The Creation and Utilization of Bioinformatics Tools to Study Patterns of the Presented HIV Immunopeptidome

A Major Qualifying Project submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the Degree in Bachelor of Science in Bioinformatics and Computational Biology

Submitted by:
Ann-Elizabeth Le
Bioinformatics/Computational Biology

Submitted on:
May 2020

Approved by:
Dmitry Korkin, PhD
Director of BCB Program
Dept. of Computer Science

ABSTRACT

Today, an estimated 38,000 new HIV infections still occur in the United States (CDC, 2020). During the early signs of immune recognition by T cells, virus-derived peptides are presented by MHC molecules on the cellular surface. In HIV infection, immune responses and immune protection are not very well defined. In this project, we analyze HIV-specific patterns during antigen processing and presentation and develop computational tools to aid in the process of the analyses. From this area of study, further advancement in protein degradation analyses and in immunopeptidome definition may help in identifying targets for efficient immune clearance and a future vaccine design.

ACKNOWLEDGEMENTS

I would like to express my deep gratitude for Professor Sylvie Le Gall for giving me invaluable advice and for giving me the opportunity to be a part of the summer program at the Ragon Institute of MGH, MIT, and Harvard for the past four years, where I have cultivated a passion for HIV research. At the institute, I am also grateful for Dr. Julie Boucau for helping me formulate multiple approaches to this project and for all of my fellow lab members and friends who have assisted me on multiple coffee runs. I would also like to offer my special thanks to Professor Dmitry Korkin for advising my project and providing me guidance throughout the process. Last but not least, I am thankful for my family and friends who have endlessly supported me through the completion of this project.

TABLE OF CONTENTS

ABSTRACT 1

ACKNOWLEDGEMENTS 2

I. INTRODUCTION 4

II. BACKGROUND 4

A. The Human Immunodeficiency Virus (HIV) 5
B. HIV Infection and the Body's Responses 5
C. The Immune System 5
D. Antigen Processing and Presentation in Subcellular Compartments 6
E. HIV Adaptation to Antigen Processing Machinery 8
F. MHC Immunopeptidomics 8

III. METHODOLOGY 9

A. Performing Peptide Degradation (Wet Lab) Experiments 9
1. HIV Peptide Degradation 9
2. Mass Spectrometry Analysis 9
3. Statistical Analysis 10
B. Organizing Peptide Degradation (Wet Lab) Experimental Data 10
1. MySQL Database: Peptide Degradation 10
2. MySQL Database: Epitopes 11
C. Web Scraping and Analysing Secondary Structure Information of Degraded Intracellular Peptides 12
D. Analyzing Amino Acid Motifs 13

IV. RESULTS 14

A. Antigen Degradation Patterns 14
B. Antigen Degradation Preferences around Specific Secondary Structures 16
C. Amino Acid Motif Identification 17

V. CONCLUSION 19

VI. REFERENCES 20

VII. APPENDICES 23

I. INTRODUCTION

It is widely believed that the human immunodeficiency virus (HIV) originated in the Democratic Republic of Congo around 1920 when the virus was transmitted from chimpanzees to humans. Although the first sporadic cases of acquired immune deficiency syndrome (AIDs) was documented prior to 1970, the widespread epidemic had not begun until the mid-to late 1970s. By this point, the virus had been predicted to have already spread to five continents, infecting somewhere between 100,000 and 300,000 people already.The large-scale infection and mortality rates of HIV have decreased through the use of drug therapies and widespread educational campaigns. Meanwhile, steps towards developing a vaccine to terminate the spread of HIV is still an ongoing struggle (The science of HIV and AIDS - overview, 2019). Despite vast advances in antiretroviral therapy, The high diversity and rapid evolution of HIV sequences within a person and across a large population is a major struggle for the design of a universal HIV vaccine (Santoro, 2013).

