```
from google.colab import drive
drive.mount('/content/drive')
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client id=947318989803-
6bn6qk8qdqf4n4q3pfee6491hc0brc4i.apps.qooqleusercontent.com&redirect uri=urn%3aietf%3awg%3aoauth%3a2.0%3a
esponse type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.g
apis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2
googleapis.com%2fauth%2fpeopleapi.readonly
Enter your authorization code:
Mounted at /content/drive
                                                                                                      In [2]:
!pip3 install torch==1.2.0+cu92 torchvision==0.4.0+cu92 -f https://download.pytorch.org/whl/torch stable
Looking in links: https://download.pytorch.org/whl/torch stable.html
Collecting torch==1.2.0+cu92
 Downloading https://download.pytorch.org/whl/cu92/torch-1.2.0%2Bcu92-cp36-cp36m-manylinux1 x86 64.whl
(663.1MB)
                                  | 663.1MB 23kB/s
Collecting torchvision==0.4.0+cu92
 Downloading https://download.pytorch.org/whl/cu92/torchvision-0.4.0%2Bcu92-cp36-cp36m-
manylinux1 x86 64.whl (8.8MB)
                                      | 8.8MB 31.7MB/s
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from torch==1.2.0+cu92) (
1.18.5)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from
torchvision==0.4.0+cu92) (1.12.0)
Requirement already satisfied: pillow>=4.1.1 in /usr/local/lib/python3.6/dist-packages (from torchvision=
=0.4.0+cu92) (7.0.0)
Installing collected packages: torch, torchvision
 Found existing installation: torch 1.5.0+cu101
    Uninstalling torch-1.5.0+cu101:
      Successfully uninstalled torch-1.5.0+cu101
  Found existing installation: torchvision 0.6.0+cu101
    Uninstalling torchvision-0.6.0+cu101:
      Successfully uninstalled torchvision-0.6.0+cu101
Successfully installed torch-1.2.0+cu92 torchvision-0.4.0+cu92
4
                                                                                                        In [0]:
import os
os.chdir('/content/drive/My Drive/Peptide MHC Project')
                                                                                                      In [0]:
import torch
import torch.nn as nn
from architecture import *
import pandas as pd
import numpy as np
print(torch. version
1.2.0 + cu 92
                                                                                                      In [0]:
# Loads Coronavirus Peptide Sequences
# Full data (in all) has too many peptides of length 11-15
corona embedding tensor = torch.load('coronavirus/subset corona embeddings.pt').cuda()
print(corona embedding_tensor.shape)
corona peptides = pd.read csv("coronavirus/subset corona peptides.txt", header=None)
corona peptides.columns = ['peptide']
# Loads MHC Allele Sequences
alpha_140_179_embedding_tensor = torch.load('embeddings/common_140_179_embeddings.pt')
alpha 50 84 embedding tensor = torch.load('embeddings/common 50 84 embeddings.pt')
allele_embedding_tensor = torch.cat((alpha_140_179_embedding_tensor, alpha_50_84_embedding_tensor), dim=1
print(allele_embedding_tensor.shape)
alleles = pd.read csv("allele sequences/30 mhc alleles.csv")
torch.Size([625, 15, 100])
torch.Size([30, 106, 100])
                                                                                                      In [0]:
# Defines model
peptide embedding dim = 100
```

peptide hidden dim = 100

```
allele embedding dim = 100
allele_hidden_dim = 100
peptide to output dim = peptide hidden dim*2*2
allele to output dim = allele hidden dim*2*2
peptide model = peptide BiGRU(peptide embedding dim, peptide hidden dim).cuda()
allele model = allele BiGRU(allele embedding dim, allele hidden dim).cuda()
output model = Output Layer (peptide model, allele model, peptide to output dim, allele to output dim).cud
output_model.load_state_dict(torch.load('/content/drive/My Drive/Peptide MHC Project/Trained Models/train
                                                                                                       Out[0]:
<All keys matched successfully>
                                                                                                        In [0]:
affinities = []
for i in range(corona_peptides.shape[0]):
  peptide affinities = []
  for j in range(alleles.shape[0]):
