Prediction of AL Amyloidosis Using Deep Learning

by

Anupa Murali

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

AL amyloidosis (amyloid light chain or primary amyloidosis) is a rare protein disorder that can be potentially fatal or can cause permanent damage to the organs in the body, especially in cases where the diagnosis does not arrive early enough or where the treatment does not begin on time. It is a type of amyloidosis, which occurs when abnormal immunoglobulin light chain (LC) proteins in the body misfold and accumulate on the heart, the kidneys, and the other organs. In order to facilitate timely diagnosis of the disease before the symptoms start fully exhibiting themselves and before the damage to the organs becomes significant, we present a computational solution in this thesis, called "DALAD", which is based on (convolutional) deep learning networks and takes in an LC sequence from a patient as the input, and determines with high confidence whether the patient has the disease or not. We develop and test multiple versions of DALAD, which are characterized by the type of sequences they have been trained on and by the types of features they incorporate to make the predictions, in order to have high performance in each of these scenarios. We establish the following for DALAD.

- DALAD is the first computational learning model to be able to accurately predict the onset of AL amyloidosis on both λ and κ LC sequences.

- DALAD comfortably beats the state-of-the-art for λ sequences in terms of accuracy measures, such as AUC score, sensitivity, and specificity. Our numbers for these three metrics are 0.89, 0.81, and 0.83, respectively, while for LICTOR, they are 0.87, 0.76, and 0.82, respectively.

- DALAD is able to utilize the features from both V and J gene segments of the LC sequences to make more accurate predictions. We additionally show via the pairwise t-test that the J gene segments do improve our performance against both λ and κ sequences.

- We provide aggregate statistics over multiple runs for each version of DALAD, along with the accuracy results for the best trained model corresponding to each version. All our findings indicate high prediction accuracy for both λ and κ sequences.

Thesis Supervisor: Bin Ma

Title: Professor, University Research Chair; Cheriton School of Computer Science, University of Waterloo

Acknowledgments

I would like to express my deepest gratitude to Professor Bin Ma, who has provided me his tireless guidance and unflagging faith in my ability to see this project through. I thank him for helping me learn so much about academia and how to approach a difficult problem repeatedly.

I would like to thank my thesis committee, Professor Helen Chen and Professor Lila Kari, whose suggestions and comments helped shape up this work. The hard questions they asked me made my efforts more focused and brought clarity.

I thank Daddy, for his guidance and many stories he told me about his days in CS grad school, and for his excitement and encouragement. I thank him for the immense amount of mathematics knowledge I learned from him, and for his inimitable love of learning, which I always found so inspiring.

I thank Amma, for modeling tenacity, grit and adaptability throughout my life, and for teaching me to be resourceful and optimistic, even when the road ahead is seemingly endless.

I thank Patti (my grandma) for her love and endless supply of delicious snacks, and for reminding me to laugh.

I thank my precious sister, Anusha, for always inspiring me with her creativity and energy, and for being someone who I love so much. I spent so many moments with her throughout my time at the University of Waterloo. When I wasn't working on my research, we were usually taking study breaks over Nutella milkshakes and anime, drawing wild pictures for each other, or venting about what's going on in our lives. Most of all, we were always there for each other no matter what and will continue to be forever. I wouldn't have gotten here if not for her love.

I thank my beloved fiancé, Vikrant, for listening to me, helping me so generously, and for bringing me dinner so many times when I was too busy or tired to cook for myself. I thank him for helping me proofread what with his immense research knowledge, bearing witness to all the ups and downs of this process, and holding my hand throughout.

Contents

List of Figures

in Section 4.1. We used `Google Colab` to train and test our models.

10. Figure 3.3 (Page 20): Creating the expanded custom germline database from IMGT.

11. Figure 3.4 (Page 21): We use IgBLAST to align the LC sequences from AL Base with the GL sequences from IMGT. This is an example of what the alignment would look like.

12. Figure 3.5 (Page 21): Aligned input sequences containing the V region. Each input sequence contains consecutive pairs of residues, side-by-side, such that each pair represents the LC residue and the corresponding GL residue at that location in the sequence (e.g., the first pair, labelled “L1/G1”, shows the LC residue and the GL residue, respectively, at the first position). This construction was made possible due to the alignment imposed by IgBLAST, as shown in Figure 3.4. The CNN’s in all the $\mathsf{DALAD}_{\mathrm{V}}$ and the $\mathsf{DALAD}_{\mathrm{VJ}}$ versions consider each of these pairs as a single feature, and end up using information from 107 such features.

13. Figure 3.6 (Page 22): Aligned input sequences containing the J region. The idea behind this representation is the same as in Figure 3.5. There are 12 such pairs or features, which are utilized by the CNN’s in all the $\mathsf{DALAD}_{\mathrm{VJ}}$ versions.

14. Figure 3.7 (Page 25): Sep model overview.

15. Figure 4.1 (Page 48): Separation histograms for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ (from the top-left, going clockwise).

16. Figure 4.2 (Page 50): ROC curves for three different situations. The one on the left in the top row compares the ROC curves of $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ just on their respective λ-only cmplement test datasets. The ROC curve on the right in the top row correspond to those of $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ on their respective κ-only complement test datasets. The figure in the bottom row show the ROC curves corresponding to $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ on their respective complement test datasets that contained a mixture of λ and κ sequences.

List of Tables

Chapter 1

Introduction

AL amyloidosis is a rare protein disorder that can be often fatal if it is not diagnosed and managed early enough. Amyloidosis, the broader group of protein disorders, which includes AL amyloidosis, is caused by misfolding of proteins, which clump together and form amyloid fibril deposits in major body organs. Immunoglobulin AL amyloidosis, the disorder under the current investigation, is specifically caused by small plasma cell antibodies produced in excess in the bone marrow that misfold, aggregate and deposit in the form of amyloid fibrils, leading to irreversible multi-organ dysfunction, typically the heart and the kidneys, and eventually to death. Here, the misfolding proteins in question are the light chain (LC) proteins in the body.

There are several options available for the treatment of the disease, which could either slow down or completely stop the process that causes the body to produce too many amyloids. They include medications and other types of treatments, such as chemotherapy, immunotherapy, or steroids, which work together to annihilate the plasma cells that manufacture these light chain proteins. However, these treatments still *cannot* reverse the damage that the organs in the body have already suffered, so it is imperative to be able to diagnose the disease in its early stages, and begin the treatment to curb the processes that cause such irreparable harm, before it becomes lethal. That said, the diagnosis of AL amyloidosis is often not easy as the signs and symptoms can be mistaken for those of other common diseases. When AL amyloidosis is suspected, in addition to blood and urine analysis for amyloid proteins, imaging of affected organs using MRI, echocardiogram, and nuclear imaging, and often even invasive biopsies of bone marrow and the affected organs are necessary. It is known that the median survival rate after the diagnosis is received is less than six months when the underlying plasma cell dyscrasia is left untreated in AL amyloidosis patients [19, 22]. Hence, it is paramount to develop novel diagnostic methodologies that are not just based on the signs and the symptoms of AL amyloidosis, but are based on the underlying molecular mechanisms of the amyloidogenic clone, which can provide evidences for predisposition much before the disease sets its course on the body. The existing literature suggests that some mutations in the light chain sequence can affect the stability of its protein structure, and can potentially lead to aggregation and deposit of the light chain on certain organs in the body [3, 13], leading to AL amyloidosis. This opens up the possibility of detecting the onset of the disease just by studying the light chain samples from the patients. Therefore, the problem of predicting AL amyloidosis based on the aforementioned strategy reduces to a problem of predicting what sequences of amino acids would result in the kind of proteins that are prone to misfolding, thereby forming amyloid fibrils.

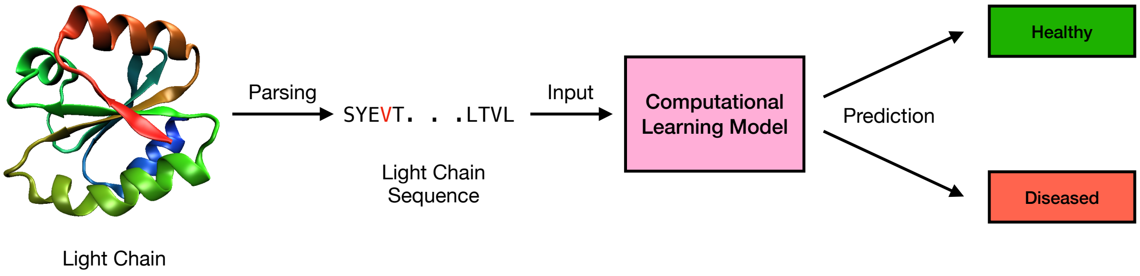


Figure 1.1: The goal is to create a computational learning model, which takes a light chain sequence from a patient as its input, and outputs a prediction saying whether the patient has AL amyloidosis or not.[1]

Our goal in this thesis is to create computational models that use LC sequences to learn about the behaviours of these sequences, and use that information to accurately predict whether a patient is suffering from the disease or not by looking at a sample of LC sequences from their body (see Figure 1.1). However, the LC sequences that lead to AL amyloidosis exhibit a high degree of

[1] “File:Thioredoxin-fold-1ert.png” by Opabinia regalis is licensed under CC BY-SA 3.0.

diversity, posing a major challenge to developing any efficient and accurate computational prediction model. This is because the large number of unique rearrangement of the variable (V) and the joining (J) region genes, compounded with a unique set of somatic mutations during the B-cell affinity maturations, makes it extremely difficult to identify the set of LC sequences that are likely to lead to AL amyloidosis. Therefore, the development of a novel computational biology method to predict AL amyloidosis from LC sequences would be a very significant achievement, both from a humanitarian perspective and also from a technical standpoint. It would also have the potential for other far-reaching implications in treating patients with AL amyloidosis, including the emergence of new drugs that can target specific mutations in the LC sequences.

In this work, we develop one such computational model, called "DALAD", which *efficiently* and *accurately* predicts the onset of the disease in patients by processing their samples of LC sequences. The novelty of our work lies in that: (1) we are able to predict whether a given LC amino acid sequence would result in misfolding proteins that eventually lead to AL amyloidosis by potentially using the combination of both the V and the J gene segments of the primary amino acid LC sequences and their corresponding germline sequences as input to a (convolutional) deep learning model, and in that (2) we can do that on both λ and κ LC sequences accurately.

1.1 Overview of Our Model and Results

Our model is a convolutional deep neural network (CNN), which uses information from LC sequences and their corresponding germline sequences to learn what mutations in the sequence tend to lead to the disease. We provide different versions of DALAD that are trained and designed for either just λ sequences or just κ sequences, or for a combination of both, and either just use the features from the V gene segment of the LC sequences (DALAD_{V} models) or the features from both the V and the J gene segments (DALAD_{VJ} models).

An AL patient usually has only one clone of abnormal free light chains that is associated with the disease. Depending on the patients, this clone can be either λ or κ. Since our work is the first one with the ability to classify both λ and κ sequences accurately, we benchmark DALAD against the state-of-the-art model (LICTOR [11]) that is designed solely for λ sequences, and provide independent results for κ sequences. In a nutshell, our results on λ sequences show remarkable on those of the state-of-the-art. For example, LICTOR has the AUC score of 0.87, the sensitivity of 0.76, and the specificity of 0.82 on λ sequences, but our best models have the AUC score of over 0.92, the sensitivity of over 0.86, and the specificity of over 0.85. The important improvement is that we are able to classify the positive sequences much more accurately, since it would be most useful for a patient having the disease to have the correct diagnosis as soon as possible.

In this thesis, we provide the experimental results for all our versions of DALAD in two forms after selecting a set of appropriate hyperparameters for each one of them.

1. We train and run each version of DALAD 100 times, and provide the aggregate statistics for different accuracy metrics, such as the AUC score, the sensitivity, the specificity, and the overall accuracy from each run.

2. We select the best trained model for each of these versions from these runs, and evaluate them on larger datasets than the ones they were previously tested on (which do not contain their respective original training datasets).

1.2 Related Work

Nearly all of the prior work in prediction of AL amyloidosis, which employs computational biology, either using mathematical or machine learning approaches, can be divided into two main categories: empirical and structure-based. Empirical approaches interpret experimental results and make predictions by identifying and considering the appropriate factors of the properties of the constituent amino acids. These properties include hydrophobicity, β-propensity and other physical properties such as solubility [22]. On the other hand, the models based on structure detect the factors responsible for amyloid aggregation by observing the existing three-dimensional (3D) structures of peptides that adopt a known fibrillar structure or native proteins that belong to distinct structural classes.

An algorithm called TANGO developed by Fernandez-Escamilla et al. predicts protein aggregation using the physico-chemical principles of beta-sheet formation, extended by the assumption that the core regions of an aggregate are fully buried [9]. Using statistical mechanics, TANGO predicts pathogenic as well as protective mutations of the Alzheimer beta-peptide, human lysozyme and transthyretin, and discriminates between beta-sheet propensity and aggregation, and therefore has the potential to predict amyloidosis [9].

There is evidence that the conversion from soluble states into cross-β fibrillar aggregates is a property shared by many different proteins, and that such fibrillar assemblies are generally characterized by a parallel in-register alignment of β-strands contributed by distinct protein molecules [23]. Therefore, assuming that a universal mechanism is responsible for β-structure formation, the algorithm PASTA (Prediction of Amyloid Structure Aggregation) developed by Trovato et al predicts amyloid structure aggregation [23].

Using the expected probability of hydrogen bond formation and expected packing density of residues, Garbuzynskiy et al developed an algorithm called FoldAmyloid, which can predict both amyloidogenic and disordered regions in protein chains [10]. FoldAmyloid exploits the fact that regions with strong expected packing density are responsible for amyloid formation. The predictions generated by FoldAmyloid are consistent with known disease-related amyloidogenic regions for eight of 12 amyloid-forming proteins and peptides in which the positions of amyloidogenic regions have been revealed experimentally [10].

The algorithm, AGGRESCAN, is based on an aggregation-propensity scale for natural amino acids derived from in vivo experiments and on the assumption that short and specific sequence stretches modulate protein aggregation [5]. This algorithm was originally developed for investigating diseases such as Alzheimer's and Parkinson's, and is not therefore specifically tested against amyloidosis. It is shown to identify a series of protein fragments involved in the aggregation of disease-related proteins and to predict the effect of genetic mutations on their deposition propensities [5].

Based on the accumulated amyloid data, it is widely accepted that protein aggregation results in beta-sheet-like assemblies that adopt either a variety of amorphous morphologies or ordered amyloid-like structures [16]. Amyloid beta-sheet aggregates have different chaperone affinities than the amorphous beta-sheet aggregates and therefore accumulate in different cellular locations and are degraded by different mechanisms. Waltz is a web-based tool that uses a position-specific scoring matrix to predict sequences that are prone to forming amyloid fibrils [16].

The hidden β-strand propensity of an amino acid sequence can be quantitatively determined by analyzing sequence-structure relationships in terms of tertiary contact. This is achieved by using the secondary structure preferences of template sequences of known secondary structure found in regions of high tertiary contact [26, 14]. The web-based tool, NetCSSP, uses a computational algorithm to detect hidden non-native sequence propensity for amyloid fibril formation and outputs a quantitative predictive value [26, 14].

A predictive algorithm called RFAmyloid that uses random forest to identify amyloid forming amino acid sequences was proposed by Mengting Niu [18]. The algorithm uses SVMProt 188-D feature extraction method based on protein composition and physicochemical properties and pse-in-one feature extraction method based on amino acid composition, autocorrelation pseudo acid composition, profile-based features and predicted structures features. The study includes experimental results on amyloid data and claims an accuracy rate of 89.19.

A machine learning method called V_LAmy-Pred explores different features that can be extracted from the Variable (V) region of immunoglobulin light chain sequences and their impact on AL amyloidosis. It considers the hydrophobicity of the complementary determining region (CDR), presence of gatekeeper residues in the FR's (framework regions) and disorderdness of the V_L region. Making use of Shannon entropy, this work establishes that V_L regions of κ light chains have lower aggregation propensity but greater sequence conservation among amyloidogenic sequences than in non-amyloidogenic sequences. On the other hand, V_L regions of λ regions have higher aggregation propensity but similar levels of sequence conservation between amyloidogenic sequences as opposed to non-amyloidogenic sequences [21]. Using these features, V_LAmy-Pred obtains 83% AUC on unseen data.

