UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculty of Sciences
School for Information Technology
Master of Statistics and Data Science

Master's thesis

Prediction of T-cell Receptor peptide binding using machine learning and ab initio principles

Ömer Sercik

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Biostatistics

Confidential

SUPERVISOR :
Prof. dr. Dirk VALKENBORG
Prof. dr. Bart CLEUREN

2021
2022

Transnational University Limburg is a unique collaboration of two universities in two countries: the University of Hasselt and Maastricht University.

UHASSELT
KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

Maastricht University

Faculty of Sciences
School for Information Technology
Master of Statistics and Data Science

Master's thesis

Prediction of T-cell Receptor peptide binding using machine learning and ab initio principles

Ömer Sercik

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Biostatistics

Confidential

SUPERVISOR :
Prof. dr. Dirk VALKENBORG
Prof. dr. Bart CLEUREN

Contents

Acknowledgement

I want to thank my supervisors prof. dr. Dirk Valkenborg and prof. dr. Bart Cleuren for their support and feedbacks during the project. Not only did they guide me through this project, but they also motivated me during my preparations for a PhD position. Bart is an excellent physicist who was also my supervisor during my bachelor project. Dirk is an engineer and statistician who is capable of having multiple meetings, reading multiple e-mails and drinking a lot of coffee, all at the same time, which is also something I've always admired. Thanks Bart, thanks Dirk, for the amazing coaching. This is not our first project where we've worked together, and I hope it won't be the last one!

Also a special thanks to my family and girlfriend for their motivation and support, and being okay that I was a "little" busy finalizing my master.

1 Abstract

Prediction of the binding between epitopes and T-cell receptors is equivalent to studying the activity of the adaptive immune response, next to B-cell activity. Thanks to the advanced technology and experimental setups, it is possible to sequence TCRs and the epitopes, leading to the possibility to analyse the specificity between the two entities. There are many possible TCRs (thanks to DNA recombination) and many possible epitopes (viral, bacterial, ...) which still makes it challenging to determine how the binding between these two proteins is governed. In this project, we aim (1) to build classifiers that will predict whether there will be a binding, given the TCR and the epitope and (2) given the epitope, to predict the sequence the TCR should have in order to have a binding. For the first task, machine learning models such as classification trees, random forest, boosting, logistic regression and KNN and deep learning models like densely connected neural networks and convolutional neural networks, are used. For the second task, only random forest models are used.

The code, data set (the version downloaded at March, 2022) and the generated data sets for the second task can be found at https://github.com/osercik/TCR-Epitope-Prediction.

2 Introduction

The immune system can be regarded as a complex network in the human body with a lot of types of cells and many types of reactions and responses that can occur. The main aim of the immune system is to defend the organism from invading pathogens. In case the invading succeeds, the immune system starts a response against the invading pathogen or against the residues (in many cases, those are proteins) the invader leaves. In this thesis, the focus relies on one type of an immune response, mediated by one type of an immune cell. In order to do so, it is important to have a general overview of the immune system. In this section, a general summary will be given. For the interested reader who wants to have a broader picture, more advanced textbooks in molecular biology are recommended, such as *Molecular Biology of the Cell* by Alberts et al. This introductory section is based on the content of this book. So are Figure 1, Figure 2, Figure 3 and Figure 5 coming from the associated chapter about the immune system (Chapter 24: The Innate and Adaptive Immune Systems).

The immune system can be divided into two parts, namely the innate immune system and the adaptive immune system. The innate immune system is the collection of the tools for defense that are present in the human body since birth, whereas the adaptive immune system is the one that is being *updated* continuously, as humans grow, become sick, get in touch with new pathogens, etc. Furthermore, an adaptive immune response can be mediated by two types of cells: B cells and T cells. Their fundamental biological difference is in the way they fight against the invasion:

- B cells are attacking the invader directly, by means of secretion of specialized proteins, or antibodies.
- T cells will rather defend the body by killing the already infected cells.

Those four terms - namely innate immune system, adaptive immune system, B cells and T cells - are important to master as they are key terms for this topic. Let's look at them in more detail.

2.1 Innate vs. Adaptive Immune System

The difference in the innate and adaptive immune system can already be read through their names. The innate immune system is that part which is present since the day of birth. It contains primitive tools to defend the human body against invading pathogens. Nevertheless, co-operation between the innate and the adaptive immune system is one of the key features which makes the immune system powerful. This co-operation is summarized in Figure 1. The (new) pathogens will be caught first by the innate immune system. The activation of the adaptive immune system will be very slow in these first stages, since the specific B cells and T cells for that (new) pathogen are low in numbers and concentration. As hours and days proceed, the adaptive immune system will start to *learn* the pathogen and produce specific B cells and T cells against it.

The innate immune system contains the following important *first* catchers: epithelial cells, pattern recognition receptors, phagocytosis and natural killers cells. The very first barrier the invader will see, are the epithelial cells. Those cells are the boundry between the inner and outer world of the body. The epithelial cells are physical and chemical barriers and are avoiding the attachement and entry of the pathogens into a cell. The next tool of the innate immune system are the pattern recognition receptors (PRRs). Those PRRs are specialized such that they memorize patterns common to pathogens, called pathogen associated molecular patterns (PAMPs). The PAMPs can be

regarded as the ID of the pathogen, recognized by a PRR. The locations of PRRs are highly variable, but mostly they are found inside cells membranes (transmembrane proteins), freely in cytosol or associated with the membranes of the endolysosomal system.

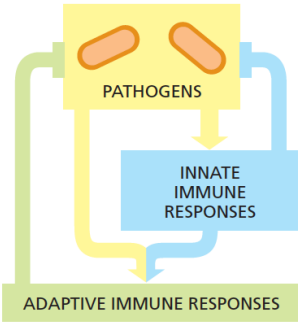

Figure 1: The Innate and Adaptive Immune System.

To give an example of the working mechanism of a PRR, let's consider TLR3, a toll like receptor. This PRR is located in the endolysosomal system and recognizes the double-stranded RNA of viruses. Next part of the innate immune system is phagocytosis. This is the proces where pathogens are seeked, engulfed and destroyed by two main phagocytic cells, namely macrophages and neutrophils. Also these cells contain PRRs and other receptors responsible for the recognition of pathogens. Macrophages are the cells that survive the fight against the pathogens, whereas neutrophils will always die, forming the main component of pus.

The last important part of the innate immune system are the natural killers cells (NKs). NKs are cells specialized such that they are able to program already infected cells (viral) to do apoptosis. The molecular pathway is summarized in Figure 2.

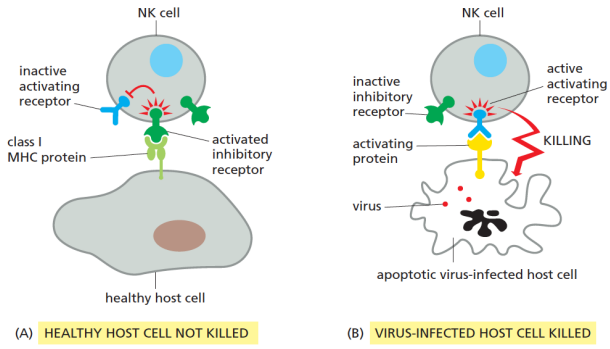

Figure 2: Molecular Pathway of NKs.

In Figure 2, both a healthy cell and an infected cell are given. The differences between the two are the levels of class I MHC and activating proteins on the surface of a normal and an infected cell.

The latter will display a low level of class I MHC proteins and a high level of activating proteins, causing a molecular signal that leads at the end to apoptosis of the infected cell. This concludes the summary of the innate immune system.

The main difference in the innate and the adaptive immune system is the level of specificity. The innate immune system has a low level of specificity. It is the first line an invader will encounter, whereas the adaptive immune system has a very high level of specificity. This leads to the need of defining the *invader* more precisely. However, in the framework of the adaptive immune system, one has to make clear difference between what type of invader: bacteria, virus, fungi, parasites.

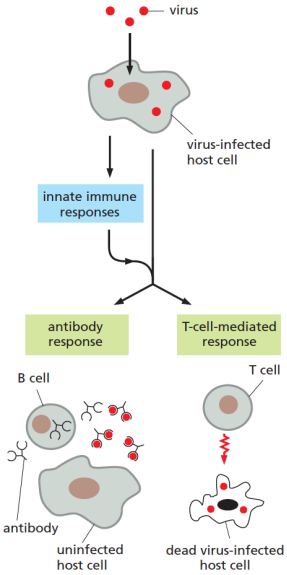

Figure 3: B Cells vs. T Cells.

The adaptive immune system can act in two possible ways:

- Antibody response. Here, the B cells will produce specialized proteins called antibodies (ABs) after activation. The ABs will bind then to the invader, which are mostly extracellular viruses and microbial toxins. After the binding of the AB, phagocytic cells will recognize these bound ABs and start the phagocytosis process.

- T cell mediated immune response. This type of immune response differs fundamentally from the antibody response. Rather than attacking invaders, T cells will recognize products of viruses, bacteria, ... that are bound to MHC protein complexes. In many cases, the *products* are proteins (viral proteins, for example), also called epitopes. These pMHC (protein - MHC) complexes will then be displayed on the surface of the infected cells, which will be recognized by T cells.

Although very complex and extensive, the immune system can be regarded as a chain of commands for recognition, binding, cleaning or killing. The focus of this project is on T cell mediated immune response. That's why as a last part of this introduction, the focus will on T cells and not on B

cells. After the T cell mediated response is explained, the translation will be made to the project and to the specific research questions. In Figure 3, the working mechanisms of B cells and T cells are shortly summarized.

2.2 T Cells and pMHC Complexes

Since the focus of this dissertation relies on T cells, let's focus on the working mechanisms of T cells in more detail. The generalization is made by using just the terms T cells, MHC and the foreign peptide (viral, bacterial, ...), but in reality, T cells and MHC complexes exist in different types. In particular:

- There are a few T cells, differing from each other by means of their functionality:
  - Cytotoxic T cells, helper T cells and regulatory T cells: these cells become effector T cells after activation.
  - Effector cytotoxic T cells: killing cells infected with a virus or other intracellular pathogen.
  - Effector helper T cells: stimulating other immune cells for an immune response, such as macrophages, dendritic cells, B cells and cytotoxic T cels.
  - Effector regulatory T cells: suppressing the activity of other immune cells.
- There are two types of MHC (Major Histocompatibility Complex) proteins, called HLA (Human Leukocyte Antigens) in humans:
  - Class I: presenting the foreign molecule to cytotoxic T cells
  - Class II: presenting the foreign molecule to helper and regulatory T cells

The high variability in the specificity of T cells, more precisely, in the T cell receptors (TCR) is due to the possibility of the V(D)J recombination in the chromosome locations encoding for the TCR protein. TCRs are proteins and like each protein produced in an organism, the origin relies in the DNA. DNA is read to produce the mRNA (transcription). This mRNA is then read again to produce the associated protein (translation). Let's take as a part of DNA $X_1, X_2, ...X_V, Y_1, Y_2, ..., Y_D, Z_1, Z_2, ..., Z_J$. Reading this from left to right each time, will deliver each time the same protein. In this case, there is no diversity. However, recombining (switching places, removing some parts, ...) will cause some diversity. A possible recombination is schematically given in Figure 4.

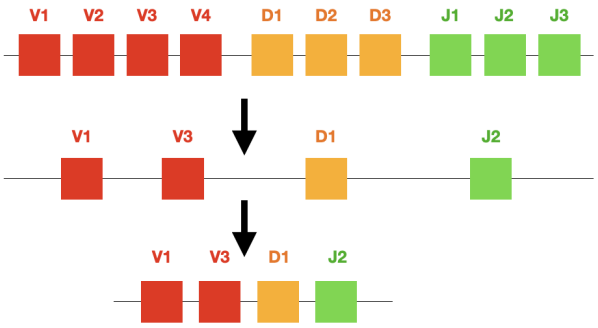

Figure 4: V(D)J recombination.

T cells are proteins consisting of two chains: an α and a β chain. The first one is generated through VJ recombination while the latter from V(D)J recombination (Krangel, 2009). The working mechanism of TCRs is presented in Figure 5. In short, T cells start an immune response as follows. As given in Figure 5, an activated dendritic cell contains a foreign protein in it. Still inside this cell, this protein binds to a MHC protein complex. This bigger complex, called pMHC now, will be displayed on the surface of the dendritic cell.

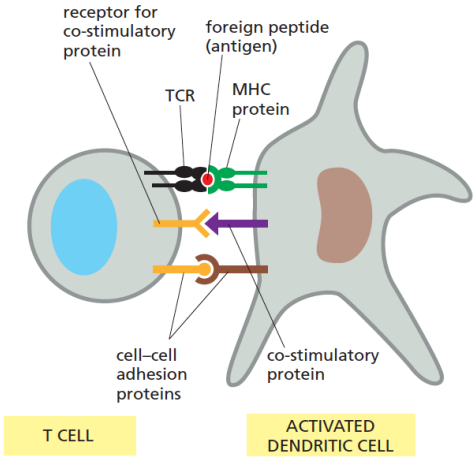

Figure 5: TCR-pMHC Binding.

