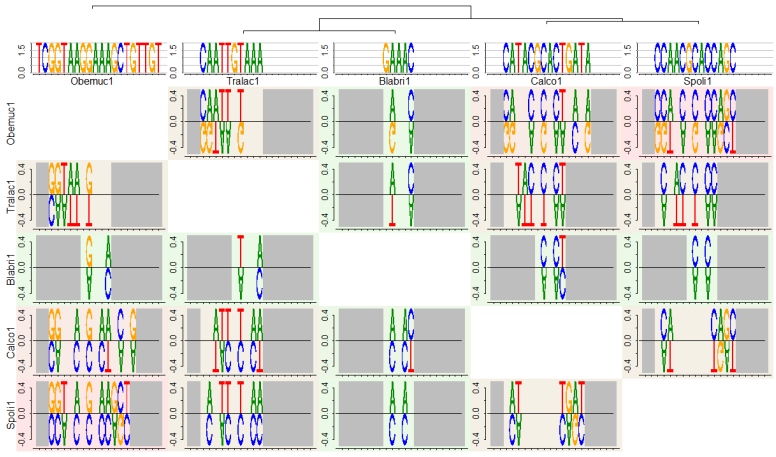


Figure 9.3: Comparison of motifs of interest of organisms evaluated by Agaricomycetes model

9.4 Final assessment of chosen models

The comparison of the selected "best models" with the phylum models and outcomes from the general models may be found in Table 9.4. The selected models demonstrate a higher capacity to accurately identify splice sites compared to the general models. In addition, the previous issue of general models struggling with underrepresented phyla has been resolved by the new models, which consistently yield high results regardless of the size of the dataset.

	"best model"		**phylum model**		**whole model**	
	donor	*acceptor*	*donor*	*acceptor*	*donor*	*acceptor*
Ascomycota	0,90	0,90	0,84	0,84	0,82	0,82
Basidiomycota	0,95	0,91	0,85	0,78	0,79	0,79
Mucoromycota	0,93	0,91	0,87	0,77	0,83	0,72
Chytridiomycota	0,88	0,89	0,77	0,65	0,71	0,71
Zoopagomycota	0,90	0,75	0,68	0,59	0,75	0,79
Blastocladiomycota	0,93	0,86	0,73	0,58	0,84	0,84

Table 9.4: Comparison of "best models" with phylum models and the general models

While constructing models, a computer failure occurred. The malfunctioning of the CPU's overhead fan resulted in a decline in performance. Due to the significant influence of hardware issues on the overall performance of the system, it is not possible to establish a dependable correlation between time and performance.

Nevertheless, in theory, the computational requirement is directly proportional to the size of the datasets. Based on this information, it can be assumed that the task of constructing large models for all 862 creatures (862 x 2 = 1724) is time-consuming and quite complex. On the other hand, creating smaller models for donors (113+223+135+31) and acceptors (135+223+31+27+120), totalling 1038 significantly reduces complexity and time. This is because the thousands of organisms are trained individually rather than all together. The non-linearity of complexity suggests that creating numerous smaller models may be a more efficient strategy compared to using a single large model.

The selected models should outperform the general models in intron detection when applied to metagenoms. This is because, unlike the general models, they were trained on more specific data, allowing them to understand both the overall patterns and the unique properties of each class. However, this efficiency is achieved by sacrificing the requirement for speed, as the total

assessment process is more time-consuming.

Chapter 10

Conclusions

This thesis aimed to investigate the efficacy of transfer learning and model explanation methods in the realm of the fungus kingdom. The primary objective was to provide the smallest number of models needed to achieve full coverage while also considering time complexity and computational resources.

Two pipelines were created: one for augmentation and another for the intention of transfer learning. Upon determining the most effective configurations for both approaches, a model created using transfer learning emerged as the superior performer, surpassing the older approach by a significant margin. During the investigation into reducing the number of necessary models, a constraint on transfer learning became apparent. While it improved results for the specific class/sub-phylum/phylum that was used, it resulted in a decline in performance for the other classes/sub-phyla/phyla. As a result, a group of ideal models was identified, suggesting a total of nine models as the optimal option - four for donors (Dothideomycetes, Agaricomycetes, Eurotiomycetes, and Mucoromycetes) and five for acceptors (Eurotiomycetes, Agaricomycetes, Mucoromycetes, Pucciniomycotina, and Sordariomycetes).

Sequence logos were chosen for explanation purposes due to their widespread application in the analysis of biological sequences. At first, a total of 38 logos were produced, with a pair for each organism. One logo was created by the donor model and the other by the acceptor. These logos demonstrated that certain models exhibited superior proficiency in identifying dimers compared to others as some models displayed a tendency to emphasise the base T instead of finding the splice sites. Consequently, the analysis of these pairs generated by DiffLogo did not result in any substantial findings. Following

that, a separate set of sequence logos was created, specifically targeting the models that were determined to be optimal.

The newly created sequence logos effectively demonstrated the strength of the models in accurately predicting splice sites within the fungal kingdom. These logos visually depict the models' capacity to precisely recognise crucial characteristics in genetic sequences. Furthermore, the comparison of the motifs of interest offered strong backing for the previous choice in selecting the most suitable models. The visual evidence provided confirms the effectiveness of the models and emphasises their potential as valuable tools for comprehending genetic patterns in the classification of the fungal kingdom. This is in accordance with the goal of creating efficient models.

This study offers the potential for further research. During experimentation, it was discovered that utilising transfer learning is advantageous. However, implementing it to determine the minimum number of required models is exceedingly difficult and time-consuming. There is a potential for exploring transfer learning to either utilise it or create a novel approach for optimising models. Additionally, as the package used for logo assessment just provides the differences, there's an opportunity to create a new instrument for logo sequences that highlight similarities.

Bibliography

[1] Chapter 11 - eukaryotic rna processing*. In Thomas D. Pollard, William C. Earnshaw, Jennifer Lippincott-Schwartz, and Graham T. Johnson, editors, *Cell Biology (Third Edition)*, pages 189–207. Elsevier, third edition edition, 2017.

[2] Denis Baručić. *Automatic intron detection in fungal genomes using machine learning*. PhD thesis, 2019.

[3] Oliver Bembom. Sequence logos for dna sequence alignments, May 11 2020. Retrieved January 6, 2024.

[4] Meredith Blackwell. The fungi: 1, 2, 3 ... 5.1 million species? *American Journal of Botany*, 98(3):426–438, 2011.

[5] Jan Brabec and Lukas Machlica. Bad practices in evaluation methodology relevant to class-imbalanced problems, 12 2018.

[6] Lori Carris, Christopher Little, and Carol Stiles. Introduction to fungi. *Plant Health Instructor*, 01 2012.

[7] Suzanne Clancy, 2008.

[8] Gavin E. Crooks, Steven E. Brenner, Gary Hon, and John-Marc Chandonia, 2004.

[9] Gavin E. Crooks, Gary Hon, John M. Chandonia, and Steven E. Brenner. Weblogo: A sequence logo generator. *Genome Research*, 14(6):1188–1190, June 2004.

