Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering
CAI Centre for Artificial Intelligence

Bachelor thesis (Computer Science)

Enhancing Immunotherapy: Predictive Models for Epitope and T Cell Binding via Deep Learning

Author	Valentin Egger Cyril Gabriele
Main supervisor	Prof. Dr. Jasmina Bogojeska
Date	07.06.2024

DECLARATION OF ORIGINALITY

Bachelor's Thesis at the School of Engineering

DECLARATION OF ORIGINALITY

Bachelor's Thesis at the School of Engineering

By submitting this Bachelor's thesis, the undersigned student confirms that this thesis is his/her own work and was written without the help of a third party. (Group works: the performance of the other group members are not considered as third party).

The student declares that all sources in the text (including Internet pages) and appendices have been correctly disclosed. This means that there has been no plagiarism, i.e. no sections of the Bachelor thesis have been partially or wholly taken from other texts and represented as the student's own work or included without being correctly referenced.

Generative AI tools were used in the process of this work. In specific, DeepL and ChatGPT were employed to improve translation on parts of the text and understanding content.

Any misconduct will be dealt with according to paragraphs 39 and 40 of the General Academic Regulations for Bachelor's and Master's Degree courses at the Zurich University of Applied Sciences (Rahmenprüfungsordnung ZHAW (RPO)) and subject to the provisions for disciplinary action stipulated in the University regulations.

City, Date:	**Name Student:**
Staad, 07.06.2024	Gabriele Cyril
Arbon, 07.06.2024	Egger Valentin

Abstract

Motivation

The human immune system is a highly intricate and individualistic phenomenon. Consequently, there is no universal solution that can be applied to all individuals to combat pathogens. It would be beneficial to ascertain which components of pathogen-specific immune responses contribute to pathogen elimination. Specifically, the objective is to identify which T cells can bind to and eliminate specific epitopes.

Results

Despite challenges such as incomplete TCR information, the absence of true negative samples, and inherent biases in available datasets due to the limited representation of the vast number of epitopes and T cells, the findings highlight the potential of deep learning to enhance personalized immunotherapy by predicting TCR-epitope bindings. This thesis introduces a novel method to address the absence of true negative samples by generating synthetic non-binding samples in an informed manner. Additionally, it emphasizes the importance of incorporating both α-chain and β-chain sequences of the TCRs and optimizing model parameters to enhance predictive accuracy, thereby facilitating more effective personalized treatments. Furthermore, experiments incorporating physicochemical properties of the protein complexes indicate that these properties do not improve model performance. Comparisons are also made to evaluate the impact of information precision in germline gene segments (V and J regions) and the MHC on model performance.

Table of Contents

1 Introduction

The ability to predict the binding between peptide major histocompatibility complexes (pMHCs) and T cell receptors (TCRs) provides valuable insights for researchers. It leads to advancements in various immunology-related sectors. This is a complex task because the TCR repertoire is highly individual.

The TCR repertoire of a patient provides useful information about their immunological history, including past and current infections, vaccination effectiveness, and autoimmune reactions. A reliable prediction tool does help in the interpretation of this information and assists in the design of personalized immunotherapies, such as cancer treatment [1], [2], [3]. Furthermore, computational analysis enables the evaluation of efficacy and potential hazards related to cross-reactivity [3]. This suggests that such a model leads to significant advancements in the field of immunotherapy [4].

A current obstacle is that the time-consuming and costly experimental approaches used to characterize TCR-pMHC interactions continue to limit the rapid advancement of immunotherapies [5]. A model that predicts these bindings provides a remedy for these experimental approaches, leading to cost reductions and faster development due to higher throughput. Additionally, these models must be capable of comprehending the molecular mechanisms that establish the affinity of a TCR for an epitope. Although sequence similarity between epitopes and the T cell appears to be crucial, TCRs can bind to multiple epitopes and vice versa. Furthermore, TCRs can exhibit cross-reactivity, meaning they can recognize and bind to multiple distinct epitopes. Consequently, this process is highly complex, as the same TCR can interact with various epitopes, adding to the challenge of accurately predicting binding affinities [6].

Another challenge in this problem is that the currently available data is predominantly derived from viral sources and only encompasses a limited portion of the target space [7]. This limitation arises due to the potential for recombination to generate a vast array of TCRs. With estimates ranging from 10^{20} to 10^{50}, the two chains of the TCR can collectively produce a vast TCR diversity that is orders of magnitude greater than the estimated $3.7 \bullet 10^{13}$ cells in the human body [8]. With regard to epitopes, only approximately 1100 different epitopes are currently available in open-source databases [9]. Furthermore, no true negative binding data samples are available. The combination of these obstacles makes this prediction task complex and challenging.

Many studies have worked on developing machine learning models to solve these problems. These approaches used a wide range of methods within the machine learning landscape [7], for example, including recent deep learning approaches such as Transformer [7], [10] as explained in the paper from Myronov et al. [11] or Korpela et al. [4]. Although there have been several attempts, many researchers agree on the fact that the biggest challenge is to generalize predictions to unseen

targets [4], [11]. Furthermore, these researchers agree on the fact that this problem aligns with the lack of diverse data.

1.1 Motivation

Due to the lack of a reliable model that generalizes well to unseen sequences in the current state of the art, the objective of this thesis is to develop a machine learning model that can predict the binding affinity between TCRs and pMHCs, especially for unseen data points. Such a model will address the needs and overcome the challenges in personalized immunotherapy design preventing potentially harmful cross-reactivities [1], [2], [3]. This thesis distinguishes between four distinct prediction tasks: one where the TCR is unseen, but the epitope is known, a second where the epitope is unseen, but the TCR is known, a third where both the TCR and the epitope are unseen and a fourth where both, the TCR and epitope, are seen.

1.2 Objectives and Requirements

This thesis aims to develop deep learning models and accurately predict the binding of a TCR and a pMHC. The different models use different data, as described in section 4.3 and the following subsections. The datasets are created by combining three data sources to train these machine learning models. The data from these open-source databases are evaluated, processed, supplemented, and adapted. These steps are described in detail in section 4.1.

Regarding binding prediction, a subtle but important distinction must be made between different prediction tasks. TCRs and pMHC, which are seen by the model during training, represent only a small number of possible bindings. The most challenging task is the binding prediction of unseen data, which in this case is unseen TCR chains and epitope sequences. The differentiation of the prediction tasks elaborated within this thesis is described in detail in section 4.1.3.

More specifically, the objectives of this thesis are the following:

- determine the current state of research and gain domain-specific knowledge
- create cohesive datasets from publicly available data
- informed creation of non-binding (negative) TCR-pMHC samples
- design and implement appropriate deep learning architectures for TCR-epitope binding prediction problem
- compare and analyze the performance of the models

The aim of this thesis is to answer the following questions:

- Given the current datasets, can a deep learning model be trained to predict the binding between TCRs and pMHCs and generalize well to sequences not seen during training?
- What is the impact of utilizing solely the TCR CDR3 β sequences in comparison to the use of both the TCR CDR3 β and the TCR CDR3 α sequences on the model's performance?

- What is the impact of the precision of the V and J region and the MHC on the performance of the in silico models?
- What effect does the incorporation of the corresponding protein complexes' physicochemical properties have on the models' performance?

2 Theoretical Background

This section analyzes the theoretical mechanisms underlying the binding process of T cells and pMHCs. This theory is fundamental to acquiring domain-specific knowledge regarding the adaptive immune response mechanisms, which in turn is essential to understanding the binding process.

2.1 The Role of T Cells in the Adaptive Immune Response

The innate immune system is the body's first line of defense against infections [12]. Nevertheless, it is only capable of neutralizing pathogens that exhibit specific molecular patterns or those that trigger interferons and other nonspecific defenses. However, the adaptive immune system's lymphocytes have evolved to detect a large variety of distinct antigens from bacteria, viruses, and other disease-causing organisms [13]. Therefore, this ability is also attributed to the TCRs on T lymphocytes. They achieve this remarkable diversity through a random process of DNA rearrangement involving recombination of the germline's V, D, and J gene segments. This process is further characterized by the addition and deletion of nucleotides at the V(D)J junctions, allowing the generation of highly variable antigen receptor repertoires [7] Laydon et al. [8]. demonstrate that the theoretical diversity for these receptors in an individual ranges from 10^{15} to 10^{20}. The effective diversity spectrum of an individual is typically between 10^{6} to 10^{10} [8]. Table 1 below provides an overview of the origin of genetic diversity.

Moreover, T lymphocytes, also called T cells, play a significant role in the adaptive immune response mechanism. When T cells encounter antigens that their receptors react to, adaptive immune responses are triggered, provided that the appropriate inflammatory signals are present to facilitate activation. T cells are activated when they encounter dendritic cells that have migrated from infection sites, acquired antigens, and moved to secondary lymphoid organs. Dendritic cells in the tissues are stimulated to engulf the pathogen and destroy it intracellularly when their pattern recognition receptors (PRRs) are activated by pathogen-associated molecular patterns (PAMPs) at the infection site. Additionally, they use receptor-independent macropinocytosis to absorb extracellular substances, such as bacteria and virus particles. As a result, peptide antigens are displayed on the major histocompatibility complex (MHC) molecules of dendritic cells, activating lymphocyte antigen receptors. After this, it is possible for a specific T cell to recognize and bind to the peptide displayed on the surface of the MHC molecule [13].

2.2 T lymphocytes and their receptor chains

T cells are essential because they can destroy intracellular invaders [13]. In contrast, antibodies act solely in the extracellular space and blood. T lymphocytes are the key players in cell-mediated immune responses. However, T cells also contribute to the immune response against extracellular organisms. Therefore, they must exhibit various effector functions [13].

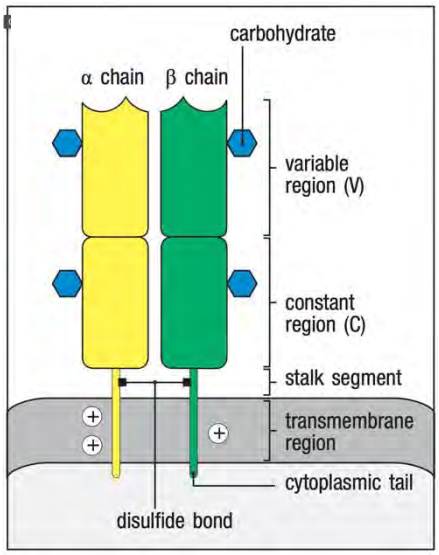


Figure 1: Illustration of the T cell receptor complex's structure [13]

Each T cell consists of two different polypeptide chains, called T-cell receptor α (TCRα) and β (TCRβ). One can observe the structure of the T cell receptor in Figure 1 [13]. This combination represents the majority, ranging from 95% to 99.5% of the T cells found in an individual. Nevertheless, a small percentage, ranging from 0.5% to 5% of T lymphocytes have a distinct paired chain, resulting in the presence of γ:δ T-cell receptors [14]. In the following, the term T cell receptor is used analogously for α:β TCRs as not otherwise stated. The complementarity-determining region 3 (CDR3) of the TCRβ is widely regarded as the most crucial component in the interaction with the peptide of the MHC [15].

2.3 CD4 and CD8 on the T lymphocyte surface

T cells can be distinguished based on their binding to MHC class I or MHC class II molecules, which determines their specific function. T cells expressing CD4 proteins on their cell surface activate other cells, while those expressing CD8 proteins are cytotoxic. CD8 recognizes MHC class I molecules, while CD4 recognizes MHC II molecule complexes. These cell proteins are called co-receptors [13]. The interaction of T cell receptors and the two MHC molecules, including the various co-receptors, is illustrated in Figure 2.

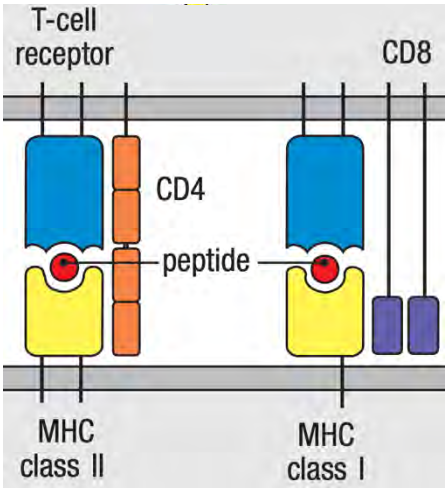


Figure 2: Diagram of T cell receptor interactions: Left, TCR with MHC class I and CD8 co-receptor. Peptide presentation is shown centrally [13].

2.4 The role of V and J Regions in T Cell Receptor Diversity

The adaptive immune system's remarkable ability to recognize millions of distinct antigens is attributed to the random recombination of multiple gene segments. The TCR gene loci contain multiple V and J gene segments that undergo rearrangement during T cell development. This process generates the vast repertoire of receptors required for antigen recognition. In addition to the combinatorial diversity provided by V and J recombination, junctional diversity further enhances the repertoire. The enzyme terminal deoxynucleotidyl transferase (TdT) adds random nucleotides at the junctions between segments. This random addition or deletion of nucleotides significantly increases the variability of the antigen-binding site, particularly the complementarity-determining region 3 (CDR3), which plays a key role in antigen binding [13].

These mechanisms collectively create a vast range of TCRs, enabling T cells to recognize a wide array of antigens and mount a tailored immune response. A summary of the various origins of T cell receptor segments is presented in Table 1.

Table 1: Sources of T cell receptor diversity and the number of human T cell receptor gene segments [13]

α:β T cell receptors		
	β	α
Number of variable segments (V)	52	~70
Number of joining segments (J)	13	61
Number of V gene pairs	5.8×10^6	
Number of junctional diversities	$\sim 2 \times 10^{11}$	
Number of total diversities	$\sim 10^{18}$	

2.5 T Cell Receptors in the Adaptive Immune Response

The TCR on the surface of a T cell plays a critical role in the adaptive immune system by recognizing immunogenic peptides, whether non-self or altered self. These peptides, known as epitopes, must be presented in conjunction with an MHC molecule on the surface of another cell to initiate an adaptive immune response [6]. In fact, TCRs can cross-react to multiple epitopes [16]. These T lymphocytes have different functions depending on the cell protein that appears on the surface. In addition, the receptors have different chains, namely α:β T-cell receptors or γ:δ T-cell receptors. To defend against a wide range of infections, a diverse repertoire of T-cell receptors with varying specificities is required. This repertoire is created through random gene recombination, which leads to great diversity [3].

3 State of the Art

Several approaches using different machine learning paradigms have been developed to tackle this complex binding prediction problem [4], [6], [17], [18]. However, a major challenge is to develop a model that can generalize to unseen peptides [6], [19], [20].

3.1 Challenges in the Available Data

Datasets such as VDJdb [21], McPAS-TCR [22], or MIRA [23] exhibit a significant bias in epitope diversity and by the absence of true false negative samples. The bias in terms of the variety is because only around 100 antigens matter for 70% of the TCR-antigen pairs. This means that the datasets available in the field are biased since they contain only around 1100 different epitopes [9]. In addition, these epitopes are mostly of viral origin [7]. The combination of all these factors limits the development of a generalizable model.

Building on the observations made by Moris et al. [6], 90% of the data points in the dataset used in their research were bindings from MHC class I and only 10% from MHC class II [6]. Combined with the previously mentioned lack of epitope diversity, addressing these issues is crucial for advancing TCR-pMHC binding prediction models. These data biases introduced by imbalances can significantly impact the model's ability to generalize across the wide range of possible TCR-pMHC interactions [24].

Furthermore, the aforementioned study conducted by Moris et al. identifies limitations in the data due to the absence of true negative samples [6]. Goffinet et al. achieved a 1:1 distribution of true positive and true negative samples by exchanging TCR sequences [17]. This was done under the assumption that the TCRs gathered from the ImmunoSEQ [25] portal do not bind to the given epitopes. While innovative, the approach of using random shuffling to create negative samples may not fully capture the nuanced nature of non-binding interactions. This method assumes that any TCR not known to bind a given epitope is a true negative, which might not always be the case. As mentioned in the Introduction section, TCR can cross-react with different epitopes. Developing more sophisticated methods for generating or identifying true negative samples could greatly enhance model performance. Nevertheless, this random shuffling method is established in the field of TCR interaction prediction [4].

Moreover, the incomplete information on TCR α and β chains in datasets poses another significant challenge. Not all datasets, and not even all data samples within a supporting dataset, contain sequence information on both the α and β chains of a specific TCR. The VDJdb [21] includes information about both chains for some data points, while the MIRA [23] does not provide any information about the TCR α chains. It is demonstrated that incorporating information on both chains improves the model's performance [19], [26]. However, paired data is still scarce, and single TCR beta sequences dominate the currently available datasets [3]. The absence of complete information limits the model's ability to accurately predict binding, as it cannot account for the full

structural and compositional context of the TCR-pMHC interaction. Efforts to fill these data gaps, such as targeted sequencing of both chains for a more comprehensive dataset, could improve the accuracy of predictions.

The suggestion by Meysman et al. to include information about the CDR1/CDR2 regions, in addition to the CDR3 region, points to another avenue for enhancing model performance [26]. This can be done implicitly by including annotated V- and J- genes corresponding to these regions or directly by including the amino acid sequence of them [20]. The CDR3 region is known for its high variability and is a critical determinant of antigen specificity [1]. Furthermore, the CDR1 and CDR2 regions contribute to antigen binding and specificity. Incorporating these regions into models could provide a more nuanced understanding of TCR-pMHC interactions, reflecting the true complexity of T-cell recognition.

But simply including more data is not the key to solving this complex prediction problem. The ImmRep workshop found no relationship between model performance and the number of training samples [26]. Moreover, Meysman et al. came to the result that many methods are limited by the lack of diversity in the datasets. This correlates with the mentioned problems encountered by the previously mentioned research and the insight gained from the exploratory data analysis done in this thesis shown in the following section. In their study, Meysman et al. also showed how well more advanced machine learning models perform compared to distance-based techniques [26]. Therefore, distance-based models can serve as baseline models with limited computational costs.

A further problem with current datasets is that the samples generated by high-throughput sequencing methods tend to contain noise, particularly from nonspecific background binding events [27].

3.2 Overview of Current State-of-the-Art Models

Creating a generalized and accurate machine learning model for unseen TCRs and pMHC remains an unsolved problem due to the complexity of the binding process and the lack of diverse datasets. This section reviews the current state-of-the-art (SOTA) models and identifies their shortcomings. Firstly, it is necessary to clarify the prediction task of the models. In addition, a brief overview of the different possible machine learning paradigms will be provided. Finally, detailed insights into the technical methods used are given.

3.2.1 Overview of the different Machine Learning Paradigms

So far, efforts to address the prediction problem have primarily employed two distinct categories of machine learning approaches namely supervised and unsupervised variants. To predict TCR specificity, preliminary attempts were made to utilize unsupervised clustering techniques. This approach assumes that sequence similarity, or the degree to which different sequences share similar features, is correlated with specificity similarity. The rationale behind unsupervised methods suggests TCR clusters with comparable sequences should attach to the same targets. These

methods typically entail the first step of establishing a distance measure, which is then utilized in the K-nearest neighbor method to select test sample labels based on the samples from the closest training set. One example of a simple distance metric is the Levenshtein distance.

The majority of models that are trained in a supervised fashion employ neural network architecture. These models encompass a diverse array of design types, including convolutional architectures, Gated Recurrent Units (GRUs), Transformers employing the self-attention mechanism, and Long Short-Term Memory (LSTM).

Weber et al. [7] mention Protein Language Models (PLMs) as a third, distinct paradigm, which is utilized in this thesis. According to their findings, every new model can be categorized into either the supervised or language sector. This increased utilization of data-intensive methodologies has led to a notable enhancement in performance outcomes when compared to other approaches [7]. However, many machine learning models, especially PLMs, are currently uninterpretable because amino acid information is encoded in complicated and abstract latent spaces [7].

3.2.2 Overview of existing Models and their Methodologies

Some research models such as iSMART (2019) [28], TCRdist (2017) [29], ALICE (2019) [30] and TCRMatch (2021) [31] used clustering algorithms.

The Random Forest algorithm was used in the epiTCR model published in 2023. According to the paper, this model outperforms models like ImRex, NetTCR2.0 and three other models. The epiTCR model was evaluated with a mean AUC at 0.98 for a specified prediction task detailed explained by the paper [32] even without using a deep network as Random Forest is a conventional machine learning method [33]. This research used data from TBAdb, a subgroup of data within PIRD [34], VDJdb [21], McPAS-TCR [22], IEBD and 10x [35].

The following models employ the deep learning paradigm, which is a data-intensive method [4].

ImRex [6] is a model proposed by Moris et al. in 2019 that employs a convolutional neural network (CNN) architecture. In the dataset they included not only information about the CDR3-epitope pair representing the generic sequence, but they also introduced a so-called interaction map where they combined physicochemical properties such as mass, hydrophobicity, hydrophilicity, isoelectric point [6]. The inclusion of these information is according to this research beneficial for the model's performance. This paper used data from the VDJdb [21] to train the ImRex model. To generate negative samples, the authors used an independent dataset of 250,000 TRB CDR3 sequences from healthy TCR repertoires, as described in detail by Dean et al [36]. After enforcing specific size criteria and removing duplicate entries, they refined the dataset to 248,895 unique sequences. Notably, this dataset had minimal overlap with the primary VDJdb [21] dataset, containing only 749 common sequences, thus providing a set of negative samples with distinct characteristics for experimental analysis [6].

A model called NetTCR2.0 was published in 2021 by Montemurro et al. Similar to ImRex, where they used a CNN architecture [6]. They also showed that it is best to use information about both

the CDR3 α and CDR3 β sequences [19]. A follow up model called NetTCR2.2 was made publicly available in December 2023 and uses an advanced architecture that was originally inspired by the work of Montemurro et al., who demonstrated that improved performance could be achieved by merging the characteristics of peptide-specific and pan-specific models [20]. The data used was gathered from IEDB [37] and VDJdb. Negative results were generated by swapping the TCRs for a given peptide with TCRs that bind to different peptides. In this case, to reduce the possibility of producing false negative results, such TCRs were limited to samples from peptides with a Levenshtein distance of at least 3 [20].

After ingesting a TCR and an epitope sequence, the TITAN model from Weber et al. uses either learned embeddings from SMILES [38] or BLOSUM62 [39] for amino acid sequence encoding. Then, context attention layers construct attention weights for each amino acid of the TCR sequence given an epitope and vice versa. This is followed by 1D convolutions of different kernel widths on both input streams. Finally, a stack of dense layers outputs the binding probability [3]. Although TITAN is finer-grained, the architecture shown in their paper is basically the same as the one suggested by Born et al. 2021 [40]. This architecture belongs to the category of attention-based neural networks. Their study generated negative data by randomly mixing TCR sequences with epitopes they are not known to bind, based on the well-established assumption that a randomly selected TCR is unlikely to bind a given epitope [6], [41]. This method prevents overestimating performance better than incorporating naive TCR sequences from external sources. In addition, this shuffling technique allows for an equal number of negative and positive samples within the dataset [3].

EpiTCR, ImRex, and TITAN are similar in their use of data, as they did not utilize information regarding CDR3 α, only CDR3 β of the TCRs.

In December 2023, a study by Korpela et al. [4] introduced the EPIC-TRACE model. This model utilizes the peptide sequence, MHC allele, V and J genes, and CDR3 domains in its architecture. Multi-head attention techniques are employed to focus on relevant features for interaction, convolution layers are used to infer binding motifs, and ProtBERT embeddings are utilized for sequence information [4], [42], [43]. The model utilizes learned linear embeddings for gene information and incorporates distinct processing pathways for each TCR chain and epitope. Its three output heads enable the model to handle various input configurations based on the availability of chain information [4]. This advancement allows for greater flexibility in the use of benchmark datasets which, for backward compatibility reasons, may not contain information on both CDR chain regions. To train the EPIC-TRACE model, they used VDJdb and IEDB data for the positive datapoints. The negative samples were synthetically constructed by shuffling, similar to above mentioned techniques.

In 2024, a new model called MATE-Pred was proposed by Goffinet et al. The main idea of this work was to combine information about the amino acid sequence, physicochemical properties, and a

contact map (C-Map) that should represent the 3D structure of the bound protein complex containing the TCR and the pMHC [17]. They followed a pipline called PiTE [15], which will be explained in more detail later in this section. The data were collected from VDJdb, IEDB and McPAS and they also generated their negative samples with a shuffling method to further improve the datasets towards a more balanced one. The study employs a methodology that utilizes MHC class I epitopes, which are combined with TCR CDR3 β sequences. The data from the VDJdb was used to create an independent test dataset. The authors stated that using an independent test set, MATE-Pred sets a new benchmark for performance and outperforms other model from the field [17].

The result of the work of Zhang et al. was a pipeline called PiTE, where different architectures were evaluated, and it was found that Transformer-like encoders outperformed the other architectures such as CNN or two others [15]. This suggests that the utilization of Transformer blocks is a viable approach for the currently developed machine learning architectures.

4 Materials and Methods

In this section, the data and models developed in this thesis are described in detail.

4.1 Data

The acquisition and preparation of data represent foundational steps in developing machine learning models, particularly in the immunotherapy domain, due to the shortcomings of the currently available data, as detailed in section 3.1. This chapter outlines a structured pipeline designed to process and prepare immunological data sourced from several specialized databases. The focus is on data related to TCR and peptide-major histocompatibility complex (pMHC) interactions, which are crucial for advancing personalized medicine and immunotherapy. Additionally, insights gained during the Explorative data anylsis (EDA) are shown in diagrams and tables.

4.1.1 Challenges in the Data Acquisition

The data acquisition process involved three distinct sources: VDJdb [21], McPAS-TCR [22], and IEDB [37]. This diversity presents a challenge in understanding and harmonizing the datasets.

4.1.1.1 Harmonizing Datasets from Different Sources

Each database employs a distinct column naming convention for the same features. This renders the normalization of these columns difficult, thus hindering the concatenation of the three datasets into a unified form. Additionally, the information in these columns is inconsistent, as detailed in section 4.1.1.3. The columns of interest have been selected and renamed accordingly.

4.1.1.2 Handling Comma-Separated Values

The IEDB [37] dataset often contains multiple comma-separated values within a single feature. For instance, a sample might list several comma-separated MHC values. To address this, each value was split into a separate row, while retaining the other features.

4.1.1.3 Inconsistent Levels of Information

The datasets did not always provide the same level of information. For example, McPAS-TCR includes a column called "*T.Cell.Type*" with values CD8 or CD4, corresponding to the MHC class the T cell binds to, as noted in section 2.3. In contrast, VDJdb has a single column indicating the MHC class. The IEDB dataset contains either MHC class I or MHC class II values exclusively. Since this work focuses on MHC class I, all MHC class II values were removed from the VDJdb and McPAS-TCR datasets, and only MHC class I datasets from IEDB were used.

4.1.2 Comprehension about Paired and Beta Datasets

Beta chain data focuses exclusively on the β-chain, offering a limited perspective on the diversity and functionality of the TCR repertoire. This constrained view potentially limits the ability to fully understand the workings of the immune system. On the other hand, paired chain data encompasses both the α- and β-chains, thereby providing a more comprehensive understanding of TCR repertoire functionality. This thesis includes both chains in the analysis. The beta chain

data is abundant, while the paired chain data is more detailed but scarce, largely due to historical technological limitations [44].

