Bachelor's Thesis Computing Science

Predicting T Cell Receptor Specificity Using TULIP

Ivan Kothaj
s1113234

June 14, 2026

First supervisor/assessor:
Dr. Johannes C. Textor

Second assessor:
Dr. Inge M. N. Wortel

Daily supervisor:
Dr. Farzaneh M. Parizi

Radboud University

Abstract

To understand the functioning of our immune system better, it is necessary to know whether a given TCR-pMHC (T Cell Receptor peptide-Major Histocompatibility Complex) pair binds. This problem is difficult due to vast variability of TCRs. Therefore, various machine learning approaches have been used to create models that are capable of predicting TCR specificity. In this study, we apply pre-trained TULIP (Transformer-based Unsupervised Language model for Interacting Peptides), without additional fine-tuning, to predict TCR specificity on the BATCAVE (Benchmark for Activation of T Cells with Cross-Reactive Avidity for Epitopes) database. TULIP has not seen the data from this database before, and hence evaluating its performance on this data is informative, as T cell recognition models often struggle on previously unseen data. We conducted two sets of experiments, first on the larger subset of the BATCAVE database, and computed AUC (Area Under the Curve) = 0.544. This result demonstrates limited predictive capability for TCR specificity. The second set of experiments was conducted on a smaller subset, structured to minimise class imbalance, and containing different mutants of an index peptide paired with the same TCR. We achieved AUC = 0.572, a more promising result; however, the Pearson correlation p-value of ≈ 0.32 substantially exceeds the significance threshold of 0.05, indicating that correlation between predicted scores and peptide activity is not statistically significant. We conclude that TULIP is not a preferable way to predict TCR specificity on the BATCAVE database, and recommend BATMAN (Bayesian Inference of Activation of T Cell Receptor by Mutant Antigens) as a more suitable option, given its training on this database. More broadly, current T cell recognition models are not yet reliable enough for real-world deployment, and future efforts should focus on improving the quality and quantity of available databases and developing models capable of generalising to unseen peptides.

Contents

List of Acronyms

AUC	Area Under the Curve
AUC-ROC	Area Under the Receiver Operating Characteristic curve
BATCAVE	Benchmark for Activation of T Cells with Cross-Reactive Avidity for Epitopes
BATMAN	Bayesian Inference of Activation of T Cell Receptor by Mutant Antigens
BERT	Bidirectional Encoder Representations from Transformers
CDR	Complementarity Determining Region
HLA	Human Leukocyte Antigen
IEDB	Immune Epitope Database
LOO-TCR	Leave-One-Out T Cell Receptor
McPAS-TCR	Manually Curated Catalogue of Pathology-Associated T Cell Receptor Sequences
MHC	Major Histocompatibility Complex
MLM	Masked Language Model
NLP	Natural Language Processing
PLM	Protein Language Model
pMHC	peptide-Major Histocompatibility Complex
ROC	Receiver Operating Characteristic
TCR	T Cell Receptor
TCR-pMHC	T Cell Receptor peptide-Major Histocompatibility complex
TULIP	Transformer-based Unsupervised Language model for Interacting Peptides

Chapter 1

Introduction

T cell recognition refers to the process by which T cells scan peptides presented on the surface of a cell and determine whether an immune reaction is necessary by distinguishing self-antigens from foreign antigens. Central to this process is T cell receptor (TCR) specificity, which refers to the ability of a TCR to selectively bind to a particular peptide-major histocompatibility complex (pMHC), determined by the structural properties of the TCR.

T cells have great cross-reactivity that makes them truly complex [1]. Such variability is important to study, as it can provide insight into various immunity-related phenomena such as autoimmune disease diagnostics, vaccine development, cancer treatment, and immunotherapy.

Over the past few years, several machine learning methods (from matrix-based approaches to protein language models) have emerged with the goal of predicting whether a TCR binds to a particular peptide based on amino acid composition. One such model is TULIP, a transformer-based unsupervised language model [2]. Specifically, when presented with a peptide, it is capable of predicting the binding likelihood to a given TCR. It is important to evaluate how such models perform on new datasets and whether they generalise beyond their original training context.

The aim of this research is to evaluate TULIP on the task of T cell recognition prediction and benchmark its performance against existing models. Given the relevance of this task to understanding autoimmune diseases and vaccine development, improving predictive accuracy remains an important research goal. We further specify our goal with following research questions:

Question 1: To what extent do TULIP scores predict TCR specificity on the BATCAVE database?

Question 2: Can TULIP predict how different mutants of the same index peptide activate a given TCR?

Question 3: Does TULIP achieve sufficient predictive performance to be used for further research?

Motivation behind choosing TULIP is that it ranks among top-performing T cell recognition models in recent benchmarking studies [3, 4, 5]. Additionally, unlike most competing models which rely on labelled training data, TULIP is unsupervised. This makes it particularly interesting to investigate whether an unsupervised approach could be a preferred method over a supervised one for T cell recognition in the future.

TULIP's transformer-based architecture is well-suited for the task of T cell recognition. Its bidirectional attention mechanism allows it to capture the full sequence context of the TCR-pMHC complex, enabling the model to predict binding likelihood, as binding depends on the overall structural compatibility between TCR and pMHC. We evaluate its performance using Pearson correlation and AUC. Our results indicate that TULIP achieves performance ranging from marginal (AUC ≈ 0.52) to modest (AUC ≈ 0.57).

The field of T cell recognition is advancing rapidly. New models are developed each year and evaluated in benchmarking competitions to identify the most successful approaches. In this study, we evaluate TULIP on data from the BATCAVE database, enabling us to compare it with the performance reported for BATMAN on the same data [6, 7].

We introduce necessary biological and technical background in **Chapter 2**. Datasets and models that were used are described in **Chapter 3**. We provide broader look into the field of T cell recognition in **Chapter 4**. Our experimental findings and corresponding evaluation methodology are detailed in **Chapter 5**. Finally, we summarise the contributions and limitations of this study in **Chapter 6**.

Chapter 2

Preliminaries

In this chapter, we introduce concepts that are necessary to understand our research. Following sections provide background on the immune system and T cells, structural composition of transformers and their various types, and statistical metrics that were used for evaluation of our results.

2.1 Background into immune system

Immune system plays an important role in protection against cancer cells and infections. One of the key components of the immune system are T cells. T cells play a role in eliminating abnormal cells, such as those presenting viral antigens. They achieve this, through their T cell receptor (TCR), by distinguishing between self-antigens, which are naturally present in the body, and foreign invaders, such as those from viruses or bacteria [2]. In this context, T cells do not recognise whole antigens directly. Instead, they recognise short fragments called peptides. Peptides that can be recognised by a TCR are referred to as epitopes. For further reading, you can learn more about T cells in Fabbri, Smart, and Pardi [8].

2.1.1 Structure of T cell receptor

The T cell receptor (TCR) is a protein located on the surface of a T cell. Most TCRs consist of two protein subparts called the α and β chain [9]. Structure of TCR is shown in Figure 2.1.

The binding site of the TCR consists of six complementarity determining regions (CDRs), with three CDRs per chain: CDR1, CDR2, and CDR3 [9]. CDR1 and CDR2 do not vary significantly between chains, whereas the CDR3 is crucial in determining which antigens can be recognised by the TCR [9, 10].