[10] H C Dube. *An Introduction to Fungi (4th edition)*, pages 23–25. Scientific Publishers (India), 2013.

[11] NYU Gencore. Gff3 format tutorial, Year of Access. Accessed on: Date of Access.

[12] Gurney Gevin. *What are neural networks?*, page 13–16. UCL Press, 1997.

[13] Georgios Giannoulis, Anastasia Krithara, Christos Karatsalos, and Georgios Paliouras. Splice site recognition using transfer learning. *Artificial Intelligence: Methods and Applications*, page 341–353, 2014.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[15] Igor V Grigoriev, Roman Nikitin, Sajeet Haridas, Alan Kuo, Robin Ohm, Robert Otillar, Robert Riley, Asaf Salamov, Xueling Zhao, Frank Korzeniewski, Tatyana Smirnova, Henrik Nordberg, Inna Dubchak, and Igor Shabalov. Mycocosm portal: gearing up for 1000 fungal genomes. *Nucleic Acids Research*, 42(D1):D699–D704, 2014.

[16] Zhang Group. Fasta: Zhang group, 2023. Accessed on: 29.12.2023.

[17] Michelle L Hastings and Adrian R Krainer. Pre-mrna splicing in the new millennium. *Current Opinion in Cell Biology*, 13(3):302–309, June 2001.

[18] Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 191–200, 2016.

[19] Alick Henrici, D. Hawksworth, Paul Kirk, B. Sutton, and D. Pegler. Ainsworth bisby's dictionary of the fungi (8th edition). 1995.

[20] Martin Indra. *Automatic Intron Detection in Metagenomes Using Neural Networks*. PhD thesis, 2020.

[21] Sheikh Inzamamuzzaman, Abdullah Al-Sakin, Ms Syeda, and Shithi Maitra. *Detection of Early Alzheimer's Disease Applying Multi-layer Neural Networks*. PhD thesis, 06 2019.

[22] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[23] Simos Kazantzidis, Anastasia Krithara, and Georgios Paliouras. Splice site prediction: Transferring knowledge across organisms. In *Bioinformatics*, 2017.

[24] E B Keller and W A Noon. Intron splicing: a conserved internal signal in introns of animal pre-mrnas. *Proceedings of the National Academy of Sciences*, 81(23):7417–7420, 1984.

[25] Alice Lacan, Michèle Sebag, and Blaise Hanczar. GAN-based data augmentation for transcriptomics: survey and comparative assessment. *Bioinformatics*, 39(Supplement$_1$) : $i111 - -i120, 062023$.

[26] Anh Vu Le, Tomáš Větrovský, Denis Barucic, Joao Pedro Saraiva, Priscila Thiago Dobbler, Petr Kohout, Martin Pospíšek, Ulisses Nunes da Rocha, Jiří Kléma, and Petr Baldrian. Improved recovery and annotation of genes in metagenomes through the prediction of fungal introns. *Molecular Ecology Resources*, 23(8):1800–1811, 2023.

[27] Nicholas Keone Lee, Ziqi Tang, Shushan Toneyan, and Peter K. Koo. Evoaug: improving generalization and interpretability of genomic deep neural networks with evolution-inspired data augmentations. *Genome Biology*, 24(1):105, May 2023.

[28] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[29] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1), 2021.

[30] Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

[31] Mason Minot and Sai T Reddy. Nucleotide augmentation for machine learning-guided protein engineering. *Bioinform Adv*, 3(1):vbac094, December 2022.

[32] Matt Ervin Mital, Rogelio Ruzcko Tobias, Herbert Villaruel, Jose Martin Maningo, Robert Kerwin Billones, Ryan Rhay Vicerra, Argel Bandala, and Elmer Dadios. Transfer learning approach for the classification of conidial fungi (genus aspergillus) thru pre-trained deep learning models. In *2020 IEEE REGION 10 CONFERENCE (TENCON)*, pages 1069–1074, 2020.

[33] Nicholas P. Money. Chapter 1 - fungal diversity. In Sarah C. Watkinson, Lynne Boddy, and Nicholas P. Money, editors, *The Fungi (Third Edition)*, pages 1–36. Academic Press, Boston, third edition edition, 2016.

[34] Miguel A. Naranjo-Ortiz and Toni Gabaldón. Fungal evolution: diversity, taxonomy and phylogeny of the fungi. *Biological Reviews*, 94(6):2101–2137, 2019.

[35] Martin Nettling, Hendrik Treutler, Jan Grau, et al. Difflogo: a comparative visualization of sequence motifs. *BMC Bioinformatics*, 16(387), 2015.

[36] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[37] Thomas P. Quinn, Sunil Gupta, Svetha Venkatesh, and Vuong Le. A field guide to scientific XAI: transparent and interpretable deep learning for bioinformatics research. *CoRR*, abs/2110.08253, 2021.

[38] J. Raviv. Decision making in markov chains applied to the problem of pattern recognition. *IEEE Transactions on Information Theory*, 13(4):536–551, 1967.

[39] Thomas D. Schneider and R.Michael Stephens. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Research*, 18(20):6097–6100, 1990.

[40] Shmuel Schwartz, João Silva, David Burstein, Tal Pupko, Eduardo Eyras, and Gil Ast. Large-scale comparative analysis of splicing signals and their corresponding splicing factors in eukaryotes. *Genome Research*, 18(1):88–103, 2007.

[41] Gabriele Schweikert, Christian Widmer, Bernhard Schölkopf, and Gunnar Rätsch. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. volume 8, pages 1433–1440, 01 2008.

[42] Ling Shao, Fan Zhu, and Xuelong Li. Transfer learning for visual categorization: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 26(5):1019–1034, 2015.

[43] C. Shorten, T. M. Khoshgoftaar, and B. Furht. Text data augmentation for deep learning. *Journal of Big Data*, 8(1), July 2021.

[44] Lavanya Shukla. Designing your neural networks, Sep 2019.

[45] Noopur Singh, Ravindra Nath, and Dev Bukhsh Singh. Splice-site identification for exon prediction using bidirectional lstm-rnn approach. *Biochemistry and Biophysics Reports*, 30:101285, 2022.

[46] J. Smith, A. Johnson, and M. Doe. *American Journal of Botany*, 98(3):426–438, 2011.

[47] Millard Susman. *Genes: Definition and Structure*. 04 2001.

[48] Ammar Tareen and Justin B Kinney. Logomaker: beautiful sequence logos in Python. *Bioinformatics*, 36(7):2272–2274, 12 2019.

[49] S. Tyekucheva and F. Chiaromonte. Augmenting the bootstrap to analyze high dimensional genomic data. *TEST*, 17(1):1–18, March 2008.

[50] Omar Wagih. ggseqlogo: a versatile R package for drawing sequence logos. *Bioinformatics*, 33(22):3645–3647, 07 2017.

[51] Sarah C. Watkinson, Lynne Boddy, and Nicholas Money. *The Fungi*. Academic Press, December 2015. `http://books.google.ie/books?id=x8qcBAAAQBAJ&pg=PR4&dq=978-0-12-382034-1&hl=&cd=1&source=gbs_api`.

