Workshop on Computational Immunobiology

Aly A. Khan[1], Evan Kiefl[2]
   [1]Course Instructor; aakhan@uchicago.edu
   [2]Course Assistant; ekiefl@uchicago.edu

THERE'S A JOKE ABOUT IMMUNOLOGY, *which Jessica Metcalf of Princeton recently told me. An immunologist and a cardiologist are kidnapped. The kidnappers threaten to shoot one of them, but promise to spare whoever has made the greater contribution to humanity. The cardiologist says, "Well, I've identified drugs that have saved the lives of millions of people." Impressed, the kidnappers turn to the immunologist. "What have you done?" they ask. The immunologist says, "The thing is, the immune system is very complicated ... " And the cardiologist says, "Just shoot me now."* — Yong, Ed. "Immunology Is Where Intuition Goes to Die", The Atlantic, Aug. 2020.

WELCOME. This workshop will introduce a few useful concepts for computationally thinking about the immune system. The field of immunology is currently in the midst of a data-driven revolution, where advances in computer science and high-throughput technologies are allowing us to approach questions that previously could not be answered using purely experimental approaches or standard reductionist techniques. In this workshop we will explore two concepts:

1. What are some ways to characterize the phenotypic diversity and population structure of immune cells?

2. How can we harness statistical modeling to make useful predictions about the immune system?

BSD courses: Other courses that may broaden your experience in quantitative and computational immunology include: Microbial 'Omics (BIOS 25420) and Quantitative Immunobiology (IMMU 34800).

WHAT IS THE IMMUNE SYSTEM? The immune system can be thought of as a loosely connected network of cells that interact to solve problems beyond the individual capacities of each single cell. At the same time, the immune system contains the capacity to exchange messages between

cells, collectively coordinate action, and retain memory of past events. The field of immunology has developed a rich and specialized body of knowledge detailing the shared biological mechanisms underlying host defense, transplantation, autoimmunity, tumor immunology, allergy and other clinical challenges.

The role of the immune system in host defense is often framed by an attacked host (e.g. self) against alien invaders (e.g. non-self; viruses, bacteria, cancer cells, etc). This conflict comes with serious consequences. The immune system must precisely function to distinguish between self and non-self, as collateral damage and misfiring can lead to serious harm (e.g. autoimmunity, allergy, etc). Strikingly, we still do not fully understand how the immune system accomplishes this exquisite role or why certain individuals develop autoimmunity.

As you progress in your graduate training, we hope you take note that there are a myriad of exciting opportunities available to biologists willing to cross and explore the traditional boundaries between immunology and computer science.

1. Visualizing the population structure of immune cells

This section is adapted from Neu et al., Trends in Immunology, 2017.

IN THE PAST DECADE advances in computing and next-generation sequencing technologies have ushered in a new era of discovery in immunology. In particular, single cell RNA-seq (scRNA-seq) has enabled an unprecedented view of gene expression in single cells. A key challenge lies in visualizing single cell gene expression data in a biologically meaningful way while remaining robust to the high levels of noise that is present in single cell data.

scRNA-seq is RNA-seq performed on an individual cell. The cellular mRNA is amplified through oligos specific for the 5′ or 3′ tail of mRNA molecules or random hexamers.

One of the most compelling applications of single-cell genomics to immunology resides in characterizing the population structure of single cells. Visualization of scRNA-seq data can help to identify rare and intermediate subpopulations that are often overlooked with bulk

RNA-seq data. The goal of visualization algorithms is to project high-dimensional data into a low-dimensional space, thus resolving cellular groups based on their transcriptional similarity without any predetermined markers dictating their identity. In this section, we review two standard dimensionality reduction algorithms used to visualize scRNA-seq data.

Exercise 1.1 — Data wrangling

Let's begin by downloading some scRNA-seq data. For this exercise we will be using data published as part of a study examining certain B cells in humans after influenza vaccination (Neu et al., JCI, 2019). We will download pre-processed supplementary data from the GEO database: `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE116500`. Scroll to the bottom of the page to locate the supplementary data.