A recent method called LICTOR (λ-**LI**ght-**C**hain **TO**xicity predicto**R**) uses machine learning to predict toxicity of immunoglobulin light chain sequences for AL amyloidosis. In an initial exploration, LICTOR uses the Fisher exact test to assess frequencies of mutations at residue positions, numbered using the Kabat-Chothia scheme, to establish that there is significant difference

($p < 0.05$) between AL and healthy light chain sequences, and hence that somatic mutations are critical in prediction of light chain toxicity. LICTOR makes use of various combinations of features extracted from the ALBase dataset in order to make the prediction. The first set of features, AMP (amino acid at each mutated position), is generated by extracting the presence or absence of germline gene mutations from each residue in the Variable and Joining regions of the light chain sequences. The second set of features, MAP (monomeric amino acid pairs) identifies whether there are mutations in residues that are in close contact (< 7.5Å) in the same monomeric 3D structure. Finally, the third set of features, DAP (dimeric amino acid pairs) identifies whether or not there are mutations in residues that are close in contact in 3D space but from different chains. This work explores a number of machine learning algorithms as well as different combinations of the above three sets of features, and concludes that Random Forest using AMP, MAP and DAP has the best performance in prediction of toxicity of Immunoglobulin LC sequences. Using Random Forest and this set of features, LICTOR achieves a 0.87 AUC score (area under the receiver operating characteristic curve), specificity of 0.82 and sensitivity of 0.76 [11]. Further, LICTOR concludes that, based on their experiments, J region has no impact on prediction of AL amyloidosis. Experimental validation of the results is conducted first *in silico* and next in a *C. elegans* model *in vivo*, by reverting two somatic mutations identified by LICTOR as contributing to AL amyloidosis [11].

The novelty of our work lies in that we are able to predict whether a given LC amino acid sequence will result in misfolded proteins that eventually lead to AL amyloidosis using deep learning. We have already seen some progress by using the combination of the germline mutations in the amino acid LC sequences and their corresponding secondary structures as input to some machine learning models. Deep learning has been shown to have the capability to learn many hidden features from rich, high dimensional data and has proven effective in complex image recognition tasks such as interpretation of satellite images and self-driving cars. As AlphaFold and some other recent techniques have had tremendous success in protein structure prediction [12, 24], we investigate whether similar techniques could be used to predict the propensity a given amino acid LC sequence has to forming amyloid fibrils in AL amyloidosis.

1.3 Organization of the Thesis

Since the computational complexity of predicting AL amyloidosis is directly related to how the light chain (LC) amino acid sequences are synthesized, we first discuss the relevant biological processes that are responsible for the antibody diversity or specifically the diversity of the LC sequences in Chapter 2. Next, in Chapter 3, we provide the technical details of our models in terms of the structure of the neural networks and the way the training and testing inputs get passed to our models. We also describe the structure of the input and the lay out the various versions of DALAD that we use in this work in the same chapter. We then move on to Chapter 4, where we elaborately state the entire process of hyperparameter selection, followed by the experimental setup and the compositions of datasets for each version, and then the complete list of experimental results for different types of testing for each version. We end the chapter with a comprehensive discussion on our findings. Finally, we provide our concluding remarks in Chapter 5, where we summarize our methods, results, and accomplishments in this work, and finish this thesis by discussing the limitations of our work, but more importantly, by laying out a few promising future directions that could assist with designing better predictive models for AL amyloidosis.

Chapter 2

The Biology of Light Chain Diversity

The light chains (LC) are an important component of antibodies or immunoglobulins, which provide the crucial defense against millions of different pathogens. An antibody molecule, shown in Figure 2.1, consists of two light chains and two heavy chains forming a "Y" shaped structure. Together, both light and heavy chains are responsible for providing an immunoglobulin repertoire of more than 10^{11} unique antibodies that constitute an impressive adaptive immune system. Since only the light chains are involved in the pathogenesis of AL amyloidosis, we will ignore the heavy chains and solely focus on the light chains in this discussion.

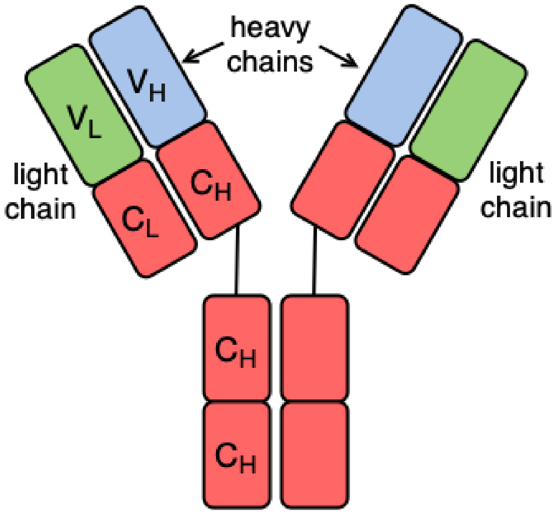


Figure 2.1: The structure of an immunoglobulin molecule.

2.1 The Synthesis of Light Chains

Since the light chains are polypeptide chains or proteins, according to the central dogma of biology, their synthesis is governed by the transcription of their coding DNA and the translation of the resulting mRNA. Therefore the understanding of the transcription of the coding DNA and the translation of the resulting mRNA is essential to the understanding of how and why AL amyloidosis occurs. For nearly all the genes found in humans, a completely ordered DNA sequences is present in the germline. However, in contrast, the encoding DNA sequence for the variable region of a light (or a heavy) chain polypeptide comes from two separate DNA segments, namely the V gene segment and the J gene segment, which are later spliced together to form the final encoding DNA sequence during the B cell maturation in the bone marrow. The V gene segment of the V region encodes approximately the first 95 to 101 amino acids of the light chain and the J gene segment of the V region encodes the remaining approximately 13 amino acids of the light chain. The V gene segments and the J gene segments are initially separated in the germline genome. The splicing of the V and J gene segments in the precursor B cells, known as the somatic V-J recombination process, is shown in Figure 2.2.

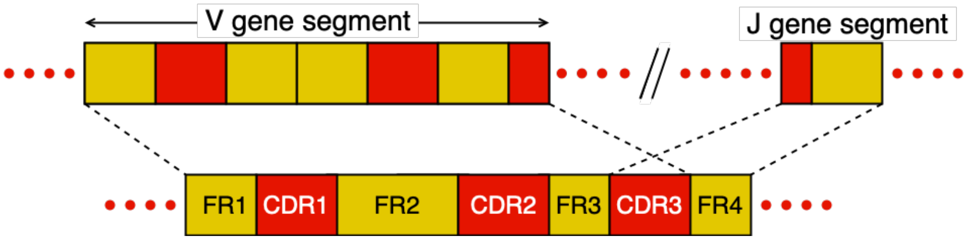


Figure 2.2: The somatic V-J recombination

There are two distinct types of light chains found in humans, namely λ and κ, which are encoded in chromosomes 22 and 2 respectively. Gene cloning and genomic sequencing have identified that there are multiple distinct gene segments for both V and J gene segments in the germline DNA. The number of V and J gene segments and the genes for the constant region that have been identified as of July, 2022 for both κ and λ light chains are listed in Table 2.1 [15].

The κ light chains are produced from nearly 39 functional V gene segments, a cluster of five J gene segments as well as a single C gene, all are located on chromosome 2. Specifically, the V and

J gene segments are arranged on chromosome 2 such that the entire cluster of V gene segments is followed by a cluster of J gene segments, which in turn is followed by a single C gene, which is responsible for the synthesis of the C region of the light chain.

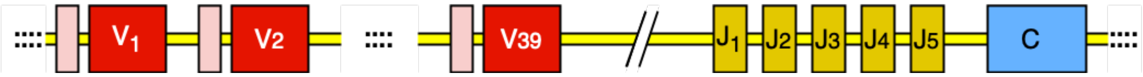


Figure 2.3: The κ genes on chromosome 2

The λ light chains are produced from nearly 32 functional V gene segments, five functional J gene segments and five C genes located on chromosome 22. Specifically, the V and J gene segments are arranged on chromosome 22 such that the entire cluster of V gene segments is followed by five sets of J gene segments. Each of the five J gene segments in turn is connected to a single C gene, which is responsible for the synthesis of the C region of the light chain.

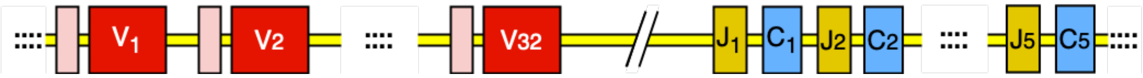


Figure 2.4: The λ genes on chromosome 22

Note that in a healthy individual, the ratio of the whole and intact κ to λ light chains is roughly 2 : 1, and the ratio of the free κ to λ light chains is around 1 : 1.5, or something that varies between 0.26 and 1.65. That said, among the polyclonal free light chains, usually only one clone becomes abnormal and causes AL amyloidosis within a patient. The abnormal AL clone can be either κ or λ. In patients with amyloidosis, however, the AL type has a κ to λ ratio of about 1 : 3.

2.2 Origin of Light Chain Diversity

The diversity of the human light chains is due to the following three main combinatorial processes.

1. The random selection and recombination of the V and the J gene segments.

2. The junctional diversity.

3. The somatic hypermutations.

Segment	**No of κ genes**	**No of λ genes**
V gene segment	39	32
J gene segment	5	5
C gene	1	5

Table 2.1: The count of various gene segments contributing to an LC sequence [15].

The first two of these processes take place in the precursor B cells and the last one takes place only in B cells in the already rearranged V-region genes due to the triggering of an immune response upon encountering an antigen.

2.2.1 Random Selection and Recombination of V and J Gene Segments

A germline LC sequence consists of one V region and one C region, where the V region is coded together by a single V gene segment and a single J gene segment, while the C region is coded by a C gene. Since there are multiple distinct gene segments for each type, and only one gene segment of each type is required for the complete assembly of the light chain, a random selection of a V and a J gene segment takes place in the precursor B cells. There are 32 distinct V gene segments and five distinct J gene segments on chromosome 22. The 32 V gene segments are clustered next to each other, while the five J gene segments are separated from each other by a C gene as shown in Figure 2.4. Therefore, a λ LC sequence can be generated in $32 \times 5 = 160$ distinct ways by the random selection of one V gene segment from out of the 32 possible ones, and one J gene segment from out of the 5 possible ones. In an analogous manner, we find that a κ LC sequence can be generated in $39 \times 5 = 195$ distinct ways. The first source of the light chain diversity is due to the recombination that takes place at the joining of the V and J gene segments. Specifically, the third hypervariable region or the CDR3 region is generated by the recombination of the V and J gene segments. The genome responsible for the generation of the first two hypervariable regions, namely CDR1 and CDR2, are found entirely within the V gene segment and therefore do not participate in the V-J recombination.

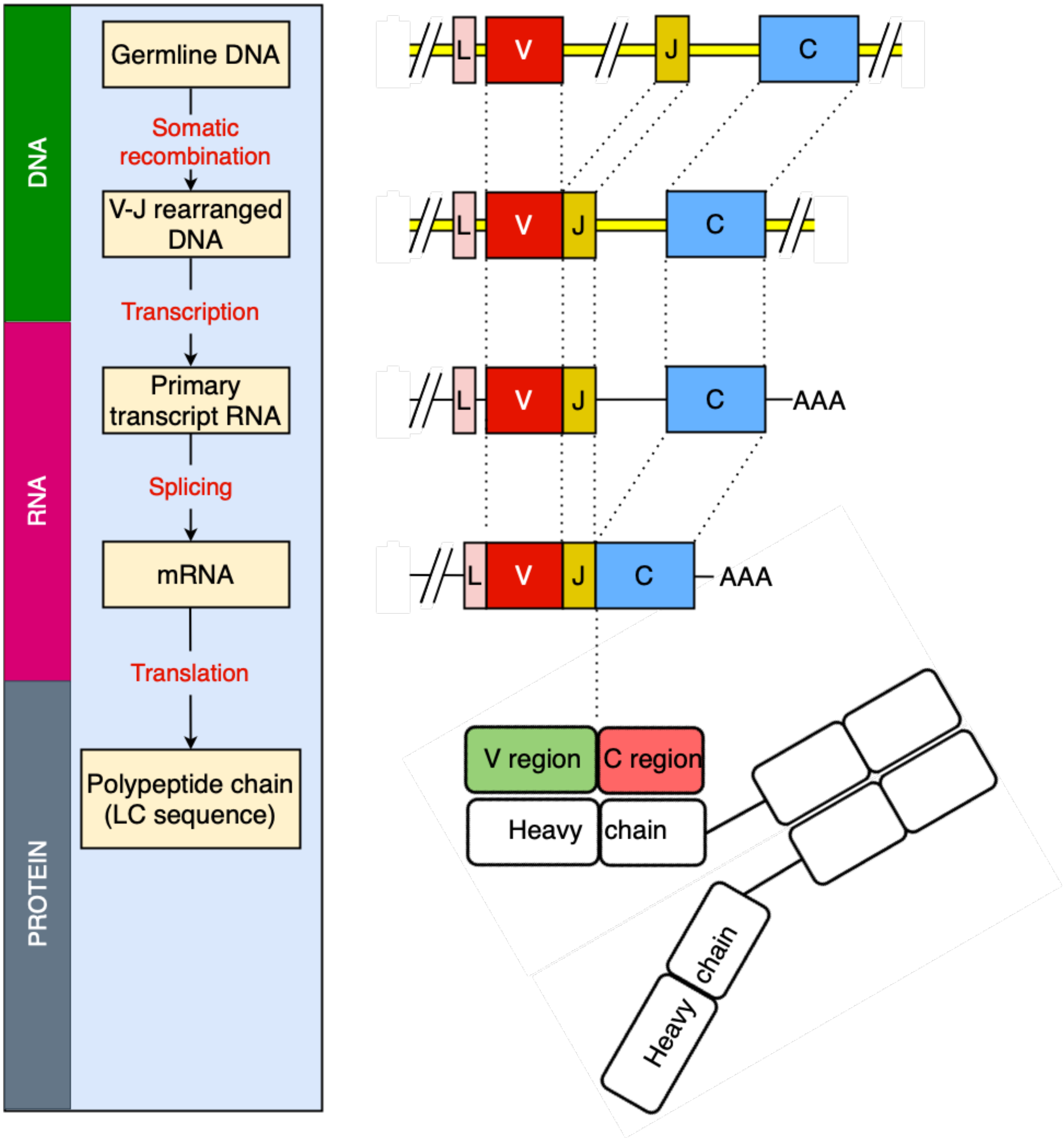


Figure 2.5: The synthesis of an LC chain from DNA. Germline DNA contains multiple copies of the V and J segment genes. Somatic recombination of the V and J gene segments produces a V-J rearranged DNA. The transcription of the V-J rearranged DNA results in a primary transcript RNA, which will have a poly-A tail added for stability. Next, during splicing, introns are removed and an *m*RNA will be produced. The translation of this *m*RNA results in an LC sequence containing a V and a C region, which in turn pairs with a heavy chain.

2.2.2 Junctional Diversity

The diversity of the CDR3 region is further increased by addition and deletion of P (or palindromic) and N (or non-template-encoded) nucleotides during the recombination process [17]. This can be observed in the LC amino acid variability plot shown in Figure 2.6. This diversity is also known as the junctional diversity as the process takes place at the joining of the V and J gene segments.

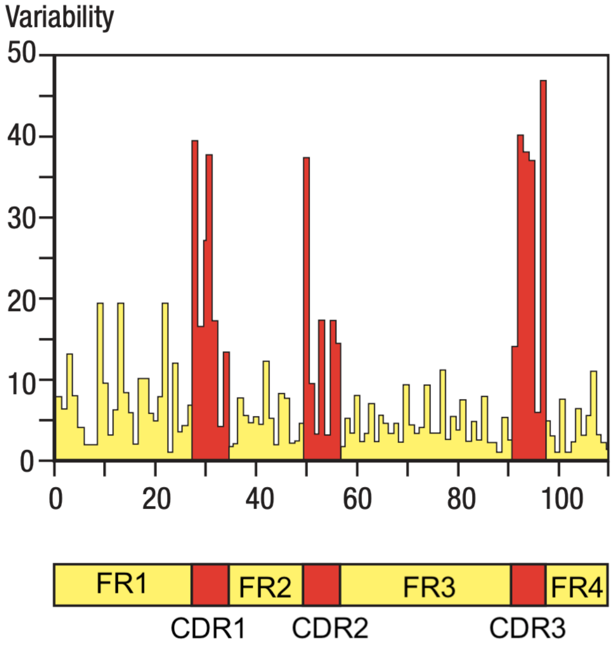


Figure 2.6: The LC amino acid variability [17]

2.2.3 Somatic Hypermutations

The third source of the light chain diversity is due to the single-point mutations that occur in mature B cells, which take place only on the re-arranged DNA that encodes the V regions. Known as the *somatic hypermutations* – because they only occur on the somatic B-lymphocytes as opposed to the meiotic recombination that occurs during the gametogenesis – they generally provide diversity that are selected for improved antigen binding. The somatic mutations, which are much needed for increasing the antibody diversity and, therefore, generally lead to healthy light chains, are suspected to be the likely culprits causing fibril formation in AL amyloidosis.

