TOWARDS ANTIGENIC PEPTIDE DISCOVERY WITH BETTER MHC-I BINDING PREDICTION AND IMPROVED BENCHMARK METHODOLOGY

Stanisław Giziński[1], **Grzegorz Preibisch**[1,2] ✉, **Piotr Kucharski**[1], **Michał Tyrolski**[1], **Michał Rembalski**[1], **Piotr Grzegorczyk**[1], **and Anna Gambin**[2]

[1]Deepflare, Warsaw, Poland.
[2]University of Warsaw, Warsaw, Poland.
✉gpreibisch@deepflare.ai

ABSTRACT

The Major Histocompatibility Complex (MHC) is a crucial component of the cellular immune system in vertebrates, responsible for, among others, presenting peptides derived from intracellular proteins. The MHC-I presentation is vital in the immune response and holds great promise in vaccine development and cancer immunotherapy. In this study, we analyze the limitations of existing methods and benchmarks for MHC-I presentation. We introduce a new benchmark to measure crucial generalization properties and models' reliability on unseen MHC molecules and peptides. Finally, we present *HLABERT*, a pre-trained language model which significantly surpasses prior methods on our benchmark and also sets new state-of-the-art on the old benchmarks.

1 INTRODUCTION

The Major Histocompatibility Complex (MHC) is a critical cell surface protein in the cellular immune system of vertebrates that plays a vital role in binding peptides derived from intracellular or extracellular proteins. It comes in two variants: MHC Class I (MHC-I) and MHC Class II (MHC-II). MHC-I molecules are found on almost all nucleated cells and present peptides from intracellular proteins to cytotoxic T-cells, which can kill cells that are infected or otherwise abnormal. MHC-II molecules are found primarily on antigen-presenting cells, such as dendric cells and macrophages, and present peptides collected from extracellular proteins to T-helper cells.

If T-cells recognize and bind to a peptide-MHC complex, an immune response can be triggered. While MHC-I is more critical in a cytotoxic response, the latter is more important in a T-helper response. Therefore, the binding of antigenic peptides to MHC-I molecules is an essential step in the cellular immune response and has significant potential in vaccine design and cancer immunotherapy.

The problem of computational prediction of peptide-MHC binding has been approached through several training methods (1). Most of them rely on artificial neural networks and differ in architecture choice and input preprocessing by different padding choices to be able to train and predict different peptide lengths.

MOTIVATIONS AND RELATED RESEARCH

Pan-specific models, as opposed to allele-specific models, are trained on datasets containing multiple alleles, thus allowing for a larger training sample size. While allele-specific models may demonstrate increased accuracy when restricted to specific alleles, the pan-specific models enable generalization across different MHC molecules. Several approaches were developed for MHC presentation prediction. The pan-specific model, *NetMHCpan*, is an ensemble of Multi-Layer Perceptions trained using the `NNAlign_MA` (2) framework and introduced the integration of both binding affinity and

eluted ligand data into training (3; 4). Another popular pan-specific model, *netMHCflurry*, is an ensemble of Convolutional Neural Networks with a Multi-Layer Perceptron classification head (5). Recently proposed, *Trans-pHLA* based on a not-pretrained transformer, shows that transformers are suited for predicting MHC-I presentation (6). *Anthem* is an allele-specific model, utilizing the AODE computational framework, a variant of Naive Bayes, with a separate model for each HLA and limited ability to generalize over new MHC, but generalizes well to unseen peptides (7). To explain the name of the two mentioned models, recall here that the Human leukocyte antigens (HLA) are part of the major histocompatibility complex (or MHC) in humans.

One-to-One methods utilize separate models for different peptide lengths. At the same time, One-to-Many and Many-to-Many approaches involve transforming input sequences to either pad or shorten the sequences to a fixed length, enabling transfer learning across different peptide lengths. *ACME* (8) is a pan-specific Many-to-Many model incorporating a Convolutional Neural Network and an attention mechanism, focusing on exploring generalization among unseen MHCs. *BVLSTM-MHC* (1) is Pan-specific Bilateral and Variable Long Short-Term Memory architecture (BVLSTM), overcoming the issue of variable peptide length without preprocessing input.

All the mentioned methods report high metrics on test sets. However, it has been shown that MHC-I presentation models are susceptible to changes in training dataset size and content (9). Due to this, we investigated existing evaluation methods and identified several weak points:

- **Weak generalization, train-test split, and imbalance.** The train-test split should be performed carefully to avoid the situation when the model was benchmarked on peptides and MHCs already seen. In previous approaches, the test set was created from unseen pairs of peptide-MHC or just unseen peptides (6; 3; 4; 7; 1; 5). This approach could result in high scores on a test set, not due to generalization but due to overfitting of the method to see sequences of peptides or MHCs. To obtain good generalization on unseen MHCs and peptides, we excluded both all MHCs and peptides seen in the test from the train set. This provides a strong generalization benchmark, with high metrics indicating a model understanding of the MHC structure role.
- **Synthetic negative examples.** In previous approaches (6; 3; 4; 7; 1; 5) negative examples, i.e. non-binding peptides, are randomly sampled parts of proteins that were not marked as binding in the assay. This approach was applied both in the train and test sets. As we think it isn't an error to sample additional negatives this way in the train set, we apprehend that adding them to the test set leads to falsely optimistic metrics on that benchmark. The random part of proteins could be much different than the product of the immunoproteasome, which is responsible for producing the peptides for presentation. Due to those synthetic samples in the test set, the benchmark results could be flawed. In our approach, a negative example is a pair (MHC, peptide) with an assay with a negative result.

