# T-Cell Receptor-Peptide Interaction Prediction with Physical Model Augmented Pseudo-Labeling

Nasy

Sep 16, 2022

# Outline

1. Introduction

2. T-Cell Receptor-Peptide Interaction Prediction with Physical Model Augmented Pseudo-Labeling

3. Conclusion

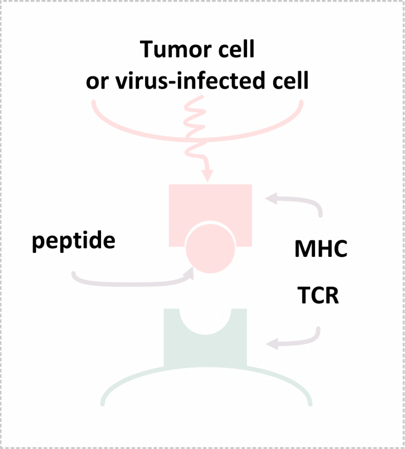
4. References

# TCR Peptide Interaction Prediction

## TCR Peptide Interaction

- The T-cell receptors (TCR) lies on the surface of the T-cell for recognition of foreign peptides.
- Peptides are presented by major histocompatibility complex (MHC) found on the surface of tumor cells or virus-infected cells.
- Common datasets for studying TCR-peptide interactions contain sequences of peptides and sequences of   chain of CDR3 of TCRs.

## Illustration of T-cell receptors (TCR) and peptide binding



# Methods

- Nearest neighbor (SwarmTCR[1])
- Distance-based minimization (TCRdist[2])
- PCA with decision tree[3]
- Random Forest[4]
- Deep Learning[5]

[1] Ryan Ehrlich, et al., "SwarmTCR: A Computational Approach to Predict the Specificity of T Cell Receptors". *BMC Bioinformatics*, vol. 22, no. 1, 7 Sept. 2021, p. 422. https://doi.org/10.1186/s12859-021-04335-w.

[2] Pradyot Dash, et al., "Quantifiable Predictive Features Define Epitope-Specific T Cell Receptor Repertoires". *Nature*, vol. 547, no. 7661, 7661 July 2017, pp. 89–93. https://doi.org/10.1038/nature22383.

[3] Yao Tong, et al., "SETE: Sequence-based Ensemble Learning Approach for TCR Epitope Binding Prediction". *Computational Biology and Chemistry*, vol. 87, 1 Aug. 2020, p. 107281. https://doi.org/10.1016/j.compbiolchem.2020.107281.

[4] Sofie Gielis, et al., "Detection of Enriched T Cell Epitope Specificity in Full T Cell Receptor Sequence Repertoires". *Frontiers in Immunology*, vol. 10, 2019. https://doi.org/10.3389/fimmu.2019.02820; Nicolas De Neuter, et al., "On the Feasibility of Mining CD8+ T Cell Receptor Patterns Underlying Immunogenic Peptide Recognition". *Immunogenetics*, vol. 70, no. 3, 1 Mar. 2018, pp. 159–68. https://doi.org/10.1007/s00251-017-1023-5.

[5] Tianshi Lu, et al., "Deep Learning-Based Prediction of the T Cell Receptor–Antigen Binding Specificity". *Nat Mach Intell*, vol. 3, no. 10, 10 Oct. 2021, pp. 864–75. https://doi.org/10.1038/s42256-021-00383-2; Yiren Jian, et al., "T-Cell Receptor-Peptide Interaction Prediction with Physical Model Augmented Pseudo-Labeling". *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '22, Association for Computing Machinery, 14 Aug. 2022, pp. 3090–97, https://doi.org/10.1145/3534678.3539075.

# Datasets

- Format
  - Positive (TCR, Peptide, MHC)
  - And lots of TCRs
- Dataset
  - VDJdb[6]
  - McPAS-TCR[7]

[6] Dmitry V. Bagaev, et al., "VDJdb in 2019: Database Extension, New Analysis Infrastructure and a T-cell Receptor Motif Compendium". *Nucleic Acids Res*, vol. 48, no. D1, 8 Jan. 2020, pp. D1057–D1062. *31588507*, https://doi.org/10.1093/nar/gkz874.

[7] Nili Tickotsky, et al., "McPAS-TCR: A Manually Curated Catalogue of Pathology-Associated T Cell Receptor Sequences". *Bioinformatics*, vol. 33, no. 18, 15 Sept. 2017, pp. 2924–29. https://doi.org/10.1093/bioinformatics/btx286.