Chapter 3

DALAD

The medical community so far has not been able to make any inroads into the diagnosis, the treatment, or the management of AL amyloidosis because of the well-known difficulty in diagnosing the disease from the signs and symptoms alone as they are often mistaken for common diseases. Therefore, we wish to explore developing new diagnostic techniques that are not based on the signs and symptoms of AL amyloidosis, but are based on the underlying molecular mechanisms of amyloidogenic clone that could predict the disease well before it sets its course on the body.

In this work, we present a (convolutional) neural network based deep learning approach, called "**D**etector for **AL A**myloidosis via **D**eep Learning" or "DALAD", for predicting whether a patient has AL amyloidosis or not. The theoretical foundation for our approach is based on the biology of light chain synthesis that we discussed in the introduction. Specifically, our work is based on the hypothesis that aberrations in one or a combination of the three biological processes responsible for the diversity of the light chains is the causative factor for the formation of amyloid fibrils. How each of the three biological processes can lead to AL amyloidosis is summarized in Table 3.1.

Biological Process	**How it Causes AL amyloidosis**
Random selection of V and J segments	Incompatible V and J segments
V-J recombination	Amylodogenic prone V-J recombination
Single point mutations	Amylodogenic prone mutations

Table 3.1: How natural biological processes lead to AL amyloidosis [15].

3.1 Overview of our Model

DALAD predicts the propensity of a given LC sequence leading to AL amyloidosis by discovering anomalies in each of the above three biological processes. Unlike the previous prediction methods discussed in the literature such as TANGO [9], PASTA [23], AGGRESCAN [5], WALTZ [16], RFAmyloid [18] and LICTOR [11], which solely rely on the V region of the LC sequences, DALAD uses both the V and J regions of the LC sequences for training as well as prediction. Our hypothesis, supported by the experimental results presented in this paper, is based on the fact that each of the three biological processes, whose aberrations contribute to AL amyloidosis (indicated in Table 3.1), involves not only the V region, but also the J region in one way or another.

In addition to using both the V and J regions as its input, DALAD also uses a custom germline database, which combines both the V and J regions. For *Homo sapiens*, the total number of λ and κ genes and therefore the total number of germline sequences representing the possible V regions is, $32 + 39 = 71$, according to the current IMGT database [15]. When we consider both the V and the J regions and the V-J recombination, the total number of possible germline sequences increases by five-fold to 355 distinct LC sequences. Therefore, the custom germline database used by DALAD includes a total of 355 distinct sequences as opposed to the 71 sequences that were generally considered in the prior research work in this area.

Finally, as illustrated in Figure 3.1, DALAD employs a 1-dimensional convolutional neural network learning model (or "1-D CNN"), which considers one feature to be a location in the amino acid sequence that contains the information of the germline sequence for that location and that of the LC sequence for that location. So, each feature essentially contains the information about the mutation at that location, and the CNN uses that feature of two pieces of information as a whole. In our model, after the convolutional module, lie the hidden layers of the neural network, followed by the output layer. The additional details about the model's architecture are provided in Figure 3.2. DALAD has different versions, as we will outline later, which use this entire structure in different ways. One way, in particular, is having two separate sub-models like the one shown in Figure 3.1 – one each for λ and κ sequences – which is one of our novel ideas behind creating our classification models that work on both types of sequences that we focus on in this manuscript.

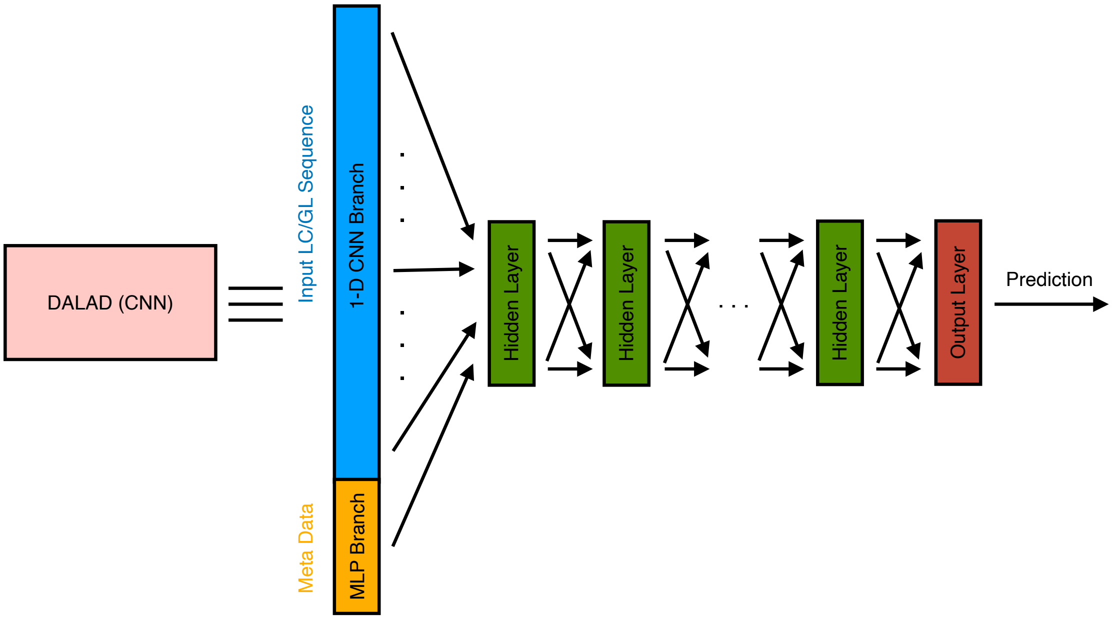


Figure 3.1: Structure of DALAD's (convolutional) deep neural network.

3.1.1 Comparison with State-of-the-Art

For comparison purposes, the performance of our model was benchmarked against the performance of LICTOR [11], a machine learning prediction model considered to be the current state-of-the-art in the literature for predicting AL amyloidosis. LICTOR, using the Random Forest algorithm, achieves a specificity and a sensitivity of 0.82 and 0.76 respectively with an AUC score of 0.87. LICTOR's prediction model is based on the prevailing hypothesis that somatic mutations in the light chains increase the likelihood of fibril formation, leading to AL amyloidosis [11, 2, 20, 6], and, therefore, it uses the differences between the light chain and the germ line sequences as the primary input in their prediction model. There are two major differences between LICTOR and our method.

1. The first is that LICTOR restricts its input to only the λ LC sequences, whereas DALAD considers both the λ and κ LC sequences. The rationale LICTOR for considering only the λ LC sequences is that the λ LC sequences are more prevalent in AL amyloidosis patients

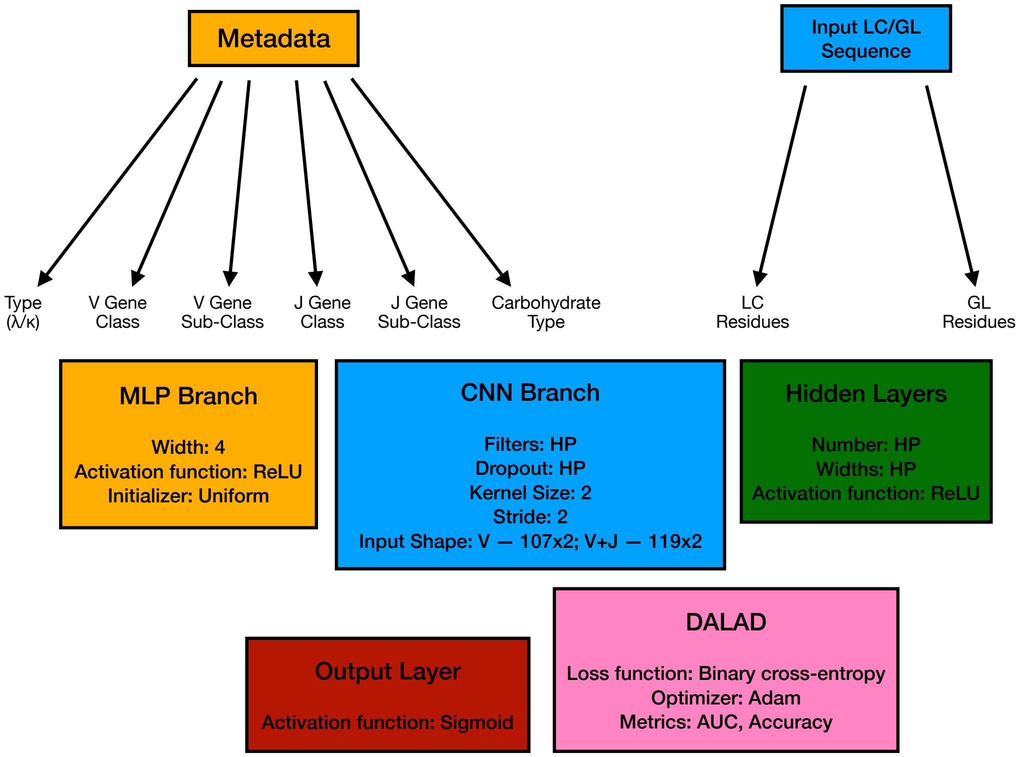


Figure 3.2: The low-level details of DALAD's architecture are stated here. "HP" indicates that those quantities are hyperparameters, and their selection is described in detail in Section 4.1. We used `Google Colab` to train and test our models.

than the κ LC sequences compared to that of healthy individuals, which is based on studies that show that the ratio of $\lambda : \kappa$ in patients with AL amyloidosis is $3 : 1$, while the ratio of $\lambda : \kappa$ in healthy individuals is $1 : 2$ [7]. However, given that one in every four AL amyloidosis patients has κ LC sequences, excluding the κ sequences from analysis will lead to reduced prediction accuracy on random test sequences. Consequently, LICTOR's model, trained only on λ LC sequences, lacks the ability to predict the toxicity of a κ LC sequence, which will invariably result in reduced prediction accuracy when used as a prediction tool in clinical settings. Hence the ability of DALAD to predict the propensity of developing AL amyloidosis using both the λ and κ sequences markedly distinguishes itself from LICTOR.

2. The second difference is that LICTOR restricts their analysis to the V region of the light chains, while our method considers both the V and J regions. We hypothesize that the random selection of V and J segments, random V-J recombination as well as any mutations occurring on the J region to influence the fibril formation as the CDR3 region is essentially formed as a result of a randomized V-J recombination. If the somatic mutations on the V region has statistical correlation to AL amyloidosis, as shown by the authors of LICTOR using Fischer's exact test [11], then based on the reason stated in the previous sentence, it is likely that the somatic mutations on the J region would have similar influence on the disease.

3.2 Input Dataset

There are two key features of DALAD that distinguish it from all of the previous machine learning approaches for predicting AL amyloidosis to-date. (1) Unlike most other works (including LICTOR, the state-of-the-art), our model can process both λ and κ sequences. (2) One set of our versions of DALAD (i.e., $\mathsf{DALAD}_{\mathrm{VJ}}$) uses both the V and the J gene segments of the LC sequences, including both the λ and κ varieties.

Our primary source of input data comes from AL Base [4], which is a curated database of light chain sequences available from the Amyloidosis Center at Boston University Medical Center and the Department of Medicine. The dataset is a collection of Ig LC sequences from patients with AL amyloidosis that was collected with the goal of enabling researchers to study their differences as well as their predicted protein sequences between those LC sequences from non-amyloidogenic patients. The AL Base dataset contains 4364 LC nucleotide and amino acid sequences, of which 808 encode monoclonal proteins that were reported to form fibrillar deposits in patients with AL amyloidosis. In addition, the dataset contains over 248 control LC sequences from patients with other plasma cell disorders without known amyloidosis, and 295 control LC sequences from healthy subjects. The last two types of sequences can be treated as noise in the data.

3.2.1 Structuring the Input

As mentioned above, DALAD uses the J regions of both the λ and κ light chains in addition to the V regions. Therefore, unlike the previous works, such as LICTOR, we needed to construct an expanded germline database, which includes not only the V region sequences, but also the J region sequences. Our hypothesis is that using an alignment tool, such as BLAST [1], against such an expanded database against light chain sequences containing both the V and J regions will defintely identify a larger set of responsible mutations. We also hypothesize that certain mutations in the V regions are likely to culminate in AL amyloidosis only in the presence of some mutations in the J regions. In other words, the fibril formation is due to the presence of a set of correlated mutations in both the V and J regions, and not just due to the mutations in the V region alone.

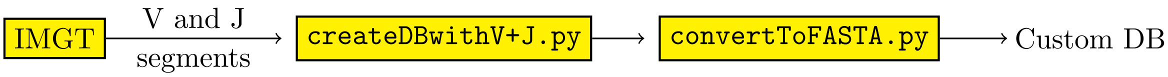
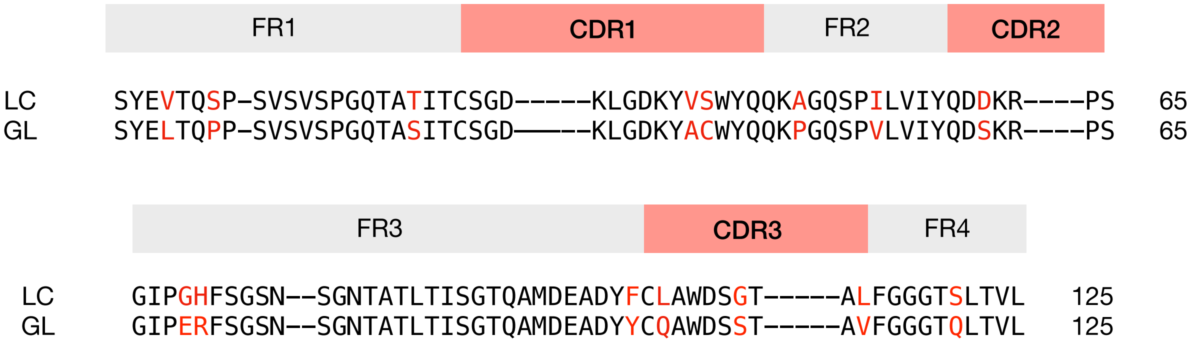


Figure 3.3: Creating the expanded custom germline database from IMGT

As a result of the availability of the additional features corresponding to the J regions, DALAD is able to use a custom 355-sequence germline database that contains all the possible combinations of the V and J regions for both the λ and the κ sequences from the IMGT repository in order to identify the potential mutations that can likely lead to AL amyloidosis. The 355 sequence custom germline database was constructed as follows. From the IMGT repository, we downloaded both the λ and κ germline sequences corresponding to the V gene segment. As of July, 2022, the IMGT repository listed 32 λ genes and 39 κ genes for the V gene segment. Further, from the IMGT repository, we also downloaded both the λ and κ germline sequences corresponding to the J gene segment. As of July, 2022, the IMGT repository listed five λ genes and five κ genes for the J gene segment. This allowed us to create our expanded custom germline database containing $32 \times 5 = 160$ germline sequences containing both the V and J regions of the λ type, and $39 \times 5 = 195$ germline sequences containing both the V and the three regions of the κ type, for a total of 355 germline sequences.

Then the light chain sequences were aligned to the corresponding germline sequence using the expanded custom germline database consisting of 355 sequences generated in the workflow shown in Figure 3.3. The custom germline database (called `GL_VJ_Lambda_Kappa.fasta`) was uploaded to the BLAST website at `https://blast.ncbi.nlm.nih.gov/Blast.cgi`. We numbered the light chain sequences using the Kabat-Chothia scheme and ran BLAST against the custom database. Using the results, we updated the input LC sequences to differentiate between the mutated positions and the non-mutated positions.



```
LC     SYEVTQSP-SVSVSPGQTATITCSGD-----KLGDKYVSWYQQKAGQSPILVIYQDDKR----PS   65
GL     SYELTQPP-SVSVSPGQTASITCSGD-----KLGDKYACWYQQKPGQSPVLVIYQDSKR----PS   65

LC     GIPGHFSGSN--SGNTATLTISGTQAMDEADYFCLAWDSGT-----ALFGGGTSLTVL   125
GL     GIPERFSGSN--SGNTATLTISGTQAMDEADYYCQAWDSST-----AVFGGGTQLTVL   125
```

Figure 3.4: We use IgBLAST to align the LC sequences from AL Base with the GL sequences from IMGT. This is an example of what the alignment would look like.