The CDR3 amino acid sequence diversity of the TCRβ chain alone is estimated to be 5×10^{11} [10]. Based on this, Joglekar and Li [11] estimates the total diversity of the TCR repertoire in a single individual to be in the range of 10^{11}–10^{12} unique TCRs.

TCR recognition is supported by co-receptors CD4 and CD8, which are non-clonally distributed proteins that co-recognise ligands of the TCR [12]. CD4 is expressed on helper or regulatory T cells and binds MHC class II molecules, while CD8 is expressed on cytotoxic T cells and binds MHC class I molecules [12].

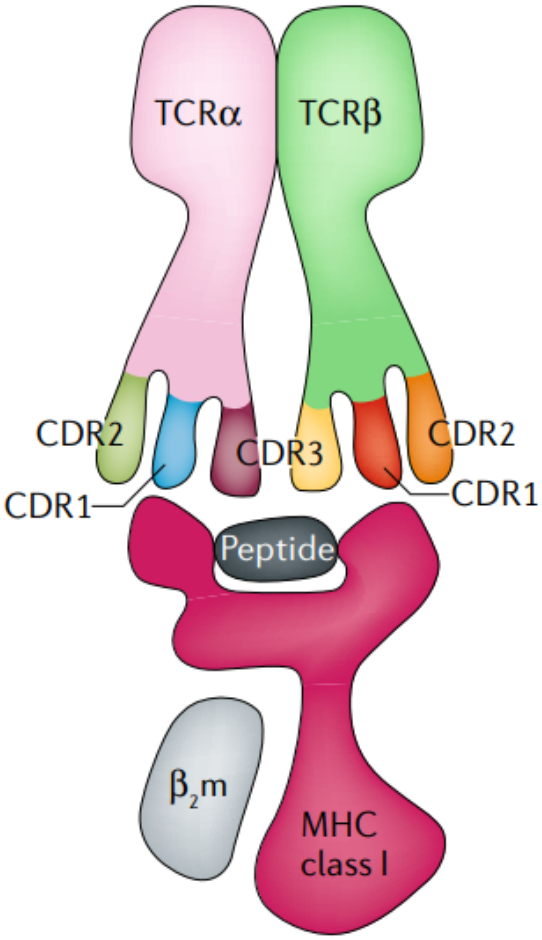


Figure 2.1: Schematic representation of TCR-pMHC binding taken from [13].

2.1.2 Types of MHC and their importance

Major histocompatibility complex (MHC) molecules partake in the presentation of protein fragments from inside the cell, called peptide antigens, to T cells [14]. MHC molecules thus form the molecular basis of TCR-peptide-MHC (TCR-pMHC) binding, as is shown in Figure 2.1, which is crucial for T cell recognition and immune system function.

According to Janeway et al. [14], human leukocyte antigen (HLA) genes encode MHC molecules in humans. MHC molecules are divided into two classes: class I and class II. MHC class I molecules are encoded by HLA-A, HLA-B, and HLA-C loci (specific positions on chromosomes where genes are located) and present intracellularly derived peptides to $CD8^+$ T cells. MHC class II molecules are primarily encoded by HLA-DR, HLA-DP, and HLA-DQ loci and present extracellularly derived peptides to $CD4^+$ T cells.

The purpose of MHC class I molecules is to alert the immune system about intracellular (inside the cell) infection to target infected cells for destruction, whereas MHC class II molecules guide intercellular co-operation between immune cells in an immune response and help to combat

extracellular (outside the cell) infections [12].

2.2 Transformers and their function

The Transformer is a neural network architecture comprising an encoder-decoder structure introduced by Vaswani et al. [15].

2.2.1 Encoder and decoder

The encoder transforms the input sequence into a hidden representation, while the decoder generates the output sequence from the encoder's representation [16], as illustrated in Figure 2.2. The encoder is composed of a multi-head self-attention mechanism followed by a position-wise feed-forward neural network.

The decoder has a similar composition as the encoder, but additionally contains an encoder-decoder attention mechanism [16]. The decoder can attend to all positions thanks to its self-attention layers [15]. It uses `softmax` function, described in subsection 2.3.1, on its output to predict distribution of probabilities for the next token [15].

Each layer of both encoder and decoder contains a fully connected feed-forward neural network. Every feed-forward network contains a hidden layer with weights and an activation function that applies a linear transformation to the input [15].

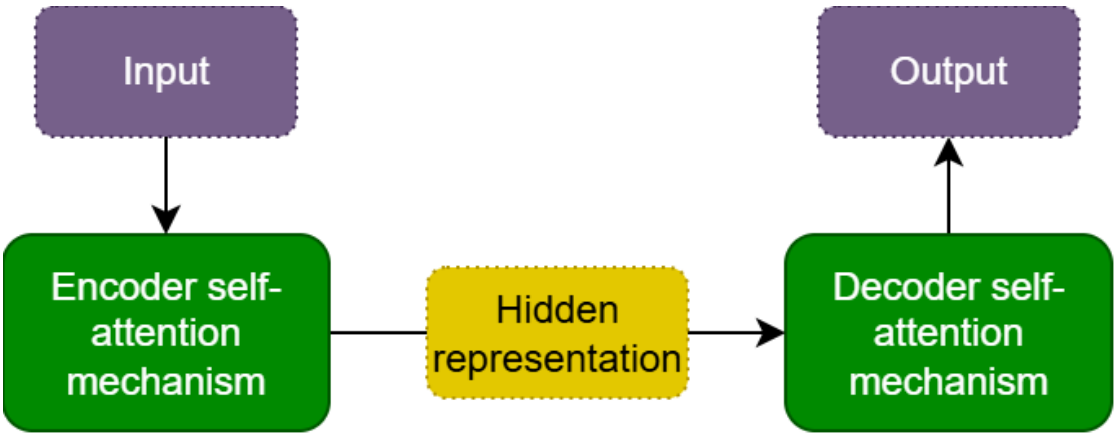


Figure 2.2: Simplified encoder-decoder architecture where encoder takes input and transforms it into a hidden representation, which is then used by decoder to compute an output.

2.2.2 Self-attention mechanism

Self-attention is a mechanism that allows a model to compute a representation of a sequence by relating each position to all other positions within the same sequence [15]. Self-attention operates on vector representations of queries, keys and values, where weights are computed based on the similarity between queries and keys and applied to the values to produce the output [15].

The multi-head self-attention mechanism is a significant improvement over previous approaches as it allows the model to apply self-attention in parallel across multiple representation subspaces as illustrated in Figure 2.3.