Many studies have shown to predict the results of vaccinations in a much more in depth look based on types and breadth of immune responses and of its protectiveness after infection. New correlates of immune protection and monitoring parameters during vaccine strategies have been researched as well. One such area of study that would be very interesting to look into would be the recognition of peptides naturally processed and presented by HIV-infected cells. A better understanding of antigen degradation and the development of computational tools that incorporate both the diversity of the degradation machinery and sequence diversity of HIV will allow for researchers to be able to identify areas of immune escape at the population level. With this, motif identification associated with efficient peptide presentation would also be an expansion of our immune knowledge. The expansion of understanding the mechanisms behind HIV antigen processing and the HIV immunopeptidome through the development of more advanced computational tools would be extremely useful. They could be used to predict antigen processing and presentation in the context of antigen variability to help identify more conserved targets for immune recognition and help improve vaccine design. (Le Gall, 2019).

Therefore, this project sought to understand the mechanism of specific protein degradation to broaden our knowledge of . Through the process of analyzing these patterns, supplemental bioinformatics methods were utilized in order to perform analyses on a much more extensive scale.

II. BACKGROUND

A. The Human Immunodeficiency Virus (HIV)

In the 1980s, the world experienced a massive outbreak of the human immunodeficiency virus (HIV) and acquired immune deficiency syndrome (AIDS) although the disease had originated decades earlier (History of AIDS, 2017). According to the World Health Organization, 75 million people have been infected with HIV and 32 million people have died from the disease since the beginning of the epidemic. By the end of 2018, 37.9 million people were living with HIV/AIDS worldwide (HIV/AIDS,).

HIV is a retrovirus that stores its genetic information using RNA and is made up of nine genes that contain instructions to make new viruses. Figure 1 depicts the process of cellular infection, integration of genetic information, and production of new viral proteins. HIV begins to infect immune system cells by attaching to immune cells that have CD4 receptors on the surface such as T cells, monocytes, macrophages, and dendritic cells. The CD4 receptor signals to the immune system about the presence of antigens. After attachment and fusion, HIV uses reverse transcriptase enzyme to convert RNA into proviral DNA inside of the host cell. This proviral DNA becomes attached to HIV's integrase enzyme and enters the cell nucleus. Once it binds to the host DNA, the HIV DNA strand integrates into the DNA of the host cell. When the cell receives a signal to become active, HIV uses the host enzyme RNA

polymerase to make mRNA to make new viral protein chains. Once these chains finish assembling into new viruses at the cell wall, they leave the cell and become infectious (The science of HIV and AIDS - overview, 2019).

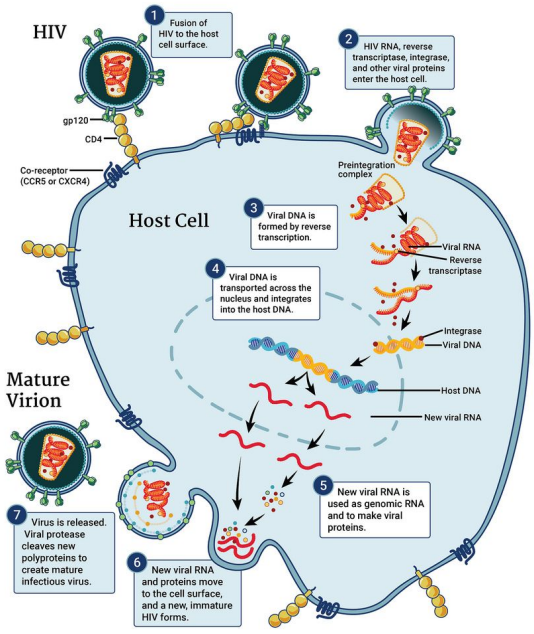


Figure 1: This illustrates the HIV replication cycle, which begins with the (1) binding of the virus to the CD4 receptor on the host cell surface. (2) A capsid carrying the virus's genome and proteins enters the cell, and (3) an HIV protein, reverse transcriptase, transcribes the viral RNA into DNA. (4) This new viral DNA is transported across the nucleus and integrates into the host cell's DNA. (5) Viral RNA is now used to create viral proteins that (6) move to the cell surface where a new, immature HIV forms. (7) The virus is released from the cell and matures into an infectious virus (Betts, 2019).