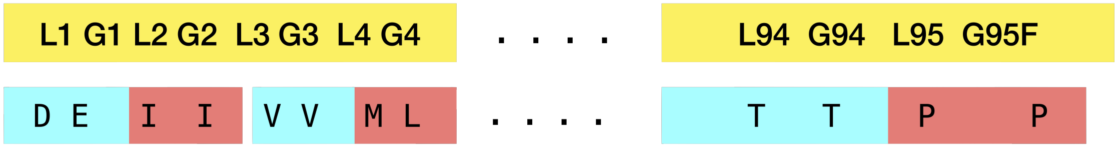


Figure 3.5: Aligned input sequences containing the V region. Each input sequence contains consecutive pairs of residues, side-by-side, such that each pair represents the LC residue and the corresponding GL residue at that location in the sequence (e.g., the first pair, labelled "L1/G1", shows the LC residue and the GL residue, respectively, at the first position). This construction was made possible due to the alignment imposed by IgBLAST, as shown in Figure 3.4. The CNN's in all the DALAD$_{V}$ and the DALAD$_{VJ}$ versions consider each of these pairs as a single feature, and end up using information from 107 such features.

We elaborate on the above process further using an example. The first line in Figure 3.4, titled

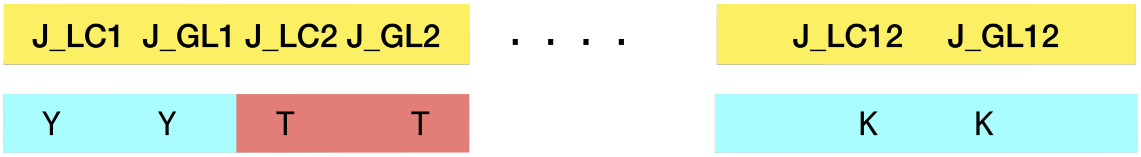


Figure 3.6: Aligned input sequences containing the J region. The idea behind this representation is the same as in Figure 3.5. There are 12 such pairs or features, which are utilized by the CNN's in all the DALAD$_{\mathrm{VJ}}$ versions.

"LC", contains the light chain sequence from ALBase, and the second line titled "GL" contains the residues in the germline sequence found in IMGT using the IgBLAST alignment tool [25] and ANARCI (**A**ntigen receptor **N**umbering **A**nd **R**eceptor **C**lassificat**I**on – a tool for numbering amino-acid sequences of antibody and T-cell receptor variable domains) [8] for annotating the alignment. Wherever there is a mutation in the light chain sequence from the database, we capture that information.

We hypothesized that both the LC and GL residues could be of use to our neural network model, as it is possible that the nature of the mutation (e.g. a mutation from a hydrophilic residue to a hydrophobic one) could contribute to a light chain sequence being amyloidogenic. So, we augmented our input to show both the light chain and germline residues. Figure 3.5 shows the structure of the sequence data for the V region that is used in our 1-D CNN. Figure 3.6 shows what the input looks like for the J region. The V and J region sequence data are concatenated for our V-J models, and the GL and the LC residues together for each locations in the sequences are parsed as a two dimensional feature by our CNN.

Remark 3.2.1. For the purpose of both training and hyperparameter selection, we balanced the datasets that we used, that is, we ensured that the ratio of the number of positives and negatives from each type of sequence was maintained within $[0.74, 0.88]$, similar to what LICTOR did for their purpose. We did this so that our trained model would not get biased towards any one particular category, especially since the number of negatives for each type of sequence was significantly more than the number of positives for that sequence in the main dataset. In real life, there are way more negative cases than the positive ones, so by balancing our dataset, we are admittedly changing the distribution of the data, too. That said, in order to have any sort of non-trivial accuracy on the positive sequences, we had to balance the dataset so as to not blur out the signal coming from those positive sequences, especially given that the overall size of the dataset we used (AL Base) was very small (close to 4000 data points).

3.2.2 Motivation for Studying κ Sequences

While our experiments with DALAD on λ-only datasets were used as a benchmark for our model against the published results of LICTOR, which uses only the λ sequences, we claim that learning to classify the κ sequences correctly is also very useful for demonstrating the power of DALAD in an actual clinical setting. This is because any random sample of individuals in a clinical setting will contain a significantly larger percentage of κ sequences than the λ sequences, and, thus, we cannot afford to ignore the κ sequences.

Type of Sequence	Healthy	Diseased	Total
λ	992	525	1517
κ	1757	166	1923

Table 3.2: Composition of the main dataset (AL Base) that we used.

In Table 3.2, we see that a major fraction of the dataset is composed of κ sequences. Given that issues with κ sequences also lead to AL amyloidosis, it is very important to be able to make accurate predictions on them, otherwise we would (1) ignore a large portion of the patients, who might have otherwise benefited from computational models, and (2) waste a big section of our data, hence, lose our prospects of achieving higher accuracy.

To provide another perspective, if we are unable to make correct predictions on the κ sequences with high confidence (say, accurate less than 50% of the times), then even if we are 90% accurate on all the λ sequences, then our overall accuracy comes out to be

$$\frac{(0.90 \times 1517) + (0.50 \times 1923)}{1517 + 1923} \approx 68\%.$$

This means that in order to get more representative accuracy numbers, we would want to be able to classify on both types of sequences accurately.

3.3 Versions of DALAD

We developed various versions of DALAD for different types of datasets, incorporating different kinds of features and types of sequences. The two types based on the types of features are those that use the information or features corresponding to the J regions of the light chain sequences, and those model that do not. Within those types of models, we had versions that worked with λ-only datasets, κ-only datasets, or with datasets containing a mixture of λ and κ sequences. We identify each version of DALAD through the use of superscripts and subscripts for its target datasets and to denote whether it used the features corresponding to the J regions, respectively. In the next chapter though, we will describe how these models were trained, and how their hyperparameters were selected.

3.3.1 $\mathsf{DALAD}_{\mathrm{V}}$ Models

The models described in this section *do not* use any features or information coming from the J regions.

$\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$. This version of $\mathsf{DALAD}_{\mathrm{V}}$ is simply trained and tested on datasets that are solely composed of λ sequences. We use this model to compare our performance against that of LICTOR's best model.

$\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$. This version is trained and tested on datasets that are only composed of κ sequences. This is a major contribution of our work, since this now enables us to classify both types of light chain sequences.

$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$. This version is trained and tested on datasets that are composed of both λ and κ sequences. This comprises a single model that gets trained on a mix of both types of sequences and gets tested on a mixture of the two types of sequences, as well.

$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$. Just like $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, this "separating" model also gets trained and tested on a mix of both λ and κ sequences, but in a different way. It is composed of two sub-models – one that is simply trained on λ sequences, and the other one that is trained just on κ sequences. When a test sequence arrives, depending on whether it is a λ or a κ sequence, it gets passed on to the appropriate sub-model to get classified.

3.3.2 $\mathsf{DALAD}_{\mathrm{VJ}}$ Models

Unlike in the case of the $\mathsf{DALAD}_{\mathrm{V}}$ models, the models in this section *do* incorporate the features corresponding to the J regions. $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$, $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$, $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$, and $\mathsf{DALAD}^{\mathrm{Sep}}_{\mathrm{VJ}}$ are the $\mathsf{DALAD}_{\mathrm{VJ}}$ analogues of $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$, $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$, $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$, and $\mathsf{DALAD}^{\mathrm{Sep}}_{\mathrm{V}}$, respectively.

3.3.3 Understanding the Sep Models

DALAD, in a nutshell, is a (convolutional) deep learning neural network. The Sep model architecture, depicted in Figure 3.7 consists of one preprocessing units and two individual deep learning units, one for the λ LC sequences and the other for the κ LC sequences.

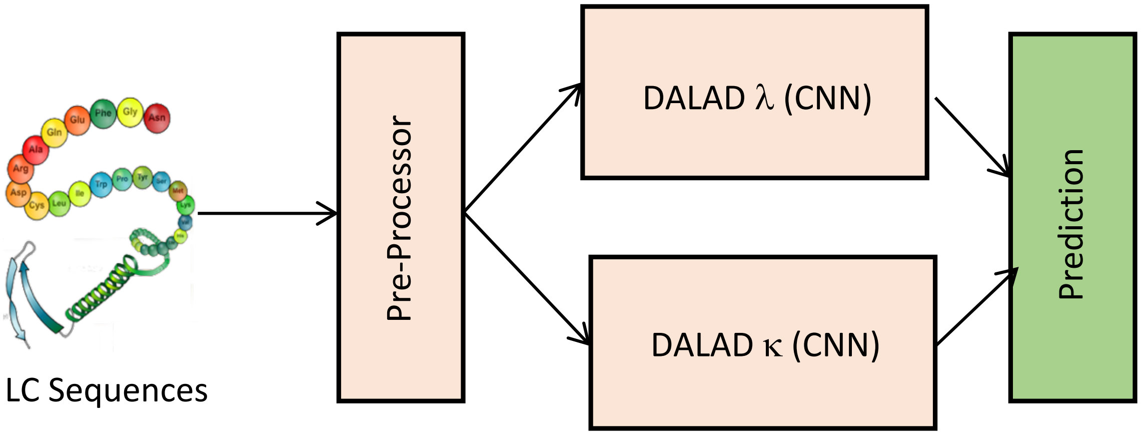


Figure 3.7: Sep model overview.

The preprocessing unit separates the λ sequences from the κ sequences and sends them to the appropriate λ or κ models for training. An identical process takes place during the prediction on random LC sequences on a previously trained Sep model. The rationale behind employing two separate neural nets for the λ and the κ sequences is that their germline sequences have distinct constituent amino acids, and, thus, the mutations occurring on a λ LC sequence cannot be aligned against a κ germline sequence, and vice versa. Note that for a fixed version, say the $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ version, the number of features for both its λ and its κ modules would be the same. We will compare the performance of the Sep models with that of the Joint models later in Chapter 4.

Chapter 4

Experiments and Discussion

In this chapter, we describe in detail the processes of hyperparameter selection, model training, and testing, along with the results of all our experiments. As mentioned in the previous chapter, we evaluate our deep learning approach on multiple kinds of datasets, and define our models using different sets of features. To recapitulate, $\mathsf{DALAD}_{\mathrm{V}}$ models only use the features from the V regions, whereas the $\mathsf{DALAD}_{\mathrm{VJ}}$ models use the features from both the V and the J regions of the sequences. Each type of model has separate versions designed for three settings: (1) for λ-only datasets; (2) for κ-only datasets; and (3) for datasets containing a mix of both λ and κ sequences (in this setting, we present two versions – Joint and Sep – as defined in Section 3.3). At the end of this chapter, we compare all the results for these versions, and provide an elaborate discussion on their consequences.

4.1 Hyperparameter Selection

For each of the aforementioned versions of DALAD, we select the set of the best hyperparameters via a search over a list of 216 candidates. Each of those candidates is defined by the number of filters (or "kernels") used in the convolutional layer, the dropout rate in the convolutional layer (a layer that nullifies the contribution of some features towards the next layer, and leaves the others unmodified), and the number and the widths of the hidden layers (if any) between the output and the convolutional layers.

General Methodology. The high-level idea was to run a set of 40 experiments each for $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, for every choice of the hyperparameters, and chose the best model in each case, which worked the best "on an average". The rationale was to choose the model that would reliably give us the best outcome consistently. For example, for $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, for a fixed set of hyperparameters, 40 experiments were performed, such that in each experiment, the model was completely reset, and different datasets was selected for the purpose of training and testing (we describe the sampling process below). Then the choice of hyperparameters, for which we got the best results on an average over the 40 runs, became our final choice for $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$. The same process was performed for the other versions of DALAD, as well.

Dataset Compositions. As mentioned above, we chose different datasets in each experimental run of all the versions of DALAD. We specify the sampling process for every version in the following. As mentioned earlier, just like in LICTOR, in each case, we balance the number of positives and negatives in every dataset to maintain approximately the same positives-to-negatives ratio in the range $[0.74, 0.88]$.

- For $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$, in each experiment, we randomly selected a set of 210 positive λ sequences and 240 negative λ sequences, and performed a $9:1$ split of each type, and merged to create datasets to train and test.

- For $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$, in each experiment, we randomly selected a set of 166 positive κ sequences and 225 negative κ sequences, and performed a $9:1$ split of each type, and merged to create datasets to train and test.

- For $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, in each experiment, we randomly selected a set of 83 positive κ sequences and 112 negative κ sequences, and performed a $9:1$ split of each type, and merged to create datasets to train and test, and did the same for a randomly chosen set of 140 positive λ sequences and 160 negative λ sequences. The training datasets from both λ and κ sequences were merged to create one large training set, and the same was done to create a large test dataset.

Note that we did not have to select the hyperparameters for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ or for $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ because they essentially rely on the accuracy of their respective λ and κ sub-models.

Hyperparameter Set. For each version of DALAD, we chose from the following set of hyperparameters. We chose the dropout value for the dropout layer inside our CNN module from the set $\mathcal{H}_D \coloneqq \{0.5, 0.7, 0.8, 1.0\}$ (the dropout value r indicates that a random r fraction of the features in the previous layer of the module would be used in the layers that lie ahead). The number of filters in that module were chosen from the set $\mathcal{H}_F \coloneqq \{32, 64\}$. Finally, the number and the widths of the hidden layers between the CNN module and the output layer were chosen from the set

$$\begin{aligned}\mathcal{H}_L \coloneqq &\{[], [64], [32], [16], [8], [64, 32], [64, 16], [64, 8],\\ &[32, 16], [32, 8], [64, 32, 16], [64, 32, 32], [64, 16, 16],\\ &[64, 32, 8], [64, 16, 8], [32, 16, 8], [32, 32, 16], [32, 32, 8],\\ &[32, 16, 16], [32, 16, 8], [16, 16, 8], [16, 8, 8], [8, 8, 8],\\ &[64, 32, 16, 8], [64, 32, 32, 16], [64, 32, 32, 8], [64, 32, 16, 16]\},\end{aligned}$$

where [] indicates that there are no hidden layers in the neural network, and for each $\ell \in \mathcal{H}_L$, such that $\ell \neq []$, the length of ℓ denotes the number of hidden layers in the network, $\ell[i]$ denotes the width of the i-th hidden layer in the network. For example, $[64, 32, 8]$ indicates that the width of the first hidden layer after the combined convolutional and multilayer perceptron branch is 64, and the following hidden layer has a width of 32, and the next (also the final layer before the output layer) layer has a width of 8. Therefore, our set of candidate hyperparameters was

$$\mathcal{H} \coloneqq \mathcal{H}_D \times \mathcal{H}_F \times \mathcal{H}_L.$$

As an example, a set of hyperparameters denoted by $(0.7, 32, [64, 32, 8])$ refers to a set of models, where the dropout rate in the convolutional module is 0.7, the number of filters in the convolutional module is 32, and the hidden layers between the first layer of our neural network (the combined multilayer perceptron and the convolutional module) and the output layer have width 64, 32, and

Version	Dropout	Number of Filters	Hidden Layers
$\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$	1.0	64	[32]
$\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$	0.5	32	[64, 32, 16]
$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$	0.5	32	[64, 32]
$\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$	0.7	64	[64]
$\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$	0.8	64	[64]
$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$	0.5	32	[64, 32]

Table 4.1: Hyperparameter choices for different versions of DALAD.

8, respectively. We refer the reader back to Figure 3.1 for a visualization of our model.

Best Hyperparameters. Table 4.1 summarises the final choice of hyperparameters for each of the aforementioned versions of DALAD.

4.2 Experimental Setup

Here, we describe our experimental setup in terms of the high-level ideas behind the process, the number of experiments performed after the hyperparameter selection for each version, the accuracy measures and the statistics used to anaylyze these experiments, and the dataset compositions for every set of experiments.