  Additionally, what is more important — is that kind of random sampling uses the assumption that negative peptides are linear epitopes. Non-linear epitopes which are the product of cis or trans-splicing, play really significant role in immune response and may account for up to 30% of ligandome (10)
- **Biased choose of positive examples.** In *NetMHCpan4.1* authors had used their previous model *NetMHCpan3* (3) to select positive entries for the test set. This allows for a *data leakage* between train and test. Last but not least the problem of limiting the length of the considered peptides should be stated. The exploration of the IEDB database (11) revealed that there are peptides longer than 12 that bind to MHC-I. Despite that, all previous methods fixed this threshold (3; 4; 5; 6; 7; 1).

OUR RESULTS

As explained above, we identified several limitations of existing methods, mostly related to evaluation and benchmark construction. In this work, we propose a new model for pan-specific MHC-I presentation prediction. Moreover, we benchmark multiple previously used methods. Our pan-specific model outperforms all previous solutions, both on previously used benchmarks and new, more restrictive benchmarks. Our results show that existing pan-specific models do not generalize well to unseen alleles and peptides. Our proposed *HLABERT* shows better generalization, both on unseen peptides and alleles and. Summarizing, our main contributions are as follows:

- identification and detailed analysis of several limitations of existing methods and benchmarks;
- creation of new benchmark for measuring the generalization of MHC-I presentation modeling;
- new model called *HLABERT* based on pre-trained transformers, which outperforms all previous methods on both new and old benchmarks.

2 MATERIALS AND METHODS

DATASET CONSTRUCTION

As we have discussed, existing benchmarks and validation methods suffer multiple limitations. Due to this fact, we have constructed our dataset in a way that tries to overcome most of them.

To create the dataset, we downloaded all measurements from IEDB (11). After getting the data, we filtered it, leaving only MHC-I. We have also chosen only MHCs originating from humans for the initial data set, removed all non-linear peptides and peptides containing ambiguous amino acids and dropped all peptides shorter than 7 and longer than 25.

All assays with positive results were assigned 1 and negative 0. In case there were multiple contradictory assay results for specific peptide-MHC pairs, we have assigned them the dominant value. In cases where the number of positive assays was equal to negative ones, the pair were removed from the dataset.

MHC sequences are gathered from IPD (12). MHC's IDs were mapped to their canonical names using *mhcgnomes* python library [1]. Then measurements from IEDB and sequences from IPD were joined using those canonical IDs.

Train-test split. The train-test split was done in the following way: we randomly chose 35 MHC alleles with measurements for more than 50 peptides and less than 7000. This set of peptide-MHC pairs constitutes our test set. From the training set, we excluded all peptides and MHC which appear in the test set. The validation set was created from peptide-MHC pairs which are removed from the training set due to having peptides the same as in the test set. This setting allows us to measure generalization both on unseen peptides and unseen MHC alleles (one part of the validation set measures the generalization of unseen peptides, and the latter measures the generalization of unseen MHCs). Our final dataset has 509540 data points in a train, 128093 in validation, and 71113 in the test. The numbers of unique peptides in the training set, validation, and test are accordingly 361151, 34050, and 54113. The numbers of unique MHC alleles are accordingly 127, 143, and 35. A visual explanation of train-test split construction is presented in Figure 1.

MHC sequences processing. MHC receptor has three α chains — α_1, α_2, α_3 (all α chains have a length of around 90 amino acids), trans-membrane part (around 20 aa), and a part which is inside the cell (around 20-40 aa) and additionally it has $\beta - 2$-microglobulin (light chain). The α_1 and the α_2-2 domains are structural parts of the peptide-binding cleft. The α_3 Ig-like domain mediates the interaction with the CD8 coreceptor, which is necessary to signal T-cell. Also, topologically, it is relatively close to peptide, the same as beta-2-microglobulin (light chain), but is further than α_1 and α_2.