[52] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1), 2016.

[53] Christopher T. Workman, Yutong Yin, David L. Corcoran, Trey Ideker, Gary D. Stormo, and Panayiotis V. Benos. enoLOGOS: a versatile web tool for energy normalized sequence logos. *Nucleic Acids Research*, 33(suppl$_2$) : $W389 - -W392, 072005$.

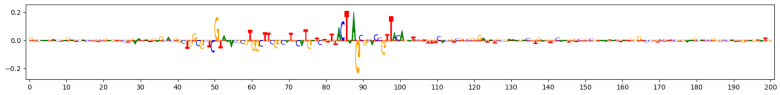
[54] Qian Xu and Qiang Yang. A survey of transfer and multitask learning in bioinformatics. *JCSE*, 5:257–268, 09 2011.

[55] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12133–12143. PMLR, 18–24 Jul 2021.

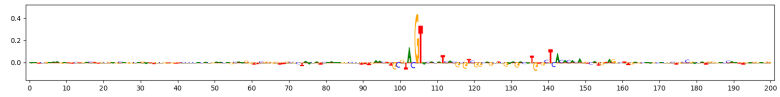
[56] Ting-He Zhang, Mario Flores, and Yufei Huang. Es-arcnn: Predicting enhancer strength by using data augmentation and residual convolutional neural network. *Analytical Biochemistry*, 618:114120, 2021.

[57] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *CoRR*, abs/1911.02685, 2019.

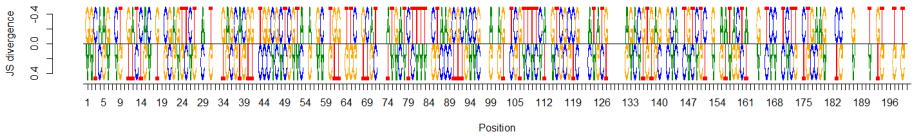
Appendices



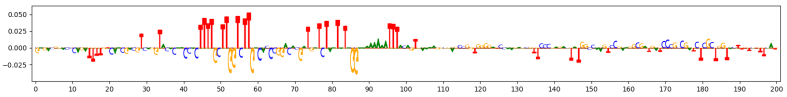
(a) : Logo of intron evaluated by acceptor model for Tralac1



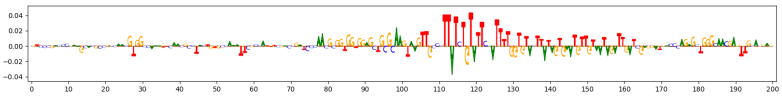
(b) : Logo of intron evaluated by donor model for Tralac1



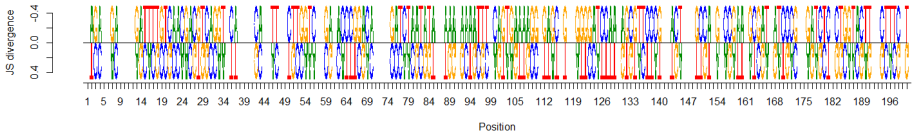
(c) : Comparison of the logos for Tralac1



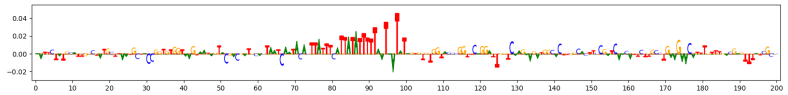
(d) : Logo of intron evaluated by acceptor model for Blabri1



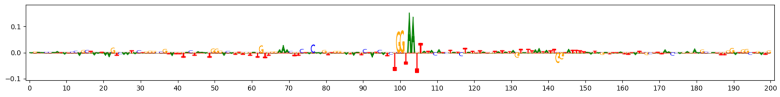
(e) : Logo of intron evaluated by donor model for Blabri1



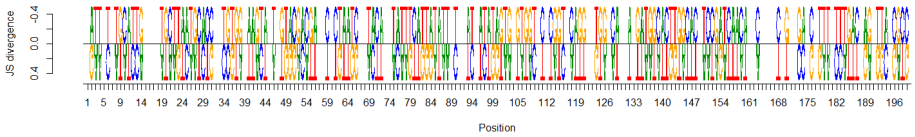
(f) : Comparison of the logos for Blabri1



(g) : Logo of intron evaluated by acceptor model for Synfus1

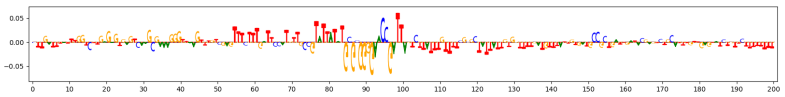


(h) : Logo of intron evaluated by donor model for Synfus1

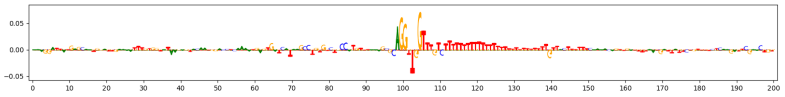


(i) : Comparison of the logos for Synfus1

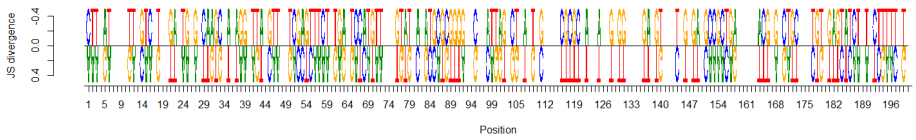
Figure 1: Visualisation of introns in different organisms evaluated by acceptor and donor models corresponding to their taxonomy, and their comparison



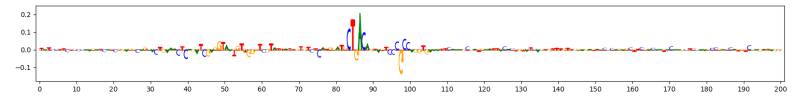
(j) : Logo of intron evaluated by acceptor model for Obemuc1



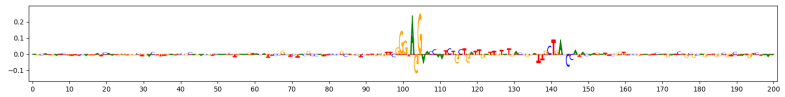
(k) : Logo of intron evaluated by donor model for Obemuc1



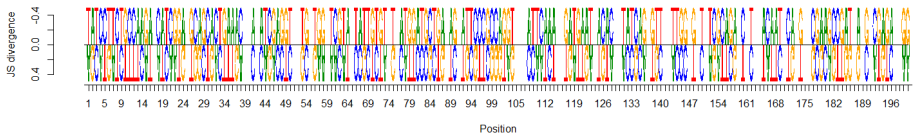
(l) : Comparison of the logos for Obemuc1



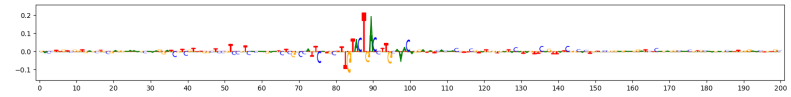
(m) : Logo of intron evaluated by acceptor model for Disac1