General Methodology. For each version of DALAD, after fixing its hyperparameters (as described in Table 4.1), we run 100 experiments with fresh choices of datasets each, and compare the aggregate statistics with those for the other DALAD versions. The purpose of looking at the aggregate statistics is just to see how the model would fare in general, in case it has to be retrained, and to get more confidence in its performance. In this set of 100 experiments, we also choose the best model (weights), and use that as a benchmark for our efficiency. That choice is made on the basis of the results for that model being above certain thresholds for different statistics. We describe the statistics briefly below. Finally, we test that model on the entire dataset that is relevant to that model (for example, the entire λ dataset for $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$).

Other Specifics. In each version of $\mathsf{DALAD}_{\mathrm{V}}$, the input layer consisted of 220 input nodes, while in each version of $\mathsf{DALAD}_{\mathrm{VJ}}$, the input layer was composed of 244 input nodes. In all versions of DALAD, the hidden layers (if any) used the ReLU activation function, while the output layer always used the Sigmoid activation function. For the purpose of compilation, we utilized the Adam optimizer under the Binary Cross Entropy loss function, and used AUC and Accuracy as the learning metrics for the process.

Accuracy Measures and Statistics Used. For each experimental run, we use the following accuracy measures for the performance of the model.

- Sensitivity or True Positive Rate (TPR). Let n_T be the number of positives, and m_T be the number of correctly classified positives. Then the TPR simply equals the fraction of the correctly classified positives, or $\frac{m_T}{n_T}$.

- Specificity or True Negative Rate (TNR). Let n_F be the number of negatives, and m_F be the number of correctly classified negatives. Then the TNR simply equals the fraction of the correctly classified negatives, or $\frac{m_F}{n_F}$.

- Accuracy (ACC). Let n_T be the number of positives, and m_T be the number of correctly classified positives. Let n_F be the number of negatives, and m_F be the number of correctly classified negatives. Then the ACC simply equals the fraction of the correctly classified examples, or $\frac{m_T+m_F}{n_T+n_F}$.

- AUC-ROC (or simply, AUC) score. Let FPR be the fraction of the negative samples that are incorrectly classified. Then the Receiver Operating Characteristic (or ROC) curve is a TPR-vs-FPR graph, and the corresponding AUC (Area Under the Curve) score is just the area under the ROC curve.

Once we have a sequence of each of the above accuracy measures (100 of each, that is) for each version, we use the following aggregate statistics to evaluate its general performance: mean, median, standard deviation, minimum, and maximum.

Dataset Compositions. As mentioned earlier (even for the process of hyperparameter selection), we chose different datasets in each of the 100 experimental runs of every version of DALAD. We specify the sampling process for each version in the following.

- For $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$, in each run, we chose all the 525 positive λ sequences in the dataset, along with a set of 600 randomly sampled negative λ sequences, and randomly split each of them in a $9:1$ ratio for the purpose of training and testing, respectively. In other words, our training dataset consisted of 90% of those 525 positive and 90% of those 600 negative λ sequences, and the remainder of the sequences formed our training dataset.

- For $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$, in each experiment, we chose all of the 166 positive κ sequences, and randomly selected a subset of 225 negative κ sequences, and performed the same kind of splitting as in the λ case for the purpose of training and testing.

- For $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, we performed the same kind of splitting as we had described in the case of hyperparameter selection, but this time, by choosing all of the 525 positive λ and 166 positive κ sequences, and random subsets of 600 negative λ and 225 negative κ sequences.

- For $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$, we performed the same sampling process as in the aforementioned cases of the Joint models.

We would like to remind the reader that in each of the above cases, we balance the number of positives and negatives in every dataset to maintain approximately the same positives-to-negatives ratio in the range $[0.74, 0.88]$. This helps achieving better accuracy guarantees while classifying the positive cases. Also, after we select the best model from each of the version, we test them on the entire dataset that is relevant to them, for example, we test the best trained $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$ models on the full λ dataset.

4.3 Experimental Results

In this section, we state the results of our experiments for all of our versions of DALAD. We start with the aggregate results for each version over 100 runs, and then move on to the results for the best trained model for each version.

4.3.1 Aggregate Results

As mentioned earlier, for each version of DALAD, we study the statistics (mean, median, standard deviation, minimum, and maximum) on all the accuracy measures (AUC score, Sensitivity, Specificity, and Accuracy) over 100 runs in order to be certain about the utility of our models.

4.3.1.1 $\mathsf{DALAD}_{\mathrm{V}}$ Models

We first provide our findings for all the $\mathsf{DALAD}_{\mathrm{V}}$ versions first. As a reminder, these models *do not* use any features corresponding to the J regions.

$\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$. We first state our results that we could use to directly compare with the performance of LICTOR, since their work focused only on the λ sequences. Our detailed aggregate statistics are stated in Table 4.2. The mean AUC score of over 87.2% already is on par with LICTOR's best model, and with a small standard deviation of 3.7%, we don't expect the performance of our model to vary too much if we are to retrain our model for some reason. The average sensitivity of over 77.7% beats that of LICTOR's best model, and so does our average specificity of over 82.8%. The average accuracy of over 80.4% indicates that in each run, our model tends to have both good sensitivity and good specificity simultaneously, which implies that if we are to select a trained model from those 100 runs, we are expected to have model that does both positive and negative classifications accurately. The low standard deviation numbers also confirm the high concentration of our accuracy measurements.

$\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$. Next, we discuss our results on κ datasets. Note that, however, we do not have a good baseline to compare them with, since our work appears to be the first one to have the ability to

Statistic	**Mean**	**Median**	**Std. Dev.**	**Minimum**	**Maximum**
AUC	0.8726	0.8753	0.0371	0.7726	0.9424
Sensitivity	0.7774	0.7924	0.0640	0.5094	0.8868
Specificity	0.8285	0.8333	0.0690	0.5833	0.9500
Accuracy	0.8045	0.8142	0.0422	0.7090	0.8938

Table 4.2: Results over 100 experiments for $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$.

Statistic	**Mean**	**Median**	**Std. Dev.**	**Minimum**	**Maximum**
AUC	0.9204	0.9284	0.0366	0.8286	0.9872
Sensitivity	0.8335	0.8235	0.0942	0.5294	1.0000
Specificity	0.8535	0.8696	0.0673	0.6522	1.0000
Accuracy	0.8450	0.8500	0.0505	0.7250	0.9750

Table 4.3: Results over 100 experiments for $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$.

tackle both λ and κ sequences efficiently and accurately. We refer the reader to Table 4.3 for the aggregate statistics over 100 runs for this version of $\mathsf{DALAD}_{\mathrm{V}}$. The aggregate numbers of $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$ appear to be significantly better than those of $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, which means that if we are to classify a κ sequence, we are more likely to be correct. The average AUC score is over 92%, while the average sensitivity and the average specificity are over 83.3% and 85.3%, respectively. As with $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, our average accuracy is very high (over 84.5% in this case), indicating that our $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$ models generally tend to classify both the negative and the positive sequences accurately. Our standard deviation numbers are also low again (except for that for sensitivity, where it is just marginally higher than that of $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$), implying high accuracy concentration of our models around the means, which are very high already.

$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$. We move on to the case, where our models may receive a dataset that contains a mix of both λ and κ sequences. We ask the reader to see Table 4.4 for details on the aggregate statistics of our overall accuracy numbers for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$. As described earlier in Section 3.3, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ does not train separately on the λ and the κ sequences, but simply consumes them together for the purpose of training. The averages of the overall AUC scores, the overall sensitivity, and the overall specificity are over 89%, 79.4%, and 82.4%, respectively, which are all very high, and much better than the respective numbers for LICTOR's best model. The high overall accuracy of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$

Test Sequence	Statistic	Mean	Median	Std. Dev.	Minimum	Maximum
$\lambda + \kappa$	AUC	0.8905	0.8948	0.0281	0.8277	0.9534
	Sensitivity	0.7944	0.7929	0.0606	0.6000	0.9429
	Specificity	0.8243	0.8193	0.0549	0.6265	0.9518
	Accuracy	0.8106	0.8137	0.0340	0.7190	0.9085
λ	AUC	0.8826	0.8807	0.0333	0.7808	0.9676
	Sensitivity	0.7885	0.7924	0.0667	0.5849	0.9623
	Specificity	0.8155	0.8167	0.0660	0.5500	0.9833
	Accuracy	0.8028	0.8053	0.0387	0.6814	0.9380
κ	AUC	0.9124	0.9143	0.0428	0.7724	0.9898
	Sensitivity	0.8129	0.8235	0.0984	0.5882	1.0000
	Specificity	0.8474	0.8697	0.0785	0.6522	1.0000
	Accuracy	0.8328	0.8250	0.0543	0.6750	0.9500

Table 4.4: Results over 100 experiments for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$.

(over 83.2%) indicates the consistency in classifying both the positive and the negative sequences correctly, that is, the model is very likely to be accurate on both types of sequences simultaneously. We also computed the average of all these statistics in the same runs for λ and κ sequences, as well, and here are the interesting observations.

- As we can see in Table 4.4, the statistics on the λ-only datasets are marginally better than those for $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$. The average AUC in the case of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ for λ sequences is over 88.2%, which is higher than that in the case of $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$ (over 87.2%). The same could be said about the average sensitivity for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ (over 78.8%), which is better than that for $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$ (over 77.7%). However, the average specificity in this case is lower than that for $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, resulting in similar accuracy scores for both $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$.

- From Table 4.4, we can see that the average AUC score of over 91.2% for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ for κ-only datasets is lower than that of $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$ (over 92%). The same is true for the average sensitivity, the average specificity, and the average accuracy, as well.

In other words, the statistics of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ on the λ sequences appear better than the ones for $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, showing that introducing a new type of sequences (κ sequences) in the training dataset does not hurt the general accuracy of the model on λ sequences, but improves it instead. On the other hand, the general accuracy of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ on κ-only datasets looks worse than if we were to

Statistic	Mean	Median	Std. Dev.	Minimum	Maximum
AUC	0.8846	0.8882	0.0251	0.8277	0.9616
Sensitivity	0.7887	0.8000	0.0635	0.6000	0.9286
Specificity	0.8328	0.8313	0.0534	0.6626	0.9398
Accuracy	0.8126	0.8137	0.0304	0.7320	0.9085

Table 4.5: Results over 100 experiments for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$.

just train on κ-only datasets, implying that the model gets misled on κ sequences when a new type of sequences (λ sequences) are introduced in the training dataset.

$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$. We now discuss the final version of $\mathsf{DALAD}_{\mathrm{V}}$ that also takes a mixture of λ and κ sequences in both training and testing datasets, but trains a separate model on each type of sequence, and uses that model to classify the respective type of sequence. Table 4.5 contains the detailed statistics for all the accuracy measures. We compare the performance of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ with that of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$. First, in terms of the average of the overall AUC scores, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ (over 88.4%) gets marginally beaten by $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ (over 89%). The average overall sensitivity of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ (over 78.8%) is also slightly lower than that of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ (over 79.4%). The average overall specificity of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ (over 83.2%), however, is better than that for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ (over 82.4%). Finally, both $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ and $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ have similar average overall accuracy scores. Since $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ contains two models (one each for λ and κ sequences), we do not compare its accuracy on the individual types of sequences again, since the discussion about $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ covers that comprehensively. That said, we would like to remark that the overall statistics of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ may appear marginally worse than those of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ because $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ does slightly better on λ-only datasets than $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ (but slightly worse on κ-only datasets), and since there are many more λ sequences in both training and testing datasets than κ sequences, the overall accuracy gets affected accordingly.

4.3.1.2 $\mathsf{DALAD}_{\mathrm{VJ}}$ Models

We finally state our numbers for all the $\mathsf{DALAD}_{\mathrm{VJ}}$ versions (which are essentially the VJ analogues of their respective $\mathsf{DALAD}_{\mathrm{V}}$ versions). As a reminder, these models *do* incorporate the features

Statistic	**Mean**	**Median**	**Std. Dev.**	**Minimum**	**Maximum**
AUC	0.8856	0.8873	0.0262	0.8075	0.9557
Sensitivity	0.8040	0.8113	0.0668	0.5849	0.9623
Specificity	0.7943	0.8000	0.0772	0.6167	0.9667
Accuracy	0.7988	0.7965	0.0355	0.7168	0.8850

Table 4.6: Results over 100 experiments for $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$.

Statistic	**Mean**	**Median**	**Std. Dev.**	**Minimum**	**Maximum**
AUC	0.9237	0.9271	0.0391	0.8261	1.000
Sensitivity	0.8518	0.8824	0.0973	0.4706	1.0000
Specificity	0.8487	0.8696	0.0842	0.6522	1.0000
Accuracy	0.8500	0.8500	0.0526	0.7000	0.9750

Table 4.7: Results over 100 experiments for $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$.

corresponding to the J regions.

$\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$. We again begin by discussing our results for just the λ models and datasets. This time, we can compare the results with those of both LICTOR and $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$. We refer the reader to Table 4.6 for all the statistical details of our experiments for $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$. The average AUC score for $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$ is over 88.5%, which beats both LICTOR's best model (87%) and our average for $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ (over 87.2%). Our average sensitivity here of over 80.4% is significantly better than that of LICTOR's best model (76%) and than the average sensitivity of $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ (over 77.7%). Our average specificity for $\mathsf{DALAD}_{\mathrm{VJ}}$ of over 79.4%, however, is lower than both LICTOR's best model's (82%) and than our average specificity of $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ (over 82.8%). Our average accuracy of over 79.8% is slightly lower than that of $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$, but that is mostly because of the lower specificity. That said, given that it is still quite high, we can again expect our model to be accurate with both positive and negative λ sequences with high probability. The low standard deviation numbers for all these metrics confirm their high concentration around their very promising averages.

$\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$. Next, we talk about our results for κ-only datasets. We state the exact details of our aggregate statistics on all the metrics over the 100 runs of $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$ in Table 4.7. We mostly compare these results with those for $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$. The average AUC score of over 92.3% in the case of

Test Sequence	Statistic	Mean	Median	Std. Dev.	Minimum	Maximum
$\lambda + \kappa$	AUC	0.8930	0.8915	0.0274	0.8217	0.9525
	Sensitivity	0.8079	0.8143	0.0576	0.6429	0.9429
	Specificity	0.8249	0.8313	0.0540	0.6747	0.9277
	Accuracy	0.8171	0.8170	0.0346	0.7255	0.9020
λ	AUC	0.8820	0.8836	0.0358	0.7931	0.9519
	Sensitivity	0.7991	0.7924	0.0615	0.6226	0.9434
	Specificity	0.8155	0.8167	0.0617	0.6500	0.9500
	Accuracy	0.8078	0.8053	0.0405	0.6991	0.9115
κ	AUC	0.9229	0.9284	0.0418	0.8031	0.9974
	Sensitivity	0.8353	0.8235	0.0964	0.5294	1.0000
	Specificity	0.8497	0.8696	0.0816	0.6087	1.0000
	Accuracy	0.8435	0.8500	0.0578	0.7000	0.9500

Table 4.8: Results over 100 experiments for $\mathsf{DALAD}^{\text{Joint}}_{\text{VJ}}$.

$\mathsf{DALAD}^{\kappa}_{\text{VJ}}$ is slightly better than that of $\mathsf{DALAD}^{\kappa}_{\text{V}}$ (over 92%). The average sensitivity of over 85.1% for $\mathsf{DALAD}^{\kappa}_{\text{VJ}}$, however, is notably better than that of $\mathsf{DALAD}^{\kappa}_{\text{V}}$ (over 83.3%). Additionally, the median sensitivity in case of $\mathsf{DALAD}^{\kappa}_{\text{VJ}}$ (over 88.2%) is significantly higher than that for $\mathsf{DALAD}^{\kappa}_{\text{V}}$ (over 82.3%), which implies that $\mathsf{DALAD}^{\kappa}_{\text{VJ}}$ is much more likely to be accurate on positive sequences than $\mathsf{DALAD}^{\kappa}_{\text{V}}$. That said, the average specificity here of over 84.8% is marginally worse than that of $\mathsf{DALAD}^{\kappa}_{\text{V}}$ (over 85.3%). The average accuracy of 85% for $\mathsf{DALAD}^{\kappa}_{\text{VJ}}$ though is marginally better than that for $\mathsf{DALAD}^{\kappa}_{\text{V}}$ (over 84.5%). As before, the high average accuracy number does indicate that $\mathsf{DALAD}^{\kappa}_{\text{VJ}}$ is expected to perform really well on both positive and negative κ sequences, and it is unlikely that its performance would be high on either just the positives or just the negatives. Finally, the low standard deviation numbers for each statistic (and similar to those for $\mathsf{DALAD}^{\kappa}_{\text{V}}$) imply high concentration of these statistics around the high average performances, implying the high stability and reliability of our model.