[1] https://github.com/pirl-unc/mhcgnomes

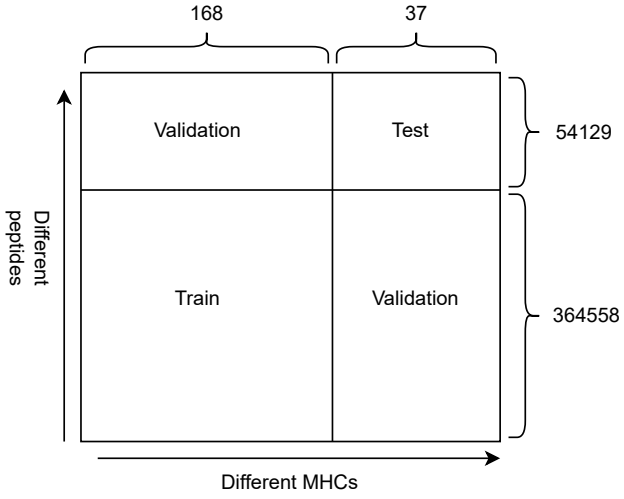

Figure 1: Graphic explanation of the data split. Axes represent different peptides and MHC alleles. Partition is done in a way that prevents any overlap of alleles or peptides between train and test split. Numbers on axes indicate number of peptides/alleles assigned to each partition.

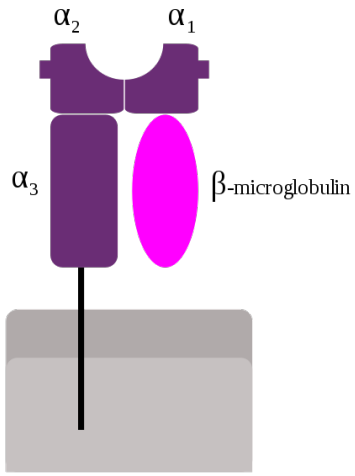

Figure 2: A schematic representation of MHC class I.

β2-microglobulin is necessary for cell surface expression of MHC class I and stability of the peptide-binding groove (13; 14; 15).

From that description, we can conclude that all regions may play a more or less significant role in peptide binding. Therefore, we decided to use full sequences of MHC. Additionally, we compare how the prediction quality differs from using the pseudo-MHC. Pseudo-MHC is a pseudo-sequence consisting of only polymorphic amino acid residues in contact with the peptide (16). The contact residues are defined as being within 4.0 Å of the peptide.

Notice that we compared those two representations of MHC because *HLABERT* was pretrained on full protein sequences, so it seemed to be much more convenient also to explore the following hypotheses: *(i)* if it will improve the prediction because it will be more similar to training data; *(ii)* if BERT is capable of training and find out which parts of the protein are the most crucial for the binding with a peptide.

MODEL ARCHITECTURE

Our model is based on the *DistillProtBERT* architecture (17). *DistillProtBERT* is a pre-trained transformer produced by the distillation of the *ProtBERT* model. Distillation is a compression technique in which a compact model is trained to reproduce the behavior of a larger model (18). *ProtBERT* (19) is a pre-trained transformer, which is trained on an objective of masked language modeling as defined in (20). In the case of *ProtBERT*, the objective is to predict the amino acids in multiple positions based on the rest of the protein. In initial experiments, we have examined the full *ProtBERT* model (19), but we have not obtained significantly better results. Due to shorter training, we decided to proceed with the distilled version.

Our architecture has two variants, as described in Figure 3. The model dualBERT consists of two copied "legs", each of them encoded either MHC or peptide. The produced hidden state is concatenated and forwarded to the last $N - k$ encoder layers. On the other hand, singleBERT encodes MHC and peptide together.

We decided to compare those two architectures because, in the method of training, BERT wasn't trained on pairs of proteins, so it could not be trained in a way to learn about relations between two proteins. Therefore it is more convenient to separate the processing of 2 proteins and then

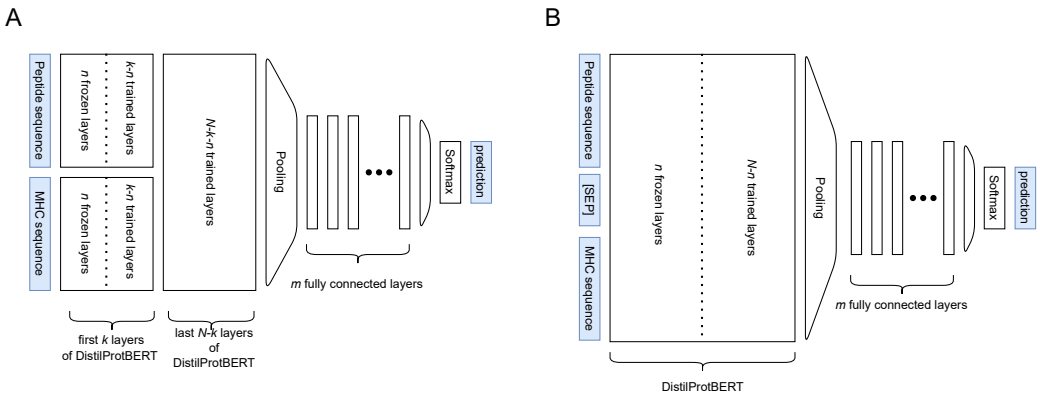

Figure 3: HLABERT architectures. Diagram A shows dualBERT variant, diagram B singleBERT variant. Symbols *m*, *n*, and *k* denote hyperparameters, *N* denotes BERT length.

build transformer blocks over those 2 representations to train the network to understand the relations between proteins in supervised way.