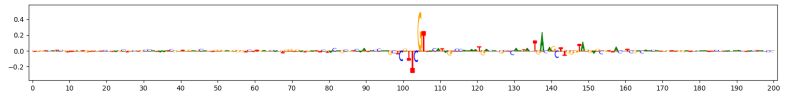
(n) : Logo of intron evaluated by donor model for Disac1



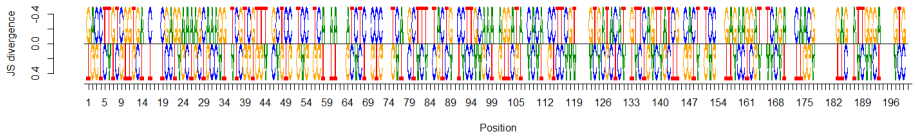
(o) : Comparison of the logos for Disac1



(p) : Logo of intron evaluated by acceptor model for Claim1

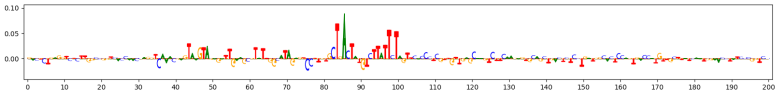


(q) : Logo of intron evaluated by donor model for Claim1

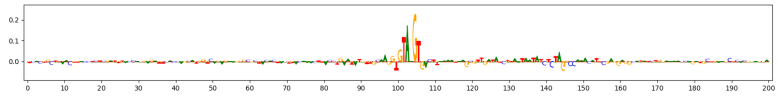


(r) : Comparison of the logos for Claim1

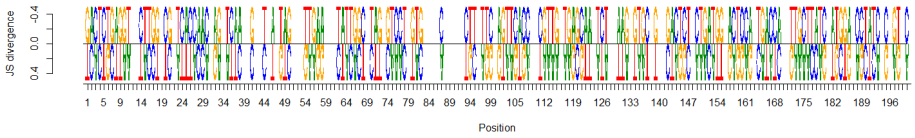
Figure 1: Visualisation of introns in different organisms evaluated by acceptor and donor models corresponding to their taxonomy, and their comparison



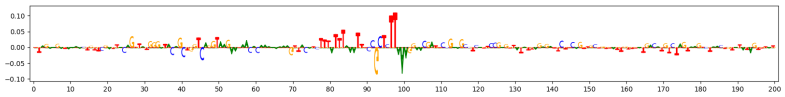
(s) : Logo of intron evaluated by acceptor model for Bissp1



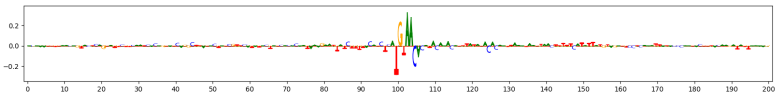
(t) : Logo of intron evaluated by donor model for Bissp1



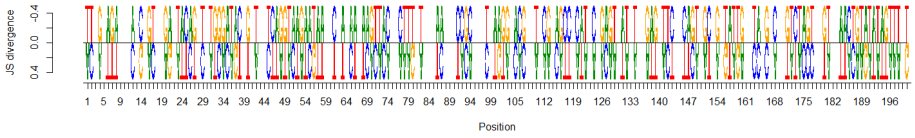
(u) : Comparison of the logos for Bissp1



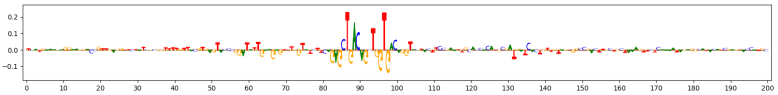
(v) : Logo of intron evaluated by acceptor model for Pilano1



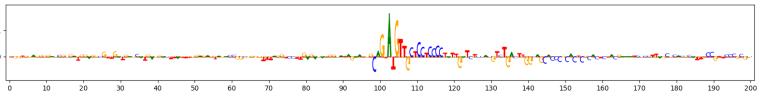
(w) : Logo of intron evaluated by donor model for Pilano1



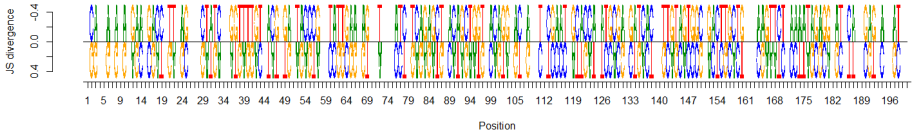
(x) : Comparison of the logos for Pilano1



(y) : Logo of intron evaluated by acceptor model for Sarco1

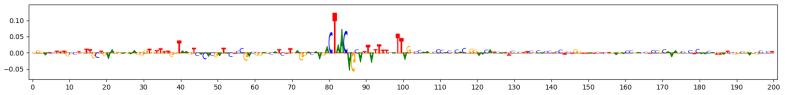


(z) : Logo of intron evaluated by donor model for Sarco1

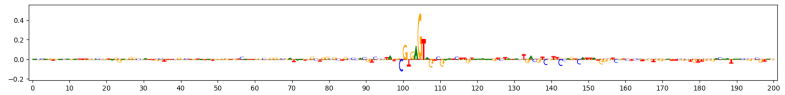


() : Comparison of the logos for Sarco1

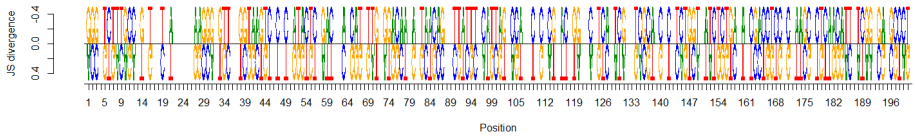
Figure 1: Visualisation of introns in different organisms evaluated by acceptor and donor models corresponding to their taxonomy, and their comparison



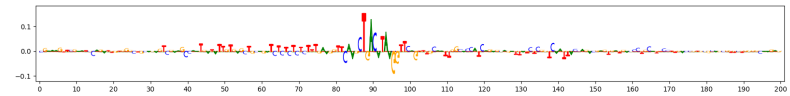
() : Logo of intron evaluated by acceptor model for Colgr1



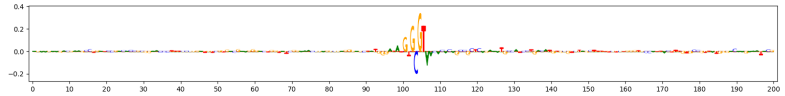
() : Logo of intron evaluated by donor model for Colgr1



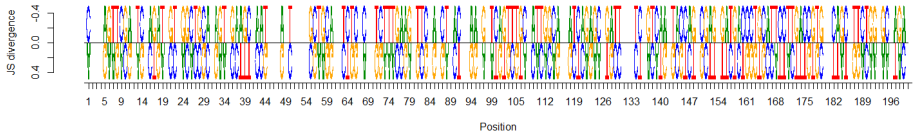
() : Comparison of the logos for Colgr1



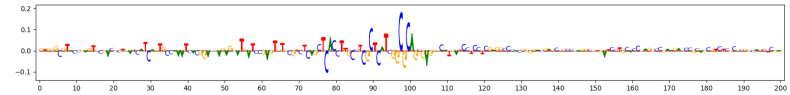
() : Logo of intron evaluated by acceptor model for Kocim1



