Personalized cancer vaccine design: A Literature Review

ChiaYu Lin

Universiteit van Amsterdam (13692577)
Vrije Universiteit Amsterdam (2729198)
c.y2.lin@student.vu.nl

Abstract

Cancer immunotherapy has been considered an important treatment for patients with end-stage cancer in recent years. In cancer immunotherapy, personalized vaccines can be produced by using tumor proteins with a high binding affinity to the MHC-I protein. This paper first explains why personalized vaccines are necessary for cancer immunotherapy, then focuses on five methodologies and their machine-learning models for predicting protein binding affinity, namely NetMHCpan-4.0, MHCflurry, MHCflurry 2.0, DeepRank, and DeepRank-GNN, benefits and drawbacks of each method. Finally, the paper concludes with a discussion of potential advancements in MHC binding affinity prediction methods.

1 Introduction

Over the years, the medical community has devoted itself to researching ways to treat cancer in patients with advanced-stage cancer. Common cancer treatments include surgery, chemotherapy, radiation therapy, and immunotherapy. Although there are several treatment options for cancer, each of them often has limitations, such as surgical removal of the cancer site, which is only effective for early-stage cancer. Radiation therapy is associated with damage to local peripheral tissues, while chemotherapy causes severe side effects as a result of the drugs administered. Among them, cancer immunotherapy is a relatively new treatment but has already established itself as a majorstay of mainstream cancer[16].

The principle of immunotherapy is to treat cancer by utilizing the patient's own immune system. The treatment is accomplished by selecting targeted tumor mutation[16] and use these mutated tumor peptides to activate the immune system's ability to combat the disease[15] and thereby eliminating the tumor. Immunotherapy has shown excellent results in the treatment of a variety of malignancies and is more efficient, better tolerated, and reduces adverse reactions compared to other conventional treatments[25].

Personalized cancer vaccines are formulated on the basis of selective mutated tumor peptides. However, a patient's cancer can contain thousands of mutations[28]. Therefore, selecting the appropriate tumor mutation peptides

to utilize as vaccine candidates represents the major challenge for cancer immunotherapy. Tumor peptides must have high binding affinities to Major Histocompatibility Complex(MHC) proteins in order to be effective vaccine candidates. Software tools such as NetMHCpan-4.0[13], MHCflurry[4], MHCflurry2.0[4], DeepRank[1], and DeepRank-GNN[2], which used state-of-the-art machine learning (ML) techniques have been demonstrated to identify tumor peptides with high MHC protein binding affinities. To date, researchers are still developing new ML methods to improve the accuracy of binding affinity predictions.

This paper is divided into four sections. First, the ***Introduction*** section is provided to explain the background of cancer immunotherapy and the necessity of using ML techniques to formulate personalized cancer vaccines. The ***Methodology*** section is followed by defining the research questions of this paper and each motivation. In the ***Discussion of Literature*** section, the content is divided into four subsections to discuss the four different software tools based on machine learning techniques that predict binding affinities. Each research question will also be answered respectively. Finally, a ***Conclusion*** section is listed to summarize the paper.

2 Methodology

In this section, the methodology that I used will be described. First, is the selection of literature papers, since the topic of the paper is related to the design of personalized cancer vaccines using ML techniques. Since the immunotherapy method using MHC binding affinity for personalized vaccines was only introduced around 2017, the predictive models introduced by the industry are still limited. From the suggestion of the e-Science Center of Amsterdam, which is an expert in personalized cancer vaccines, I selected five of the most representative software tools of the modern era that were most mentioned in the papers for comparison as my literature research. Two of the tools, DeepRank[1] and DeepRank-GNN[2], were developed by the e-Science Center Amsterdam, where I will be doing my internship, so it was very helpful to get a quick overview of the ML mechanism used for cancer vaccine design. Google Scholar is used to searching for publications that provide background information on the biology related to the formulation of the personalized cancer vaccine. In this paper, I will focus on machine learning models the five software tools designed to predict good vaccine candidates instead of discussing how to design cancer vaccines in practice. Therefore, the research questions and the motivation is as followings:

- RQ1: *What are the future directions of machine learning techniques for personalized cancer vaccine design?*
  - The goal of this research question is to investigate how the accuracy and performance of the future machine learning model for predicting the binding affinity of MHC complexes to formulate personalized cancer vaccines can be improved and implemented. To answer this question, I need to delve into the machine-learning models currently being used in

the field for the design of personalized cancer vaccines to gain a better understanding of the issue they are dealing with. As a result, I have come up with the following two subquestions.