More precisely, it is being fed by batches of sequences in the form:

[CLS] Peptide Sequence [SEP] MHC Sequence [END]

where [CLS], [SEP], and [END] are special tokens, standing for *classification*, *separation*, and sequence end. Variant DualBERT gets

[CLS] Peptide Sequence [END]

[CLS] MHC Sequence [END]

to the first and second leg respectively.

On the head of each of the variants, we use a concatenation of embeddings from multiple poolings. More precisely, given sequence length L, embedding hidden dimension H and embedding vectors $u = [v_1, ..., v_L], u \in \mathbb{R}^{L \times H}$, it produces concatenation of embeddings:

$$\text{ConcatEmbedding} = [\text{CLSEmb}, \text{MeanEmb}, \text{MeanSqrtEmb}, \text{MaxEmb}] \in \mathbb{R}^{4H}$$

$\text{CLSEmb} = v_{\text{ClsIdx}}$ (where ClsIdx is an index of [CLS] token) picks a synthetic token, creates embedding from it, and during the forward pass in the transformer encoder, accumulates all information needed for classification in it.

$\text{MeanEmb} = \frac{\sum_{j=1}^{L} v_j}{L \cdot H} \in \mathbb{R}^H$ aggregates information from all L embeddings by taking the mean of them. Similarly we take MeanSqrtEmb as $\frac{\sum_{j=1}^{L} v_j}{\sqrt{L\dot{H}}}$.

$\text{MaxEmb} = [v_{\arg\max_{i=1...L}(v_{ij}),j}]_{j=1...H}$ takes max for each dimension over all L embeddings.

The vector after pooling is processed by the classification head, which consists of n fully connected layers; a batch normalization layer (21) and dropout (22) follow each layer. Model output is processed by softmax (23). The model is optimized using cross-entropy loss (24) with $L2$ regularization (25). Figure 3 presents the model architecture.

MODEL TRAINING, HYPERPARAMETERS SEARCH AND PERFORMANCE EVALUATION

As the model and training had multiple hyperparameters which impact the model performance, we had to find optimal values of those hyperparameters.

The most important variable hyperparameters were: oversampling proportions, loss functions and their hyperparameters, methods of pooling, dropout, hidden dimensions, and the number of layers of classification (m), and last the binary parameter — complete sequence vs. pseudo-MHC.

As the space of possible parameters is large, we had to perform multiple grid searches to reason about the optimal hyperparameters. Finally, almost 400 experiments have been performed.

Oversampling proportion. Without oversampling model was overfitting to the dominant class, because of the data imbalance, we oversampled the training dataset with different frequencies to determine which frequency led to the best performance on the validation set.

Complete sequence vs. pseudo-MHC. To compare if pseudo-MHC is better than full MHC sequence, we compared multiple models which differ only on the use of full or pseudo-MHC sequence. Additionally, we would like to check if BERT is able to generalize without preprocessing of MHC sequence.

Architecture. As a result, we have a model and training with the following hyperparameters: dimensionality of transformer hidden space: 1024; layers sizes and dropout of head feed-forward network: $[2048, 2048, 2048]$, 0.3 respectively; optimizer: Adam; learning rate: $5e-7$; weight decay: $5e-3$; batch size: 64; the number of epochs: 60. Moreover, we have checked if freezing weights of initial encoder blocks increase performance. In the dualBERT setup, this leads to overfitting, thus we finally stick to the singleBERT version. In this variant, however, freezing doesn't improve our results, so we finally don't freeze anything.

Performance evaluation. As our test is slightly imbalanced (85% of examples are positive), we decided to use include metrics that are suited for imbalanced data. Also, as we focus on antigen discovery, positive examples are more essential for us; therefore, it also shaped the choice of our metrics. We used a few metrics for binary classification: Accuracy, Recall, Precision, F1, AUC, and PR-AUC. The detailed definitions can be found in the Appendix.

3 Results and Discussion

We compared our model *HLABERT* to the models described in the introduction. We used training, and test data from the most popular benchmark (3). We outperformed other models on the test set in terms of AUC. The results of that comparison are described in Table 1

As we can see, our model surpasses other models on AUC and have similar results to NetMHCpan4.1. However, most of the models dropped a magnitude compared to AUC. We asked ourselves the question: why?

Then we explored that most of the models seemed to overfit to 0 because of ill-posed training. Synthetic negative peptides were over 95% of the dataset. As we explored different oversampling techniques, it seems that in this problem, models are really sensitive to a class imbalance in the training set. We could optimize the model to achieve better results than netMHCpan4.1 on that task, however - due to code unavailability and impossible to reproduce model, we didn't know if that web-API model was retrained on that test set, and we would compare our test to their train. Therefore, we decided to rethink the benchmarking method.