$\mathsf{DALAD}^{\text{Joint}}_{\text{VJ}}$. We proceed to the case, where our models may receive a dataset that contains a mix of both λ and κ sequences again. We refer reader to Table 4.8 for details on the aggregate statistics of our overall accuracy numbers for $\mathsf{DALAD}^{\text{Joint}}_{\text{VJ}}$. Just like $\mathsf{DALAD}^{\text{Joint}}_{\text{V}}$, $\mathsf{DALAD}^{\text{Joint}}_{\text{VJ}}$ does not train separately on the λ and the κ sequences, but simply takes them all together as inputs for training. The averages of the overall AUC scores, the overall sensitivity, and the overall specificity are over 89.3%, 80.7%, and 82.4%, respectively, which are all very high, and much better than the respective numbers for LICTOR's best model. The high overall accuracy of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ (over 81.7%) indicates the consistency and high utility in classifying both the positive and the negative sequences correctly, that is, the model is very likely to be accurate on both types of sequences simultaneously. We also computed the average of all these statistics in the same runs for λ and κ sequences, as well, and just like in the case of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, we list a few interesting observations.

- As we can see in Table 4.8, most of the statistics for $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ on the λ-only datasets are marginally better than those for $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$. The average AUC in the case of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ for λ sequences is over 88.2%, which is lower than that in the case of $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$ (over 88.5%). The same could be said about the average sensitivity for $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ (over 79.9%), which is slightly worse than that for $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$ (over 80.4%). On the other hand, the average specificity in this case (over 81.5%) is quite higher than that for $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$ (over 79.4%), resulting in a marginally higher average accuracy score for $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ than $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$.

- From Table 4.8, we can see that the average AUC score of over 92.2% for $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ for κ-only datasets is lower than that of $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$ (over 92.3%).The average sensitivity in the case of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ (over 83.5%) is significantly lower than that for $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$ (over 85.1%). The average specificity of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ (over 84.9%), however, is very marginally better than that for $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$. This results in the average overall accuracy of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ on κ-only test datasets being lower than that of $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$.

Unlike in the case of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, where the model was performing better on λ-only test datasets than $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, but worse on κ-only datasets than $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$, it is hard to conclude something of that flavour for $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$. This is because the performances of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$ are comparable on λ-only test datasets, and the same could be said for the performances of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$ on κ-only test datasets. Finally, to compare the general performance of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ with that of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, we would like to point out from Tables 4.8 and 4.4 that the average overall statistics of all the accuracy metrics (except for the sensitivity, where $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ dominates slightly) when testing on mixed datasets have similar values for both the models. The same could be said for the average performance of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ on λ-only datasets, as well.

Statistic	Mean	Median	Std. Dev.	Minimum	Maximum
AUC	0.8945	0.8940	0.0264	0.8365	0.9573
Sensitivity	0.8073	0.8143	0.0583	0.6428	0.9286
Specificity	0.8130	0.8253	0.0648	0.6386	0.9277
Accuracy	0.8104	0.8105	0.0335	0.7320	0.8889

Table 4.9: Results over 100 experiments for $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$.

$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$. We finally move on to the second version of $\mathsf{DALAD}_{\mathrm{VJ}}$ that also takes a mixture of both λ and κ sequences in training and test datasets. Just like $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ though, $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ also has two sub-models, such that one trains and tests only on λ sequences, and the other trains and tests only on κ sequences. We state all the relevant aggregate statistics for $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ in Table 4.9. We compare $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ with $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ first, but only in terms of the overall performance on the mixed datasets (since we have already done a full review of the performances of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$ on individual types of sequences). As we can see from Tables 4.9 and 4.8, the averages of all the accuracy metrics (except for specificity, which is a little higher for $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$) are quite similar. This can be justified by our comparisons of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ with $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$ on λ-only sequences, and our comparisons of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ with $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$ on κ-only sequences, because they were almost as good as each other in their respective cases. We finally compare $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ with $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ in terms of the overall performance on mixed datasets only (since the individual sub-models have already been compared above). The average AUC score and the average sensitivity of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ (over 89.4% and over 80.7%, respectively) are higher than those of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ (over 88.4% and over 78.8%, respectively), respectively. The average specificity of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ (over 81.3%), however, is lower than that of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ (over 83.2%), which brings their respective average overall accuracy scores very close to one another.

4.3.2 Best Individual Results

In this section, we describe the results of the best trained models from each version on the appropriate test datasets. For each of $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, while executing those 100 experimental runs, we saved their respective “best” (we will describe what that means below) trained models, and tested them individually.

Remark 4.3.1. As mentioned earlier, to select each model, we chose a subset of its relevant dataset (for example, the λ dataset for $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$), and split that for training and testing. For example, for $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$, we had selected all the 525 positive λ sequences and randomly selected 600 of the negative λ sequences, and had finally used 90% of each for training, and the rest for testing. In our training and testing of these individual models, not all the available data was used. This was because there are many more negative sequences than positive sequences in AL Base. To balance the numbers of positive and negative sequences, many negative sequences were never used in the training and the testing sets. So, after selecting the best model for $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$, we tested it on that original test dataset again to make sure that it is correctly classifying the positive sequences it had not been trained on because there were no other positive sequences to test on. Then we tested it again on all the λ sequences in the main dataset, which did not contain any of the training λ sequences, because there were still over 350 negative λ sequences that the model had not been either trained or tested on. The same process was repeated for all the other trained models for the other versions. One of the goals of these experiments was to use these additional negative sequences to see how much our trained models would generalize.

We first informally define a few notations. When we talk about a "complement" dataset for a model, we are referring to the complement of the training dataset for that model with respect to the original relevant dataset. For example, for $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$, we used 1012 λ sequences out of a total of 1517 λ sequences for training, so the "complement" dataset refers to the set of the remaining 505 λ sequences. By the "original" dataset, we are referring to the original test dataset that was used to test the model before selecting it (which consists of 113 λ sequences for $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$). Note that the complement datasets and the original datasets for $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$ could be different because the subset of the negative sequences to test and train on was chosen randomly from the original λ dataset. The same is true for the other models, as well. Finally, the original and the complement datasets for any given model would have the same set of positive sequences because we use all the relevant sequences in the relevant main dataset due to the scarcity of positive sequences.

We describe the test datasets and the process of selection of the trained models for each version of DALAD below.

- For $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, we chose from the set of models, whose: (1) AUC score was at least 0.91; (2) sensitivity was at least 0.84; and (3) specificity was at least 0.88. At the same time, we tried to maximize each accuracy metric. We then tested the selected model on (1) the entire complement dataset that contained all the λ sequences (that were not used for training), and (2) the original test dataset it was tested on.

- For $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$, we chose from the set of models, whose: (1) AUC score was at least 0.93; (2) sensitivity was at least 0.89; and (3) specificity was at least 0.91. We simultaneously tried to maximize each accuracy metric. We then tested the selected model on (1) the entire complement dataset that contained all the κ sequences (that were not used for training), and (2) the original test dataset it was tested on.

- For $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, we chose from the set of models, whose: (1) AUC score was at least 0.91; (2) sensitivity was at least 0.84; and (3) specificity was at least 0.88. At the same time, we attempted to maximize each accuracy metric. We then tested the selected model on (1) the entire complement dataset that contained all the λ and the κ sequences (that were not used for training), and (2) the original test dataset it was tested on.

- For $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$, we chose from the set of models, whose: (1) AUC score was at least 0.91; (2) sensitivity was at least 0.84; and (3) specificity was at least 0.90. At the same time, we tried to maximize each accuracy metric. We then tested the selected model on (1) the entire complement dataset that contained all the λ sequences (that were not used for training), and (2) the original test dataset it was tested on.

- For $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$, we chose from the set of models, whose: (1) AUC score was at least 0.95; (2) sensitivity was at least 0.90; and (3) specificity was at least 0.91. We simultaneously tried to maximize each accuracy metric. We then tested the selected model on (1) the entire complement dataset that contained all the κ sequences (that were not used for training), and (2) the original test dataset it was tested on.

- For $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, we chose from the set of models, whose: (1) AUC score was at least 0.90; (2) sensitivity was at least 0.83; and (3) specificity was at least 0.85. At the same time, we

Dataset	**Version**	**AUC**	**Sensitivity**	**Specificity**	**Accuracy**
Complement	$\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$	0.9067	0.8491	0.8142	0.8178
	$\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$	0.8977	0.8491	0.8628	0.8614
Original	$\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$	0.9377	0.8491	0.9000	0.8761
	$\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$	0.9308	0.8491	0.9333	0.8938

Table 4.10: Results for the best $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$ models on λ sequences.

attempted to maximize each accuracy metric. We then tested the selected model on (1) the entire complement dataset that contained all the λ and the κ sequences (that were not used for training), and (2) the original test dataset it was tested on.

Remark 4.3.2. We would also like to note that the thresholds we chose to select the best model for each version of DALAD were higher than the respective average statistics for those accuracy metrics.

$\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$. In Table 4.10, we compare the accuracy metrics for $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$ with respect to their respective complement and original datasets. As we can see for their respective complement datasets, both the models beat LICTOR's best model in terms of the AUC score (close to 90% in both our versions, compared to the 87% of LICTOR) and the sensitivity (close to 85% in both versions, compared to the 76% of LICTOR). The specificity of $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ (over 81.4%) is slightly lower than that of LICTOR's best model (82%), but the specificity of $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$ (over 86.2%) is much higher than that of both. With respect to their original datasets though, both our models surpass LICTOR's best model by huge margins in all accuracy measures. We would like to mention, however, that the specificity of both these models on their respective complement datasets was lower than the thresholds chosen for their respective versions while selecting them (0.88 and 0.90, respectively). The reason is that the complement datasets had 392 more unseen negative λ sequences, so we were bound to lose some accuracy on them, but that said, the specificity numbers for both these models are not that much lower than they were on their original test datasets, which shows that our models do generalize. Finally, to compare $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$ with each other, both of them have similar performance numbers, except for the specificity, which is significantly

Dataset	**Version**	**AUC**	**Sensitivity**	**Specificity**	**Accuracy**
Complement	$\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$	0.9425	0.9412	0.8360	0.8372
	$\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$	0.9652	1.0000	0.8932	0.8944
Original	$\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$	0.9949	0.9412	0.9565	0.9500
	$\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$	0.9974	1.0000	0.9565	0.9750

Table 4.11: Results for the best $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$ models on κ sequences.

higher for $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$, which makes us believe that the latter would be a better option in practice.

$\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$. In Table 4.11, we can see the high very high AUC scores for both these models on both their respective complement and their respective original datasets. The AUC score of $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$ is notably lower than that of $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$ with respect to their complement datasets (by over 2%). The sensitivity of $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$ is higher than that of $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$, but that number of positive test sequences in each test dataset is 17 for both the models, so the difference is just that $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$ misclassified one positive κ sequence, whereas $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$ classified all the 17 positive κ sequences correctly. The notable difference comes in the specificity though. Both models have similar numbers for specificity on their respective original datasets, but the specificity of $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$ on its complement dataset (over 89.3%) is significantly higher than that of $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$ on its own complement dataset (over 83.6%), despite both having a drop in that metric compared to the their numbers on their respective original test datasets (for the same reason as we had described for our λ-specific models). The original thresholds for specificity for both these models while choosing them were the same (0.91). This does show that $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$ is a much better model to use in practice because both models were simply trained on 149 positive κ sequences and 202 negative κ sequences, but the performance of $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$ on the remaining unseen 1555 negative κ sequences was very noteworthy, also showing that this model was generalizing well with very good accuracy for both positive and negative κ sequences.

$\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$. In Table 4.12, we compare the performance of $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$ on their respective original and complement test datasets in terms of the overall results on the mixture of λ and κ sequences, but also separately on just the λ and just the κ sequences.

Dataset	Version	AUC	Sensitivity	Specificity	Accuracy
Complement	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$	0.9096	0.8571	0.8401	0.8406
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$	0.9304	0.8857	0.8401	0.8416
Complement-λ	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$	0.8989	0.8679	0.8407	0.8436
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$	0.9218	0.8679	0.8496	0.8515
Complement-κ	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$	0.9155	0.8235	0.8399	0.8397
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$	0.9281	0.9412	0.8373	0.8384
Original	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$	0.9380	0.8571	0.9036	0.8824
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$	0.9468	0.8857	0.9157	0.9020
Original-λ	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$	0.9299	0.8679	0.8833	0.8761
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$	0.9456	0.8679	0.9333	0.9027
Original-κ	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$	0.9616	0.8235	0.9565	0.9000
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$	0.9514	0.9412	0.8696	0.9000

Table 4.12: Results for the best $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ models on λ and κ sequences.

First, we compare $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ with each other in terms of their respective accuracy metrics on their respective original and complement datasets containing a mixture of λ and κ sequences. In terms of the original test datasets, all the metrics of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ were higher than those of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, especially for the sensitivity (by close to 3%). In terms of the complement test datasets, $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ was superior in terms of the AUC score (by over 2%), which suggests that it would be a better model to use in practice, even though the specificity numbers for both on their respective complement datasets were less than the thresholds for specificity while choosing them (0.88 and 0.85, respectively). In terms of classification of just the λ sequences, $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ was much superior on its original test dataset than $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ was on its own original dataset as far as the negative sequences were concerned (by close to 5%), but their performances on λ sequences converged to similar numbers on their respective complement datasets. That said, the AUC score of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ was higher (by over 2%) on its complement dataset than that of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$. Finally, on just the κ sequences, both had very similar AUC scores for both their respective original and respective complement datasets, but had very different sensitivity and specificity numbers. $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ had a much higher sensitivity for κ sequences than that of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ (by an unbelievable 12%). On the original datasets though, the specificity of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ for κ sequences was lower than that of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ by close to 9%. However, the sensitivity number on κ sequences

Dataset	**Version**	**AUC**	**Sensitivity**	**Specificity**	**Accuracy**
Complement	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$	0.9239	0.8714	0.8311	0.8324
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$	0.9148	0.8857	0.8864	0.8864
Complement-λ	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$	0.9067	0.8491	0.8142	0.8178
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$	0.8977	0.8491	0.8628	0.8614
Complement-κ	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$	0.9425	0.9412	0.8360	0.8372
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$	0.9652	1.0000	0.8932	0.8944
Original	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$	0.9508	0.8714	0.9157	0.8954
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$	0.9511	0.8857	0.9398	0.9150
Original-λ	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$	0.9377	0.8491	0.9000	0.8761
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$	0.9308	0.8491	0.9333	0.8938
Original-κ	$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$	0.9949	0.9412	0.9565	0.9500
	$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$	0.9974	1.0000	0.9565	0.9750

Table 4.13: Results for the best $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ models on λ and κ sequences.

for both the models dropped and converged to values close to 84%, which was not a huge drop for $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, but it was for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$. We would like to make similar observations for both these models again like we did for their aggregate results.

- On λ sequences in their respective complement datasets, the all the accuracy metrics of both the Joint models were better than those of $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$ (except for the specificity of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, which was lower than that of $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$ by close to 2%). This is again a similar trend that we had noticed earlier in the aggregate results for these models, implying that the presence of the κ sequences in the training dataset might actually help in classifying the λ sequences more accurately, which is surprising. As an additional note, one would observe that both $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ beat LICTOR on the λ-only datasets.
- For κ sequences, both $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ suffer loss in all the accuracy metrics, just as we had observed in their aggregate statistics earlier. This seems consistent with our conjecture that the presence of λ sequences in the training dataset hurts the performance on κ sequences.

$\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$. We finally analyze the results for the best models of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ by looking at Table 4.13. We essentially just draw comparisons between $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ and

$\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$, and among the performance of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, but only in terms of their performance on datasets having a mixture of λ and κ sequences (since we have already discussed previously the comparison of both the sub-models of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ with $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ on λ-only and κ-only test datasets). We note that the numbers for the AUC scores for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ on their respective original test datasets were very similar, and so were their respective numbers for sensitivity on their respective original datasets. That said, the specificity of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ on its original test dataset was higher than that of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ on its original test dataset by over 2%. However, these gaps became larger when tested on their respective complement datasets. The AUC score of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ was higher than that of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ by close to 1%, but the sensitivity of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ was higher than that of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ by close to 1.5%. The main difference was noticed in the overall specificity though – the specificity of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ was higher than that of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ by more than 5.5%. This can be explained by the difference in the performance of their respective sub-models, which we have already discussed above. Therefore, it appears that $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ is the better model to use in practice. Now, we compare these two models with their Joint analogues. On the original test datasets, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ beats $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ on all the accuracy measures, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ beats $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ on all the accuracy measures, as well. On the complement test datasets, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ has a higher AUC score than that of $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, but with slightly lower specificity than the latter. On the other hand, $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ has a slightly lower AUC score than that of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, but has a much higher specificity (by over 4.5%). Given the performance of its sub-models, as well, this indicates that $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ would be the best and the most stable model to use among these four choices.