This displayed complex will be recognized by a T cell through its TCR. Binding of the TCR on this pMHC complex will cause the infected cell to die. As explained earlier, there are two types of MHCs: class I and class II. Although they share the same working mechanism, the origin of the immune response differs:

- Class I: the foreign protein is being produced inside the infected cell itself. For example, the invader in this case can be a virus. Once inside the cell, it will use the transcription - translation tools to produce its viral proteins. Those will be recognized by the class I MHC protein, causing the pMHC formation. This complex will then be transported to the cell membrane, to be displayed to a TCR.
- Class II: the foreign protein itself is now the invader. Once this foreign protein has entered the host cell, the class II MHC protein will recognize it, causing then again the pMHC binding and display on the surface.

As it is clear, fundamentally, one has a protein - protein - protein binding. Proteins can be regarded as a long chain of amino acids. As such, the types of amino acids present in the cell, as well as their order with respect to each other inside the same chain, and with respect to each other on different chains, will highly influence the reaction. For some sequences of amino acids, there will be an immune response (strong binding between TCR and pMHC). However, for some sequences, this binding will be weak, or even impossible. This leads to the research questions, discussed in the next section.

3 Research Questions

As discussed earlier, the focus of this project will be on T cell mediated responses. T cells will bind to infected cells, through a TCR-pMHC complex. The TCR, however, only gets in contact with the foreign protein and not with the MHC. As such, the very starting point of this project are the sequences of the TCR and the (for example, viral) protein. The first research question is then:

Given the sequence of the TCR and given the sequence of the foreign protein (epitope), what is the probability of having a binding?

This first research question is called the probability prediction (PP) task. Using the sequence of the two components, the probabilities $P(\texttt{Binding})$ and $P(\texttt{No Binding})$ will be determined. The second research question is:

Given the sequence of the epitope, what should the sequence of the TCR be in order to have an immune response?

This is called the sequence prediction (SP) task. Given the sequence of the epitope, the task is to determine the sequence of the TCR such that $P(\texttt{Binding})$ is high.

4 Data

The data set used is the VDJ data base (`https://vdjdb.cdr3.net/`), which is a publicely available data base. It contains TCR-epitope pairs ever observered, based on published results of TCR specificity assays and personal communications and submissions. The TCR component of the pair is actually only the TCR-β chain. For some pairs, both the α and β chains are available. However, for most pairs, only information about the β chain is available. In this section, two important aspects of the data will be discussed:

- Data exploration. What information do the raw data give? How many couples are there? What is the average number a epitopes that a TCR binds to and vice versa? ...
- Graphical assessment of couple clusters. Are there some clusters in the couples? In particular, a graphical assessment will be done to check if there are epitopes and TCRs that are likely to interact with each other.

This is an important aspect of the project: it is the very first step to get insight in couples and interaction among the different entities (TCRs and epitopes).

4.1 Data Expolaration

The data was downloaded in March, 2022. This raw data can also be viewed on the website of the VDJ data base. There were no missing values and the only cleaning procedure was removing the redundancies in the data. The raw data contain 80717 observations. After removing the redundancies, 65283 observations were left. Table 1 gives a summary of the cleaned data.

Table 1: Summary of the Cleaned Data.

Number of Observations	65283
Number of Unique Couples	38819
Number of TCR-αs	23392
Number of TCR-βs	36248
Number of Epitopes	461
Mean Number of Epitopes a TCR-β binds to	1
Mean Number of TCR-βs an Epitope binds to	85

Based on this simple summary, some important conclusions can be made about the nature of the data set. First of all, one epitope can bind with multiple TCRs and one TCR can bind to multiple epitopes. However, there is some imbalance in this feature: the mean number of TCRs an epitope binds to is way higher than the mean number of epitopes a TCR binds to. Also, because of the fact that not for all couples both chains are present, in all the models only the TCR-β will be used, in order to have consistency for each couple. The affinity of the binding between a TCR and an epitope is governed by both the α and β chains (Krogsgaard et al., 2005). However, using only the β chain can still lead to high accuracy, but including the α chain will nothing but just improve the results (Springer et al., 2020).

4.2 Graphical Assessment of Couple Clusters

In Figure 6 (left), a graph is given for the couples in the cleaned data set (only the TCR-β). As one can notice, there are groups: some are large, some are small. Also note here that 1 group is just 1 epitope surrounded by the TCRs the epitope binds to. In Figure 6 (right), a more simplified version is displayed (both graphs are generated by `Gephi`): the nodes are representing only the epitopes and an edge between two epitopes is drawn if and only if they share at least one common TCR. More formally, suppose X and Y are epitopes. Let the sets $\mathcal{S}_X$ and $\mathcal{S}_Y$ be the sets of TCR the two epitopes bind to, respectively. If $\mathcal{S}_X \cap \mathcal{S}_Y \neq \emptyset$, the an edge $\mathcal{E}$ is drawn to connect X and Y. This is done for each possible pair of epitopes. In Figure 7, a zoomed version of Figure 6 (left) is given.

Based on Figure 6 and Figure 7, it is clear that some groups form cluster. In particular, there are groups connected with each other by means of a common TCR, forming a cluster. Note again that 1 group is an epitope and collection of TCRs that this epitope binds to. A cluster is meant to be a collection of epitopes sharing common TCRs. This implies an important feature, namely the following. Both TCRs and epitopes are nothing but just proteins. Proteins are characterized by their amino acids. The amino acid sequence making up the protein is key to its chemical and physical properties. The fact that there are clusters means that some of these proteins share common or at least related chemical and physical features, making them being identical or similar from a chemical and physical point of view. This gives rise to feature engineering, discussed in §5.

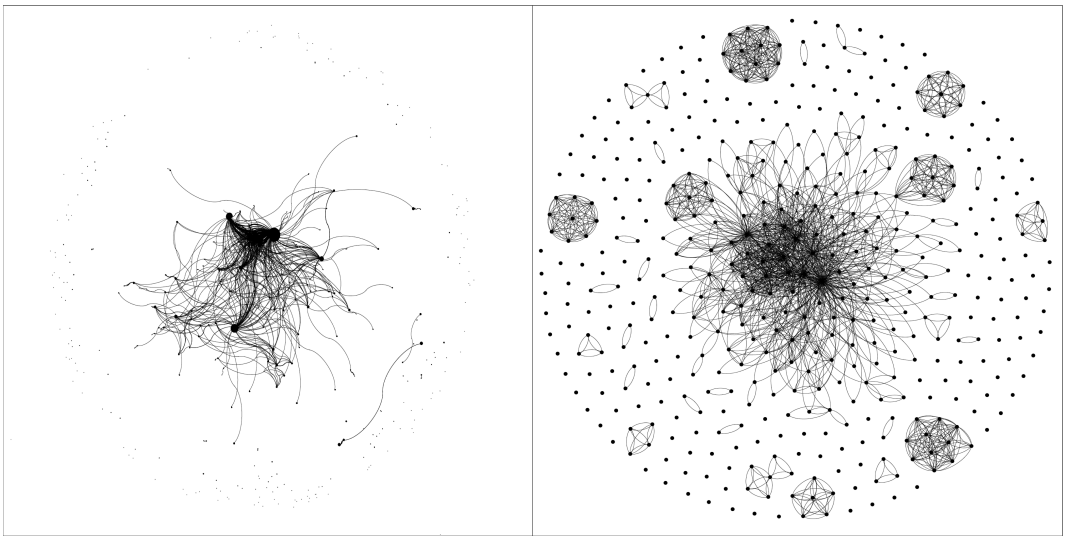

Figure 6: Graphical Display of Clusters.

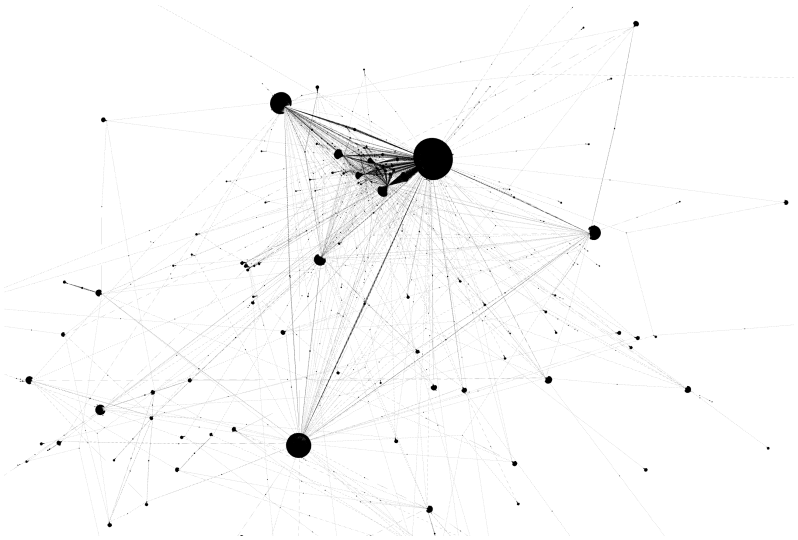

Figure 7: Zoom into the Clusters.

5 Methodology

Like mentioned earlier, the project consists of two main tasks. The first task - being PP - is the prediction of the probability of a binding, given the sequences of the TCR-β and the epitope. The second task - SP - is predicting the sequence of the TCR-β, given the sequence of the epitope. For the first task, both machine learning and deep learning methods are used, whereas for the second task, only machine learning is considered. An overview of the used methods and models is given in Table 2.

Table 2: Summary of the used Methods and Models.

Probability Prediction Task	**Models/Methods**
Machine Learning	Tree, Random Forest, Boosting, Logistic Regression, KNN
Deep Learning	Densely Connected NN, Convolutional NN
Sequence Prediction Task	**Models/Methods**
Machine Learning	Random Forest

For both machine learning and deep learning, the important factor in the process of the building such models is the way one defines the inputs. It is important from both a statistical and biological point of view to have enough features and relevant features. That's why this is the first important task that was considered in this project: feature engineering.

5.1 Feature Engineering

The very beginning inputs are the amino acid sequences of the TCR-β and the epitope. Those are proteins highly characterized by their amino acid sequences. This also implies that their sequences will have an dramatically large effect on the way they bind with each other. Amino acid sequences will also have an effect on the chemical and physical nature of these molecules. In `R`, the software used to do all the analyses, the package `Peptide` (Osorio et al., 2021) contains functions, which take as input an amino acid sequence, and output a chemical or physical feature. For both the TCR-β and epitope, the following features are engineered: length of the protein, charge of the protein, instability index of the protein, hydrophobicity of the protein, molecular mass of the protein, Boman index of the protein and the aliphatic index of the protein. Those are shortly discussed below.

Length

The length of a protein is the number of amino acids that make up that protein. A protein with sequence $X_1X_2...X_n$ with each X_i one of the 20 amino acids, has length n. In `R` this is easily obtained by using the `nchar()` function, which actually takes as input a string and outputs the length of the input string.

Charge

To calculate the charge, the function `charge()` is used from the `Peptide` package. The input is the amino acid sequence of the protein as a string. The net charge is then computed based on the Henderson-Hasselbalch equation (Moore, 1985). Note also the returned charge is the one assuming pH = 7.

Instability Index

This is an index indicating the stability of a protein. The function `instaIndex()` from the `Peptide` package is available to calculate this. A protein with an II smaller then 40 is considered as stable. An II larger than 40 might be an indication that the protein is unstable (Guruprasad et al., 1990).

Hydrophobicity

Using the `hydrophobicity()` function, one can also compute the GRAVY hydrophobicity index

of the protein. The hydrophobicity of a protein is an important factor in the stabilization of the protein folding. In order to have a function protein, the folding process should pass correctly: structure is key in the functioning of a protein.

Molecular Mass

Molecular mass is computed manually using the masses of individual amino acids. However, note that after a binding between two amino acids, one molecule of water, H_2O is released. This implies that if one wants to calculate the molecular mass of a protein sequence of n amino acids, then the masses of these n amino acids need to be added. At the end, $n-1$ times the molecular mass of water, which is 18 units, need to be substracted to correct for its $n-1$ times releases.

Boman Index

This is another protein index: the function `boman()` is computing this index. The return value is an indication for the potential interaction the protein can undergo with other proteins. In particular, a Boman index above 2.48 indicates a high binding potentian to other proteins (Boman, 2003).

Aliphatic Index

This is calculated with the `aIndex()` function. It is the relative value occupied by aliphatic side chains (alanine, valine, isoleucine and leucine). It can be regarded as a positive factor for the increase of the thermostability of globular proteins (Ikai, 1980).

After having discussed the relevant engineered features, there are still two important aspects to clearify: generating of negative samples and the split of training and test sets.

5.2 Generation of Negative Samples

The data base used in this project contains TCR-epitope couples that were ever observed in nature. So all samples in this data base are positive samples. However, the purpose of the models built is to predict the probability of observing a binding and not observing a binding, given the sequence (and as such, the chemical and physical features) of the TCR-β and epitope. There are no negative samples in the data base but those are needed to build the classifiers. This is solved by a method where random couples are generated. If these randomly generated couples are not in the data base, then they are considered as negative samples. Figure 8 gives a schematic overview.

