A Survey on Machine Learning Techniques on Immunotherapy

CS 466 Intro to Bioinformatics

Edward(Qianfu) Tang[1] and Zhengyu Li[2]

[1]Bioinformatics, University of Illinois, Urbana-Champaign
[2]Computer Science, University of Illinois, Urbana-Champaign

December 2021

Abstract

The field of immunotherapy is the treatment of cancer by activating or suppressing the immune system. There are a lot of underlying challenges [1] that remains for such therapy, which includes building a pre-clinical model that translates to human immunity, determining the dominant cancer driver, or so on. One of the main challenge lies on the recognition of T cells, which helps control immune system and kill cancer cells. In this survey, we aim to make a comprehensive study of how machine learning models are used in immunotherapy and how the architectures of those models help tackle difficult tasks in immunotherapy, and how the performance of these models compare to each other.

Background

Human bodies are vulnerable and prone to diseases and viruses. Thankfully, the immune system in our body serves as a defense against external pathogens, and their function is to recognize and kill the tumor and cancer cells [2]. However, cancer and tumor cells can also block natural immune response, and that is when immunotherapy comes in to restore the active response of one's immune system. This is not an easy task because there is by no means an universal solution to all types cancer cells and all patients. Being able to customize immunotherapy for individual patients against individual diseases and determine whether someone will be responding positively is a great challenge and sometimes we need to use the power of machine learning to help.

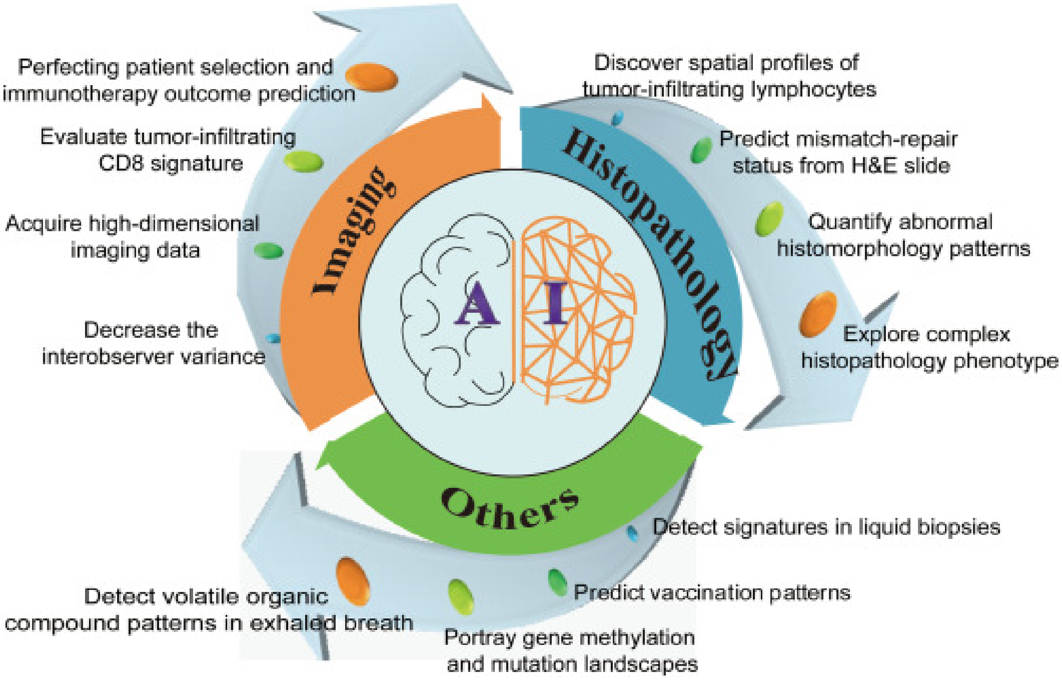
Perfecting patient selection and immunotherapy outcome prediction
Evaluate tumor-infiltrating CD8 signature
Acquire high-dimensional imaging data
Decrease the interobserver variance
Imaging
Histopathology
Discover spatial profiles of tumor-infiltrating lymphocytes
Predict mismatch-repair status from H&E slide
Quantify abnormal histomorphology patterns
Explore complex histopathology phenotype
A I
Others
Detect signatures in liquid biopsies
Predict vaccination patterns
Portray gene methylation and mutation landscapes
Detect volatile organic compound patterns in exhaled breath

Figure 1 : The application of AI-based technologies in immunotherapy and their potential clinical consequences [3].

