# PRIME

1. Ref: Gfeller et al. Improved predictions of antigen presentation and TCR recognition with MixMHCpred2.2 and PRIME2.0 reveal potent SARS-CoV-2 CD8+ T-cell epitopes. Cell Systems, 2023.
2. Uploaded peptide sequences separately for each HLA to Webserver: http://ec2-18-188-210-66.us-east-2.compute.amazonaws.com:3000/
3. No mouse or MHCII predictions available, only HLAI predictions available.
4. We used the PRIME score for prediction.

# IEDB

1. Ref for MHC I: Calis JJA, Maybeno M, Greenbaum JA, Weiskopf D, De Silva AD, Sette A, Kesmir C, Peters B. 2013. Properties of MHC class I presented peptides that enhance immunogenicity. PloS Comp. Biol. 8(1):361. PMID: 24204222
2. For MHC I, uploaded peptide sequences separately for each MHC to Webserver <https://nextgen-tools.iedb.org/pipeline?tool=tc1> with prediction model “Class I pMHC Immunogenicity” with default (1,2, C terminal) positions to mask and Peptide Length(s) “as is”. Prediction for HLA-B\*81:01 returned error.
3. Ref for MHCII: <http://tools.iedb.org/mhcii/reference/>
4. For MHC II data (binding predictions only), uploaded peptide sequences separately for each MHC to Webserver <http://tools.iedb.org/mhcii/> with “Prediction method:” set to “NetMHCIIPan 4.1 EL”, and length is selected “as is”.

**pMTnet\_Omni**

1. Ref: “pan-MHC and cross-Species Prediction of T Cell Receptor-Antigen Binding”, <https://www.biorxiv.org/content/early/2023/12/12/2023.12.01.569599> for pMTnet\_Omni and 'Deep learning-based prediction of T cell receptor-antigen binding specificity.'(https://www.nature.com/articles/s42256-021-00383-2) Lu, T., Zhang, Z., Zhu, J. et al. 2021 for pMTnet.
2. Prediction is available only for TCRs with at least the CDR3b sequence.
3. For TCRs with full sequence, we formatted the data according to <https://pmtnet-omni-document.readthedocs.io/en/latest/input_format/index.html> (more details in the code to generate input) and used the online webserver of pMTnet\_Omni at <https://dbai.biohpc.swmed.edu/pmtnet/analysis-omni.php> for predictions. We used the logit column as prediction.
4. For TCR2-T with only CDR3 sequence available, we used pMTnet (v1), and used the webserver <https://dbai.biohpc.swmed.edu/pmtnet/analysis-base.php>

# NetTepi

1. Ref: NetTepi: an integrated method for the prediction of T cell epitopes. Trolle T, Nielsen M. Immunogenetics (2014). 66(7-8):449-56.
2. Only HLAI predictions are available. Among them, no predictions exist for HLA-B\*81:01.
3. We uploaded peptide sequences separately for each HLA and for 9mers and 10mers to the webserver at <https://services.healthtech.dtu.dk/services/NetTepi-1.0/> , set default values for Relative weight on stability prediction: 0.16 and Relative weight on T cell propensity prediction: 0.10 and the respective peptide lengths.
4. T Cell Propensity score gave better AUC than combined score, so we report the former score.

# NetTCR-2.2

1. Ref: Jensen, M. F., & Nielsen, M. (2023). NetTCR 2.2—Improved TCR specificity predictions by combining pan- and peptide-specific training strategies, loss-scaling and integration of sequence similarity. bioRxiv. <https://doi.org/10.1101/2023.10.12.562001>
2. From IMGT (e.g., <https://www.imgt.org/IMGTrepertoire/index.php?section=LocusGenes&repertoire=genetable&species=human&group=TRAV>) we get CDR12ab region AA sequences for different TR genes in data.
3. Only MHC-I predictions are available, as mentioned in the webserver. Even though there is no MHC input, the maximum peptide length allowed is 12 (as per the webserver instructions), which is smaller than all MHCII peptide lengths with TCR sequence. Also, the prediction needs CDR3ab as well as CDR12 ab sequences.
4. We submitted the TCR and peptide sequences to the webserver at <https://services.healthtech.dtu.dk/services/NetTCR-2.2/> with the default parameters Similarity scaling factor (α) : 10 and Percentile rank threshold: 100.

