# Few Shot Learning for TCR-Epitope Binding Affinity Prediction

**Ajay Kannan**
**Faculty Mentor: Dr. Heewook Lee**

## Introduction

T cells are a subset of lymphocytes that play an important role in the immune response. T cells are activated when the T cell receptor (TCR) is stimulated by an antigen. The TCR is a complex of integral membrane proteins that can bind to an antigen presented by major histocompatibility complex (MHC) molecules on the host cell membrane. Each TCR sequence can bind to a set of specific antigens, requiring the host to generate a large number of TCRs. The binding specificity between a TCR and an antigen is determined by both TCR and antigen sequences. The most important part of an antigen for binding specificity is the epitope, also known as an antigenic determinant.

The recent availability of TCR-epitope binding affinity data from public databases such as IEDB, VJDdb, and McPAS has fueled several machine learning models that are capable of predicting whether a given pair of TCR and epitope binds or not. However, the currently available databases are still nascent and the number of epitopes that are reported in these databases is limited, causing the models to perform poorly on out-of-sample epitopes. In this study, we aim to improve out-of-sample epitopes performance by developing a TCR-epitope binding affinity prediction model.

## Related Works

Several deep learning models have been developed to computational predict the binding affinity of a given TCR-epitope pair. NetTCR [1] is a CNN-based model that has shown that a sequence-based model allows for identifying TCRs binding a given cognate peptide-MHC target out of a large pool of non-binding TCRs. ATM-TCR [2] uses multi-head self-attention which encapsulates the self-attention to capture biological contextual information and to improve its generalization. ERGO [3] employs a pre-trained encoder based on unlabeled TCRs to better capture what TCR sequences are like to improve its prediction performance. While these methods perform fairly (AUC around 0.75) on TCR-epitope pairs where the epitope sequence was seen in training examples (in-sample epitope), they perform poorly on the pairs where the epitope was never seen in training examples (out-of-sample epitope).

Fortunately, there is a machine learning method that addresses this problem. Few-shot learning [4] is an approach first used in image classification especially to manage out-of-sample data. The use of Siamese Neural Networks (SNN) under the framework of Few-shot learning has been shown to be effective at evaluating semantic textual similarity [5]. Measuring semantic similarity is analogous to measuring similarity between two close epitopes as they are hypothesized to bind to similar TCRs.

## Our Approach

We aim to develop a few shot learning approach for TCR-epitope binding affinity prediction that achieves improved performance on out-of-sample epitopes. In order to do so, we find similar TCRs from a given set of pairs of TCRs and Epitopes. That is, from the dataset

which consists of a TCR and epitope pair, predicting one more TCR which binds to the epitope. Traditionally, a neural network learns to predict multiple classes. This poses a problem when we need to add or remove new classes to the data. In this case, we have to update the neural network and retrain it on the whole data set. Also, deep neural networks require a large volume of training data. On the other hand, SNN, a few shot learning technique, is a class of neural network architectures that contain two or more identical sub-networks.

Using an SNN-like model, we plan to train the TCR-Epitope dataset with unseen TCRs with pre-trained embedding. Contrastive loss is used to measure models performance. It is a distance-based loss as opposed to a more conventional error-prediction loss. This loss function is used to learn embeddings in which two similar points have a low Euclidean distance and two dissimilar points or TCRs have a large Euclidean distance. Similar TCRs are supposed to have high affinity scores and can combine with the given epitope. The architecture uses a pre-trained amino-acid embedding model as the embedding network before the SNN-like model. Each TCR represented by a sequence of letter strings is embedded by the embedding network, yielding a sequence of real-valued vectors. Then our model predicts whether the two TCRs are binding to the same epitope by comparing the two embeddings.

We would like to utilize the budget given to us by MORE to purchase storage and computing equipment for developing and testing our prediction model.

**Model Training**

We have pre-trained the amino-acid embedding model on millions of unlabeled TCR data collected from ImmunoSeq. We fine-tune the embedding of TCRs. The parameters of the embedding networks are initialized with the pre-trained weights, and the two embedding networks share identical parameters that are simultaneously updated. We will experiment with stacking a varying number of dense layers on top of each embedding network. The last layer with self-supervised contrastive loss (SCL) discriminates epitope-binding-identity for a given pair of TCR embeddings. That is, if TCR1 and TCR2 bind to the same epitope, then the model minimizes the distance between their embeddings. Suppose they bind to different epitopes, then the model maximizes the distance between their embeddings.

**Impact of Research**

Solving the TCR-epitope binding problem is essential to developing novel clinical applications such as cancer immunology, the looming pandemic crisis, etc. Evaluating which TCRs will bind to target epitopes is challenging, as there are over $10^{15}$ rearrangements of T-cells, each with possible recombination resulting in a distinct TCR. By increasing prediction performance on unseen epitopes, our model can provide a more confident and accurate pool of cognate TCRs that are likely to bind to target epitopes in immunotherapy strategy development. Automating this process can significantly reduce the time and the cost needed for traditional wet lab assays when developing TCR-mediated therapies. The outcome of the project will further the Fulton research theme for health and aid the medical field.

**References**

[1] Montemurro, A., Schuster, V., Povlsen, H.R., Bentzen, A.K., Jurtz, V., Chronister, W.D., Crinklaw, A., Hadrup, S.R., Winther, O., Peters, B. and Jessen, L.E., 2021. NetTCR-2.0

enables accurate prediction of TCR-peptide binding by using paired TCRα and β sequence data. *Communications Biology*, *4*(1), pp.1-13.

[2] Cai, M., Bang, S., Zhang, P. and Lee, H., 2022. ATM-TCR: TCR-Epitope Binding Affinity Prediction Using a Multi-Head Self-Attention Model. *Frontiers in Immunology*, *13*.

[3] Springer, I., Besser, H., Tickotsky-Moskovitz, N., Dvorkin, S. and Louzoun, Y., 2020. Prediction of specific TCR-peptide binding from large dictionaries of TCR-peptide pairs. *Frontiers in Immunology*, p.1803.

[4] Dhillon, G.S., Chaudhari, P., Ravichandran, A. and Soatto, S., 2019. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*.

[5] Ranasinghe, T., Orăsan, C. and Mitkov, R., 2019, September. Semantic textual similarity with Siamese neural networks. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)* (pp. 1004-1011).

# Personal Statement

**Ajay Kannan**

Hailing from India, I did my Undergraduate Studies in computer science and engineering in College of Engineering, Guindy, Anna University, India. During my second year of undergrad, I joined a non-profit research organization. I was interested in the ongoing research in the field of Human-Computer Interaction (HCI) & gesture recognition using Machine Learning. At that time I thought it was for a noble cause, helping deaf and mute people. Though it is not directly or a medical device it aids the daily living of handicapped people. I went on to publish two papers in the field, notably static gesture recognition in Global IOT Summit, Geneva.

That is when my life took a turn. In 2017, I contracted a brain aneurysm and took a break for three years for rehab. As I was taking care of my health parallely I was studying through online courses. One such was bioinformatics and I liked the basic concept. So I applied and got a Data science internship in a bioinformatics company called Sigtuple. They were working with concepts like AI, CNN, different neural networks etc with biological data from blood and urine samples across India. This sparked the research interest in me to read on bioinformatics more.

After joining Arizona State University, I was constantly looking for subjects which had the marriage of computer science and biology. That is when I learnt about the course algorithms in computational biology by Dr Lee. I was interested in Dr. Lee's lectures and research. He introduced to me the research on TCR-epitope and that his research team is working on the state of the art models. Although I lack the academic background in biology, I have a strong background in software development and machine learning which can help me during the developmental phase. With Dr. Lee's help, I can build a strong foundation in the problem of the TCR-epitope topic and I have already discussed it with him. I have a strong foundation in machine learning and deep learning which can help me to implement the problem at hand and solve it. With my previous research skills and my work in the internship in bioinformatics, I believe that I can use this opportunity to further my research career.