Supplementary file	Size	Download	File type/resource
GSE116500_Limma_adj_4.csv.gz	7.2 Mb	(ftp)(http)	CSV

Once the supplementary data is downloaded, you can uncompress the file. The resulting csv file contains single cell gene expression data for nearly 300 B cells. Let's analyze the data using R:

```
# load data
geo <- read.csv('./GSE116500_Limma_adj_4.csv',
    row.names = 1, header = TRUE)
```

How many genes and cells are in the data? Let's try looking at the dimensions of the data:

```
# number of rows (Genes) and columns (cells)
dim(geo)
num_genes <- dim(geo)[1]
num_cells <- dim(geo)[2]
```

To made things easier for subsequent steps, let's take the transpose of the data so that each row denotes a cell and

Dimensionality reduction is a process to reduce the number of variables to a compressed set of principal variables. More specifically, dimensionality reduction can be understood as projecting the data from the original high-dimensional space into much lower-dimensional space, while (roughly) capturing the concerned statistical properties (e.g., variation, distribution) and/or structure property (e.g., clusters). High-dimensional data after dimensionality reduction are easier to store and faster for downstream computation. Moreover, when projecting the data to two- or three-dimensional space, it is also easier for visualization.

GEO is a database managed by the NIH and functions as a public repository of high throughput sequencing and microarray data.

Figure 1: The supplementary data can be found at the bottom of the page, and has has already been pre-processed.

each column denotes a gene. The transpose of a matrix is an operation which flips a matrix over its diagonal; that is, it switches the row and column indices. This can facilitate certain types of linear algebraic operations and calculations which operate on columns by default.

```
# current data with genes by cells
dim(geo)

# transpose data to cells by genes
geo <- t(geo)
dim(geo)
```

Our assumption is that most gene expression in scRNA-seq data contains random noise due to technical variation. We aim to focus our analyses on genes with high variability, which may have a biological basis. For the purposes of this exercise we simply calculate variance of gene expression as a way to rank genes, but we note other methods (e.g. coefficient of variation) can be used as well.

```
# Let's identify highly variable genes based on
# variance.
SDs <- apply(geo, 2, var)

# we usually pick the top 1000 - 2000 highly
#   variable genes (HVGs)
hvgs <- names(sort(SDs, decreasing = TRUE))[1:2000]
geo.hvg <- as.data.frame(geo[,hvgs])
dim(geo.hvg)
```

We should now have a matrix of 295 cells with the top 2000 most highly variable genes. One fact that we did not reveal earlier is that these cells are plasmablasts, which are B cells that secrete antibodies. We will now classify the type (or isotype) of antibodies these B cells are secreting by comparing the expression of IgA and IgG genes. We will classify each cell with their most highly expressed isotype:

Immunoglobulin isotype can be thought of as a molecular classification for antibodies.

```r
# There are multiple genes that encode subclasses
# of the two isotypes in this data: IgA and IgG

# Grab max IgA values across all genes
IgA <- cbind(geo.hvg$IGHA1,geo.hvg$IGHA2)
IgA_max <- as.matrix(apply( IgA, 1, max))
#Set column name to 'IgA'
colnames(IgA_max) <- 'IgA'

# Grab max IgG values across all genes
IgG <- cbind(geo.hvg$IGHG1,geo.hvg$IGHG2,
    geo.hvg$IGHG3,geo.hvg$IGHG4)
IgG_max <- as.matrix(apply( IgG, 1, max))
#Set column name to 'IgG'
colnames(IgG_max) <- 'IgG'

#Determine if IgA is higher or IgG is higher
Ig <- cbind(IgA_max,IgG_max)
Ig_max <- colnames(Ig)[(apply( Ig, 1, which.max))]
```

We have now classified cells as IgA or IgG expressing
plasmablasts.