# ImRex

1. Ref: Moris, Pieter, Joey De Pauw, Anna Postovskaya, Sofie Gielis, Nicolas De Neuter, Wout Bittremieux, Benson Ogunjimi, Kris Laukens, and Pieter Meysman. “Current Challenges for Unseen-Epitope TCR Interaction Prediction and a New Perspective Derived from Image Classification.” Briefings in Bioinformatics 22, no. 4 (2020). <https://doi.org/10.1093/bib/bbaa318>.
2. Downloaded from GitHub: <https://github.com/pmoris/ImRex>
3. To run ImRex, go to its main folder in “C:\Users\amita\Downloads\local\_runs\_softwares\ImRex-master\ImRex-master”. So go to miniconda terminal, cd to the above folder, then activate conda environment deepTCR (NOT imrex-env) and finally run the code with input output arguments

* cd C:\Users\amita\Downloads\local\_runs\_softwares\ImRex-master\ImRex-master
* Conda activate deepTCR
* python ./src/scripts/predict/predict.py --model ./models/pretrained/2020-07-24\_19-18-39\_trbmhcidown-shuffle-padded-b32-lre4-reg001/2020-07-24\_19-18-39\_trbmhcidown-shuffle-padded-b32-lre4-reg001.h5 --input imrex\_input\_peptides.csv --output imrex\_output\_predictions.csv

The results will be in imrex\_output\_predictions.csv file.

1. While, running the above, received the following output messages in terminal:
   1. Filtered CDR3 sequences to length: (10, 20)
   2. Filtered epitope sequences to length: (8, 11)
2. From GitHub readme: “…model was trained on the VDJdb dataset (August 2019 release) that was filtered on human TRB data, no 10x data and restricted to 10-20 (CDR3) or 8-11 (epitope) amino acid residues,..”

# ERGO-II

1. Ref:  Springer I, Tickotsky N and Louzoun Y (2021), Contribution of T Cell Receptor Alpha and Beta CDR3, MHC Typing, V and J Genes to Peptide Binding Prediction. *Front. Immunol. 12:664514.* [doi: 10.3389/fimmu.2021.664514](https://www.frontiersin.org/articles/10.3389/fimmu.2021.664514/full)
2. Downloaded from GitHub: <https://github.com/IdoSpringer/ERGO-II/tree/master>
3. To run ERGO, go to its main folder in “C:\Users\amita\Downloads\local\_runs\_softwares\ERGO-II-master\ERGO-II-master”. So go to miniconda terminal, cd to the above folder, then activate conda environment and finally run the code with input output arguments

* cd C:\Users\amita\Downloads\local\_runs\_softwares\ERGO-II-master\ERGO-II-master
* Conda activate ergo-env
* python Predict.py mcpas ergo\_input\_peptides.csv (could also use vdjdb in place of mcpas)

The results will be in results.csv file renamed to ergo\_mcpas\_output.csv file. Mcpas model has better AUC than vdjdb and so we report that.

1. Main GitHub link: <https://github.com/IdoSpringer/ERGO-II/tree/master> (note that the codes Model.py and Predict.py required some changes before they could be run, including renaming a directory “TCR\_Autoencoder” to “Models/AE” in code since TCR\_autoencoder directory is missing. We also added cpu only in load torch settings)

# epiTCR

1. Ref: epiTCR: a highly sensitive predictor for TCR-peptide binding, Bioinformatics 2023 May 4;39(5):btad284
2. GitHub: <https://github.com/ddiem-ri-4D/epiTCR/tree/main>
3. To run epiTCR, go to its main folder in “C:\Users\amita\Downloads\local\_runs\_softwares\ epiTCR-main\ epiTCR -main”. So go to miniconda terminal, cd to the above folder, then activate conda environment epitcr-env and finally run the code with input output arguments

* cd C:\Users\amita\Downloads\local\_runs\_softwares\epiTCR-main\epiTCR-main
* Conda activate epitcr-env
* python predict.py --testfile epitcr\_input\_peptides\_mhci.csv --modelfile models/rdforestWithMHCModel.pickle --chain cem > epitcr\_output\_mhci.csv