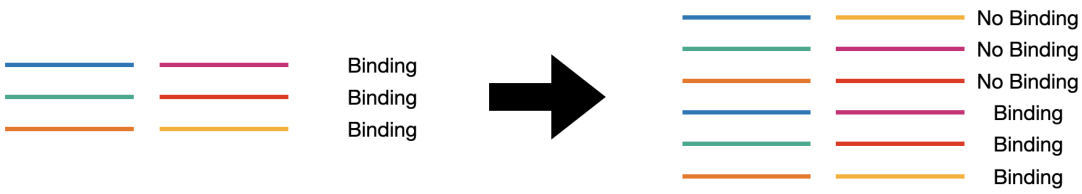

Figure 8: Generation of Negative Samples.

The reasoning behind this method is rather biological: the binding between a TCR and epitope - or in general, between two proteins - has a very high specificity. Besides this, nature has a very high number of TCRs and epitopes. This implies that the probability of a binding between a randomly chosen TCR and randomly chosen epitope is very small, which allows one to consider it as a negative sample (Weber et al., 2021, Moris et al., 2020, Fischer et al., 2020).

After generation of the negative samples, 38819 positive samples and 39850 negative samples were present in the (final) data.

5.3 Splitting the Train and Test Sets

In general machine learning problems, a model M characterized by parameters $(\beta_0, ..., \beta_k)$ is built such that it has a high predictive power. This means that the model should be built such that it does not only well on the data used to build the model (training data set) but also well on a data set never seen before (test data set). Therefore, the data set at hand should be splitted in two disjoint parts: training set and test set. Let the data set at hand be $\mathcal{Z}$ and let $\mathcal{N}$ and $\mathcal{M}$ be the training and test set, respectively. Then:

- $\mathcal{Z} = \mathcal{N} \cup \mathcal{M}$
- $\mathcal{N} \cap \mathcal{M} = \emptyset$

Then model $M = (\beta_0, ..., \beta_k)$ is built using $\mathcal{N}$ obtaining $\hat{M} = (\hat{\beta}_1, ..., \hat{\beta}_k)$ and the predictive power of $\hat{M}$ is evaluated comparing the predictions for $\mathcal{M}$ with the true values of $\mathcal{M}$. However, for this project, the samples are characterized in a different way. The samples are couples $Z = XY$ where Z is the couple, X is the TCR-β and Y is the epitope. For this type of samples, it might be the case that a couple indeed does not occur in the training set if it is a part of the test set. However, the entities (TCR-βs and epitopes) might occur again in the training set, but forming couples with different entities (epitopes and TCR-βs). So the splitting of training set $\mathcal{N}$ and test set $\mathcal{M}$ can happen in different ways. The following splittings are considered.

Let again the total data set be $\mathcal{Z}$ and $\mathcal{N}$ and $\mathcal{M}$ be the training and test set, respectively. A sample in $\mathcal{Z}$ can be written as $Z_{ij} = X_i Y_j$ where Z_{ij} is the couple, X_i the TCR-β and Y_j the epitope. Then, if $Z_{ij} \in \mathcal{M}$ then $Z_{ij} \notin \mathcal{N}$:

- Couple split. Here, it is allowed that the TCR-β and the epitope can still occur in the training set, but with different epitopes and TCR-βs: $X_i Y_k \in \mathcal{N}$ with $k \neq j$ and $X_l Y_j \in \mathcal{N}$ with $l \neq i$.
- TCR split. Here, it is allowed that the epitope occurs in the training set with different TCR-βs, but the TCR-β is not allowed to occur in the training set: $X_i Y_k \notin \mathcal{N}, \forall k$ and $X_l Y_j \in \mathcal{N}$ with $l \neq i$.
- Epitope split. Here, it is allowed that the TCR-β occurs in the training set with different epitopes, but the epitope is not allowed to occur in the training set: $X_i Y_k \in \mathcal{N}$ with $k \neq j$ and $X_l Y_j \notin \mathcal{N}, \forall l$.
- Full split. Both TCR-β and the epitope are not allowed to occur in the training set: $X_i Y_k \notin \mathcal{N}, \forall k$ and $X_l Y_j \notin \mathcal{N}, \forall l$.

In Figure 9, these different splitting methods are summarized.

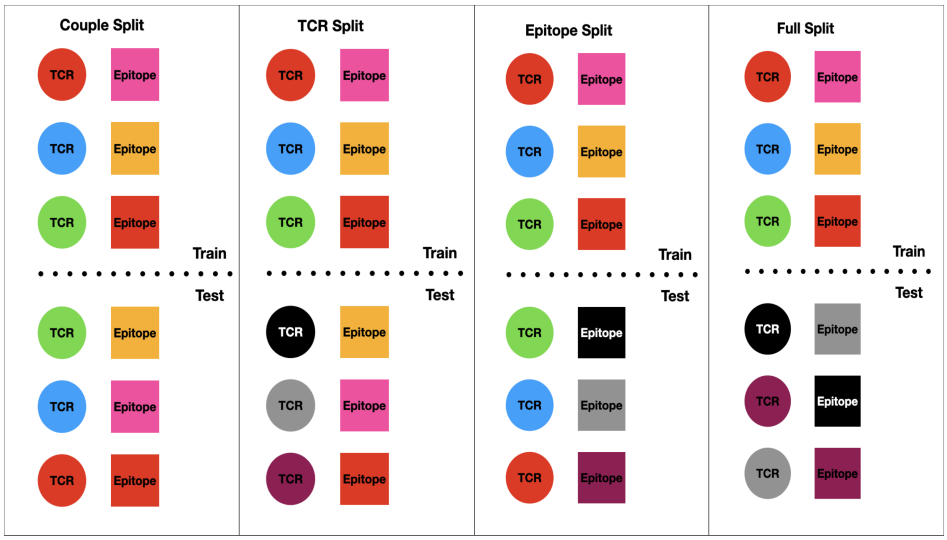

Figure 9: Different Splitting Methods.

Before jumping into the results, let's discuss shortly the models used and the considered metrics to evaluate the models' performance.

5.4 Models and Metrics

The PP task is a classication task: the inputs are the generated features of the TCR-β and the epitope and the output is whether the two entities (TCR-β and epitope) will bind. To be more precise, the output is $P(\texttt{Binding})$ and $P(\texttt{No Binding})$ and the model will predict a positive binding if $P(\texttt{Binding}) > P(\texttt{No Binding})$ and a negative binding if $P(\texttt{Binding}) < P(\texttt{No Binding})$. The metrics to evaluate the PP task are the following:

- Overall accuracy: the proportion of couples assigned to the correct class. In this case, the classes are `Binding` and `No Binding`.
- Sensitivity: the proportion of positive couples that are assigned to the `Binding` class. In particular, it is the probability $P(\texttt{Prediction = Binding} \mid \texttt{True = Binding})$.
- Specificity: the proportion of negative couples that are assigned to the `No Binding` class. In particular, it is the probability $P(\texttt{Prediction = No Binding} \mid \texttt{True = No Binding})$.

Let's now look at the models used in the PP task.

Tree

To be more precise, what is used is rahter a classification tree. The variables to build the classification tree are - like mentioned earlier - the chemical and physical features generated for the TCR-β and epitope. So in total the tree is built using 14 variables. The purpose of a classification tree is to divide the variable space, which is in this case a 14 dimensional space, into J distinct regions $R_1, ..., R_J$. Once these regions are created - such that the training error is minimized - a

test couple can be taken and assigned to a class. Figure 10 gives a very simple example of how a classification tree works.

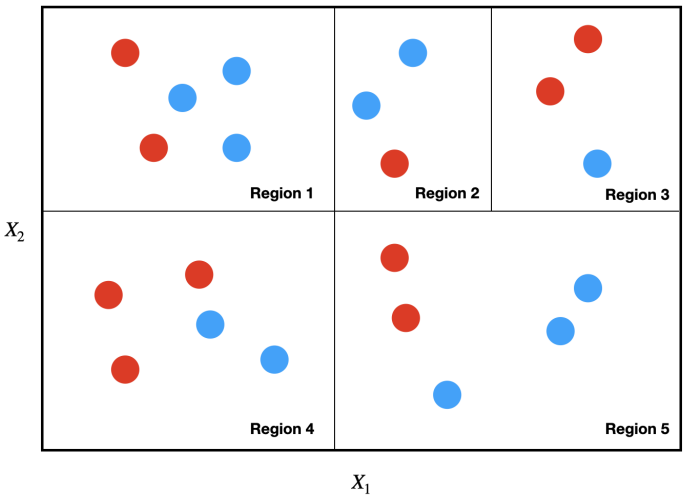

Figure 10: Example of a Random Tree.

As seen in Figure 10, the predictor space is a 2 dimensional space, which is a plane. This plane is divided such that the training classification error rate is minimized. How is a never seen test observation, with features (x_1, x_2) classified to either the class `Blue` or `Red`? What has to be done, is just determining to which region on the plane the vector (x_1, x_2) belongs. In case this vector belongs to `Region 1` then that test observation will be assigned to class `Blue`. If the test observation (x_1, x_2) belongs to `Region 4` then this test observation will be assigned to class `Red`.

Random Forest and Boosting

The same principle of a classification trees applies on random forests and boosting. The difference is now that in stead of building just one classification tree, many trees are built and all of them are used to make the class prediction. At the end, the class that was predicted the most is taken as the class assigned to the test observation. The difference between random forests and boosting is that in random forest, the trees are built independently from each other. In boosting, however, the trees are grown sequentially.

Logistic Regression

Logistic regression is a modelling approach for a binary classification task. Suppose Y is a binary categorical variable (0/1 coding) and let's say there is only one predictor X. What is modelled here is the probability that an observation belongs to class 1:

$$\pi = \beta_0 + \beta_1 \cdot X \tag{1}$$

However, if this model is fitted to a data, then it is possible that for a test observation, a prediction is made with $\pi < 0$ if that observation is likely to belong to class 0 or $\pi > 1$ if that observation

is likely to belong to class 1. This is a mathematical issue since probabilities are supposed to be nonnegative and within the range of [0, 1]. So equation (1) needs to be reformulated such that for each prediction, it is assured that the predicted probability is inside [0, 1]:

$$\pi = \frac{\exp(\beta_0 + \beta_1 \cdot X)}{1 + \exp(\beta_0 + \beta_1 \cdot X)} \tag{2}$$

Manipulating further gives:

$$\begin{aligned} \frac{\pi}{1-\pi} &= \exp(\beta_0 + \beta_1 \cdot X) \\ \ln\left(\frac{\pi}{1-\pi}\right) &= \beta_0 + \beta_1 \cdot X \end{aligned} \tag{3}$$

It is the last equation of (3) that is fitted on the data. The coefficients (β_0, β_1) are estimated using the maximum likelihood estimating method. Once those are obtained, $(\hat{\beta}_0, \hat{\beta}_1)$, a prediction can be made for a test observation x_T. Using equation (3) (last equation), with estimates $(\hat{\beta}_0, \hat{\beta}_1)$ and $X = x_T$, the probability $\hat{\pi}$ is estimated. If $\hat{\pi} > 0.5$ then the test observation with feature x_T will be assigned to class 1.

K Nearest Neighbours

KNN is the most basic binary classification approach. Unlike other statistical and machine learning models, in KNN, no model is actually trained. There are no model parameters needed to be estimated from the data. In fact, it is about remembering the data and placing a test observation in the data space. After that, K nearest neighbours are examined and the class with the highest occurence is assigned to this test observation. An example is given in Figure 11. The training data is plotted, for example, Y against X or X_2 against X_1. As it is clear, the possible classes are `Red` and `Blue`. After the training data is remembered (by plotting), a test observation is placed inside this plot. As example, $K = 5$ is taken. This means that the 5 nearest neighbours of the black test observation are examined. 2 of them belong to `Blue` and 3 of them to `Red`. This implies that the black test observation will be assigned to class `Red`.

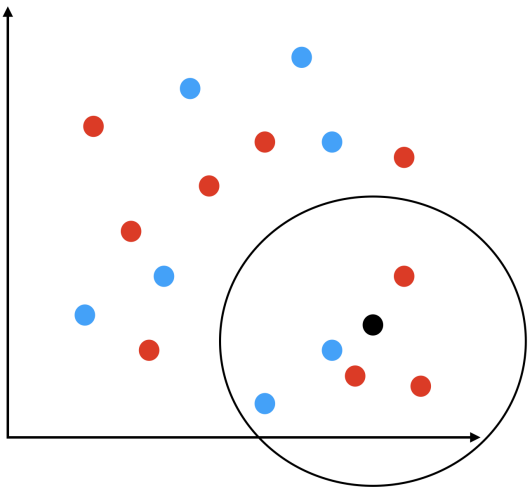

Figure 11: KNN Example.

The larger K is taken, the more observation there are to make the prediction. To be safe, mostly K is taken as an odd number, to prevent ending up with equal number of (training) observations in each of the two classes.

Deep Learning

Deep learning is part of machine learning with a lot of applications in data science, especially in image recognition tasks. Deep learning has the same goal as any other machine learning model, but from the view point of model architecture, deep learning models are fundamentally different. A short overview of deep learning will be given. However, for an extensive deep learning training, *Deep Learning with Python* by François Chollet is highly recommended. To understand how deep learning models work, the best way is to show it with an example.