B. HIV Infection and the Body's Responses

Once the body is infected with HIV, there are three stages at which different symptoms occur: primary or acute HIV infection, chronic or clinical latency infection, and eventually AIDs (The science of HIV and AIDS - overview, 2019). During acute HIV infection, the earliest stage of HIV infection, the viral load generally increases and the number of CD4 T cells decreases. During the second stage of HIV, chronic or clinical latency infection, patients normally receive antiretroviral treatment, resulting in rapid HIV control and partial restoration of immune functions (Deeks, 2007). However, untreated patients can develop extreme HIV symptoms, which can potentially lead to the most severe form of HIV infection in less than one year, AIDS. At this stage, a person is more vulnerable to cancer and other life-threatening infections (History of AIDS, 2017).

C. The Immune System

The immune system is a highly complex and powerful interconnecting network made up of tissues, cells, and proteins whose primary function is to defend the body from non-self pathogens. This defense mechanism enables the body to prevent potential infection and illness from disease-causing organisms, such as bacteria and viruses. Two subgroups of the immune system are the innate and adaptive, both of which are critical in responses to foreign particles (Thompson, 2015). The innate immune system is the more rapid of the two responses, acting as the initial line of defense to prevent the spread of pathogens throughout the body. Innate immune cells recognize specific patterns found on microbial surface components of pathogens, which leads to induction of pro-inflammatory cytokines in infected cells and activation of adaptive immune cells (Levy, 2001). The adaptive immune system is a slower but more targeted approach when responding to pathogens by distinguishing between self and non-self molecules in order to destroy the latter. Both B and T lymphocytes are white blood cells which are responsible for humoral and cell-mediated immunity, respectively (Alberts, 2002). Research continues to be conducted on all forms of the immune system to find treatments and protection against infectious diseases, such as vaccines. Vaccines trials are continuously being conducted to seek for preventative measures against diseases and illnesses, such as the human immunodeficiency virus (HIV) (Understanding HIV/AIDs, 2019).

D. Antigen Processing and Presentation in Subcompartments

During initial exposure to an antigen, the body's adaptive immune system works to recognize and destroy the invading virus. In cell-mediated immune responses, activated T cells interact directly with a foreign antigen that is presented to them on the surface of a host cell (Alberts, 2002). In HIV infection, immune responses are not able to clear infection and the mechanisms behind this are not well defined. T cell recognition of antigen-presenting cells highly depends on the expression of peptides bound to major histocompatibility complex class I (MHC-I) and class II (MHC-II) molecules. The antigen processing machinery plays an important role in the conversion of antigens into MHC-I- and MHC-II-bound peptides for protective T cell responses (Blum, 2013).

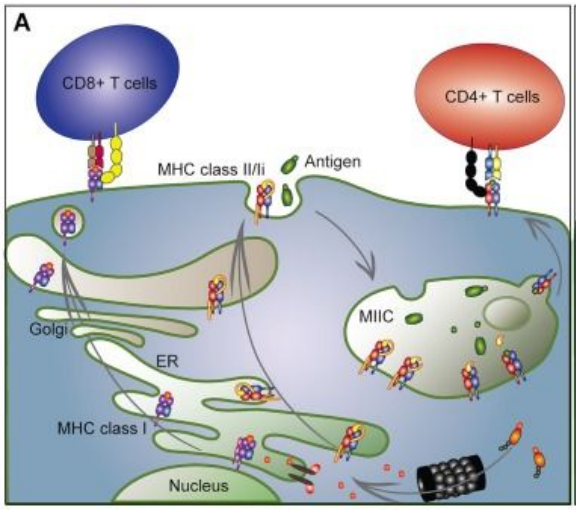


Figure 2. A) Classical antigen presentation pathways. MHC class I molecules come together in the ER with chaperone proteins until peptide binding. Ubiquitinated antigens are degraded by the proteasome, and the cut remaining peptides are transported into the ER lumen by transporters associated with antigen presentation (TAPs). The peptides are loaded onto MHC class I, tapasin is released and the peptide-MHC class I complex is transported through the Golgi to the cell surface where they are recognized by CD8+ T cells. MHC class II molecules assemble in the ER with chaperone invariant chain II, which mediates travel of MHC class II from the ER through the Golgi to the endosomal pathway. Peptide-loaded MHC class II molecules are released from the endosomal compartment to the cell surface where they are recognized by CD4+ T cells (Malmstrøm, 2013).