The results will be in epitcr\_output\_mhci.csv

file in the main directory

1. MHCI Allele sequences are sourced from <https://github.com/ddiem-ri-4D/epiTCR/blob/main/data/hlaCovertPeudoSeq/HLAWithPseudoSeq.csv.zip>
2. While running, got these warnings:
3. UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.2.0 when using version 1.1.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:

https://scikit-learn.org/stable/model\_persistence.html#security-maintainability-limitations

warnings.warn(

C:\Users\amita\miniconda3\envs\epitcr-env\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 1.2.0 when using version 1.1.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:

<https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations>

1. epiTCR paper says: “CDR3β sequences of 8–19 amino acids were provided as TCR input, and peptide sequences of 8–11 amino acids were given as the corresponding peptide.” The code returns as error if the lengths exceed this limit, so MHCII and TIL1383I TCR predictions could not be done.

# iTCep

# Ref: iTCep: a deep learning framework for identification of T cell epitopes by harnessing fusion features. Front. Genet. 14:1141535. doi: 10.3389/fgene.2023.1141535

# Model downloaded from GitHub: <https://github.com/kbvstmd/iTCep/>

1. For peptides with CDR3b sequence: To run iTCep, go to its main folder in “C:\Users\amita\Downloads\local\_runs\_softwares\iTCep-master\iTCep-master>”. So go to miniconda terminal, cd to the above folder, then activate conda environment itcep-env and finally run the code with input output arguments

* cd C:\Users\amita\Downloads\local\_runs\_softwares\iTCep-master\iTCep-master
* Conda activate itcep-env
* python predict.py --input itcep\_input\_with\_tcr.xlsx --output itcep\_output\_with\_tcr.csv

The results will be in itcep\_output\_with\_tcr.csv file in the main directory.

1. For peptides without matching CDR3b sequence, the model predicts a list of binding TCRs CDR3b sequences. We do not use these predictions
2. The main paper says: “…the peptide-CDR3 pair sequences were padded to the maximum length of 32 and were…” Longer pairs in the input data did not return any result in the output file, so no MHCII predictions.

# Attn-TAP

1. Ref: AttnTAP: A Dual-input Framework Incorporating the Attention Mechanism for Accurately Predicting TCR-peptide Binding. Front. Genet., <https://doi.org/10.3389/fgene.2022.942491>
2. Model downloaded from GitHub: <https://github.com/Bioinformatics7181/AttnTAP>
3. For peptides with CDR3b sequence: To run Attn-TAP, go to its main folder in “C:\Users\amita\Downloads\local\_runs\_softwares\AttnTAP-main\ AttnTAP-main >”. So go to miniconda terminal, cd to the above folder, then activate conda environment attntap-env finally run the code with input output arguments

* cd C:\Users\amita\Downloads\local\_runs\_softwares\ AttnTAP-main\AttnTAP-main>
* Conda activate attntap-env
* python ./Codes/AttnTAP\_test.py --input\_file=attntap\_input.csv --output\_file=attntap\_output\_mcpas.csv --load\_model\_file=./Models/cv\_model\_0\_mcpas\_0.pt
* python ./Codes/AttnTAP\_test.py --input\_file=attntap\_input.csv --output\_file=attntap\_output\_vdjdb.csv --load\_model\_file=./Models/cv\_model\_0\_vdjdb\_0.pt

The results for two models (mcpas and vdjdb) will be in attntap\_output\_mcpas.csv and attntap\_output\_vdjdb.csv. The mcpas model has better AUC, so we report that.

# ATM-TCR

1. Ref: ATM-TCR: TCR-Epitope Binding Affinity Prediction Using a Multi-Head Self-Attention Model

Michael Cai et al, Front. Imm. 2022.

1. Model downloaded from GitHub: <https://github.com/Lee-CBG/ATM-TCR>
2. For peptides with CDR3b sequence: To run ATM-TCR, go to its main folder in “C:\Users\amita\Downloads\local\_runs\_softwares\ATM-TCR -main\ATM-TCR -main >”. So go to miniconda terminal, cd to the above folder, then activate conda environment atmtcr-env finally run the code with input output arguments

* cd C:\Users\amita\Downloads\local\_runs\_softwares\\ATM-TCR -main\ATM-TCR -main>
* Conda activate atmtcr-env
* python main.py --infile data/combined\_dataset.csv --indepfile atmtcr\_input.csv --mode test --cuda False

The results will be in result/pred\_original\_atmtcr\_input.csv file.