    peptide = torch.reshape(corona_embedding_tensor[i, :, :], (1, 15, 100))
    allele = torch.reshape(allele embedding tensor[j, :, :], (1, 106, 100))
    binding pred = output model(peptide, allele).item()
    peptide affinities.append(binding pred)
  affinities.append(peptide_affinities)
  if i % 100 == 0:
    print(i)
0
100
200
300
400
500
600
                                                                                                        In [0]:
n alleles = 30
all affs = []
HLA A affs = []
HLA_B_affs = []
HLA C affs = []
[all affs.extend(el) for el in affinities]
[HLA A affs.extend(el[:10]) for el in affinities]
[HLA B affs.extend(el[10:20]) for el in affinities]
[HLA_C_affs.extend(el[20:]) for el in affinities]
avg aff = sum(all affs) / len(all affs)
meanHLA A = sum(HLA A affs) / len(HLA A affs)
meanHLA B = sum(HLA B affs) / len(HLA B affs)
meanHLA C = sum(HLA C affs) / len(HLA C affs)
print ("Mean Binding Affinity between peptides and 30 MHC-I alleles:", round (avg aff, 3))
print("Mean Binding Affinity between peptides and 10 MHC-I alleles with gene:")
print("- HLA-A: {}".format(round(meanHLA A, 3)))
print("- HLA-B: {}".format(round(meanHLA B, 3)))
print("- HLA-C: {}\n".format(round(meanHLA C, 3)))
Mean Binding Affinity between peptides and 30 MHC-I alleles: 0.714
Mean Binding Affinity between peptides and 10 MHC-I alleles with gene:
- HLA-A: 0.549
- HLA-B: 0.706
- HLA-C: 0.888
                                                                                                        In [0]:
# Shows binding affinities by percentile
x = sorted(all affs)
print("25th percentile:", round(x[int(len(x)\star.25)], 3))
print("50th percentile:", round(x[int(len(x)*.5)], 3))
print("75th percentile: f\n" % round(x[int(len(x)*.75)], 3))
# Almost all peptides will certainly bind to at least 1 MHC allele
n peptides = len(affinities)
near_one_all = []
near zero all = []
either all = []
for pep_affs in affinities:
  near one = []
  near zero = []
```

```
either = []
  for aff in pep_affs:
    near one.append(aff >= 0.9)
    near zero.append(aff <= 0.1)</pre>
    either.append(aff >= 0.9 or aff <= 0.1)
  near one all.append(near one)
  near zero all.append(near zero)
  either all.append(either)
pct_near_one = sum([sum(near_one) / n_alleles for near_one in near_one_all]) / n_peptides
pct_near_zero = sum([sum(near_zero) / n_alleles for near_zero in near_zero_all]) / n_peptides
pct either = sum([sum(either) / n alleles for either in either all]) / n peptides
print("Percentage of binding affinities over 0.9: {}%".format(round(pct_near_one*100, 1)))
print("Percentage of binding affinities under 0.1: {}%".format(round(pct near zero*100, 1)))
print("Either (under 0.1 or over 0.9): {}%".format(round(pct either*100, 1)))
pctHLA A = sum([sum(in range[:10]) / 10 for in range in near one all]) / n peptides
pctHLA B = sum([sum(in range[10:20]) / 10 for in range in near one all]) / n peptides
pctHLA C = sum([sum(in range[20:]) / 10 for in range in near one all]) / n peptides
print("Percent over 0.9, by gene:")
print("- HLA-A: {}%".format(round(pctHLA A*100, 1)))
print("- HLA-B: {}%".format(round(pctHLA B*100, 1)))
print("- HLA-C: {}%".format(round(pctHLA_C*100, 1)))
25th percentile: 0.29
50th percentile: 0.997
75th percentile: 1.000000
Percentage of binding affinities over 0.9: 63.4%
Percentage of binding affinities under 0.1: 21.0%
Either (under 0.1 or over 0.9): 84.4%
Percent over 0.9, by gene:
- HLA-A: 44.7%
- HLA-B: 61.8%
- HLA-C: 83.8%
                                                                                                       In [0]:
# Finds average affinity by protein
# Extracts each protein's amino acid sequence
corona = pd.read_csv("coronavirus_netchop.csv")
aa_indices = corona.index[corona['pos'] == 1].tolist()
n proteins = len(aa indices)
protein list = []
for i in range(n_proteins):
  if i < len(aa indices) - 1:</pre>
    end index = aa indices[i + 1]
  else:
    end index = corona.shape[0]
  protein = "".join(corona['AA'].iloc[aa_indices[i]:end_index].to_list())
  protein list.append(protein)
# Finds which proteins each peptide is in
protein affinities = [[] for in range(n proteins)]
for i in range(n peptides):
  peptide = corona_peptides.peptide.iloc[i]
  for j in range(n proteins):