Exercise 1.2 — Data Visualization

PRINCIPAL COMPONENT ANALYSIS (PCA) is a linear
dimensionality reduction algorithm that is often the first-
step in visualizing high-dimensional data. PCA takes
an input of correlations between cells based on gene ex-
pression data, and identifies principal components corre-
sponding to linear combinations of genes, which cumu-
latively capture the variability of the total dataset. When
the data is projected against the first few components,
which account for the largest amount of variation, distinct
populations can be visually and biologically interpreted.
Let's use the top two principal components to visualize
our data:

PCA is a linear dimensionality reduc-
tion algorithm, used to project high
dimension data into a few 'components'
that capture most of the variability in
the data. It is a popular visualization
technique that can help identify pat-
terns or connections between samples.

```r
# Let's perform PCA
geo.pca <- prcomp(geo.hvg, center = TRUE,
    scale = TRUE)
plot_pc_data <- data.frame(
    PC1=geo.pca[["x"]][,'PC1'],
    PC2=geo.pca[["x"]][,'PC2'])
```

In order to visualize our scRNA-seq data using PCA, we will need to use some functions from `ggplot2`. Let's load the R package:

```r
library("ggplot2")
```

As a general rule of thumb, if you get an error message saying there is no package titled `ggplot2` you may need to first install the appropriate package:

```r
install.packages("ggplot2")
```

Let's visualize our data:

```r
#plot PCA results with Ig status
ggplot(plot_pc_data, aes(x=geo.pca[["x"]][,'PC1'],
    y=geo.pca[["x"]][,'PC2'], color=Ig_max)) +
    geom_point(shape=1) + theme_minimal() +
    geom_point(aes(color = Ig_max)) +
    theme(legend.position = "top")
```
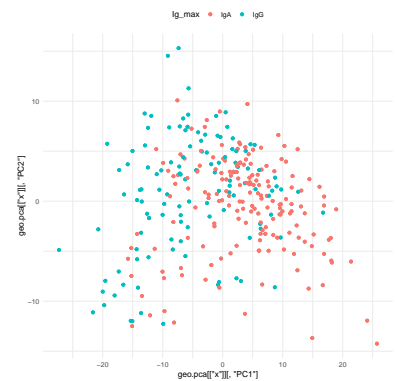
Do IgA cells and IgG cells separate well using PCA?

T-DISTRIBUTED STOCHASTIC NEIGHBOR EMBEDDING (T-SNE) is a widely used nonlinear dimensionality reduction algorithm. Unlike PCA, which seeks to capture variance in data, t-SNE seeks to explicitly preserve the local structure of the original data. t-SNE constructs a probability distribution to describe the data set such that pairs of similar cells are assigned a high probability, while dissimilar pairs are assigned a much smaller probability. Thus, cells that are similar in the high-dimensional space will cluster together (due to high probability) in

Figure 2: Visualizing our scRNA-seq data with PCA.

t-SNE is a nonlinear dimensionality reduction method, which seeks to preserve the local structure of data in high-dimensional space when projected into low-dimensional space.

low-dimensional space. This ability to explicitly maintain clustering of similar cells is an advantage of t-SNE over direct linear transformation such as PCA. This approach is very effective with scRNA-seq data, and has been used to resolve transcriptionally distinct populations that are indistinguishable with PCA. We will need to use some functions from `Rtsne`, so let's load the R package:

```
library("Rtsne")
```

If you get an error message saying there is no package titled `Rtsne` you may need to first install the package.

```
# Let's use the top 10 PC for t-SNE
geo.tsne <- Rtsne(geo.hvg, theta = .001,
        perplexity = 30, initial_dims = 10)
plot_tsne_data <- data.frame(
    tsne1=geo.tsne$Y[,1],
    tsne2=geo.tsne$Y[,2])
```

Let's visualize our data:

```
#plot PCA results with Ig status
ggplot(plot_tsne_data, aes(x=geo.tsne$Y[,1],
    y=geo.tsne$Y[,2], color=Ig_max)) +
    geom_point(shape=1) + theme_minimal() +
    geom_point(aes(color = Ig_max)) +
    theme(legend.position = "top")
```
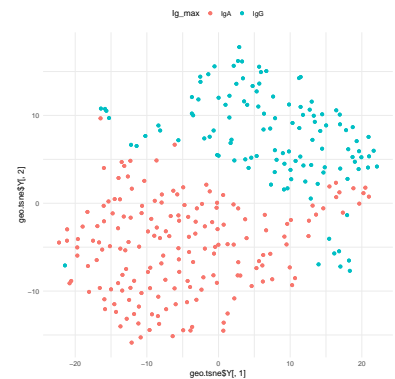
Do IgA cells and IgG cells separate well using t-SNE?