4.1.3 Differentiation in the Prediction Task

Most of the comparisons of the different state-of-the-art (SOTA) models and corresponding methods are not coherent because different datasets were used. Additionally, the resulting values of the different metrics need to be considered carefully taking the details of the specific prediction tasks into account. An acronym for the prediction task case is used in this project to distinguish between the different relevant binding prediction subtasks. The acronym is TPP and stands for TCR-Peptide Pairing. TCR includes the CDR3 chains, and peptide is the column epitope in the datasets of this thesis.

The objective of this approach is to ascertain whether a model has already seen a TCR or epitope during training, before making a prediction. The initial task involves the peptide and TCR being seen (TPP1). The second prediction task is limited to the peptide being seen while the TCR is unseen (TPP2). A third scenario is when neither the TCR nor the peptide is seen (TPP3). The TPP4 scenario occurs when the TCR is seen but the peptide is not.

TPP4 samples are a byproduct of the reclassification of the paired test set, explained in section 4.1.5.1, and the process of generating negative samples, e.g. when the peptide seen in a TPP1 sample is replaced with an unseen peptide from a TPP3 row, a TPP4 sample is created. This does not provide any additional benefit, but it is a verifiable fact. The process of creating negative samples is explained in section 4.1.6.

Table 2: TCR-Peptide Pairing Tasks

	TCR	Peptide
TPP1	Seen	Seen
TPP2	Unseen	Seen
TPP3	Unseen	Unseen
TPP4	Seen	Unseen

TPP3 is considered the most general prediction because it predicts the binding affinity of unseen TCRs and unseen peptides. Therefore, it's the most interesting task [4]. According to Korpela et al. [4], only TPP1 and TPP2 have been solved with appropriate results.

In the model, the binding prediction is treated as a binary classification problem. This implies that the evaluation of every epitope-TCR interaction is based on the occurrence of a binding event, which is a binary outcome (bind or not bind). The problem is simplified for practical implementation, despite the complexity of epitope-TCR pairing in immunological processes. In such processes, the target epitope is typically unknown, and each TCR might theoretically be paired with numerous epitopes (a multi-label classification scenario) [26].

The simplified approach enables the implementation of the binary cross-entropy loss, which is a suitable method for optimizing binary classification models. This approach treats each possible epitope-TCR combination as an independent binary classification problem.

4.1.4 Data Acquisition Pipeline

The data for this study was manually downloaded from three primary immunological databases. Only human data is used.

- **VDJdb**: A curated database that contains TCR sequences paired with known antigens [21].
- **McPas**: This database is similar to VDJdb but is specifically focused on TCR sequences known to be associated with various pathologies. [22].
- **IEDB**: The IEDB encompasses a comprehensive range of immune epitope data, encompassing antibody, T cell, and MHC binding contexts associated with infectious, allergic, autoimmune, and transplant-related diseases. [37].

The flow chart in Figure 3 illustrates the general process by which the data is prepared for the model at an abstract level.

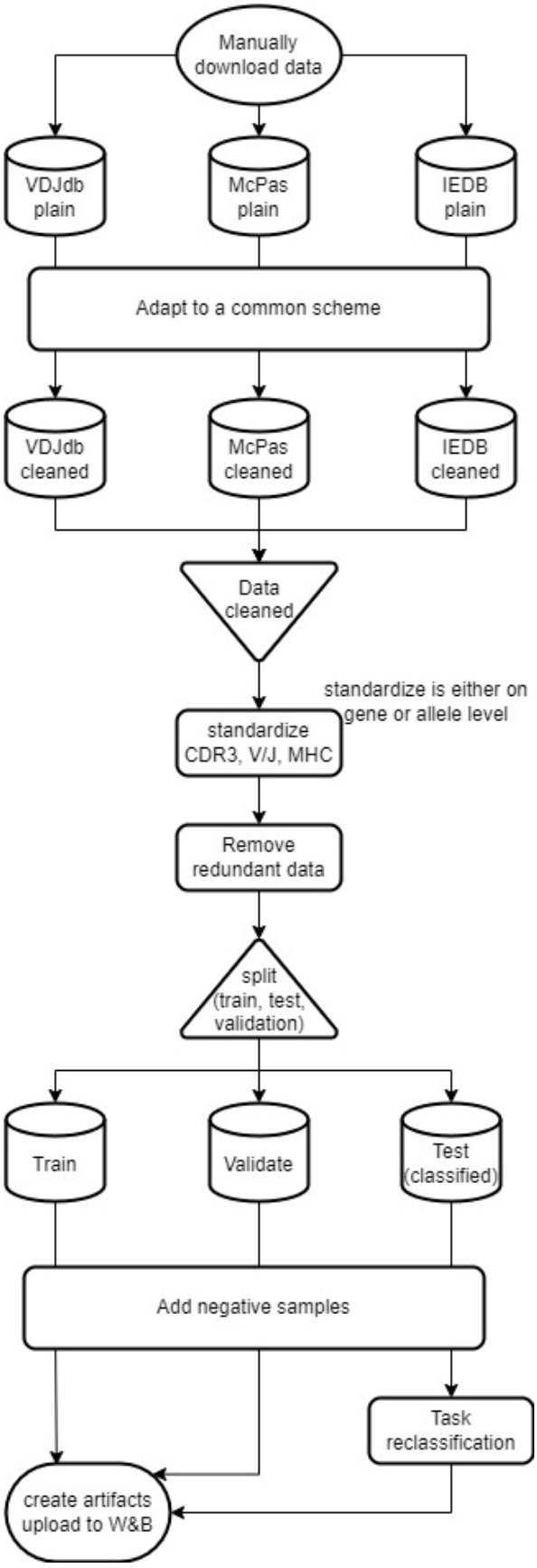


Figure 3: Data Acquisition Pipeline

After downloading and harmonizing the data from the three sources, the CDR3, V/J and MHC attributes are standardized with an open-source python package called tidytcells [45]. Entries that could not be standardized have been deleted.

After standardization, a custom method was employed to remove duplicated data. The standard duplicate removal approach provided by Python packages is flawed because it treats rows with different amounts of filled columns as non-duplicates, even if they have the same values in the filled columns. To avoid this, entries are initially sorted according to the number of existing values. Thereafter, each entry is compared with the subsequent entries. These subsequent entries are marked as duplicates and removed if the values of them are identical or missing. Thus, the cells of subsequent entries that are empty are disregarded, which is equivalent to assuming that they have the same value as the entry being compared.

The split into train, test and validation is illustrated in Figure 4. The inclusion of negative examples is implemented after the split, as otherwise it proved challenging to distribute the TPP tasks in the test set. The reclassification is necessary because the newly added negative entries are not classified. Finally, the data is uploaded to the Weights & Biases platform.

4.1.5 Data Split

The data split is similar for the beta and paired datasets. The paired dataset has some specific characteristics regarding the seen/unseen classification. The data is divided in three subsets needed for the model development and evaluation: 70% of the data makes up the training set, 15% makes up the validation set, and the remaining 15% form the test set. The process can be seen in detail in Figure 4. The distribution is based on conventional practice.

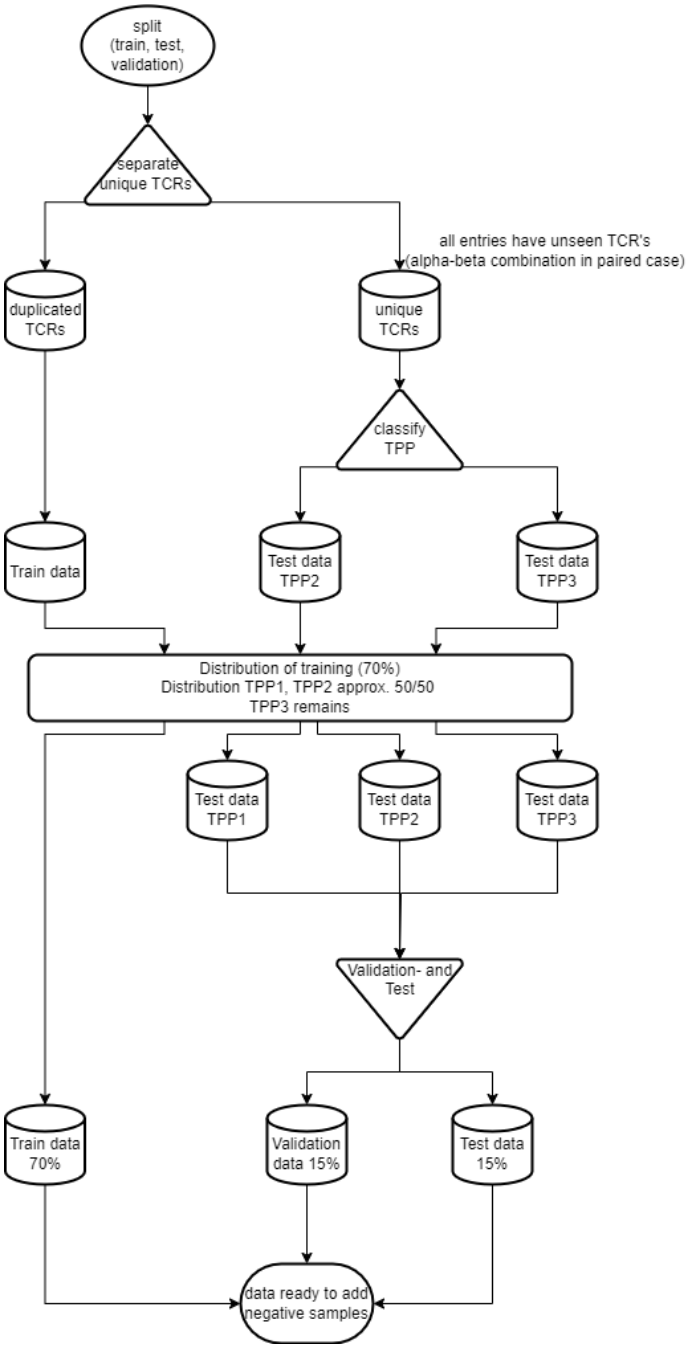


Figure 4: Data Split Pipeline

The initial step involves the harmonized and standardized dataset being divided into two distinct parts, the train dataset, and the test dataset, without consideration of the distribution. The test dataset comprises all entries with unique TCRs, while the train dataset contains the remaining duplicates. Every entry from test data gets classified as explained in section 4.1.5.1. Given that the test dataset comprises a set of unique TCRs, only TPP2 and TPP3 are possible tasks. Due to the small number of entries with unique epitopes, there are too few rather than too many TPP3 tasks. For this reason, these remain in the test dataset.

The objective is to achieve a distribution ratio of 70% for training data and 30% for testing data, which is later divided into validation and test datasets. The procedure is to either transfer entries from the train dataset to the test set, or alternatively, to transfer TPP2 entries from the test dataset to the train dataset. It is guaranteed in the application code that any data points transferred from the training dataset to the test dataset can be classified as TPP1. This is a prerequisite for the ensuing distribution of TPP1 and TPP2 in the test dataset.

As has been the case throughout all application runs, the TPP2 entries have constituted the majority. Consequently, several TPP2 entries from the test set have been exchanged with TPP1 entries from the training set, ensuring that the test set contains an equal number of TPP1 and TPP2 entries.

Finally, the test dataset is divided into test and validation dataset. It is ensured that half of the TPP1, TPP2, and TPP3 samples remain in the test dataset. However, these entries must then be reclassified, as the entries in the validation dataset are seen and were considered unseen when they were still contained in the test dataset.

4.1.5.1 Test Dataset - Prediction Task Classification

Each datapoint in the test dataset is classified as either TPP1, TPP2, TPP3, or TPP4, based on section 4.1.3. Consequently, each CDR3 and epitope value in the test dataset is evaluated to determine whether it exists in either the training or validation dataset. The dataset contains a limited number of distinct epitopes. Therefore, it is not feasible to introduce an arbitrary number of TPP3 or TPP4 samples into the test set without removing data points from the training or validation set.

This thesis deals with paired chain data as well as beta chain only data, as explained in section 4.2.3. For the beta chain only dataset, the classification is straight forward but for the paired chain dataset, the TCR consists of two chains. The question then arises as to what is considered to be seen. Two possible scenarios exist: either the occurrence of the same combination in the seen data must be verified, or the occurrence of either α- or β-chain must be observed in the seen data. A third scenario would be if both, the α- and β-chain, were in the seen data, but the specific combination of chains is not relevant. This scenario may be analyzed in the future but is not addressed in this thesis.

Paired Normal Classified

A TCR is identified as seen, when α- and β-chain occur in the same combination in the seen data (validation or train datasets). This is the classification that was introduced first in the application code.

Paired Reclassified

In the reclassified test dataset, a TCR is seen if either α- or β-chain occurs in the seen data. Compared to the normal classified test dataset, this results in an increased number of TCRs being seen, leading to some TPP3 samples being reclassified as TPP4 and some TPP2 samples being reclassified as TPP1.

4.1.6 Negative Samples Creation

This study employed an innovative methodology to address the challenge of the absence of negative samples. Negative samples are defined as synthetically created data samples where the T cell does not bind to the epitope. An established method in the field to create negative samples is to randomly shuffle the epitopes and, if available, the corresponding MHC [4]. The assumption here is that the randomly selected epitope does not bind with the T cell.

Instead of using the established random method of epitopes, the amino acid sequences of epitopes were previously embedded using the ProtT5 PLM [43]. By doing so, this process ensured that the epitope selected for the negative sample was dissimilar to the epitope the T cell binds to in the true positive sample.

This process resulted in the computation of per-epitope embeddings, obtained by averaging the embedding of each amino acid comprising the epitope yielding a 1024-dimensional vector. Furthermore, the negative sample generation embeddings are conducted on a per-epitope basis to minimize the computational cost, while maintaining the desired level of accuracy. A detailed description of the embedding computation process is available on the ProtTrans GitHub repository of the developers, the link can be found in the paper [43].

After obtaining the per-epitope embeddings, they are used in the informed epitope shuffling process. A random epitope of the dataset is selected as a potential negative sample. The embedding of the positive and the embedding of the randomly selected potential negative epitope are compared using the cosine similarity [46]. If the dissimilarity of the embeddings is within a manually set range of $[-1, 0.75] = \{x \in R \mid -1 \leq x \leq 0.75\}$, the randomly selected epitope is accepted as a negative sample. If it is accepted the epitope sequence and the MHC is concatenated with the information of the T cell of the true positive sample and added to the dataset as a negative sample, indicated by setting the "*Binding*" feature to 0. Otherwise, a new randomly selected sample is chosen, and the same algorithm is applied again.

The interval threshold is not adjusted nor optimized during the training process. It is set based on the following reasoning: a cosine similarity of 1.0 indicates that the embeddings are identical in terms of the information they contain [46]. The upper bound is set to 0.75 to balance between ensuring some dissimilarity and allowing enough epitopes to be identified. This specific upper bound, rather than a higher one, is chosen because the number of epitopes in currently available public datasets is quite limited [9].

To ensure the prediction task classification described in 4.1.5.1 will not be affected, negative samples are synthetically generated for train, validation, and test individually. This ensures that no new CDR3 sequences or epitopes are added to the seen data. After adding the negative samples, the ones in the test dataset are classified.

By employing this methodology, an approximately balanced (1:1) distribution between positive and negative samples can be achieved. This method was developed to address the dearth of negative

binding samples, which represents a significant challenge in the field. [6]. To the best of the authors' knowledge, this methodology represents a completely novel approach.

4.1.7 Encoding of the Epitope, TCR CDR3 α and TCR CDR3 β Sequences

The ProtT5 [43] model used in the negative samples generation is also used to compute the embeddings for the sequences. In contrast to the embeddings used in the negative sample generation, the embeddings for the epitope, TCR CDR3 α and TCR CDR3 β sequences are computed per-amino-acid. This means the resulting embeddings have different shapes. For example, if the input sequence is “CALRDYKLSF”, which has a length of ten, the embedding before applying padding results in a vector shape [10, 1024]. This high granularity level of the embeddings guarantees an accurate representation of the sequence as these embeddings contain crucial biophysical characteristics [43].

The ProtT5 model is a general PLM, which means it is neither specifically built for T cells nor epitopes. For the prediction discussed in this study, another suitable alternative might be a domain-specific PLM, such as TCR-BERT [47]. However, the research from Lin et al. demonstrated that embeddings from a general PLM performed comparably to embeddings using a domain-specific PLM [48].

One of the primary challenges is the generalizability of the models used to predict the TCR-pMHC binding affinity. Therefore, it was decided to utilize embeddings from a general PLM as the input for the epitope, TCR CDR3 α, and TCR CDR3 β sequences.

To reduce the computational time required for training, the embeddings are pre-computed and mapped to the corresponding entries within the PyTorch Dataset at the outset of the model’s training process. This process is repeated for each dataset, thus resulting in a single iteration for the train, test, and validation datasets.

4.1.8 Physicochemical Properties - Additional incorporated Information

In addition to the amino acid embeddings of the corresponding epitope, TCR CDR3 α and TCR CDR3 β chain information, a pre-computed 101-dimensional vector of physicochemical properties is provided to specific models. Each element of the vector corresponds to a feature. Some examples of those features are QSAR [49], BLOSUM indices, Kidera factors, VHSE-scales, and others. These properties are computed using the entire amino acid sequence of the corresponding epitope, TCR CDR3 α and TCR CDR3 β. The sequences are passed to the “peptides” Python package. A similar method was employed in the research by Goffinet et al. [17].

A substance's physical and chemical qualities are referred to as its physicochemical properties. These characteristics pertain to different facets of proteins, including their physical characteristics and chemical makeup or interactions. Understanding protein structure, function, stability, and interactions necessitates understanding these characteristics [13].

Prior to transferring these physicochemical properties to the model, the values are adjusted to a range between $[-1, 1] = \{x \in R \mid -1 \leq x \leq 1\}$. To prevent the leakage of data, the features of the validation and test datasets are scaled using the scalers obtained from the train dataset. This is done to prevent the inadvertent transfer of information from the train dataset to the validation or test datasets.

4.2 Insights into the Datasets

All source datasets included data from different species, for example mice and apes. This study is limited to human data. Furthermore, the study is limited to MHC class I only. This is partly due to the lack of MHC class II datapoints and partly due to the existence of biological differences.

4.2.1 Comparing MHC Classes

MHC class I molecules present peptides from endogenous sources and MHC class II molecules present peptides from exogenous sources [50]. The way binding occurs and how the molecules are structured is not the same for the two classes, although they are similar. MHC class I and MHC class II both comprise two chains. While only one chain is variable in MHC class I, both chains are variable in the second class. No polymorphisms are known for the non-variable chain of MHC class I, Beta2Microglobulin [51]. The ability of MHC class II molecules to bind peptides of variable lengths is a significant factor limiting the accuracy of prediction tools [52].

4.2.2 Comparison of Precision in V/J Regions and MHCs Between Genes and Alleles

In the context of biology, a gene is defined as a segment of deoxyribonucleic acid (DNA) that encodes a functional product, typically a protein. Genes are responsible for various structural and functional aspects of all living organisms, including the immune system components. An allele is a variant form of a gene. Therefore, the allele is a more precise value. The presence of different alleles can result in variations in the protein products, which in turn affect the phenotype of an organism [13].

In the context of immunobiology, these variations are crucial, as they contribute to the diversity of immune responses, allowing organisms to combat a wide range of pathogens effectively. For instance, the MHC is highly polymorphic. The genes of such organisms have a multitude of different alleles, each of which encodes a molecule that presents a different peptide to T cells [13].

4.2.3 Datasets for experiments

This study developed four distinct datasets to investigate the binding interactions of TCRs with peptides presented by the MHC. Specifically, two datasets were constructed for beta chain only data, one enriched with precision gene annotations and the other with precision allele annotations. Similarly, two additional datasets were compiled for paired chain data, again distinguishing between precision gene and precision allele annotations. Each of these datasets can be employed with or without incorporating physicochemical properties, thereby facilitating the implementation of flexible approaches in the predictive modelling of TCR-peptide interactions.

4.2.3.1 Beta chain datasets

The two datasets, which contain only the β-chain information, have the following attributes:

TRBV: V-Region of the β-chain, plays a key role in TCR diversity (section 2.4)

TRBJ: J-Region of the β-chain, plays a key role in TCR diversity (section 2.4)

TRB_CDR3: CDR3 sequence of β-chain, an important segment of the variable chains in TCRs

MHC: Molecule which presents the epitope

Epitope: Part of an antigen recognized from T Cells

4.2.3.2 Paired chain datasets

The two paired chain datasets, which contain the α- and β-chain information, have the same attributes as the beta chain only dataset, but additionally they have the following α chain specific informations:

TRAV: V-Region of the α-chain, plays a key role in TCR diversity (section 2.4)

TRAJ: J-Region of the α-chain, plays a key role in TCR diversity (section 2.4)

TRA_CDR3: CDR3 sequence of α-chain, an important segment of the variable chains in TCRs

4.2.4 Exploratory Data Analysis

A total of four different datasets have been created, each of which can be used with or without physicochemical properties. Two datasets for beta chain only differ in precision, one at the gene level and one at the allele level. The same applies to the two paired chain datasets.

4.2.4.1 Overview

In Table 3, an overview of the datasets is provided. The data sourced from IEDB, McPas-TCR and VDJdb will be examined. This means that negative examples are currently not considered, as they are artificially generated.

Table 3: Overview Datasets

	Paired Gene	Paired Allele	Beta Gene	Beta Allele
Rows	48161	52167	179822	199492
Missing cells	22.8%	21.8%	19.7%	19%

The small difference in the number between genes and alleles datasets comes from duplicate elimination. Gene precision contains less information, resulting in more duplicates. The number of missing cells is illustrated in Figure 5. The CDR3 and epitope information is never missing and, therefore, not interesting in this diagram. TRAV and TRAJ are paired specific attributes.

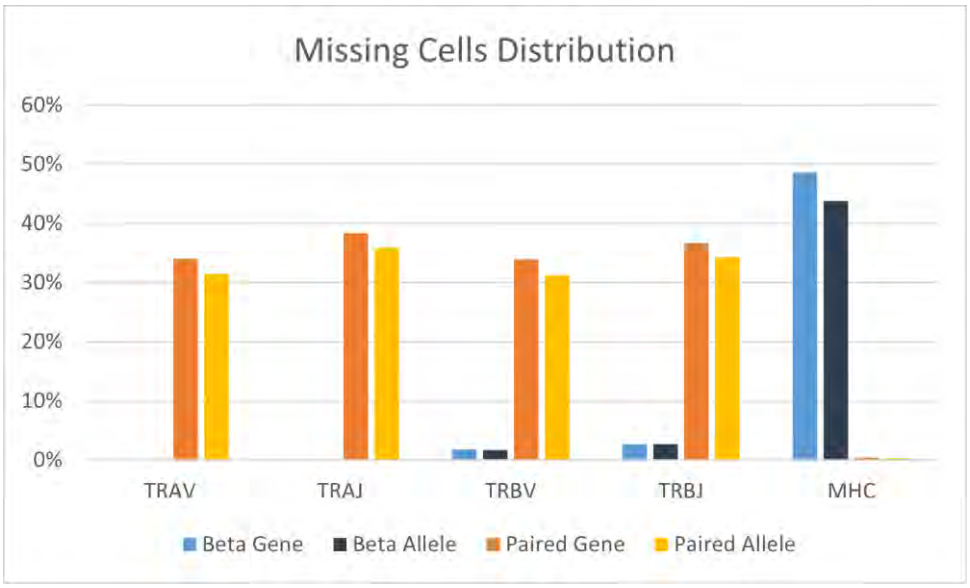


Figure 5: Missing Cells Distribution

The diagram illustrates that the paired data lacks V/J information, and beta lacks MHC data.

4.2.4.2 Distinct Values

The number of distinct values per attribute varies significantly. TRAV, TRAJ, and TRA_CDR3 are paired specific attributes. The V/J and MHC attributes exhibit such a limited range of distinct values that they are not discernible in Figure 6.

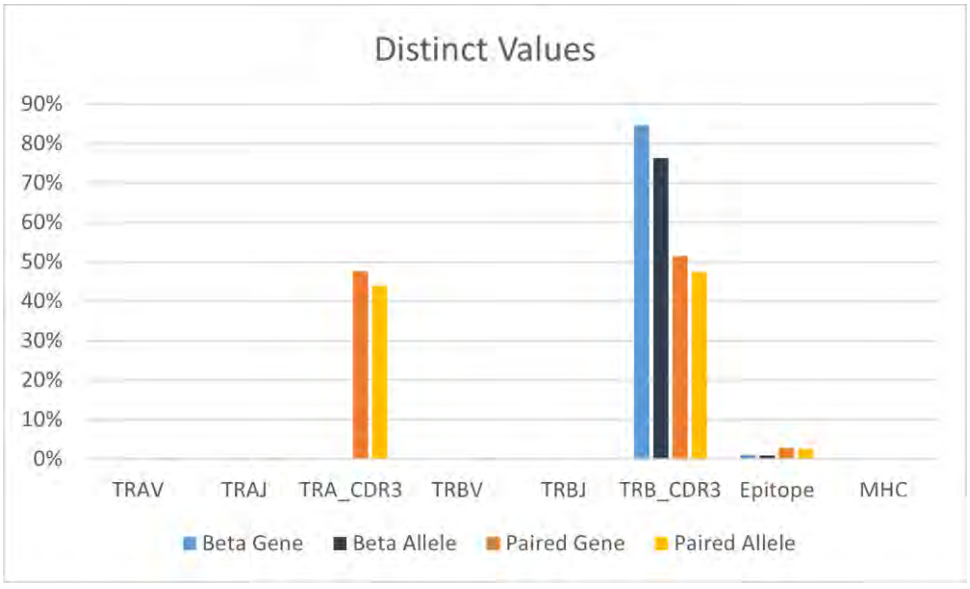


Figure 6: Distinct Values Distribution

Nevertheless, it is evident that there are more distinct CDR3 values than epitopes.

4.2.4.3 Attributes Distribution

The frequency of occurrence and whether the attributes offer a wide variety of values or whether some values must be repeated have been established. The question then arises as to the distribution of the individual attribute values. Additional diagrams are included in Appendix B: EDA, whenever needed.

CDR3

In the paired dataset, the CDR3 region is available for both the α- and β-chains. Table 4 shows an overview of the distribution.

Table 4: CDR3 distribution

		Paired Gene	Paired Allele	Beta Gene	Beta Allele
Distinct CDR3 Sequences	α-chain	22971	22971	-	-
	β-chain	24803	24803	152160	152160
Distinct CDR3 Sequences	α-chain	47.7%	44.0%	-	-
	β-chain	51.5%	47.5%	84.6%	76.3%

No comparison can be made between the paired and beta datasets for the α-chain. However, it is noteworthy that the paired dataset has less variation for the β-chain CDR3 sequences than the beta dataset.