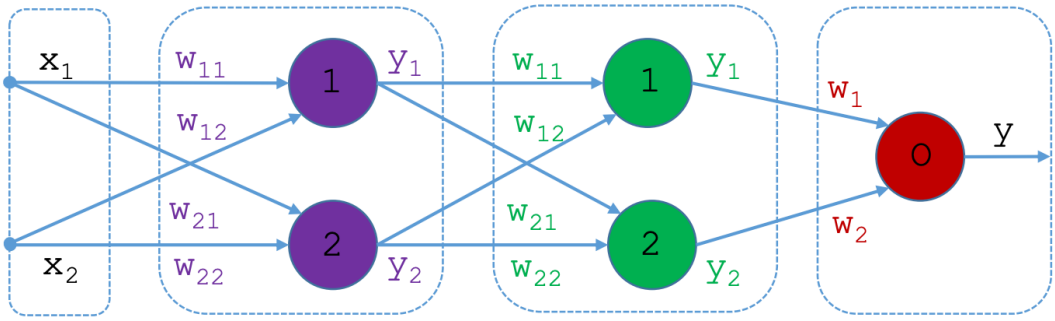

Figure 12: Example of a Deep Learning Model.

In Figure 12, a very simple deep learning model (neural network, NN) is shown. As can be seen, the model consists of 4 compartments, or layers. The very first and very last one is called the input and output layer, respectively. The two layer between them are called hidden layers. Those are characterized by the input they receive and the output they return. That output is generated by the weights and bias terms, which are the intrinsic properties of a layer. The input layer is:

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{4}$$

As such, equation (4) is also the input for the first hidden layer. Based on Figure 12, the output of the first hidden layer and thus the input for the second hidden layer is:

$$\begin{aligned} y_I &= \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ &= \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b_I \end{aligned} \tag{5}$$

Note that $b_I \in \mathbb{R}^2$. The output of the second layer and so the input for the final layer is then:

$$\begin{aligned} y_{II} &= \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ &= \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + b_{II} \end{aligned} \tag{6}$$

And again, $b_{II} \in \mathbb{R}^2$. The output layer returns:

$$y = \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + b_O \tag{7}$$

Here, $b_O \in \mathbb{R}$. This is not necessarely the final step for a deep learnig model. Another important intrinsic property of a NN is the nonlinearity it introduces in the output y. Depending on the task at hand, different nonlinearities are possible. In this context, namely a binary classification task, the appropriate nonlinearity on y is the `sigmoid` nonlinearity:

$$y_f = \sigma(y) = (1 + e^{-y})^{-1} \tag{8}$$

Here, y is as previously shown and y_f the very final output of the neural network. The working principle of the sigmoid nonlinearity can be related to that of the logistic regression: it assigns a probability to both classes and predicts the class for which the probability is the highest.

6 Results

In this section, the results of both the PP and SP tasks are discussed. First, the results of the PP task will be considered. Remember that for the PP task, both machine learning (5) and deep learning models (2) were considered, for each of the 4 possible splits of the data set.

6.1 Probability Prediction

6.1.1 Machine Learning

As mentioned here and also in many other machine learning books, prediction models are all about building a model and testing its performance by applying it on a never seen data. The prediction power of the model on the new data set tells something about its performance. However, it is not always possible to have a test data by hand and as such, the one available to build the model on should be used efficiently: splitting it in a training part whereon the model is built. Once the model is built, it is tested on the other part, called the test set. For each possible split (see Figure 9 again):

- Couple split:
  - Validation: the data is splitted in two (roughly) equally sized parts. On one part, the model is built. On the other part, the model is tested.
  - 5 fold cross validation: the data is splitted in 5 (roughly) equally sized parts. First, the first part is left out and the model is trained on the other for parts. Then the model is tested on the left out part. This is repeated 5 times (each time leaving one part out and training on the 4 remaining parts). And the end, the average of the 5 performances is taken.
- TCR split:
  - Validation: select 100 TCR-βs randomly. Then split the data into two parts: the first part is the one for which the TCR-β is one of the selected 100 TCR-βs and the other part is the one for which this is not the case. Train the model on the latter one and test it on the first one.
  - 5 fold cross validation: repeat the above 5 times, and take the average of the performances.

- Epitope split:
  - Validation: select 50 epitopes randomly. Then split the data into two parts: the first part is the one for which the epitope is one of the selected 50 epitopes and the other part is the one for which this is not the case. Train the model on the latter one and test it on the first one.
  - 5 fold cross validation: repeat the above 5 times, and take the average of the performances.
- Full split:
  - Validation: select 200 TCRs and 200 epitopes randomly. Then split the data into two parts: the first part is the one for which the epitope and/or TCR-β are one of the selected 200 epitopes and/or TCR-βs and the other part is the one for which this is not the case. Train the model on the latter one and test it on the first one.
  - 5 fold cross validation: repeat the above 5 times, and take the average of the performances.

For KNN, the approach is slightly different:

- Couple split: split the data in two (roughly) equally sized parts. One part is the test set. For each element in the test set (each couple, in the context of this project), a prediction (`Binding` or `No Binding`) is made based on the K nearest couples to this element. It is assigned to the class for which more than $\sim K/2$ of the training couples belong to.
- TCR split: select 100 TCR-βs randomly. Then split the data into two parts: the first part is the one for which the TCR-β is one of the selected 100 TCR-βs and the other part is the one for which this is not the case. The first part of the test set. For each element in the test set (each couple, in the context of this project), a prediction (`Binding` or `No Binding`) is made based on the K nearest couples to this element. It is assigned to the class for which more than $\sim K/2$ of the training couples belong to.
- Epitope split: select 100 epitopes randomly. Then split the data into two parts: the first part is the one for which the epitope is one of the selected 100 epitopes and the other part is the one for which this is not the case. The first part of the test set. For each element in the test set (each couple, in the context of this project), a prediction (`Binding` or `No Binding`) is made based on the K nearest couples to this element. It is assigned to the class for which more than $\sim K/2$ of the training couples belong to.
- Full split: select 200 TCR-βs and 200 epitopes randomly. Then split the data into two parts: the first part is the one for which the epitope and/or TCR-β are one of the selected 200 epitopes and/or TCR-βs and the other part is the one for which this is not the case. The first part of the test set. For each element in the test set (each couple, in the context of this project), a prediction (`Binding` or `No Binding`) is made based on the K nearest couples to this element. It is assigned to the class for which more than $\sim K/2$ of the training couples belong to.

KNN is applied for varying K. For each K, KNN is performed 5 times and at the end of the Kth loop, the average of these 5 is taken. Furthermore, for all machine learning models, the inputs are the engineered chemical and physical features, denoted as:

$$X_{11}, ..., X_{17}, X_{21}, ..., X_{27} \tag{9}$$

Here, $X_{11}, ..., X_{17}$ are length, charge, instability index, hydrophobicity, molecular mass, Boman index and aliphatic index of the TCR-β, respectively, and, $X_{21}, ..., X_{27}$ are length, charge, instability

index, hydrophobicity, molecular mass, Boman index and aliphatic index of the epitope, respectively.

Classification Tree

In Figure 13, the results of the classification tree are given. The test approach is the validation approach. For the couple split and TCR split, the results are way better than those of the epitope split and full split.

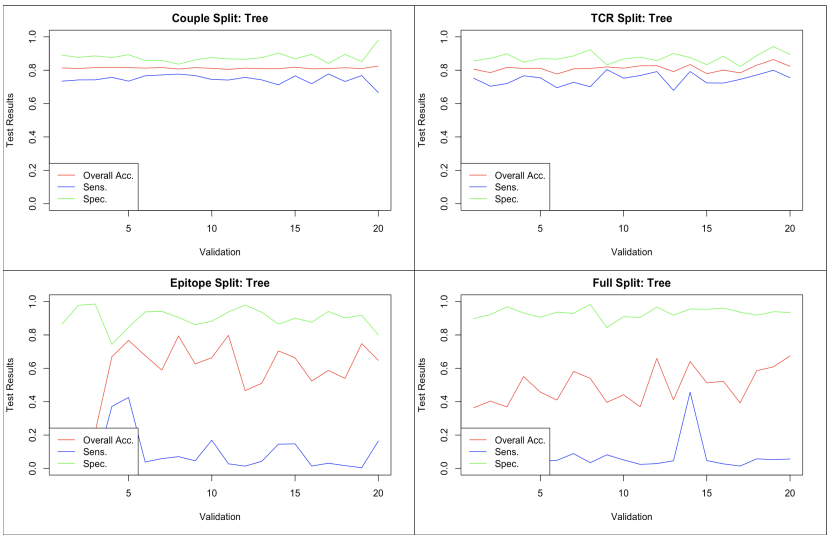

Figure 13: Classification Tree: Validation.

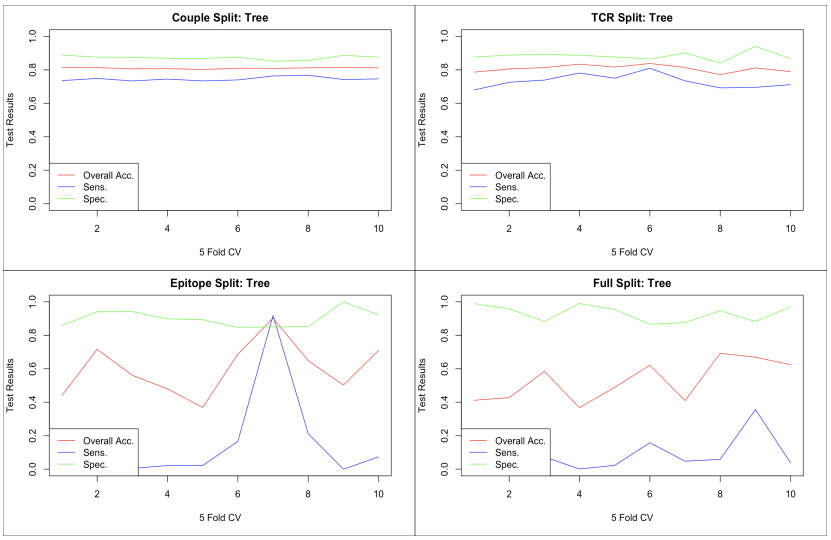

Figure 14: Classification Tree: 5 Fold CV.

Also note that the validation approach was performed 20 times. After each validation, the overall accuracy (red), the sensitivity (blue) and the specificity (green) are computed. Next, Figure 14 gives the result of the classification tree for which a 5 fold cross validation was performed. Compared to the validation approach, the curves are smoother. But the same comments apply also here: the results of the couple split and TCR split are way better than the results of the epitope split and full split.

Random Forest

The same approach is also taken for the random forest model. In Figure 15, the validation results are displayed and in Figure 16, the 5 fold CV results. The same comments also apply here: the couple split and TCR split are doing way better than the epitope split and full split.

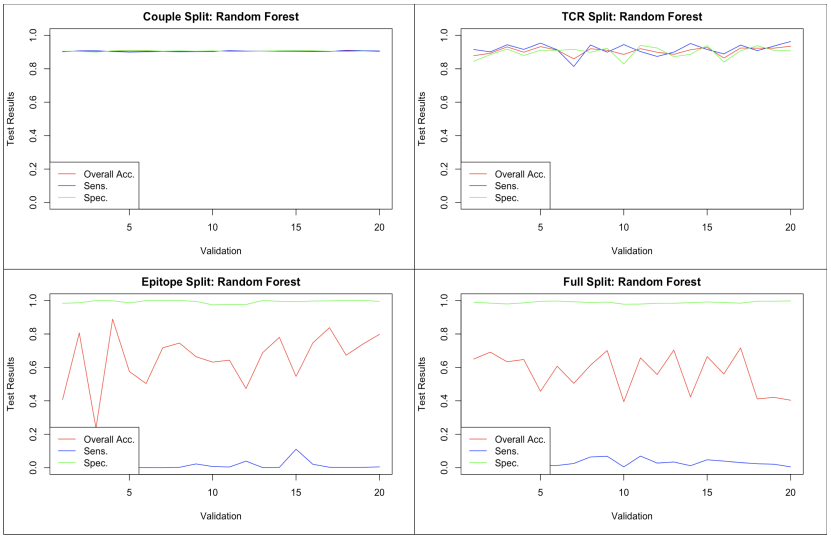

Figure 15: Random Forest: Validation.

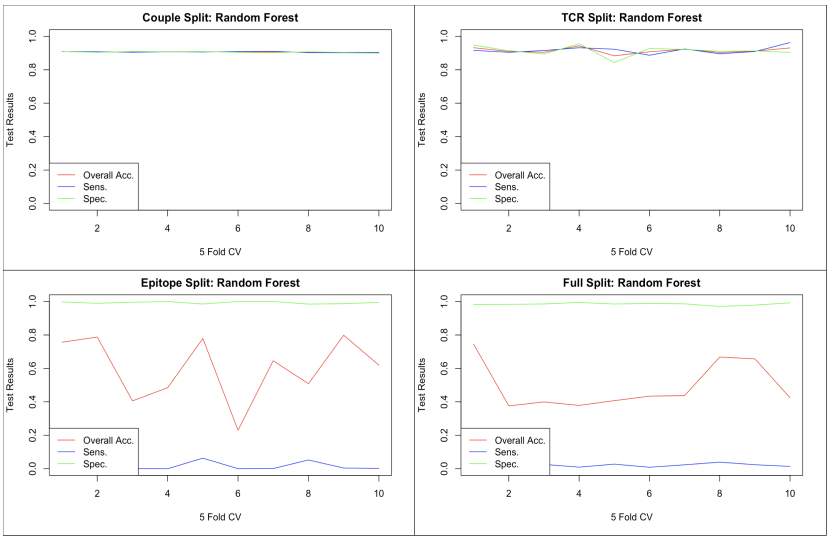

Figure 16: Random Forest: 5 Fold CV.