# Paper

T-Cell Receptor-Peptide Interaction Prediction with Physical Model Augmented Pseudo-Labeling[8]

[8] Yiren Jian, et al., "T-Cell Receptor-Peptide Interaction Prediction with Physical Model Augmented Pseudo-Labeling". *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '22, Association for Computing Machinery, 14 Aug. 2022, pp. 3090–97, https://doi.org/10.1145/3534678.3539075.

# Problem

Current datasets for training deep learning models of this purpose remain constrained without diverse TCRs and peptides.

# Solution

Extend training dataset

# Solution

- Data-augmented psudo-label of TCR-peptide pairs
  - Use teacher model to generate pseudo-labels and retrain the model with them
- Physical modeling of TCR-peptide interaction
  - Molecular dynamic (MD)
  - Docking energy

# What is Docking energy

Docking is a computational method for predicting the structures of protein complex (e.g., dimer of two molecules) given the structure of each monomer. It searches the configuration of the complex by minimizing an energy scoring function.

In this work, they use the final docking energy (of the optimal structure of the complex) between a TCR and peptide as the surrogate binding label for the TCR-peptide pair.

# Dataset

- Dataset $\mathcal{D}$
  - VDJdb[9]
  - McPAS-TCR[10]
- Labeled (Training dataset, $\mathcal{D}_{train}$)
  - TCR-peptide pairs with known binding affinity (1 positive, 0 negative)
- Unlabeled
  - TCRdb (no peptide) with peptide from $\mathcal{D}$.
  - $\mathcal{D}_{auxiliary}$

[9] Dmitry V. Bagaev, et al., "VDJdb in 2019: Database Extension, New Analysis Infrastructure and a T-cell Receptor Motif Compendium". *Nucleic Acids Res*, vol. 48, no. D1, 8 Jan. 2020, pp. D1057–D1062. *31588507*, https://doi.org/10.1093/nar/gkz874.

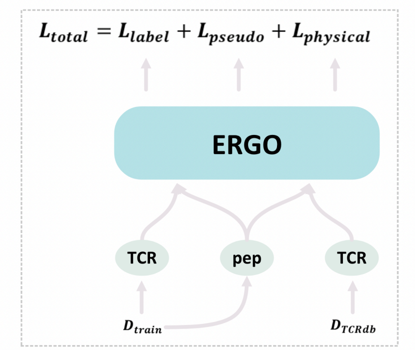
[10] Nili Tickotsky, et al., "McPAS-TCR: A Manually Curated Catalogue of Pathology-Associated T Cell Receptor Sequences". *Bioinformatics*, vol. 33, no. 18, 15 Sept. 2017, pp. 2924–29. https://doi.org/10.1093/bioinformatics/btx286.

# Method

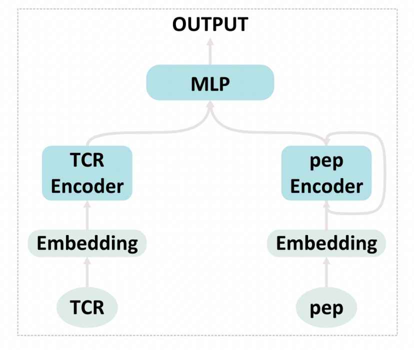
There are four steps in a single training step:

- Learning from labeled dataset $\mathcal{L}_{label}$
- Learning from physical modeling $\mathcal{L}_{phy}$
- Learning from data-augmented pseudo-labeling $\mathcal{L}_{pseudo-label}$
- Look ahead meta-update

## Overview



## ERGO



# Learning from labeled dataset $\mathcal{L}_{label}$

- $pred = f_\theta(t, p)$
  - $t$ is the TCR
  - $p$ is the peptide
  - The embedding of TCR and peptide from ERGO[11].
    - TCRs use LSTM or AE
    - Peptides use LSTM
  - $f_\theta$ is the model
    - $f_\theta = MLP(concat(t, p))$
- $\mathcal{L}_{label} = BCE(pred, y)$

[11] Ido Springer, et al., "Prediction of Specific TCR-Peptide Binding From Large Dictionaries of TCR-Peptide Pairs". *Frontiers in Immunology*, vol. 11, 2020. https://doi.org/10.3389/fimmu.2020.01803.