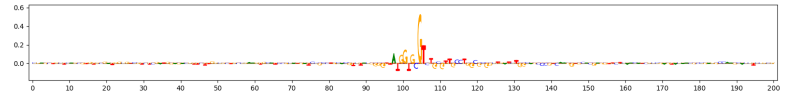
() : Logo of intron evaluated by donor model for Kocim1



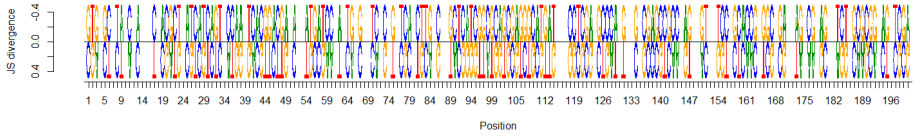
() : Comparison of the logos for Kocim1



() : Logo of intron evaluated by acceptor model for Spoli1

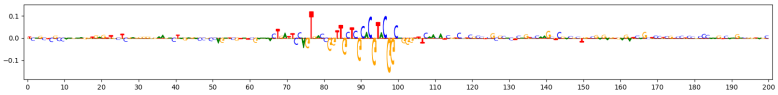


() : Logo of intron evaluated by donor model for Spoli1

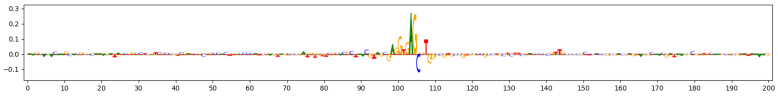


() : Comparison of the logos for Spoli1

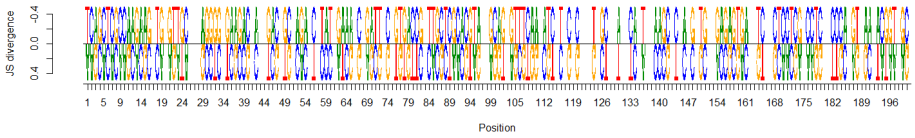
Figure 1: Visualisation of introns in different organisms evaluated by acceptor and donor models corresponding to their taxonomy, and their comparison



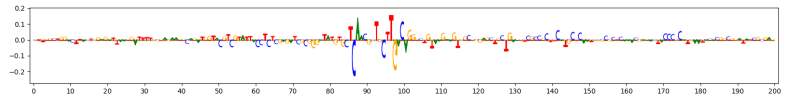
() : Logo of intron evaluated by acceptor model for Jamsp1



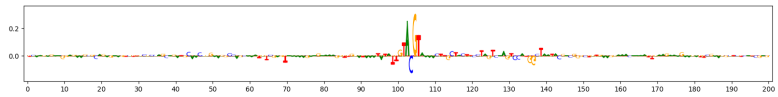
() : Logo of intron evaluated by donor model for Jamsp1



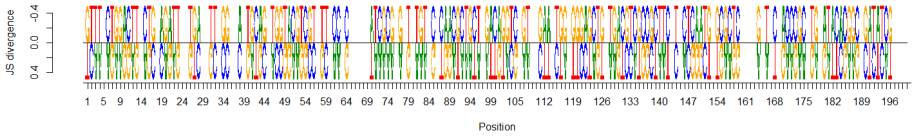
() : Comparison of the logos for Jamsp1



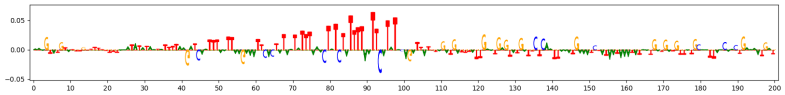
() : Logo of intron evaluated by acceptor model for Saico1



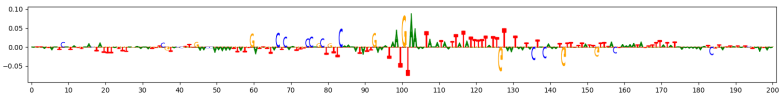
() : Logo of intron evaluated by donor model for Saico1



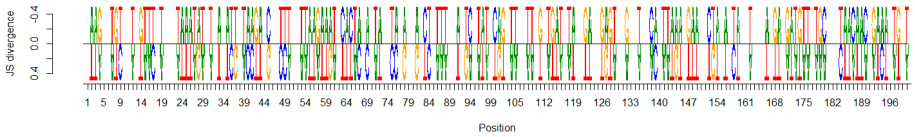
() : Comparison of the logos for Saico1



() : Logo of intron evaluated by acceptor model for Anasp1

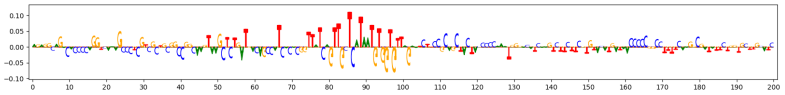


() : Logo of intron evaluated by donor model for Anasp1

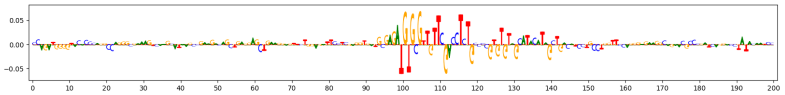


() : Comparison of the logos for Anasp1

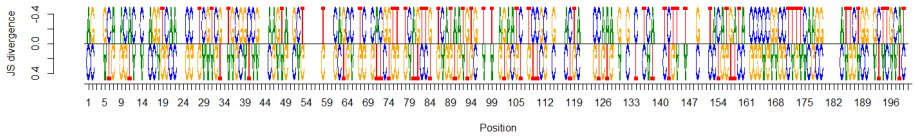
Figure 1: Visualisation of introns in different organisms evaluated by acceptor and donor models corresponding to their taxonomy, and their comparison



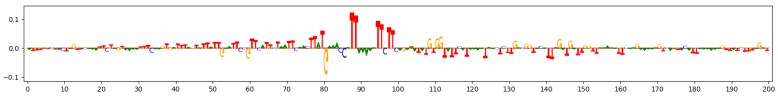
() : Logo of intron evaluated by acceptor model for Calco1



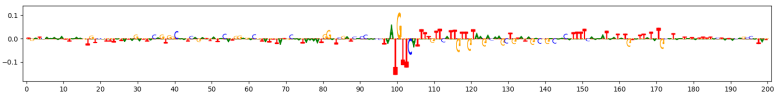
() : Logo of intron evaluated by donor model for Calco1