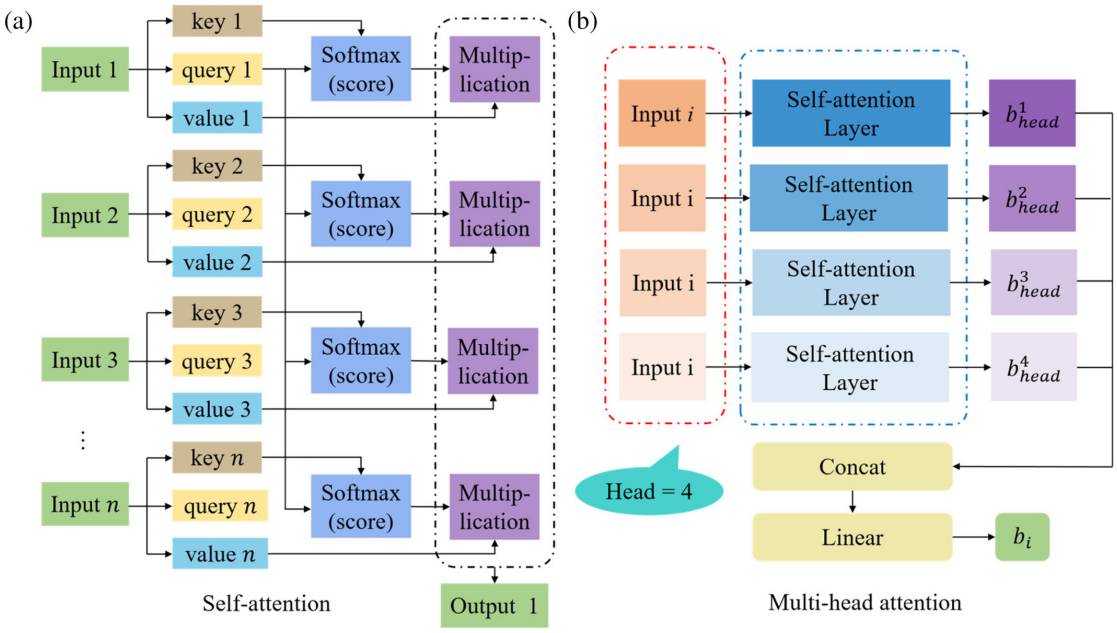


Figure 2.3: Self-attention mechanism (left) and multi-head self-attention mechanism (right) taken from [16].

2.2.3 Masked language models

Masked language models (MLMs) are a type of model trained using masked language learning, where a random token in a sequence is swapped with a mask token and the model has to predict it based on the other tokens in the sequence.

A well-known example of an MLM for natural language processing (NLP) tasks is Bidirectional Encoder Representations from Transformers (BERT) [17]. NLP is a field concerned with the computational processing and understanding of human language, encompassing tasks such as text classification and question answering.

Protein language models (PLMs) apply the principles of language modelling to biological sequences. TULIP, for instance, is a PLM that uses masked language modelling, training the model to predict masked amino acids in a sequence, whose order determines a specific protein. The functioning of both NLP and PLM models is illustrated in Figure 2.4.

2.2.4 Purpose of tokens

Tokens are a crucial part of MLMs as they segment the input data into discrete units and produce sequence representations that are later used for classification tasks [17], while also being necessary for training the MLMs. subsection 2.2.5 and subsection 2.2.6 describe specific tokens used by BERT and TULIP.

2.2.5 BERT tokens

The BERT paper introduced the following tokens as part of its pipeline for NLP tasks [17]:

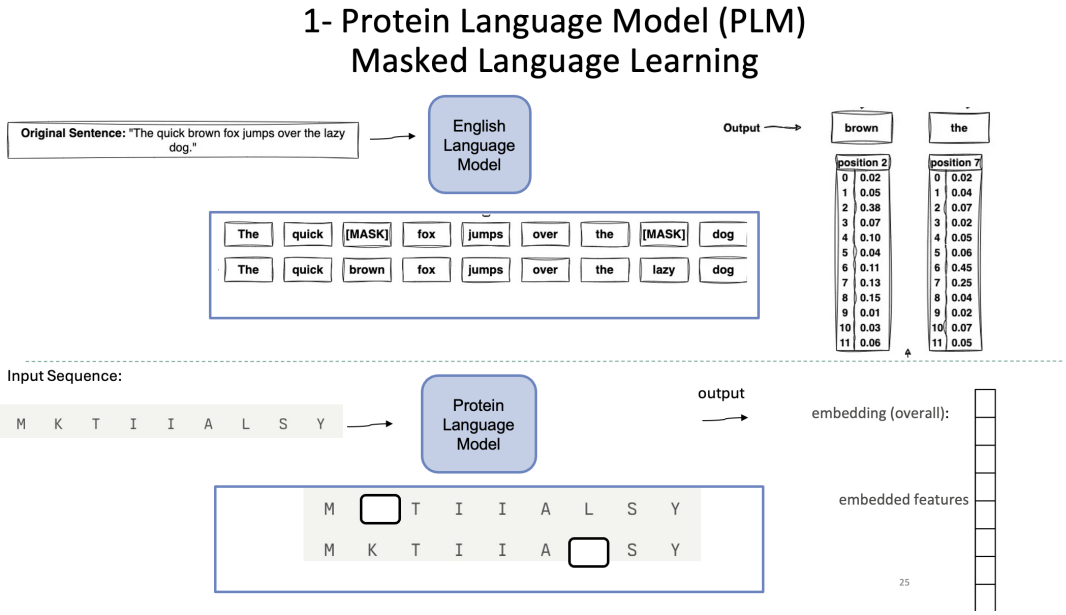


Figure 2.4: Comparison between two types of masked language models, natural language processing model (top) and protein language model (bottom).

`[CLS]` is a classification token prepended to every input sequence. Using self-attention, its final hidden state aggregates information from all other tokens in the sequence. It produces a single vector representation of the input that is used for classification.

`[SEP]` is a separator token that separates two segments from each other. This token is not directly used in TULIP, as it is replaced by two tokens `[SOS]` and `[EOS]` which perform a similar task.

`[MASK]` is a token used to replace the target token that the MLM has to predict during the training phase.

2.2.6 TULIP tokens

TULIP introduces several additional tokens beyond those defined in the BERT paper [2]:

`[PAD]` is a token used to pad sequences to a uniform length.

`[SOS]` is a token that denotes the start of the input sequence.

`[EOS]` is a token that denotes the end of the input sequence.

`[UNK]` is a token that denotes unknown characters.

`[MIS]` is a token that denotes missing information at a given position in the input sequence. It is used to handle incomplete data entries, which are common in biological databases, by explicitly marking absent values.

2.2.7 TULIP architecture

There exist numerous variants of the Transformer architecture. In this work, we focus on the transformer-based unsupervised language model TULIP, which was designed for TCR specificity prediction [2].

TULIP is an MLM model based on the BERT framework introduced by Devlin et al. [17]. This architecture enables bidirectional contextual learning and can be readily adapted to various downstream tasks, including T cell recognition.

TULIP has three encoders and three decoders, each specialised based on their input. As illustrated in Figure 2.5, α `Encoder`, β `Encoder`, and `e Encoder` take as input an embedded representation of CDRα, CDRβ, and epitope, respectively. Each of them computes a hidden representation that is passed to the decoders.

α `Decoder`, β `Decoder`, and `e Decoder` compute the log-likelihood of their associated variable conditioned on the hidden representation from the other two encoders and the embedded MHC representation.

