Evolving Training Sets for Peptide Discrimination via Evolutionary Algorithms

Loek Gerrits (s1032343), Evangelos Spithas (s1125593), Bart van Nimwegen

June 12, 2025

1 Introduction

1.1 Background

In human bodies, the immune system is responsible for detecting pathogens and exterminating them. One of the tools at its disposal, is the T cells. T cells grow in the thymus, and each one is responsible for identifying and attacking specific cells. However, that means that it is possible for healthy cells to be attacked as well. In order to mitigate this, human cells are presented to the T cells, and the T cells that attack them are eliminated. This process is called Negative Selection (NS). In practice, the amount of possible cells is very large, and as a result only a sub-set of self peptides are presented in the T cells.

Artificial Immune Systems (AIS) are systems that draw inspiration from the human immune system, similarly to how Neural Networks (NNs) are inspired by the human nervous system. A big challenge in using the NS algorithm is selecting an effictive sub-set of peptides to train on. Selecting the optimal subset is important because it direcly impacts the algorithm's ability to differentiate between healthy and harmful peptides. Therefore being able to find the optimal subset to use for the NS algorithm is crucial to improve the accuracy of the NS algorithm.

In this project, we would like to explore how we can find this optimal subsets to train an NS algorithm for peptide selection. We will do this using 3 different methods, randomly sampling a subset, using a greedy algorithm and an evolutionary algorithm to produce them. Afterwards, we will train the negative selection algorithm with each one of them, and evaluate its performance against different sets of harmful peptides, such as HIV and ebola cells.

1.2 Relevance

1.3 Problem description

1.4 Related work

[1]

2 Methodology

2.1 Datasets

We use various The datasets we are using are

2.2 Optimizers

In order to study the effectiveness of evolving a dataset through an evolutionary algorithm, we will compare 1) a random dataset, 2) a greedily optimized dataset,

our evolved datasets (and its resulting performance metrics) to a random dataset

we create a random dataset which will be our control condition.

2.2.1 Greedy Algorithm

The greedy algorithm optimizes ...

Because the greedy algorithm has a high complexity, computing the fully optimized dataset using all human peptides and all possible motifs was infeasible. Therefore, this study limited the number of peptides and motifs.

2.2.2 Evolutionary Algorithm

AA composition

AA frequency

Exchangeability

2.3 Negative Selection

In order to run the negative selection experiments we will use the jar that is provided by `https://johannes-textor.name/negativeselection.html`. The training of the algorithm will take place with the three training sets as described in section 2.1. All peptides have a fixed length of 6, and we will evaluate the effectiveness of the NS algorithm by experimenting with a variable length of contiguous selectors, ranging between 1 and 6.

In order to asses the performance of the NS, we will use as test sets, a subset of the self peptides, with size 1216, and a number of anomalous peptides that belong to the following categories: ebola, hepatitis b... For each peptide in the test datasets, the NS algorithm can return either the number of patterns in the repertoire that match it, or the normalised value $log_2(1 + x)$, where x is the number of matching patterns. We will use this normalised value, as the number of selectors can be unwieldy to work with. Afterwards, we can classify each peptide as anomalous or self peptide, if its score exceeds the value r. After classifying the data, we will estimate the receiver operating characteristic curve (AUC) to quantify how well our estimated datasets can optimise the NS algorithm.

2.4 Experiment

2.5 Analysis

3 Results

4 Discussion

5 Conclusion

The end.

References

[1] Inge MN Wortel et al. “Is T cell negative selection a learning algorithm?” In: *Cells* 9.3 (2020), p. 690.