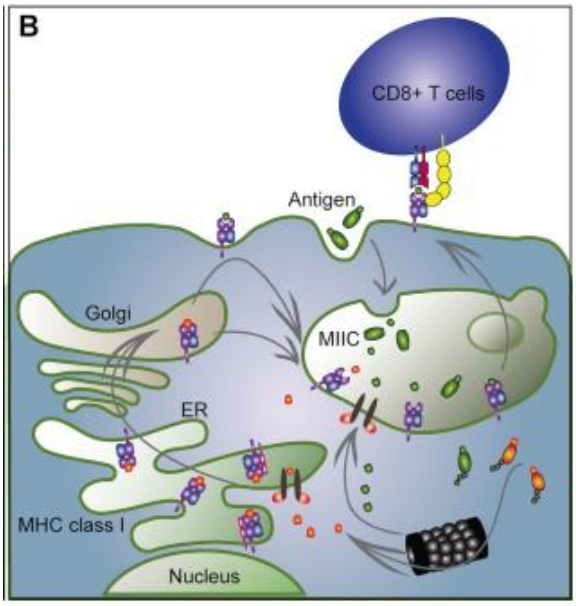


Figure 3. B) Alternative (cross-presentation) pathway. MHC class I molecules carrying signal motifs in the cytoplasmic tail are transported to the endosomal pathway where endocytosed antigens are degraded. Peptides are loaded directly in endosomes in a TAP-independent manner, or the antigens can translocate to the cytosol for proteasomal degradation. The antigens are either loaded onto MHC class I in the ER or transported back by TAP transporters. Peptide-loaded MHC class I molecules are then released to the cell surface for antigen presentation to CD8+ cells (Malmstrøm, 2013).

HIV infects cell subsets expressing CD4 and productive infection requires that HIV successfully fuses at the plasma membrane that delivers the virus into the cytosol, which continues to uncoat and go through reverse transcription during its transport to the nucleus to become transcribed and translated (Churchill, 2016). Productive infection is not as common since there are a multitude of factors related

to host restriction and degradation machinery (Chauveau, 2017).

Within the cell, there are various compartments in which proteases and peptidases degrade antigens during antigen processing; different degradation patterns from this contribute to defining the amount of peptides able to load onto MHC molecules.

Cytosolic proteasomes unfold and degrade into fragments that are further processed into peptides by cytosolic peptidases. Some of these peptides are transferred to the endoplasmic reticulum (ER) to be degraded by other aminopeptidases before being loaded onto MHC-I (Lázaro, 2015). Exogenous antigens can be processed by cathepsins in endosomes and lysosomes. Degraded peptides from the cytosol and ER are then cross-presented in the MHC-I or transported into the late endosomal MHC-II compartment for MHC-II loading and presentation (Blu, 2013). Other proteasomes, peptidases, and cathepsins are involved with processing HIV epitopes but not much is known about specific epitopes required for antigen processing and presentation (Le Gall, 2007).

Due to the difference in degradation patterns, studies have shown that some epitopes are produced in all compartments while others are produced in more specific compartments. Within the variable timing and nature of peptide degradation, peptidase activity seems to change throughout cellular subsets targeted by the virus (21, 22). Macrophages express the highest levels of hydrolytic activities due to their phagocytic functions while dendritic cells express high lysosomal activities but low cytosolic peptidase activities (11, 23). Because of these differences in peptidase activity, this changed degradation patterns of HIV antigens which in turn affected the kinetics and amount of epitopes produced. Epitope production may have implications to the MHC-peptidome environment and possibly efficiency of immune recognition by epitope-specific CD8 T cells. Peptidase hydrolytic activity across cellular compartments are distinct, therefore making their mechanistic impact on antigen processing unclear (Dinter, 2015). External factors such as interferon gamma has been shown to increase expression of peptidases such as ERAP1 or LAP, the expression of immunoproteasome catalytic subunits, TAP or MHC-I, which in turn increased antigen processing and presentation of MHC-peptides (Serwold, 2002).

