EpitopePredictions vs MHCnuggets

How they work and how they differ.

# Introduction

We have two programs (MHCnuggets and Epitope Predictions) which both predict how likely a certain epitope is to be shown on the outside of the cell membrane with different haplotypes. This is important in the recognition of one's own body cells. If a cell presents an epitope that is not recognised by the immune system, it will be killed, which is a good thing, as the cell could be malfunctioning or infected by a virus. This presentation of epitopes also plays a key role in vaccine development. In order for a vaccine to be effective you want the epitopes to be presented as often as possible, so the immune system can quickly detect it and develop antibodies against the pathogen. Also, this can be used to make sure that this happens on every different haplotype, so people with different immune systems all get immune quickly. However, these programs output some very different results and thus it is unknown if the given predictions are trustworthy or not. In this paper we discuss the differences between these programs, why they are caused, and how this affects the usefulness of the results.

# 

# Hypothesis

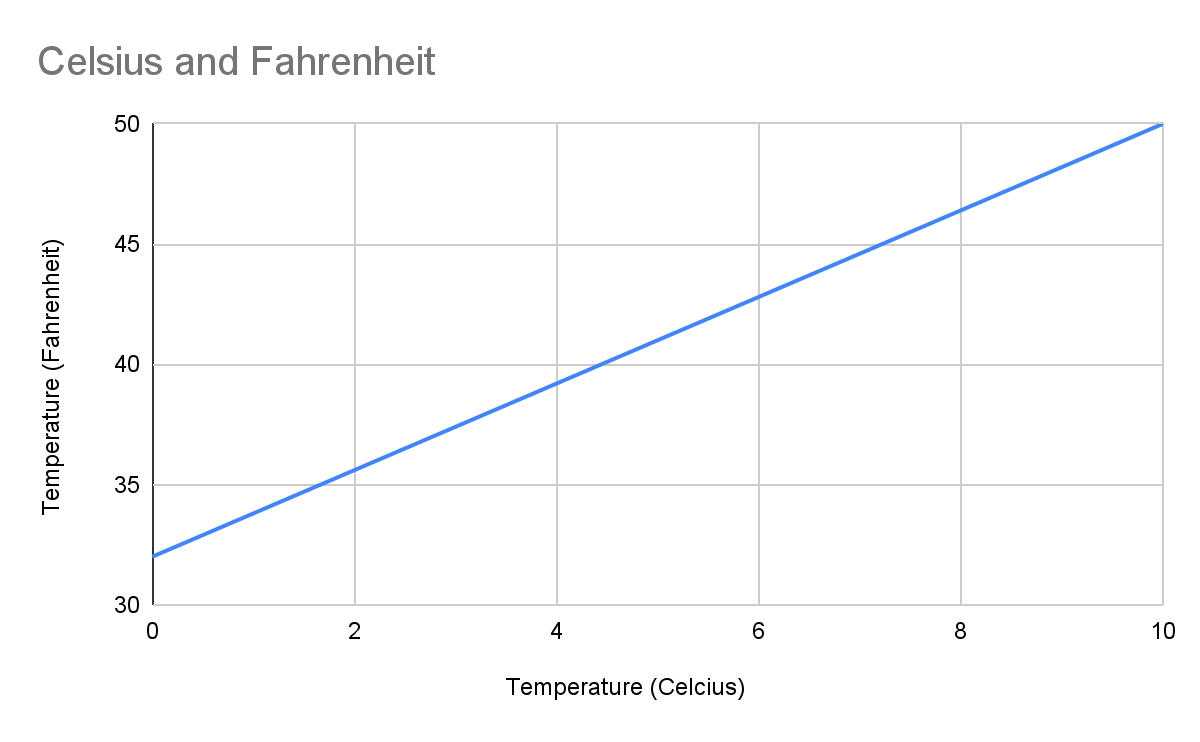
Since both programs are trying to predict the same thing, we expected the results to be equal, or at least very similar. However, as will be shown in the results section of this paper, both programs output different results. We suspected this could be caused by the use of different scales for both programs, just like thermometers can give temperature in celsius or fahrenheit. In this case the numbers differ, but the temperature stays the same. We thought the same might be happening here. If this is happening, we expect the highest results of EpitopePredictions to also be the highest result from MHCnuggets, and vice versa.

