

# Drug Response Prediction

# Motivation

Use Machine Learning to predict how a patient or cell line reacts to a drug

# Problem Statement

- Dataset of  $\{(c, d, r)\}$ 
  - Cell line  $c$ , drug  $d$ , response of  $c$  to  $d$  in Log IC50
- Learn function  $f$ 
  - $f: (c, d) \rightarrow r$
- Evaluate performance on test set
  - 5 fold CV

# Methodology

- Learn a representation for cell
- Learn a representation for drug
- Learn joint representation for both
- Alternative method
  - Learn drug-specific cell representations

# Cell Line

- Cell line consists mostly of gene expressions
  - Optionally include copy number, mutation flag
- Each cell line is a set of genes
- Genes have relationships
  - Pathways, Protein-Protein Interactions (PPIs)
- Topologically forms a graph

# Cell Line Representation Ideas

- Set of genes
  - Self-supervised or end-to-end
- Graphs of genes and relationships
- Bio-informed network of genes
  - Architecture design with pathways or PPIs

# Drugs

- Chemical molecules
- String of atoms and bonds





PubChem CID	2244
Structure	<div> 2D</div> <div> 3D</div> <div> Crystal</div>
Primary Hazards	<div> Irritant <a href="#">Laboratory Chemical Safety Summary (LCSS) Datasheet</a></div>
Molecular Formula	<b>C<sub>9</sub>H<sub>8</sub>O<sub>4</sub></b> CH <sub>3</sub> COOC <sub>6</sub> H <sub>4</sub> COOH
Synonyms	aspirin ACETYSALICYLIC ACID 50-78-2 2-Acetoxybenzoic acid 2-(Acetoxy)benzoic acid <a href="#">View More...</a>

Image credit: PubChem for [Aspirin](#)

# Drug Representation Ideas

- Fix drug and represent cell only
- Sequence of atoms and bonds
- Graph of atoms as nodes and bonds as relationships
  - Atom features
    - Atom symbol, hydrogen degree, aromatic flag, ...
- Combination of sequence and graph



# Fusion

- Use drug and cell representation
- Fuse the two
  - Concatenation
  - Add
- Response Prediction Head
- Mean Squared Error optimization
  - L2 norm of prediction and ground truth Log IC50

# Overview and Goals

- 5 fold cross validation
- Data
  - Cancer Cell Line Encyclopedia
  - Genomics of Drug Sensitivity in Cancer
  - PubChem
- No need to include everything
  - Could limit the set of genes to certain pathways, ...

# Incorporating Structure Information into TCR- Epitope Prediction

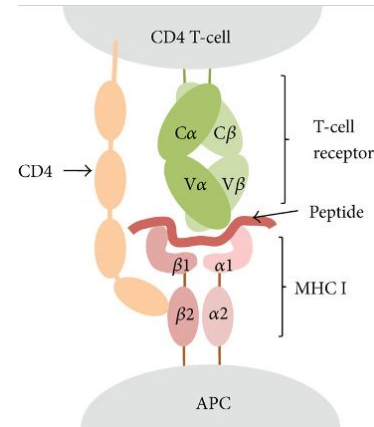
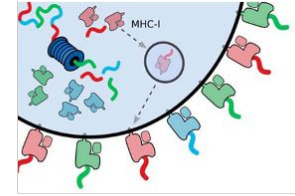
# Background

TCRs: immune system's sensors that recognize specific antigens invade human body

Epitopes: Peptide, a small shard of antigens

MHC molecules: Major Histocompatibility Complex are proteins that display and carry peptide (epitope) fragments on to cell surface and forming pMHC

Binding: a specific recognition and interaction between TCR-pMHC

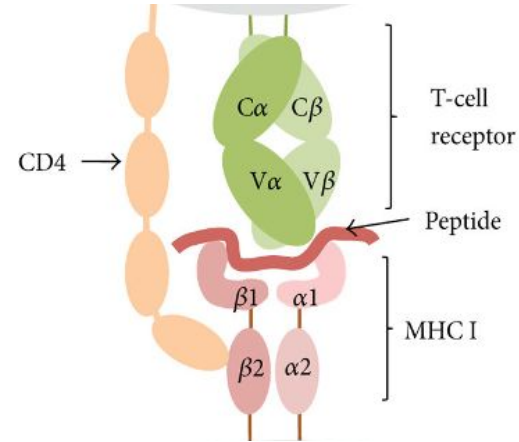


# Problem

Knowing T-cell receptor and epitope could exist/is stable in such (RHS) a system is important for vaccine design, autoimmune disease and cancer immunotherapy and ...