Note in both the classification tree and the random forest the very low sensitivity and very high specificity in the epitope split and full split. Especially for random forest, it is clear that the models don't do a good job for epitope split and full split: every test couple has a very high probability to be assigned to the `No Binding` class, regardless the true class.

Boosting

Boosting is another tree based method used in this project. Figure 17 gives the validation results and Figure 18 gives the 5 fold CV results. And also here, the same comments made for the previous models apply here. It is also clear that for the epitope split and full split, again, every test couple has a high chance to be assigned to the `No Binding` class, whether or not the couple actually binds.

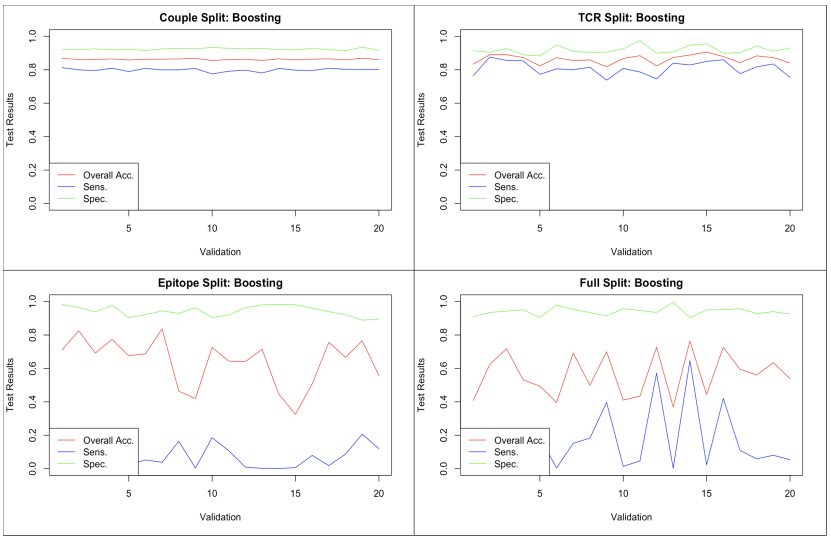

Figure 17: Boosting: Validation.

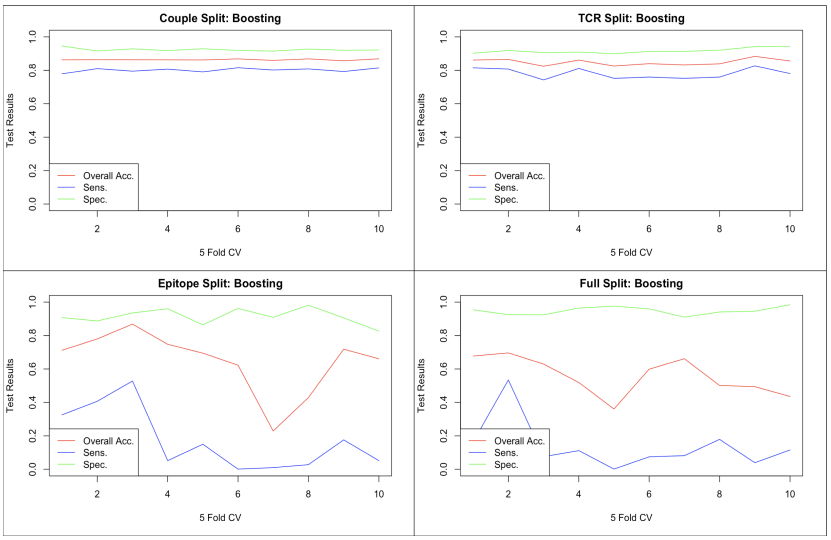

Figure 18: Boosting: 5 Fold CV.

This concludes the tree based models. Next, logistic regression will be discussed.

Logistic Regression

Logistic regression is another classification model, though it is not a tree based method like the three above mentioned methods. In this case, a model is fitted. This means that some parameters will be estimated, using the data. To have an idea about what features are of significance, a logistic regression model is fitted to the whole data base. In particular, the following model is fitted to the

full data:

$$\begin{aligned}\ln\left(\frac{\pi}{1-\pi}\right) &= \beta_0 + \beta_1 \cdot X_{11} + \beta_2 \cdot X_{12} + \beta_3 \cdot X_{13} + \beta_4 \cdot X_{14} \\ &+ \beta_5 \cdot X_{15} + \beta_6 \cdot X_{16} + \beta_7 \cdot X_{17} + \beta_8 \cdot X_{21} \\ &+ \beta_9 \cdot X_{22} + \beta_{10} \cdot X_{23} + \beta_{11} \cdot X_{24} + \beta_{12} \cdot X_{25} \\ &+ \beta_{13} \cdot X_{26} + \beta_{14} \cdot X_{27}\end{aligned} \tag{10}$$

With X_{ij} like in equation (9). The parameter estimates are displayed in Table 3.

Table 3: Logistic Regression on Full Data.

Coefficient	Estimate	Standard Error	z value	$P(Z > \|z\|)$
β_0	2.244	$9.473 \cdot 10^{-2}$	23.684	$< 2 \cdot 10^{-16}$
β_1	$7.592 \cdot 10^{-3}$	$1.257 \cdot 10^{-2}$	0.604	0.54576
β_2	$2.136 \cdot 10^{-2}$	$8.309 \cdot 10^{-3}$	2.570	0.01016
β_3	$9.784 \cdot 10^{-5}$	$3.297 \cdot 10^{-4}$	0.297	0.76662
β_4	$-8.646 \cdot 10^{-2}$	$4.238 \cdot 10^{-2}$	-2.040	0.04135
β_5	$-3.290 \cdot 10^{-4}$	$1.210 \cdot 10^{-4}$	-2.718	0.00657
β_6	$2.321 \cdot 10^{-2}$	$1.856 \cdot 10^{-2}$	1.251	0.21103
β_7	$1.686 \cdot 10^{-3}$	$6.033 \cdot 10^{-4}$	2.795	0.00519
β_8	$5.378 \cdot 10^{-1}$	$1.110 \cdot 10^{-2}$	48.466	$< 2 \cdot 10^{-16}$
β_9	$5.028 \cdot 10^{-1}$	$7.131 \cdot 10^{-3}$	70.514	$< 2 \cdot 10^{-16}$
β_{10}	$1.173 \cdot 10^{-3}$	$2.530 \cdot 10^{-4}$	4.638	$3.52 \cdot 10^{-6}$
β_{11}	$1.423 \cdot 10^{-1}$	$2.469 \cdot 10^{-2}$	5.763	$8.24 \cdot 10^{-9}$
β_{12}	$-7.286 \cdot 10^{-3}$	$1.000 \cdot 10^{-4}$	-72.835	$< 2 \cdot 10^{-16}$
β_{13}	$5.894 \cdot 10^{-2}$	$1.270 \cdot 10^{-2}$	4.641	$3.47 \cdot 10^{-6}$
β_{14}	$3.116 \cdot 10^{-3}$	$3.012 \cdot 10^{-4}$	10.344	$< 2 \cdot 10^{-16}$

It is not of interest to do a model selection procedure and so on. The model in equation (10) is fitted to the whole data just out of interest. However, Table 3 does contain some important messages:

- The probability of having binding between the TCR-β and epitope is mainly dominated by the epitope: all the parameter estimates $\beta_8, ..., \beta_{14}$ are highly significant.
- The features of the TCR-β that effects the binding significantly are the charge, hydrophobicity, the molecular mass and the aliphatic index of the TCR-β: β_2, β_4, β_5 and β_7 respectively.

This is an important notification from both a statistical and biological point of view. The binding between a TCR-β and an epitope is mainly determined by the chemical and physical features of the epitope.

In Figure 19 and Figure 20, like the previous models, the validation and 5 fold CV results are given, respectively. Also here the couple split and TCR split are doing better than the epitope split and full split. However, the epitope and full split are doing better than the epitope and full split of the previous models, especially when one is looking at the sensitivity of the epitope and full split.

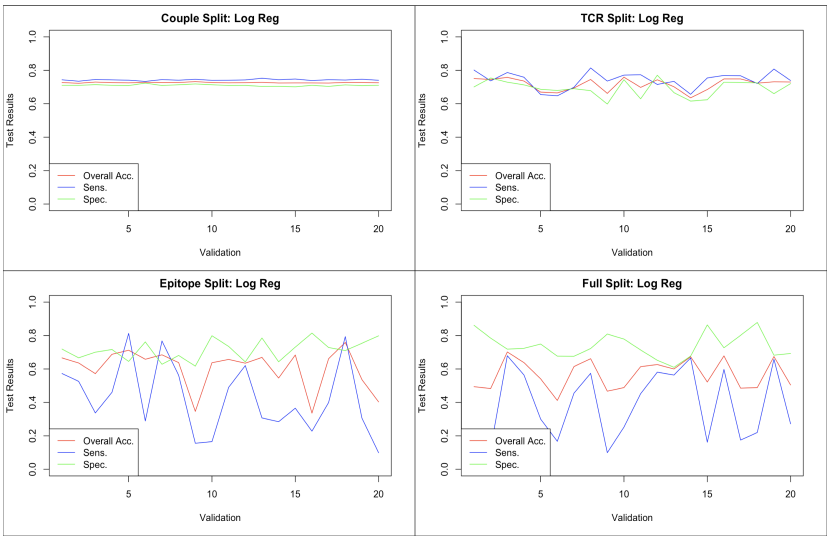

Figure 19: Logistic Regression: Validation.

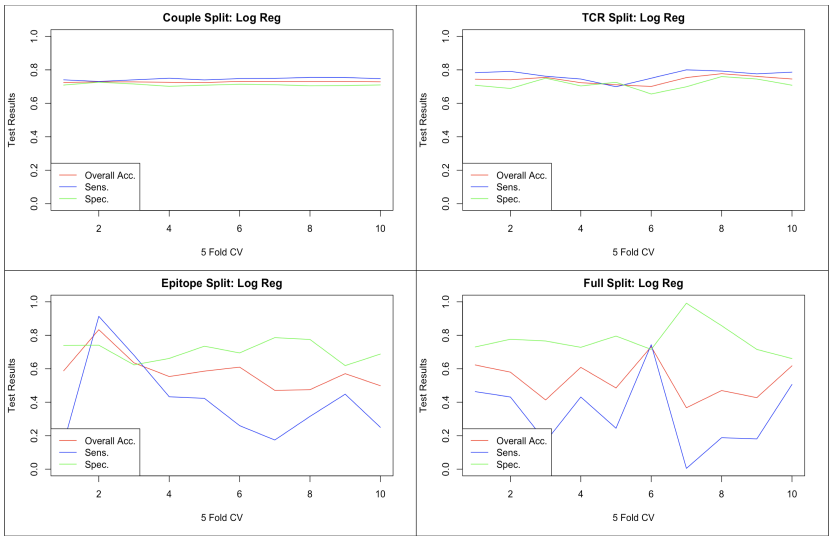

Figure 20: Logistic Regression: 5 Fold CV.

This concludes actually the last model. Next, KNN is discussed. But like mentioned earlier, in KNN, actually there is no model that is trained. In particular, there are no parameters estimated using a training set and tested on a test set. It is just about remembering the data and using that to do predictions for test observations.

KNN

In Figure 21, the results of KNN are displayed. Unlike the previous machine learning models,

here there are some differences in the results. First of all, like earlier, the couple split and TCR split are doing way better. However:

- For couple split and TCR split, the higher K the worse the test results.
- For epitope split, the higher K the better the test results.
- For full split, K hasn't a large effect on the results at all.

Because it was observed in Figure 21 that K had a positive effect on the test results of the epitope split, K was enlargened further till $K = 490$. These results are given in Figure 22.

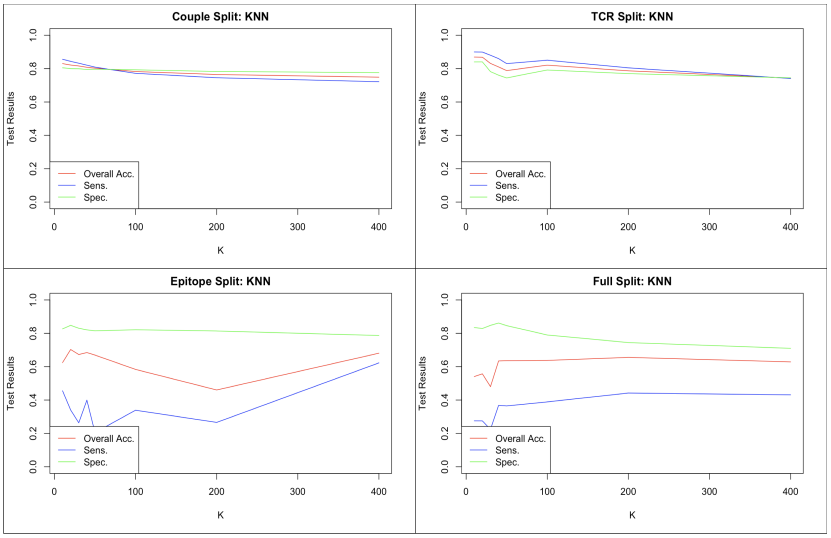

Figure 21: KNN.