# Methods

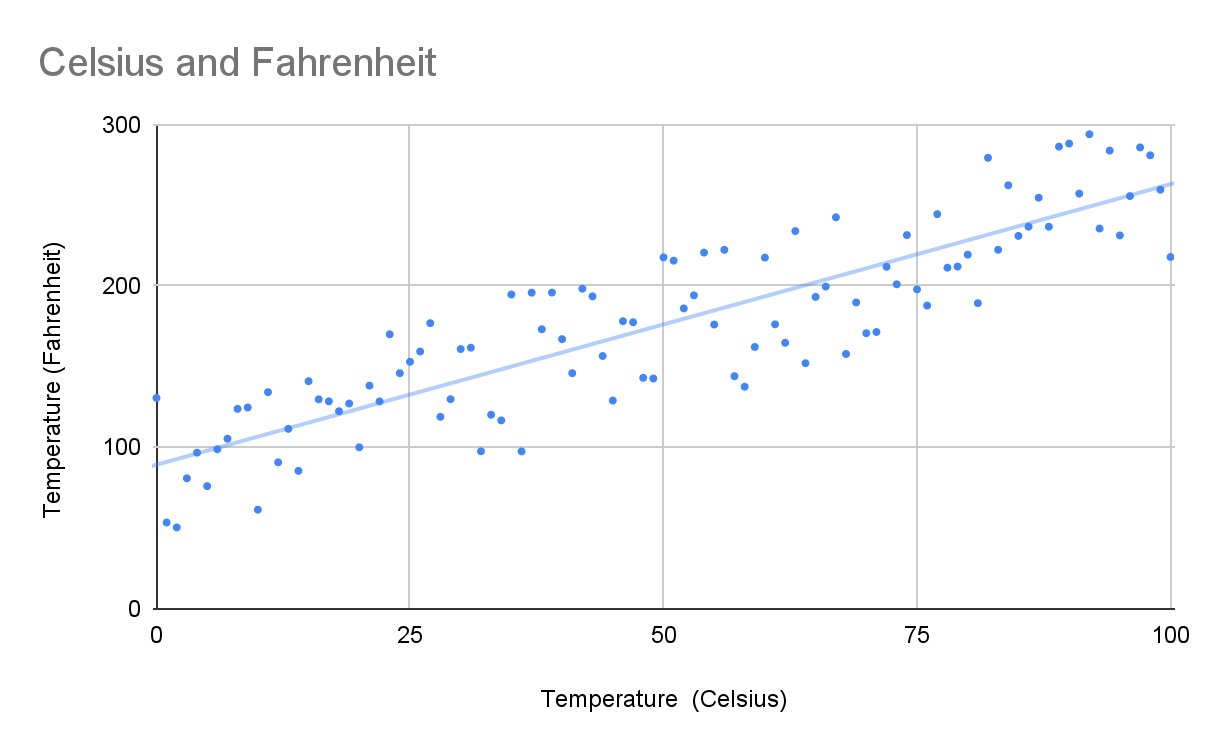
### Correlation

Here we will be showing our methods for determining whether or not there is a correlation between the predicted results from both programs. We demonstrate this using some examples.