We use binding affinity to reflect the prone of TCR-epitope to format such a protein complex.

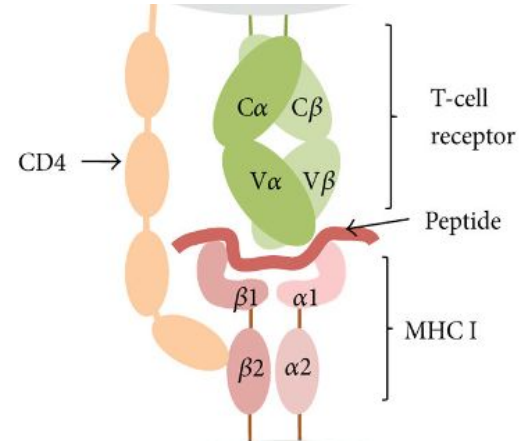
Since the scarce of exact value of binding affinity, researchers quantized it as a **binary classification problem** in computational biology



# Useful details

**TCR:** TCRs are  $\alpha$ ,  $\beta$  heterodimers of about 240–270 amino acids each, with N-terminal Variable (V) domains that determine specificity. Each V domain contains framework regions and three complementarity-determining regions (CDRs), with CDR3- $\beta$  especially critical for binding

**pMHC:** Epitopes are short peptides (8–15 AA) displayed by MHC molecules, forming peptide–MHC (pMHC) complexes. These serve as recognition targets for TCRs and are central to adaptive immunity.



# Past problem abstraction

## CDR3-beta–epitope sequences based binding prediction:

1. Limited paired  $\alpha$ – $\beta$  data, sparse MHC context
2. The high cost of 3D structural determination
3. Researchers show that CDR3 beta chain contributes most to the interaction between TCR-pMHC interactions in most known structures

**We had the following problem definition: next page**

# Sequence based: TCR-epitope binding problem

**Feature Space** Let  $\mathcal{A}$  be the amino acid alphabet ( $|\mathcal{A}| = 20$ ). A CDR3- $\beta$  sequence is  $c \in \mathcal{A}^{L_c}$  with  $8 \leq L_c \leq 20$ , and an epitope is  $e \in \mathcal{A}^{L_e}$  with  $8 \leq L_e \leq 15$ . Input space:

$$\mathcal{X} = \{(c, e)\}.$$

**Labels**

$$y = \mathbf{1}\{c \text{ binds } e\}, \quad y \in \{0, 1\}.$$

**Objective**

$$f_\theta : \mathcal{X} \rightarrow [0, 1], \quad f_\theta(c, e) \approx P(y = 1 \mid c, e).$$



## Availability of high quality TCR-pMHC data

Paired TCR data: high throughput characterization of protein and single cell characterization methods, we have more paired full sequences of TCR data

3D structure data: More crystal structures examined by cryo-electron microscopy and computational determined 3D structures generated by AlphaFold

**We have a chance to restate the TCR-epitope binding problem: next page**

# Structure-enhanced TCR-epitope binding problem

**Feature Space** - TCR: paired  $\alpha$  and  $\beta$  chains

$$c^\alpha \in \mathcal{A}^{L_\alpha}, \quad c^\beta \in \mathcal{A}^{L_\beta}.$$

- pMHC: epitope  $e \in \mathcal{A}^{L_e}$  and MHC  $m \in \mathcal{A}^{L_m}$ . - Auxiliary 3D coordinates (experimental or predicted):

$$X_T = \{x_i\}_{i=1}^{N_T}, \quad X_M = \{z_j\}_{j=1}^{N_M}, \quad x_i, z_j \in \mathbb{R}^3.$$