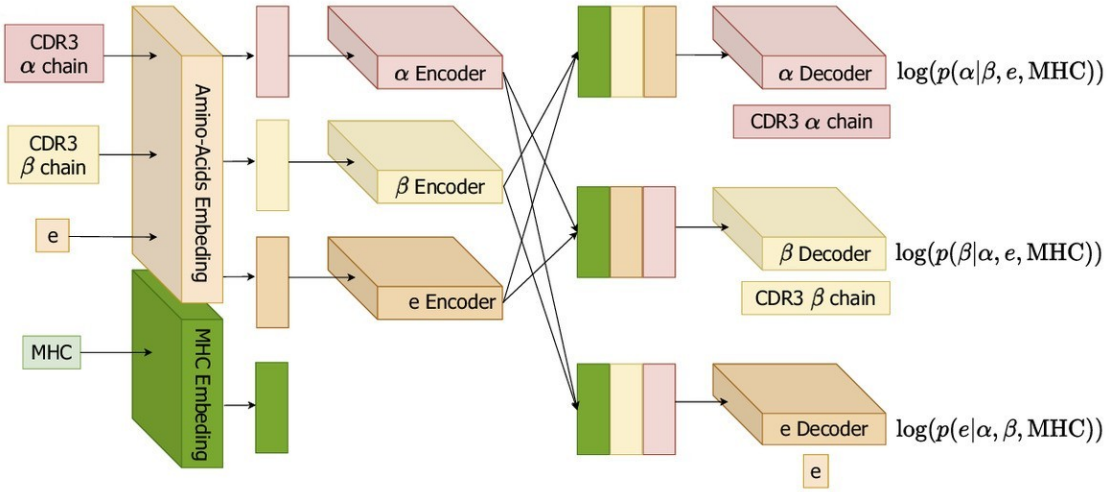


Figure 2.5: TULIP encoder-decoder architecture taken from [2].

2.3 Statistical and computational methods

In this section, we introduce statistical and machine learning metrics that are either a part of TULIP's architecture or used by us to evaluate its performance.

2.3.1 Softmax function

The `softmax` function converts a collection of values into a probability distribution by computing the exponentials of each element divided by the sum of all exponentials [18]. It is defined as:

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_k e^{x_k}} \tag{2.1}$$

where x_j represents a value of a vector x at position j and e represents Euler's number.

2.3.2 Pearson correlation

Pearson's correlation coefficient r, introduced by Pearson [19], measures linear association between two continuous variables. Coefficient r ranges from -1 to 1 where 0 means no linear correlation, a positive correlation indicates that as one variable increases so does the other, while a negative correlation indicates that as one variable increases the other decreases.

The p-value is used to determine the statistical significance of the correlation, where p-value $<$ 0.05 represents a significant finding. It denotes the probability of observing the computed correlation, under the null hypothesis of no linear correlation between two variables. In this study, we compute r using `scipy.stats` [18], defined as:

$$r = \frac{\sum(x - m_x)(y - m_y)}{\sqrt{\sum(x - m_x)^2 \sum(y - m_y)^2}} \tag{2.2}$$

where m_x and m_y are the means of input arrays x and y respectively.

2.3.3 Area under the ROC curve

Receiver operating characteristic (ROC) curves are used to evaluate and visualise the performance of classifiers [20]. They plot the true positive rate against false positive rate across different classification thresholds. They are often used for domains with skewed class distribution and unequal classification error costs.

Area under the ROC curve (AUC) summarises the overall performance of the classifier as a single scalar value, representing the probability that the model ranks a randomly chosen positive instance higher than a randomly chosen negative one [20]. The confusion matrix, in Figure 2.6, visualises how true and false positive rates are computed.

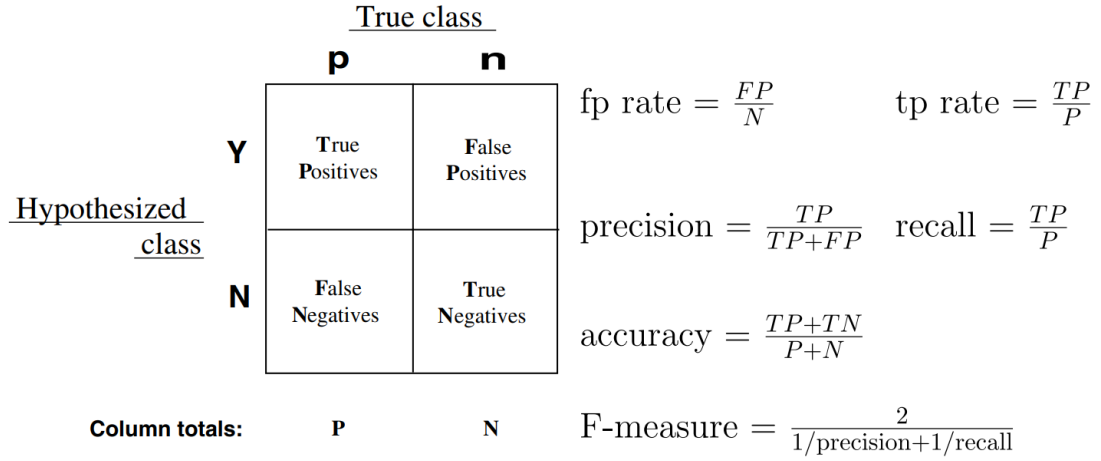


Figure 2.6: Confusion matrix and performance metrics as is found in [20].

Some problems require the false-positive rate to be below a specific threshold; otherwise the particular approach is not useful [21]. This is particularly relevant in medical applications,

where false positives can have serious consequences, motivating the use of partial AUC metrics. $\mathrm{AUC}_{0.1}$ is a partial AUC metric that integrates the ROC curve only up to a false-positive rate of 0.1 [21]. The following formula is taken from [5], which will be further discussed in **Chapter 4**:

$$\text{average } \mathrm{AUC}_{0.1} = \frac{1}{N} \sum_{1}^{N} \mathrm{AUC}_{0.1}(p) \tag{2.3}$$

where N is the number of peptides in the test set and $\mathrm{AUC}_{0.1}(p)$ is $\mathrm{AUC}_{0.1}$ for peptide p.

Chapter 3

Methods

In this chapter, we describe how we processed our datasets and explain how we adapted the original TULIP implementation, obtained from [22], for the purpose of this study. The datasets used for evaluation were obtained from [7] and preprocessed as described in section 3.1.

3.1 Datasets

TULIP has been trained on various databases as described in its paper [2] which are further discussed in section 4.1. To evaluate TULIP's performance on T cell recognition, we use the data from the BATCAVE database [7] which is examined in more detail in section 4.3. We developed a script called `extract_database.py` that preprocesses the datasets as follows:

- removes entries corresponding to mouse data, retaining only human samples
- removes entries that are missing CDR3 sequence information
- retains only the following columns: `index`, `CDR3`α, `CDR3`β, `MHC`, `peptide`, `activation`, `peptide activity`

Additionally, the `activation` column, renamed to `binder`, was converted from ternary labels (0: non-binder, 1: weak binder, 2: strong binder) to binary labels (0: non-binder, 1: binder), where weak and strong binders were merged into a single positive class. This data transformation is required for AUC computation, which operates on binary class labels.

We created two subsets from the BATCAVE database, namely `BATCAVE_subset.csv` and `mutant_subset.csv`, that are used for experiments regarding the first and second research questions, respectively. The `BATCAVE_subset.csv` contains over 10,000 entries, with multiple TCRs per peptide, whereas the `mutant_subset.csv` contains approximately 2,000 entries, with one TCR per `index peptide`. So each TCR is associated with multiple `mutants` of one `index peptide`. It contains an additional `index peptide` column and is derived from the subset, which was filtered to minimise class imbalance.