- RQ1.1: *What are the requirements for a suitable vaccine candidate?*
  - The goal of this research question is to understand what the requirements are for a mutated tumor peptide to become a good vaccine candidate. This answer will be relevant to how machine learning models are trained and what are the expected test outputs from the model.
- RQ1.2: *What are the major software tools for predicting vaccine candidates? What are the key challenges for these software tools in predicting vaccine candidates?*
  - The goal of this research question is to discuss the five main software tools NetMHCpan-4.0, McFlurry, McFlurry2.0, DeepRank and DeepRank-GNN and to find the challenges these software tools are facing by comparing their advantages and drawbacks.

3 Discussion of Literature

In this section, I will review the five current state-of-art tools for binding affinity prediction in the scope of personalized vaccination. In Section 3.1, a brief summary of the articles will be described based on the time of publication of the tools. In Section 3.2, a detailed description of the machine learning models of the tools will be provided, including the architecture of the model and how the model was trained and tested. In Section 3.3, I will discuss the limitation each tool faces and how it can be improved.

3.1 Literature Summary

Jurtz et al. [13] proposed a method **NetMHCpan-4.0**, which makes predictions of the binding of cellular peptides and MHC-I peptides based on their *binding affinity(BA)* and *eluted ligand(EL) information*. The greatest contribution of NetMHCpan-4.0 is that it improved the traditional use of only cellular BA data to predict the binding of MHC to peptides. The problem with using only BA data to predict the MHC binding of peptides leads to a correspondingly high false positive rate which in turn leads to the unsuitability of the personalized vaccine. The IEDB(Immune Epitope Database)[7] has a large amount of publicly available BA and EL data. Therefore, NetMHCpan-4.0 used the data provided by IEDB to train the machine learning model, with the combination of two data domains, BA and EL, to make the prediction of MHC binding more reliable. Jurtz's team tested NetMHCpan-4.0 by comparing its performance using different data domains. While the regular NetMHCpan-4.0 method is trained using BA + EL data, the model was also trained using BA data and EL data only. The result showed that the Area Under Curve (AUC) score of the BA+EL model received the highest score around 0.95, followed by the model trained with just

BA data, which is slightly lower than the performance of BA+EL, and finally the model trained with only EL data, whose AUC score is only close to 0.85. As a result, by merging the BA and EL data-set, the method NetMHCpan-4.0 may effectively increase the prediction accuracy. In addition, the following brief introductions to binding affinity and eluted ligands are provided:

Binding Affinity
In the biomedical field, the binding of two proteins can be seen as a reversible, rapid equilibrium process. Binding Affinity is a value that describes the strength of the binding interaction between two proteins[14] and is often seen as an indicator of the binding between the tumor peptides to MHC-I proteins in cancer immunotherapy.

Eluted Ligand
The eluted ligand is a naturally occurring peptide that can be eluted from antigen presenting cells (APCs)[22]. The information of the elution ligand is independent of the MHC cells and is not present on the surface of the cells. It provides beneficial biomedical data that can be used to predict appropriate peptides.

O'Donnell et.al [20] proposed an open-source software tool **MHCflurry** to predict the binding of MHC-I peptides. Before the development of MHCflurry, the training process of MHC-I binding tools training process, such as NetMHCpan-4.0, could only be operated by developers and its use was primarily restricted to private research purposes. In response to the surge in the discovery of tumor neoantigens, O'Donnell's team has released a Python-based package, MHCflurry, which is open-sourced and simple-to-install[4]. MHC-flurry features a configurable interface, and its machine-learning model can be modified and re-trained according to the user's needs. Similarly to NetMHCpan-4.0, MHCflurry also trained the model based on binding affinity data and MHC ligands. However, MHCflurry supports more variable peptide lengths between 8 and 15 using a fixed-length encoding algorithm. The first four and last four alleles of the peptide are regarded as the "anchor location" that is most related to MHC. Following that, the inputted peptides are represented as a 15-length sequence with the anchor position alleles fixed and the residue alleles filled in with an X character[20]. To measure the effectiveness of MHCflurry, O'Donnell's team applied a mass spectrometry (MS) data set that comprises a substantial amount of MHC data[18]. According to the outcome, MHCflurry can identify up to 7000 predictions per second, which is approximately 400 times as fast as NetMHCpan 4.0. Furthermore, MHCflurry also presents a better prediction accuracy for peptides of lengths other than 9[20].