In Figure 7, the x-axis represents a list of all the distinct CDR3 sequences, sorted by the number of occurrences, and on the y-axis, the corresponding count:

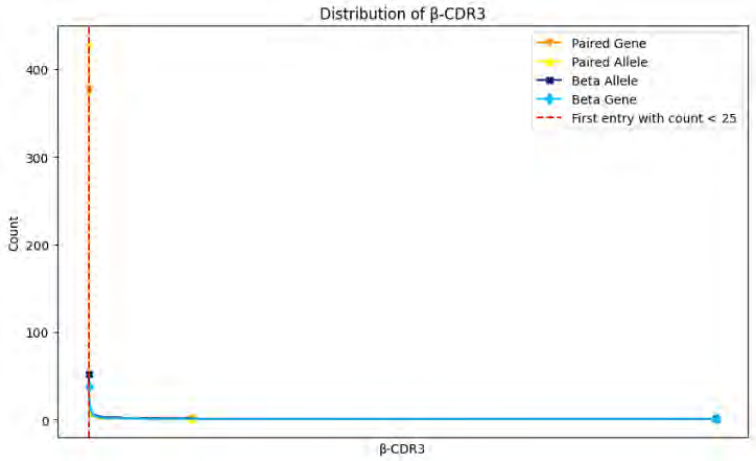


Figure 7: β-CDR3 distribution shown as graph

The two lines from the beta dataset extend from the far left to the far right, while the two paired lines extend from the far left to the first quarter. This is because of the different dataset sizes. The plot shows a large gap between the most frequent β-CDR3 sequences of the paired and beta datasets. The distribution of the beta dataset is more balanced than the distribution of the paired dataset. This can be seen even better in Figure 8:

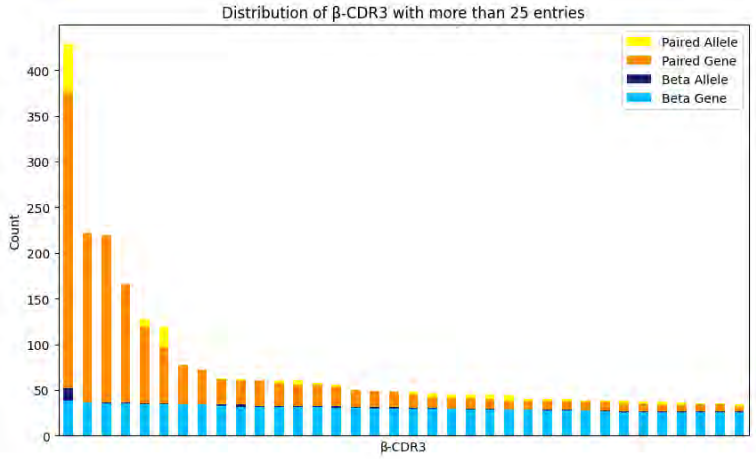


Figure 8: β-CDR3 distribution cropped shown as histogram

The paired dataset contains a few CDR3 sequences that occur with high frequency, the values in the beta dataset are more evenly distributed.

In the diagrams of the α-chain, the most frequent CDR3 sequence appears around 250 times. Nevertheless, the distribution looks similar and can be found in the Appendix B: EDA, section 10.2.1.1.

V/J

The V and J regions are significantly more represented in the beta dataset, as seen by the number of missing cells in Table 5. Also, there are way more distinct V-Regions than distinct J-Regions.

Table 5: V/J distribution

		Paired Gene	Paired Allele	Beta Gene	Beta Allele
Distinct V-Region	α-chain	48	112	-	-
	β-chain	60	128	67	165
Distinct V-Region	α-chain	0.2%	0.3%	-	-
	β-chain	0.2%	0.4%	<0.1%	0.1%
Missing Cells V-Region	α-chain	34%	31.5%	-	-
	β-chain	33.9%	31.3%	1.8%	1.7%
Distinct J-Region	α-chain	55	111	-	-
	β-chain	14	28	16	31
Distinct J-Region	α-chain	0.2%	0.3%	-	-
	β-chain	<0.1%	0.1%	<0.1%	2.7%
Missing Cells J-Region	α-chain	38.4%	35.9%	-	-
	β-chain	36.7%	34.3%	2.7%	2.7%

Figure 9 shows the distribution of the V-Region values. The x-axis describes the values and the y-axis the corresponding number of occurrences.

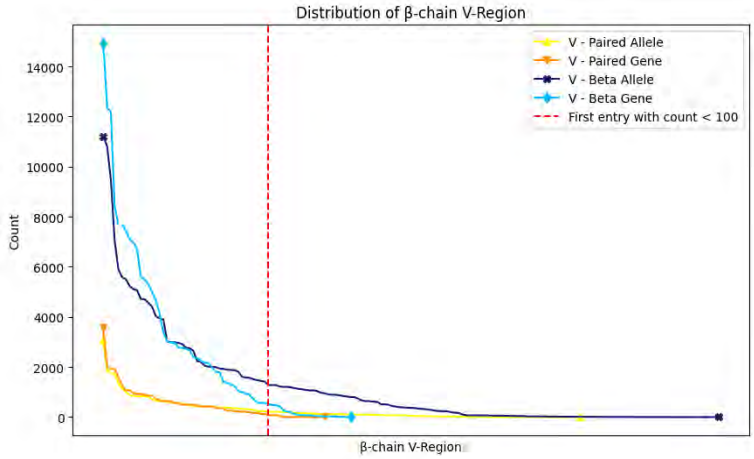


Figure 9: β-chain V-Region distribution shown as graph

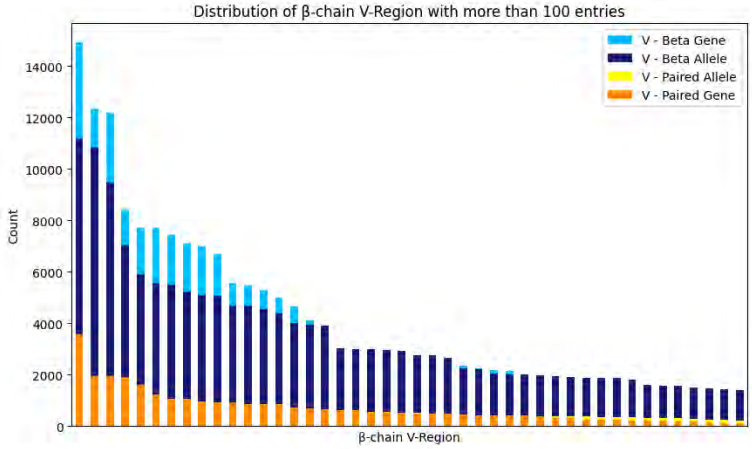


Figure 10: β-chain V-Region distribution cropped shown as histogram

In Figure 9 the distribution is suboptimal. On the left-hand side of the graph, the frequency of the values declines rapidly. This left-hand area, which ends at the red line, is shown as a histogram in Figure 10.

Now about the J-Region. The most frequent entry, in this case, occurs almost 40000 times. That's more than twice the amount of the most frequent entry of V-Region. In this case, too, the curve drops very quickly, especially in the beta datasets, as illustrated in Figure 11.

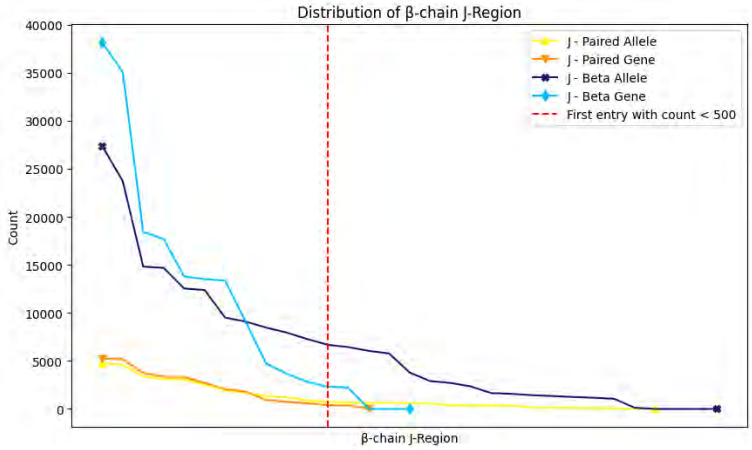


Figure 11: β-chain J-Region distribution shown as graph

A detailed view of the left side of the red line is given in Figure 12.

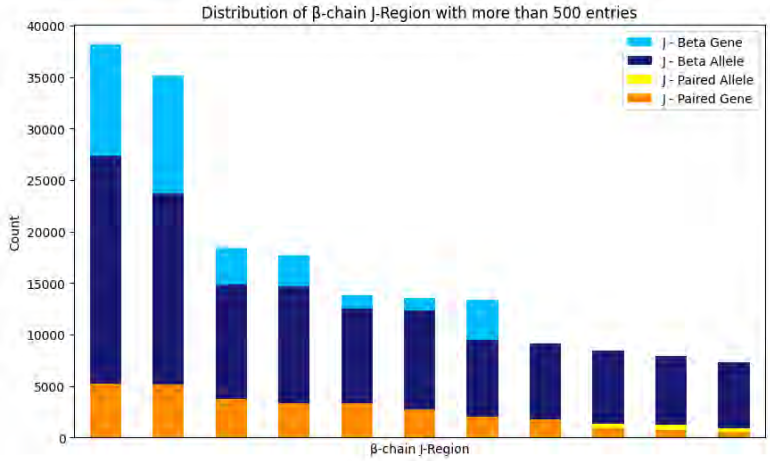


Figure 12: β-chain J-Region distribution cropped shown as histogram

The paired data seems to have a more balanced distribution, but as shown in Figure 5, the paired datasets are also missing more than 30% of the J-Region cells.

Another investigation option is the combination of variables. It would be beneficial to ascertain whether the values are distributed systematically and whether they consistently occur in pairs. Since the two paired datasets each have values for the α- and β-chain, there are six pie charts

together with the two beta datasets. Consequently, two of them are shown, as the other four are very similar. The remaining graphs are added to the Appendix B: EDA, section 10.2.2.

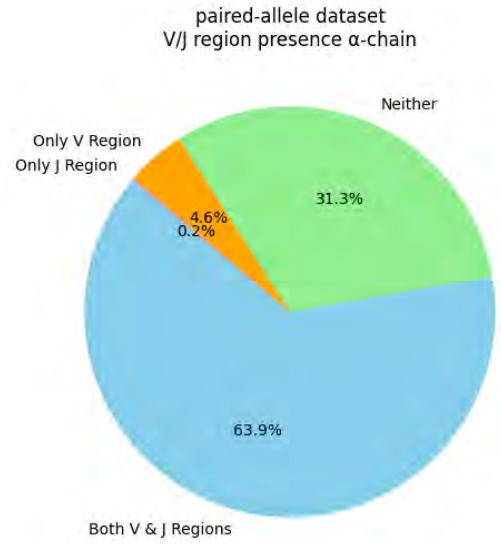


Figure 13: α-chain V/J region presence

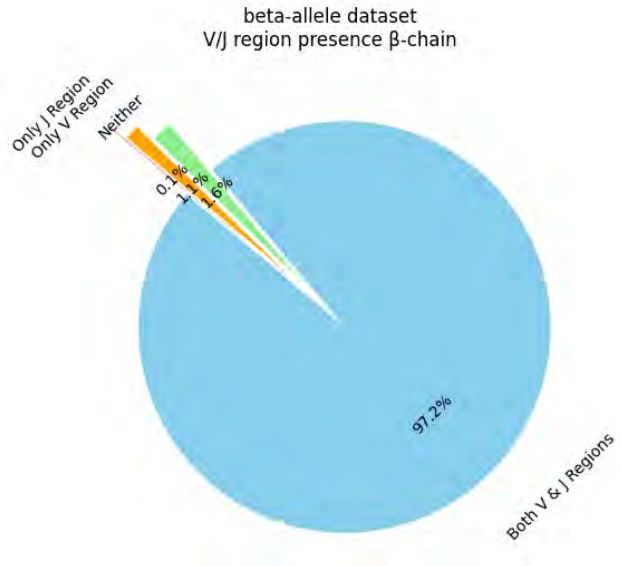


Figure 14: β-chain V/J region presence

The pie charts, Figure 13 and Figure 14, again show that the beta dataset covers the V/J values much better. The proportion where only either the V or the J region is set is small; usually, both are set or none.

MHC

Table 6 illustrates the difference in the number of missing MHC values between the paired and beta datasets. It also shows that four MHC genes are identical in both datasets, and the allele datasets contain variations of these four genes.

Table 6: MHC distribution

	Paired Gene	Paired Allele	Beta Gene	Beta Allele
Distinct MHC	4	78	4	111
Missing cells	0.4%	0.2%	48.6%	43.8%

Two illustrations are necessary to examine the distribution of MHC because the number of different values between gene and allele precision differs significantly.

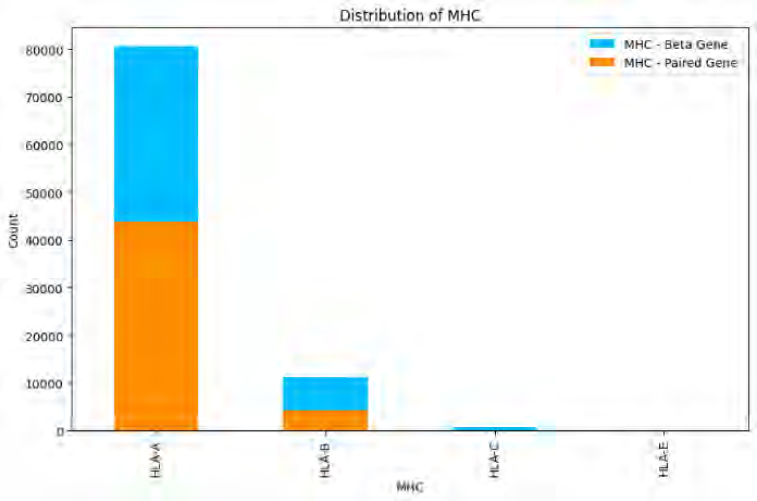


Figure 15: MHC Gene distribution

In the gene dataset, as illustrated in Figure 15, the proportions between the four MHC values for beta and paired appear quite similar. The MHC gene HLA-A is overrepresented, whereas HLA-E is only present eight times in both datasets. HLA-C is found 49 times in the paired dataset and 753 times in the beta dataset. The distribution of the allele datasets is illustrated in Figure 16.

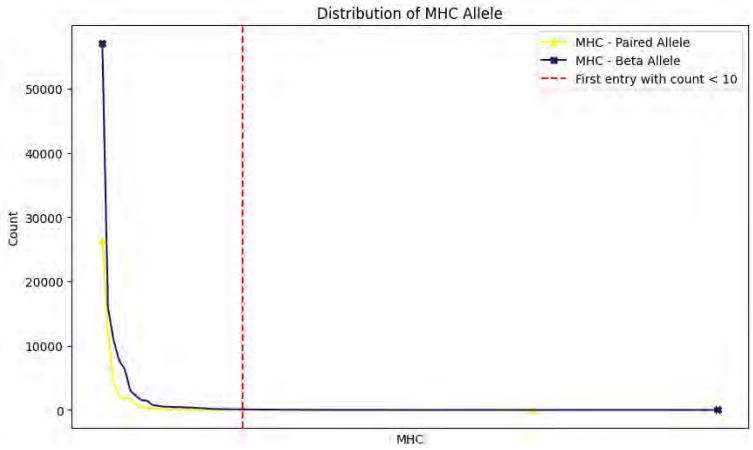


Figure 16: MHC Allele distribution shown as graph

A closer look to the left-hand side of the red line is provided in Figure 17.

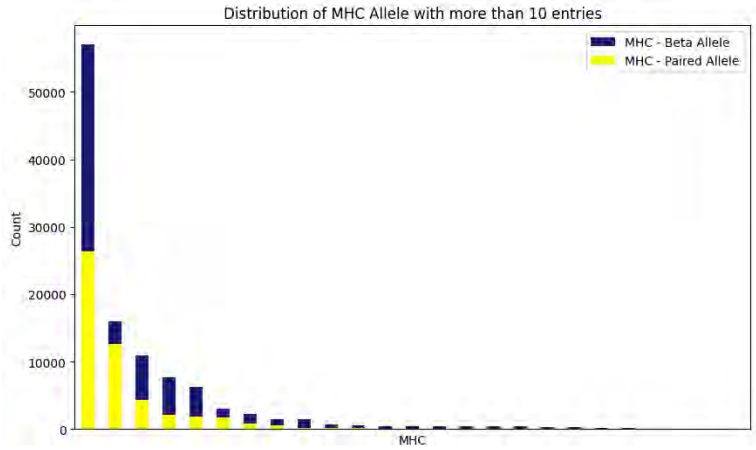


Figure 17: MHC Allele distribution cropped shown as histogram

The two plots for the allele datasets, Figure 16 and Figure 17, show that the variations of the HLA-A genes are not very well distributed. The two alleles “hla-a*02:01” and “hla-a*03:01” account for 75% of all alleles contained in the paired dataset and 65% in the beta dataset. Across all datasets, there is a consistent pattern of a few highly frequent sequences followed by a tail of rare sequences.

Epitope

The epitope is present in every data point. The distinction between the gene and allele datasets resides solely in the number of entries, as the gene datasets exhibit a smaller size than the allele datasets. The number of distinct epitopes stays the same for gene and allele precision.

Table 7: Epitope distribution

	Paired Gene	Paired Allele	Beta Gene	Beta Allele
Distinct Epitopes	1377	1377	1890	1890

The comparison of the distribution between beta and paired dataset is evident. The graph in Figure 18 shows that although the paired dataset is much smaller, the most frequently occurring epitope occurs almost 8000 times more than the most frequently occurring epitope in the beta dataset.
The most frequently occurring epitope in the paired dataset occurs in about half of all entries. This epitope is also the second most frequently occurring epitope in the beta dataset. The most occurring epitope in the beta datasets occurs in about 9% of all entries. In both cases, there are a few epitopes that occur very often, and the others are very rare. This is very evident in the paired dataset and can be recognized even better in Figure 19, which shows the left part up to the red marking in Figure 18.

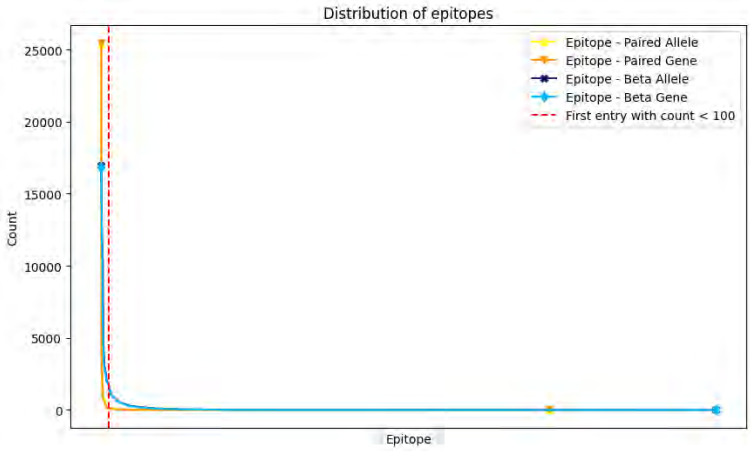


Figure 18: Epitope distribution shown as a graph

The two bars on the far left in Figure 19 represent the two alleles "hla-a*02:01" and "hla-a*03:01." It is evident that these two alleles are overrepresented in the datasets.

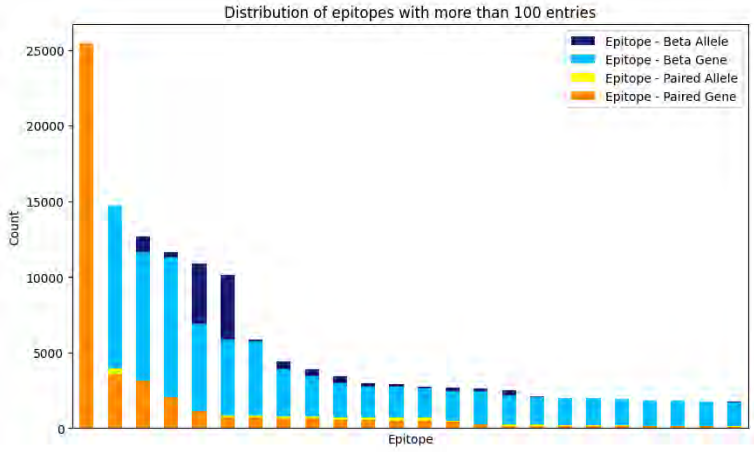


Figure 19: Distribution of the most represented epitopes as a histogram

4.2.4.4 Negative Samples

Until now, only the positive data have been subjected to analysis. This is defined as the entries of TCR and epitopes that bind. However, a machine learning binary classification model also needs negative examples. The negative examples have been created individually for the train, validation and test dataset, as explained in section 4.1.6. The distribution of the attributes is similar to those of the positive data, with the exception of MHC and epitope data. The ratio between positive and negative data points should be 1:1, in example the same number of positive and negative data points. The distribution was nearly achieved, although minor discrepancies remain unexplained, as evidenced by Table 8. Train, Validation and Test data are concatenated for this purpose.

Table 8: Negative data distribution

	Paired Gene	Paired Gene Test	Paired Gene Validation	Paired Gene Train	Paired Allele	Paired Allele Test	Paired Allele Validation	Paired Allele Train
rows positive only	48161	7226	7224	33711	52167	7826	7826	36515
rows positive only (%)	50.27	50.13	50.09	50.35	50.24	50.14	50.14	50.28
rows negative only	47635	7189	7199	33247	51673	7783	7782	36108
rows negative only (%)	49.73	49.87	49.91	49.65	49.76	49.86	49.86	49.72
rows all	95796	14415	14423	66958	103840	15609	15608	72623
	Beta Gene	Beta Gene Test	Beta Gene Validation	Beta Gene Train	Beta Allele	Beta Allele Test	Beta Allele Validation	Beta Allele Train
rows positive only	179822	26974	26973	125875	199492	29925	29922	139645
rows positive only (%)	50.03	50.08	50.08	50.01	50.05	50.04	50.06	50.05
rows negative only	179581	26884	26886	125811	199089	29873	29849	139367
rows negative only (%)	49.97	49.92	49.92	49.99	49.95	49.96	49.94	49.95
rows all	359403	53858	53859	251686	398581	59798	59771	279012

MHC and Epitope Distribution

The distribution of these attributes changes because the MHC and epitopes are exchanged for each positive data point to create a negative data point. The distribution diagrams look similar, but the number and order of the values have changed. For example, the epitope “KLGGALQAK” is in first place in the positive dataset and “GILGFVFTL” is in second place. This order is reversed for the negative dataset.

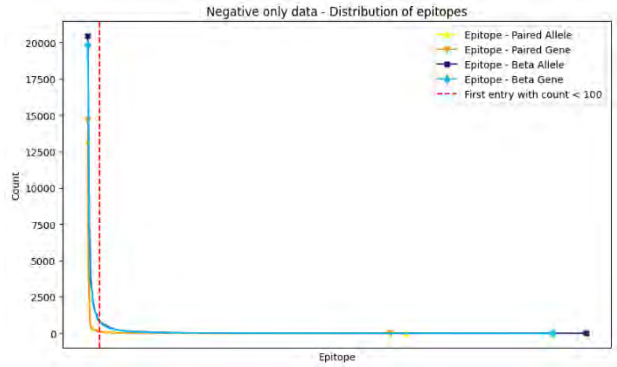


Figure 20: Epitope distribution of negative only data shown as graph

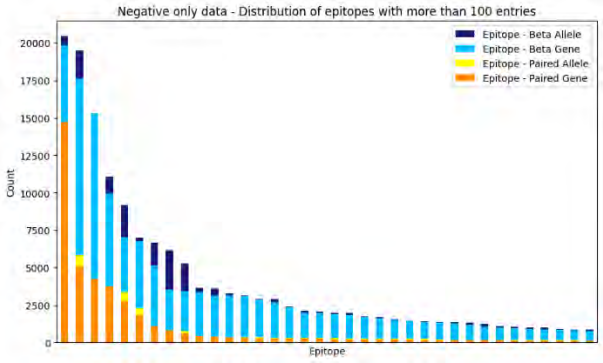


Figure 21: Epitope distribution of negative only data cropped shown as histogram

Figure 20 and Figure 21 can be compared to Figure 18 and Figure 19. It remains the case that a small number of epitopes occur with very high frequency, while many epitopes occur with low frequency. It can be observed that the first epitope, which occurs less than 100 times, is present in the negative data at a later point in time. This is the reason because Figure 21 has more bars than Figure 19. Furthermore, the most frequently occurring epitope in the paired datasets has been observed to occur 15000 times rather than the previously reported 25000 times. In contrast, the number of the most frequently occurring epitopes in the beta dataset is exhibiting a slight upward trend. The shapes of the positive and negative data are similar, with the only difference being the order in which they appear.

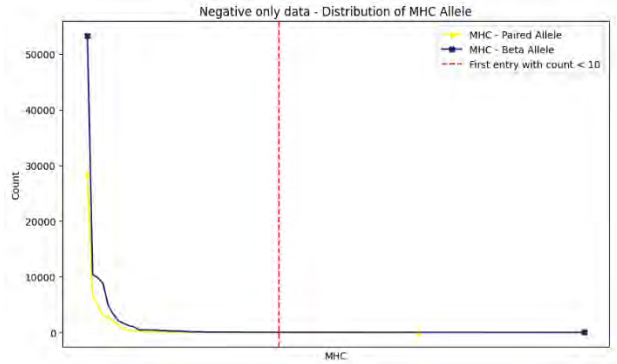


Figure 22: MHC Allele distribution of negative only data shown as graph

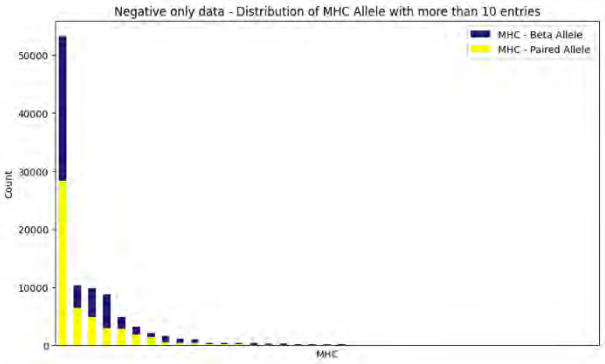


Figure 23: MHC Allele distribution of negative only data cropped shown as histogram

As illustrated in Figure 22 and Figure 23, the distribution of MHC alleles exhibits only minimal variation, compared to Figure 16 and Figure 17. No differences could be identified in the distribution of MHC genes, except that the total number is slightly smaller. This is because there is slightly less negative than positive data.

Duplicated rows

The negative examples are generated after the dataset has been split into three distinct subsets: train, validation, and test. This process has resulted in the creation of overlapping duplicates. Upon examination of the datasets in isolation, it is evident that no duplicates are present within the Train, Validation, and Test subsets. However, upon merging these subsets, duplicates emerge.

Unfortunately, there are also duplicates that differ only in the binding property. This is an error and can occur in this way, for example. In the train dataset, a negative sample is created by exchanging the MHC and epitope properties, as explained in section 4.1.6. However, the new combination appears in the test dataset as a positive sample. It is then clear that the negative sample is not negative and should not appear in any dataset. These incorrectly declared samples are declared as incorrect negative samples in Table 9.

Table 9: Duplicated data

	Paired Gene	Paired Allele	Beta Gene	Beta Allele
distinct duplicates	281	248	577	538
all duplicates	554	489	1147	1068
incorrect negative samples	64	52	189	103

The initial row in Table 9 describes how many different duplicated samples occur. The second row indicates the number of affected rows, which means the number of all duplicated samples. The third row describes the number of negative rows that are not correct and shouldn't be in the dataset. These are also duplicates if the binding property is ignored. Note that all distinct duplicates and all incorrect negative samples are a part of the "*all duplicates*" row.

4.2.4.5 TPP Task Samples

The analysis considers the final dataset, including negative and positive samples. Table 10 lists the number of task samples in the test datasets.