Only MHC class I molecules are included, as MHC class I is more extensively represented in available databases compared to MHC class II. Both subsets contain only peptides of length 9 and 10, and their representative samples are illustrated in section A.2.

The subsets include only CDR3 sequences, as CDR3 is the principal determinant of antigen binding specificity due to its direct interaction with the peptide [23]. Therefore, CDR3 serves as the main input feature in T cell recognition models.

3.2 Model

The original TULIP model was trained on $n = 209{,}779$ non-redundant data entries containing an epitope and at least one chain of the TCR [2]. We downloaded the codebase from [22]. We did not pre-train or fine-tune the model as this was beyond the scope of this study due to computational constraints. We modified the prediction pipeline, specifically the component responsible for `score` computation after model initialisation and token setup.

The first set of experiments, addressing our first research question, was conducted on `BATCAVE_subset` with the following procedure.

We grouped different TCRs paired with the same peptide, denoted as `target peptide`. For each `target peptide`, TULIP predicted interaction `scores` for all corresponding TCR-pMHC pairs. The resulting `scores` were merged with `binder` and `peptide activity` columns into a single data frame for further analysis.

Additionally, we saved intermediate results containing Pearson correlation and AUC per `target peptide`. These intermediate results are not discussed further in this study; however, they are available at our GitHub repository for further inspection (see section A.1).

The second set of experiments, addressing our second research question, follows an almost identical procedure but was conducted on `mutant_subset` and intermediate results are not saved. Instead of grouping by `target peptide`, we grouped different `mutant peptides` sharing the same `index peptide` and TCR-pMHC pair. For each `index peptide` and its corresponding TCR-pMHC pair, TULIP predicted interaction `scores` for all associated `mutant peptides`.

For both sets of experiments, we computed the overall Pearson correlation on the merged data frame. We also computed AUC using `scikit-learn` [24]. A detailed description of our evaluation approach and findings is provided in **Chapter 5**. The modified code and datasets can be accessed via our GitHub repository (see section A.1).

Chapter 4

Related Work

In this chapter, we examine currently used TCR databases and the state-of-the-art models for T cell recognition. We also discuss BATMAN, whose performance is compared with TULIP in **Chapter 5**.

4.1 TCR Databases

As established in subsection 2.1.1, the TCR repertoire is enormous. This poses a challenge for T cell recognition models, which tend to achieve significantly higher predictive accuracy on data encountered during training or finetuning, referred to as seen data, compared to previously unseen data. An approach to minimise this problem is to train the models on various complementary databases.

McPAS-TCR (manually curated catalogue of pathology-associated T cell receptor sequences) is a database that focuses on the association between TCR sequences and various diseases. It collected data from 118 publications into one database that contains around 5,100 TCR sequences [25].

VDJdb is a database that focuses on TCR-pMHC binding interactions rather than disease association. It contains over 5,000 TCR sequences, which were obtained from new donor samples reported in more than 100 publications [26].

The immune epitope database (IEDB) contains data on B cell and T cell epitopes, along with their associated MHC molecules and peptides, with a focus on autoimmune, transplant, and allergic diseases. It has been continuously updated for over 20 years and in its latest update from 2024 contains around 1.6 million peptides from more than 25,000 publications [27].

VDJdb, McPAS-TCR and IEDB were used to train TULIP as well as the majority of the T cell recognition models that participated in the competitions described in section 4.2.

4.2 State-of-the-art in T cell recognition

Predicting the specificity of TCR-pMHC interactions is a difficult task. State-of-the-art models have made significant progress in the last couple of years. Many models achieved $\text{AUC}_{0.1} \geq 0.7$ depending on peptide; however, this performance was limited to peptides observed during training

[5]. When tested on previously unseen peptides their performance dropped significantly, staying only slightly above chance level [5].

Models are benchmarked in competitions, such as Immune Repertoire (IMMREP23), to identify the most effective approaches and further the research progress [5]. Recently, the field has shifted focus towards evaluating generalisation to unseen peptides. Current state-of-the-art models are improving, but still have a long way to go as the best ones achieved $AUC_{0.1}$ around 0.6 on unseen pMHC pairs.

These results were reported in the latest IMMREP25 competition [28]. TULIP was evaluated in this competition as well; however, it achieved $AUC_{0.1}$ between 0.50–0.51 on unseen peptides, falling behind the other models [28]. This shows how quickly this field is advancing, as TULIP was one of the best performing models in the previous competition in 2023, but in 2025 its performance had been surpassed by more recently developed models.

4.3 BATMAN

BATMAN (Bayesian Inference of Activation of TCR by Mutant Antigens) is a T cell recognition model that operates on different principles than TULIP [6]. Unlike the models discussed previously, BATMAN uses a distance matrix together with positional weights for prediction, focusing on the impact of single amino acid mutations on TCR activation.

BATMAN was trained on the BATCAVE (Benchmark for Activation of T Cells with Cross-Reactive Avidity for Epitopes) database. The BATCAVE database contains over 22,000 TCR-pMHC pairs with a focus on TCR cross-reactivity [6]. Models of this type and transformer-based models such as TULIP are typically developed and evaluated in different contexts, as they address different aspects of T cell recognition.

BATMAN reports within-TCR performance (mean AUC = 0.836), where the model is trained and evaluated on peptide data associated with the same TCR [6]. It also reports leave-one-out-TCR (LOO-TCR) performance (mean AUC = 0.687), where the model is trained on data from all but one TCR and evaluated on the held-out TCR, which is encountered only during testing [6]. These results are not directly comparable to the competition results discussed in section 4.2, as different datasets were used for training and evaluation.

Chapter 5

Results

In this chapter, we present and interpret the findings from our experiments. The following two sections address the first and second research questions, respectively. All figures in this chapter were created using `matplotlib` [29].

5.1 TULIP performance on the BATCAVE subset

TULIP computes the probability of binding using the following formula taken from [2]:

$$\log(p(e|\alpha, \beta, \text{MHC})) - \log(p(e|\text{MHC})) \tag{5.1}$$

where $p(e|\alpha, \beta, \text{MHC})$ is the probability of epitope e given TCRα, TCRβ and MHC, and $p(e|\text{MHC})$ is the probability of epitope e given only MHC.

The resulting log scores are converted into a probability distribution using the `softmax` function. TULIP returns `scores`, which are negated, such that higher values correspond to stronger predicted binders, and the final values range from approximately -3 to -60, where the closer the value is to 0, the higher the binding probability. Based on this, TULIP ranks the TCR-pMHC complexes from the most to the least likely binders for each target peptide.

We plotted `peptide activity` from `BATCAVE_subset` against `score` predicted by TULIP illustrating the relationship between the two, which can be seen in Figure 5.1a. We computed the overall Pearson correlation of $r = -0.044$ and a corresponding p-value of 5.3×10^{-6} on these data.

Pearson correlation is around 0, which indicates that there is neither positive nor negative correlation between our `score` and the `peptide activity`. The p-value of $5.3 \times 10^{-6} < 0.05$ indicates statistical significance; however, in this context it confirms that the absence of correlation is a statistically reliable finding, not a result of insufficient data.