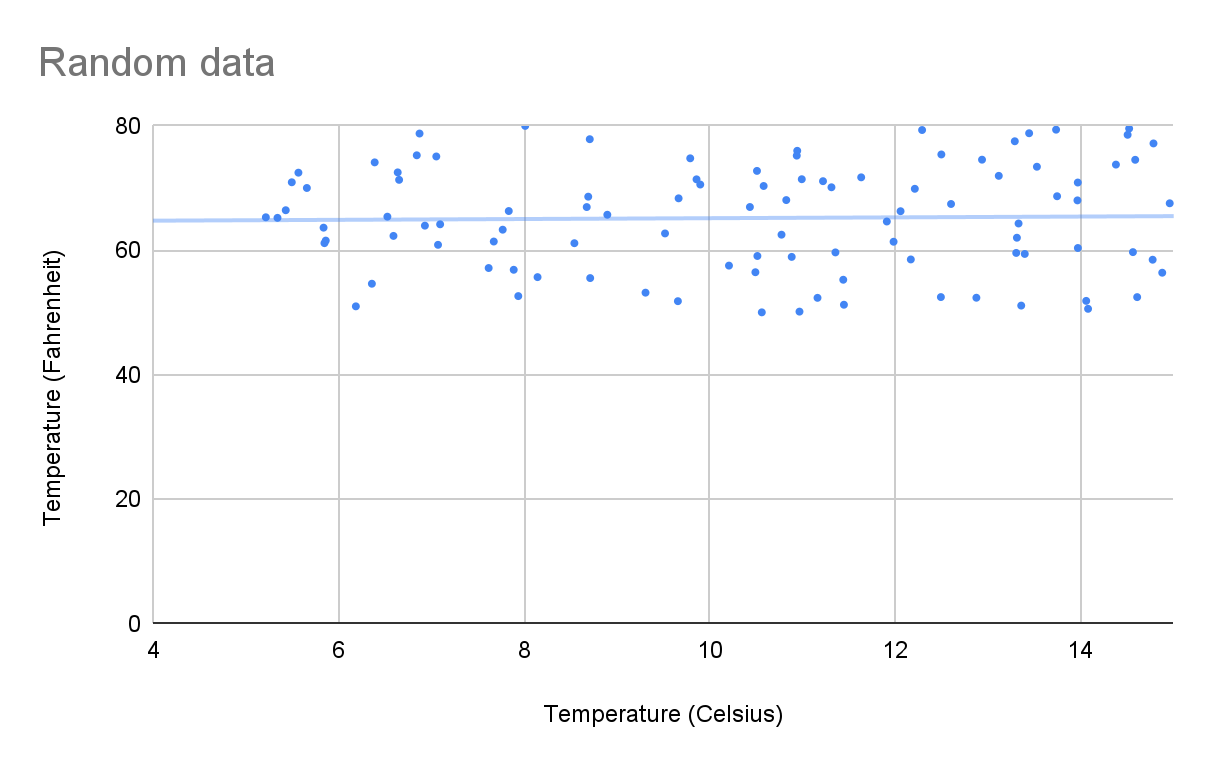
In this table we have the temperature in two different units: Celsius and Fahrenheit. If we were to plot this table, we would get the graph below. The x-axis shows the temperature in Celsius and the y-axis shows the temperature in Fahrenheit. In this example it is easy to see the correlation: as we increase the value of the x-axis, the y-axis follows. While such a correlation is easy to spot, it becomes more difficult when there is noise involved. For this example we expect a trendline with the following formula: Temperature (**°**F) = (Temperature(**°**C) × 9/5) + 32. If we look at the graph we see it starts at 32, which results in the +32. The slope of the graph is equal to , which corresponds to the expected slope.



This example is exactly the same as the previous example, but we added some noise to the measurements. Though, at first glance, it might not seem like there is a clear correlation between the two, a trendline shows that this assumption is false. We still have the same correlation between the temperature in Celsius and the temperature in Fahrenheit. Here, the trendline has an equation of



For our last example we have a graph with random data. Here we see there is no correlation: as we increase the value on the x-axis, the y-axis stays the same, this can be seen using the trendline, which has a equation of .



We also have a way to quantify this correlation, using the Pearson Correlation Coefficient.

### Pearson Correlation

Using the Pearson Correlation Coefficient we are able to find a correlation between the results of MHCnuggets and EpitopePredictions for each different haplotype. See the formula down below.