Table 10: TPP Task samples distribution

Number of Entries	TPP1	TPP2	TPP3	TPP4
Paired Gene	5879	7816	546	174
Paired Gene Reclassified	9015	4680	412	308
Paired Allele	7434	7955	145	75
Paired Allele Reclassified	10949	4440	100	120
Beta Gene	25602	27699	437	120
Beta Allele	29311	30308	132	47

Figure 24 illustrates the distribution of samples in the different TPP tasks. The sample ratio between TPP1 and TPP2 samples is approximately 1:1. The two paired reclassified datasets represent an exception because of the reclassification, which leads to less TPP2 and more TPP1 samples, as explained in section 4.1.5.1. Furthermore, only a few samples are in the TPP3 and TPP4 tasks. This is due to the lack of distinct epitopes, which prevents the creation of many samples for the TPP3 and TPP4 tasks.

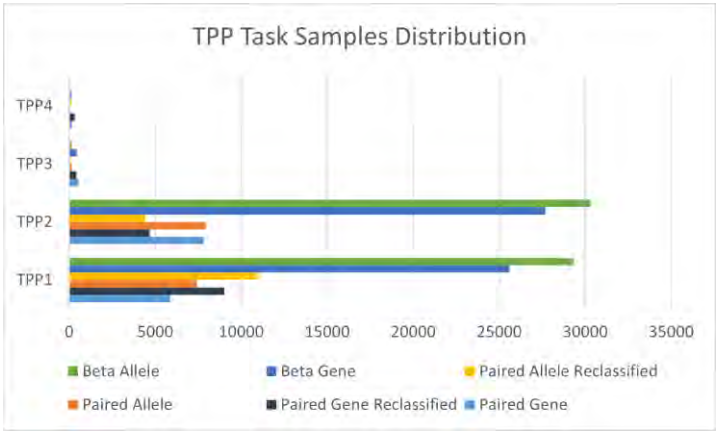


Figure 24: TPP Task samples distribution

In addition to examining the number of entries for each task, it is also important to assess how the unseen data for different tasks varies in similarity to the seen data. All values in the train and validation dataset are declared as seen. The minimum Levenshtein distance to the seen CDR3 values is determined for each unseen CDR3 value in the test dataset. The objective is to ascertain whether the unseen CDR3 sequences in TPP2 and TPP3 samples exhibit the same Levenshtein distance to the seen CDR3 sequences.

Table 11: CDR3 Levenshtein distances

Mean Levenshtein distance (CDR3)	Paired Gene	Paired Allele	Beta Gene	Beta Allele
TRA TPP2	1.19	1.17	-	-
TRA TPP3	1.34	1.12	-	-
TRA TPP2 (reclassified)	1.67	1.69	-	-
TRA TPP3 (reclassified)	1.65	1.50	-	-
TRB TPP2	1.83	1.76	1.95	1.91
TRB TPP3	2.14	1.88	1.91	1.92
TRB TPP2 (reclassified)	2.39	2.37	-	-
TRB TPP3 (reclassified)	2.36	2.18	-	-

In Table 11, the Levenshtein distance is the mean of the minimum distances between the CDR3 values from test dataset, compared to the epitopes from the train and validation datasets. This is particularly interesting because the TPP2 and TPP3 tasks both have unseen TCR. The difference between them is that TPP2 has seen epitope, while for TPP3, the epitope is unseen. The beta dataset has no reclassified test set and no α-chain, so some values are missing.

Due to the proximity of the values, it is challenging to draw conclusions from the table. A boxplot is generated for each test dataset to facilitate more robust statements, showcasing the smallest Levenshtein distance to the seen CDR3 sequences. The boxplots in Figure 25 and Figure 26 show that certain data points in the paired test dataset have a Levenshtein distance of zero. This is no longer the case with the reclassified paired test dataset. The difference between the normal paired classification and the reclassified case is explained in chapter 4.1.5.1.

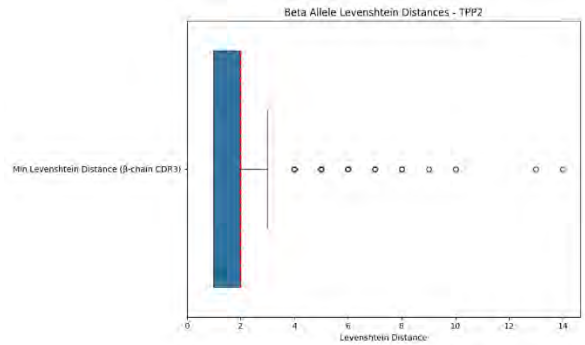


Figure 25: Boxplot Levenshtein CDR3 - TPP2 Beta Allele

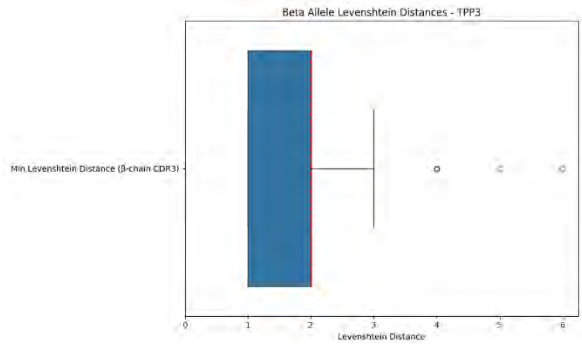


Figure 26: Boxplot Levenshtein CDR3 - TPP3 Beta Allele

With one exception, the TPP2 and TPP3 tasks differ in the number of upward outliers. The exception is the Levenshtein distance of the α-chain of the normally classified paired gene dataset, where TPP2 differs from TPP3 not only in the number of outliers but also in the lower quartile and lower whisker, as can be seen in the following figures Figure 27 and Figure 28.

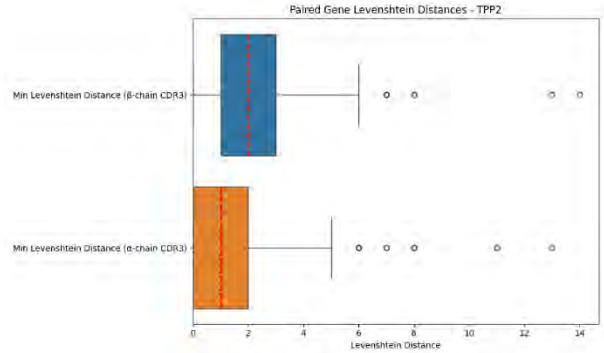


Figure 27: Boxplot Levenshtein CDR3 - TPP2 Paired Gene

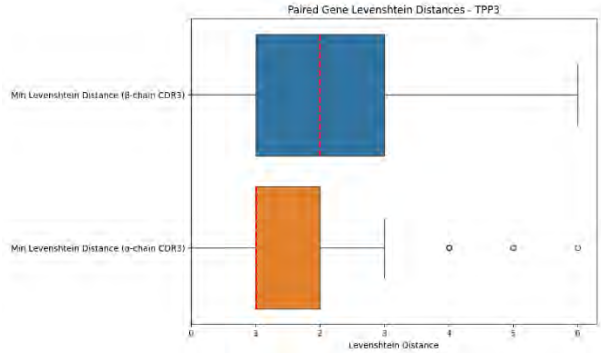


Figure 28: Boxplot Levenshtein CDR3 - TPP3 Paired Gene

A comparison of the normally classified paired datasets with the reclassified paired datasets reveals that the distance between the lower and upper whisker is consistently smaller in the reclassified dataset. Furthermore, the lower whisker or even the lower quartile is always set at zero in the normally classified paired dataset, in contrast to the reclassified dataset which never has a zero. The remaining boxplots are in Appendix B: EDA section 10.2.3.

4.2.4.6 Conclusion

The analysis revealed that the paired datasets exhibited a high degree of missing data in the V/J region, with over 30% of values being absent. In contrast, the beta datasets exhibited a notable degree of missing data in the MHC region, with over 40% of values being absent. In terms of distinct values, the beta datasets exhibit a more balanced distribution of CDR3 sequences compared to the paired datasets. However, attributes such as V/J, MHC, and epitope have fewer distinct values, with MHC primarily composed of the HLA-A class. The process of creating negative samples is flawed, resulting in the generation of negative samples that should be positive. However, the impact of this is mitigated by the relatively low number of flawed samples. Finally, the Levenshtein distance analysis indicates minimal differences, suggesting the potential for improvement by implementing alternative approaches.

4.3 Models

In total eight different models are developed. Four models, referred to here as the Vanilla Models, process all available data without considering physicochemical properties. To utilize these properties, models designated as the Physicochemical Models are created. The various models share a similar architectural framework, which is further elaborated upon in the subsequent sections. Additionally, a differentiation must be done between the models utilizing Gene precision for the V- and J-Regions as well as for the MHC. The combination of these differences resulted in the number of different models.

4.3.1 The Architecture of a Transformer

To capture dependencies between the different input sequences fed into the model, regardless of their distance from one another, the employed design in this thesis is based on the self-attention mechanism. This allows the model to produce output sequences that accurately reflect these dependencies.

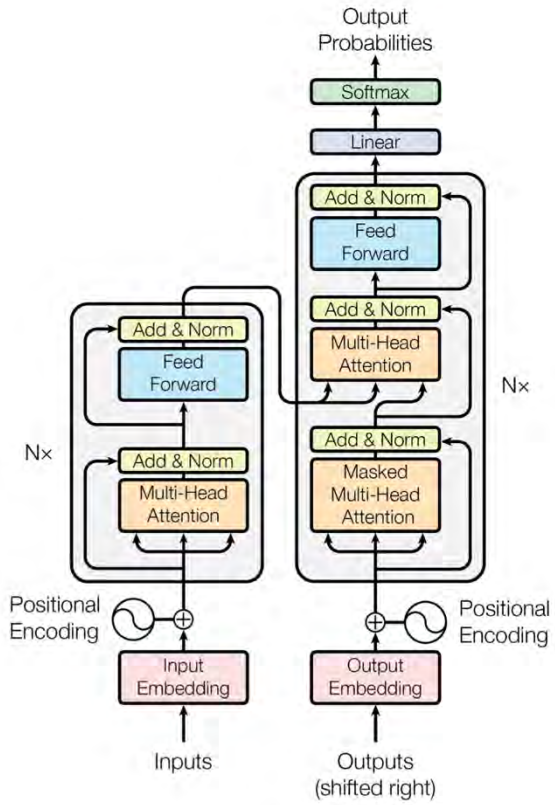


Figure 29: Architecture of the Transformer model: The encoder (left) processes input embeddings with multi-head attention and feed-forward layers. The decoder (right) includes an additional masked multi-head attention to handle shifted output embeddings, generating final output probabilities [10].

Vaswani et al. [10] introduced the self-attention mechanism, which enables the model to compute a series of attention scores to determine the relative relevance of various items in the input

sequence. The model is able to focus on relevant segments of the sequence, regardless of their position, due to the weighted representations of the input elements that are enabled by these scores. By ensuring that even remote dependencies within the sequence may be represented efficiently, this method improves the model's ability to identify complex relationships and patterns [10]. The main components of the Transformer are location-wise feed-forward networks, which apply transformations to each location individually, and multi-head self-attention layers, which enable the model to focus on several portions of the input sequence simultaneously [10]. As illustrated in the accompanying Figure 29 which is derived from the paper that introduced this architectural concept, a transformer can be comprised of an encoder and decoder component [10]. However, in this thesis, only the encoder is utilized, as it is a downstream task in this case of classification.

4.3.2 Regularization Methods to Prevent Overfitting: Dropout and Weight Decay

In machine learning, regularization techniques such as dropout and weight decay are employed to prevent overfitting. Throughout training, dropout randomly "drops out" a fraction of neurons, causing the network to identify and discard redundant representations, thereby enhancing generalization [53]. By adding a penalty to the loss function that is proportionate to the weights' magnitude, weight decay encourages the model to maintain smaller weights, which reduces complexity and improves generalization [53]. The objective of both techniques is to generate a model with better generalization ability for unseen data.

4.3.3 General Architecture

Essentially, the architectural framework utilized is a transformer encoder, as detailed in section 4.3.1. Consequently, as discussed in section 1.2 and compared further in section 3.2.1, the models in this study are categorized within the deep learning paradigm of machine learning. The architecture employs tunable hyperparameters, including the learning rate, the weight decay [54], and the dropout probability rate [53]. Dropout is applied to both the linear fully connected multilayer perceptrons (MLPs) components and within the transformer multi-head attention blocks.

The dropout layers of the linear layers are applied after the final batch normalization layer but prior to the final linear layer. The dropout in the attention heads is applied after the batch normalization and immediately after the activation function, which is ReLU in this case [55]. These architectural decisions regarding the placement of the dropout layer are made in accordance with the guidelines provided by Kim et al. [56].

The EPIC-TRACE model [4] initially provided the conceptual framework for our architecture. In the EPIC-TRACE model, epitope, TCR α, and TCR β chain information are treated separately. The epitope embeddings are then concatenated with the TCR CDR3 α and TCR CDR3 β sequence embeddings, thereby integrating the chain information of the T cell. Building upon this foundation, our model diverges significantly from that of Korpela et al. [4]. The models developed in this thesis

have a single pipeline after the concatenation, in contrast to the three different output heads present in the models of Korpela et al. [4].

By employing a modular architecture, the models are able to accommodate missing values for subsequently concatenated information, such as the V-Region, J-Region, and MHC. These missing values, represented by NumPy "*nan*" values, are assigned corresponding embeddings learned during training. In their forward method, all models obtain epitope embeddings shaped by Σx1024, where Σ corresponds to the length of the longest epitope sequence, referred to as "*max_length*" in following figures of the specific architectures. The TCR α CDR3 region embeddings and the TCR β CDR3 region embeddings are similarly padded to the longest sequence in all distinct train, test, and validation datasets. Subsequently, the aforementioned embeddings are subjected to processing by the Transformer encoders, which yield the relevance of each amino acid within a sequence with respect to all other sequences in the complex and the epitope sequence to which the CDR3 region may be binding.

In addition, all models obtain the index of the corresponding trained embedding of the V-Region, J-Region for each chain and the same MHC embedding for both parallel halves.

Ultimately, the two sides of the model are flattened and then fed through a classifier implemented as a MLP, using residual connections.

4.3.3.1 Explanation of Models Using Less Parameters

This section provides an explanation of the models that utilize fewer parameters.

For the models that employ Gene precision for the three additional embeddings representing the V/J-Region and the MHC, rather than Allele precision, a model with a slightly reduced number of parameters is necessary due to the shorter sequences, which result in smaller embedding layers within the model. This adaptation is performed by the model in a posteriori manner.

4.3.3.2 Additional Information for the Physicochemical Models

Physicochemical models provide additional information regarding the physicochemical properties of the entire complex. For the TCR CDR3 beta, TCR CDR3 alpha, and epitope, this information is inputted. Subsequently, the concatenated sequences are processed by a Transformer block, which enables the capture of the importance of each amino acid with respect to all the others, as well as the physicochemical properties of the complex.

4.3.4 Paired Vanilla Model

The Paired Vanilla Model uses the information about the epitope, TCR CDR3 α, TCR CDR3 β, V and J region and the MHC but not including the physicochemical properties as described in section 4.1.8. It is important to note that this model is designed for processing data where the CDR3 region of both chains, namely TCR CDR3 α and TCR CDR3 β, are available. As this model does not need to handle the numerical inputs of the physicochemical properties the architecture is decently smaller compared to the Physicochemical Model. Essentially it does not have the MLP which

converts the numerical input into embeddings. As one can see in Figure 30 the architecture includes in total four MLPs, where the three used for the embedding of the V and J region as well as for the MHC are the same implemented using the PyTorch Embedding class.
The weights of this Embeddings class are learned during training.The two Transformer Blocks are implemented using multi-head attention [10], layer norm [57], dropout [53] and instantiated with four heads.

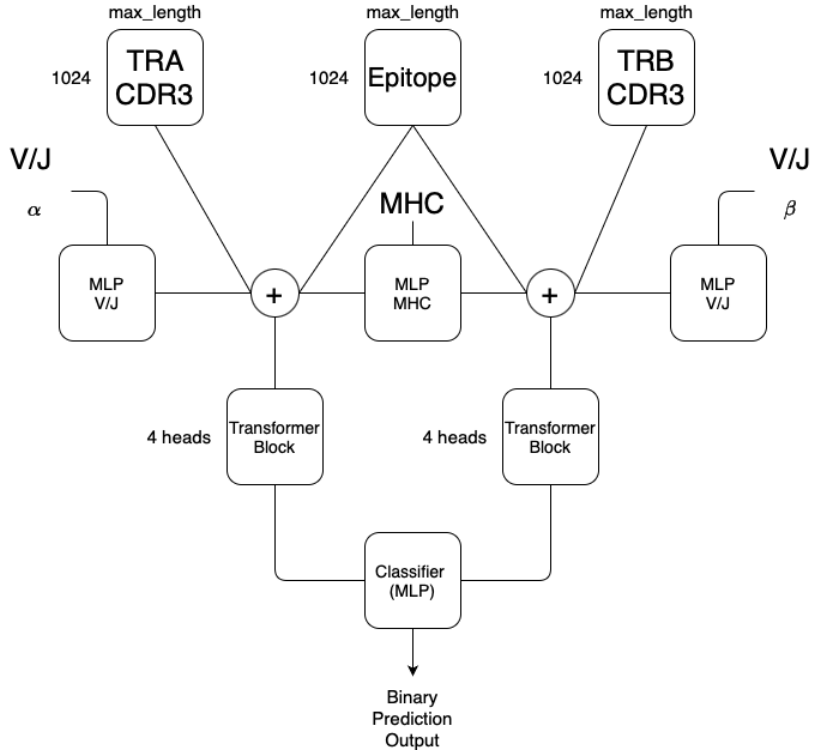


Figure 30: Architecture of the Paired Vanilla model

The hidden dimensions of the linear feed forward part of the Transformer blocks are computed as illustrated in the following equation:

Equation 1: Calculation of hidden dimensions in the Transformer blocks

$$hidden\ dimensions\ =\ 1.5\ \cdot\ 1024$$

1024 is the dimensionality of the embeddings. The previously mentioned hidden dimensions of the Transformer block are calculated in an identical manner for every model, regardless of its specific variation. Consequently, no further sample formula is required for the other models, as they are computed in an identical manner.
The input for the Classifier which essentially a MLP is flattened. This implies that the input dimensionality of the Paired Vanilla model variation is calculated as follows:

Equation 2: Calculation of input dimensionality for the Classifier block in a Paired Vanilla model

$$input\ dimensions\ =\ 4\ \cdot\ (max\_length\ +\ 3)\ \cdot\ 1024$$

Nevertheless, this varies for each model variation, in contrast to the hidden dimensions seen in Equation 1. This block incorporates batch normalization [58], dropout [53]. In total this model has 15.8 million trainable parameters for the Gene variation and 16.1 million for the Allele version of the model.

4.3.5 Beta Vanilla Model

The Beta Vanilla model is in its nature identical to the Paired Vanilla model described in section 4.3.4, however, the Beta Vanilla model includes solely information about the TCR CDR3 β, V region, J region and MHC sequences. The hidden dimensions of the transformer blocks are defined the same way. As the TCR CDR3 α is not included in this model, the input dimensions to the Classifier block are calculated slightly different:

Equation 3: Calculation of input dimensionality for the Classifier block in a Beta Vanilla model

$$input\ dimensions = 2 \cdot (max\_length + 3) \cdot 1024$$

As it is a Vanilla model, also no physicochemical properties are included as one can see in Figure 31 which illustrates the detailed architecture. The benefit of this model is that as mentioned in section 4.1.2 considerably more data samples are available for the TCR CDR3 β only dataset.

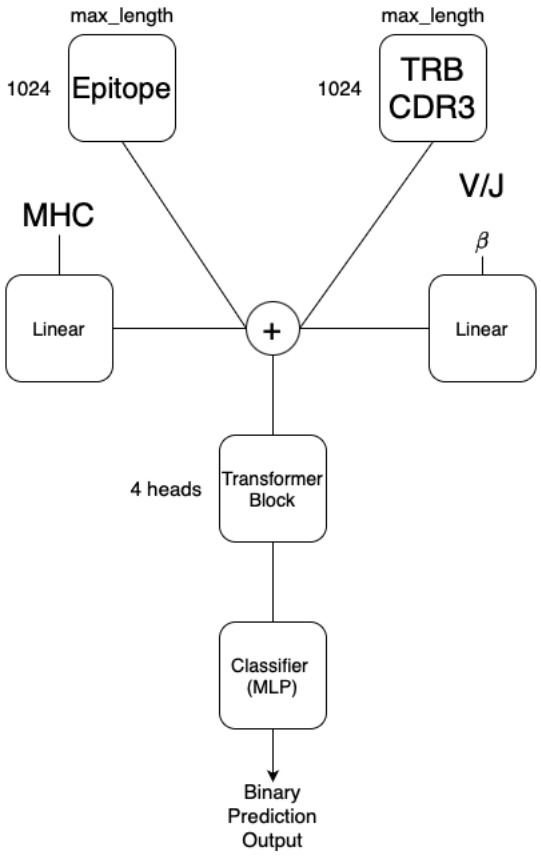


Figure 31: Architecture of the Beta Vanilla Model

Nevertheless, this model uses the same regularization techniques as previously mentioned so for example batch normalization [58], dropout [53]. This model has in total 13.3 million trainable parameters for the Gene version of the model and 13.5 million parameters for the Allele model.

4.3.6 Paired Physicochemical Model

The Paired Physicochemical Model extends the Paired Vanilla model described in section 4.3.4. by incorporating additional properties. Like the Paired Vanilla model, it uses the epitope, V/J-Regions, and MHC information, along with both TCR CDR3 α and TCR CDR3 β regions as input. In addition to these inputs, the Paired Physicochemical Model includes the physicochemical properties of the epitopes and the TCR CDR3 α and TCR CDR3 β regions.

These additional physicochemical properties are processed by a MLPs, called "*Physico MLPs*". This is illustrated in detail in Figure 32 as one can see in the input of the "Physico MLPs".

The embeddings resulting from the "Physico MLPs" are concatenated with the embeddings obtained from the ProtT5 PLM. The concatenated information is then passed to a transformer encoder with two attention heads. This configuration enables the model to assess the significance of each amino acid within the input sequence in relation to all others, considering the physicochemical properties that describe the entire complex. The purpose of the subsequent Transformer Encoder with four Attention Heads is to capture the relationship between the pre-processed sequences of the TCR chains and the physicochemical properties of both the TCR chains and the potential binding epitope. This allows the model to contextualize the interactions between these components, integrating the physicochemical properties.

The Gene version of this model has 45.7 million trainable parameters, and the Allele version of the same architecture has 46.0 million parameters. The number of trainable parameters for this architecture is greater than that of other architectures due to the use of five Transformer blocks in total.

In addition to the three distinct input sequence heads, this model exhibits physicochemical properties for each. Consequently, the input dimensions of the classifier block are calculated in a distinctive manner:

Equation 4: Calculation of input dimensionality for the Classifier block in a Paired Physicochemical model

$$4 \cdot (max\_length + 4) \cdot 1024$$

The calculation of the hidden dimensions of the Transformer blocks remains identical to that described in section 4.3.4 and illustrated in Equation 1.

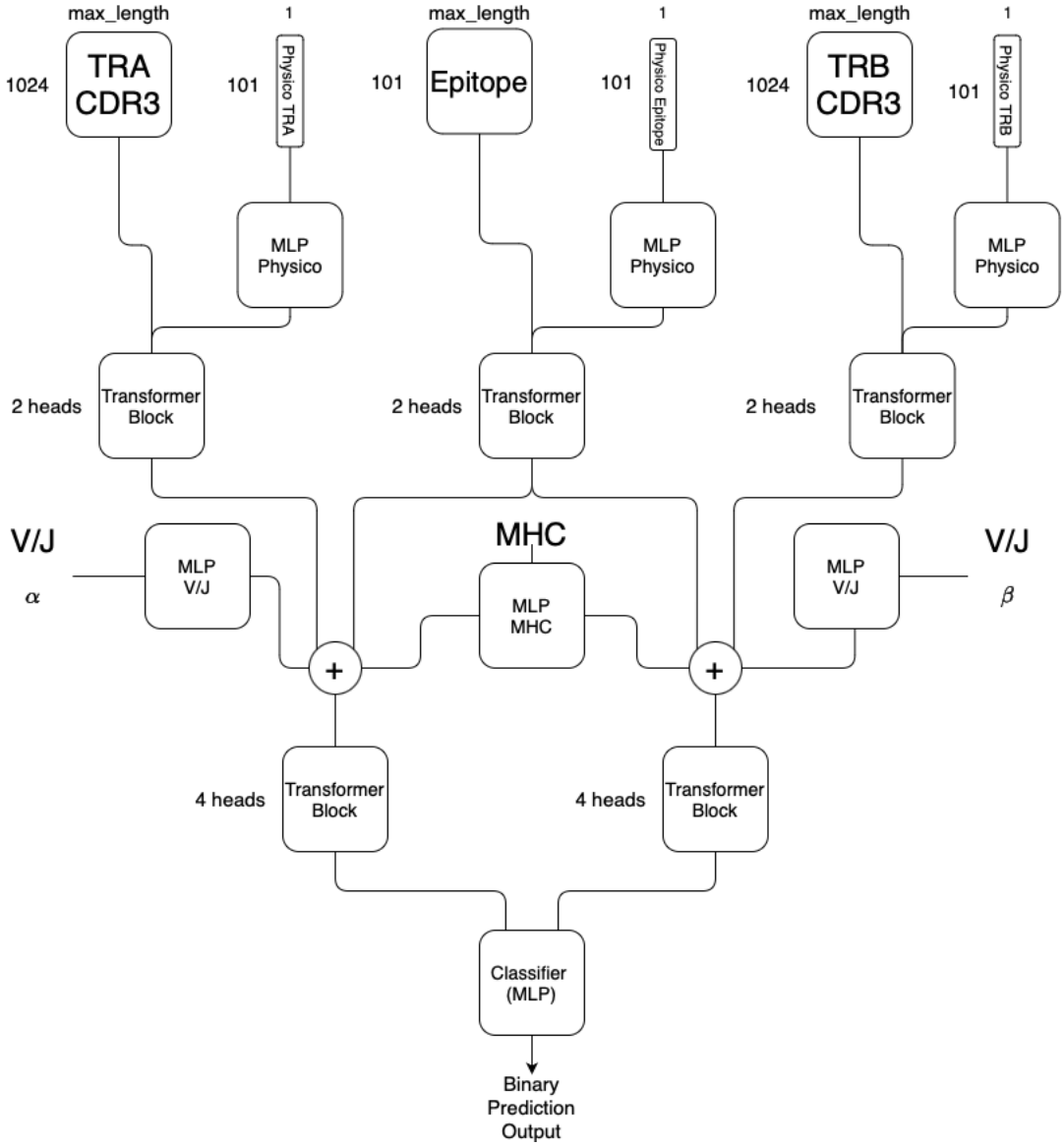


Figure 32: Architecture of the Paired Physicochemical Model

4.3.7 Beta Physicochemical Model

The Beta Physicochemical Model incorporates physicochemical information, as described in section 4.3.6, in a manner analogous to the Paired Physicochemical Model. However, in the model described in this section, only the TCR CDR3 β is considered, as it is a beta-only model.