In Figure 22, it can be seen that K has a positive effect further till $K = 460$. After this value, the results aren't changing that much. The irony is clear here: till now, all the models were doing worse for epitope split. However, KNN, which is a very naive and simple machine learning tool, beats the previous models for epitope split.

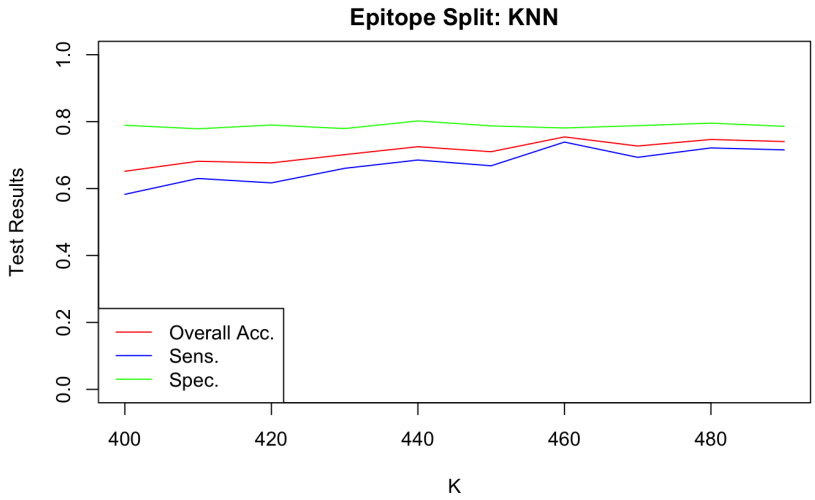

Figure 22: KNN for Epitope Split: $K > 400$.

This also concludes the machine learning models for the PP task. To wrap up, it was all about splitting the data, building a model using one part of it (training set) and testing it on the other part of it (test set). For each model, each split, validation was performed 20 times and 5 fold CV was performed 10 times. It is a good place here to summarize all these results. This is done in Table 4. For each approach (validation or 5 fold CV) for each split, the mean across the 20 validations or 10 5 fold CV is computed. Except for KNN: here, the best result is given with the associated K value. For example, for TCR split, the best overall accuracy was reached at $K = 10$ with a value of 0.86944494, the best sensitivity at $K = 10$ with a value of 0.8998198 and the best specificity at $K = 20$ with a value of 0.8406022.

Table 4: Overall Summary of the Machine Learning Models.

Classification Tree: Validation	Overall Accuracy	Sensitivity	Specificity
Couple Split	0.8150482	0.7298446	0.8979584
TCR Split	0.8152281	0.7473084	0.8817116
Epitope Split	0.544526	0.07850809	0.9061205
Full Split	0.5118133	0.1169892	0.9227894
Classification Tree: 5 Fold CV	Overall Accuracy	Sensitivity	Specificity
Couple Split	0.8135062	0.7289017	0.896371
TCR Split	0.7995548	0.7318406	0.8667098
Epitope Split	0.5844105	0.06040861	0.9253473
Full Split	0.5225675	0.1274053	0.9020948
Random Forest: Validation	Overall Accuracy	Sensitivity	Specificity
Couple Split	0.9053622	0.9045359	0.9061675
TCR Split	0.9135992	0.9142624	0.9134359
Epitope Split	0.6105673	0.03103918	0.9916374
Full Split	0.5148326	0.02801767	0.9896988
Random Forest: 5 Fold CV	Overall Accuracy	Sensitivity	Specificity
Couple Split	0.9076062	0.9086823	0.9065563
TCR Split	0.8941355	0.8922518	0.8971301
Epitope Split	0.6826154	0.02892883	0.9933588
Full Split	0.5161582	0.02578254	0.9876092
Boosting: Validation	Overall Accuracy	Sensitivity	Specificity
Couple Split	0.8616152	0.801117	0.9204967
TCR Split	0.8550022	0.789033	0.9197636
Epitope Split	0.6231994	0.1085707	0.9371842
Full Split	0.5522293	0.1589395	0.9428394
Boosting: 5 Fold CV	Overall Accuracy	Sensitivity	Specificity
Couple Split	0.8612563	0.7982842	0.9224232
TCR Split	0.8738907	0.8176934	0.9316371
Epitope Split	0.6595764	0.1211394	0.9348453
Full Split	0.5658854	0.0909858	0.9388287
Logistic Regression: Validation	Overall Accuracy	Sensitivity	Specificity
Couple Split	0.7264165	0.7428696	0.7104416
TCR Split	0.716704	0.7423419	0.6921754
Epitope Split	0.6062556	0.4271195	0.7138515
Full Split	0.5683752	0.3842602	0.7399126
Logistic Regression: 5 Fold CV	Overall Accuracy	Sensitivity	Specificity
Couple Split	0.7277538	0.745389	0.7105909
TCR Split	0.7411175	0.7684695	0.7143444
Epitope Split	0.5822667	0.4056686	0.7065091
Full Split	0.5325604	0.336069	0.7739058
KNN	Overall Accuracy	Sensitivity	Specificity
Couple Split	0.8302258 ($K = 10$)	0.8545392 ($K = 10$)	0.8064460 ($K = 10$)
TCR Split	0.8694494 ($K = 10$)	0.8998198 ($K = 10$)	0.8406022 ($K = 20$)
Epitope Split (1)	0.6820125 ($K = 40$)	0.6820125 ($K = 400$)	0.8250659 ($K = 30$)
Epitope Split (2)	0.7813236 ($K = 470$)	0.7780726 ($K = 470$)	0.8018179 ($K = 460$)
Full Split	0.6911866 ($K = 100$)	0.4889432 ($K = 100$)	0.8622801 ($K = 50$)

Next, the results of deep learning models for the PP task will be discussed.

6.1.2 Deep Learning

For the PP task, also deep learning models were considered. The aim is the same as previous models. The data is divided according to the split of interest. A neural network is trained on the training set and this NN is then tested on the other set. So the output is the same: class prediction, which is either `Binding` or `No Binding`. However, for deep learning, the input differs. For machine learning, the input were the engineered features of both the TCR and epitope. Here, the input is the amino acid sequence of both the TCR and epitope, encoded to a numerical representation according to the function:

$$\mathcal{A} \mapsto \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\} \tag{11}$$

With $\mathcal{A} = \{$A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V$\}$, the collection of all possible amino acids. An important note to be made is the following. The length of the TCR-βs and epitopes changes through the data, but the maximal length of an epitope and TCR-β is 20 and 38, respectively. For deep learning models, it is of importance that all the couples have the same length. Because of this all epitopes are translated to a numerical representation (vector) with a length of 20 and, for example, if the epitope has a length of 14, then till position 14 the function in equation (11) will be used. After that, the other remaining position will be filled with 0s, till a length of 20. The same is done for a TCR-β, but a length of 38. So the smart reader will conclude now that the input is a vector of length 58. For each split, two models are being considered. First, a densely connected neural network and after that, a convolutional neural network. Again, for an extensive description of the difference, the book of mr. Chollet is highly recommended. Here, rather a short description will be given. Deep learning models are built using `Tensorflow` and `Keras` in `Python`.

A densely connected neural network will be trained such that it will be looking at patterns in the input that are rather global. Note that the input is actually a translated amino acid sequence. While a densely connected neural network is being trained, it will look at the sequence each time and search for global patterns. A convolutional neural network is rather looking for local patterns. For example, it will look for whether there are group of amino acids emerging in specific positions inside the amino acid chain. Before giving the test results of the neural networks for each split, Figure 23 and Figure 24 are summarizing the model architectures of the densely connected network and the convolutional neural network, respectively.

The densely connected neural network is the one looking for global patterns in the amino acid sequences of both the TCR-β and the epitope. As it is clear from the figure, the architecture is such that each time the units in the layers are increasing. Just like in machine learning, where the model performance is being improved when more covariates are added, for a neural network, the model performance can also be improved by adding more and more layers and introducing more and more units inside the layers.

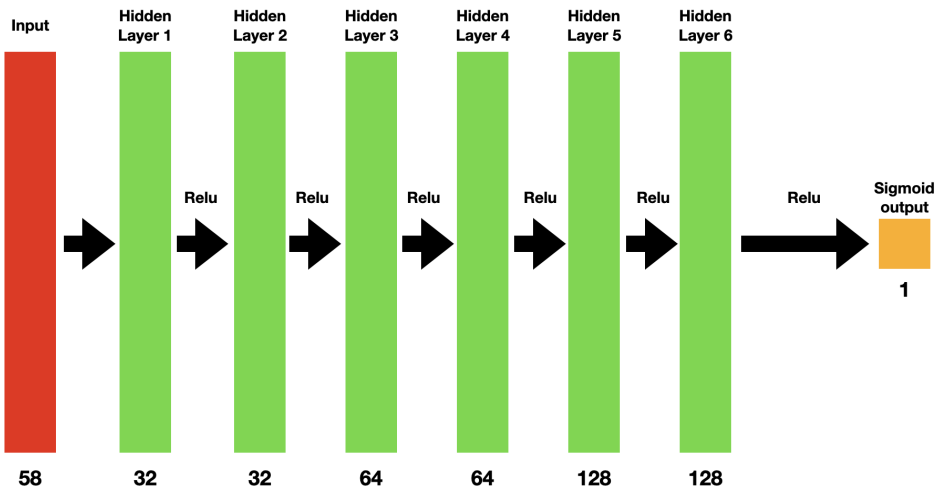

Figure 23: Densely Connected Neural Network.

In Figure 24, where the convolutional neural network is summarized, a lot of unknown terms might occur. The detailed description will be skipped over, but let's go through the architecture and discuss shortly what is happening anyway. When the input is arriving in the `1D Conv.` layer, what is happening is that the network is scanning the input and it looking for emerging local patterns, specific patterns in specific locations inside the input. After that, as it might be clear, the dimension is reduced by a factor of 2. From all the saved patterns in the 56×32 convolution, only the most important ones are kept, giving a dimension of 28×32. This is done further and further until a convolution of 7×128 is reached. This final convolution is vectorized at the end and being connected with the output again, which is the same as the densely connected neural network.

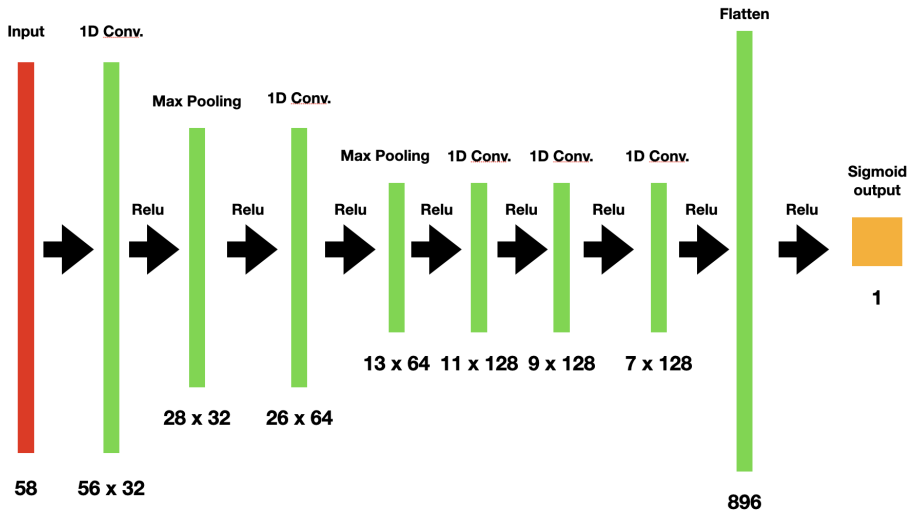

Figure 24: Convolutional Neural Network.

The only aspect not discussed yet is the `relu` operation, standing for rectified linear unit function, which is:

$$\begin{aligned} \text{relu}(x) &= 0 \text{ for } x < 0 \\ &= x \text{ for } x \geq 0 \end{aligned} \tag{12}$$

In Table 5, the test results of the densely connected NN and ConvNet for each split are given. The same overall conclusion can also be made here: the results of the couple split and TCR split are better than the results of the epitope split and full split.

Table 5: Overall Summary of the Deep Learning Models.