Discussion

While single cell transcriptional profiles have high dimensionality due to the thousands of genes profiled, their intrinsic dimensionalities are typically much lower. Thus, unsupervised low dimensional projections can reveal salient structure in large-scale scRNA-seq datasets. However, the choice of dimensionality reduction algorithms used to visualize and to infer cell similarity needs careful thought for immunology applications.

Figure 3: Visualizing our scRNA-seq data with t-SNE.

2. Peptide-MHC interactions

Nearly every cell in our body is decorated with a class of molecules known as major histocompatibility complex (MHC). The function of the MHC is to bind peptide fragments (antigens) derived from pathogens and display them on the cell surface for recognition by components of the immune system. In particular, alien peptides may be recognized by cytotoxic T cells, which can destroy infected cells. Understanding the binding affinity of MHC proteins and the repertoire of cognate peptide ligands is important for advancing our understanding of the antigenic landscape in infectious diseases, autoimmunity, vaccine design, and cancer immunotherapy.
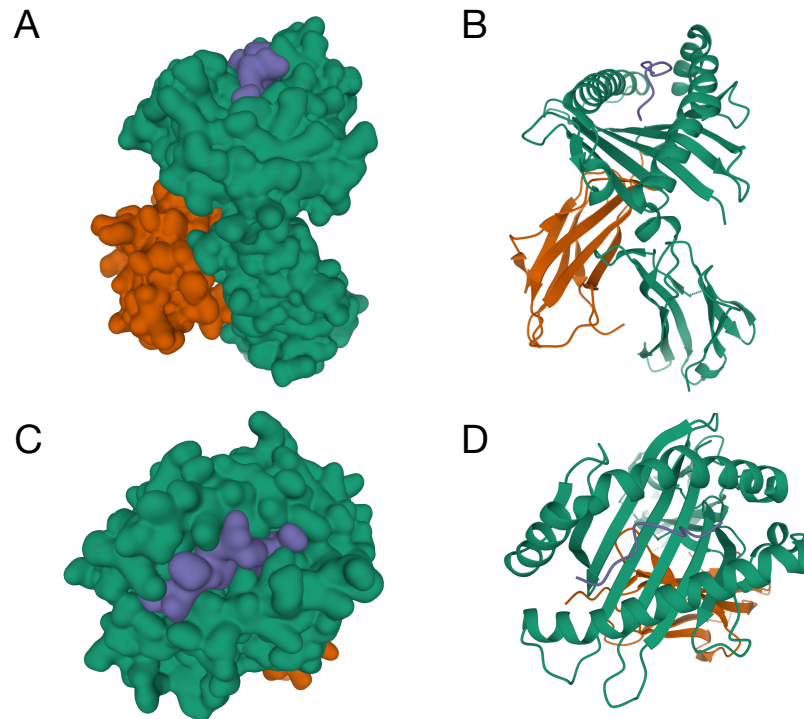

Figure 4: Structure of a peptide-MHC complex. (A) Front view of the crystal structure 1I4F depicting a space-filled molecular surface representation of the MHC (HLA-A*02:01) protein bound with a peptide (derived from MAGEA4, in purple), and (B) a backbone ribbon representation of the MHC. The heavy chain of MHC (alpha), which contains the binding cleft, is depicted in green. The supporting light chain (Beta-2 Microglobulin) is depicted in brown. (C,D) Top view of the complex showing the surface of the MHC (green) and the exposed surface of the bound peptide (purple). The exposed part of the peptide is referred to as the "TCR-interacting surface" of the peptide-MHC complex.

Due to the importance of this process, peptide-MHC binding has been experimentally studied in a variety of ways. For example, the relative binding ability of different peptides to a specific MHC molecule can be directly assessed by competition experiments. The result of such

experiments is a set of relative binding energies for different MHC-peptide combinations. In this section, we will model the repertoire of peptide ligands associated with a specific MHC molecule.