= correlation coefficient

= values of the x-variable in a sample

= mean of the values of the x-variable

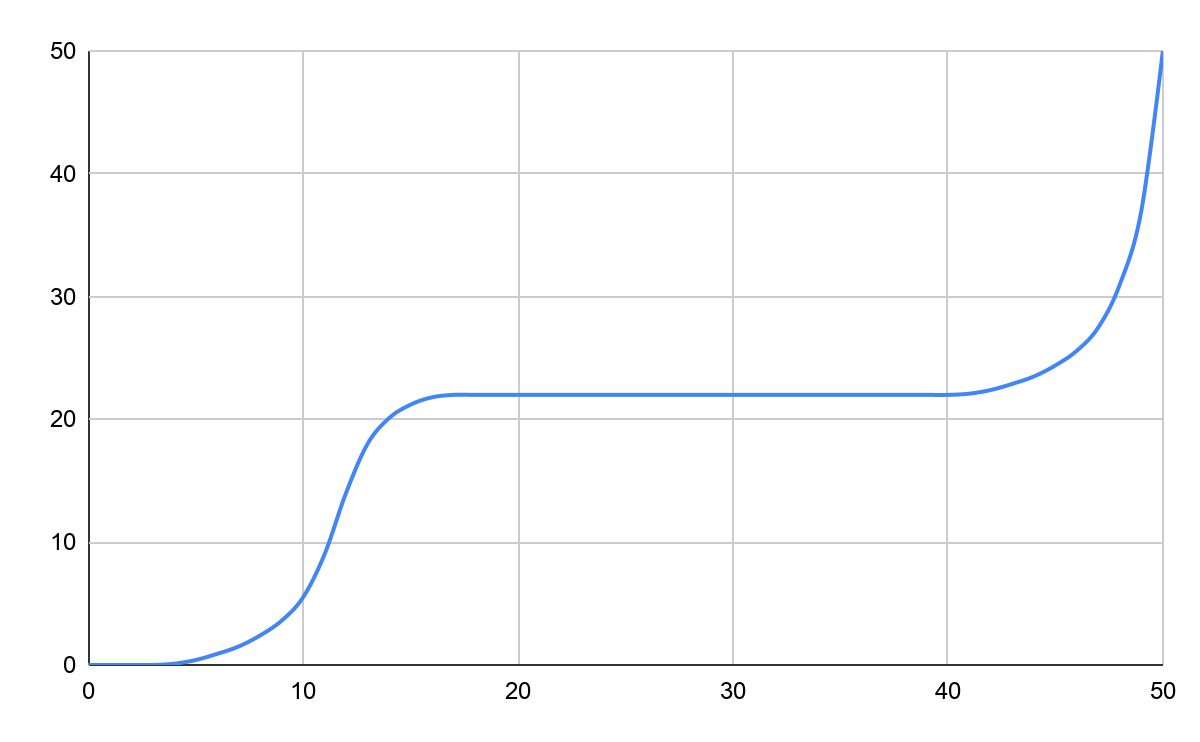
= values of the y-variable in a sample

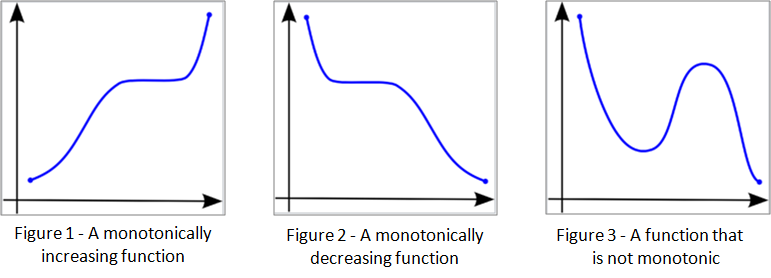
= mean of the values of the y-variable

[Formula explanation]

### Spearman Correlation

While it seems as if this provides solid evidence as to whether there is a correlation or not, the Pearson correlation coefficient has some drawbacks. We use the following graph:

It is clear there is a correlation between the variables, but this correlation is non-linear. If we were to input these values into the Pearson formula, we get a correlation of 0.843, which is still considered quite strong, but the Pearson Correlation Coefficient is not meant to be used on a non-linear graph like this one. To work around this problem, we will also be testing if there is a non-linear correlation. For this we use the Spearman’s Correlation, which can be used to determine correlation between monotonic, or monotone, sets of data. Monotonic meaning they both tend to go up, or down, together.



This is where the Spearman Correlation comes into play. Using the following formula, we are able to determine how strong a monotonic correlation is between the two datasets, if there is any.

*rs* = correlation coefficient

*N* = the number of data points

*Di* =

[Formula explanation]

When we input the same graph in the spearman formula, we get a correlation of 0.9463. Which is very close to one, meaning there is a very strong correlation between the variables in the graph. If this graph were less perfect, and predictable, the Pearson Correlation could indicate there is no correlation, or a very weak one, while there may still be a monotonic relation present. That is why we will also be testing every haplotype to see the spearman correlation, to determine if there is a monotonic relationship.