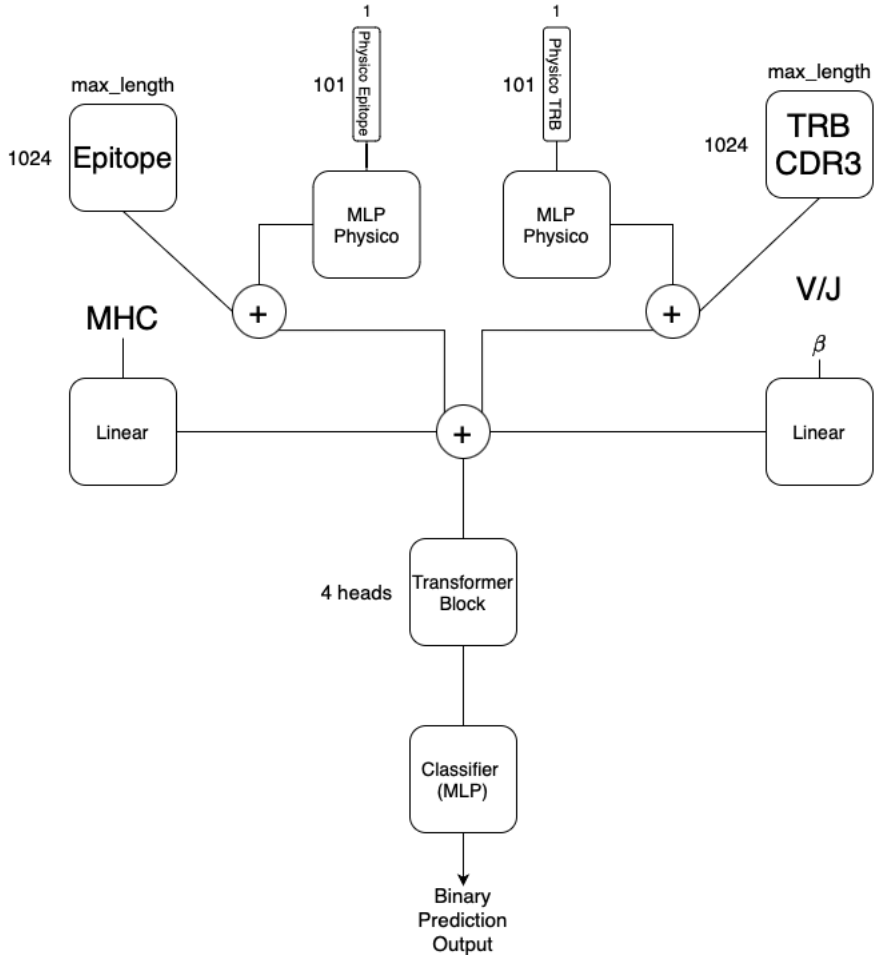


Figure 33: Architecture of the Beta Physicochemical model

Consequently, the TCR CDR3 α and their corresponding physicochemical properties are not included. Additionally, epitope information is used. The model architecture depicted in Figure 33 provides insight into the way the information is utilized and integrated within the machine learning model.

In this variation of the model, the dimensionality of the classifier is calculated as follows:

Equation 5: Calculation of Input Dimensionality for the Classifier Block in a Beta Physicochemical Model

$$input\ dimensions = 2 \cdot (max\_length + 4) \cdot 1024$$

13.6 million trainable parameters are trained for the Gene precision variation of this model, and 13.8 million are trained for the Allele variation.

4.4 Evaluation of the Models

The models are validated using the area under the receiver operating characteristic curve (ROC AUC) and the previously mentioned average precision (AP). These metrics are implemented by the torch package and utilized in the validation phase, respectively, at the conclusion of the training process following the evaluation of the model in the test step. These metrics were selected for their established use in the field, as evidenced by references [3], [4], [19]. Moreover, both metrics are meaningful in evaluating the model's ability to distinguish between binding and non-binding. This because AP combines the recall of the model as well as the precision as depicted in Equation 6:

Equation 6: Mathematical formulation of the average precision (AP) metric

$$\mathrm{AP} = \sum_{\mathrm{n}=1}^{\mathrm{N}} (\mathrm{R_n} - \mathrm{R_{n-1}})\mathrm{P_n}$$

Parameters in Equation 6:

- N is the number of thresholds,
- $\mathrm{P_n}$ is the precision at the n-th threshold,
- $\mathrm{R_n}$ is the recall at the n-th threshold, and
- $\mathrm{R_{n\text{-}1}}$ is the recall at the n-1-th threshold.

The plotting of the true positive rate (sensitivity) against the false positive rate (one-specificity) at varying threshold levels enables the ROC AUC to assess the efficacy of a binary classification model [59] as one can see in Equation 7:

Equation 7: Mathematical formulation of the ROC AUC metric

$$\mathrm{ROC\ AUC} = \int_0^1 \mathrm{TPR(t)}\ \mathrm{d}\big(\mathrm{FPR(t)}\big)$$

Parameters in Equation 7:

- $\mathrm{TPR(t)}$ is the true positive rate at threshold t, and
- $\mathrm{FPR(t)}$ is the false positive rate at threshold t.

The evaluation metrics were calculated individually for each task and then determined as a global value as an average for the respective metric of the respective model.

4.5 Training of the Models

All models are implemented using the PyTorch Lightning framework, which is built on top of PyTorch. This framework helps prevent code duplication and provides convenient functions to facilitate the training script. Nevertheless, to maintain the application code as modularized as

possible, a distinct training script, model script, and dataset class are implemented for each model variation described in the section.

The loss function is defined as the binary cross-entropy loss seen in Equation 8. This loss function is particularly well-suited to binary classification tasks, as it effectively measures the discrepancy between the predicted probabilities and the actual class labels. This makes it an optimal choice for scenarios where the outputs are probabilities that must be closely aligned with either 0 or 1. The equation of this loss function is provided in Equation 8:

Equation 8: Binary Cross-Entropy Loss function

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

Parameters in Equation 8:

- N is the number of samples,
- y_i is the true label of the i-th sample (0 or 1) and
- p_i is the predicted probability for the i-th sample.

To ensure reproducibility and trackability of the experiments, which are the training runs, the Weights and Biases platform is used. From there, the datasets are automatically downloaded when starting a run. Furthermore, all hyperparameters are logged, which is essential for ensuring the reproducibility of the different training. For all datasets and, therefore, for all model variations except the Beta Vanilla variations (both precision levels), a batch size of 128 is employed. For the Beta Vanilla models a batch size of 256 is used.

The choice of optimization method, either Adam or stochastic gradient descent (SGD), is dependent on the Bayesian optimization approach employed during hyperparameter tuning [60], [61].

Following each epoch, a validation step is undertaken to monitor the learning with respect to the validation dataset. The test phase is conducted once after the final training epoch to assess the model and provide an evaluation based on the two metrics AP and ROC AUC.

To achieve more effective generalization, stochastic weight averaging (SWA) is enabled during training with a 10% learning rate, which is derived from the learning rate utilized in the initial 45 epochs. SWA, a technique that averages different locations along the trajectory of SGD, is an optimization strategy that employs a cyclical or constant learning rate [62]. SWA leverages SGD's capacity to sample the network's loss surface at a cyclical or continuous learning rate, in addition to weight averaging [62]. The method produces stochastic weights, which assist in the identification of wider optima, thereby leading to better generalization [62].

As a final regularization technique, early stopping is employed with a patience of five epochs. This means that if, after five epochs, no improvement in the global AP value evaluated on the validation dataset is achieved, the training process is terminated.

4.6 Handling the Different Tasks in the Hyperparameter Tuning Process

To automate the hyperparameter tuning of the individual models, the Weights and Biases Sweeps are used. Bayesian hyperparameter optimization was selected as the optimization method. For detailed insights, please refer to the .yaml files in the GitHub repository. The metric to maximize during the process is the AP over all tasks referred to as global AP.

As this thesis differentiates between distinct prediction tasks based on whether the sequences are seen or unseen (as detailed in section 4.1.3), one might typically hyperparameter tune for a specific task, such as TPP3. However, the objective of this thesis is to examine the model's performance regarding including TCR CDR3 beta only or both chains, additional physicochemical properties and the precision of the V/J-Region and the MHC. Consequently, it was determined due to time limitations and restriction in computational resources that hyperparameter tuning should be conducted based on the global average precision (AP) metric.

5 Results

This section presents and illustrates the outcomes of each experiment conducted in a comprehensive manner. It includes detailed analyses of the performance metrics, comparisons between different models, and the impact of various factors such as paired and beta-only data, physicochemical properties, gene, and allele precision aiming to provide a thorough understanding of the experimental results.

5.1 Overview of the Performance of the Different Models

The following subsections provide an overview of the performance of the models. A total of eight models were trained and evaluated. The number of models is determined by the combination of paired and beta models, with the inclusion or exclusion of physicochemical properties and the precision level of gene or allele, respectively. The first table presents a general overview of all trained models. The global values for both metrics in the following tables are calculated using the test dataset.

5.1.1 Relativization of the Comparison between Different Models

The comparisons between the different models presented in this thesis must be considered in the context of their inherent limitations. This is because the test datasets upon which the models were tested, for instance between Beta Vanilla Gene and Paired Vanilla Gene, were disparate. This complicates direct comparisons between these models. A detailed discussion and reasoning behind this matter is provided in 6.1.

5.1.2 General Overview of the Performance of the Models

As illustrated in Table 12 models incorporating physicochemical properties exhibit inferior performance relative to those that do not. Moreover, the comparison reveals that in most cases and tasks, the Paired models outperform those that utilize only TCR CDR3 β sequences, despite the suboptimal performance of the Paired Physicochemical model. This aligns with previous research indicating that incorporating information from both the TCR CDR3 β and the TCR CDR3 α sequences enhances performance [19], [26].

In contrast, the content of the Table 12 shows that Beta Vanilla in combination with the Allele precision achieves the best TPP3 performance over all model combinations, the Gene precision models included.

A further insight from the performance evaluation is that models including physicochemical properties perform worse than those that do not include them.

However, it is important to note that these comparisons involve different test datasets, making direct comparison challenging as mentioned in 5.1.1.

Table 12: Summary of performance metrics across model variations

Gene Precision				
Paired Vanilla Model	global	TPP1	TPP2	TPP3
Binary Average Precision (AP)	0.8469	**0.9366**	0.8035	0.7432
ROC AUC	0.8664	**0.9477**	0.8234	0.7229
Paired Physicochemical Model	global	TPP1	TPP2	TPP3
Binary Average Precision (AP)	0.6029	0.6341	0.5943	0.6207
ROC AUC	0.6552	0.6863	0.6409	0.6133
Beta Vanilla Model	global	TPP1	TPP2	TPP3
Binary Average Precision (AP)	0.7517	0.8139	0.7644	0.8103
ROC AUC	0.7474	0.8049	0.7848	0.6558
Beta Physicochemical Model	global	TPP1	TPP2	TPP3
Binary Average Precision (AP)	0.7535	0.8148	0.7601	0.8043
ROC AUC	0.7490	0.7977	0.7735	0.6390
Allele Precision				
Paired Vanilla Model	global	TPP1	TPP2	TPP3
Binary Average Precision (AP)	**0.8954**	0.9230	**0.8751**	0.8132
ROC AUC	**0.9009**	0.9288	**0.8785**	0.7581
Paired Physicochemical Model	global	TPP1	TPP2	TPP3
Binary Average Precision (AP)	0.6443	0.6683	0.6219	0.7147
ROC AUC	0.6777	0.7069	0.6494	0.6503
Beta Vanilla Model	global	TPP1	TPP2	TPP3
Binary Average Precision (AP)	0.8115	0.8063	0.8163	**0.9352**
ROC AUC	0.8367	0.8303	0.8485	**0.8283**
Beta Physicochemical Model	global	TPP1	TPP2	TPP3
Binary Average Precision (AP)	0.8059	0.8006	0.8107	0.9195
ROC AUC	0.8254	0.8259	0.8251	0.7932

The optimal values are presented in bold font within the table. Appendix C: Run Name to Performance Mapping Table provides the corresponding run name, which leads to the specific performance.

5.1.3 Comparison of the Performance Between Beta-Only and Paired Models

A detailed comparison is presented between models that utilize solely the information pertaining to the CDR3 region of the β-chain and those that incorporate data from the CDR3 of both the β- and the α-chain. As the performance of the Paired Physicochemical models are below par, only the Vanilla models are considered in the average metric calculation. The best values for each metric and task in Table 13 are presented in bold font. The values presented in Table 13 represent the mean values. For instance, the value for the Paired Models, TPP1, ∅AP (0.9298), is calculated as the Paired Vanilla Model TPP1 Gene value (0.9366) added with the Paired Vanilla Model TPP1 Allele value (0.9230), and then divided by the number of samples (here two) to obtain the mean value. It should be noted that the global values are not included in the calculation of these values, as they already represent a kind of mean value. The standard deviation of the corresponding values can be observed on the right side. All the raw data can be extracted from Table 12.

Table 13: Comparison of the mean performance values between the Paired and the Beta models

Paired Models	global	TPP1	TPP2	TPP3
∅ AP	**0.8712** ± 0.0343	**0.9298** ± 0.0096	**0.8393** ± 0.0506	0.8193 ± 0.0495
∅ ROC AUC	**0.8837** ± 0.0244	**0.9383** ± 0.0134	**0.8510** ± 0.0390	0.7405 ± 0.0249
Beta Models	**global**	**TPP1**	**TPP2**	**TPP3**
∅ AP	0.7816 ± 0.0423	0.8101 ± 0.0054	0.7904 ± 0.0367	**0.8728** ± 0.0883
∅ ROC AUC	0.7921 ± 0.0631	0.8167 ± 0.0180	0.8167 ± 0.0450	**0.7421** ± 0.1220

A comparison of the Paired and Beta models, as illustrated in Table 13, reveals that the Paired models outperform the Beta models in the tasks TPP1 and TPP2. This is evidenced by the bold font marking the best performances of the corresponding tasks. Notwithstanding, the Beta model demonstrates superior performance in the TPP3 task, which is the most generalized task, as all sequences are unseen during the training phase of the model. In the TPP3 task, the Beta models exhibited a mean AP performance that was 6.53% higher than that of the Paired models. It is important to note that this comparison utilizes different test datasets, which should be considered when interpreting the results. Consequently, direct comparison between the Paired and Beta models is challenging due to the distinct test datasets used as mentioned in 5.1.1.

5.1.4 Comparison of the Performance Between Gene and Allele Precision Models

Due to the suboptimal performance of the Paired Physicochemical models, all Physicochemical models were excluded from this comparison. Consequently, this overview pertains solely to the Vanilla models, encompassing the Genes and Alleles variants. Thus, an average value and the standard deviation of the combination of the Allele models of the Beta and Paired models were determined, as were the same calculations for the gene models of the Paired and Beta variants.

Table 14: Comparison of the mean performance values between the Gene and the Allele models

Gene Models	global	TPP1	TPP2	TPP3
∅ AP	0.7993 ± 0.0673	**0.8753** ± 0.0868	0.7840 ± 0.0276	0.7768 ± 0.0474
∅ ROC AUC	0.8069 ± 0.0841	0.8763 ± 0.1010	0.8041 ± 0.0273	0.6894 ± 0.0474
Allele Models	**global**	**TPP1**	**TPP2**	**TPP3**
∅ AP	**0.8535** ± 0.0593	0.8647 ± 0.0825	**0.8457** ± 0.0416	**0.8742** ± 0.0863
∅ ROC AUC	**0.8688** ± 0.0454	**0.8796** ± 0.0697	**0.8635** ± 0.0212	**0.7932** ± 0.0496

As illustrated in Table 14, the models that incorporate Allele precision exhibit superior performance when compared to the Gene models. In the TPP1 task evaluated with the mean AP, the Gene Models exhibit superior performance. Notably, there is a considerable disparity in performance observed on both metrics for the TPP3 task. In this scenario, the Allele models demonstrated an average improvement of 12.54% in comparison to the mean results obtained from the AP evaluation of the gene models.

It is significant to remember that various test datasets were used in this comparison, which should be taken into account when evaluating this results.

5.1.5 Comparison of the Performance Between Physicochemical and Vanilla Models

Due to the significant differences in performance between the Paired Physicochemical models and Beta Physicochemical models, these variations were not included in the previous comparisons.

Now, the Physicochemical models are evaluated to compare them with the Vanilla models, which do not consider physicochemical properties.

Table 15: Comparison of the mean performance values between the Vanilla and the Physicochemical models

Vanilla Models	global	TPP1	TPP2	TPP3
∅ AP	**0.8264** ± 0.0605	**0.8700** ± 0.0694	**0.8148** ± 0.0458	**0.8255** ± 0.0800
∅ ROC AUC	**0.8379** ± 0.0658	**0.8779** ± 0.0708	**0.8338** ± 0.0397	**0.7413** ± 0.0719
Physicochemical Models	**global**	**TPP1**	**TPP2**	**TPP3**
∅ AP	0.7017 ± 0.0942	0.7295 ± 0.0916	0.6968 ± 0.1050	0.7648 ± 0.1275
∅ ROC AUC	0.7268 ± 0.0769	0.7542 ± 0.0680	0.7222 ± 0.0915	0.6740 ± 0.0810

Table 15 demonstrates that the Vanilla models consistently outperform the Physicochemical models across all tasks. Moreover, the greater standard deviation of the Physicochemical models indicates that the performance of the models varies more between the Physiochemical variations than with the Vanilla models.

Each of these mean values are evaluated as an average across all combinations of the test datasets, making this comparison more reliable. The inclusion of every test dataset variation ensures a comprehensive and balanced evaluation.

5.2 Importance of Hyperparameter Tuning

In the initial stages of the thesis, the possibility of omitting the hyperparameter tuning process was considered due to the constraints imposed by limited temporal and computational resources.

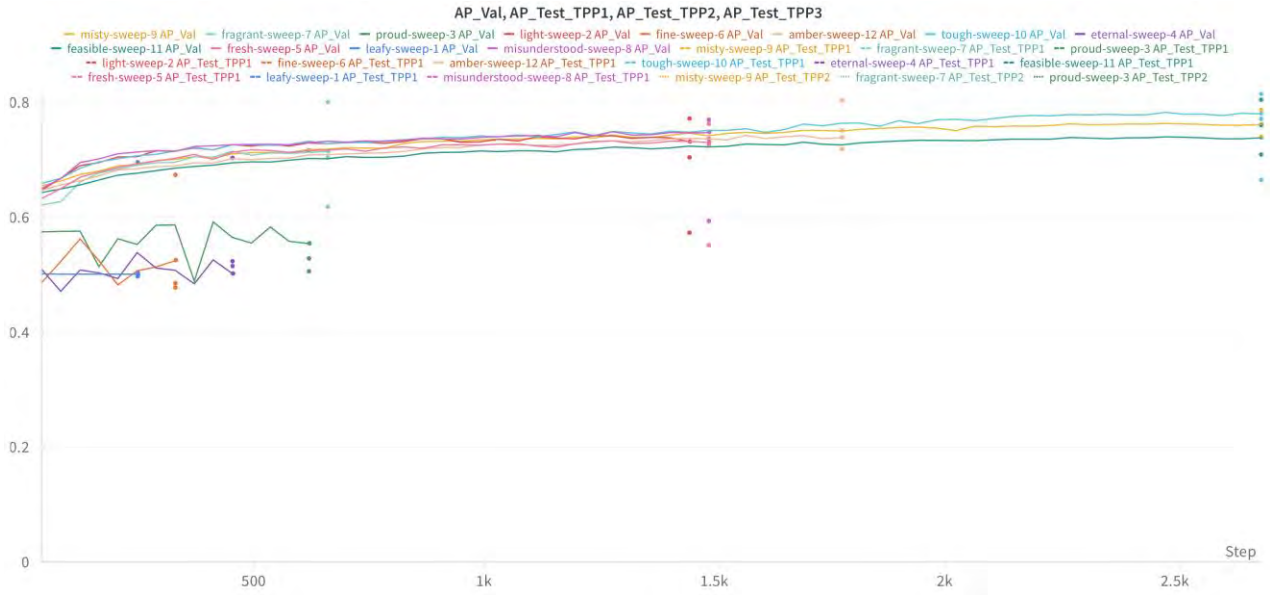


Figure 34: Graphical representation of the learning process: Average Precision (AP) over time of the Beta Physicochemical model with Gene precision

Nevertheless, Figure 34 elucidates the critical importance of hyperparameter tuning within the context of the data and model utilized in this thesis. As examples, the different runs of the Beta Physicochemical during the hyperparameter tuning are used in Figure 34. In the x-axis, the number of training steps is illustrated, while on the y-axis, one can see the “AP_Val” metric evaluated on the validation dataset, resulting in a graph. The AP metrics are corresponding to each task calculated on the test dataset (AP_Test_TPP1, AP_Test_TPP2 and AP_Test_TPP3). Every run of the hyperparameter tuning process of the Beta Physicochemical with Gene precision is depicted in Figure 34. The worst performance on the global AP evaluated with the test dataset has the fine-sweep-5 run with 0.4808. In contrast, the best performance has the misty-sweep-9 with 0.7535 for the global AP on the test dataset. The performance of the misty-sweep-9 is 56.66% superior to that of the worst run, indicating the need for hyperparameter tuning. The other models exhibited a comparable pattern.

5.3 Observations on Optimizer Selection

Notable to mention is that best models of each subcategory are achieved using the SGD optimizer [61] instead of Adam [60]. This observation implies that Adam has a lack of generalization, which was as well observed by a paper released in 2021 [63]. This could be circumvented by tuning the

inner parameters of the Adam optimizer, which would result in higher computational effort as one would have to tune three parameters, namely ε, β_1 and β_2 [60]. Nevertheless, Adam is frequently selected when hyperparameter tuning is not performed, as it functions effectively without tuning of these three hyperparameters [60].

6 Discussion

This thesis investigated the possibility of designing a machine learning model that employs deep learning building blocks in its architecture and is capable of making generalized predictions about the binding affinity between T cells and epitopes using currently available open data. Furthermore, experiments were conducted to analyze the impact of utilizing the CDR3 sequences of both the α- and β-chains on the performance of the models. A further differentiation was analyzed with respect to the precision of the gene segments of the V/J-Region as well as for the MHC. A third investigation was also made with experiments to show the influence of the additional physicochemical properties of the protein complexes.

6.1 Explanation of Using Different Test Datasets Between Different Models

When comparing the various models discussed in this thesis, it is crucial to consider the inherent constraints of each model. Notably, differences in the datasets used for testing, such as between Paired Vanilla Allele and Beta Vanilla Allele, pose significant challenges. Additionally, direct comparisons with other state-of-the-art (SOTA) models are not feasible. To achieve such comparisons, either the models presented in this thesis would need to be tested using the test data from other studies, or the SOTA models would need to be evaluated using the test dataset from this thesis.

Two different test datasets were chosen for the Paired and Beta models. The Beta model benefits from significantly more available data for the beta chain, while the Paired models include additional information, which previous work has shown to improve model performance [19], [26]. However, the Paired model cannot be tested using only beta chain sequences, necessitating distinct handling for these variations. To evaluate the Beta model, using the comparatively small test dataset of the Paired model may not yield meaningful results due to its limited size relative to the training dataset. The same consideration applies to the comparison between the Allele and Gene test datasets. The split algorithm described in section 4.1.5 has not been reliably tested to create two test datasets that differ solely in the V, J region and MHC precision. Moreover, it has not been proven that the epitope and TCR sequences are identical, further complicating direct comparisons between these models.

Additionally, the majority of SOTA models are evaluated using cross-validation. This approach is not employed in this thesis due to limitations in time. Nevertheless, a single-time split was conducted in accordance with the explanations provided in section 4.1.5.

6.2 Analysis of the Performance Using a Reclassified Paired Dataset

It is important to note that when evaluating the Paired models, they consider both the information about the TCR CDR β and the CDR3 α sequences. For the TPP3 task, a T cell is considered as "unseen" if both CDR3 sequences are not observed in the same combination. However, one could

argue that one region of the T cell declared as unseen might be seen during the training either in the training dataset or the validation dataset. To tackle this, a new dataset called "*reclassified*" was created where both the TCR CDR β and the TCR CDR α sequences are unseen, or in other words not included in the validation or training dataset.

Due to the constraints of time, the investigation done with the "*reclassified*" test dataset is limited to the Paired Physicochemical models.

A comparative analysis of the performance of the models evaluated on the Gene and Allele datasets is presented in Table 16. The test dataset, which has been previously utilized in all previous evaluations, is referred to as the "*normal*" test dataset. The "*reclassified*" dataset represents the scenario in which both CDR3s are unseen, rather than just the combination.

Table 16: Comparison between Paired Physicochemical model performances Evaluated on the normal and reclassified test dataset

Gene		
Paired Physicochemical Model	TPP3 (normal)	TPP3 (reclassified)
AP	0.6207	0.6900
ROC AUC	0.6133	0.5692
Allele		
Paired Physicochemical Model	TPP3 (normal)	TPP3 (reclassified)
AP	0.7147	0.8129
ROC AUC	0.6503	0.6298

Because of the reclassification of the TPP3 task, the sequences contained in the others (TPP1 and TPP2) changed too. For completeness they are included in Appendix C: Run Name to Performance Mapping Table but not further investigated.

The hypothesis was that the Paired Physicochemical models would perform worse for the reclassified TPP3 tasks than for the normal classified TPP3 tasks. This because as mentioned in section 4.1.5.1, both CDR3 sequences are unseen in the reclassified dataset.

Nevertheless, one can see that the performance is increasing regarding the AP metric. This phenomenon can be attributed to the fact that the sequences selected for the reclassified TPP3 task are, on average, more similar to the sequences observed in the training set than those in the normal test dataset. Therefore, investigations were done about the mean Levenshtein distances compared between the sequences in the reclassified and the initial test dataset, which refute this statement. The results are presented in section 4.2.4.5. The finding there is that the normal dataset has a lower mean Levenshtein distance, indicating that reclassified test dataset contains less similar sequences compared to the normal test dataset.

This implies that the model should perform better when evaluated using the normal test dataset. However, as illustrated in Table 16, the performance evaluated using the reclassified test dataset is not for every metric and model variation worse, which makes the obtained results not entirely explainable with the Levenshtein investigations. Further, more comprehensive analyses must be conducted in future studies.

6.3 Evaluating Model Generalization to Unseen CDR3 and Epitope Sequences

The task corresponding to the unseen CDR3 and epitope sequence is the TPP3. In a paired model, both the CDR3 α and β sequences are incorporated, representing the entire complex. This means that either the combination of both chains is declared as unseen, or neither chain is allowed to be seen. Conversely, in a beta model, which incorporates only the CDR3 β sequence, only the β-chain is unseen. In addition, for both model variations, the epitope sequence is also unseen.

Given the differences in the test datasets used for the various model variations, it is important to approach the following interpretations with caution. As previously mentioned in 5.1.1 and further elaborated in 6.1, direct comparisons are not feasible due to these discrepancies. Consequently, the values of the state-of-the-art (SOTA) model are provided for informational purposes only and should not be considered a definitive benchmark for the models presented in this thesis.

The model exhibiting the highest performance within the Paired models is the Paired Vanilla model, which uses Allele-precision with an AP value of 0.8127 and an ROC AUC value of 0.7581. The Beta Vanilla model, using the Allele-precision methodology, proved to be the most effective overall, with an AP of 0.9352 and a ROCAUC of 0.8283. Both models were evaluated using individual test datasets.

For comparison, the EPIC-TRACE model achieved an ROC AUC of 0.906 ± 0.000 for the TPP2 (unseen TCR, seen epitope) task and an AP of 0.691 ± 0.001 for the same task [4]. For the TPP3 task EPIC-TRACE achieved an AP of 0.294 ± 0.007 and an ROC AUC of 0.693 ± 0.008.