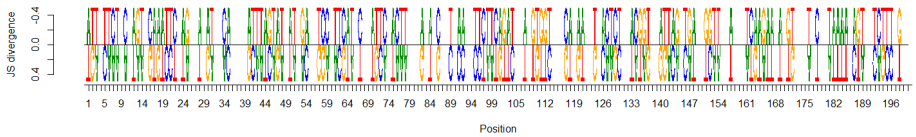
() : Comparison of the logos for Calco1



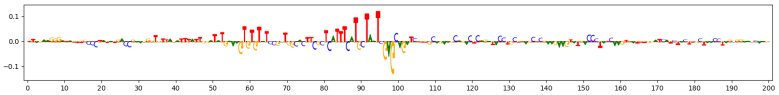
() : Logo of intron evaluated by acceptor model for Gloin1



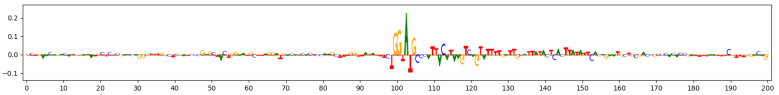
() : Logo of intron evaluated by donor model for Gloin1



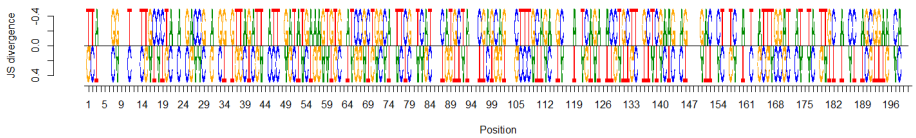
() : Comparison of the logos for Gloin1



() : Logo of intron evaluated by acceptor model for Symko1

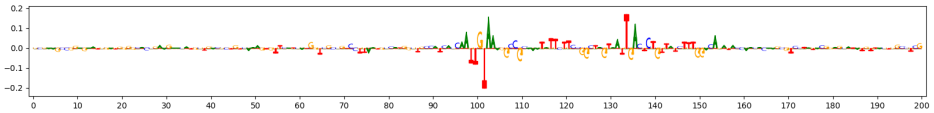


() : Logo of intron evaluated by donor model for Symko1

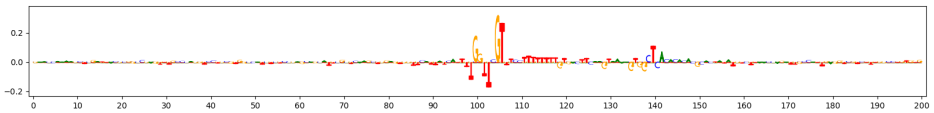


() : Comparison of the logos for Symko1

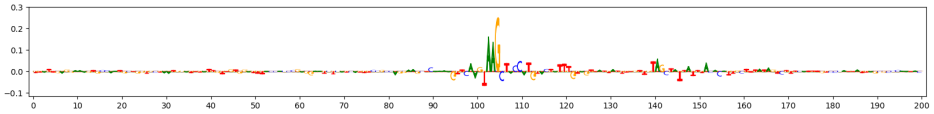
Figure 1: Visualisation of introns in different organisms evaluated by acceptor and donor models corresponding to their taxonomy, and their comparison



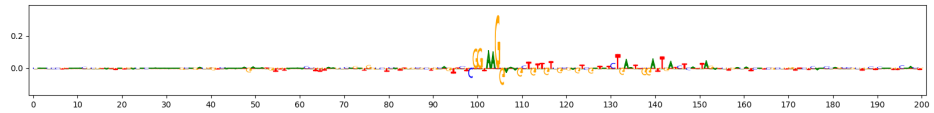
(a) : Tralac1



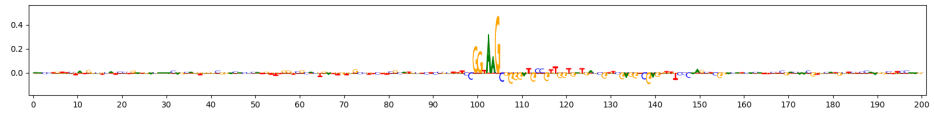
(b) : Blabri1



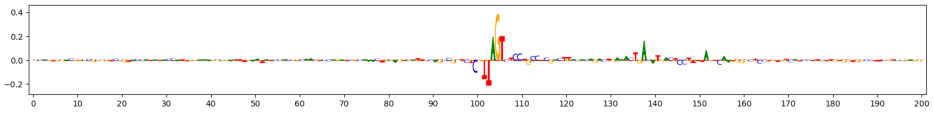
(c) : Synfus1



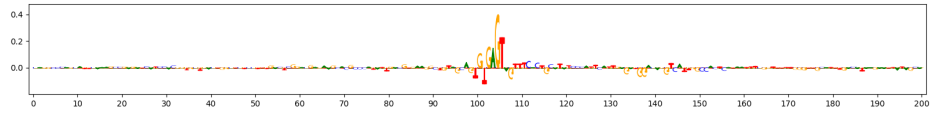
(d) : Obemuc1



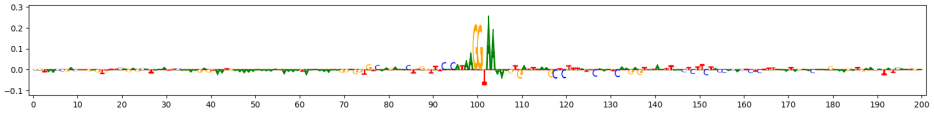
(e) : Disac1



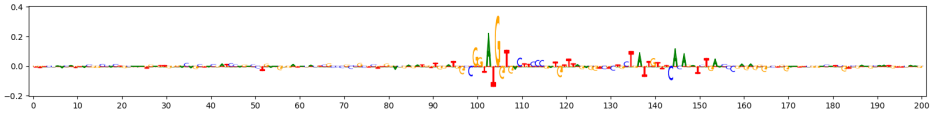
(f) : Claim1



(g) : Bissp1

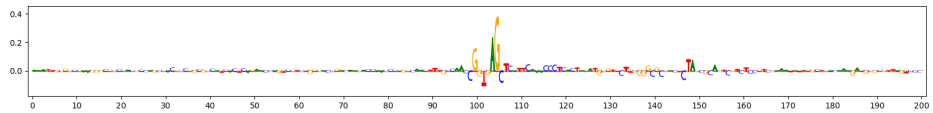


(h) : Pilano1

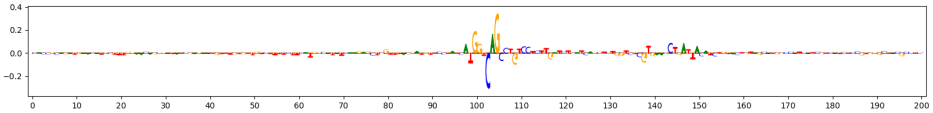


(i) : Sarco1

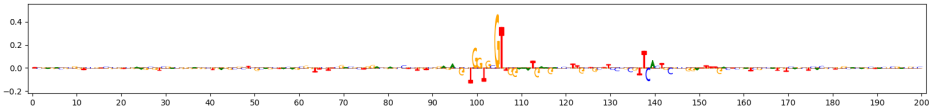
Figure 2: Visualisation of introns in different organisms evaluated by donor models from the optimal set