Two years later, O'Donnell et al. [21] published an improved version of MHCflurry, known as **MHCflurry 2.0**. O'Donnell's team developed two separate predictors to increase the accuracy of prediction for MHC-I peptides. The BA predictor focuses on predicting the binding affinity of class 1 MHC, just as

the common MHC-I predictor. Although the selection of MHC-I proteins with higher affinity is the most basic criterion for formulating a personalized cancer vaccine, O'Donnell's team decided to include other presentation pathways in the antigen delivery process as a requirement as well. Therefore, the second predictor, also referred as the antigen process (AP) predictor, is created based on allele-independent effects [21]. O'Donnell's team first trained the BA predictor utilizing cellular affinity information and MS data sets. The AP predictor model was then constructed using a training set produced by the BA predictor. The training set contains identified peptides(hit) and unidentified peptides(decoys), where the hits and decoys are both recognized by the BA predictor[21]. A small data collection of allele-independent data that were not used in the BA predictor was then applied to model the AP predictor. As a result, MHCflurry 2.0 surpasses NetMHCpan4.0 and its previous version MHCflurry with a 40% improvement in positive prediction value(PPV) due to the integration of the BA and AP predictor[21].

Renaud et al. [23]proposed a deep learning Python package **DeepRank**[1] that uses 3D convolutional neural networks (CNNs) to analyze protein-protein interactions (PPIs). Through the PPIs, one can observe the whole process of cellular interaction. This is crucial for predicting how proteins will function in cancer immunotherapy. Renaud's team collected more than 7000 pieces of 3D structure data sets of protein-protein complexes from the Protein Data Bank(PDB) database[5], which is a valuable resource database for structural biology research. Renaud's team further processed the PDB files to intercept and analyze only the interface residue between the two protein chains. The atoms that are covered by the two protein chains at an agreed-upon distance are referred to as the interface residue. The complexes of a residue interface are then mapped using a Gaussian mapping technique onto a 3D grid. The mapped 3D grid data is then displayed as 3D images, with each grid point comprising values related to various interface properties. The 3D images of PPI are stored in a HDF5 file format which is used for the training and testing of Deep Rank's CNN model[23]. The DeepRank platform has two functions, classification, and prediction. For classifications, Deep Rank can determine whether the inputted PPI 3D image is a biological or a crystal artifact. For prediction, DeepRank can predict the binding affinities of the two proteins based on the PPI 3D images, which is the primary feature related to immunotherapy. DeepRank also demonstrates its high processing efficiency by using MPI to perform parallel processing on a large number of PDB files, thus outperforming the two existing methods, PRODIGY and PISA.

The following year, Réau et al.[24] published a new version of the DeepRank Python package[2], known as **DeepRank-GNN**. DeepRank-GNN, as its names indicate, chose GNN to create the model as opposed to DeepRank, which employs CNN as its deep learning model. The reason for abandoning CNN in favor of GNN is that CNN has some drawbacks, such as that it is not sufficient to extract the overall structural information because it uses data from the Euclidean space to represent images[17]. Additionally, the PPIs data re-

quires argumentation before it is fed into the model, which will affect the overall performance. Similarly to DeepRank, DeepRank-GNN focuses on analyzing the interface residue of the PPIs. In contrast, the PPIs for DeepRankGNN are transformed into graphs rather than images, with the interface residue's atoms serving as the graph nodes and the two protein chains' contact residues serving as the graph edges. The PPI graphs are then stored in the HDF5 file format. DeepRank-GNN was evaluated by Réau's team using DOVE and HADDOCK scores. As a result, DeepRank-GNN demonstrated a notable improvement in the performance of PPI data processing, and DeepRank-GNN also required significantly less storage than DeepRank.

3.2 Model Description

NetMHCpan-4.0
NetMHCpan[13] built its model on NNAlign[19], a platform to construct neural network models and evaluate the performance of peptide-MHC interactions. A design schematic of the neural network model of NetMHCpan-4.0 is shown in Figure 1. The BA and EL data used to train the model are from IEDB[7], with a total of 85217 data sets inputted. The neurons in the hidden layer were trained using 10 randomly generated configurations and 5-fold cross validation, yielding a combination of 100 networks.The output layer contains two output neurons: the first neuron returns the score for binding affinity, while the second neuron outputs the score of eluted ligand.