4.3.2.1 Additional Graphs and Figures

We provide separation histograms for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ on their respective orignal test datasets that contained a mixture of λ and κ sequences (Figure 4.1). The x-axis denotes the log odds ratios (or the logit's) of the prediction likelihoods, i.e., if the prediction value is p, then the x-axis contains values of the form $\log\left(\frac{p}{1-p}\right)$. Each of the graph shows a clear separation between the positive and the negative sequences due to our classifiers, which in another

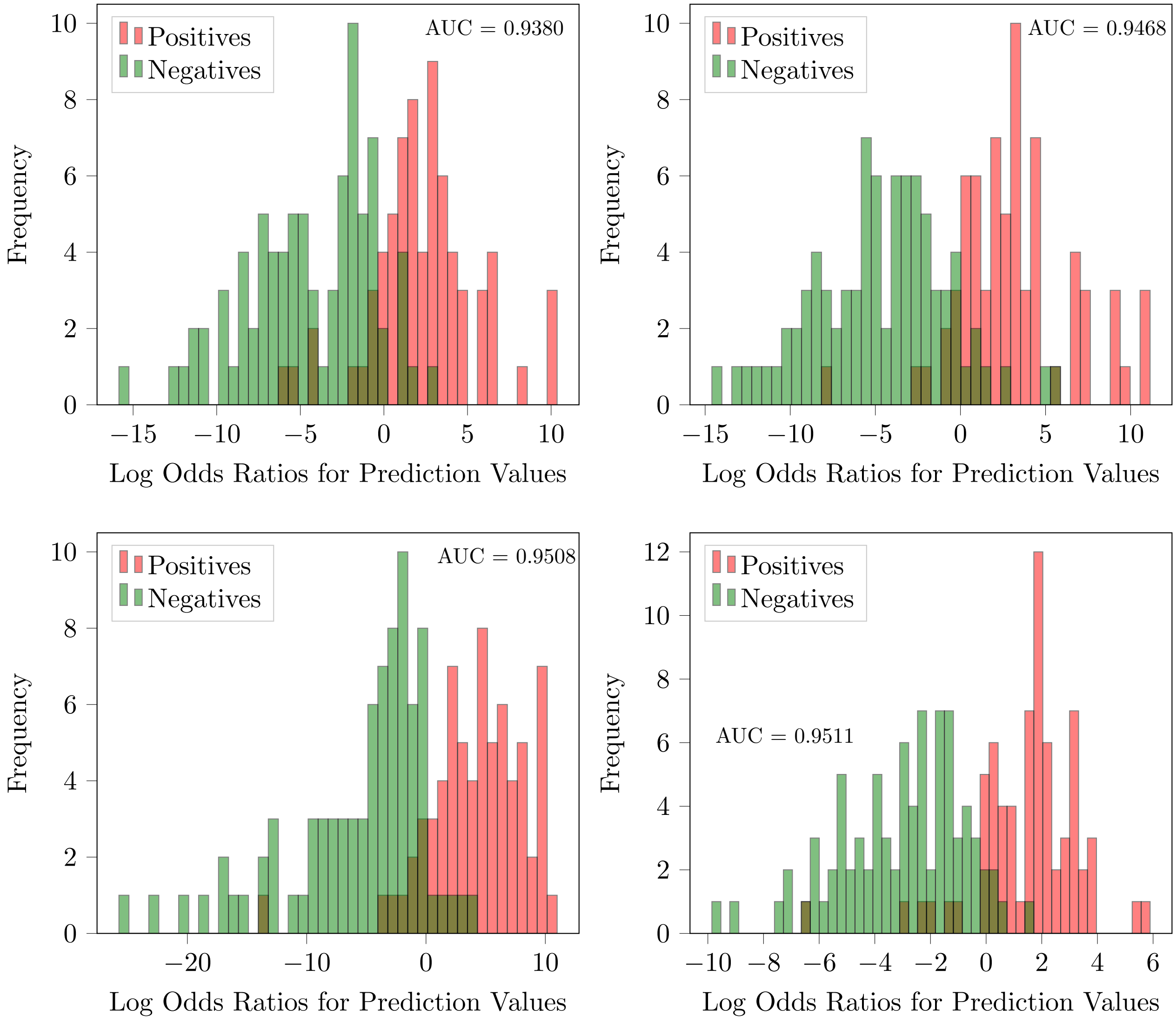


Figure 4.1: Separation histograms for $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ (from the top-left, going clockwise).

way, depicts the accuracy of these models.

We finally present the superimposed ROC curves for three different sets of models based on their respective complement test datasets (Figure 4.2).

- $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ (only for λ sequences), and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ (only for λ sequences).

- $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$, $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$ (only for κ sequences), and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ (only for κ sequences).

Model		AUC	Sensitivity	Specificity
Average Performance over 100 Runs	$\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$	0.8726	0.7774	0.8285
	$\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$	0.8856	0.8040	0.7943
	$\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$	0.8826	0.7944	0.8243
	$\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$	0.8930	0.8079	0.8249
Performance of Best Model on Complement Test Set	$\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$	0.9067	0.8491	0.8142
	$\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$	0.8977	0.8491	0.8628
	$\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$	0.8989	0.8679	0.8407
	$\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$	0.9218	0.8679	0.8496
LICTOR		0.8700	0.7600	0.8200

Table 4.14: Comparing the AUC score, the sensitivity, and the specificity of LICTOR with those of $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$, $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$, $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$, and $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$, both in terms of the average performances of our models and in terms of the performance of our best models on their respective complement test datasets. This comparison can only be done on λ sequences because of the domain limitation of LICTOR.

- $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$, $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$, $\mathsf{DALAD}^{\mathrm{Sep}}_{\mathrm{V}}$, and $\mathsf{DALAD}^{\mathrm{Sep}}_{\mathrm{VJ}}$ for the performance on test datasets that contain both λ and κ sequences.

4.4 Discussion

Here, we provide further discussions and a recapitulation of the consequences of our experiments. We aim to provide a more abstract picture of our findings, and try to bring the attention of the reader to the main takeaways.

4.4.1 Comparison with LICTOR

We have compared the performance of $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$, $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$, $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$, and $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$ with that of LICTOR's best model on λ sequences, both with respect to our aggregate statistics of the relevant accuracy measures over 100 runs of each of our version, and with respect to the best choices of models for each of our versions (see Table 4.14). To remind the reader, the AUC score of LICTOR's best model was 0.87, and its sensitivity and accuracy were 0.76 and 0.82, respectively.

In the aggregate statistics, the AUC scores of $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$, $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$, $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$, and $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$ were 0.8726, 0.8856, 0.8826, and 0.8820, respectively, which show that all these versions were

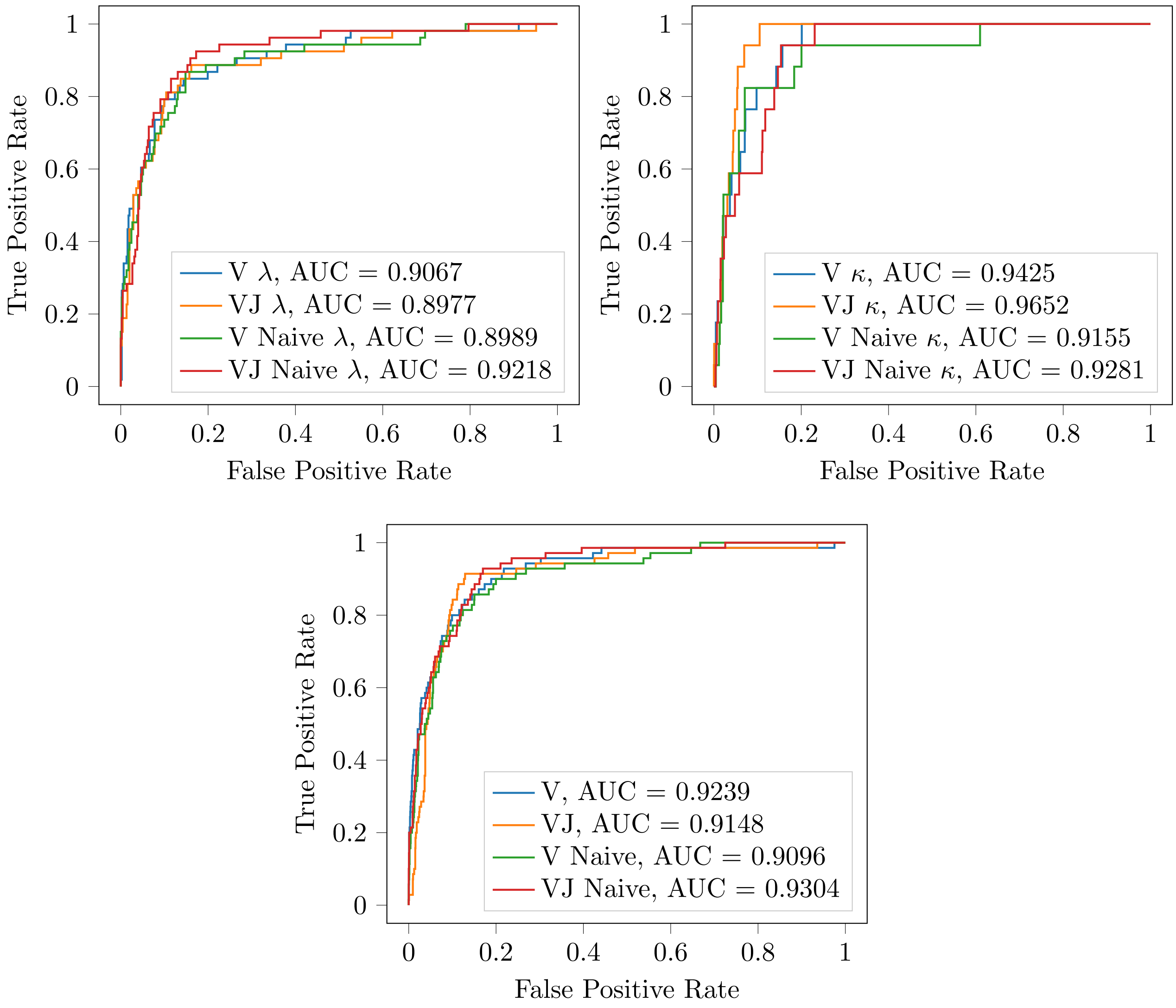


Figure 4.2: ROC curves for three different situations. The one on the left in the top row compares the ROC curves of $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ just on their respective λ-only cmplement test datasets. The ROC curve on the right in the top row correspond to those of $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ on their respective κ-only complement test datasets. The figure in the bottom row show the ROC curves corresponding to $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ on their respective complement test datasets that contained a mixture of λ and κ sequences.

definitely better than LICTOR's best model on an average in terms of the AUC scores. The average sensitivity on λ sequences of these versions were 0.7774, 0.8040, 0.7885, and 0.7991, respectively, which are all higher than that of LICTOR's best model, especially in the case of $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$. The specificity of these versions were 0.8285, 0.7943, 0.8155, and 0.8155, respectively. In this case, except for $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$, all the other models had an average specificity that was either very close or higher than that of LICTOR's best model. This ultimately shows that on an average, our models are expected to perform better than LICTOR's best model.

In terms of the best respective models for the aforementioned versions, $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$, $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$, $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$, and $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$ had AUC scores on the λ sequences of 0.9067, 0.8977, 0.8989, and 0.9218, respectively, which are much higher than the 0.87 AUC score of LICTOR's best model. The sensitivity of our best models in the same order were 0.8491, 0.8491, 0.8679, and 0.8679, respectively, which are again very significantly higher than that of LICTOR's best model (by at least 9%). Finally, the specificity scores of these four models were 0.8142, 0.8628, 0.8407, and 0.8496, respectively. Even for this metric, all models (except for $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$) show superior numbers than LICTOR's best model. Therefore, as far as the best models are concerned, all our models perform significantly better than the state-of-the-art.

Hence, in addition to being functional for κ sequences, our methodology is very accurate on λ sequences, as well, as both the aggregate statistics and the accuracy metrics for our best models show. Note that our pre-processing scheme of the λ sequences is similar to that of LICTOR, so our improvements could not have happened just due to pre-processing alone. We were able to use some additional information, like the GL sequences (which were also utilized by LICTOR, but in a different way) and the J gene segments (which, as LICTOR concluded, were not useful for their model), which along with the design of our model, gave the superior performance.

4.4.2 $\mathsf{DALAD}_{\mathrm{V}}$-vs-$\mathsf{DALAD}_{\mathrm{VJ}}$

In this part of the discussion, we ask the following question: do the additional features corresponding to the J regions contribute anything?

From the average statistics, it is difficult to infer what might be better. For example, for the

λ models, $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$ has notably better average AUC score and average sensitivity than $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$, but its average specificity is lower. In case of $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$, the average AUC scores and the specificity look similar, but there is a 2% increase in the sensitivity for $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$. For the Joint models, there isn't much change in the average statistics for the overall accuracy metrics, which is mostly because the $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$ had more accurate predictions for κ sequences than $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$, but had less accurate predictions for λ sequences than $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$, and this made its overall accuracy quite similar to that of $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$. So, it appears that on an average, the features corresponding to the J regions make predictions a little more accurate in different cases for different versions of DALAD, for example, by increasing the sensitivity of the κ models.

Now, we compare the performance of the best $\mathsf{DALAD}_{\mathrm{VJ}}$ models with that of the best $\mathsf{DALAD}_{\mathrm{V}}$ models. In case of $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$, the sensitivity and the AUC scores are within 1% of each other, but the specificity of $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$ is higher than that of $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ by close to 5%, which is a massive improvement. In case of $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$, the AUC score of $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$ is higher than that of $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$ by over 2%, which is a big improvement. The sensitivity of $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$ is also much higher, but as we discussed previously, we cannot conclude much from that because there aren't that many positive data points available to test on. However, the specificity of $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$ is higher than that of $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$ by almost 6%, which is a huge improvement, especially given that there are way more negative sequences. In this case, the J regions do seem to provide extra accuracy. In case of the Joint models, the AUC score of $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$ is higher than that of $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$ by over 2%, and the sensitivity of $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$ is also higher than that of $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$ by close to 3%. On the λ sequences alone, $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$ seem to have similar performance, but on κ sequences alone, $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$ has a much better sensitivity than that of $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$. Based on all this information about the best models, the $\mathsf{DALAD}_{\mathrm{VJ}}$ models do exhibit better performance than their $\mathsf{DALAD}_{\mathrm{V}}$ counterparts.

Finally, we perform additional t-tests over those 100 runs for: (1) $\mathsf{DALAD}^{\lambda}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\lambda}_{\mathrm{VJ}}$; (2) $\mathsf{DALAD}^{\kappa}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\kappa}_{\mathrm{VJ}}$; and (3) $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{V}}$ and $\mathsf{DALAD}^{\mathrm{Joint}}_{\mathrm{VJ}}$. The results are recorded in Table 4.15. For the λ and κ versions, the negative t-values and the p-values being less than 0.05 indicate clear improvements in the AUC scores when adding the J regions in. This is not true for

Version	***p*-Value**	**Sign of *t*-Value**
DALAD^{λ}	0.000000	Negative
DALAD^{κ}	0.017979	Negative
$\mathsf{DALAD}^{\text{Joint}}$	0.432382	Negative

Table 4.15: For λ, κ, and Joint models, we compute the approximate p-values for the t-paired sampled tests between the AUC scores of their respective $\mathsf{DALAD}_{\text{V}}$-vs-$\mathsf{DALAD}_{\text{VJ}}$ versions.

the Joint models though – that said, the average statistics and the metrics for the best models in this case indicate otherwise.

4.4.3 Comparing Joint and Sep Models

One of our novel ideas in this work was to have a model for classification of both λ and κ sequences, which was composed of two sub-models ($\mathsf{DALAD}_{\text{V}}^{\text{Sep}}$ and $\mathsf{DALAD}_{\text{VJ}}^{\text{Sep}}$) – one each for λ and κ sequences. The goal was to present this idea as a proof-of-concept, and compare it with the model, which was just composed of one neural network that trained on both types sequences together without separating them ($\mathsf{DALAD}_{\text{V}}^{\text{Joint}}$ and $\mathsf{DALAD}_{\text{VJ}}^{\text{Joint}}$). As we saw in Section 4.3, the Sep models perform better in different ways than the corresponding Joint models, both on an average and in the individual runs for our best respective models.