Densely Connected NN	Accuracy	Sensitivity	AUC-ROC
Couple Split	0.8674	0.8757	0.9321
TCR Split	0.8647	0.8704	0.9421
Epitope Split	0.6064	0.0520	0.4004
Full Split	0.6792	0.2449	0.5494
ConvNet	Accuracy	Sensitivity	AUC-ROC
Couple Split	0.8722	0.8725	0.9130
TCR Split	0.8889	0.8796	0.9217
Epitope Split	0.6127	0.0818	0.4881
Full Split	0.7008	0.1323	0.5249

In all the machine learning and deep learning models, couple split and TCR split results are way better than the results of the epitope split and full split. In §7, this will be discussed in more detail.

6.2 Sequence Prediction

In this part of the project, called the SP part, given the sequence of the epitope, the appropriate sequence of the TCR-β will be predicted. This is a challenging task in both computer science and biology (targeted immunotherapy): building a (virtual) TCR-β such that this TCR-β will bind to the given epitope. The approach here is highly different than that of the PP task, though the main idea remains the same: a training set is used to build models and these built models are used on a test set. The test set is generated by picking one epitope randomly from the data. The other part remains as a training set and the set with the randomly selected epitope with its TCR-βs is considered as the test set. In particular, it is that randomly selected epitope for which a TCR-β will be built. The inputs in this SP task are both the sequence of the epitopes and the chemical and physical features of them.

In Figure 25, the SP task method is summarized. As mentioned several times throughout this report, the maximal length of the TCR-β in the total data is 38. This means that the random forest models built using the training observations will predict a sequence of length 38. Also, because predicting just one string will have an evaluation metric with only the option `correctly predicted` or `falsely predicted`, rather a matrix will be predicted with the number of rows equal to 21 (20 amino acids and `X`, standing for no amino acid) and number of columns equal to 38, being the position inside the TCR-β chain. Therefore, if this matrix called M, then m_{ij} is standing for the probability to have amino acid i on position j, with the amino acids ordered in a vector like given in the deep learning section, with additional `X` as the 21th element of this vector. Note that one epitope binds with multiple TCR-βs, which means that such a matrix can also be generated using the true TCR-βs of this epitope which it binds to. This will also allow to compare the predicted probability matrix with the true matrix, leading to a metric called the position wise error, which is

defined as the absolute value of the column difference of the predicted matrix and the true matrix. More formally, if M is the predicted matrix and N the true matrix, then jth error rate is defined as:

$$\epsilon_j = \frac{1}{21} \sum_{i=1}^{21} |M_{ij} - N_{ij}| \tag{13}$$

For $j = 1, ..., 38$. As a last comment, the following is very important to note.

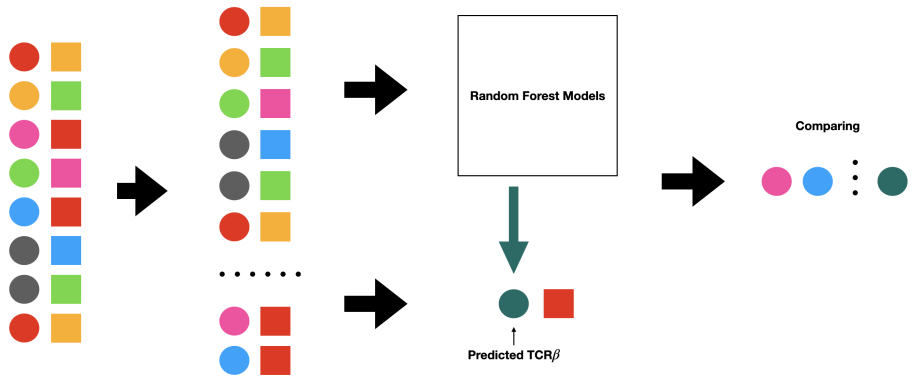

Figure 25: Summary for the SP task.

A random forest model will be built for each position and for each amino acid. In particular, for position j, for each amino acid, a random forest model will be built. Out of that model, the probability of having that amino acid on position j and not having that amino acid on position j will be determined. This is implying that, to build a TCR-β, $21 \times 38 = 798$ random forest models are used! This requires some computational power. However, because of the long run time, 4 randomly selected epitopes will be evaluated. Note that it is not that these 4 epitopes are selected as a test set: these will be selected one by one so for each epitope the training set will contain 460 epitopes, and this set will be used to build a TCR-β for the randomly selected epitope. As a consequence, $\sum_{i=1}^{21} M_{ij} = 1$ is not necessarily true, whereas $\sum_{i=1}^{21} N_{ij} = 1$ is true, for all j.

This method is summarized in Figure 26. The 4 randomly selected epitopes are TTDPSFLGRY, KLFEFLVYGV, LSLRNPILV and KRWIIMGLNK. To compare the probability matrices (predicted vs true), the matrices are converted to a heatmap and these two heatmaps are then compared. This is given in Figure 27. From left to right each time the predicted and true probability matrices are given. From top to bottom, one has the epitope TTDPSFLGRY, KLFEFLVYGV, LSLRNPILV and KRWIIMGLNK. Note that the highest variability in amino acids is seen between positions 2 and 16. This is due to the fact all TCR-βs are starting with the amino acid `C` and only a few TCR-βs have a long length. Comparing the heatmaps based on color and color differences won't be that informative. That's why the position wise error rate is defined as mentioned earlier.

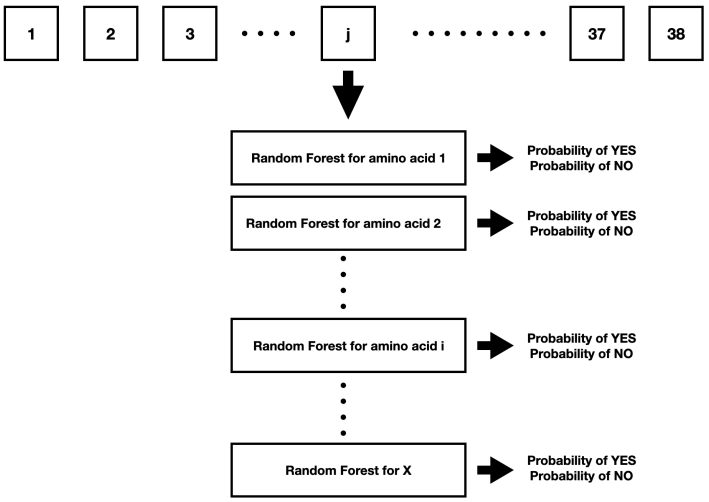

Figure 26: Building a TCR-β using Random Forest Models.

The position wise error rates for the 4 randomly selected epitopes are given in Figure 28. The high variability between the position 2 and 16 is also reflected on this graph. For all the epitopes, the error rate shows an increasing trend starting from position 2 and reaching a maximum value around position 10. After position 10, the error rate is decreasing to a value of 0. Note again that each position wise error is the mean absolute difference of the predicted and true probabilities of having each amino acid at that position.

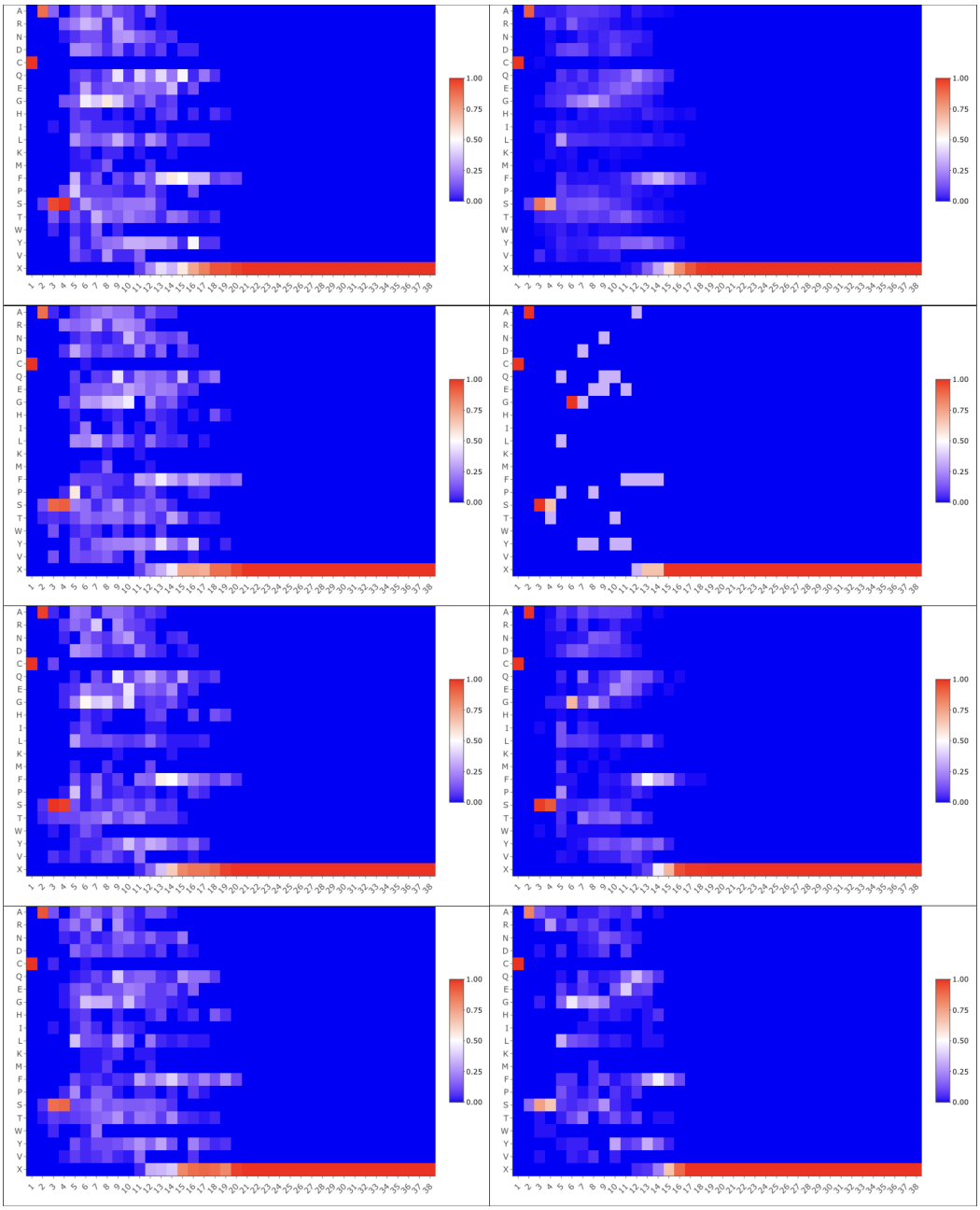

Figure 27: Sequence Prediction: Predicted vs True

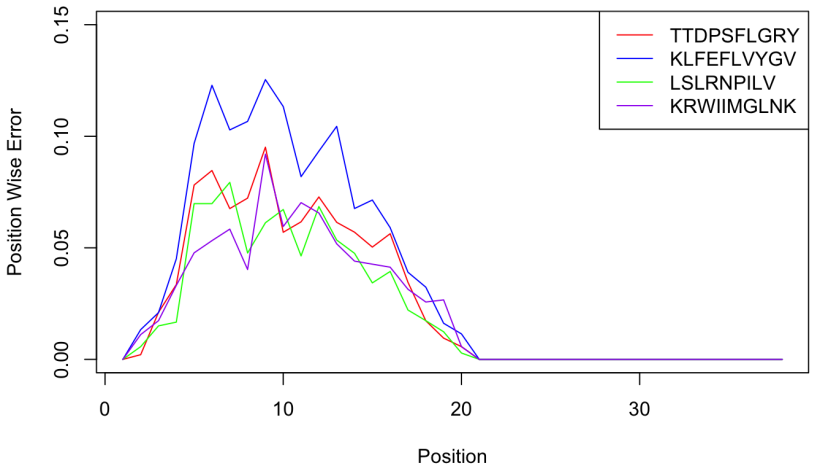

Figure 28: Position Wise Errors for the Epitopes.

Next in §7, the results will be discussed and interpreted furhter.

7 Discussion and Interpretation

7.1 PP Task

In the PP task, the metrics overall accuracy, sensitivity and specificity were used to evaluate the model's performance on the test set. In general, the best test results were obtained for couple split and TCR split, whereas the results for the epitope split and full split were dramatically bad. The following is hypothesized as the reason for this.

In the full data, the number of couples is 78669. From this, 38819 are positive and 39850 are negative. This is no magic at all: 38819 is coming from the cleaned raw data and 39850 are the created negative samples. The mean number of epitopes a TCR binds to and doesn't bind to are both 1. The mean number of TCR-βs an epitope binds to and doesn't bind to is 84 and 86, respectively. Given the TCR-βs or epitopes, no imbalances are seen in positive versus negative samples. Will this be conserved for each split? The danger exists that this balance can be disturbed because of the imbalances between the TCR-βs and epitopes. Check Table 6: this summarizes the discussed ones for each split.

So for the full data, two aspects were noted:

- There is a balance between positive and negative samples (50/50 balance). Let's call this the marginal balance.
- This is conserved for both TCR-βs and epitopes. However, when one compares TCR-βs and epitopes, some imbalance is introduced. So there is also conditional balance (i.e. balance given only TCR-βs or epitopes), but this conditional balances differ.