	AUCROC	PRAUC
MHCFlurry	0,84	0,42
NetMHCpan*	0,95	0,82
ACME	0,92	0,58
HLABERT	0,95	0,78

Table 1: Results of models on the benchmark proposed by *NetMHCpan-4.1* Authors.
(*) NetMHCpan was not retrained, due to lack of available code. Due to this, metrics of the test set are not reliable, as it was not stated clearly if the model available as the web application was later trained on a full set containing the examples from test set.

Generalization of the model is the final goal of machine learning because it implies how the model will perform in a real-world scenario. Therefore, rigorous benchmarking of the method is a crucial step in the lifespan of every machine-learning model. It is tough when no additional data is provided by the model's author or is insufficient — we believe that benchmarks should be easily reproducible while still being as simple as possible.

	AUCROC			PRAUC		
	train	val	test	train	val	test
MHCFlurry	.92	.84	.78	.99	.93	.95
NetMHCpan*	.89	.88	.90	.99	.96	**.98**
ACME	.88	.87	.32	.99	.96	.96
Phla	.94	.91	.85	.99	.96	.96
Not-Pretrained HLABERT	.96	.91	.84	.96	.97	.96
Dual HLABERT	.96	.88	.77	.96	.95	.94
HLABERT	.94	.92	**.91**$\pm$.0001	.94	.97	**.98**$\pm$.0002

Table 2: Results of models on our benchmark. Models labeled with a star (*) were not retrained due to unavailable code for training or the inability to run training. We tested them on a subset of the test containing HLA alleles not present in the reported training and test set.

Previous benchmarks had limitations such as improper train-testsplit and random negative peptides in test set, which didn't allow the proper measurement of a generalization on unseen MHCs and peptides. The benchmark we proposed resolves these issues - it allows testing how the model will generalize on completely new MHC or pathogen while being much easier to reproduce

As you can see in Table 2 there is a significant drop in generalization on most of the metrics compared to those reported in original papers, which implies that these models don't generalize as well as reported.

Additionally, previous benchmarks contained synthetically sampled peptides as a negative example in the test set, as well as bias on positive examples, due to the usage of the previous model to choose positive peptides. A new benchmark without those limitations allows for a less biased measurement of model performance. Lack of the random negative samples means that the benchmark is more suited for the use-case with known the results of protein cleavage, for example, from mass spectrometry analysis of proteolytic peptides (MAPP) (26).

Our model achieved state-of-the-art results compared to other methods, including *NetMHCpan4.1*, measuring performance on both previous and new benchmarks. However, due to the lack of access to the full model and training data from *NetMHCpan4.1*, proper validation was impossible. Additionally, full hyperparameters of the network have not been published, making it nearly impossible to reproduce their results. Despite these limitations, *NetMHCpan4.1* was validated on our benchmark using the web-API version of *NetMHCpan4.1*, and our results showed improvement. It should be noted that some of the MHC and peptides in the test were also included in the previous training set, suggesting that our model outperforms *NetMHCpan4.1* even further than the metrics indicate.

We found that models trained on full MHC sequence have similar performance on the test set to pseudo MHC. It shows that full sequence doesn't give better results, but it is a hypothesis that needs more exploration. Appropriate choice of mask may make it easier to train and achieve better results.

Additionally, we have compared the not-pretrained HLABERT with pretrained version (see table 2). We have found that not pretrained models achieve worse metrics. Taking into account the out-of-distribution characteristics of our test set, we hypothesize that pretraining has a positive influence on generalization capabilities.

Notice that our method doesn't need additional preprocessing of the sequence, which sometimes might need additional experiments to reason about the structure of the protein.

4 Conclusions and future work

In this work, we have provided an analysis of the shortcomings of previous benchmarks and model validation approaches. As a result of that analysis, we created a new benchmark without those limitations. Then we benchmarked the essential models in the field and compared results with our model. We analyzed the generalization of other models and compared them to ours. We achieved state-of-the-art results on our benchmark and external benchmark provided by Authors of *NetMHCpan4.1*.

Our work is by no means comprehensive in the goals that had been realized. There are multiple steps that could be undertaken:

- To ensure an understanding of the performance of available architectures, measure the performance of other pre-trained models such as T5 (19);
- To provide a further understanding of the model's generalization, create a cross-species benchmarking dataset in which, the model would be trained on N-1 species and tested on the Nth species that the model has not seen before. Such a setting will allow measuring whether the model generalizes over MHC-I polymorphism;
- To provide further information on the model generalization, extend the dataset using non-linear peptides, which are the results of splicing.
- Add more negative samples to have a proportion of negatives to positives similar to real-world data.

5 Availability and acknowledgment

The model and data is available for download from: https://github.com/deepflare-public/HLABERT

Computational part of this study was supported in part by the Poznań Supercomputing and Networking Center (grant number 603) and PLGrid Infrastructure (grant ID plgdeepflare).