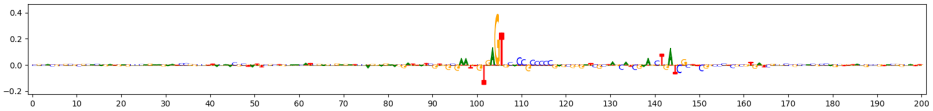
(j) : Colgr1



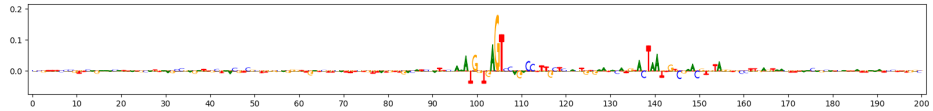
(k) : Kocim1



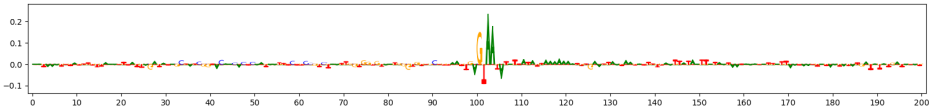
(l) : Spoli1



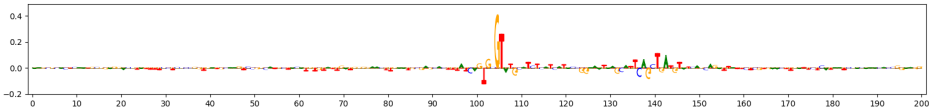
(m) : Jamsp1



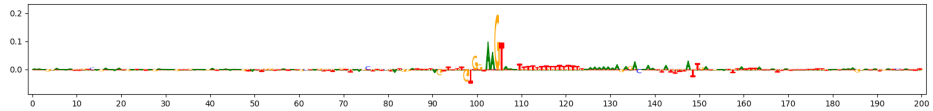
(n) : Saico1



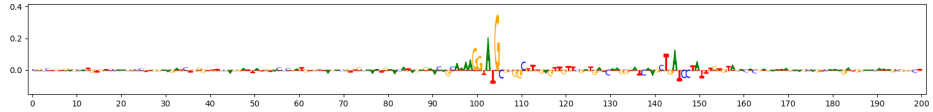
(o) : Anasp1



(p) : Calco1

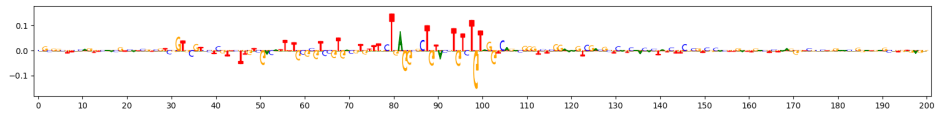


(q) : Gloin1

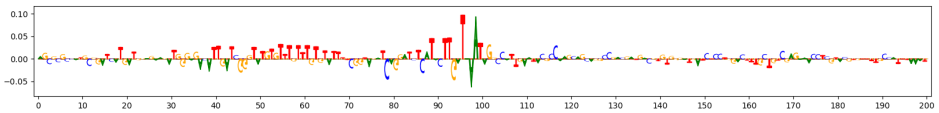


(r) : Symko1

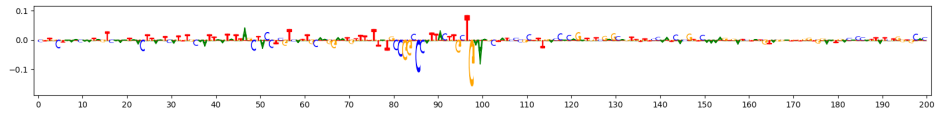
Figure 2: Visualisation of introns in different organisms evaluated by donor models from the optimal set (continued)



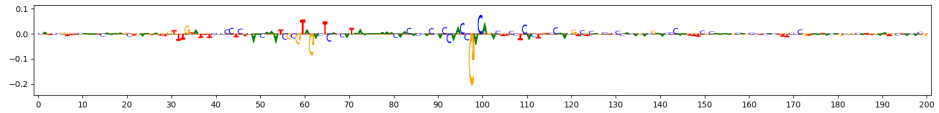
(a) : Tralac1



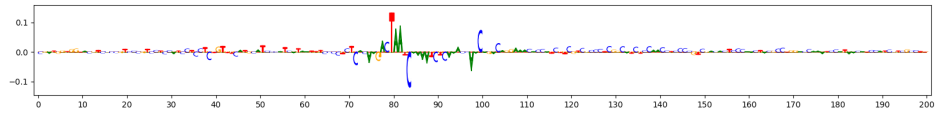
(b) : Blabri1



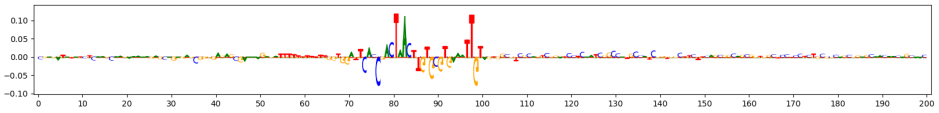
(c) : Synfus1



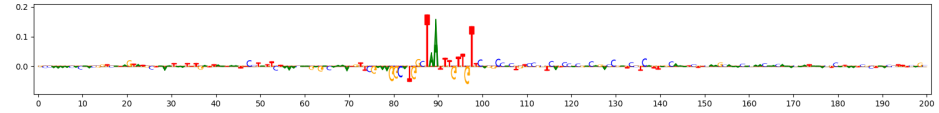
(d) : Obemuc1



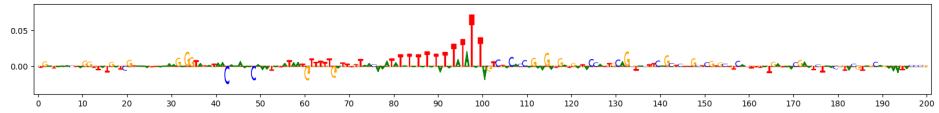
(e) : Disac1



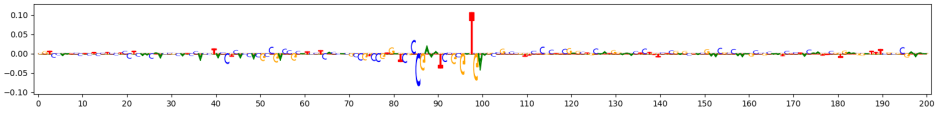
(f) : Claim1



(g) : Bissp1

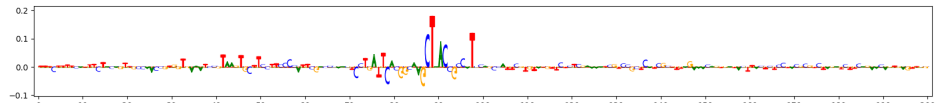


(h) : Pilano1

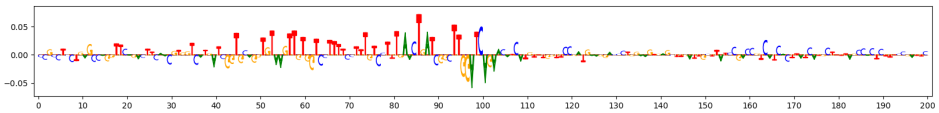


(i) : Sarco1

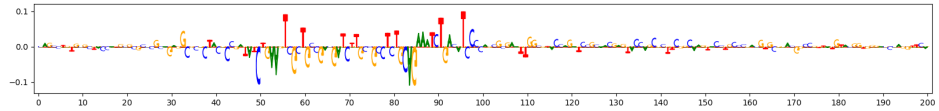
Figure 3: Visualisation of introns in different organisms evaluated by acceptor models from the optimal set