# DeepTR

1. Probing T cell response by improved TCR repertoires featurization with deep generative model. Web site: [https://bioinfo.uth.edu/DeepTR.](https://bioinfo.uth.edu/DeepTR)
2. Only HLAI predictions available, no HLA-B\*81:01 prediction available
3. Uploaded peptide data for individual TCRs separately to webserver at <https://bioinfo.uth.edu/DeepTR/Prediction.php> and downloaded the results.

# TCRPrediction

1. Ref: Attention network for predicting T-cell receptor–peptide binding can associate attention with interpretable protein structural properties, <https://www.frontiersin.org/journals/bioinformatics/articles/10.3389/fbinf.2023.1274599/full>
2. Downloaded model from GitHub: <https://github.com/kyoheikoyama/TCRPrediction/tree/main>
3. For peptides with CDR3b sequence: To run TCRPrediction, go to its main folder in “cd C:\Users\amita\Downloads\local\_runs\_softwares\TCRPrediction-main\TCRPrediction-main\>”. So go to miniconda terminal, cd to the above folder, then activate conda environment tcrprediction-env finally run the code with input output arguments

* cd C:\Users\amita\Downloads\local\_runs\_softwares\TCRPrediction-main\TCRPrediction-main\scripts>
* conda activate tcrprediction-env
* python predict.py --model\_key entire\_cross\_newemb --checkpointsjson ../hpo\_params/checkpoints.json --input\_filepath ../data/tcrprediction\_input\_peptides.csv

The results will be in ../data/tcrprediction\_input\_peptides\_entire\_cross\_newemb.csv file. Pred1 is used for prediction.

# HeteroTCR

1. Ref: HeteroTCR: A heterogeneous graph neural network-based method for predicting peptide-TCR interaction, Zilan Yu, Mengnan Jiang & Xun Lan, <https://www.nature.com/articles/s42003-024-06380-6>
2. Downloaded model from GitHub: <https://github.com/yuzilan/HeteroTCR>
3. Prediction was available for peptide with length <15 with CDR3b sequence available.
4. The code requires running both NetTCR2 featurization and HeteroTCR predictions. In the code/predict.py provided, we had to add map\_location=torch.device('cpu') in line 105: model.load\_state\_dict(torch.load(PATH,map\_location=torch.device('cpu'))) to run it locally on CPU. Also had to discard version numbers and some packages in the yml files provided to install them locally. Please refer to the exported env yml files for envs in which we ran the codes.
5. We put the input test.tsv file in data/batman\_datasets/try folder and ran

* cd Downloads/local\_runs\_softwares/HeteroTCR-main/HeteroTCR-main\code
* conda activate nettcr-env
* python extra\_test\_feature.py -sd my\_datasets -td try -tmd exp\_datasets\_iedb\_McPAS\_CNN
* conda deactivate
* conda activate heterotcr-env
* python predict.py -sd batman\_datasets -td try -tmd exp\_datasets\_iedb\_McPAS\_Hetero

The output pred.tsv file will be in the try folder as well

# TITAN

1. Ref: TITAN: T-cell receptor specificity prediction with bimodal attention networks, <https://academic.oup.com/bioinformatics/article/37/Supplement_1/i237/6319659>
2. Model downloaded from GitHub: <https://github.com/PaccMann/TITAN/tree/main?tab=readme-ov-file>
3. For TCR-pMHC pairs with full TCR sequence, we put the TCR ID, epitope ID (.smi file), and TCR-epitope pair files in batman\_inputs folder and ran

* C:\Users\amita\Downloads\local\_runs\_softwares\TITAN-main\TITAN-main
* conda activate titan-env
* python scripts/flexible\_model\_eval.py batman\_inputs/titan\_input\_peptides.csv batman\_inputs/tcr\_seq\_id.csv batman\_inputs/epitope\_seq\_id.smi trained\_model bimodal\_mca titan\_output

The output scores will be in trained\_model/results/titan\_output.npy file

1. To get output scores for all TCR-peptide pairs, we had to change drop\_last=False in scripts/flexible\_model\_eval.py code test\_loader field