It is necessary to make a remark regarding these metrics, as the research conducted by Korpela et al. had a ratio of 1:5 for the true positive samples and the generated true negative samples [4]. This has a direct impact on the values of these two metrics, as the distribution of true positive and true negative samples, and vice versa, changes in relation to the dataset used in this thesis, which has nearly an exact distribution of 1:1 between positive and negative data points [64].

For a more accurate assessment, a comparative study using test datasets from other papers is essential. However, the results of the most successful model in the TPP3 task, tested on the data developed from this thesis, suggest that developing a moderately performing generalized model is feasible.

6.4 The Impact of Utilizing CDR3 Sequences From Only β-Chain Versus Both β- and α-Chains

As indicated in prior research, incorporating CDR3 sequences from both chains enhances the model's performance [19], [26]. The experiments conducted in this thesis yielded the same result. As illustrated in Table 13, the Paired models consistently demonstrate superior performance across nearly all averaged metrics and task combinations. In the TPP3 task the Beta Models exhibit superior performance. Care should be taken when interpreting the results obtained from this thesis,

as the test datasets for the different model variations differ, as noted earlier in 5.1.1 and explained in more detail in 6.1.

Nevertheless, the incorporation of the CDR3 α sequence provides the models with additional information, enabling them to reproduce the structural characteristics of the TCR more accurately. This results in enhanced performance. The superior performance of the Beta Vanilla model in the TPP3 evaluation with the AP metric can be attributed to the beta chain only datasets containing nearly four times more data. In addition the distribution of the CDR3 sequences are more balanced. A 2023 study demonstrated that the inclusion of additional data enhances generalization capabilities [65]. This highlights a disadvantage of using paired chain information, as datasets containing both chains are limited. The scarcity of such data, particularly for the dataset presented in this thesis, is detailed in Section 4.2.3.2. This limitation in available paired data is a common issue observed in all available datasets, as identified in previous work [3].

6.5 Interpretation of Allele and Gene Precision in V and J Regions and MHC

The work by Meysman et al. proposed the inclusion of the CDR1 and CDR2 regions in addition to the CDR3 region of the TCR [26]. This is because, although the CDR1 and CDR2 regions are determined by V gene usage, they contribute additional variability to the chains and play a crucial role in facilitating contacts between the TCR and the epitope-MHC complex[26]. Subsequent research done in previous work indicated that the inclusion of CDR1 and CDR2 regions can be indirectly achieved by incorporating the V/J-Regions of the corresponding TCR [20]. In this thesis, the CDR1 and CDR2 regions were included indirectly by adding information about the V/J-Regions. Experiments were conducted to assess the prediction performance of the models utilizing the V/J-Regions, as well as the MHC, with either gene or allele precision. The experiments demonstrated that the utilization of Allele precision is, on average, more effective than Gene precision in all task and averaged metric combinations, apart from the TPP1 AP evaluation. In this case, the Gene precision models exhibited a slight advantage, with an improved AP of 1.23%. It is important to note that different test datasets were utilized in this comparison, and this should be considered when interpreting the results.

The superior performance of the models using Allele precision relative to those using Gene precision may be attributed to the model's ability to differentiate between a greater number of groups within these regions and MHC. Further abbreviations studies must be conducted to provide a more precise justification for this assertion.

6.6 Explanation of Including Physicochemical Properties as Additional Descriptors for Protein Complexes

The rationale behind the incorporation of a vector representing the physicochemical properties of the protein complex was to provide additional information relevant to the binding prediction problem. Nevertheless, the experiments conducted for both the Beta and the Paired models

demonstrated that the inclusion of these vectors had a detrimental impact on performance. In all models and tasks, the performance declined when the physicochemical properties were included, except for the TPP1, AP evaluation with Gene precision.

Initially, the underperformance of these models led to the hypothesis that the scaling of the feature vector was being conducted in an unfavorable manner. To test this hypothesis, two variations were considered: one with standardization and one with the min-max normalization method, which transforms the information into an interval of $[-1, 1] = \{x \in R \mid -1 \leq x \leq 1\}$. The results indicated that the performance was negatively affected by the addition of these features, regardless of the normalization method employed.

Moreover, the He Initialization was employed for the linear layers of the Transformer blocks and the Classifier, as well as for the Embedding layer [66]. However, this did not improve the model's performance. A slight variation in the architectural approach was attempted, yet this did not result in enhanced performance.

As a third point of interest, experiments were conducted in which the patience of the early stopping callback was increased to 15 epochs, a change from the previous setting of 5 epochs. These modifications did not result in any large changes in the observed behavior.

Given that the inclusion of the physicochemical properties negatively affects the performance of both variations, further comparisons between the gap of the Paired Physicochemical and Beta Physicochemical models are conducted.

It can be observed that the Beta variation performs significantly better when using physicochemical properties compared to the Paired model. The mean performance of the Beta Physicochemical models, considering only the TPPs and excluding the global value, is 0.8183 ± 0.0533 for the AP metric and 0.7757 ± 0.0699 for the ROC AUC. The paired physicochemical models yielded a mean AP of 0.6423 ± 0.0428 and a ROC AUC value of 0.6579 ± 0.0335. Consequently, the Vanilla Physicochemical models exhibited a mean AP metric of 19.64% inferiority compared to the Beta Physiochemical model. Evaluated with the ROC AUC metric a 15.19% lower performance is observed. This discrepancy is of considerable magnitude and thus requires an explanation for its occurrence in future work.

This comparison is more reliable because each mean value is calculated as an average across all possible combinations of the test datasets. By including every test dataset variation, a thorough and fair comparison is achieved.

6.6.1 Improving Physicochemical Results Traceability via Recorded Metrics

Note that the following illustrations focus on the ROC AUC and the AP performances only of the models with the best global metrics evaluations.

The metrics with the prefix "*_Val*" are evaluated with the validation dataset during training, while the other metrics, except for "*train_loss*", are evaluated using the test dataset and illustrated as dots in the figures because they are computed once after the ending epoch.

To gain further insight into the distinction between the moderate-performing Beta Physicochemical models and the poorly performing Paired Physicochemical models, this section presents a visual analysis of the metrics.

For better readability, not all best Physicochemical models are depicted, only the ones with the best global metrics. The names of these models can be seen in the legends of the plots.

An overview of the best individual performances in each task is provided in Appendix C: Run Name to Performance Mapping Table and illustrated in Appendix E: Illustration of All Physicochemical Properties and Their Metrics Evaluations.

Figure 35, Figure 36 and Figure 37 illustrate that for the Beta Physicochemical models, the loss is decreasing while the evaluation metrics are increasing. On the left-hand side, the train loss functions of the best performing Paired Physicochemical models are plotted. They are very short and horizontal, outlined with a blue rectangle and annotated with "*Paired Physicochemical Metrics*". For a more detailed illustration, individual figures are provided below (Figure 38 and Figure 39).

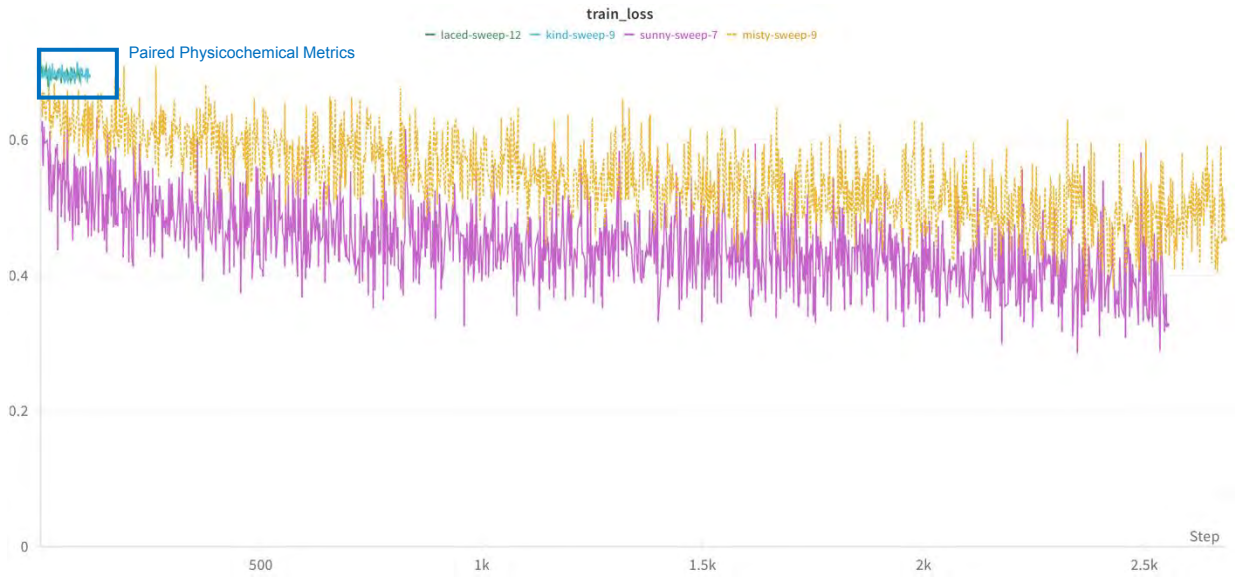


Figure 35: Comparison of training loss of different Physicochemical Models over training steps

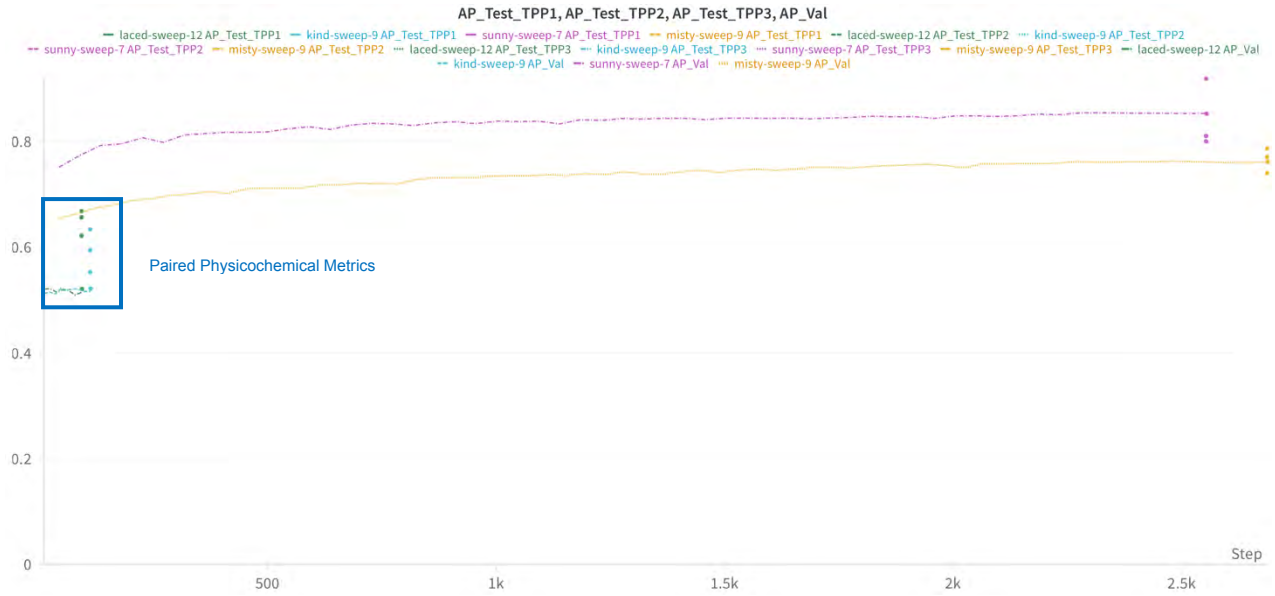


Figure 36: Comparison of AP_Val of different physicochemical models over training steps, illustrated with APs of different tasks evaluated on corresponding test datasets.

One can gain insights into the correlation between the loss and the evaluation metrics. For the Beta Physicochemical model, the loss is on a decreasing trajectory, whereas the evaluation metrics are exhibiting an upward trend. In contrast, for the Paired Physicochemical models, the loss remains constant, indicating that the AP and ROC AUC values remain unchanged. This is consistent with the observations presented in Figure 36 and Figure 37.

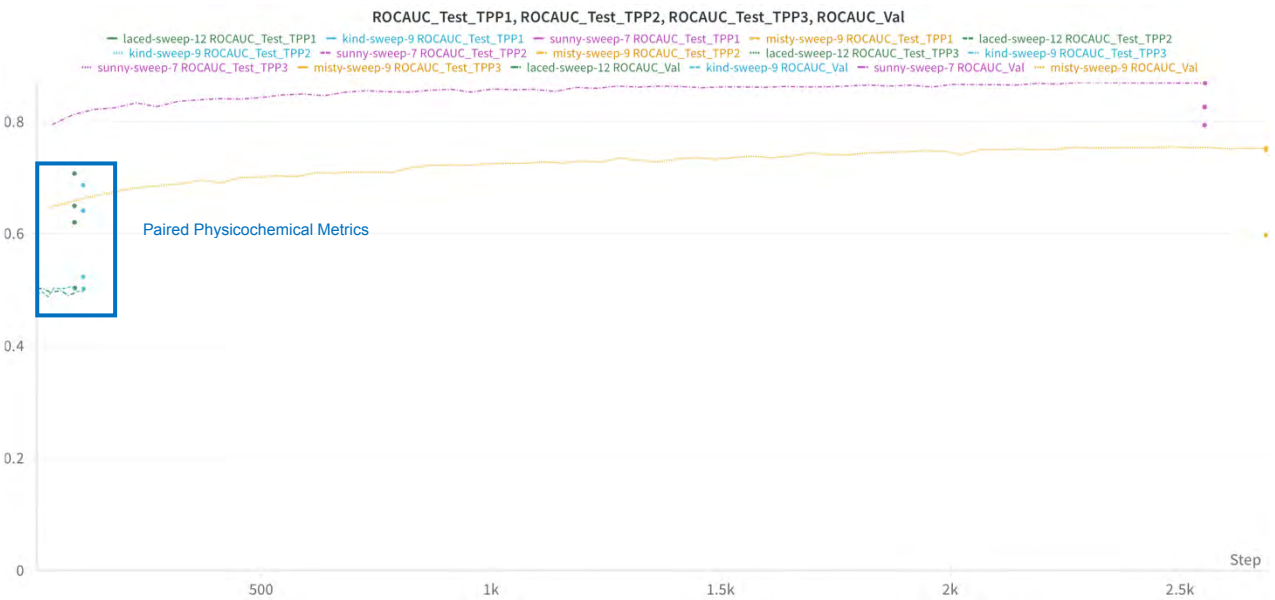


Figure 37: Comparison of ROCAUC of different physicochemical models over training steps, illustrated with ROCAUCs of different tasks evaluated on corresponding test datasets.

The following figures, Figure 38 and Figure 39, illustrate the training of Paired Physiochemical models in comparison to a single Beta Physiochemical model evaluation. It is evident that the model is not learning.

Dots represent the results of the evaluations conducted with the test datasets, while those evaluated with the validation dataset are graphs. Is because the test evaluation is done once after the very last epoch whether the validation of the models is done after every epoch.

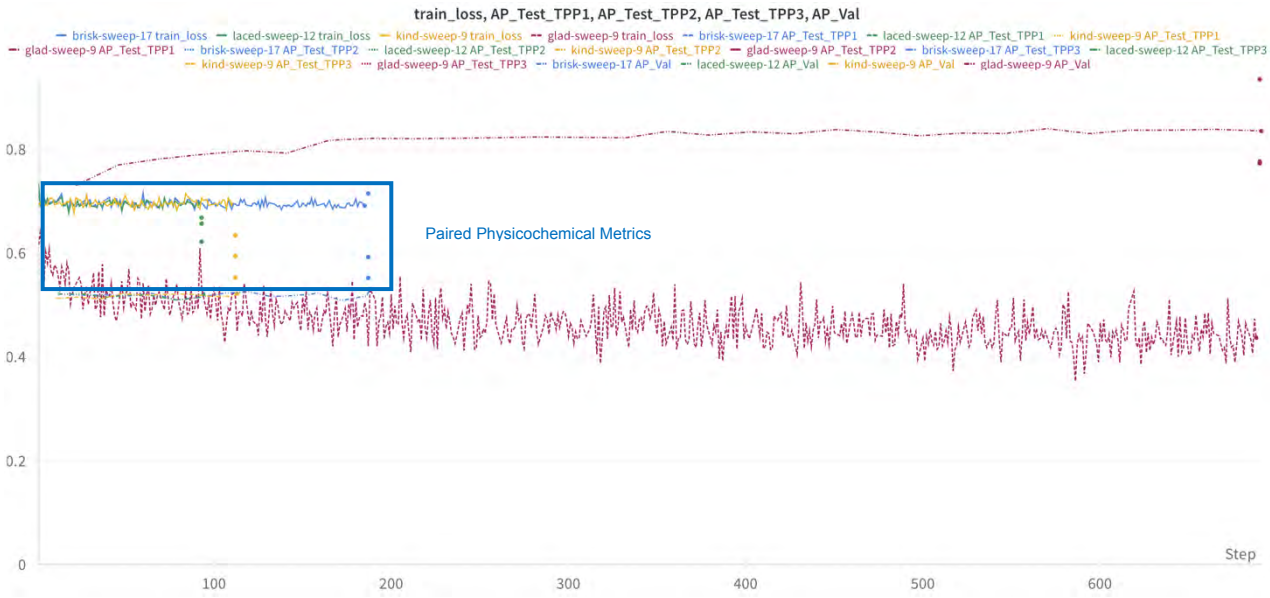


Figure 38: Comparison of training loss and AP metrics between underperforming Paired Physicochemical models and a single Beta Physicochemical model. Each line represents a different model variant, illustrating trends in train_loss, AP_Test_TPP1, AP_Test_TPP2, AP_Test_TPP3 and A_Val over the training steps.

A noteworthy observation is that the metrics evaluated on the test dataset for the Paired Physicochemical models exhibit a notable lower value than those evaluated with the validation dataset. This is not the case for the Beta Physicochemical model. There is a significant discrepancy between the evaluations conducted on the test dataset and those conducted with the validation dataset. This is despite the application of regularization techniques described in section 4.5, including SWA, early stopping, dropout layers, and weight decay [67].

A similar behavior is observed for ROC AUC metrics, as depicted in Figure 39.

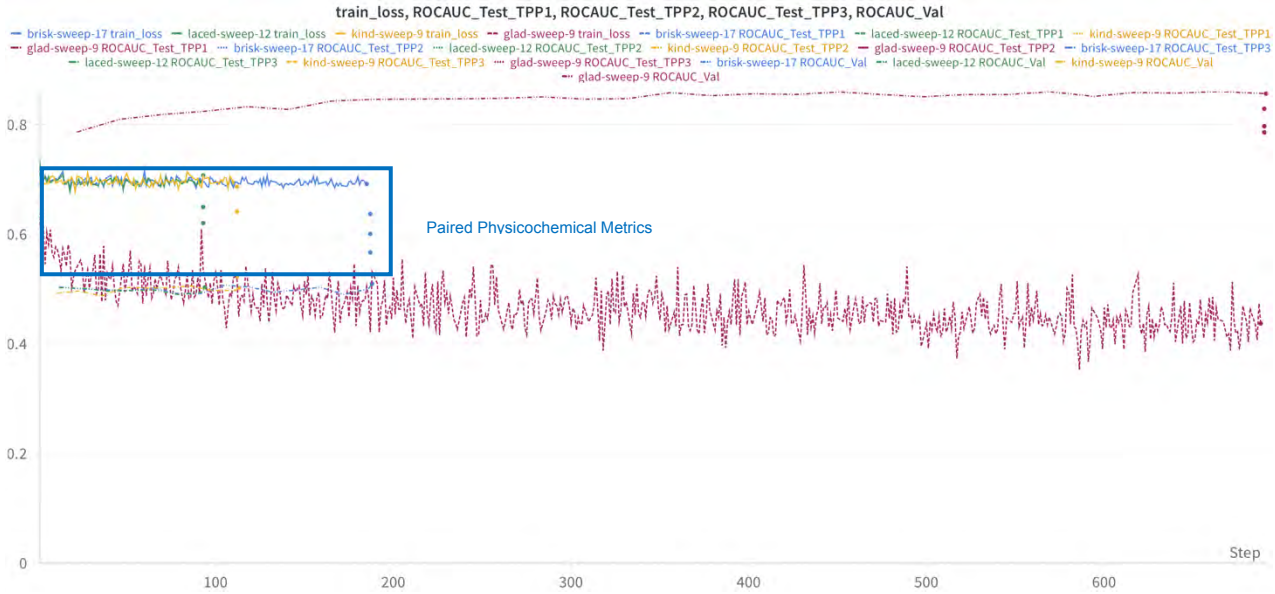


Figure 39: Comparative performance analysis of underperforming Paired Physicochemical models against a single Beta Physicochemical model, focusing on training loss and ROC AUC metrics. Each line represents a distinct model variant, depicting the evolution of train_loss, ROCAUC_Test_TPP1, ROCAUC_Test_TPP2, ROCAUC_Test_TPP3, and ROCAUC_Val over the training steps.

These interpretations suggest that, given the currently available paired data, incorporating physicochemical properties is not feasible. However, for the beta-only dataset, the impact is less pronounced. Although the Beta Vanilla models generally outperform the Beta Physicochemical models, the Gene variations difference is insignificant.

For the following analysis, the global AP and global ROC AUC are included in the mean calculations. For instance, the mean AP of the Beta Vanilla Gene models is 0.7851, with a ROC AUC of 0.7482. Similarly, the Beta Physicochemical Gene models have a mean AP of 0.7832 and a mean ROC AUC of 0.7398.

The difference is more noticeable in models using Allele precision. The Beta Vanilla model shows a mean AP of 0.8423 and a mean ROC AUC of 0.8360, whereas the Physicochemical variant has a mean AP of 0.8342 and a ROC AUC of 0.8174.

Comparisons within a specific model variation, such as Beta Vanilla Gene versus Beta Physicochemical Gene, can be made directly since they are evaluated using the same test dataset.

The motivation for incorporating these physicochemical properties originated from a published study that exclusively utilized TCR CDR3 β sequences in its analysis [17]. The addition of the physicochemical properties showed that paper a moderate positive effect on the performance [17].

6.7 Comparative Analysis of Evaluation Metrics Between All Model Variations

Figure 40 clearly demonstrates the differences between the variations with respect to the training loss. The Paired Physicochemical models, which have been annotated as spring-sweep-10, bright-

sweep-9, and resilient-sweep-8, are not learning at all, seen in the top left corner. In contrast, the three Beta Physicochemical models, designated as sunny-sweep-7, swept-sweep-6, and misty-sweep-5, exhibited significantly enhanced learning. The training time of the Paired Models was notably shorter, a consequence of the early stopping strategy employed during training, as discussed in section 4.4. Consequently, if the Paired Physicochemical models are not learning, the average precision of the model evaluated with the validation dataset is also not increasing, and thus the training is terminated.

A comparable pattern emerges when comparing the Paired Vanilla with Allele Precision, namely resilient-sweep-17, icy-sweep-16, and fluent-sweep-14, with the Beta Vanilla runs, which also employ Allele Precision and are labeled as colorful-sweep-12 and toasty-sweep-11.

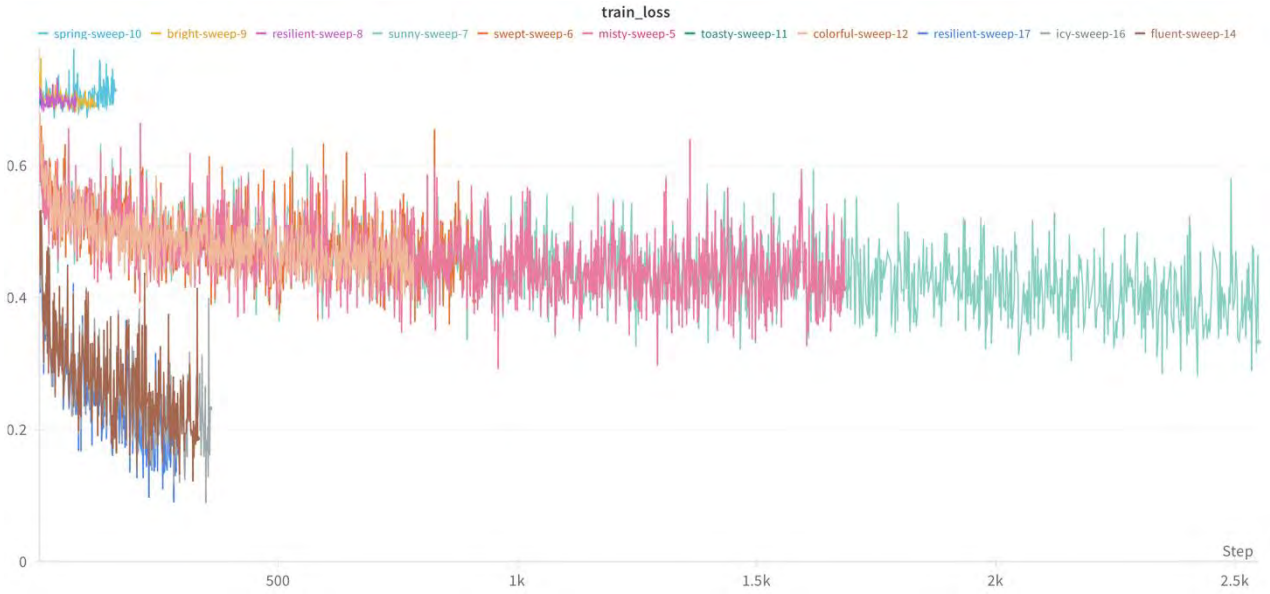


Figure 40: Training Loss Over Time for Different Model Variations

This figure displays the training loss over time for various model variations, including Paired Physicochemical models (spring-sweep-10, bright-sweep-9, resilient-sweep-8), Beta Physicochemical models (sunny-sweep-7, swept-sweep-6, misty-sweep-5), and others. The Paired Physicochemical models exhibit the highest training loss, indicating poor learning performance, while the Beta Physicochemical models show significantly improved learning outcomes. The differences in training loss trajectories highlight the varying effectiveness of each model type throughout the training process.

One can also interpret the training loss over training steps as an indicator for the different performances. This is because the loss over time negatively correlates with the performance measured with the test dataset. The Paired Physicochemical models have the highest training loss, which corresponds to the worst performance. The Beta Physicochemical models have a similar performance as the Beta Vanilla models, as can be seen in Figure 40, which show a similar behavior of the training loss over training steps. The same correlation applies for the Paired Vanilla models. The lowest loss is observed in the Paired Physicochemical models, which also exhibit the highest performance in terms of AP and ROC AUC.

To correlate the loss functions with the APs and ROC AUCs of the models, Figure 41 and Figure 42 are included.

Figure 41: Average Precision (AP) Evaluation Over Training Steps

This figure illustrates the Average Precision (AP) evaluation over training steps for various model configurations. The Paired Physicochemical models (spring-sweep-10, bright-sweep-9, resilient-sweep-8) exhibit lower AP values, indicating poorer performance, while the Beta Physicochemical models (sunny-sweep-7, swept-sweep-6, misty-sweep-5) show improved AP values. Each line represents the AP progression during training, and the dots denote the final test AP values for each model. The graph highlights the performance differences between the Beta and Paired models, as well as the impact of the physicochemical enhancements.

These figures represent the test values for all tasks and models as colored dots, indicating that the evaluations were conducted only once after the final training epoch. The graphs are evaluated using the validation dataset after every epoch.