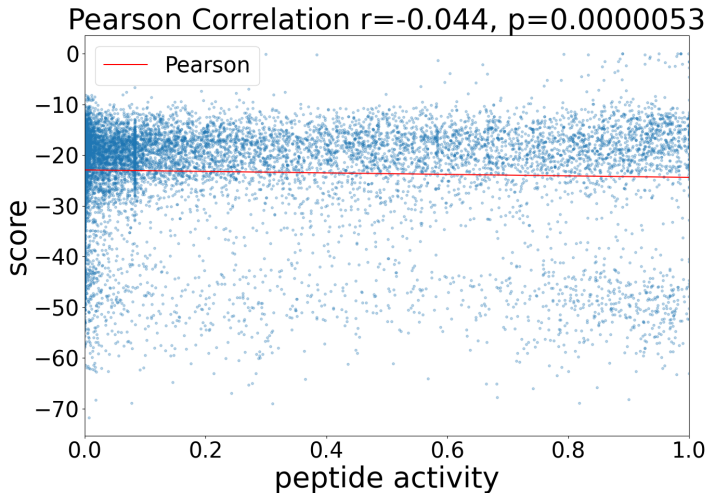
We computed both AUC and $\text{AUC}_{0.1}$ to compare TULIP's performance both with the IMMREP competitions results and with BATMAN. Both metrics were computed on `binder` from `BATCAVE_subset` and `score`. We obtained an AUC of 0.544, as shown in Figure 5.1b. The $\text{AUC}_{0.1}$ was 0.522, as shown in Figure 5.1e. These results indicate that TULIP performs only marginally above chance level, with both metrics exceeding 0.5 by less than 0.05.

We also attempted binary classification using a threshold, where we chose a boundary threshold of -20, selected empirically after testing multiple threshold values. We transformed our data by creating a new column called `predicted_binder`, assigning class 1 if the `score` exceeds the threshold and class 0 otherwise.

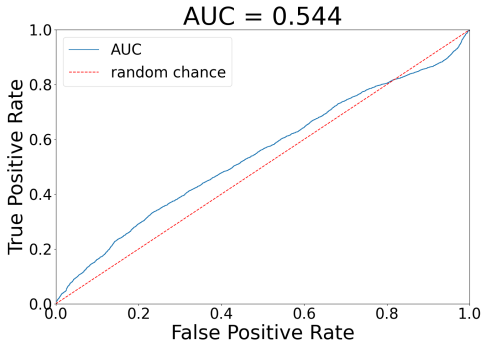
Using this transformed data, we computed AUC between our `predicted_binder` and `binder` from `BATCAVE_subset`. This was motivated by visual inspection of Figure 5.1a, which suggested a possible linear separation between binders and non-binders. However, our result was AUC = 0.534, which was lower than the AUC on the non-transformed data, suggesting that no clear linear decision boundary exists at this threshold.

Furthermore, we investigated whether TULIP incorporates MHC type into its predictions and whether doing so improves predictive accuracy. In Figure 5.1d, we present AUC = 0.525, computed on a modified version of our dataset in which all MHC types were replaced with a uniform placeholder string. Since this AUC was lower than the AUC of 0.544 obtained on the original data, shown in Figure 5.1b. We conclude that TULIP utilises MHC information and that its inclusion improves predictive accuracy.

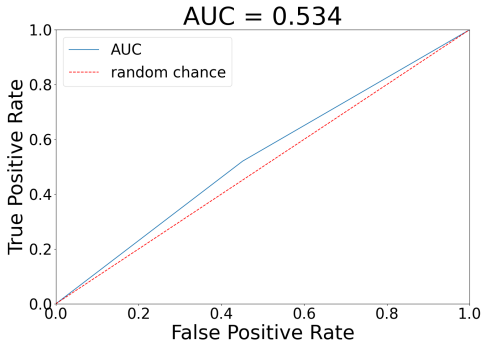
This experiment is particularly relevant because some datasets do not include HLA typing, and therefore it is important to be aware of a possible drop in accuracy on such data. Additionally, this experiment confirmed that TULIP can process input with incorrect MHCs, as it did not crash but rather still ran normally by using `[UNK]` token to replace nonsensical data. However, it is important to note that if the MHC column is completely missing in the input file, then the program will crash, as identified through testing. The reason is that TULIP expects certain types of headers in its input file, so if the data do not contain MHC column, it is necessary to modify the input by adding dummy values.



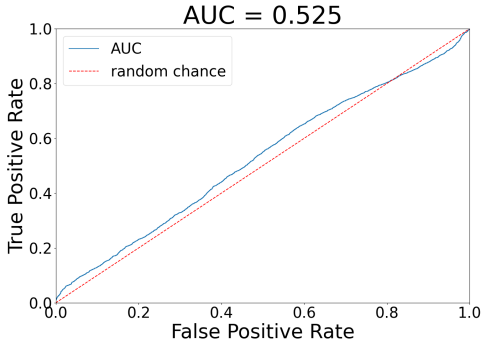
(a) Scatter plot of TULIP `scores` against `peptide activity` with Pearson correlation $r = -0.044$.



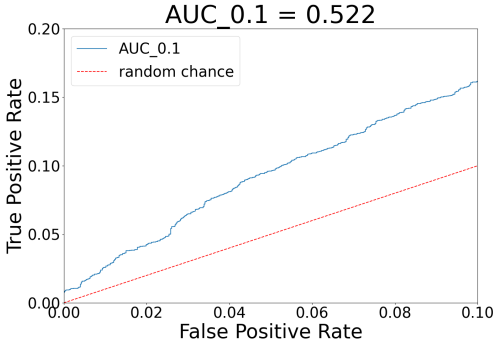
(b) ROC curve between `binder` and `score` with AUC = 0.544.



(c) ROC curve for binary threshold classification between `binder` and `predicted binder` with AUC = 0.534.



(d) ROC curve between `binder` and `score` on data excluding MHCs with AUC = 0.525.



(e) Partial ROC curve between `binder` and `score` with $AUC_{0.1} = 0.522$.

Figure 5.1: Evaluation of TULIP's prediction performance on `BATCAVE_subset`.

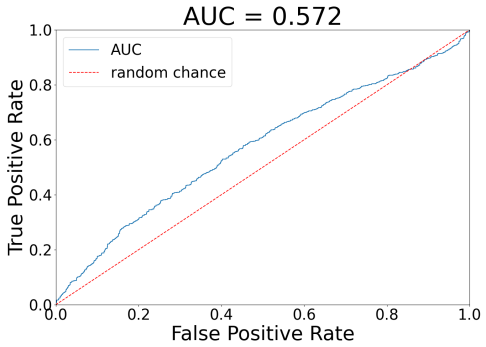
5.2 TULIP mutant prediction

We investigated whether TULIP can predict on data consisting of `mutant peptides` sharing the same `index peptide`-TCR pair.