When looking at the CS (couple split) and TS (TCR split), for both training and test sets, it is seen that the both the marginal balance and conditional balances are preserved. However, for the ES (epitope split) and FS (full split), it is seen that this is disturbed: the marginal balance doesn't exist anymore and the conditional balance for the epitopes is gone too. Through all the splits, the conditional balance of TCR-β is 50/50 whereas for the epitope, this 50/50 dissapears. It is this imbalace that is hypothesized being responsible for the bad results.

Table 6: Evaluation of the Different Splits.

CS Train / Test	Number of Couples	40000	38669
	Number of Positives	19751	19068
	Number of Negatives	20249	19601
	Mean number of Epitopes a TCR binds to	1	1
	Mean number of Epitopes a TCR doesn't bind to	1	1
	Mean number of TCRs an Epitope binds to	43	42
	Mean number of TCRs an Epitope doesn't bind to	44	43
TS Train / Test	Number of Couples	78435	234
	Number of Positives	38708	111
	Number of Negatives	39727	123
	Mean number of Epitopes a TCR binds to	1	1
	Mean number of Epitopes a TCR doesn't bind to	1	1
	Mean number of TCRs an Epitope binds to	84	1
	Mean number of TCRs an Epitope doesn't bind to	86	1
ES Train / Test	Number of Couples	57009	21660
	Number of Positives	21507	17312
	Number of Negatives	35502	4348
	Mean number of Epitopes a TCR binds to	1	1
	Mean number of Epitopes a TCR doesn't bind to	1	1
	Mean number of TCRs an Epitope binds to	53	346
	Mean number of TCRs an Epitope doesn't bind to	87	87
FS Train / Test	Number of Couples	51511	27158
	Number of Positives	29235	9584
	Number of Negatives	22276	17574
	Mean number of Epitopes a TCR binds to	1	1
	Mean number of Epitopes a TCR doesn't bind to	1	1
	Mean number of TCRs an Epitope binds to	112	32
	Mean number of TCRs an Epitope doesn't bind to	86	57
Full Data	Number of Couples	78669	
	Number of Positives	38819	
	Number of Negatives	39850	
	Mean number of Epitopes a TCR binds to	1	
	Mean number of Epitopes a TCR doesn't bind to	1	
	Mean number of TCRs an Epitope binds to	84	
	Mean number of TCRs an Epitope doesn't bind to	86	

Through the logistic regression model that was fitted on the full data, it was seen that all the chemical and physical features of the epitope were highly significant. The model was fitted on a balanced data which means that all differences seen are not by accident, but caused by the covariates, in this case the ones of the epitope (and some of the TCR-β). Splitting the data such that the marginal and conditional balances aren't disturbed might be a computationally challenging task. A good way to avoid this is finding a data set with balanced number of TCR-βs and epitopes.

To get such a data set, one could try data ensembling: collecting lots of such data sets and merging them into one big data. Such an approach might also have the benefit that the train and test sets get larger.

7.2 SP Task

Like mentioned earlier in the SP task, the noteworhty aspect here is the fact that the highest error rate was seen around position 10. Also note that that this is maximal for the epitope KLFEFLVYGV, the second row in Figure 27 and the blue curve in Figure 28. When looking at Figure 27 at the true matrix of this epitope, the color pixel are more discrete compared to the other three true probability matrices. This is due to the fact that this epitope has a low number of TCR-βs it binds to:

- TTDPSFLGRY: binds to 244 TCR-βs.
- KLFEFLVYGV: binds to 3 TCR-βs.
- LSLRNPILV: binds to 130 TCR-βs.
- KRWIIMGLNK: binds to 66 TCR-βs.

For the other three, because these bind to more TCR-βs compared to the KLFEFLVYGV , there is larger variety of possible amino acids, which reduces the error probability. However, overall, because for all epitopes the TCR-βs have more variety of amino acids around position 10, the error rate will around this position always higher compared to the other positions.

8 Possible Drawbacks

Let's summarize the whole project and the tasks performed. There were two aims here:

- Given the sequence of the TCR-β and given the sequence of the epitope, what is the probability that these two proteins will have a succesful binding?
- Given the sequence of the epitope, what is the appropriate sequence the TCR-β should have in order to have a binding?

The data at hand was the data set available on the website of VDJDB. The first challenge has emerged here already. This data contains only positive samples: couples ever observed. For the first task, actually, the goal is to build a classifier that will assign a (test) couple to the class of `Binding` or `No Binding`. To build this classifier, during the building process, both positive samples and negative samples should be seen. The negative samples were generated by randomly picking a TCR-β and an epitope out of the data. If they weren't a couple, they were assigned to be a negative couple. This approach is also discussed in the literature. However, this approach can be validated by building a classifier built using a data set with positive and negative samples, both proven in a biological lab for example. Such a data set (by preference, large enough) can then be used to train a classifier and test it on the randomly generated couples, which are hypothesized to be negative.

Another aspect to keep in mind is that, while all information is available of the epitope, for the TCR, only the sequence of the TCR-β is used. This simplification is done because the TCR-β chain (and also the α chain) of the CDR3 region of the TCR is that part of the whole TCR that is coming in contact with the epitope (Jorgensen et al., 1992). However, by using this approach, one makes

the strong assumption that the 3D structure of the TCR doesn't matter at all, and the probability of having a binding between the TCR and the epitope is highly determined by this region only. As one can realize this is a very strong assumption and maybe not valid at all. The two components discussed here are proteins. Like mentioned in every biology book about proteins, the amino acid sequence determines the 3D structure of the protein, which in its turn determines the functionality of the protein. Proteins have specific 3D structures, highly related to their function they have to perform. A mismatch in the amino acid sequence can lead to the protein's disability to perform its task. To conclude the comments on the PP task, a good classifier should be built using:

- a data set containing positive and negative samples (balanced), both validated biologically,
- the sequence of the components and
- the 3D structure of the components.

For the last one, a mapping is needed: the 3D structure information should be mapped to a vector as an input for the machine learning or deep learning models.

For the SP task, to determine what amino acid i should be on the jth position, random forest models were built for each amino acid, for each position, leading to the 798 random forest models. However, it must be noted that the inputs are actually chain of categorical values. For example, an epitope of length 8 can be written as $X_1X_2X_3...X_8$. The method explained in Figure 25 and Figure 26 leads to the assumption that for each sequence (chain) the positions are independent from each other:

$$P(X_j|X_i, \forall i \neq j) = P(X_j) \tag{14}$$

To what extend is this true? Is it indeed true that X_j is independent of the other positions? Or is there some dependency? For the latter one, then, the framework of Markov modelling is needed. And if the two chains (epitope and TCR-β) will have an effect to each other, then one needs bivariate Markov modelling.

9 Ethics, Societal Relevance and Stakeholder Awareness

Working with sequenced TCRs brings a number of ethical concerns with it. First of all, knowing the TCR is equivalent to knowing the part of the DNA of the patient that encodes for the TCR of interest. Many articles can also be found in the literature about this issue, one of them being Morris et al., 2006, making a clear distinction between the individual issues and population issues. Individual issues are the release of such data bases and the identifiability, adecaucy of the content and the reporting the results. Population issues are of broader sense, like stereotyping and stigmatization, inclusion and differential benefits and community and culturally specific issues. Like they also mention in their conclusion, those ethical issues should be clearly addressed by developing effective strategies to inform a much broader set of issues in the ethics of a (medical) research.

On the other hand, however, the societal relevance and the possible stakeholders are very clear. Like also mentioned in §11, prediction of an immune response using machine learning and deep learning (or other AI tools) can be of great help during emergencies like epidemics or pandemics. Big stakeholders here are pharma industries, but also other companies, since a pandemic can lead to measures, leading to a reduced economic activities. Being able to make predictions for an immune response can be of great help during vaccine development, which can accelarate the development.

10 Conclusion

In this project, the probability of having a binding between a TCR and an epitope was discussed. Also the sequence of the TCR-β was determined, given the sequence of the epitope. The models used were the classification tree, random forest, boosting and KNN. The software used here was `R` and the required packages for this models were `tree`, `randomForest`, `gbm` and `class`, respectively. Also logistic regression was used, but no special packages were required for this. The metrics were overall accuracy, sensitivity and specificity for the PP task and position wise error rate for the SP task. For the PP task, for CS and TS, the overall accuracy, sensitivity and specificity were showing that model is performing well on the test set. A sensitivity of 0.8 means, given all the positive couples, the model will assign 0.8 of them to the `Bindig` class, whereas a specificity of 0.8 means, given all the negative couples, the the model will assign 0.8 of them to the `No Binding` class. Each time for each split, the overall accuracy lies nicely between the sensitivity and specificiy, because of the 50/50 balance of positives/negatives. The results of ES and FS were worse than those of CS and TS.

11 Future Research and Challenges

What is the future of this project and other projects related to this? Remember the very recent pandemic caused by the corona virus. Because of this pandemic, some measures were taken, leading to a restricted social life and a reduced economic activity. However, to what extend can the world of the 21st century allow such a measure again, and for how long? The history shows already one thing: it was not the first pandemic, and it won't be the last one.

Reducing down the economic activity is something the governments allow in extreme circumstances. However, the best thing to do, is avoiding this. Let's take the corona virus as an example and keep the terminology as easy as possible. Once in the body, the corona virus will code for its spike proteins. In order to get immunized against this virus, it should be recognized by the immune system. In particular, the spike protein can be seen as the ID of the virus and it is this ID that should be recognized in order to start an immune response. The immune system should code for another protein that has to recognize this ID: specialized antibodies. The good reader might notice immediately that in this case, it is about antibody response and not about T-cell mediated immune response. The principle remains the same: the spike protein is now the equivalent to the epitope and the antibody is the equivalent to the TCR-β.

Given that this won't be the last pandemic (as currently the media are talking about monkeypox virus), having a pandemic is actually equivalent to having the following problem:

- The sequence of the disease causer is known. How to fight against this?

Further analysis of this question leads actually to the SP task of the project. Furthermore, if some experiments were done with this disease causer or other ones, and one wants to check if some new antibodies or TCRs will work or not, this leads to the PP task (TCR split or couple split). Having a classifier with maximal sensitivity and maximal specificity will be of great help in order to develop vaccines. Vaccines will accelerate the worldwide immunization against the disease causer. That's why data bases like VDJDB and experiments leading to such data bases are of great value, not only for biology and medicine, but also for statistics, computer science, mathematics and (statistical) physics.

12 Literature

Alberts, B. et al. *Molecular Biology of the Cell*, Garland Science, New York, 2015.

Chollet, F. *Deep Learning with Python*, Manning, New York, 2021.

Osorio, D., Rondon-Villarreal, P. & Torres, R. Peptides: A package for data mining of antimicrobial peptides. The R Journal. 7(1), 4-14 (2015).

Moore, D.S. (1985). Amino acid and peptide net charges: A simple calculational procedure. Biochemical Education, 13, 10-11.

Guruprasad K, Reddy BV, Pandit MW. Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. Protein Eng. 1990 Dec;4(2):155-61. doi: 10.1093/protein/4.2.155. PMID: 2075190.

Boman, H. G. (2003). Antibacterial peptides: basic facts and emerging concepts. Journal of Internal Medicine, 254(3), 197-215.

Ikai (1980). Thermostability and aliphatic index of globular proteins. Journal of Biochemistry, 88(6), 1895-1898.

Weber, Anna & Born, Jannis Rodriguez Martinez, Maria. (2021). TITAN: T-cell receptor specificity prediction with bimodal attention networks. Bioinformatics. 37. i237-i244. 10.1093/bioinformatics/btab294.

Moris, Pieter & De Pauw, Joey & Postovskaya, Anna & Gielis, Sofie & Neuter, Nicolas & Bittremieux, Wout & Ogunjimi, Benson & Laukens, Kris & Meysman, Pieter. (2020). Current challenges for unseen-epitope TCR interaction prediction and a new perspective derived from image classification. Briefings in Bioinformatics. 22. 10.1093/bib/bbaa318.

Fischer, David & Wu, Yihan & Schubert, Benjamin & Theis, Fabian. (2020). Predicting antigen specificity of single T cells based on TCR CDR 3 regions. Molecular Systems Biology. 16. 10.15252/msb.20199416.

Jorgensen, J., Esser, U., Fazekas de St. Groth, B. et al. Mapping T-cell receptor–peptide contacts by variant peptide immunization of single-chain transgenics. Nature 355, 224–230 (1992).https://doi.org/10.1038/355224a0

Krogsgaard M, Davis MM. How T cells "see" antigen. Nat Immunol. (2005) 6:239-45. doi: 10.1038/ni173

Springer I, Besser H, Tickotsky-Moskovitz N, Dvorkin S, Louzoun Y. Prediction of Specific TCR-Peptide Binding From Large Dictionaries of TCR-Peptide Pairs. Front Immunol. 2020;11:1803. Published 2020 Aug 25. doi:10.3389/fimmu.2020.01803

Krangel MS (2009) Mechanics of T cell receptor gene rearrangement. Curr Opin Immunol 21:133-139

Morris W. Foster, Richard R. Sharp, Ethical issues in medical-sequencing research: implications of genotype–phenotype studies for individuals and populations, Human Molecular Genetics, Volume 15, Issue suppl_1, 15 April 2006, Pages R45–R49, https://doi.org/10.1093/hmg/ddl049