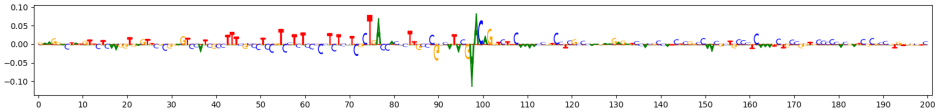
(j) : Colgr1



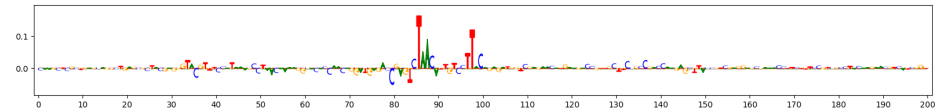
(k) : Kocim1



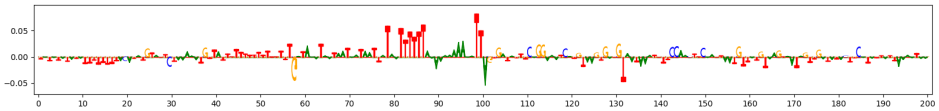
(l) : Spoli1



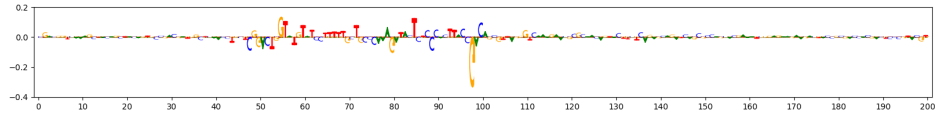
(m) : Jamsp1



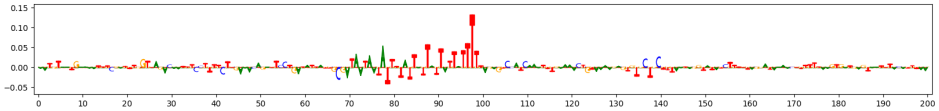
(n) : Saico1



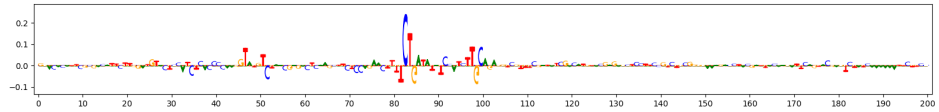
(o) : Anasp1



(p) : Calco1



(q) : Gloin1



(r) : Symko1

Figure 3: Visualisation of introns in different organisms evaluated by acceptor models from the optimal set (continued)

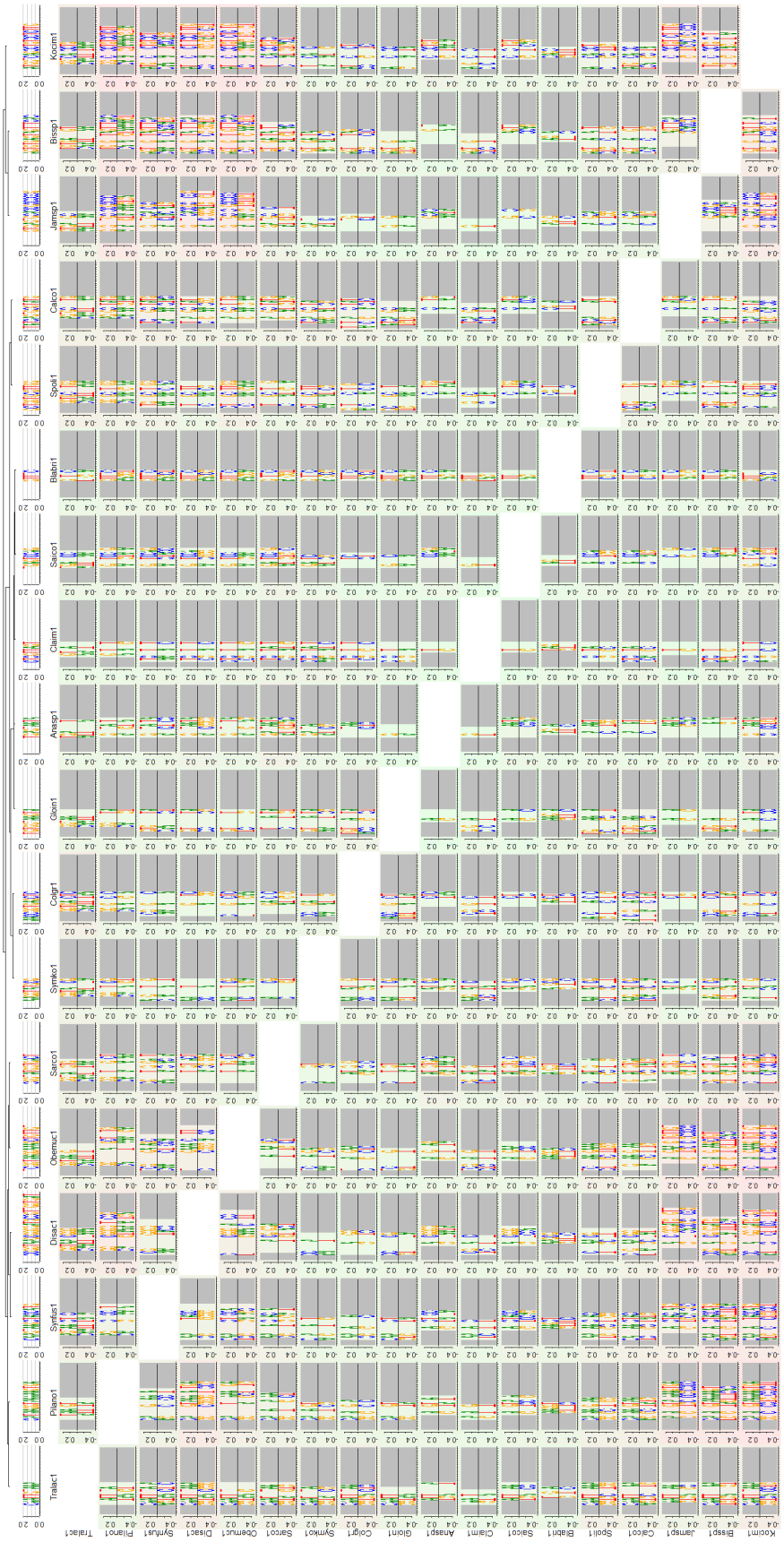


Figure 4: Comparison of motifs of interest of all models