One type of such immunotherapy uses immune checkpoint blockers (ICB), which involves using drugs to tell the immune system to ignore the shutdown orders from cancer cell. However, due to various types of cancers and patients, this kind of therapy only got response from 1/3 of patients. In order to identify why this is the case, and to get relatively accurate predictions on whether a patient will respond to ICB, researchers have developed artificial intelligence with various machine learning models to conquer this task.

Before we dive into the architecture, we first need to explore our immune system. T-cell, also known as T lymphocyte, is a type of white blood cell that determines the specificity of immune response to antigens (foreign substances) in the body. As Mösch et al. stated in their paper[4], therapies based on the power of T cells can enhance T cell recognition by introducing TCRs that preferentially direct T cells to tumor sites (TCR-T therapy) or through vaccination to induce T cells while it is active; and also alternatively, through creating a microenvironment favorable for cytotoxic cell (which binds to and kills infected cells and cancer cells.) activity through the ICB aforementioned. CD8+ T cells can detect and destroy malignant cells by binding to peptides presented on cell surfaces by MHC (major histocompatibility complex) class I molecules. This provides a molecular biology background of this study.

Motivation

In 2003, Nielson et al.[5] first proposed NetMHC, a neural network based model that predicts T-cell epitopes (peptides derived from antigens and recognized by the T-cell receptor (TCR) when bound to MHC molecules displayed on the cell surface of APCs). The model used Blosum matrix and Hidden Markov Model based encoding for amino acids before training. Nowadays, NetMHC has upgraded to 4.0 and has several variants that are based off the architecture. There are also state-of-the-art models such as pMHC–TCR [6] published by Lu et al. in 2021 that is trained from a long-short term memory (LSTM) deep-learning network that represents protein sequence of antigens numerically. Our goal is to analyze how those models perform, especially the pros and cons of different architectures in those models, especially the NetMHC variants.

NetMHC

Model

The HMM model was constructed using the hmmbuild command from the Hmmer package [7]. An epitope similarity score S for the 9-amino-acid long peptide is calculated as

$$S = \sum_{i=1,\ldots,9} 2 * \log(P_i/Q_i)/\log(2)$$

where P_i is the probability for finding a given amino acid on position i in the HMM and Q_i is the probability for finding the amino acid in the Swiss Prot database[8].

The encoding consists of several steps. The first one is a sparce encoding where each amino acid is encoded as a 20-digit binary number (a single 1 and 19 zeros) resembling one-hot vector; the second is the Blosum50 encoding in which the amino acids are encoded as the Blosum50 score for replacing the amino acid with each of the 20 amino acids. The last encoding is in the form of HMM described above. The model is trained and tested by cross-validating the first dataset, and then the performance is evaluated by Pearson correlation coefficient:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2(y_i - \bar{y})^2}}$$

The neural network architecture used is a conventional feed-forward network with an input layer with 180 neurons, one hidden layer with 2–10 neurons, and a single neuron output layer. Back-propagation is used to update the weight of the neural network.

Data Collection

All machine learning models require training, testing and validation data. In the orginial NetMHC model, there existed two sets of data to begin with. The first set- the training and testing data was consisted of 528 nine-mer amino acids peptides for which the binding affinity to the human leukocyte antigen (HLA) class I molecule A*0204 has been measured using an improved spun column chromatography technique originated from a 1995 paper [9]. The dataset prepared through this technique is later referred to as Buus dataset. The data preprocessing involves the purification of MHC Class 1 molecules (K^k) with K^k-specific antibody, manual peptide synthesis, radioiodination of $\beta_2 m$, insulin and of peptide using a chloramine T method, binding of peptides to MHC molecules, and spun column chromatography.

The second set of data is used to train the hidden Markov model, which is originated from the SYFPEITHI database described in [10]. This database is made available online at http://www.syfpeithi.de, and comprises more than 7000 peptide sequences known to bind class I and class II MHC molecules.