# Learning from physical modeling $\mathcal{L}_{phy}$

- Molecular dynamic (MD): accurate but slow
- Docking energy: HDOCK[12]
- TCR/Peptide -> BLAST+ -> MSA -> MODELLER -> Structure -> Docking energy
  - Top 25% Negative
  - Bottom 25% Positive
- $pred' = f_\theta(t', p')$
  - $(t', p')$ become tuples in $\mathcal{D}_{auxiliary}$
- $\mathcal{L}_{phy} = BCE(pred', y)$

[12] Yumeng Yan, et al., “The HDOCK Server for Integrated Protein–Protein Docking”. *Nat Protoc*, vol. 15, no. 5, 5 May 2020, pp. 1829–52. https://doi.org/10.1038/s41596-020-0312-x.

# Learning from data-augmented pseudo-labeling $\mathcal{L}_{pseudo-label}$

- $prob = f_{teacher}(t', p')$
- $pred' = f_{\theta}(t', p')$
- $\mathcal{L}_{pseudo-label} = \texttt{KL} - \texttt{divergence}(pred', prob)$

# Look Ahead Meta-Update I

- Learning from labeled dataset
  - $out = model(t, p)$
  - $\mathcal{L}_{label} = BCE(out)$
  - $model.update(\mathcal{L}_{label})$
- Learning from data-augmented pseudo-labeling
  - $out = model(t', p')$
  - $out' = model_{teacher}(t', p')$
  - $\mathcal{L}_{pseudo-label} = KL(out, out')$
  - $model.update(\mathcal{L}_{pseudo-label})$
  - $param = model.param$
- Learning from physical modeling
  - $out = model(t', p')$
  - $\mathcal{L}_{phy} = BCE(out)$
  - $model.update(\mathcal{L}_{phy})$
- Look ahead meta-update
  - Learning Rate * 2
  - $\mathcal{L} = BCE(model(t, p))$

# Look Ahead Meta-Update II

- If $\mathcal{L} > \mathcal{L}_{label}$
  - $model.param = param$

# Look Ahead Meta-Update

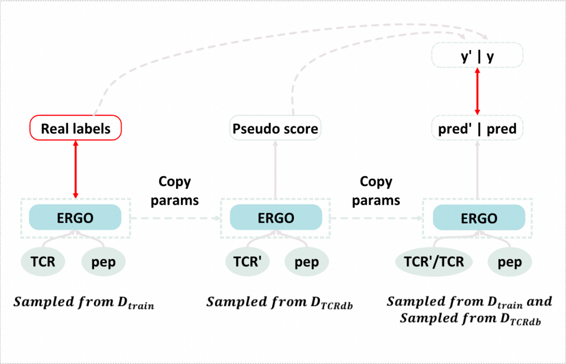
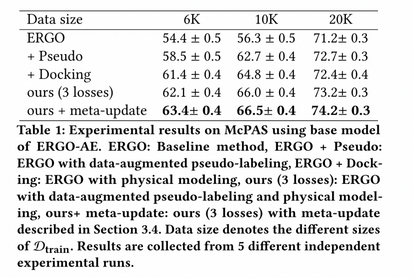


Figure: Overview of learning from data-augmented pseudolabeling. An ERGO model is first learned with TCRs and peptides sample from Dtrain, and this model is used as the teacher model. Then, this teacher model is used for pseudolabeling TCR-peptide pairs from auxiliary dataset. Finally, we re-train an ERGO model with the original dataset and the extended pseudo-labeled dataset.

# Results McPAS

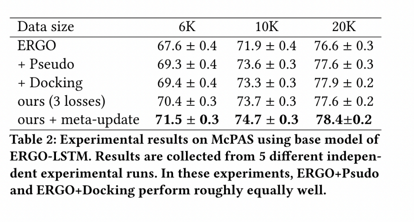
## LSTM



| Data size | 6K | 10K | 20K |
|---|---|---|---|
| ERGO | 54.4 ± 0.5 | 56.3 ± 0.5 | 71.2± 0.3 |
| + Pseudo | 58.5 ± 0.5 | 62.7 ± 0.4 | 72.7± 0.3 |
| + Docking | 61.4 ± 0.4 | 64.8 ± 0.4 | 72.4± 0.4 |
| ours (3 losses) | 62.1 ± 0.4 | 66.0 ± 0.4 | 73.2± 0.3 |
| ours + meta-update | **63.4± 0.4** | **66.5± 0.4** | **74.2± 0.3** |

**Table 1: Experimental results on McPAS using base model of ERGO-AE. ERGO: Baseline method, ERGO + Pseudo: ERGO with data-augmented pseudo-labeling, ERGO + Docking: ERGO with physical modeling, ours (3 losses): ERGO with data-augmented pseudo-labeling and physical modeling, ours+ meta-update: ours (3 losses) with meta-update described in Section 3.4. Data size denotes the different sizes of $\mathcal{D}_{\text{train}}$. Results are collected from 5 different independent experimental runs.**