Input space:

$$\mathcal{X}_{\text{aug}} = \{(c^\alpha, c^\beta, e, m, X_T, X_M)\}.$$

**Labels**

$$y = \mathbf{1}\{\text{TCR binds pMHC}\}, \quad y \in \{0, 1\}.$$

**Objective**

$$f_\theta : \mathcal{X}_{\text{aug}} \rightarrow [0, 1], \quad f_\theta(\cdot) \approx P(y = 1 \mid c^\alpha, c^\beta, e, m, X_T, X_M).$$

# Our goal

Construct a pipeline that contains 3D information including experimentally and computationally determined structures into TCR-epitope binding prediction

Hints:

1. Experimental determined structures is very limited, and computationally determined structures might be full of noise. Training an end2end pure structure-based model might be very challenging.
2. AlphaFold has other output generated along with coordination of each atoms. For example, Predicted aligned error (PAE) is generated with each atom positions representing AF2's confidence about this prediction.
3. Learning from previous literatures, using a subset of 3D structure info could be helpful. For example, only take CDR3 and epitope 3D structure from whole structure is a potentially a good chose.

# Other useful information

PDB / TCR3d

AlphaFold2 / Multimer

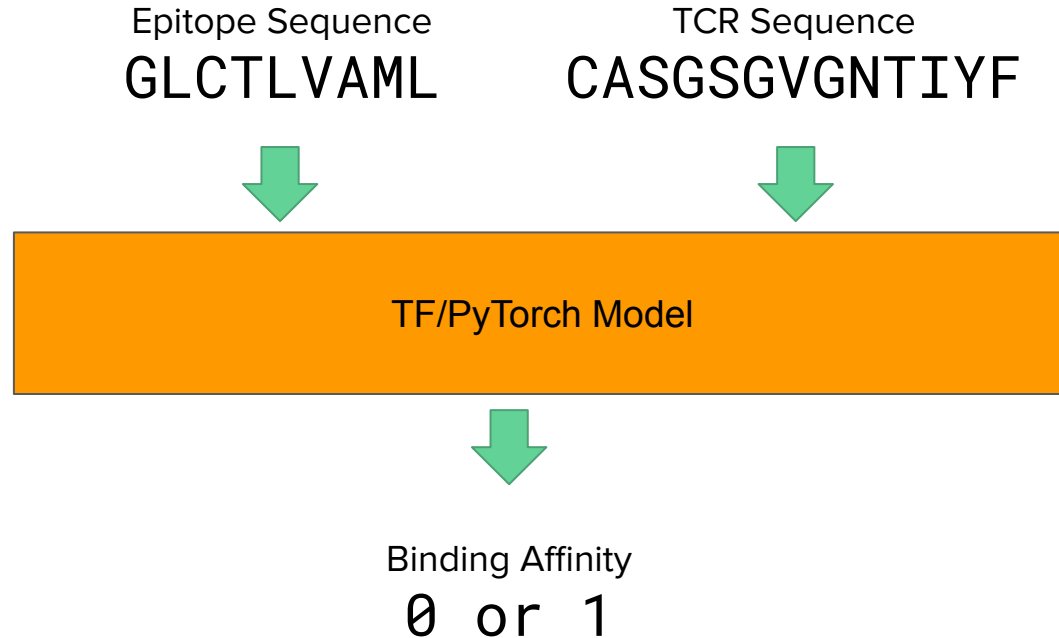
TCRdock / TCRfold

See project description materials.