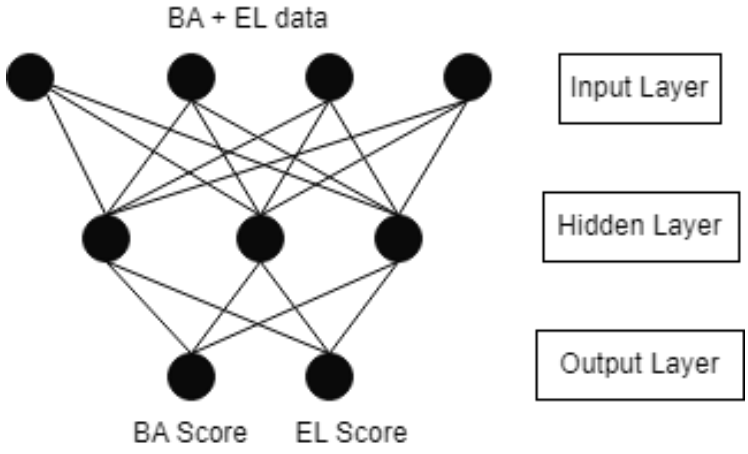

Fig. 1: The Neural Network Model of NetMHCpan-4.0 [13]

MHCflurry
The open source Keras neural network package[3] is applied to implement MHCflurry

in Python. MHCflurry encodes each residue of the peptide and converts it into a sequence of length 15 and assembles a 15x21 matrix as input to the MHCflurry model. The binding affinity of each allele was calculated as the geometric mean of the results from the set of 8 to 16 neural networks in each of the 320 MHCflurry models. The geometric mean results are further converted to a scale of 0 to 1, with larger values representing the stronger binding of the allele to the MHC-I peptide.

The binding affinity and MHC ligands datasets used for model training come from the IEDB[7]. Before training, the data sets are filtered and those without MHC-I and nonidentifiable names are discarded, leaving only those peptides identified by mass spectrum, for a total of 143,898 quantitative and 439,978 qualitative binding affinity data[20]. MHCflurry is derived from 40 frameworks that are trained on allele-specific affinity data or trained on only quantitative affinity data. Four replications were trained for each of these 80 potential outcomes, resulting in 320 models for each allele[20]. 90% of the data is used to train each neural network, and the remaining 10% is used as test data to generate the MHCflurry neural network model[20].

MHCflurry 2.0

As an improvement version of MHCflurry, MHCflurry 2.0 is also implemented based on the neural network library, Keras[3]. As mentioned in the previous section, MHCflurry is made up of two predictors, the BA predictor and the AP predictor, and thus, the two predictor model is built separately. For the BA predictor, MHCflurry 2.0 expands on its previous version, from requiring a separate neural network model to train each allele, to being able to train up to 14993 MHC-I alleles on a single neural network model[21]. The input to the neural network model contains the encoded peptide converted sequence and the amino acid sequence of the selected 37 positions in the allele. Similarly to MHCflurry, the model outputs the binding affinity results and scales them from 0 to 1. As mentioned in the previous section, the AP predictor is trained based on the hits and decoys from the BA predictor. In MHCflurry 2.0, O'Donnell's team defined a binding affinity less than 0.5744 as a "hit". MHCflurry 2.0 features 140 training models in total, whereas MHCflurry 2.0 is made up of 35 architectures and uses four duplicated data sets[21]. In addition, in each neural network, 90% of the data is used for training and the remaining 10% is used as test sets.

DeepRank

DeepRank uses the widely used Pytorch[6] deep learning framework to implement its neural network model[23]. The training procedure begins with the model receiving HDF5 files containing multiple 3D PPI images with features and labels. Users can simply filter the PPI images from the HDF5 files that they intend to input into the model based on the features and labels of the PPIs[23]. Those PPI images that match the values set by the user are fed into the neural network model and trained through a series of CNN layers, such as convolutional layers, pooling layers, and finally fully connected layers. The data sets trained for the

model comprised a total of 5739 PPI complexes. The data set is divided into two parts: training and validation, while the training portion comprises 80% of the data set[23]. The outcome of the model, the binding affinity values, are stored in an HDF5 file for further use. The overview of the DeepRank neural network model is shown in Figure 2.