But this is not where it ends. As there are non-linear graphs that are not monotonic, like a parabola. For this type of correlation we will be plotting the haplotypes as well, to see if there is a different type of correlation visible.

# Biology

Before we dive into the two models and the results it is first necessary to understand what kind of values these models actually predict and what those values mean in the physical world. Therefore we will look into the biological side.

## How are antigens presented to the immune system?

The HLA (human leukocyte antigen) system is a part of the immune system that determines whether a cell in the body is invasive, like a bacteria, and should thus be killed, or a body’s own cell, like a blood cell or a liver cell.

This is usually determined by checking whether the antigen that is presented on a MHC-I (Major Histocompatibility Complex) or a MHC-II molecule. These two molecules have nearly the same name, but differ a fair amount. MHC-I molecules are found on all nucleated cells, cells with a nucleus. However MHC-II molecules are only found on the surfaces of antigen presenting cells (APCs) like macrophages or phagocytes. Thus MHC-I is used for the detection of infected cells by, for example viruses or the detection of bacteria and MHC-II is used for presenting the antigens of a virus or bacteria to other cells in the immune system, for example, T helper cells or B cells.[4]

For viruses this can be determined when a certain cell is infected and is showing a foreign antigen on one of it’s MHC-I proteins. This antigen can be one of the body’s known antigens or, when the cell is infected by a virus, the virus’ foreign antigen. If a cell is presenting an antigen a Tc cell will determine whether the antigen is foreign or not, if this is the case the Tc cell will kill the cell by disintegrating it’s membrane with proteins. It will also start cloning itself rapidly to be able to more quickly notify other cells necessary in the defense against the virus.

For bacteria this process is nearly the same, the only difference is that a bacterium is a cell in itself, while a virus is not. Thus a bacterium is able to present an antigen itself in contrast to an infected cell presenting the antigen “for” the virus. Because the virus is unable to present the antigen as it is not a cell.

## How does ic50 relate to this?

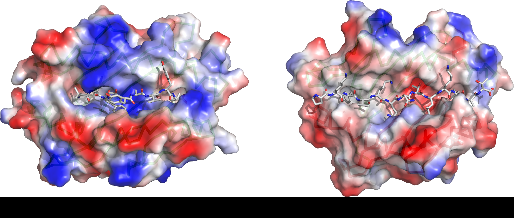
# 

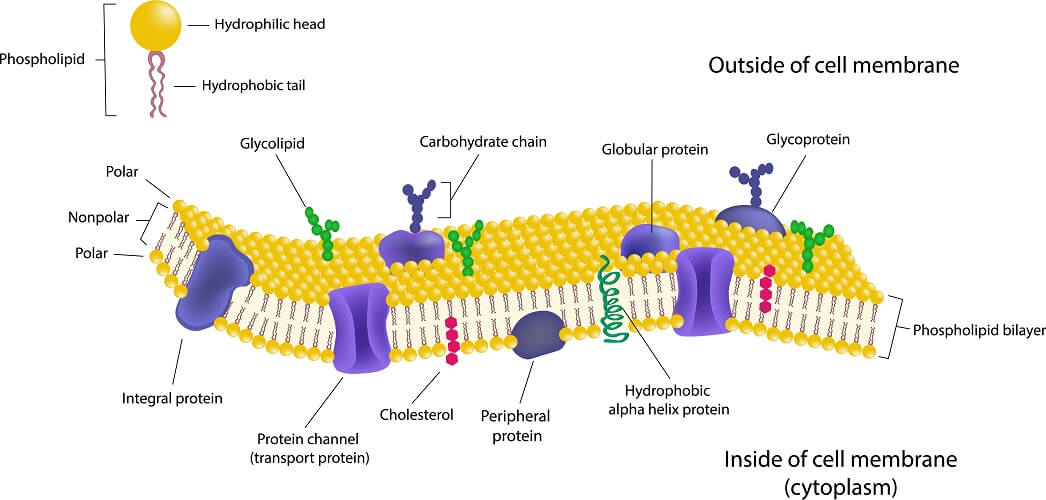
# Epitope Predictions

Epitope Prediction is one of the programs used for the prediction of an IC50-value. It was developed by Johannes Textor. Epitope Predictions uses a stabilized matrix [1] to calculate this IC-50 value.