    if peptide in protein_list[j]:
      protein_affinities[j].append(affinities[i])
# Calculates average binding affinity per protein
print("Number of proteins:", n proteins)
for i in range(n proteins):
  count = len(protein affinities[i])
  if count == 0:
    continue
  avg aff = sum([sum(pep affs) / len(pep affs) for pep affs in protein affinities[i]]) / count
  print("Mean Binding Affinity between 30 MHC-I alleles and the {} peptides found in protein {}: {}"
         .format(count, i+1, round(avg aff, 3)))
```

```
Number of proteins: 14
Mean Binding Affinity between 30 MHC-I alleles and the 14 peptides found in protein 2: 0.75
Mean Binding Affinity between 30 MHC-I alleles and the 25 peptides found in protein 3: 0.83
Mean Binding Affinity between 30 MHC-I alleles and the 4 peptides found in protein 4: 0.728
Mean Binding Affinity between 30 MHC-I alleles and the 10 peptides found in protein 5: 0.68
Mean Binding Affinity between 30 MHC-I alleles and the 4 peptides found in protein 7: 0.669
Mean Binding Affinity between 30 MHC-I alleles and the 1 peptides found in protein 8: 0.451
Mean Binding Affinity between 30 MHC-I alleles and the 447 peptides found in protein 9: 0.712
Mean Binding Affinity between 30 MHC-I alleles and the 287 peptides found in protein 10: 0.696
Mean Binding Affinity between 30 MHC-I alleles and the 98 peptides found in protein 11: 0.711
Mean Binding Affinity between 30 MHC-I alleles and the 4 peptides found in protein 12: 0.814
Mean Binding Affinity between 30 MHC-I alleles and the 16 peptides found in protein 13: 0.612
Mean Binding Affinity between 30 MHC-I alleles and the 2 peptides found in protein 14: 0.721
                                                                                                      In [0]:
# Protein 11 is the Spike Protein
count = len(protein affinities[10])
spike avg = sum([sum(pep affs) / len(pep affs) for pep affs in protein affinities[10]]) / count
print("Spike protein (11) mean binding affinity: {}".format(round(spike avg, 3)))
\label{eq:meanHLA_A = sum([sum(pep_affs[:10]) / 10 for pep_affs in protein_affinities[10]]) / count} \\
meanHLA B = sum([sum(pep affs[10:20]) / 10 for pep affs in protein affinities[10]]) / count
meanHLA_C = sum([sum(pep_affs[20:]) / 10 for pep_affs in protein_affinities[10]]) / count
print("By gene:")
print("- HLA-A: {}".format(round(meanHLA A, 3)))
print("- HLA-B: {}".format(round(meanHLA_B, 3)))
print("- HLA-C: {}".format(round(meanHLA C, 3)))
Spike protein (11) mean binding affinity: 0.711
By gene:
- HLA-A: 0.518
- HLA-B: 0.731
- HLA-C: 0.885
                                                                                                      In [0]:
# Breaks peptides apart by length
length affinities = [[] for _ in range(8)]
for i in range(n peptides):
  peptide = corona peptides.peptide.iloc[i]
  length affinities[len(peptide)-8].append(affinities[i])
# Calculates mean binding affinity by peptide length
for i in range(8):
  all = []
  [all.extend(peps) for peps in length_affinities[i]]
  avg aff = sum(all) / len(all)
  print("{} peptides of length {}".format(len(length affinities[i]), i+8))
  print("- Mean binding affinity:", round(avg_aff, 2))
  sorted affs = sorted(all)
  count = len(sorted affs)
  first_idx = [aff >= 0.9 for aff in sorted_affs].index(True)
  pct over = round((count - first idx) / count * 100)
  print("- Percentage of binding affinities above 0.9: {}%".format(pct over))
100 peptides of length 8
- Mean binding affinity: 0.82
- Percentage of binding affinities above 0.9: 76%
250 peptides of length 9
- Mean binding affinity: 0.57
- Percentage of binding affinities above 0.9: 45%
150 peptides of length 10
- Mean binding affinity: 0.7
- Percentage of binding affinities above 0.9: 61%
25 peptides of length 11
- Mean binding affinity: 0.89
- Percentage of binding affinities above 0.9: 86%
25 peptides of length 12
- Mean binding affinity: 0.94
- Percentage of binding affinities above 0.9: 90%
25 peptides of length 13
- Mean binding affinity: 0.98
- Percentage of binding affinities above 0.9: 96%
25 peptides of length 14
- Mean binding affinity: 0.98
- Percentage of binding affinities above 0.9: 97%
25 peptides of length 15
- Mean binding affinity: 0.94
- Percentage of binding affinities above 0.9: 93%
```

In [0]: