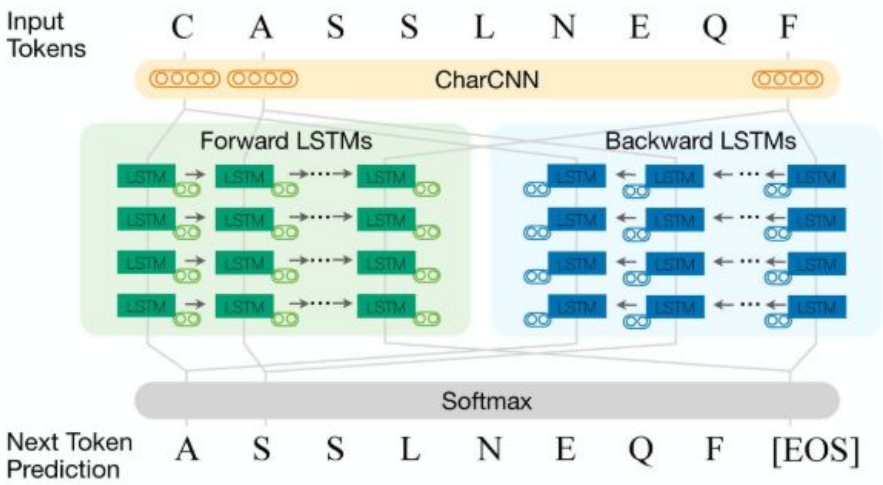
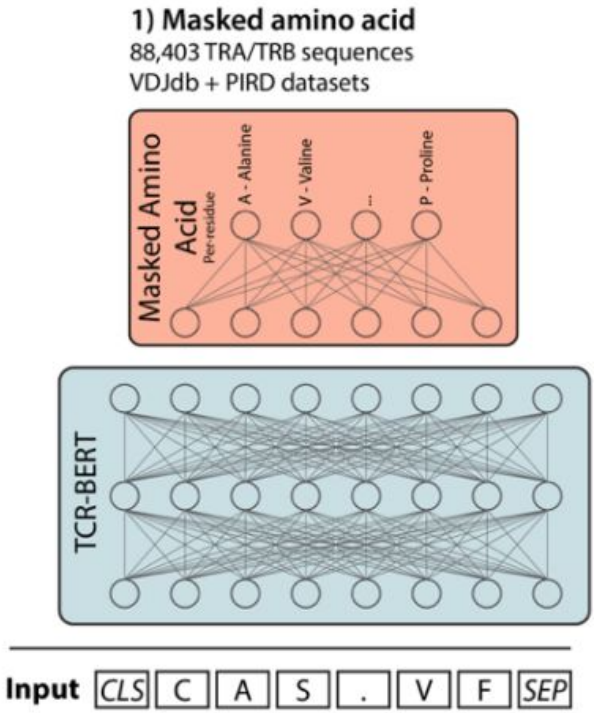
Comparison of TCR Embedding Models with varying vocabularies (k-length peptides)

Popular LM based TCR embeddings





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
- 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

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

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.

Questions?