Exercise 2.1 — Data wrangling

Let's begin by downloading some peptide-MHC binding data. High-throughout peptide screening through competition experiments have resulted in large datasets cataloging binding affinities between various MHC molecules and peptides. For this exercise we will be using data from the The Immune Epitope Database (IEDB). We will download pre-processed peptide-MHC binding data from the IEDB database: http://tools.iedb.org/mhci/download/ Note the file labeled towards the top of the page: binding_data_2013.zip

IEDB is a public database of immune epitope information. The database contains data related to antibody and T cell epitopes for humans, non-human primates, rodents, and other animal species. In particular, the database contains extensive MHC class I binding data from a variety of different antigenic sources.

Dataset used for retraining the IEDB class I binding prediction tools.

- Description of the dataset: The dataset is largely identical to that of Kim et al (2014), described above, but includes additional data that was not publicly available at the time.
- Date of the dataset generation: 2013
- Details on the dataset generation: The dataset was compiled from three sources: the IEDB, the Sette lab, and the Buus lab. If a peptide/allele combination had more than 1 measurement among the three sources, its geometric mean was taken.
- Data format: Compressed text file containing binding data.
- Dataset availability: binding_data_2013.zip

Figure 5: The peptide-MHC binding data is pre-processed and can be found towards the top of the page.

Once the peptide-MHC binding data is downloaded, you can uncompress the file. The resulting tab-delimited file contains nearly 200,000 peptide-MHC combinations. Let's analyze the data again using R:

```r
# load data
iedb <- read.csv('./bdata.20130222.mhci.txt',
    header = TRUE, sep = "\t")

# let's use head to view a snippet of the data
head(iedb)
```

You should see the first few lines of the file, including the header for the columns. Let's take a moment to interpret what the values mean for each of the columns:

species This is the species from which a specific MHC
allele was evaluated for peptide binding.

mhc This is the specific MHC allele.

peptide_length MHC class I molecules bind peptides
that are predominantly 8-10 amino acid in length. Tra-
ditionally, there has been a focus on 9mer peptides
when mapping HLA-I restricted T cell epitopes.

sequence This is the sequence of the peptide.

inequality This reflects the uncertainty for some of the
peptide MHC binding data, where there some reported
affinities are either an upper-bound or lower-bound to
the true binding affinity.

meas The predicted output is given in units of IC50 nM.
Therefore a lower number indicates higher affinity. As a
rough guideline, peptides with IC50 values $<$50 nM are
considered high affinity.

WHY IS THIS INTERESTING? Several T-cell based can-
cer immunotherapies are being engineered to drive anti-
tumor immune responses for specific antigens presented
by the human MHC allele HLA-A*02:01. In cancer, so-
matic mutations altering the amino acid sequence of en-
dogenous protein coding genes can result in the genera-
tion of tumor-specific HLA-presented antigenic peptide
epitopes (or neo-antigens). These neo-antigens have the
potential to activate cytotoxic T lymphocytes (CD8+ T
cells) of the host immune system through HLA class I
molecules, thereby provoking an anti-tumor immune re-
sponse. It stands to reason that if we knew the binding
specificity of a given MHC, we could evaluate different
somatic mutations in a cancer sample and determine if it
could be presented by the cancer.

Given the data available, can we model the repertoire
of high affinity peptides that are presented by the human
HLA-A*02:01 allele? Let's try to model the distribution in
a position specific manner:

```
# let's select human,   'HLA-A*02:01',
# peptides of length 9
# and binding affinity < 50
filtered_iedb = subset(iedb, species=='human'
    & mhc=='HLA-A*02:01'
    & meas < 50
    & peptide_length == 9)


# let's grab the peptide sequences
peptide_sequences = filtered_iedb[,4]
```

Exercise 2.2 — Data Visualization

One way to visualize the repertoire of high affinity peptides that can bind to HLA-A*02:01 is to use a sequence logo plot. First, the relative frequency of each amino acid at each position is calculated. This can be referred to as a positional weight matrix (PWM). Second, the logo plot depicts the relative frequency of each character by stacking characters on top of each other, with the height of each character proportional to its relative frequency. The total height of the letters depicts the information content of the position, in bits. Here, we will use an R package called `ggseqlogo` to calculate the position specific frequencies for all high affinity 9mer peptides and visualize the sequence logo.