# Timeline

**Fall 2023 (pre-MORE) -**

- Background Research for the method
    - Research on Siamese Neural Network
    - Research on Amino-acid embedding
- Gathering the TCR-Epitope from various sources
    - Assimilate into a one dataset after compiling
- Get the model from the pre-trained amino-acid embedding

**Spring 2023 -**

- Implementing Siamese network like architecture with pre-training of amino-acid embedding
    - Developing the architecture of SNN
    - Training the model with pre-trained data
    - Validation and Testing the model with unseen epitopes
- Comparing AUCs of unseen epitopes of various architectural
    - Siamese network like structure (i.e., Pre-training + Siamese)
    - Comparing without SNN (i.e., Pre-training + Dense Layers)
- Comparing with AUCs of unseen epitopes with the existing approaches
    - Comparing without embedding
        - Run ATM-TCR model and compare with our model
        - Run NetTCR model and compare with our model
    - Comparing with embedding
        - Run ERGO model and compare with our model
        - Run catELMo + ATM-TCR model and compare with our model
- Produce the Results

# Ajay Kannan

+1 602 748 9642 ajaykannan@gmail.com
linkedin.com/in/ajay-kannan-34a04013b/ - github.com/AjayKannan97

## Education

**Arizona State University, Ira A. Fulton Schools of Engineering** **Jan. 2021 – Present**
*Master of Science in Computer Science* *Tempe, AZ, USA*

**Anna University, College of Engineering, Guindy** **Aug. 2014 – May 2019**
*Bachelor of Engineering in Computer Science and Engineering* *Chennai, TN, India*

## Relevant Coursework

- Artificial Intelligence
- Full Stack Dev
- Data Analysis
- Internet of Things

## Experience

**Sigtuple** **August – December 2021**
*Data Science Intern* *Bangalore, India*

- Worked on Image processing on real samples, procured from biological data across India. Learnt about Siamese Network, GANs, Neural Style transfer and Alpha Blending.

**Microsoft, IDC** **May 2017 – July 2017, May 2018 – July 2018**
*Software Engineer Intern* *Hyderabad, India*

- Worked on heat-map generation for sensor values based on satellite image, as a part of **"FarmBeats"**, a Microsoft Initiative in 2018.
- Initiated a new project called **Shopping on Cortana** at Microsoft as part of my summer internship in 2017.

**Solarillion Foundation** **August 2015 – December 2018**
*Research Assistant* *Chennai, India*

- Have completed a project and submitted a paper on Generic Dynamic Gesture Recognition using accelerometer system (Machine Learning) from two public datasets. This paper was presented in the FICC 2018, Singapore. **Title:** *A Generic Multi-modal Dynamic Gesture Recognition System using Machine Learning*
- Have completed a project and submitted a paper on a Gesture Recognition Glove for American Sign Language using accelerometer sensors. An efficiency of 94-96 percent for the glove using 5 accelerometers was achieved. Presented it at Global IoT Summit 2017. **Title:** *Low-Cost Static Gesture Recognition System Using MEMS Accelerometers*

## Projects

**ASU projects** | *Python, Google Cloud Console, Scala, Spark, Postgres* **Jan - April 2021**

- Statistical ML: Did three projects on Logistic Regression & Naive Bayes, Clustering and Convolutional Neural Network.
- Data Visualization: Did a group project on Census Data for Salary Classification using Exploratory data analysis.
- Data processing at Scala: Did a group project on a geo-spatial database and running multiple spatial queries.
- Bio-Inspired AI & Optimization - Was a part of group of 2 and did a project on Wolf pack algorithm.
- Artificial Intelligence - Did the coursework projects on Q-Learning, Search algorithms and PDDL on ROS

**Final year project - Food classification and Ingredients identification: South-Indian cuisine** | *Python* **April 2019**

- Summary - Deep Learning techniques were used to classify South-Indian food data sets (Images). The accuracy graph obtained from the comparison of the different algorithms was used to find the best pre-processing and CNN Architecture for the given data set. Further, Calorie, nutrient and fat values were computed using the ingredients detected from food.