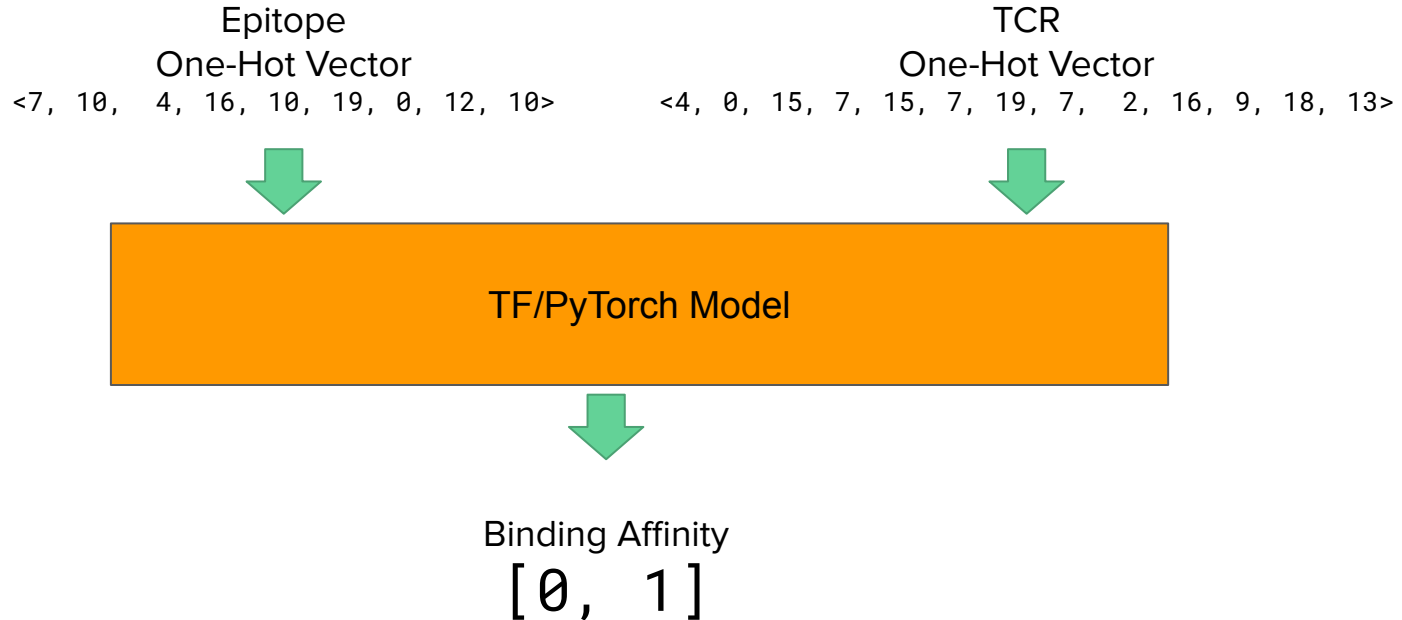
# Q& A

# TCR Embedding Model - Data Selection

# Inputs and Outputs



# Inputs and Outputs\*



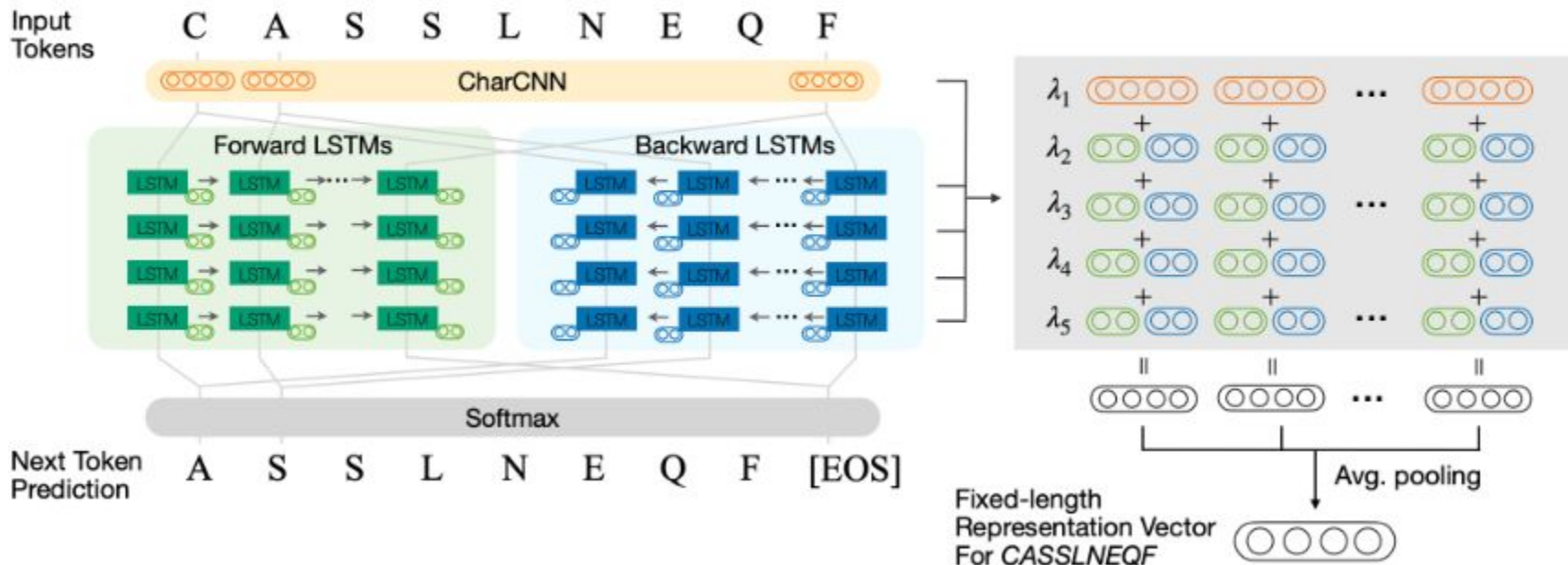


# An embedding model of interest

## catELMo

Amino acid sequence representation model

a



# Questions

- **Scaling of training data:** With random selection of pretraining data, how much data is required to achieve good downstream prediction performance, and at what point does performance plateau as the dataset grows?
- **Data selection strategies:** Beyond random selection, can we design strategies to select more informative subsets of TCR sequences such that the resulting embedding model achieves comparable performance with fewer training examples?