DeepRank-GNN
DeepRank-GNN uses the Graph Interaction Network (GINet) to form the neural network model[24]. The GINet architecture is a new GNN architecture that can strengthen convolution network feature representations[27]. DeepRank-GNN imports HDF5 files which includes multiple PPI graphs to the model for training. In the GINet architecture, each PPI graph is further divided into two subgraphs, one of which is an internal graph connecting atoms within the same protein and the other is an external graph connecting atoms between two proteins[24]. The internal and external subgraphs are successively transferred to a series of GNN layers, such as the graph convolution layer, pooling layer, and the fully connected layer. Lastly, a scatter mean procedure was employed to combine the two subgraphs before constructing the final representative graph. The output graphs of the DeepRank-GNN model are also saved in the HDF5 file format for further binding affinity analysis[24].

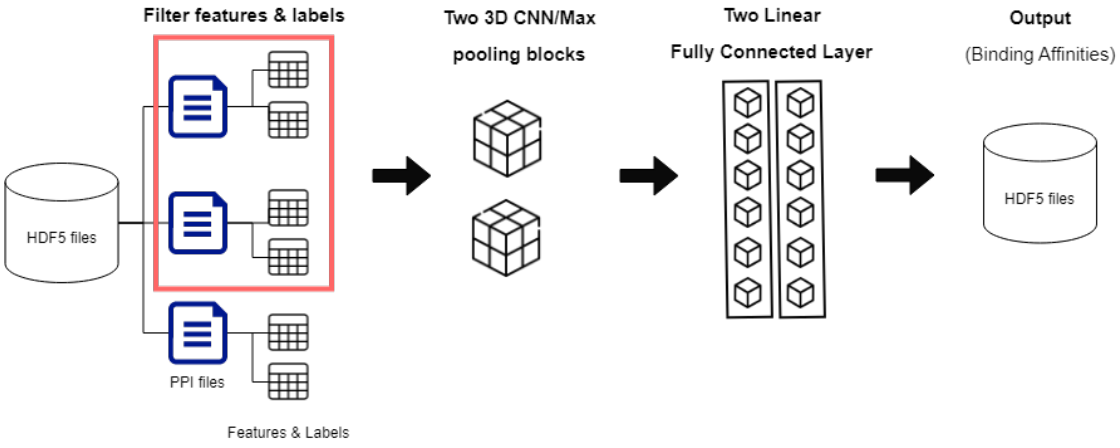

Fig. 2: The Neural Network Model of DeepRank [23]

3.3 Challenges and Future Directions

NetMHCpan-4.0 was the first method that predict MHC-I interaction based on information both on the binding affinities and eluted ligands. However, NetMHCpan-4.0's execution speed is its major flaw. MHCflurry was introduced to solve the

problem of NetMHCpan-4.0 execution efficiency with 396 times faster execution speed. In addition, MHCflurry features fixed-length encoding that allows proteins of different lengths to be input while preserving the anchor positions, which is most relevant to MHC-I. MHCflurry does, however, have the limitation of only supporting a specific set of alleles. Its improved version, MHCflurry 2.0 expands the number of supporting alleles from 112 to 14993 types. Furthermore, MHCflurry 2.0 enhances its MHC-I prediction ability by implementing two separate predictors, the BA and AP predictors. A significant flaw with MHCflurry 2.0 is that the data used for training the AP predictor is already detected by the BA predictor, which may cause the accuracy of the AP predictor to be overstated. DeepRank, a platform that uses the CNN deep learning technique, was published. DeepRank converts PPIs into images for model training; the model itself has the benefit of flexibility because the user can select the PPIs depending on features and labels. However, DeepRank has several major drawbacks such as the 3D grid size used for the model is fixed, so large PPI may not fit into the model, and CNN may require more data argumentation. DeepRank-GNN addresses these issues by implementing the model using GNN, which presents PPIs via graphs as opposed to images since the neural network model can manage any size of graph. However, the performance of DeepRank-GNN still has large space for improvement. Finally, existing models suffer from the same problem, that is, the number of alleles that can be supported is still insufficient, and the variable length of proteins that can be processed is similarly constrained.