In general, the evaluation metrics for at least certain models increase over time, which indicates that the models are learning. The training loss over training steps is a direct indicator of a model's performance.

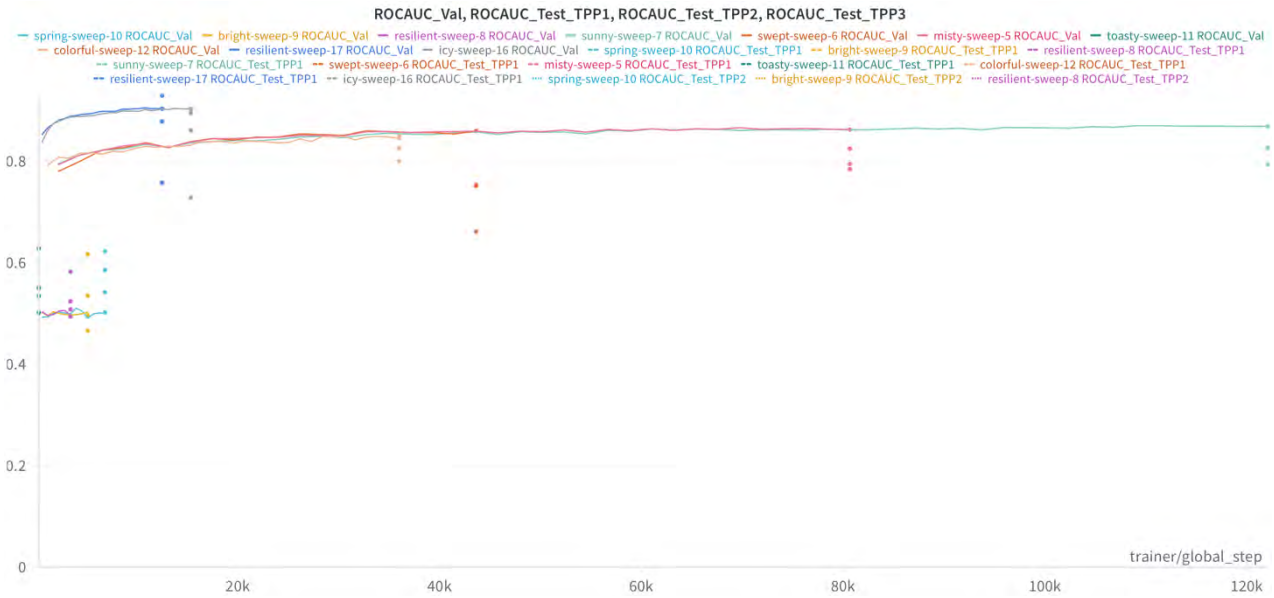


Figure 42: ROC AUC Evaluation Over Training Steps

This figure presents the ROC AUC (Area Under the Receiver Operating Characteristic curve) evaluation over training steps for different model configurations. The Paired Physicochemical models (spring-sweep-10, bright-sweep-9, resilient-sweep-8) show lower ROC AUC values, indicating subpar performance, whereas the Beta Physicochemical models (sunny-sweep-7, swept-sweep-6, misty-sweep-5) demonstrate higher ROC AUC values, reflecting better performance. Each line indicates the ROC AUC progression during training, and the dots represent the final test ROC AUC values for each model. The graph underscores the performance discrepancies between Beta and Paired models and the influence of physicochemical enhancements on model efficacy.

6.8 Limitation of This Thesis

One limitation of this thesis is the restriction of the comparisons that can be made between certain variations of models in this work and, above all, with models from other works as described in detail in section 6.1.

The publicly accessible data imposes a further limitation, which is accompanied by associated limitations regarding true negative samples.

Additionally, due to time constraints, this thesis prioritized addressing the research questions and objectives over exploring the relative importance of the models' input features. Computational limitations were also considered. For instance, the hyperparameter tuning process focused on the global AP metric rather than on a specific task, as explained in section 4.6.

7 Outlook

This section lists insights that were not addressed in this thesis and provides an overview of potential future work.

7.1 Overall Test Dataset

As noted in the comparisons between the models developed within this thesis and the SOTA models, different test datasets were used for different model variations. To ensure consistent and fair comparisons, the same test dataset needs to be used. This approach guarantees that the performance metrics are directly comparable, and the evaluations are reliable. In further work one can create such a comprehensive test dataset and evaluate all the models with it.

7.2 Improvement of the Data Split in Order to Generate More TPP3 Data Samples

Figure 4 illustrates that upon creating the data split, all unique TCRs are initially filtered out to facilitate the generation of the test set and the required number of samples for the TPP2 and TPP3 tasks. This approach could be improved by also paying attention to unique or rarely occurring epitopes, which would most likely result in more TPP3 samples in the test set.

7.3 Improved Hyperparameter Tuning for Task-Specific Optimization

The models resulting from the hyperparameter tuning in this thesis are based on global metrics accumulated from all three tasks: TPP1, TPP2, and TPP3. The result is that neither of the models is perfectly tuned towards a specific task. A future work based on this thesis could be improved by adjusting the dataset split so that it is done only for one task. In this case, the metric to maximize during the hyperparameter tuning process would be the one specifically for that task, for example, AP TPP3.

7.4 Enhanced Embedding Development through Full-Length TCR Sequences

One additional experiment is to assess the efficacy of creating embeddings for distinct portions of the TCR for the full chain sequence. As an alternative to the approach taken in this thesis, one could create embeddings for the full-length TCR, consisting of the TCR CDR3 α, TCR CDR3 β, and epitope sequences. One potential solution is to use a library like stitchr (https://jamieheather.github.io/stitchr/), which generates the complete TCR sequence based on the given features of the chain. In a similar vein, Korpela et al. [4] employed this methodology for the creation of embeddings when the necessary information to obtain the complete TCR sequence was available. The data containing the full-length TCRs was prepared for analysis, but due to time and computational constraints, the corresponding experiments could not be conducted.

7.5 Analyzing the Underperformance of Physicochemical Models

The hypothesis for adding the physicochemical property vector to the protein complexes was that the model would learn to incorporate this higher-level information. However, as observed in this

thesis, this was not the case, and especially the performances of the Paired Physicochemical models are poor. Future work should investigate the causes of this behavior.

7.6 Enhancing TCR-pMHC Prediction through Transfer Learning on General Protein-Protein Binding Data

An additional experiment involves pre-training the existing models with general protein-protein binding data. Subsequently, the models are fine-tuned using the TCR-pMHC-specific binding affinity data. The specific datasets are the ones utilized in the experiments.

The protein symbols of the general-purpose protein-protein binding data were obtained from BioGid. Subsequently, the amino acid sequences were extracted from the UniProt database via its Application Programming Interface (API) using the protein symbols. Consequently, a substantial dataset comprising approximately 1.2 million true positive data samples can be generated, encompassing unspecific protein-protein-binding data. In addition to this, the artificially generated examples of negative data would also be incorporated into the set.

A dataset of general-purpose protein-protein binding data is available for analysis, although the execution of the requisite experiments was prevented due to resource limitations.

Further considerations were undertaken to determine the precise methodology that could have been employed to conduct such an experiment. As the dataset includes two amino acid sequences for the Beta Models, it is evident how one would pretrain them, given that these models have two input heads (see Figure 31 and Figure 33). This is not the case for the Paired models, as they have three input heads, as illustrated in Figure 30 and Figure 32. However, for the paired models, the available data is significantly less, making this approach even more crucial. One potential approach to pretraining the model is to consistently pass one sequence to the epitope input head, and then to equally alternate the input between the TCR CDR3 α and TCR CDR3 β input heads.

This ensures that the model architecture remains unaltered and that both variations can be trained with the same dataset. Similarly, incorporating physicochemical properties is feasible for both Beta and Paired models.

8 Conclusion

This thesis offers several significant contributions to the field of bioinformatics:

1. **Application Code for Generating Datasets including Negative Samples**
    - An open-source project was developed to generate datasets from three publicly available data sources. The application code addresses a common issue found in previous work, where data is often dispersed across multiple files and difficult to understand. The resulting dataset is comprehensive and organized in a single tabular format (.tsv file), with optional additional files, such as embeddings or physicochemical properties, included during training. Modularized code was created to ensure an understandable implementation of the models and training processes. This improves clarity and organization, aiding researchers in building upon existing work more efficiently.
    - The project introduces a novel method for generating synthetic non-binding data samples through informed shuffling.

It is important to note that different test datasets were used for the various model variations, so caution is required when interpreting these results among different models.

2. **Analysis of Paired vs. Beta-Only Datasets**
    - The thesis analyzed the impacts of using paired datasets compared to TCR CDR3 beta-only datasets. It is important to note that the results need to be relativized due to the different model variations having different test datasets. Nevertheless, the findings indicate that, overall, the Paired models exhibit higher performance. However, in the TPP3 task, where both the T cell and the epitope are unseen, the Beta model outperformed the Paired model, achieving the highest AP in this particularly challenging scenario.
3. **Impact of Precision Levels**
    - The study investigated the impact of two different precision levels for the V/J-Regions, as well as the MHC. Subject to the different test datasets, the performance of the Allele precision models indicates a higher performance. On average, the Allele models demonstrated a 12.54% increase in average precision (AP) in the most complex task, where all sequences were previously unseen (TPP3).
4. **Effect of Physicochemical properties**
    - Experiments were conducted with an additional vector describing the physicochemical properties of protein complexes. It was observed that the performance of Beta models either remains the same or slightly decreases, while the performance of Paired models collapses to the extent that they fail to learn. The insights gained indicate that, given the currently available paired data, incorporating physicochemical properties is not viable. Nevertheless, for the beta-only dataset, the impact is less severe. Although the Beta Vanilla models

generally outperform the Beta Physicochemical models, the difference in the Gene variations is not significant.

The comparison between the Physicochemical models and the Vanilla models is fair because it incorporates the evaluation of the average metric, ensuring a comprehensive and balanced evaluation.

Future Work and Conclusion

For future work, it would be beneficial to investigate the causes of the poor performance of Paired models with the added physicochemical properties. Furthermore, including a dataset from an independent source as a test set provides a more robust evaluation of the models after hyperparameter tuning. Additional proposals are described in detail in section 7.

In conclusion, the currently public available data is insufficient to fully describe the complexity of the immune response. However, if the data situation improves, it is likely that a robust model could be developed, offering significant added value in the health sector.

9 References

9.1 Bibliography

[1] J. Zhang, W. Ma, and H. Yao, “Accurate TCR-pMHC interaction prediction using a BERT-based transfer learning method,” *Brief. Bioinform.*, vol. 25, no. 1, p. bbad436, Nov. 2023, doi: 10.1093/bib/bbad436.

[2] C. Graham, R. Hewitson, A. Pagliuca, and R. Benjamin, “Cancer immunotherapy with CAR-T cells – behold the future,” *Clin. Med.*, vol. 18, no. 4, pp. 324–328, Aug. 2018, doi: 10.7861/clinmedicine.18-4-324.

[3] A. Weber, J. Born, and M. Rodriguez Martínez, “TITAN: T-cell receptor specificity prediction with bimodal attention networks,” *Bioinformatics*, vol. 37, no. Supplement_1, pp. i237–i244, Aug. 2021, doi: 10.1093/bioinformatics/btab294.

[4] D. Korpela, E. Jokinen, A. Dumitrescu, J. Huuhtanen, S. Mustjoki, and H. Lähdesmäki, “EPIC-TRACE: predicting TCR binding to unseen epitopes using attention and contextualized embeddings,” *Bioinformatics*, vol. 39, no. 12, p. btad743, Dec. 2023, doi: 10.1093/bioinformatics/btad743.

[5] G. Li *et al.*, “T cell antigen discovery via trogocytosis,” *Nat. Methods*, vol. 16, no. 2, pp. 183–190, Feb. 2019, doi: 10.1038/s41592-018-0305-7.

[6] P. Moris *et al.*, “Current challenges for epitope-agnostic TCR interaction prediction and a new perspective derived from image classification.” Dec. 18, 2019. doi: 10.1101/2019.12.18.880146.

[7] A. Weber, A. Pélissier, and M. R. Martínez, “T cell receptor binding prediction: A machine learning revolution.” arXiv, Dec. 27, 2023. Accessed: May 08, 2024. [Online]. Available: http://arxiv.org/abs/2312.16594

[8] D. J. Laydon, C. R. M. Bangham, and B. Asquith, “Estimating T-cell repertoire diversity: limitations of classical estimators and a new approach,” *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 370, no. 1675, p. 20140291, Aug. 2015, doi: 10.1098/rstb.2014.0291.

[9] D. Hudson, R. A. Fernandes, M. Basham, G. Ogg, and H. Koohy, “Can we predict T cell specificity with digital biology and machine learning?,” *Nat. Rev. Immunol.*, vol. 23, no. 8, pp. 511–521, Aug. 2023, doi: 10.1038/s41577-023-00835-3.

[10] A. Vaswani *et al.*, “Attention Is All You Need.” arXiv, Aug. 01, 2023. Accessed: May 15, 2024. [Online]. Available: http://arxiv.org/abs/1706.03762

[11] A. Myronov, G. Mazzocco, P. Król, and D. Plewczynski, “BERTrand—peptide:TCR binding prediction using Bidirectional Encoder Representations from Transformers augmented with random TCR pairing,” *Bioinformatics*, vol. 39, no. 8, p. btad468, Aug. 2023, doi: 10.1093/bioinformatics/btad468.

[12] S.-R. Woo, L. Corrales, and T. F. Gajewski, “Innate Immune Recognition of Cancer,” *Annu. Rev. Immunol.*, vol. 33, no. 1, pp. 445–474, Mar. 2015, doi: 10.1146/annurev-immunol-032414-112043.

[13] K. M. Murphy and C. Weaver, *Janeway’s immunobiology*, 9th edition. New York London: GS, Garland Science, Taylor & Francis Group, 2017.

[14] Y. Zhao, C. Niu, and J. Cui, “Gamma-delta (γδ) T cells: friend or foe in cancer development?,” *J. Transl. Med.*, vol. 16, no. 1, p. 3, Dec. 2018, doi: 10.1186/s12967-017-1378-2.

[15] P. Zhang, S. Bang, and H. Lee, “PiTE: TCR-epitope Binding Affinity Prediction Pipeline using Transformer-based Sequence Encoder,” in *Biocomputing 2023*, Kohala Coast, Hawaii, USA: WORLD SCIENTIFIC, Nov. 2022, pp. 347–358. doi: 10.1142/9789811270611_0032.

[16] G. Petrova, A. Ferrante, and J. Gorski, “Cross-Reactivity of T Cells and Its Role in the Immune System,” *Crit. Rev. Immunol.*, vol. 32, no. 4, pp. 349–372, 2012, doi: 10.1615/CritRevImmunol.v32.i4.50.

[17] E. Goffinet, R. Mall, A. Singh, R. Kaushik, and F. Castiglione, “MATE-Pred: Multimodal Attention-based TCR-Epitope interaction Predictor.” arXiv, Dec. 05, 2023. Accessed: May 08, 2024. [Online]. Available: http://arxiv.org/abs/2401.08619

[18] E. Jokinen *et al.*, “Predicting recognition between T cell receptors and epitopes using contextualized motifs.” May 24, 2022. doi: 10.1101/2022.05.23.493034.

[19] A. Montemurro *et al.*, “NetTCR-2.0 enables accurate prediction of TCR-peptide binding by using paired TCRα and β sequence data,” *Commun. Biol.*, vol. 4, no. 1, p. 1060, Sep. 2021, doi: 10.1038/s42003-021-02610-3.
[20] M. F. Jensen and M. Nielsen, “NetTCR 2.2 - Improved TCR specificity predictions by combining pan- and peptide-specific training strategies, loss-scaling and integration of sequence similarity.” Oct. 16, 2023. doi: 10.1101/2023.10.12.562001.
[21] M. Shugay *et al.*, “VDJdb: a curated database of T-cell receptor sequences with known antigen specificity,” *Nucleic Acids Res.*, vol. 46, no. D1, pp. D419–D427, Jan. 2018, doi: 10.1093/nar/gkx760.
[22] N. Tickotsky, T. Sagiv, J. Prilusky, E. Shifrut, and N. Friedman, “McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences,” *Bioinformatics*, vol. 33, no. 18, pp. 2924–2929, Sep. 2017, doi: 10.1093/bioinformatics/btx286.
[23] M. Klinger *et al.*, “Multiplex Identification of Antigen-Specific T Cell Receptors Using a Combination of Immune Assays and Immune Receptor Sequencing,” *PLOS ONE*, vol. 10, no. 10, p. e0141561, Oct. 2015, doi: 10.1371/journal.pone.0141561.
[24] N. V. Chawla, N. Japkowicz, and A. Kotcz, “Editorial: special issue on learning from imbalanced data sets,” *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 1–6, Jun. 2004, doi: 10.1145/1007730.1007733.
[25] S. Nolan *et al.*, “A large-scale database of T-cell receptor beta (TCRβ) sequences and binding associations from natural and synthetic exposure to SARS-CoV-2.” Aug. 04, 2020. doi: 10.21203/rs.3.rs-51964/v1.
[26] P. Meysman *et al.*, “Benchmarking solutions to the T-cell receptor epitope prediction problem: IMMREP22 workshop report,” *ImmunoInformatics*, vol. 9, p. 100024, Mar. 2023, doi: 10.1016/j.immuno.2023.100024.
[27] W. Zhang *et al.*, “A framework for highly multiplexed dextramer mapping and prediction of T cell receptor sequences to antigen specificity,” *Sci. Adv.*, vol. 7, no. 20, p. eabf5835, May 2021, doi: 10.1126/sciadv.abf5835.
[28] H. Zhang *et al.*, “Investigation of Antigen-Specific T-Cell Receptor Clusters in Human Cancers,” *Clin. Cancer Res.*, vol. 26, no. 6, pp. 1359–1371, Mar. 2020, doi: 10.1158/1078-0432.CCR-19-3249.
[29] P. Dash *et al.*, “Quantifiable predictive features define epitope-specific T cell receptor repertoires,” *Nature*, vol. 547, no. 7661, pp. 89–93, Jul. 2017, doi: 10.1038/nature22383.
[30] M. V. Pogorelyy *et al.*, “Detecting T cell receptors involved in immune responses from single repertoire snapshots,” *PLOS Biol.*, vol. 17, no. 6, p. e3000314, Jun. 2019, doi: 10.1371/journal.pbio.3000314.
[31] W. D. Chronister *et al.*, “TCRMatch: Predicting T-Cell Receptor Specificity Based on Sequence Similarity to Previously Characterized Receptors,” *Front. Immunol.*, vol. 12, p. 640725, Mar. 2021, doi: 10.3389/fimmu.2021.640725.
[32] M.-D. N. Pham *et al.*, “epiTCR: a highly sensitive predictor for TCR–peptide binding,” *Bioinformatics*, vol. 39, no. 5, p. btad284, May 2023, doi: 10.1093/bioinformatics/btad284.
[33] N. K. Chauhan and K. Singh, “A Review on Conventional Machine Learning vs Deep Learning,” in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, Greater Noida, Uttar Pradesh, India: IEEE, Sep. 2018, pp. 347–352. doi: 10.1109/GUCON.2018.8675097.
[34] W. Zhang *et al.*, “PIRD: Pan Immune Repertoire Database,” *Bioinformatics*, vol. 36, no. 3, pp. 897–903, Feb. 2020, doi: 10.1093/bioinformatics/btz614.
[35] 10x Genomics, “A new way of exploring immunity: linking highly multiplexed antigen recognition to immune repertoire and phenotype.” [Online]. Available: https://pages.10xgenomics.com/rs/446-PBO-704/images/10x_AN047_IP_A_New_Way_of_Exploring_Immunity_Digital.pdf
[36] J. Dean *et al.*, “Annotation of pseudogenic gene segments by massively parallel sequencing of rearranged lymphocyte receptor loci,” *Genome Med.*, vol. 7, no. 1, p. 123, Dec. 2015, doi: 10.1186/s13073-015-0238-z.
[37] R. Vita *et al.*, “The Immune Epitope Database (IEDB): 2018 update,” *Nucleic Acids Res.*, vol. 47, no. D1, pp. D339–D343, Jan. 2019, doi: 10.1093/nar/gky1006.

[38] D. Weininger, “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules,” *J. Chem. Inf. Comput. Sci.*, vol. 28, no. 1, pp. 31–36, Feb. 1988, doi: 10.1021/ci00057a005.
[39] S. Henikoff and J. G. Henikoff, “Amino acid substitution matrices from protein blocks.,” *Proc. Natl. Acad. Sci.*, vol. 89, no. 22, pp. 10915–10919, Nov. 1992, doi: 10.1073/pnas.89.22.10915.
[40] J. Born *et al.*, “Data-driven molecular design for discovery and synthesis of novel ligands: a case study on SARS-CoV-2,” *Mach. Learn. Sci. Technol.*, vol. 2, no. 2, p. 025024, Jun. 2021, doi: 10.1088/2632-2153/abe808.
[41] D. S. Fischer, Y. Wu, B. Schubert, and F. J. Theis, “Predicting antigen specificity of single T cells based on TCR CDR 3 regions,” *Mol. Syst. Biol.*, vol. 16, no. 8, p. e9416, Aug. 2020, doi: 10.15252/msb.20199416.
[42] A. Elnaggar *et al.*, “CodeTrans: Towards Cracking the Language of Silicon’s Code Through Self-Supervised Deep Learning and High Performance Computing,” 2021, doi: 10.48550/ARXIV.2104.02443.
[43] A. Elnaggar *et al.*, “ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 7112–7127, Oct. 2022, doi: 10.1109/TPAMI.2021.3095381.
[44] J. A. Carter *et al.*, “Single T Cell Sequencing Demonstrates the Functional Role of αβ TCR Pairing in Cell Lineage and Antigen Specificity,” *Front. Immunol.*, vol. 10, p. 1516, Jul. 2019, doi: 10.3389/fimmu.2019.01516.
[45] Y. Nagano and B. Chain, “tidytcells: standardizer for TR/MH nomenclature,” *Front. Immunol.*, vol. 14, p. 1276106, Oct. 2023, doi: 10.3389/fimmu.2023.1276106.
[46] A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, “Cosine similarity to determine similarity measure: Study case in online essay assessment,” in *2016 4th International Conference on Cyber and IT Service Management*, Bandung, Indonesia: IEEE, Apr. 2016, pp. 1–6. doi: 10.1109/CITSM.2016.7577578.
[47] K. Wu *et al.*, “TCR-BERT: learning the grammar of T-cell receptors for flexible antigen-xbinding analyses.” Nov. 20, 2021. doi: 10.1101/2021.11.18.469186.
[48] Z. Lin *et al.*, “Evolutionary-scale prediction of atomic-level protein structure with a language model,” *Science*, vol. 379, no. 6637, pp. 1123–1130, Mar. 2023, doi: 10.1126/science.ade2574.
[49] S. Sadeghian-Rizi, A. Sakhteman, and F. Hassanzadeh, “A quantitative structure-activity relationship (QSAR) study of some diaryl urea derivatives of B-RAF inhibitors,” *Res. Pharm. Sci.*, vol. 11, no. 6, p. 445, 2016, doi: 10.4103/1735-5362.194869.
[50] P. J. Van Den Elsen, “Expression Regulation of Major Histocompatibility Complex Class I and Class II Encoding Genes,” *Front. Immunol.*, vol. 2, 2011, doi: 10.3389/fimmu.2011.00048.
[51] S. I. Van Kasteren, H. Overkleeft, H. Ovaa, and J. Neefjes, “Chemical biology of antigen presentation by MHC molecules,” *Curr. Opin. Immunol.*, vol. 26, pp. 21–31, Feb. 2014, doi: 10.1016/j.coi.2013.10.005.
[52] P. Wang, J. Sidney, C. Dow, B. Mothé, A. Sette, and B. Peters, “A Systematic Assessment of MHC Class II Peptide Binding Predictions and Evaluation of a Consensus Approach,” *PLoS Comput. Biol.*, vol. 4, no. 4, p. e1000048, Apr. 2008, doi: 10.1371/journal.pcbi.1000048.
[53] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.
[54] A. Krogh and J. Hertz, “A Simple Weight Decay Can Improve Generalization,” in *Advances in Neural Information Processing Systems*, J. Moody, S. Hanson, and R. P. Lippmann, Eds., Morgan-Kaufmann, 1991. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf
[55] A. F. Agarap, “Deep Learning using Rectified Linear Units (ReLU).” arXiv, 2018. doi: 10.48550/ARXIV.1803.08375.
[56] B. J. Kim, H. Choi, H. Jang, D. Lee, and S. W. Kim, “How to Use Dropout Correctly on Residual Networks with Batch Normalization.” arXiv, Feb. 13, 2023. Accessed: May 15, 2024. [Online]. Available: http://arxiv.org/abs/2302.06112

[57] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer Normalization,” 2016, doi: 10.48550/ARXIV.1607.06450.
[58] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” 2015, doi: 10.48550/ARXIV.1502.03167.
[59] A. P. Bradley, “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997, doi: 10.1016/S0031-3203(96)00142-2.
[60] S. J. Reddi, S. Kale, and S. Kumar, “On the Convergence of Adam and Beyond.” arXiv, Apr. 19, 2019. Accessed: Jun. 01, 2024. [Online]. Available: http://arxiv.org/abs/1904.09237
[61] S. Ruder, “An overview of gradient descent optimization algorithms.” arXiv, Jun. 15, 2017. Accessed: Jun. 01, 2024. [Online]. Available: http://arxiv.org/abs/1609.04747
[62] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, “Averaging Weights Leads to Wider Optima and Better Generalization.” arXiv, 2018. doi: 10.48550/ARXIV.1803.05407.
[63] Aman Gupta, Rohan Ramanath, Jun Shi, and S. Sathiya Keerthi, “Adam vs. SGD: Closing the generalization gap on image classification.” [Online]. Available: https://www.opt-ml.org/papers/2021/paper53.pdf
[64] F. S. Nahm, “Receiver operating characteristic curve: overview and practical use for clinicians,” *Korean J. Anesthesiol.*, vol. 75, no. 1, pp. 25–36, Feb. 2022, doi: 10.4097/kja.21209.
[65] X. Zhou, Y. Jiang, and M. Bansal, “Data Factors for Better Compositional Generalization.” arXiv, Nov. 07, 2023. Accessed: Jun. 01, 2024. [Online]. Available: http://arxiv.org/abs/2311.04420
[66] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.” arXiv, 2015. doi: 10.48550/ARXIV.1502.01852.
[67] X. Ying, “An Overview of Overfitting and its Solutions,” *J. Phys. Conf. Ser.*, vol. 1168, p. 022022, Feb. 2019, doi: 10.1088/1742-6596/1168/2/022022.