A **PWM** is a type of scoring matrix in which amino acid substitution scores are inferred separately for each position from a collection of aligned protein sequences.

Notably, the Matthew Stephens Lab at UChicago has also developed a sequence logo tool called `Logolas`.

```
# First let's install a seqlogo tool
install.packages("ggseqlogo")
library(ggseqlogo)

# You can also install Logolas
# BiocManager::install("Logolas")
# library(Logolas)
```

Let's pass the list of 9mers to `ggseqlogo` to visualize sequence logo:

```
ggseqlogo(peptide_sequences, seq_type='aa')
```

What positions of the high affinity peptides to seem to be highly specific for binding HLA-A*02:01? How does this compare with high affinity binding specificities of other HLA, for example, HLA-C*06:02?

Discussion

Although we have made tremendous progress in modelling peptide-MHC interactions over the past several decades, connecting which T cells interact with which MHC-bound antigens remains a challenge. An efficent solution to this problem would have broad applications in improving our understanding of T cells in health, autoimmunity, and cancer (and potentially a free trip to Sweden). This remains a challenging task, in part, due to the enormous number of potential T cell receptors, the diversity of the MHC and bound antigen peptides. However, new computational methods may help resolve TCR-pMHC interactions by integrating and learning complex patterns from diverse high-throughput experimental approaches.
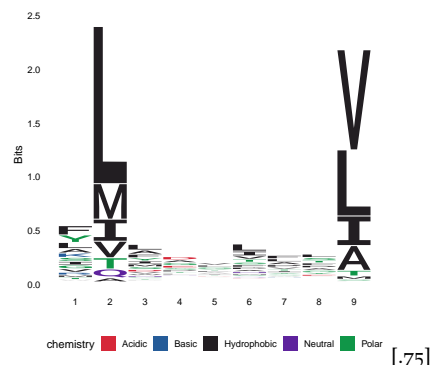

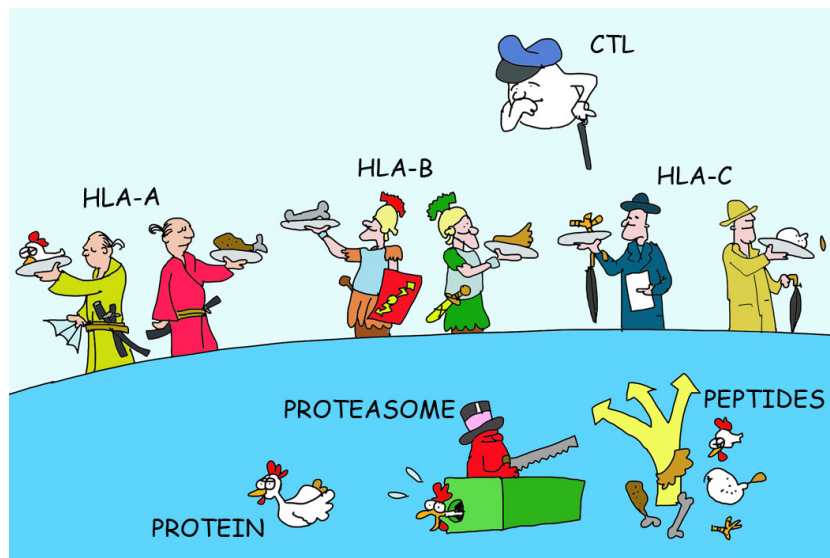Figure 6: Sequence logo of high affinity peptides for HLA-A*02:01.

Figure 7: MHC class I overview (Rock, Kenneth L., Eric Reits, and Jacques Neefjes. "Present yourself! By MHC class I and MHC class II molecules." Trends in immunology 37.11 (2016): 724-737.) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5159193/figure/F1/