**College Projects** | *Python, Arduino* **2014 - 2019**

- Implemented a product price prediction system for a dataset containing images and meta features of garments.
- Implemented an image captioning system which uses a bi-directional RNN and image net for captioning a given image.
- Recreated a DC motor Control using Arduino and Interrupts.

## Technical Skills

**Languages**: Python, Java, C++, C#, HTML/CSS, JavaScript, SQL, Scala, Postgres, Bash, Arduino, ROS
**Developer Tools**: Keras for Neural Networks, NLTK Tool, Anaconda Packages, Numpy, Scipy, Tweepy, VS Code, Google Cloud Platform, Android Studio, Arduino Platform
**Technologies/Frameworks**: MacOS, Windows, Linux, GitHub, Raspberry-Pi

## Leadership / Extracurricular

- Group Lead - IoT, Solarillion Foundation - Was the group leader for Internet of Things group and worked as teaching assistant for sometime. *2016 - 2017*
- Artist - Theatron, CEG - Acted in multiple plays including plays in Crea-Shakthi and an active member. *2014 - 2017*

# Arizona State University

## Unofficial Transcript

**Name:** **Ajay Kannan**
**Student ID:** **1219387832**

---

Print Date: 05/17/2022
External Degrees
Anna University
Bachelor of Engr 07/01/2019

**Beginning of Graduate Record**

**2021 Spring**

| Course | Description | Attempted | Earned | Grade | Points |
|---|---|---|---|---|---|
| **CSE 511** | **Data Processing at Scale** | 3.000 | 3.000 | A | 12.000 |
| **CSE 575** | **Statistical Machine Learning** | 3.000 | 3.000 | A- | 11.001 |
| **CSE 578** | **Data Visualization** | 3.000 | 3.000 | A- | 11.001 |

| | | | Attempted | Earned | Points |
|---|---|---|---|---|---|
| Term GPA: | 3.78 | Term Totals | 9.000 | 9.000 | 34.002 |
| Cum GPA: | 3.78 | Cum Totals | 9.000 | 9.000 | 34.002 |

**2022 Spring**

| Course | Description | Attempted | Earned | Grade | Points |
|---|---|---|---|---|---|
| **CSE 551** | **Foundations of Algorithms** | 3.000 | 3.000 | A- | 11.001 |
| **CSE 571** | **Artificial Intelligence** | 3.000 | 3.000 | B | 9.000 |
| **CSE 598** | **Special Topics** | 3.000 | 3.000 | A | 12.000 |
| Course Topic: | Bio-inspired AI and Optimizat | | | | |

| | | | Attempted | Earned | Points |
|---|---|---|---|---|---|
| Term GPA: | 3.56 | Term Totals | 9.000 | 9.000 | 32.001 |
| Cum GPA: | 3.67 | Cum Totals | 18.000 | 18.000 | 66.003 |

**2022 Fall**

| Course | Description | Attempted | Earned | Grade | Points |
|---|---|---|---|---|---|
| **CSE 535** | **Mobile Computing** | 3.000 | 0.000 | NR | 0.000 |
| **CSE 573** | **Semantic Web Mining** | 3.000 | 0.000 | NR | 0.000 |
| **CSE 574** | **Planning/Learning Methods AI** | 3.000 | 0.000 | NR | 0.000 |

| | | | Attempted | Earned | Points |
|---|---|---|---|---|---|
| Term GPA: | 0.00 | Term Totals | 0.000 | 0.000 | 0.000 |
| Cum GPA: | 3.67 | Cum Totals | 18.000 | 18.000 | 66.003 |

END OF TRANSCRIPT