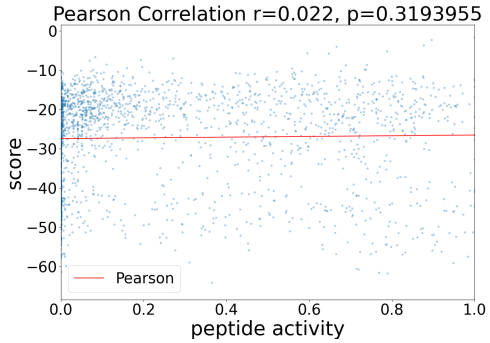
We computed AUC = 0.572 between `binder` from `mutant_subset` and `score`, as illustrated in Figure 5.2a. This AUC of 0.572 is higher than the 0.544 obtained on `BATCAVE_subset`. We propose two possible explanations for this: first, the original dataset was constructed to minimise class imbalance; second, TULIP groups predictions by `index peptide` rather than `target peptide` as in the previous experiment, while using `mutant peptide` for `score` prediction.

Furthermore, we assessed whether `scores` are significantly correlated with `peptide activity`, as illustrated in Figure 5.2b. We computed Pearson correlation of $r = 0.022$ and p-value of ≈ 0.32. The p-value of ≈ 0.32 exceeds the significance threshold of 0.05, indicating that the observed correlation is not statistically significant.

One possible explanation, for the high p-value, we propose is that by grouping on the `index peptide` but computing `score` for `mutant peptide`, we introduce an inconsistency between the grouping criterion and the scored peptide.



(a) ROC curve between `binder` and `score` for mutant peptides with AUC = 0.572.



(b) Scatter plot of `scores` against `peptide activity` for mutant peptides with Pearson correlation $r = 0.022$.

Chapter 6

Conclusions

6.1 Addressing the research questions

To address our first research question—**To what extent do TULIP scores predict TCR specificity on the BATCAVE database?**—we conducted the first set of experiments in section 5.1. We computed AUC = 0.544 for TULIP on the BATCAVE database. This is only marginally above the chance level, suggesting weak predictive performance. For comparison, BATMAN reported a within-TCR performance (mean AUC = 0.836) on the BATCAVE database, substantially outperforming TULIP. It should be noted that these results are not directly comparable because BATMAN was trained on this database while TULIP was not, and because our AUC is computed globally, whereas BATMAN reports mean AUC across TCRs. We conclude that TULIP `scores` demonstrate limited predictive capability for TCR specificity on the BATCAVE database and that BATMAN is a preferable model for this task.

Addressing our second research question—**Can TULIP predict how different mutants of the same index peptide activate a given TCR?**—we conducted the second set of experiments in section 5.2. We evaluated TULIP's performance from two perspectives. First, we assessed its classification performance using AUC, obtaining AUC = 0.572 on the subset containing `index peptides` grouped with the same TCRs. This result is promising given that the state-of-the-art models predict only up to $\text{AUC}_{0.1} \approx 0.6$ on unseen data. Second, we assessed whether TULIP `scores` are significantly correlated with `peptide activity` using Pearson correlation. The resulting p-value of ≈ 0.32 considerably exceeds the significance threshold of 0.05, indicating no statistically significant correlation. Therefore, while TULIP shows modest classification performance, the lack of significant correlation limits confidence in its prediction of how different mutants of the same `index peptide` activate a particular TCR.

Our third research question—**Does TULIP achieve sufficient predictive performance to be used for further research?**—is addressed by the combined findings of both sets of experiments. We observed that TULIP showed a measurable improvement over random classification, with performance ranging from marginal (AUC ≈ 0.52) to modest (AUC ≈ 0.57). TCR specificity prediction remains a largely unsolved challenge, where the state-of-the-art models achieve only moderate performance. Nevertheless, we conclude that this level of predictive performance is not sufficient for TULIP to be used for further research.

6.2 Limitations

These results are partially expected, as TULIP was evaluated on a new database containing unseen data, and was not pre-trained or fine-tuned on it. The database was also structured differently than those used in TULIP's original training. Therefore, without pre-training TULIP on unseen peptides, its performance is limited. Due to limited time and computational resources, it was not possible to pre-train or fine-tune the model on this data.

This study is further limited in scope, as TULIP was evaluated on a single database. Additionally, only MHC class I data were used, leaving TULIP's performance on MHC class II TCR-pMHC pairs unexamined, due to the scarcity of MHC class II data in available databases.

6.3 Future work

As the quantity and quality of available databases improve, and more refined models are developed, reliable T cell recognition may become achievable in the near future. The development of more general models capable of addressing multiple tasks within the T cell recognition setting could provide new insights into immunological challenges and enable more targeted treatments for cancer and autoimmune diseases. Based on our findings, current models, including TULIP, are not yet ready for real-world deployment, and future efforts should focus on developing models that achieve reliable performance on unseen data.

Bibliography

[1] Andrew K. Sewell. "Why must T cells be cross-reactive?" In: *Nature Reviews Immunology* 12.9 (Sept. 2012), pp. 669–677. ISSN: 1474-1741. DOI: 10.1038/nri3279. URL: https://www.nature.com/articles/nri3279.

[2] Barthelemy Meynard-Piganeau et al. "TULIP: A transformer-based unsupervised language model for interacting peptides and T cell receptors that generalizes to unseen epitopes". In: *Proceedings of the National Academy of Sciences* 121.24 (June 11, 2024), e2316401121. DOI: 10.1073/pnas.2316401121. URL: https://www.pnas.org/doi/10.1073/pnas.2316401121.

[3] Felix Drost et al. "Benchmarking of T cell receptor-epitope predictors with ePytope-TCR". In: *Cell Genomics* 5.8 (Aug. 13, 2025), p. 100946. ISSN: 2666-979X. DOI: 10.1016/j.xgen.2025.100946. URL: https://www.sciencedirect.com/science/article/pii/S2666979X25002022.

[4] Aisha Shah et al. *Unpaired TCRalpha + TCRbeta sequencing is sufficient for training machine learning TCR-epitope recognition predictors*. Mar. 18, 2026. DOI: 10.64898/2026.03.16.711991. URL: https://www.biorxiv.org/content/10.64898/2026.03.16.711991v1. Pre-published.

[5] Morten Nielsen et al. "Lessons learned from the IMMREP23 TCR-epitope prediction challenge". In: *ImmunoInformatics* 16 (Dec. 1, 2024), p. 100045. ISSN: 2667-1190. DOI: 10.1016/j.immuno.2024.100045. URL: https://www.sciencedirect.com/science/article/pii/S2667119024000156.

[6] Amitava Banerjee et al. "T cell receptor cross-reactivity prediction improved by a comprehensive mutational scan database". In: *Cell Systems* 16.8 (Aug. 20, 2025), p. 101345. ISSN: 2405-4712. DOI: 10.1016/j.cels.2025.101345. URL: https://www.sciencedirect.com/science/article/pii/S2405471225001784.

[7] *meyer-lab-cshl/BATMAN-paper*. Meyer Laboratory, Jan. 2, 2026. URL: https://github.com/meyer-lab-cshl/BATMAN-paper.

[8] Monica Fabbri, Chanel Smart, and Ruggero Pardi. "T lymphocytes". In: *The International Journal of Biochemistry & Cell Biology* 35.7 (July 1, 2003), pp. 1004–1008. ISSN: 1357-2725. DOI: 10.1016/S1357-2725(03)00037-2. URL: https://www.sciencedirect.com/science/article/pii/S1357272503000372.

[9] K. Christopher Garcia and Erin J. Adams. "How the T Cell Receptor Sees Antigen—A Structural View". In: *Cell* 122.3 (Aug. 12, 2005), pp. 333–336. ISSN: 0092-8674, 1097-4172. DOI: 10.1016/j.cell.2005.07.015. PMID: 16096054. URL: https://www.cell.com/cell/abstract/S0092-8674(05)00745-2.