9.2 List of Figures

9.3 List of Tables

9.4 List of Equations

9.5 List of Abbreviations

Abbreviation	Term
AP	Average Precision
AP_Test_TPP1	Average Precision on Test TPP1
AP_Test_TPP2	Average Precision on Test TPP2
AP_Test_TPP3	Average Precision on Test TPP3
AP_Val	Average Precision on Validation Set
API	Application Programming Interface
C-Map	Contact Map
CDR3	Complementarity Determining Region 3
CNN	Convolutional Neural Network
DNA	Deoxyribonucleic Acid
FPR	False Positive Rate
GRU	Gated Recurrent Unit
HLA	Human Leukocyte Antigen
LSTM	Long Short-Term Memory
MHC	Major Histocompatibility Complex
MLP	Connected Multilayer Perceptrons
PAMP	Pathogen-Associated Molecular Pattern
PLM	Protein Language Model
pMHC	Peptide-Major Histocompatibility Complex
PRR	Pattern Recognition Receptor
QSAR	Quantitative Structure-Activity Relationship
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
ROC AUC	Receiver Operating Characteristic Area Under the Curve
SGD	Stochastic Gradient Descent
SOTA	State-Of-The-Art
SWA	Stochastic Weight Averaging
TCR	T Cell Receptor
TCRα	T Cell Receptor alpha
TCRβ	T Cell Receptor beta
TdT	Terminal Deoxynucleotidyl Transferase
TPP	TCR-Peptide Pairing
TPR	True Positive Rate
TRA_CDR3	TCR CDR3 sequence of α-chain
TRAJ	TCR J-Region of the α-chain
TRAV	TCR V-Region of the α-chain
TRB_CDR3	TCR CDR3 sequence of β-chain
TRBJ	TCR J-Region of the β-chain
TRBV	TCR V-Region of the β-chain

10 Appendix

10.1 Appendix A: Project Description

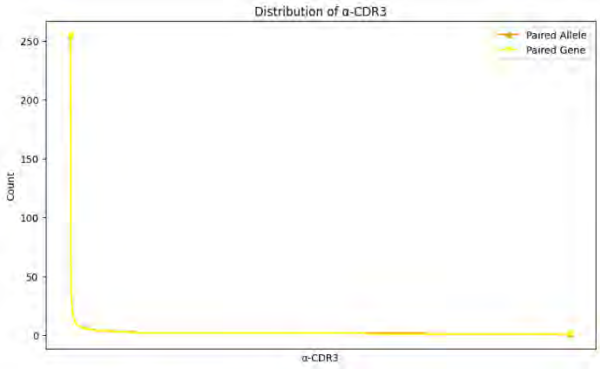
T-cells are a type of white blood cell that play a crucial role in the adaptive immune response. The ability of T-cells to recognize and bind to specific epitopes is essential for the immune system's ability to mount an appropriate immune response. Understanding the binding interactions between T-cell epitopes and TCRs is crucial for studying immune responses, designing vaccines, and developing immunotherapies for various diseases, including infections, autoimmune disorders, and cancer. Therefore, the goal in this work is to explore the ability of utilizing molecule representations based on large protein language models combined with state-of-the-art deep learning approaches to predict the T-cell epitope binding affinity and this way make one step towards personalized immunotherapy.

10.2 Appendix B: EDA

10.2.1 CDR3 Sequences

10.2.1.1 α-chain

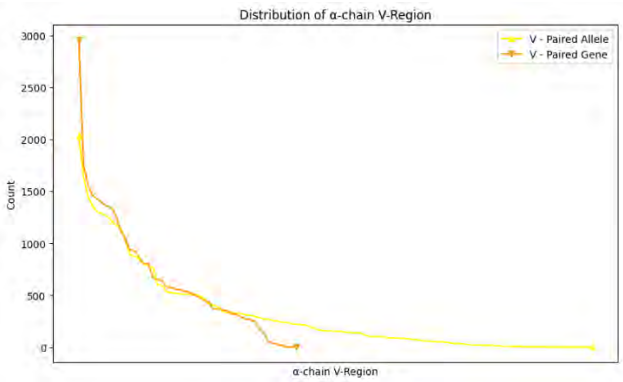
The plot shows the distribution of the α-CDR3 sequence.

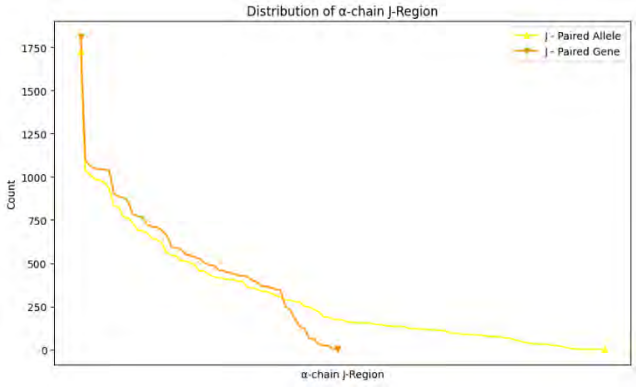


10.2.2 V/J Region

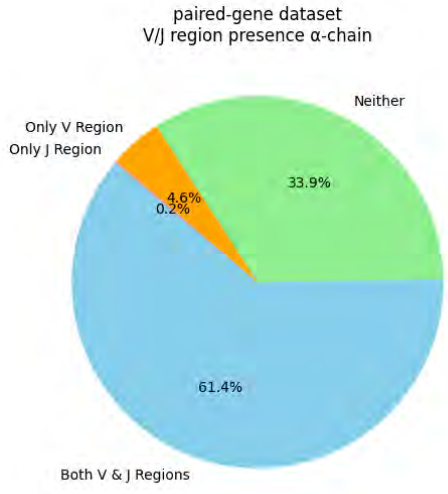
10.2.2.1 α-chain

The plots show the distribution of the V/J-Region in α-chain. This is only for the paired dataset.



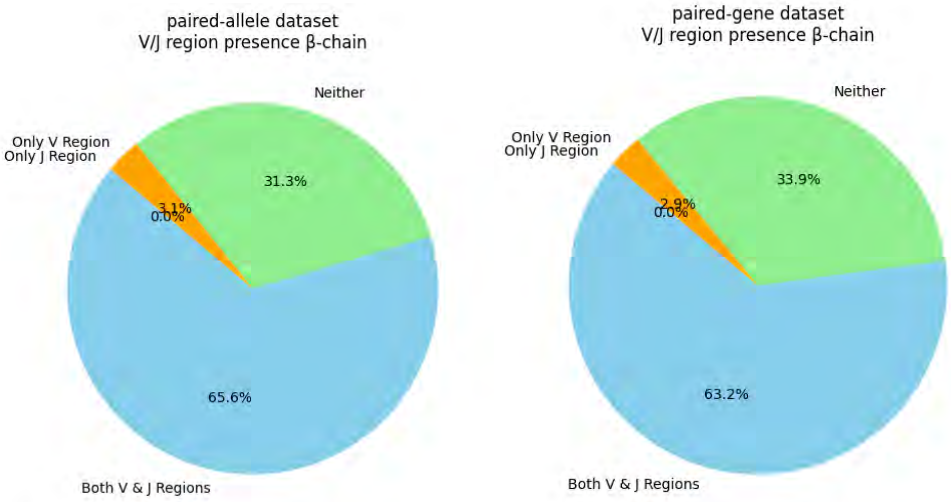


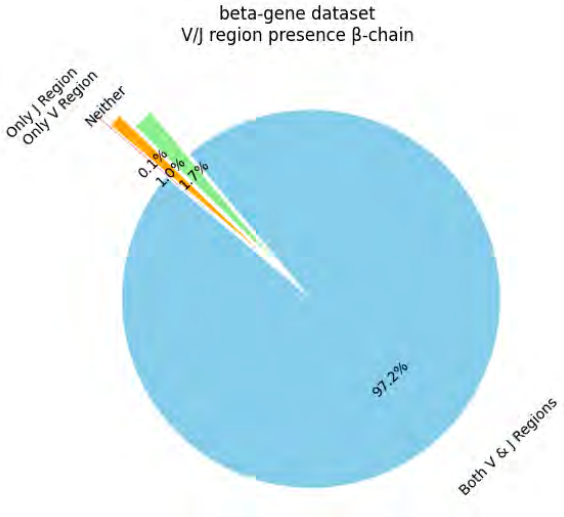
The pie plots show detailed information about the V/J-Region presence in α-chain. This is only for the paired dataset.



10.2.2.2 β-chain

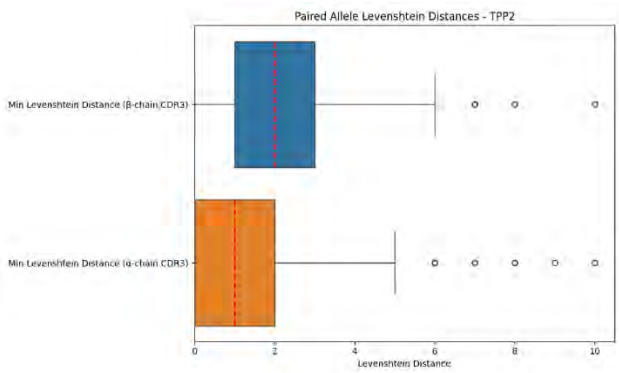
The pie plots show detailed information about the V/J-Region presence in β-chain, for paired and beta datsets.

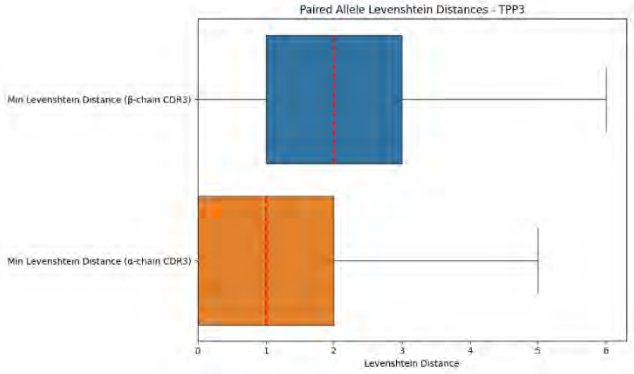


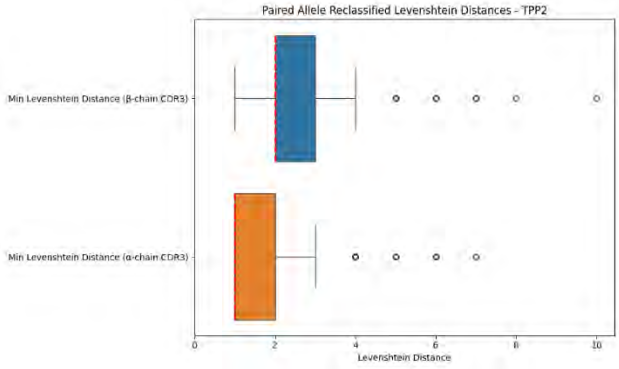


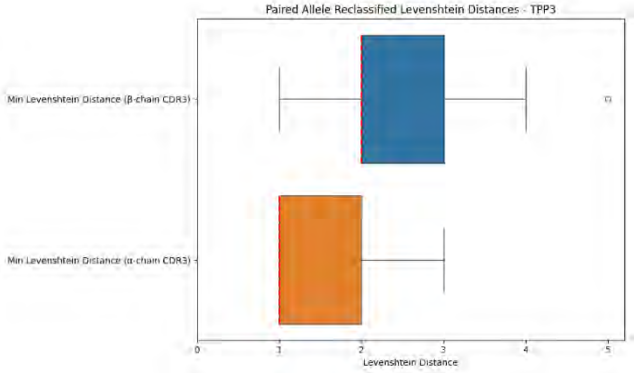
10.2.3 Tasks (Levenshtein Boxplots)

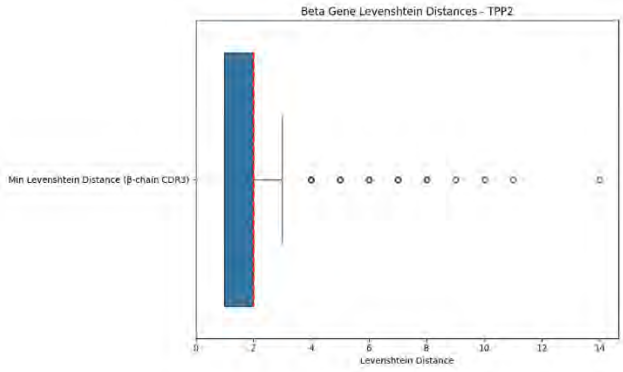
The boxplots show the distribution of the Levenshtein distances regarding the CDR3 sequences.

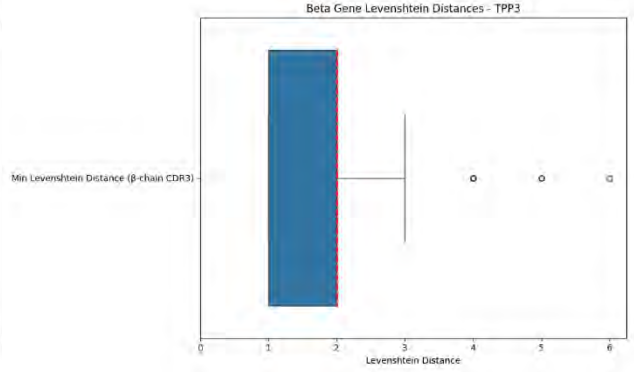

Paired Allele Levenshtein Distances - TPP2
Min Levenshtein Distance (β-chain CDR3)
Min Levenshtein Distance (α-chain CDR3)
0
2
4
6
8
10
Levenshtein Distance

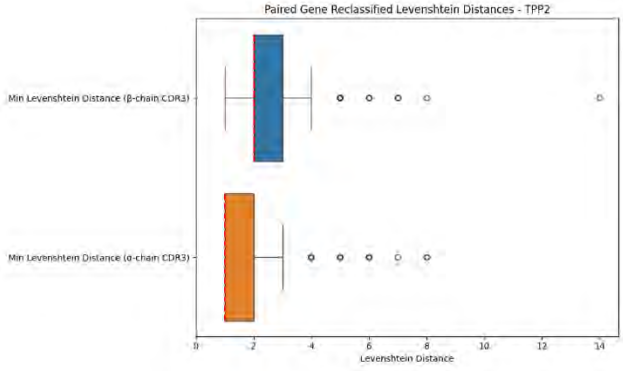

Paired Allele Levenshtein Distances - TPP3
Min Levenshtein Distance (β-chain CDR3)
Min Levenshtein Distance (α-chain CDR3)
0
1
2
3
4
5
6
Levenshtein Distance

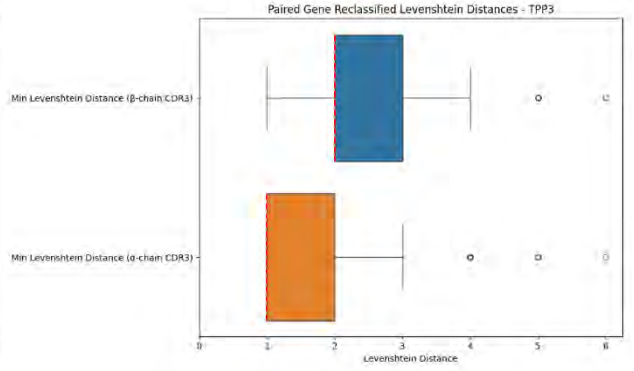

Paired Allele Reclassified Levenshtein Distances - TPP2
Min Levenshtein Distance (β-chain CDR3)
Min Levenshtein Distance (α-chain CDR3)
0
2
4
6
8
10
Levenshtein Distance


Paired Allele Reclassified Levenshtein Distances - TPP3
Min Levenshtein Distance (β-chain CDR3)
Min Levenshtein Distance (α-chain CDR3)
0
1
2
3
4
5
Levenshtein Distance


Beta Gene Levenshtein Distances - TPP2
Min Levenshtein Distance (β-chain CDR3)
0
2
4
6
8
10
12
14
Levenshtein Distance


Beta Gene Levenshtein Distances - TPP3
Min Levenshtein Distance (β-chain CDR3)
0
1
2
3
4
5
6
Levenshtein Distance


Paired Gene Reclassified Levenshtein Distances - TPP2
Min Levenshtein Distance (β-chain CDR3)
Min Levenshtein Distance (α-chain CDR3)
0
2
4
6
8
10
12
14
Levenshtein Distance


Paired Gene Reclassified Levenshtein Distances - TPP3
Min Levenshtein Distance (β-chain CDR3)
Min Levenshtein Distance (α-chain CDR3)
0
1
2
3
4
5
6
Levenshtein Distance

10.3 Appendix C: Run Name to Performance Mapping Table

The Run Name corresponds to the model's name.

Gene Precision								
Paired Vanilla Model	global	Run Name	TPP1	Run Name	TPP2	Run Name	TPP3	Run Name
AP	0.8469	trim-sweep-23	0.9366	playful-sweep-17	0.8035	trim-sweep-23	0.7432	valiant-sweep-15
ROC AUC	0.8664	fanciful-sweep-31	0.9477	proud-sweep-14	0.8234	faithful-sweep-27	0.7229	wandering-sweep-24
Paired Physicochemical Model	global	Run Name	TPP1	Run Name	TPP2	Run Name	TPP3	Run Name
AP	0.6029	kind-sweep-9	0.6341	kind-sweep-9	0.5943	kind-sweep-9	0.6207	comic-sweep-3
ROC AUC	0.6552	kind-sweep-9	0.6863	kind-sweep-9	0.6409	kind-sweep-9	0.6133	comic-sweep-3
Beta Vanilla Model	global	Run Name	TPP1	Run Name	TPP2	Run Name	TPP3	Run Name
AP	0.7517	helpful-sweep-11	0.8139	hearty-sweep-8	0.7644	fragrant-sweep-14	0.8103	elated-sweep-13
ROC AUC	0.7474	helpful-sweep-11	0.8049	hearty-sweep-8	0.7848	fragrant-sweep-14	0.6558	neat-sweep-5
Beta Physicochemical Model	global	Run Name	TPP1	Run Name	TPP2	Run Name	TPP3	Run Name
AP	0.7535	misty-sweep-9	0.8148	tough-sweep-10	0.7601	feasible-sweep-11	0.8043	feasible-sweep-11
ROC AUC	0.7490	misty-sweep-9	0.7977	tough-sweep-10	0.7735	feasible-sweep-11	0.6390	fragrant-sweep-7
Allele Precision								
Paired Vanilla Model	global	Run Name	TPP1	Run Name	TPP2	Run Name	TPP3	Run Name
AP	0.8954	resilient-sweep-17	0.9230	resilient-sweep-17	0.8751	deep-sweep-15	0.8132	sandy-sweep-9
ROC AUC	0.9009	resilient-sweep-17	0.9288	resilient-sweep-17	0.8785	deep-sweep-15	0.7581	deep-sweep-15

Paired Physicochemical Model	global	Run Name	TPP1	Run Name	TPP2	Run Name	TPP3	Run Name
AP	0.6443	laced-sweep-12	0.6683	laced-sweep-12	0.6219	laced-sweep-12	0.7147	brisk-sweep-17
ROC AUC	0.6777	laced-sweep-12	0.7069	laced-sweep-12	0.6494	laced-sweep-12	0.6503	leafy-sweep-29
Beta Vanilla Model	global	Run Name	TPP1	Run Name	TPP2	Run Name	TPP3	Run Name
AP	0.8115	olive-sweep-10	0.8063	olive-sweep-10	0.8163	olive-sweep-10	0.9352	glad-sweep-9
ROC AUC	0.8367	colorful-sweep-12	0.8303	olive-sweep-10	0.8485	colorful-sweep-12	0.8283	glad-sweep-9
Beta Physicochemical Model	global	Run Name	TPP1	Run Name	TPP2	Run Name	TPP3	Run Name
AP	0.8059	sunny-sweep-7	0.8006	misty-sweep-5	0.8107	sunny-sweep-7	0.9195	sunny-sweep-7
ROC AUC	0.8254	sunny-sweep-7	0.8259	sunny-sweep-7	0.8251	sunny-sweep-7	0.7932	sunny-sweep-7

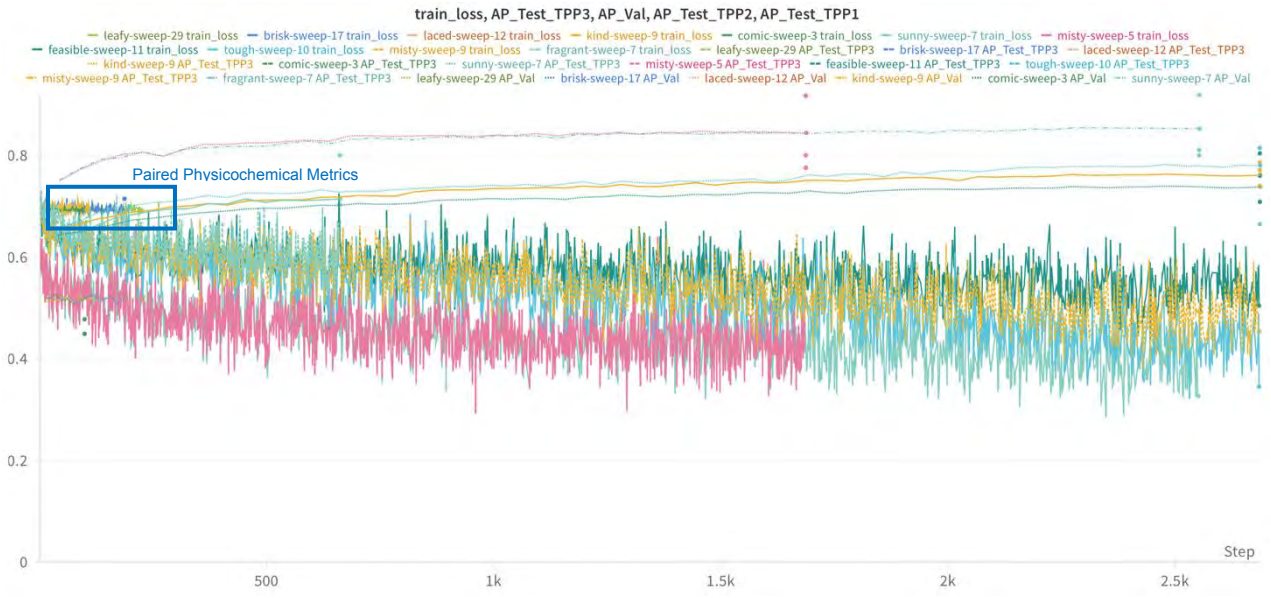
10.4 Appendix D: Table Reclassified All Tasks

Gene								
Paired Physicochemical Model	global (normal)	global (reclassified)	TPP1 (normal)	TPP1 (reclassified)	TPP2 (normal)	TPP2 (reclassified)	TPP3 (normal)	TPP3 (reclassified)
AP	0.6029	0.6029	0.6341	0.6127	0.5943	0.6097	0.6207	0.6900
ROC AUC	0.6552	0.6552	0.6863	0.6622	0.6409	0.6560	0.6133	0.5692
Allele								
Paired Physicochemical Model	global (normal)	global (reclassified)	TPP1 (normal)	TPP1 (reclassified)	TPP2 (normal)	TPP2 (reclassified)	TPP3 (normal)	TPP3 (reclassified)
AP	0.6443	0.6443	0.6683	0.6429	0.6219	0.6517	0.7147	0.8129
ROC AUC	0.6777	0.6777	0.7069	0.6844	0.6494	0.6581	0.6503	0.6298

10.5 Appendix E: Illustration of All Physicochemical Properties and Their Metrics Evaluations

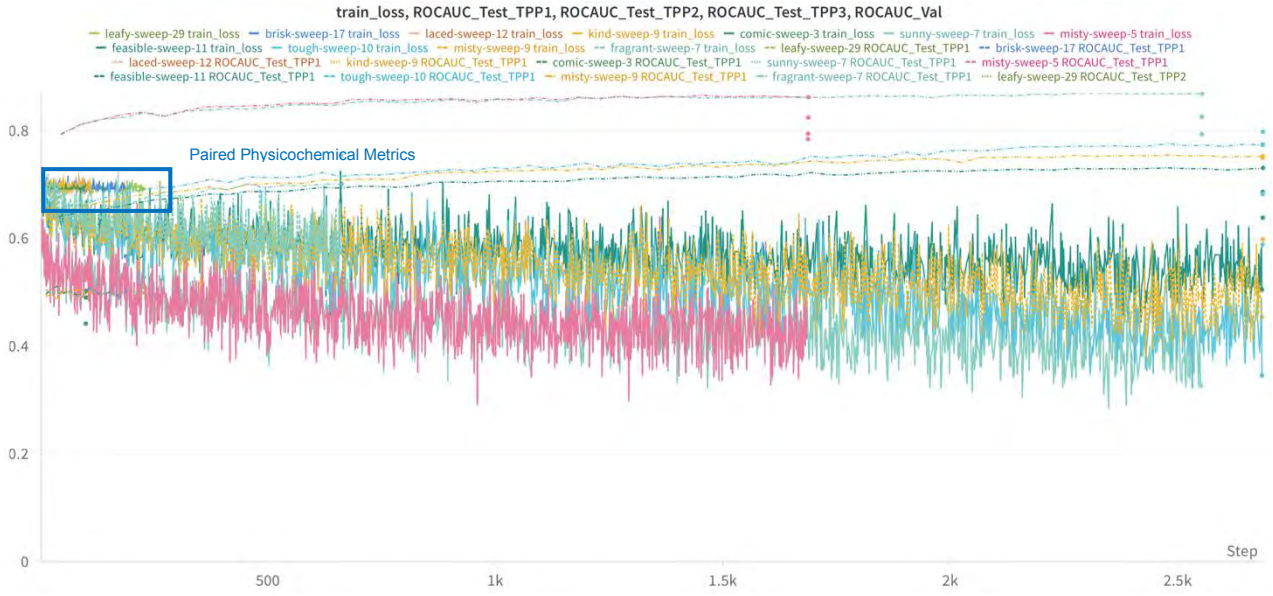
10.5.1 Illustration of the AP Metrics in Combination with the Loss

Comparison of training loss and Average Precision (AP) metrics across multiple Physicochemical models over training steps. Each line represents a different model variant, showing the evolution of train_loss, AP_Test_TPP3, AP_Val, AP_Test_TPP2, and AP_Test_TPP1



10.5.2 Illustration of the ROC AUC Metrics in Combination with the Loss

Performance comparison of multiple Physicochemical models based on training loss and ROC AUC metrics across different training steps. Each line represents a different model variant, depicting the progression of train_loss, ROCAUC_Test_TPP1, ROCAUC_Test_TPP2, ROCAUC_Test_TPP3 and ROCAUC_Val.


train_loss, ROCAUC_Test_TPP1, ROCAUC_Test_TPP2, ROCAUC_Test_TPP3, ROCAUC_Val
leafy-sweep-29 train_loss
brisk-sweep-17 train_loss
laced-sweep-12 train_loss
kind-sweep-9 train_loss
comic-sweep-3 train_loss
sunny-sweep-7 train_loss
misty-sweep-5 train_loss
feasible-sweep-11 train_loss
tough-sweep-10 train_loss
misty-sweep-9 train_loss
fragrant-sweep-7 train_loss
leafy-sweep-29 ROCAUC_Test_TPP1
brisk-sweep-17 ROCAUC_Test_TPP1
laced-sweep-12 ROCAUC_Test_TPP1
kind-sweep-9 ROCAUC_Test_TPP1
comic-sweep-3 ROCAUC_Test_TPP1
sunny-sweep-7 ROCAUC_Test_TPP1
misty-sweep-5 ROCAUC_Test_TPP1
feasible-sweep-11 ROCAUC_Test_TPP1
tough-sweep-10 ROCAUC_Test_TPP1
misty-sweep-9 ROCAUC_Test_TPP1
fragrant-sweep-7 ROCAUC_Test_TPP1
leafy-sweep-29 ROCAUC_Test_TPP2
Paired Physicochemical Metrics
0.8
0.6
0.4
0.2
0
500
1k
1.5k
2k
2.5k
Step

10.6 Appendix F: Source Code

The code and initial datasets of this work, along with all additional instructions necessary to reproduce or build upon it, are accessible to the public via:

https://github.com/vegger/BA_ZHAW