# What to expect?

- **Scaling of training data:** You will need to train the embedding model with various data size, and then test its performance on TCR-epitope binding prediction task.
- **Data selection strategies:** You will need to come up with strategies to do select most informative sequences to get similar performance as random selection?

# Resources

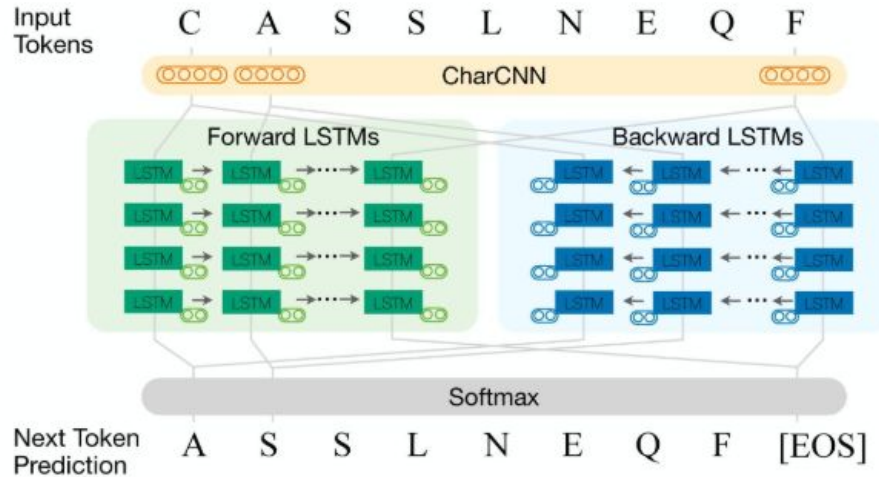
- Refer to this first.
  - Code & Data <https://github.com/Lee-CBG/catELMo>
  - Paper. <https://elifesciences.org/reviewed-preprints/88837>
- Detailed Data and code to be shared. TA will make an announcement about this.

## Other notes

- No restrictions on prediction model structures. But we suggest keep it consistent for fair comparison of your results.