## AE

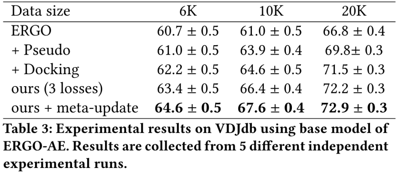


| Data size | 6K | 10K | 20K |
|---|---|---|---|
| ERGO | 67.6 ± 0.4 | 71.9 ± 0.4 | 76.6 ± 0.3 |
| + Pseudo | 69.3 ± 0.4 | 73.6 ± 0.3 | 77.6 ± 0.3 |
| + Docking | 69.4 ± 0.4 | 73.3 ± 0.3 | 77.9 ± 0.2 |
| ours (3 losses) | 70.4 ± 0.3 | 73.7 ± 0.3 | 77.6 ± 0.2 |
| ours + meta-update | **71.5 ± 0.3** | **74.7 ± 0.3** | **78.4±0.2** |

**Table 2: Experimental results on McPAS using base model of ERGO-LSTM. Results are collected from 5 different independent experimental runs. In these experiments, ERGO+Psudo and ERGO+Docking perform roughly equally well.**

# Results VDJdb

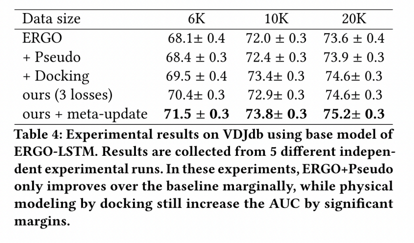
## LSTM



| Data size | 6K | 10K | 20K |
|---|---|---|---|
| ERGO | 60.7 ± 0.5 | 61.0 ± 0.5 | 66.8 ± 0.4 |
| + Pseudo | 61.0 ± 0.5 | 63.9 ± 0.4 | 69.8± 0.3 |
| + Docking | 62.2 ± 0.5 | 64.6 ± 0.5 | 71.5 ± 0.3 |
| ours (3 losses) | 63.4 ± 0.5 | 66.4 ± 0.4 | 72.2 ± 0.3 |
| ours + meta-update | **64.6 ± 0.5** | **67.6 ± 0.4** | **72.9 ± 0.3** |

**Table 3: Experimental results on VDJdb using base model of ERGO-AE. Results are collected from 5 different independent experimental runs.**

## AE

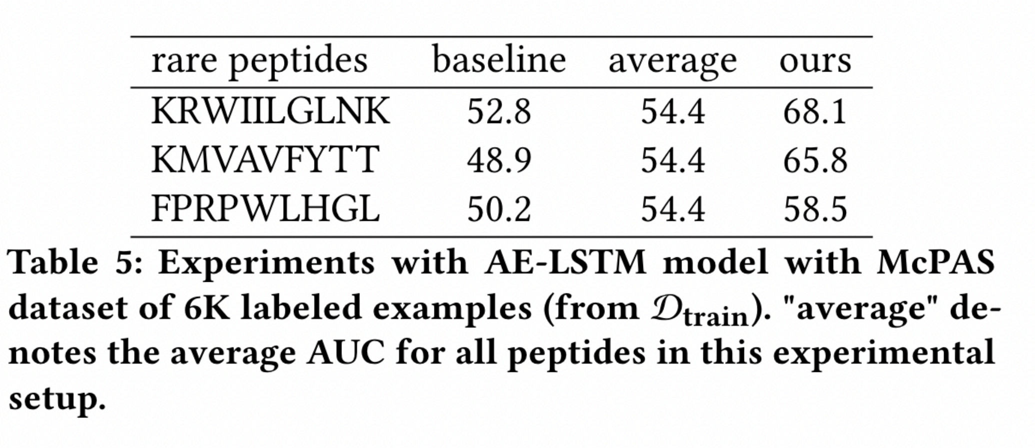


| Data size | 6K | 10K | 20K |
|---|---|---|---|
| ERGO | 68.1± 0.4 | 72.0 ± 0.3 | 73.6 ± 0.4 |
| + Pseudo | 68.4 ± 0.3 | 72.4 ± 0.3 | 73.9 ± 0.3 |
| + Docking | 69.5 ± 0.4 | 73.4± 0.3 | 74.6± 0.3 |
| ours (3 losses) | 70.4± 0.3 | 72.9± 0.3 | 74.6± 0.3 |
| ours + meta-update | **71.5 ± 0.3** | **73.8± 0.3** | **75.2± 0.3** |