3D modeling approaches may be an ideal way to predict vaccine candidates, which is not limited by the protein variability length and can also identify more rare MHC allels through 3D structures. Geometric deep learning is a newly developed technology used in machine learning techniques for 3D modeling and is optimized for molecular sciences[11]. The feature of geometric deep learning is that its neural models are generated from non-Euclidean domains, for example, graphs[12]. This technique allows the representation of cells to be rendered graphically from 3D models instead of converting them into 2D grids for model training. The atoms in the cell represent the nodes in the graph, and the associations between the atoms represent the edges of the graph[11]. Therefore, future models may apply geometric deep-learning techniques to address problems in existing models.

4 Conclusion

In this literature study, I studied and analyzed five methodologies, namely NetMHCpan-4.0, MHCflurry, MHCflurry 2.0, DeepRank, and DeepRank-GNN to investigate how to formulate a suitable personalized cancer vaccine with machine learning technology. Based on my research questions, three conclusions can be drawn from the study:

1. Mutated tumor peptides must bind MHC proteins with high affinities to become suitable vaccine candidates.

2. Although existing methods have been improving machine learning models to predict protein binding affinity with increasing accuracy and speed, there is still a shortage of supported alleles and protein length variability, which limits the accuracy of prediction.
3. Future methodology for determining vaccine candidates could focus on training mutated tumor peptide 3D modeling using the geometric deep learning technique.

Some nice algorithms used in the existing predicting tools can also be followed in future methodology. For example, the fixed-length encoding algorithm used in MHCflurry enables the ability to support variable length alleles, the idea of interface residue between the two protein chains from DeepRank also improves the accuracy by focusing on the most relevant part of the cells. Lastly, the protein-protein interactions (PPIs) used in DeepRank and DeepRank-GNN support generating the overview of cellular interaction, which is useful to represent input data for future models based on GDL.

The limitation of this literature study is that only the five most-mentioned binding affinity prediction methods in the field and their machine-learning models were introduced. Other minority approaches such as MixMHCpred and DeepMHC also provide good MHC prediction and may also be included in future references. In addition, limited research and discussion were conducted on the pre-processing part of the data before applying them to model training. Moreover, the performance analysis tools used in the model were not thoroughly covered. Future research can take into account all of these restrictions.

A Terminology Table

Term	**Definition**
Allele	An allele is a gene variant that resides in the same location[8].
Antigen Process	A immune process that expresses the trigger of immune cells.
AUC	The Area Under Curve (AUC) is a performance analysis method for measuring how much the model is capable of distinguishing between classes.
Binding Affinities	The strength of binding interactions between two molecules.
Cancer Immunotherapy	Cancer immunotherapy is a type of therapy that treats cancer by activating self-immune system.
Mass Spectrometry	An analytical approach for evaluating the mass-to-charge ratio (m/z) of one or more molecules within a sample[10].
MHC Protein	The major histocompatibility complex (MHC) is a group of closely related genes on vertebrate DNA that code for cell surface proteins required by the immune system[9].
PPI	Protein-protein Interactions(PPI) is the area of a protein's surface where it interacts with another[26].