As the algorithm evolves through time, the variants of NetMHC has been applied to more and more different databases. Among those databases, there is one specific database that caught our eyes, and that is the renowned Immune Epitope Database (or abbreviated as IEDB, see www.iedb.org). Funded by National Institute of Allergy and Infectious Diseases (NIAID), the database integrated experimental data regarding various T cell epitope and antibody. Furthermore, it has also gathered some prediction tools on their website, among which some of them are for users to do the predictions online. Several prediction algorithms are supported, including IEDB recommended (right now the recommended algorithm is NetMHCpan EL 4.1), Consensus, NetMHCpan BA 4.1, ANN 4.0, SMMPMBEC, SMM, CombLib, PickPocket, NetMHCcons, and NetMHCstabpan.

The Evolution of NetMHC Algorithms

The first generation of NetMHC model was published in 2003. At that time, it was a novel idea to develop a model that consists of a combination of sparse encoding, Blosum encoding, and input derived from HMM. It used neural network as a tool of peptide-MHC binding. In studies related to the first model, 81 different human allele were used, including but not limited to HLA-A,HLA-B, HLA-C, and HLA-E. The algorithm has changed a lot in this almost 20-year-span, with the latest model being NetMHCpan - 4.1. While the NetMHC 2.0 model has been discontinued, the NetMHC 3.0 model is upgraded that it is able to predict 8, 10, 11-mer peptide binding using 9-mer trained predictor [11], and is trained on the IEDB data in addition of the SYFPEITHI data. The major breakthrough of NetMHCpan-4.0+ models and all models before that, is that the previous algorithms only used binding data from outside the human body. This completely ignored the potential steps when the antigens are being processed and transported within the human body. Apart from that, NetMHCpan also simutaneously integrated data from binding affinity and MS eluted ligand. This had generated the new model some significantly better results than before.

Comparison between NetMHC algorithms

Ok, now we have all these NetMHC algorithms in place. But how shall we choose from these algorithms that all seem quite compelling? A 2019 study [12] compared 13 different MHC I binding algorithms form an independently collected database. The tested algorithms in this study include NetMHC 4.0, NetMHC 3.4, NetMHCpan 4.0, NetMHCpan 3.0, NetMHCpan 2.8, MHCflurry 1.2, PickPocket 1.1, IEDB SMM, IEDB SMMPMBE-Cand SYFPEITHI. Among which, some of them already had newer versions (such as NetMHCpan), but the experiment still provides a set of reliable methods when we want to compare these algorithms.

Basically, the algorithms are evaluated using receiver operating characteristic curve (or abbreviated as an ROC curve, a curve that is created by drawing the true positive rate, or TR, against the false positive rate, or FPR, at various threshold settings) and the area under the curve (or abbreviated as AUC). Note that from statistics:

$TPR = Sensitivity = Recall = TP/(TP + FN)$, and
$FPR = 1 - Specificity = FP/(FP + TN)$

According to the authors, several TPR and FPR threshold are recommended when testing, namely when: (i) $FPR \leq 0.33$, (ii) $TPR \geq 2 * FPR$, and when (iii) the threshold yielding the highest possible sensitivity within the limits defined in (i) and (ii).

The study shows that when we consider pooled peptide lengths, all algorithms performed similarly. However for different lengths of peptides and HLA, the following algorithms differed significantly:

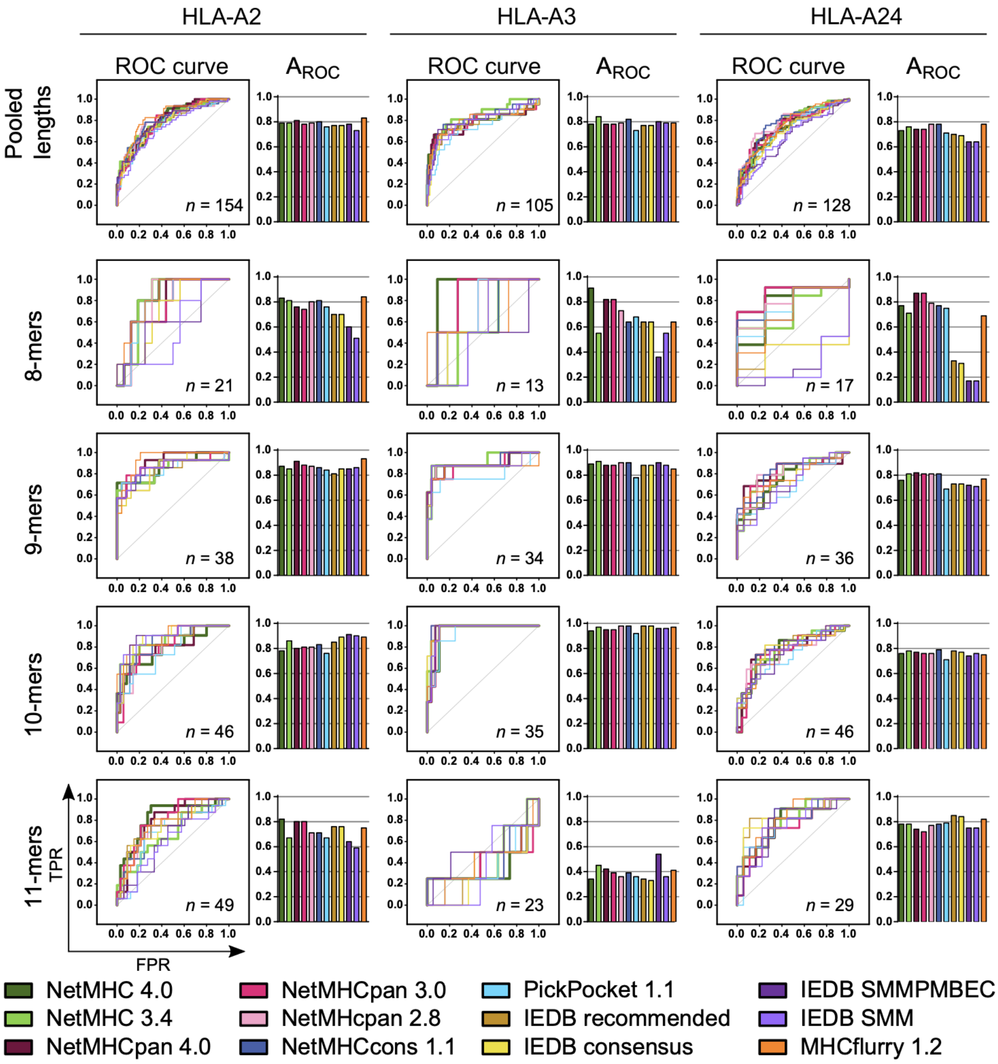

Figure 2: Performances of NetMHC variants on 8 to 11 mers on different antigens.

Hence judging from the results, it seems that pan-specific algorithms that are based on neural networks almost always delivers the best results, which makes NetMHCpan the top of the list. In comparison, algorithms such as IEDB SMM and IEDB SMMPMBEC seems less compelling, scoring at the bottom for the most of the time. The newest algorithms to heir date, NetMHCpan 4.0 and MHCflurry 1.2 do not seem to differentiate others a lot, and it seems that the best way of choosing the right algorithm is still based on the length of the peptide.

Furthermore, the choice of parameters are also discussed in this paper. Although it seems that no significant conclusions were drawn (as there were little universal rules being provided in the paper), the paper did introduce different threshold for strong, intermediate, and weak bindings. This could serve as a more standardized way of examining algorithms in future studies.

Extended Model

In 2021, Lu et al. proposed pMHC-TCR binding prediction network (pMTnet) to predict TCR binding specificities of the neoantigens—and T cell antigens in general—presented by class I MHC. This model is transfer-learning-based, and human tumour sequencing data is being processed to make a series of novel observations regarding the sources of immunogenicity, prognosis and treatment response to immunotherapies. The embedding of pMHCs is based on the NetMHCpan, a variant for NetMHC model mentioned above that based on the result, deemed to have the best performance.

Unlike NetMHC models, pMHCnet predicts TCR–pMHC pairing in independent experimental data, so it needs embeddings for both pMHC and TCR data. pMHC is embedded using a LSTM network to numerically represent sequences of antigens and MHC, in the meanwhile, when embedding TCR, CDR3 regions of TCR β chains, which is the key determinant of specificity in antigen recognition, are being focused on. The amino acid symbol is encoded using Atchley factors [13] and then a stack autoencoder is used. After training the pretrained NetMHC-pan model on the new data, the result shows that on the validation set, the area under the ROC of pMTnet achieved > 0.8 on them and outperformed competing software, which further showcases the strength of NetMHC-pan on learning other tasks.