**Table 4: Experimental results on VDJdb using base model of ERGO-LSTM. Results are collected from 5 different independent experimental runs. In these experiments, ERGO+Pseudo only improves over the baseline marginally, while physical modeling by docking still increase the AUC by significant margins.**

## Results Rare Peptides

- A rare peptide KRWIILGLNK has only AUC score of 52.8,
- while this method achieves 68.1.
- Note that the average AUC for all peptides is 54.4.



| rare peptides | baseline | average | ours |
|---|---|---|---|
| KRWIILGLNK | 52.8 | 54.4 | 68.1 |
| KMVAVFYTT | 48.9 | 54.4 | 65.8 |
| FPRPWLHGL | 50.2 | 54.4 | 58.5 |

**Table 5: Experiments with AE-LSTM model with McPAS dataset of 6K labeled examples (from $\mathcal{D}_{\text{train}}$). "average" denotes the average AUC for all peptides in this experimental setup.**

- Goal: Improve the prediction of TCR-peptide interactions
- Solution:
  - Docking energies as the physical properties between TCR-peptide pairs
  - Data-augmented pseudo-labeling
  - Look ahead meta-update
  - Experiments on VDJdb and McPAS datasets

# References I

Bagaev, Dmitry V., et al., “VDJdb in 2019: Database Extension, New Analysis Infrastructure and a T-cell Receptor Motif Compendium”. *Nucleic Acids Res*, vol. 48, no. D1, 8 Jan. 2020, pp. D1057–D1062. *31588507*, https://doi.org/10.1093/nar/gkz874.

Dash, Pradyot, et al., “Quantifiable Predictive Features Define Epitope-Specific T Cell Receptor Repertoires”. *Nature*, vol. 547, no. 7661, 7661 July 2017, pp. 89–93. https://doi.org/10.1038/nature22383.

De Neuter, Nicolas, et al., “On the Feasibility of Mining CD8+ T Cell Receptor Patterns Underlying Immunogenic Peptide Recognition”. *Immunogenetics*, vol. 70, no. 3, 1 Mar. 2018, pp. 159–68. https://doi.org/10.1007/s00251-017-1023-5.

Ehrlich, Ryan, et al., “SwarmTCR: A Computational Approach to Predict the Specificity of T Cell Receptors”. *BMC Bioinformatics*, vol. 22, no. 1, 7 Sept. 2021, p. 422. https://doi.org/10.1186/s12859-021-04335-w.

Gielis, Sofie, et al., “Detection of Enriched T Cell Epitope Specificity in Full T Cell Receptor Sequence Repertoires”. *Frontiers in Immunology*, vol. 10, 2019. https://doi.org/10.3389/fimmu.2019.02820.

# References II

- Jian, Yiren, et al., "T-Cell Receptor-Peptide Interaction Prediction with Physical Model Augmented Pseudo-Labeling". *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '22, Association for Computing Machinery, 14 Aug. 2022, pp. 3090–97, https://doi.org/10.1145/3534678.3539075.
- Lu, Tianshi, et al., "Deep Learning-Based Prediction of the T Cell Receptor–Antigen Binding Specificity". *Nat Mach Intell*, vol. 3, no. 10, 10 Oct. 2021, pp. 864–75. https://doi.org/10.1038/s42256-021-00383-2.
- Springer, Ido, et al., "Prediction of Specific TCR-Peptide Binding From Large Dictionaries of TCR-Peptide Pairs". *Frontiers in Immunology*, vol. 11, 2020. https://doi.org/10.3389/fimmu.2020.01803.
- Tickotsky, Nili, et al., "McPAS-TCR: A Manually Curated Catalogue of Pathology-Associated T Cell Receptor Sequences". *Bioinformatics*, vol. 33, no. 18, 15 Sept. 2017, pp. 2924–29. https://doi.org/10.1093/bioinformatics/btx286.
- Tong, Yao, et al., "SETE: Sequence-based Ensemble Learning Approach for TCR Epitope Binding Prediction". *Computational Biology and Chemistry*, vol. 87, 1 Aug. 2020, p. 107281. https://doi.org/10.1016/j.compbiolchem.2020.107281.

# References III

Yan, Yumeng, et al., “The HDOCK Server for Integrated Protein–Protein Docking”. *Nat Protoc*, vol. 15, no. 5, 5 May 2020, pp. 1829–52. https://doi.org/10.1038/s41596-020-0312-x.