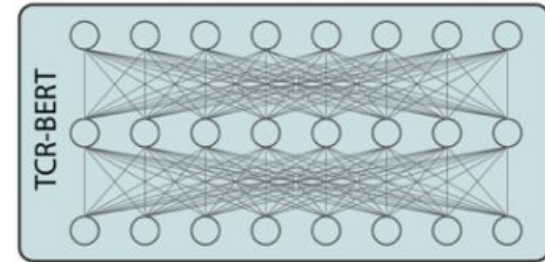
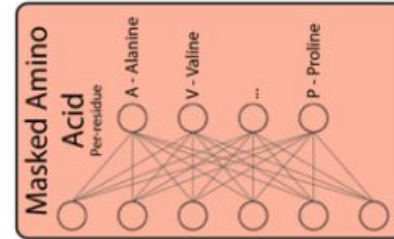
# Comparison of TCR Embedding Models with varying vocabularies (k-length peptides)

# Popular LM based TCR embeddings



## 1) Masked amino acid

88,403 TRA/TRB sequences  
VDJdb + PIRD datasets



Input [CLS] [C] [A] [S] [.] [V] [F] [SEP]

<https://elifesciences.org/reviewed-preprints/88837>

<https://proceedings.mlr.press/v240/wu24b/wu24b.pdf>

# Questions and expectations

- Whether alternative vocabulary definitions could lead to more informative embeddings?
  - **k-mer vocabularies:** Instead of using individual amino acids, what happens if we build the vocabulary from k-length peptides (subsequences)? You will retrain embedding models using different k values and study their impact on downstream TCR–epitope prediction performance.
  - **Dynamic vocabularies (optional):** Can you design methods to dynamically adjust the tokenization with biological implications?
- We follow the same pipeline as the previous project
  - First learn an embedding model
  - Then monitor prediction model's performance



# Resources


You are encouraged to explore models beyond those listed.

- TCRBert <https://github.com/wukevin/tcr-bert>
- catELMo <https://github.com/Lee-CBG/catELMo>
- Detailed model and data to be announced.

# Computing resources – SOL

- [sol.asu.edu](https://sol.asu.edu)
- You should request a sol account. Free GPUs and computing hours offered by ASU.
- <https://github.com/pzhang84/CodexCommand/blob/main/sol.md>

# Q & A



**To ace the project(s), you need to write code** (python recommended) **to work with data**

I need a Machine? and package manager?

**sol** (asu proprietary; [requires access](#)): [quick setup guide](#)

sol support channel (by admins of asu sol): <https://app.slack.com/client/EBY1XTCCR/CMTPR329M>

**Mamba** (open-source; sol mandates this, no other package manager): [docs](#)

mamba is a tool for managing packages and environments. Languages it supports: Python, C / C++, Rust, Go, JavaScript, Ruby, Deno, Zig, and more

# Demo of a model

## a. Move your data and model to sol:

- data generally hosted on [github repo](#), or [zenodo.org](#)
- Steps
  1. Download data and code to your laptop.
  2. Upload your files to sol: login into ASU network using Cisco Secure Login (if off-campus)
    - a. (recommended) Use GUI: login to [sol.asu.edu](#) -> hover on Files -> click on Home Directory -> upload files from local computer
    - b. Use local terminal to [ssh into sol.asu.edu](#) and use a linux command (scp) to transfer files
      - i. from local to [sol.asu.edu](#):

```
scp local_file asurite@sol.asu.edu:/path/on/sol
```
      - ii. from [sol.asu.edu](#) to local:

```
scp asurite@sol.asu.edu:/path/on/sol local_file
```

## b. Use the terminal using bash commands like: scp or filezilla or winscp to transfer files.

- Run your code on sol
  1. Interactive mode (recommended; synchronous; request for a VS Code Tunnel)
  2. Sbatch yourfile.sh (asynchronous; submit jobs on server)
- Submit your jobs as soon as possible, as it could be crowded at the end of semester and waiting time could be long.



# Tips?

1. Time management:
  - a. Set up environment (40% of time)  
Train models (40% of time)  
Documentation (20% of time)
2. Keep an eye out for your fairshare score: essentially gets you ahead in the queue for batch jobs



**Questions?**