The graph in Figure 4 was taken from `https://commons.wikimedia.org/wiki/File:MHC_Class_1.svg` under CC BY 2.5.

References

[1] Limin Jiang, Hui Yu, Jiawei Li, Jijun Tang, Yan Guo, and Fei Guo. Predicting MHC class I binder: existing approaches and a novel recurrent neural network solution. *Briefings in Bioinformatics*, 22(6):bbab216, November 2021.

[2] Bruno Alvarez, Birkir Reynisson, Carolina Barra, Søren Buus, Nicola Ternette, Tim Connelley, Massimo Andreatta, and Morten Nielsen. NNAlign_ma; MHC Peptidome Deconvolution for Accurate MHC Binding Motif Characterization and Improved T-cell Epitope Predictions. *Molecular & cellular proteomics: MCP*, 18(12):2459–2477, December 2019.

[3] Vanessa Jurtz, Sinu Paul, Massimo Andreatta, Paolo Marcatili, Bjoern Peters, and Morten Nielsen. NetMHCpan-4.0: Improved Peptide–MHC Class I Interaction Predictions Integrating Eluted Ligand and Peptide Binding Affinity Data. *The Journal of Immunology*, 199(9):3360–3368, November 2017.

[4] Birkir Reynisson, Bruno Alvarez, Sinu Paul, Bjoern Peters, and Morten Nielsen. NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. *Nucleic Acids Research*, 48(W1):W449–W454, July 2020.

[5] Timothy J. O'Donnell, Alex Rubinsteyn, Maria Bonsack, Angelika B. Riemer, Uri Laserson, and Jeff Hammerbacher. MHCflurry: Open-Source Class I MHC Binding Affinity Prediction. *Cell Systems*, 7(1):129–132.e4, July 2018.

[6] Yanyi Chu, Yan Zhang, Qiankun Wang, Lingfeng Zhang, Lingfeng Zhang, Xuhong Wang, Xuhong Wang, Yanjing Wang, Yanjing Wang, Dennis Russell Salahub, Dennis Russell Salahub, Qin Xu, Qin Xu, Jianmin Wang, Xue Jiang, Xue Jiang, Yi Xiong, Dong-Qing Wei, Donq-qing Wei, and Dong-Qing Wei. A transformer-based model to predict peptide–HLA class I binding and optimize mutated peptides for vaccine design. *Nature Machine Intelligence*, 4(3):300–311, March 2022. MAG ID: 4220655797.

[7] Shutao Mei, Fuyi Li, Dongxu Xiang, Rochelle Ayala, Pouya Faridi, Geoffrey I. Webb, Patricia T. Illing, Jamie Rossjohn, Tatsuya Akutsu, Nathan P. Croft, Anthony W. Purcell, and Jiangning Song. Anthem: a user customised tool for fast and accurate prediction of binding between peptides and HLA class I molecules. *Briefings in Bioinformatics*, 22(5):bbaa415, September 2021.

[8] Yan Hu, Ziqiang Wang, Hailin Hu, Fangping Wan, Lin Chen, Yuanpeng Xiong, Xiaoxia Wang, Dan Zhao, Weiren Huang, and Jianyang Zeng. ACME: pan-specific peptide–MHC class I binding prediction through attention-based deep neural networks. *Bioinformatics*, 35(23):4946–4954, December 2019.

[9] Yohan Kim, John Sidney, Søren Buus, Alessandro Sette, Morten Nielsen, and Bjoern Peters. Dataset size and composition impact the reliability of performance benchmarks for peptide-MHC binding predictions. *BMC Bioinformatics*, 15(1):241, December 2014.

[10] Wayne Paes, German Leonov, Thomas Partridge, Annalisa Nicastri, Nicola Ternette, and Persephone Borrow. Elucidation of the Signatures of Proteasome-Catalyzed Peptide Splicing. *Frontiers in Immunology*, 11:563800, September 2020.

[11] Randi Vita, Swapnil Mahajan, James A. Overton, Sandeep Kumar Dhanda, Sheridan Martini, Jason R. Cantrell, Daniel K. Wheeler, Alessandro Sette, and Bjoern Peters. The Immune Epitope Database (IEDB): 2018 update. *Nucleic Acids Research*, 47(D1):D339–D343, January 2019.

[12] Dominic J Barker, Giuseppe Maccari, Xenia Georgiou, Michael A Cooper, Paul Flicek, James Robinson, and Steven G E Marsh. The IPD-IMGT/HLA Database. *Nucleic Acids Research*, 51(D1):D1053–D1060, January 2023.