Conclusion

We mainly explored how the architectures of NetMHC-based models and the fact that ML can match the pace with modern medicine regarding generated data and the detection of phenotypic varieties that sneak through human screening [3] is encourging to the field of immunotherapy, and how different neural networks encode the complicated amino acid, antigen sequence, etc is inspiring to future researchers on how machines can learn certain pattern more efficiently than others.

References

[1] Priti S. Hegde and Daniel S. Chen. “Top 10 Challenges in Cancer Immunotherapy”. In: *Immunity* 52.1 (2020), pp. 17–35. ISSN: 1074-7613. DOI: https://doi.org/10.1016/j.immuni.2019.12.011. URL: https://www.sciencedirect.com/science/article/pii/S1074761319305308.

[2] *Machine learning helps in predicting when immunotherapy will be effective.* https://www.sciencedaily.com/releases/2021/06/210630135027.htm.

[3] Zhijie Xua, Xiang Wang, Shuangshuang Zeng, Xinxin Ren, Yuanliang Yan, Zhicheng Gong. *Applying artificial intelligence for cancer immunotherapy.* https://www.sciencedirect.com/science/article/pii/S2211383521000459.

[4] Anja Mösch et al. “Machine Learning for Cancer Immunotherapies Based on Epitope Recognition by T Cell Receptors”. In: *Frontiers in Genetics* 10 (2019), p. 1141. ISSN: 1664-8021. DOI: 10.3389/fgene.2019.01141. URL: https://www.frontiersin.org/article/10.3389/fgene.2019.01141.

[5] Morten Nielsen, Claus Lundegaard,Peder Worning, Sanne Lise Lauemøller, Kasper Lamberth, Søren Buus, Søren Brunak, Ole Lund. *Reliable prediction of T-cell epitopes using neural networks with novel sequence representations.* https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2323871/.

[6] Tianshi Lu, Ze Zhang, James Zhu, Yunguan Wang, Peixin Jiang, Xue Xiao, Chantale Bernatchez, John V. Heymach, Don L. Gibbons, Jun Wang, Lin Xu, Alexandre Reuben, Tao Wang. *Deep learning-based prediction of the T cell receptor–antigen binding specificity.* https://www.nature.com/articles/s42256-021-00383-2#citeas.

[7] S R Eddy. *Profile hidden Markov models.* https://pubmed.ncbi.nlm.nih.gov/9918945/.

[8] A Bairoch, R Apweiler. *The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000.* https://pubmed.ncbi.nlm.nih.gov/10592178/.

[9] S. Buus , A. Stryhn, K. Winther, N. Kirkby, L. O. Pedersen. *Receptor-ligand interactions measured by an improved spun column chromatography technique. A high efficiency and high throughput size separation method.* https://pubmed.ncbi.nlm.nih.gov/7537104/.

[10] H Rammensee, J Bachmann, N P Emmerich, O A Bachor, S Stevanović. *SYFPEITHI: database for MHC ligands and peptide motifs.* https://pubmed.ncbi.nlm.nih.gov/10602881/.

[11] Claus Lundegaard, Kasper Lamberth, Mikkel Harndahl, Søren Buus, Ole Lund, Morten Nielsen, *NetMHC-3.0: accurate web accessible predictions of human, mouse and monkey MHC class I affinities for peptides of length 8–11.* https://academic.oup.com/nar/article/36/suppl_2/W509/2505813?login=true.

[12] Maria Bonsack, Stephanie Hoppe, Jan Winter, Diana Tichy, Christine Zeller, Marius D. Küpper, Eva C. Schitter, Renata Blatnik and Angelika B. Riemer. *Performance Evaluation of MHC Class-I Binding Prediction Tools Based on an Experimentally Validated MHC–Peptide Binding Data Set.* https://cancerimmunolres.aacrjournals.org/content/7/5/719.abstract.

[13] William R. Atchley et al. “Solving the protein sequence metric problem”. In: *Proceedings of the National Academy of Sciences* 102.18 (2005), pp. 6395–6400. ISSN: 0027-8424. DOI: `10.1073/pnas.0408677102`. eprint: `https://www.pnas.org/content/102/18/6395.full.pdf`. URL: `https://www.pnas.org/content/102/18/6395`.