References

1. DeepRank. https://github.com/DeepRank/deeprank
2. DeepRank-GNN. https://github.com/DeepRank/Deeprank-GNN
3. Keras. https://github.com/keras-team/keras
4. MHCFlurry. https://github.com/openvax/mhcflurry
5. Protein Data Bank. https://www.rcsb.org/
6. Pytorch. https://pytorch.org/
7. The Immune Epitope Database. https://www.iedb.org/
8. allele. https://www.nature.com/scitable/definition/allele-48/ (2014)
9. major histocompatibility complex. https://www.britannica.com/science/major-histocompatibility-complex (2023)
10. What is mass spectrometry? https://www.broadinstitute.org/technology-areas/what-mass-spectrometry (2023)
11. Atz, K., G.F..S.G.: Geometric deep learning on molecular representations. Nat Mach Intell **3**, 1023–1032 (2021). https://doi.org/https://doi.org/10.1038/s42256-021-00418-8
12. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: Going beyond euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017). https://doi.org/10.1109/MSP.2017.2693418
13. Jurtz V, Paul S, A.M.M.P.P.B.N.M.: Netmhcpan-4.0: Improved peptide-mhc class i interaction predictions integrating eluted ligand and peptide binding affinity data. Journal of immunology **199**(9), 3360–3368 (2017). https://doi.org/https://doi.org/10.4049/jimmunol.1700893
14. Kastritis, P., Bonvin, A.: On the binding affinity of macromolecular interactions: Daring to ask why proteins interact. Journal of the Royal Society, Interface / the Royal Society **10**, 20120835 (02 2013). https://doi.org/10.1098/rsif.2012.0835
15. Kirschner, D., P.J.: Modeling immunotherapy of the tumor – immune interaction. Journal of Mathematical Biology **37**, 235–252 (1998)
16. Koury, J., L.M.C.C.C.L.G.J.H.D.H.J.H.F.K.P.T.G..T.A.: Immunotherapies: Exploiting the immune system for cancer treatment. Journal of immunology research **2018**, 1–16 (2018)
17. Lu, Y., Chen, Y., Zhao, D., Liu, B., Lai, Z., Chen, J.: Cnn-g: Convolutional neural network combined with graph for image segmentation with theoretical analysis. IEEE Transactions on Cognitive and Developmental Systems **13**(3), 631–644 (2021). https://doi.org/10.1109/TCDS.2020.2998497
18. Murphy, J.P., Konda, P., Kowalewski, D.J., Schuster, H., Clements, D., Kim, Y., Cohen, A.M., Sharif, T., Nielsen, M., Stevanovic, S., Lee, P.W., Gujar, S.: Mhc-i ligand discovery using targeted database searches of mass spectrometry data: Implications for t-cell immunotherapies. Journal of Proteome Research **16**(4), 1806–1816 (2017). https://doi.org/10.1021/acs.jproteome.6b00971
19. Nielsen M, A.M.: Nnalign: a platform to construct and evaluate artificial neural network models of receptor-ligand interactions. Nucleic acids research **45**, 344–349 (2017). https://doi.org/https://doi.org/10.1093/nar/gkx276
20. O'Donnell TJ, Rubinsteyn A, B.M.R.A.L.U.H.J.: Mhcflurry: Open-source class i mhc binding affinity prediction. Cell Systems **7**(1), 129–132.e4 (2018). https://doi.org/https://doi.org/10.1016/j.cels.2018.05.014
21. O'Donnell TJ, Rubinsteyn A, L.U.: Mhcflurry 2.0: Improved pan-allele prediction of mhc class i-presented peptides by incorporating antigen processing. Cell Systems **11**(1), 42–48.e7 (2020). https://doi.org/https://doi.org/10.1016/j.cels.2020.06.010

22. Paul S, Grifoni A, P.B.S.A.: Major histocompatibility complex binding, eluted ligands, and immunogenicity: Benchmark testing and predictions. Frontiers in immunology **10**, 3151 (02 2020). https://doi.org/https://doi.org/10.3389/fimmu.2019.03151
23. Renaud N, Geng C, G.S.A.F.R.L.M.D.R.M.B.A.X.L.: Deeprank: a deep learning framework for data mining 3d protein-protein interfaces. Nat Commun **12**(1), 7068 (2021). https://doi.org/https://doi.org/10.1038/s41467-021-27396-0
24. Réau, M., Renaud, N., Xue, L.C., Bonvin, A.M.J.J.: DeepRank-GNN: a graph neural network framework to learn patterns in protein–protein interfaces. Bioinformatics **39**(1) (11 2022). https://doi.org/10.1093/bioinformatics/btac759
25. Tan, S., Li, D., Zhu, X.: Cancer immunotherapy: Pros, cons and beyond. Biomedicine Pharmacotherapy **124**, 109821 (2020). https://doi.org/https://doi.org/10.1016/j.biopha.2020.109821
26. Tonddast-Navaei, S., .S.J.: Are protein-protein interfaces special regions on a protein's surface? The Journal of chemical physics **143**(24) (2015). https://doi.org/https://doi.org/10.1063/1.4937428
27. Wu, T., Lu, Y., Zhu, Y., Zhang, C., Wu, M., Ma, Z., Guo, G.: Ginet: Graph interaction network for scene parsing (2020). https://doi.org/10.48550/ARXIV.2009.06160
28. Zhang, X., S.P.K.P.G.S..G.W.E.: Personalized cancer vaccines: Targeting the cancer mutanome. Vaccine **35**, 1094–1100 (2017). https://doi.org/10.1016/j.vaccine.2016.05.073