[13] Elisabeth Coudert, Sebastien Gehant, Edouard de Castro, Monica Pozzato, Delphine Baratin, Teresa Neto, Christian J A Sigrist, Nicole Redaschi, Alan Bridge, The UniProt Consortium, Alan J Bridge, Lucila Aimo, Ghislaine Argoud-Puy, Andrea H Auchincloss, Kristian B Axelsen, Parit Bansal, Delphine Baratin, Teresa M Batista Neto, Marie-Claude Blatter, Jerven T Bolleman, Emmanuel Boutet, Lionel Breuza, Blanca Cabrera Gil, Cristina Casals-Casas, Kamal Chikh Echioukh, Elisabeth Coudert, Beatrice Cuche, Edouard de Castro, Anne Estreicher, Maria L Famiglietti, Marc Feuermann, Elisabeth Gasteiger, Pascale Gaudet, Sebastien Gehant, Vivienne Gerritsen, Arnaud Gos, Nadine Gruaz, Chantal Hulo, Nevila Hyka-Nouspikel, Florence Jungo, Arnaud Kerhornou, Philippe Le Mercier, Damien Lieberherr, Patrick Masson, Anne Morgat, Venkatesh Muthukrishnan, Salvo Paesano, Ivo Pedruzzi, Sandrine Pilbout, Lucille Pourcel, Sylvain Poux, Monica Pozzato, Manuela Pruess, Nicole Redaschi, Catherine Rivoire, Christian J A Sigrist, Karin Sonesson, Shyamala Sundaram, Alex Bateman, Maria-Jesus Martin, Sandra Orchard, Michele Magrane, Shadab Ahmad, Emanuele Alpi, Emily H Bowler-Barnett, Ramona Britto, Hema Bye A-Jee, Austra Cukura, Paul Denny, Tunca Dogan, ThankGod Ebenezer, Jun Fan, Penelope Garmiri, Leonardo Jose da Costa Gonzales, Emma Hatton-Ellis, Abdulrahman Hussein, Alexandr Ignatchenko, Giuseppe Insana, Rizwan Ishtiaq, Vishal Joshi, Dushyanth Jyothi, Swaathi Kandasaamy, Antonia Lock, Aurelien Luciani, Marija Lugaric, Jie Luo, Yvonne Lussi, Alistair MacDougall, Fabio Madeira, Mahdi Mahmoudy, Alok Mishra, Katie Moulang, Andrew Nightingale, Sangya Pundir, Guoying Qi, Shriya Raj, Pedro Raposo, Daniel L Rice, Rabie Saidi, Rafael Santos, Elena Speretta, James Stephenson, Prabhat Totoo, Edward Turner, Nidhi Tyagi, Preethi Vasudev, Kate Warner, Xavier Watkins, Rossana Zaru, Hermann Zellner, Cathy H Wu, Cecilia N Arighi, Leslie Arminski, Chuming Chen, Yongxing Chen, Hongzhan Huang, Kati Laiho, Peter McGarvey, Darren A Natale, Karen Ross, C R Vinayaka, Qinghua Wang, and Yuqi Wang. Annotation of biologically relevant ligands in UniProtKB using ChEBI. *Bioinformatics*, 39(1):btac793, January 2023.

[14] Andreas Blees, Dovile Januliene, Tommy Hofmann, Nicole Koller, Carla Schmidt, Simon Trowitzsch, Arne Moeller, and Robert Tampé. Structure of the human MHC-I peptide-loading complex. *Nature*, 551(7681):525–528, November 2017. Number: 7681 Publisher: Nature Publishing Group.

[15] George F. Gao, José Tormo, Ulrich C. Gerth, Jessica R. Wyer, Andrew J. McMichael, David I. Stuart, John I. Bell, E. Yvonne Jones, and Bent K. Jakobsen. Crystal structure of the complex between human CD8 and HLA-A2. *Nature*, 387(6633):630–634, June 1997. Number: 6633 Publisher: Nature Publishing Group.

[16] Morten Nielsen, Claus Lundegaard, Thomas Blicher, Kasper Lamberth, Mikkel Harndahl, Sune Justesen, Gustav Røder, Bjoern Peters, Alessandro Sette, Ole Lund, and Søren Buus. NetMHCpan, a Method for Quantitative Predictions of Peptide Binding to Any HLA-A and -B Locus Protein of Known Sequence. *PLOS ONE*, 2(8):e796, 2007. Publisher: Public Library of Science.

[17] Yaron Geffen, Yanay Ofran, and Ron Unger. DistilProtBert: a distilled protein language model used to distinguish between real proteins and their randomly shuffled counterparts. *Bioinformatics*, 38(Supplement_2):ii95–ii98, September 2022.

[18] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, February 2020. arXiv:1910.01108 [cs].

[19] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing, May 2021. arXiv:2007.06225 [cs, stat].

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456, Lille, France, 2015. JMLR.org.

[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, January 2014.

[23] John S. Bridle. Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. In Françoise Fogelman Soulié and Jeanny Hérault, editors, *Neurocomputing*, pages 227–236. Springer Berlin Heidelberg, Berlin, Heidelberg, 1990.

[24] I. J. Good. Rational Decisions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 14(1):107–114, January 1952.