[10] Harlan S. Robins et al. "Overlap and Effective Size of the Human CD8+ T Cell Receptor Repertoire". In: *Science Translational Medicine* 2.47 (Sept. 2010), 47ra64–47ra64. DOI:

10.1126/scitranslmed.3001442. URL: https://www.science.org/doi/full/10.1126/scitranslmed.3001442.

[11] Alok V. Joglekar and Guideng Li. "T cell antigen discovery". In: *Nature Methods* 18.8 (Aug. 2021), pp. 873–880. ISSN: 1548-7105. DOI: 10.1038/s41592-020-0867-z. URL: https://www.nature.com/articles/s41592-020-0867-z.

[12] Ronald N. Germain. "T-cell development and the CD4–CD8 lineage decision". In: *Nature Reviews Immunology* 2.5 (May 2002), pp. 309–322. ISSN: 1474-1741. DOI: 10.1038/nri798. URL: https://www.nature.com/articles/nri798.

[13] Nicole L. La Gruta et al. "Understanding the drivers of MHC restriction of T cell receptors". In: *Nature Reviews Immunology* 18.7 (July 2018), pp. 467–478. ISSN: 1474-1741. DOI: 10.1038/s41577-018-0007-5. URL: https://www.nature.com/articles/s41577-018-0007-5.

[14] Charles A. Janeway et al. *Immunobiology: The Immune System in Health and Disease*. 5th ed. New York: Garland Science, 2001. ISBN: 978-0-443-07098-3.

[15] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[16] Jian Jiang et al. "Transformer technology in molecular science". In: *WIREs Computational Molecular Science* 14.4 (2024), e1725. ISSN: 1759-0884. DOI: 10.1002/wcms.1725. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1725.

[17] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. NAACL-HLT 2019. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423/.

[18] Pauli Virtanen et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python". In: *Nature Methods* 17.3 (Mar. 2020), pp. 261–272. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0686-2. URL: https://www.nature.com/articles/s41592-019-0686-2.

[19] Karl Pearson. "VII. Mathematical contributions to the theory of evolution.—III. Regression, heredity, and panmixia". In: *Philosophical Transactions of the Royal Society of London, Series A: Containing Papers of a Mathematical or Physical Character* 187 (Dec. 31, 1896), pp. 253–318. ISSN: 0264-3952. DOI: 10.1098/rsta.1896.0007. URL: https://doi.org/10.1098/rsta.1896.0007.

[20] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters*. ROC Analysis in Pattern Recognition 27.8 (June 1, 2006), pp. 861–874. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2005.10.010. URL: https://www.sciencedirect.com/science/article/pii/S016786550500303X.

[21] Donna Katzman McClish. "Analyzing a Portion of the ROC Curve". In: *Medical Decision Making* 9.3 (Aug. 1, 1989), pp. 190–195. ISSN: 0272-989X. DOI: 10.1177/0272989X8900900307. URL: https://doi.org/10.1177/0272989X8900900307.

[22] Barthelemy Meynard-Piganeau. *barthelemymp/TULIP-TCR*. Jan. 15, 2026. URL: https://github.com/barthelemymp/TULIP-TCR.

[23] *Frontiers — Contribution of T Cell Receptor Alpha and Beta CDR3, MHC Typing, V and J Genes to Peptide Binding Prediction*. URL: https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2021.664514/full.

[24] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *MACHINE LEARNING IN PYTHON* (). URL: https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?.

[25] Nili Tickotsky et al. “McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences”. In: *Bioinformatics* 33.18 (Sept. 15, 2017), pp. 2924–2929. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx286. URL: https://doi.org/10.1093/bioinformatics/btx286.

[26] Mikhail Shugay et al. “VDJdb: a curated database of T-cell receptor sequences with known antigen specificity”. In: *Nucleic Acids Research* 46.D1 (Jan. 4, 2018), pp. D419–D427. ISSN: 0305-1048. DOI: 10.1093/nar/gkx760. URL: https://doi.org/10.1093/nar/gkx760.

[27] Randi Vita et al. “The Immune Epitope Database (IEDB): 2024 update”. In: *Nucleic Acids Research* 53.D1 (Jan. 6, 2025), pp. D436–D443. ISSN: 1362-4962. DOI: 10.1093/nar/gkae1092. URL: https://doi.org/10.1093/nar/gkae1092.

[28] Eve Richardson et al. *IMMREP25: Unseen Peptides*. Apr. 1, 2026. DOI: 10.64898/2026.03.30.715276. URL: https://www.biorxiv.org/content/10.64898/2026.03.30.715276v1. Pre-published.

[29] John D. Hunter. “Matplotlib: A 2D Graphics Environment”. In: *Computing in Science & Engineering* 9.3 (May 2007), pp. 90–95. ISSN: 1558-366X. DOI: 10.1109/MCSE.2007.55. URL: https://ieeexplore.ieee.org/document/4160265.

Appendix A

Appendix

A.1 Our code

Our modified version of TULIP code and datasets used in this research are available in the following GitHub repository:

`https://github.com/ivankothaj/predicting-TCR-specificity-TULIP`

A.2 Samples from our subsets

index	CDR3b	CDR3a	peptide	MHC	binder	peptide activity
5730	CASSLWEKLAKNIQYF	CAMRGDSSYKLIF	ALWGPDPAAA	HLA-A*02:01	1	0.3125
5731	CASSLWEKLAKNIQYF	CAMRGDSSYKLIF	FLWGPDPAAA	HLA-A*02:01	1	0.115625
5732	CASSLWEKLAKNIQYF	CAMRGDSSYKLIF	ILWGPDPAAA	HLA-A*02:01	1	0.39375
5733	CASSLWEKLAKNIQYF	CAMRGDSSYKLIF	LLWGPDPAAA	HLA-A*02:01	0	0.090625
5734	CASSLWEKLAKNIQYF	CAMRGDSSYKLIF	MLWGPDPAAA	HLA-A*02:01	1	0.1375

Table A.1: Sample of the BATCAVE subset used for overall evaluation of TULIP.

index	CDR3b	CDR3a	peptide	MHC	index peptide	binder	peptide activity
0	CASSLGIDAIYF	CIVRGLNNAGNMLTF	APQDLNTML	HLA-B*81:01	TPQDLNTML	1	0.647810273159145
1	CASSLGIDAIYF	CIVRGLNNAGNMLTF	CPQDLNTML	HLA-B*81:01	TPQDLNTML	1	0.6811609263657957
2	CASSLGIDAIYF	CIVRGLNNAGNMLTF	DPQDLNTML	HLA-B*81:01	TPQDLNTML	1	0.3150460213776722
3	CASSLGIDAIYF	CIVRGLNNAGNMLTF	EPQDLNTML	HLA-B*81:01	TPQDLNTML	1	0.561935866983373
4	CASSLGIDAIYF	CIVRGLNNAGNMLTF	FPQDLNTML	HLA-B*81:01	TPQDLNTML	1	0.6118987529691212

Table A.2: Sample of the subset used to evaluate prediction of TCR activation by different mutant peptides.