## Stabilized matrix

A stabilized matrix is [info]

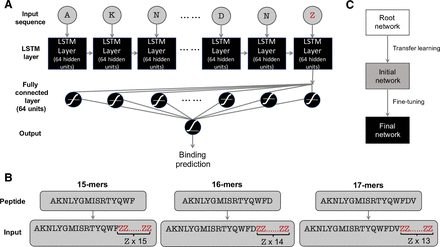
It uses the sum of the hydrophobicity value [2], the physical property of a molecule that is seemingly repelled from a mass of water, of each of the amino acids to calculate how likely they are to appear on the cell membrane. This is where MHC class I is important. The MHC I molecule is built up of α- and β-chains [], which differ between haplotypes. These chains form a groove, in which a peptide can bind. The MHC I groove is closed and because of this mostly short epitopes can properly bind to it, though research has shown longer epitopes do sometimes bind to MHC I []. These epitopes are mostly between 9 and 11 amino acids long. However, epitopes with different lengths often use alternative binding grooves, which complicates predictions []. Because of this Epitope Predictions uses epitopes that are 9 amino acids long, because they are most common [3], and to simplify predictions.



The membrane of a human cell is built up of phospholipids, which consist of a hydrophobic tail and a hydrophilic head. This means that a hydrophobic amino acid is less likely to appear on the outside of a cell membrane. See the image below.

# MHCnuggets

MHCnuggets is another program which predicts an IC50-value. MHCnuggets is developed to to work around the limitations other predictors had, by using a neural network that predicts the peptide-MHC bindings. It uses a method called LSTM(long short-term memory) which is good at processing peptides which can be of any length. The neural network got trained with a method called transfer learning, which in this case means the network uses the data(binding affinity, IC50-value) from other alleles to predict the IC50-value of an unknown (to the program) allele.



visualisation of the neural network

# 

# R-code

The scripts that are used to generate the results can be found [here](https://github.com/richelbilderbeek/ep_vs_mhcn). They roughly work as follows: [the first script](https://github.com/richelbilderbeek/ep_vs_mhcn/blob/master/create_dataset.R) is used to generate random peptides (line 20 to 31) and after that the two programs MHCnuggets and EpitopePrediction are used to predict the ic50 values of the randomly generated peptides (line 33 to 41), these are stored in the file “ep\_vs\_mhcn.csv” (line 43).

# 

# Results

|  |  |  |
| --- | --- | --- |
| Haplotype | Pearson Correlation | Spearman Correlation |
| HLA-A\*01:01 | -0.074 | -0.251 |
| HLA-A\*02:01 | 0.042 | 0.043 |
| HLA-A\*03:01 | 0.055 | 0.08 |
| HLA-A\*24:02 | 0.138 | 0.469 |
| HLA-A\*26:01 | 0.218 | 0.434 |
| HLA-B\*07:02 | 0.309 | 0.734 |
| HLA-B\*08:01 | 0.329 | 0.762 |
| HLA-B\*15:01 | 0.227 | 0.812 |
| HLA-B\*18:01 | 0.119 | 0.221 |
| HLA-B\*27:05 | 0.252 | 0.73 |
| HLA-B\*39:01 | 0.156 | 0.623 |
| HLA-B\*40:02 | 0.171 | 0.556 |
| HLA-B\*58:01 | 0.31 | 0.785 |

Here we have the correlation-values as calculated using both Pearson’s and Spearman’s formulas. The Pearson Correlation gave a value between -0.074 and 0.309 and the Spearman Correlation ranged from -0.251 to 0.812.

# [RJCB: I think the plot looks pretty. I do not understand that the red lines do not go through the red lines. You table has two columns, on for the PC and one for the SC, so where do those other values come from?] [JB: dit is een trendlijn van het zwevend gemiddelde van de afgelopen 2 waardes, dat is ook waarom de halftransparante lijnen pas bij de 2e waarde beginnen.]