[25] Andrew Y. Ng. Feature selection, L_1 vs. L_2 regularization, and rotational invariance. In *Twenty-first international conference on Machine learning - ICML '04*, page 78, Banff, Alberta, Canada, 2004. ACM Press.

[26] Hila Wolf-Levy, Aaron Javitt, Avital Eisenberg-Lerner, Assaf Kacen, Adi Ulman, Daoud Sheban, Bareket Dassa, Vered Fishbain-Yoskovitz, Carmelo Carmona-Rivera, Matthias P. Kramer, Neta Nudel, Ifat Regev, Liron Zahavi, Dalia Elinger, Mariana J. Kaplan, David Morgenstern, Yishai Levin, and Yifat Merbl. Revealing the Cellular Degradome by Mass Spectrometry Analysis of Proteasome-Cleaved Peptides. *Nature biotechnology*, page 10.1038/nbt.4279, October 2018.

[27] Takaya Saito and Marc Rehmsmeier. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, 10(3):e0118432, March 2015.

A PERFORMANCE EVALUATION - METRICS DEFINITIONS

Definition of variables: True Positive (TP) - Number of correctly predicted binders by the model, True Negative (TN) - Number of correctly predicted binders by the model, False Negative (FN) - Number of binders predicted as non-binders by the model, False Positive (FP) - Number of non-binders predicted as binders by the model.

Accuracy is a metric that quantifies the proportion of correctly predicted examples out of all examples. Mathematically, it can be expressed as:

$$\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$$

Recall is a measure of the ability of a binary classifier to identify all positive instances correctly. It is defined as the ratio of true positive predictions to the sum of true positive and false negative predictions. In probabilistic interpretation, it is the probability of correctly predicting a positive example, given that it is a positive instance. Mathematically, recall can be expressed as:

$$\frac{\text{TP}}{\text{TP} + \text{FN}}$$

Precision is a measure of the accuracy of positive predictions by a binary classifier. It is defined as the ratio of true positive predictions to the sum of true positive and false positive predictions. In probabilistic interpretation, it is the probability of a positive prediction being correct, given that the classifier predicted it as positive. Mathematically, precision can be expressed as:

$$\frac{\text{TP}}{\text{TP} + \text{FP}}$$

F1 Score is a metric that combines precision and recall by taking their harmonic mean. This metric is especially useful in imbalanced datasets, where there is a trade-off between precision and recall. The F1 score can be expressed as:

$$\frac{2\ \text{Precision}\ \text{Recall}}{\text{Precision} + \text{Recall}}$$

False Positive Rate (FPR) is a measure of the proportion of negative instances that are incorrectly classified as positive by a binary classifier. In probabilistic words, it is the probability that the model will label the negative sample incorrectly:

$$\frac{\text{FP}}{\text{TN} + \text{FP}}$$

Area Under the Receiver Operating Characteristic Curve (AUROC) is a metric used to evaluate the performance of binary classifiers. The AUROC measures how the trade-off between sensitivity and specificity changes, depending on the threshold, by computing the area under the ROC curve. The ROC curve plots the true positive rate (sensitivity) against the false positive rate at different threshold settings. The AUROC in probabilistic interpretation is the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one, assuming that positive instances rank higher than negative instances.

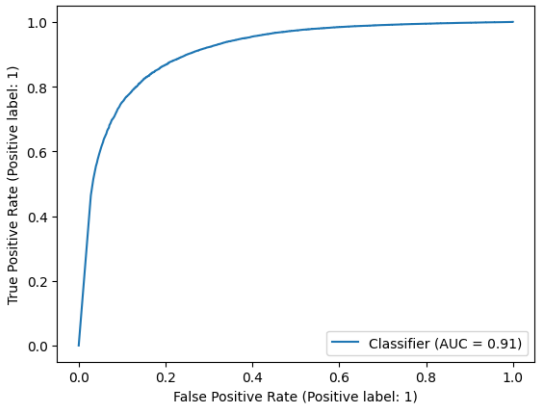

Figure 4: AUR-ROC at test set of HLABERT

PR-AUC - To handle the trade-off between sensitivity and precision, which depends on the threshold, we compute the area under the PR curve, which plots the true positive rate (Sensitivity) against the precision at different threshold settings for a binary classifier. It is proven that it is a better metric than AU for strongly imbalanced datasets. That can be approximated as averaged precision:

Precision-Recall Area Under the Curve (PR-AUC) is a metric used to evaluate the performance of binary classifiers in cases where the positive class is rare or when false negatives are more detrimental than false positives (27). The PR-AUC handles the trade-off between sensitivity and precision, which is dependent on the threshold, by computing the area under the PR curve. The PR curve plots the true positive rate (sensitivity) against the precision at different threshold settings. The PR-AUC is a better metric than AUROC for strongly imbalanced datasets and can be approximated as the average precision, expressed as:

$$\sum nP_n(R_n - R_{n-1})$$

where P_n is Precision at index n and R_n is Recall at index n.