On analyzing Tables 4.12 and 4.13, we concluded that $\mathsf{DALAD}_{\text{VJ}}^{\text{Sep}}$ was the best model to use in practice because of its high and consistent numbers on all the accuracy statistics (which we believe would generalize well, too).

4.4.3.1 Individual Performance on λ and κ datasets

On λ and κ sequences separately, $\mathsf{DALAD}_{\text{VJ}}^{\text{Sep}}$ was performing better than both $\mathsf{DALAD}_{\text{V}}^{\text{Joint}}$ and $\mathsf{DALAD}_{\text{VJ}}^{\text{Joint}}$ (except for the sensitivity on λ sequences, where it was a little lower for $\mathsf{DALAD}_{\text{VJ}}^{\text{Sep}}$ compared to that of $\mathsf{DALAD}_{\text{VJ}}^{\text{Joint}}$, but we observed that the availability of κ sequences somehow seemed to improve the performance of the Joint models on the λ sequences). We believe that while mixing in both types of sequences worsens the performance of $\mathsf{DALAD}_{\text{V}}^{\text{Joint}}$ and $\mathsf{DALAD}_{\text{VJ}}^{\text{Joint}}$ on the κ sequences, their performance on the λ sequences gets better because the mutation behaviours in the λ sequences that lead to the onset of AL amyloidosis also include those similar to the ones in the κ

sequences, but they happen less frequently in the λ sequences. This way, the mutation behaviours from the κ sequences add to the information about the λ mutations, and improves the performance on the λ sequences. At the same time, especially because we had much fewer κ sequences to train $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ on, the other mutation behaviours from the λ sequences might have confused those models when testing on the κ sequences, thereby hurting their performance on the κ sequences.

Based on this, one suggestion that we could offer to improve the performance of $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$ even further is that while training its λ module, we could mix a certain number of κ sequences, as well, to improve its numbers on λ sequences. The κ module could still remain the same. With some additional tweaking to the hyperparameters, and with more data, we could develop a combined (that works on both λ and κ sequences) model that is even better than $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$.

4.4.4 Better Performance on κ Sequences

One major observation for the keen reader is that the models trained and tested solely on κ sequences performed significantly better than the models trained and tested on λ sequences. Unfortunately, we do not have an answer for this yet, but we do have two plausible explanations for this.

1. We conjecture that the mutations that happen in κ sequences during the onset of AL amyloidosis are probably more significant and prominent than those in case of λ sequences. In other words, the κ sequences exhibit behaviours in case of the patient having the disease that are much more identifiable and those that we can more easily categorize. These behaviours are also what we aim to understand in future from the perspective of biology, too.

2. We also believe that a larger dataset with κ sequences would have made it easier for us to test our models, and compare them with their performance on the λ sequences. This would have just given higher-confidence (and even more representative) accuracy numbers for comparison.

4.4.5 Dataset Size and Composition

Despite the promising and high-quality accuracy numbers of our models of DALAD, we believe that having more data could have helped improve our models even further. What we had available to us was a meagre set of about 4000 light chain sequences (or data points), which included both λ and κ sequences. Given that the two types of sequences behave and mutate differently in the positive and the negative cases, such a small number of samples for each was very challenging while attempting to obtain models that could characterize all these behaviours accurately for both these types of sequences. At the same time, when we're incorporating more features (like those from the J regions) into our model, we would ideally like to have more training samples in order for the models to generalize and start exhibiting their true potential for even more accurate predictions. In some sense, having more features increases the "dimensionality" of the models, which means more training samples for reliable training. That said, our DALAD_{VJ} models show better accuracy in many settings than their DALAD_{V} counterparts not because of any coincidence, since we also had aggregate statistics for each of their accuracy metrics over many runs each that confirmed our hypothesis.

This issue was further exacerbated by the fact that there were many more negative examples than positive examples, especially in case of the κ sequences, where the number of negative sequences was more than 10 times the number of positive sequences. Hence, we could not have utilized a major fraction of the κ dataset for training because otherwise our models could have become biased towards classifying any light chain as negative, thereby, significantly lowering the sensitivity. For instance, the number of positive κ sequences was 166, while the number of negative κ sequences was 1757. Since we performed a $9:1$ split for training and testing after balancing the dataset (which involved using all 166 positive sequences, but just 225 randomly selected negative sequences), we ended up having to ignore 1532 negative sequences, which was nearly 80% of the entire κ dataset, and nearly 40% of the main dataset. We had to balance the λ dataset, as well, but the problem was not as pressing as in the κ case, since the λ dataset had 525 positive sequences and 992 negative sequences, and by selecting all the positive sequences and 600 random negative sequences, we were not losing much from the main dataset, as compared to the κ models.

Chapter 5

Conclusion

We conclude this thesis by summarising: (1) what we managed to achieve through our work; (2) what the limitations are and what could have helped to improve our performance; (3) what we conjecture based on our findings; and (4) what a few potential future directions could be.

5.1 Our Accomplishments

We developed a set of new machine learning models (DALAD) to predict the onset of AL amlyloidosis by looking at the light chain sequences from patients. Our models were based on convoultional neural networks, which combined the convolutional module with a deep neural network consisting of multiple hidden layers. We developed multiple versions of DALAD that tackled the prediction problem on different kinds of datasets – λ-only datasets, κ-only datasets, and datasets containing a mixture of both λ and κ sequences.

Features of DALAD. There are two key distinguishing features of DALAD. (1) Unlike most other prior works, our model can tackle both λ and κ sequences. (2) Our $\mathsf{DALAD}_{\mathrm{VJ}}$ versions use both the V and the three regions of the LC sequences, including both the λ and κ varieties.

Using a Custom Germline Database. DALAD uses a custom 355-sequence GL database that contains all the possible combinations of the V and J regions for both the λ and the κ sequences from the IMGT repository in order to detect the potential mutations that could potentially lead to AL amyloidosis.

Versions and Experimental Procedures. Our models included: (1) $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$, which were trained and tested solely on λ sequences; (2) $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$ that were trained and tested only on κ sequences; (3) $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$, which had just one model each that was trained and tested on a mixture of both λ and κ sequences simultaneously; and (4) $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Sep}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Sep}}$, which had two sub-models, one each for λ and κ sequences alone, that were trained and tested on the respective types of sequences separately. We first selected the hyperparameters for each of these versions through multiple runs of training and testing on smaller datasets than the ones we finally used to train and test. We then tested the accuracy of our models with their respective hyperparameter choices by testing each over 100 runs, and computing the aggregate statistics for each of the accuracy measures that we considered in this work. Finally, we selected the best trained model for each version of DALAD, and evaluated it on the relevant complement of its respective training dataset.

Superior Accuracy on λ Sequences. On λ sequences alone, our DALAD models comfortably beat the state-of-the-art models (like LICTOR's best model) in terms of different useful accuracy metrics, such as AUC score, sensitivity, and specificity. This is true both in terms of the average statistics and the accuracy measures for our best models. The models that we used for this were $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$.

High Accuracy on κ Sequences. We achieved even higher accuracy on κ sequences than we did for λ sequences. This is true again both in terms of the aggregate statistics over multiple runs of each model, and in terms of the individual accuracy numbers for our best trained models for each relevant version. The versions we used in this process were $\mathsf{DALAD}_{\mathrm{V}}^{\kappa}$, $\mathsf{DALAD}_{\mathrm{VJ}}^{\kappa}$, $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$, and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$.

Incorporating the J Region. From our experimental results in Chapter 4 and our discussion in Section 4.4, we concluded that the J regions do appear to help in achieving higher accuracy while attempting to classify both the λ and the κ sequences correctly. The improvement happens in different accuracy metrics for different cases, but we believe that with more data and more in-depth research, new models could be discovered in future that would utilize the information from the J regions even further to get better accuracy in their predictions.

Other Conclusions. In Section 4.4, we talked about our conjectures about the behaviours of λ and κ sequences in case they are truly positive. This was based on our findings that training jointly on λ and κ sequences together (as in $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$) improved the accuracy while classifying the λ sequences (as opposed to training simply on the λ sequences alone), but at the same time, decreased the accuracy on the κ sequences (as opposed to training only on the κ sequences alone).

5.2 Limitations and Future Directions

We finally discuss the limitations of our work, and propose a few future directions to facilitate better classification models for this problem.

Limitations. As mentioned in Section 4.4, the lack of data did hinder what we could have potentially done while working on this problem. Incorporating more features helped us, but in order to see its full potential, we would have liked to have larger datasets available to us for both λ and κ sequences, which had a larger number of both positive and negative sequences for each. The other issue we discussed in Section 4.4 was that $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ seemed to be performing better than $\mathsf{DALAD}_{\mathrm{V}}^{\lambda}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\lambda}$ on the λ sequences. We proposed a new idea that in our Sep models, the module for λ sequences could be trained on a small number of κ sequences, as well, which could increase the accuracy numbers in that case, too. That said, this calls for more data again. The final concern that we would like to highlight is that the set $\mathcal{H}$ of hyperparameters that we had for each version of DALAD contained 216 choices. With more resources and data, we could have broadened this universe of hyperparameters even further, and could have possibly selected even better values for each version.

Future Directions. We believe that our work has opened up new avenues for research on computational methods for predicting the onset of AL amyloidosis. As discussed above, with more resources and data, even larger and more accurate deep neural networks (possibly convolutional) could be constructed with better choices of hyperparameters. Next, incorporating new features could also help obtaining more accurate predictions. We mentioned earlier that the J region does have more potential, but at the same time, one could also look into other features that could be potentially useful for this task, for example, the secondary structures. More features means more information, but at the same time, as we have remarked a few times already now, having enough data is important in order to be able to make full use of that information in order to be able to generalize well and get better accuracy. Another interesting direction, but in terms of just biology, comes from our findings from the experiments on $\mathsf{DALAD}_{\mathrm{V}}^{\mathrm{Joint}}$ and $\mathsf{DALAD}_{\mathrm{VJ}}^{\mathrm{Joint}}$ (that these models tend to perform better on λ test sequences than the models trained simply on λ sequences alone), which indicate that, despite having different mutations and behaviours in case of positive sequences, the λ and the κ sequences may exhibit certain similar behaviours, and it could be useful to understand these similarities to help facilitate more accurate predictions. Finally, our model Sep structure could be used for prediction in cases of other diseases, as well, where there are multiple kinds of data exhibiting conflicting or misleading behaviours available, and learning jointly on all of them together may not be feasible either due to computational or accuracy bottlenecks.

Bibliography

[1] Stephen F. Altschul et al. "Basic local alignment search tool". In: *Journal of Molecular Biology* 215.3 (1990), pp. 403–410. ISSN: 0022-2836. DOI: `https://doi.org/10.1016/S0022-2836(05)80360-2`. URL: `https://www.sciencedirect.com/science/article/pii/S0022283605803602`.

[2] Elizabeth M Baden et al. "Structural insights into the role of mutations in amyloidogenesis". en. In: *J. Biol. Chem.* 283.45 (Nov. 2008), pp. 30950–30956.

[3] Merrill D Benson, Juris J Liepnieks, and Barbara Kluve-Beckerman. "Hereditary systemic immunoglobulin light-chain amyloidosis". en. In: *Blood* 125.21 (May 2015), pp. 3281–3286.

[4] Kip Bodi et al. "AL-Base: a visual platform analysis tool for the study of amyloidogenic immunoglobulin light chain sequences". en. In: *Amyloid* 16.1 (Mar. 2009), pp. 1–8.

[5] Oscar Conchillo-Solé et al. "AGGRESCAN: a server for the prediction and evaluation of "hot spots" of aggregation in polypeptides". en. In: *BMC Bioinformatics* 8.1 (Feb. 2007), p. 65.

[6] D P Davis et al. "Both the environment and somatic mutations govern the aggregation pathway of pathogenic immunoglobulin light chain". en. In: *J. Mol. Biol.* 313.5 (Nov. 2001), pp. 1021–1034.

[7] Angela Dispenzieri, Morie A Gertz, and Francis Buadi. "What do I need to know about immunoglobulin light chain (AL) amyloidosis?" en. In: *Blood Rev.* 26.4 (July 2012), pp. 137–154.

[8] James Dunbar and Charlotte M Deane. "ANARCI: antigen receptor numbering and receptor classification". en. In: *Bioinformatics* 32.2 (Jan. 2016), pp. 298–300.

[9] Ana-Maria Fernandez-Escamilla et al. "Prediction of sequence-dependent and mutational effects on the aggregation of peptides and proteins". en. In: *Nat. Biotechnol.* 22.10 (Oct. 2004), pp. 1302–1306.

[10] Oxana V Galzitskaya, Sergiy O Garbuzynskiy, and Michail Yurievich Lobanov. "Prediction of amyloidogenic and disordered regions in protein chains". en. In: *PLoS Comput. Biol.* 2.12 (Dec. 2006), e177.

[11] Maura Garofalo et al. "Machine learning predicts immunoglobulin light chain toxicity through somatic mutations". In: *bioRxiv* (2020). DOI: `10.1101/849901`. eprint: `https://www.biorxiv.org/content/early/2020/12/15/849901.full.pdf`. URL: `https://www.biorxiv.org/content/early/2020/12/15/849901`.

[12] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". en. In: *Nature* 596.7873 (Aug. 2021), pp. 583–589.

[13] Pamina Kazman et al. "Fatal amyloid formation in a patient's antibody light chain is caused by a single point mutation". en. In: *Elife* 9 (Mar. 2020).

[14] Changsik Kim et al. "NetCSSP: web application for predicting chameleon sequences and amyloid fibril formation". en. In: *Nucleic Acids Res.* 37.Web Server issue (July 2009), W469–73.

[15] Marie-Paule Lefranc et al. "IMGT®, the international ImMunoGeneTics information system® 25 years on". en. In: *Nucleic Acids Res.* 43.Database issue (Jan. 2015), pp. D413–22.

[16] Sebastian Maurer-Stroh et al. "Exploring the sequence determinants of amyloid structure using position-specific scoring matrices". en. In: *Nat. Methods* 7.3 (Mar. 2010), pp. 237–242.

[17] K.M. Murphy and C. Weaver. *Janeway's Immunobiology: Tenth International Student Edition with Registration Card.* Titolo collana. W.W. Norton, 2022. ISBN: 9780393884913. URL: `https://books.google.ca/books?id=uiabzgEACAAJ`.

[18] Mengting Niu et al. "RFAmyloid: A web server for predicting amyloid proteins". en. In: *Int. J. Mol. Sci.* 19.7 (July 2018).

[19] M I F J Oerlemans et al. “Cardiac amyloidosis: the need for early diagnosis”. en. In: *Neth. Heart J.* 27.11 (Nov. 2019), pp. 525–536.

[20] Luis del Pozo Yauner et al. “Influence of the germline sequence on the thermodynamic stability and fibrillogenicity of human lambda 6 light chains”. en. In: *Proteins* 72.2 (Aug. 2008), pp. 684–692.

[21] Puneet Rawat et al. “Exploring the sequence features determining amyloidosis in human antibody light chains”. In: *Scientific Reports* 11.1 (July 2021), p. 13785. ISSN: 2045-2322. DOI: 10.1038/s41598-021-93019-9. URL: https://doi.org/10.1038/s41598-021-93019-9.

[22] Usman A Tahir et al. “Predictors of mortality in light chain cardiac amyloidosis with heart failure”. en. In: *Sci. Rep.* 9.1 (June 2019), p. 8552.

[23] Antonio Trovato et al. “Insight into the structure of amyloid fibrils from the analysis of globular proteins”. en. In: *PLoS Comput. Biol.* 2.12 (Dec. 2006), e170.

[24] Zhiyong Wang et al. “Protein 8-class secondary structure prediction using conditional neural fields”. en. In: *Proteomics* 11.19 (Oct. 2011), pp. 3786–3792.

[25] Jian Ye et al. “IgBLAST: an immunoglobulin variable domain sequence analysis tool”. en. In: *Nucleic Acids Res.* 41.Web Server issue (July 2013), W34–40.

[26] Sukjoon Yoon and William J Welsh. “Detecting hidden sequence propensity for amyloid fibril formation”. en. In: *Protein Sci.* 13.8 (Aug. 2004), pp. 2149–2160.