# Conclusion

As can be seen in the results (figure x), there appears to be a slight correlation according to Pearson’s coefficient (an average r-value of 0,1732). But this is not high enough to be considered statistically significant, for that to be true the value has to be above 0,3, which is true for some haplotypes though. *Indicates a slightly linear graph?*

Interestingly, the average correlation calculated by Spearman’s coefficient is much higher, namely 0,4614. This can be considered as statistically significant. *Indicates a fairly monotonic graph, thus the next point in this case usually has a higher value than the one before that. Not necessarily linear? And what does that mean exactly in this situation?*

It can also be seen that the (trend)lines are very comparable. The differences in the y-values are often comparable, but the Pearson graph almost constantly has lower values, *this might mean that the graph is partially linear and partially monotonic, more research has to be done to verify this?*

# 

# Discussion

# Definitions

Epitope: the part of an antigen molecule to which an antibody attaches itself. An epitope is a specific protein on the surface of an antigen and is used for recognition of cells.

IC50: a quantitative measure that indicates how much of a particular inhibitory substance (e.g. drug) is needed to inhibit, in vitro, a given biological process or biological component by 50%. The biological component could be an enzyme, cell, cell receptor or microorganism. For our research this means a lower IC50 corresponds to an epitope beine more likely to be presented.

Haplotype: a group of genes **(jeans)** within an organism that was inherited together from a single parent. A different haplotype can correspond to a different MHC I-complex, which in itself means that different epitopes are more or less likely to be presented to the immune system.

# 

# References

[1] Peters, Bjoern, and Alessandro Sette. "Generating quantitative models describing the sequence specificity of biological processes with the stabilized matrix method." *BMC bioinformatics* 6.1 (2005): 1-9.

[2] Sanchez-Trincado, Jose L., Marta Gomez-Perosanz, and Pedro A. Reche. "Fundamentals and methods for T-and B-cell epitope prediction." *Journal of immunology research* 2017 (2017).

[3] Trolle, Thomas, et al. "The length distribution of class I–restricted T cell epitopes is determined by both peptide supply and MHC allele–specific binding preference." *The Journal of Immunology* 196.4 (2016): 1480-1487.

[4] Janeway, Charles A, and Jr. “The major histocompatibility complex and its functions.” *Immunobiology: The Immune System in Health and Disease. 5th edition.*, U.S. National Library of Medicine (1970).

[5] Research done by Joshua van Waardenberg

[richelbilderbeek/meesterproef\_joshua](https://github.com/richelbilderbeek/meesterproef_joshua)

[6] Initial research done by Richel Bilderbeek

[richelbilderbeek/ep\_vs\_mhcn](https://github.com/richelbilderbeek/ep_vs_mhcn)

[7] Spearman’s vs Pearson’s correlation coefficient

<https://statisticsbyjim.com/basics/spearmans-correlation/>

[8] Various calculators for correlation coefficients

<https://www.socscistatistics.com/>

[]MHC class I

<https://en.wikipedia.org/wiki/MHC_class_I>

[] hydrophobicity

<https://en.wikipedia.org/wiki/Hydrophobe>

[] Epitope Predictions

<https://github.com/jtextor/epitope-prediction>

[] Stabilized matrix

<https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-6-132>

[] Interpretation of Pearson’s coefficient

<https://www.questionpro.com/blog/pearson-correlation-coefficient/>

### 

### Images:

[] Cell membrane

<https://biologydictionary.net/cell-membrane/>

Visualisation of the neural network

<https://cancerimmunolres.aacrjournals.org/content/8/3/396.long>

MHCnuggets

<https://cancerimmunolres.aacrjournals.org/content/8/3/396.long>

LSTM

<https://www.researchgate.net/publication/13853244_Long_Short-term_Memory>

# 

# Notes

Epitope Predictions gebruikt een stabilized matrix, waarbij de hydrofobiciteit van de aminozuren wordt opgeteld. Epitope Predictions is sneller dan MHCnuggets, maar misschien minder betrouwbaar. JTextor

MHCnuggets gebruikt een getraind neuraal netwerk en is langzamer, maar misschien nauwkeuriger.