In terms of immunogenicity, HIV Gag p24, Env, Nef have been found to be the most immunogenic while Tat and Vpu are the least immunogenic. The conservation of the MHC-I peptides across variants or strains vary and are hard to pinpoint patterns in (Le Gall, 2007). There have been many studies dedicated to identifying degradation patterns of HIV antigens by different compartments such as proteasomes, cytosol, or endolysosomes (12, 21, 22). These studies have identified various areas of faster peptide degradation resulting in fragments that are too short to load onto MHC-I or areas of slower peptide degradation resulting in nested peptide fragments that are potentially loaded onto MHC-I (Le Gall, 2007). Another factor that may contribute to CD8 immune responses within an HIV protein focuses on the kinetics of epitope production. Flanking epitope motifs are important in potentially predicting production of peptide fragments and thus peptide presentation by MHC-I or MHC-II.

Despite the critical role that the MHC peptidome plays in immune recognition, the kinetics behind the processing and presentation of peptides to immune cells is not fully understood. Thus far, it has been mainly inferred based on T cell responses in HIV-infected persons (Le Gall, 2007). Some algorithms like NetChOP, which predicts proteasomal cleavage sites and others that analyze TAP or MHC binding help shed light upon degradation features, but do not account for peptidases involved in protein degradation, specific degradation machinery in different cellular compartments, and viral diversity (Lázaro, 2015). HIV proteins are already highly variable within an individual, making it even more so at a population level (Llano, 2013). Further research is needed to identify kinetics of peptide degradation in each cell subset to assess whether the different compartments

result in more optimal targets for immune recognition in infected cells.

E. HIV Adaptation to Antigen Processing Machinery

Antigen processing is a complex multistep process that involves intracellular degradation of self- and pathogen-derived proteins. Once they are degraded, they become peptides that are loaded onto major histocompatibility complex (MHC) molecules and displayed at the cell surface. Immune T cells detect and clear infected cells that are presented as peptides derived from intracellular degradation of the pathogen. MHC-I complexes display 8-11aa of optimal size long peptides to CD8 T cells while MHC-II complexes display peptides of non-canonical sizes of up to 16aa to CD4 T cells (Kourjian, 2014).

Degradation patterns of HIV-derived proteins into epitopes displayed by MHC-I or MHC-II have an effect on HIV-specific immune responses and the recognition of infected cells (Kourjian, 2014). In one experiment, a cell line expressing soluble MHC, primary CD4 T cells infected with relicative HIV, B cells infected with non-replicative HIV, 293 T cells transfected with HIV, and cells infected with an MVA vector expression HIV antigen fragment were all found to show that HIV-derived come mostly from Gag proteins aside from other less abundant proteins. It was also suggested that certain areas of the Gag proteins were more efficiently presented across different cell types, HIV expression systems, or HLA types (Yaciuk, 2014).

Longer peptides with extensions on the N- or C-terminal ends were found to not be easily predictable by MHC-I binding anchors. Therefore, the specifics of location and loading of peptides onto MHC are unknown along with the structure of the MHC-peptide complexes and stability of the MHC-peptide at the cell surface (The science of HIV and AIDS - overview, 2019). Potential identified MHC-bound peptide biases during CD8 recognition can help define targets for immune recognition of infected cells displayed by dendritic cells after vaccination (Churchill, 2016). Another aspect to consider in the variability of HIV peptide presentation are defective ribosomal products (DRiPs) which may play a part in early peptide presentation. Alternate reading frame translation products have been found in HIV-infected cells; however, their role and translational byproducts in HIV replication is unknown (Goldwich, 2008). The underlying mechanisms in the presentation of HIV peptides across different HIV proteins are still not understood due to the lack of datasets that exist on the HIV-derived immunopeptidome (Berger, 2015). This knowledge is another critical aspect in helping to identify targets for immune recognition and immunogen design (Yaciuk, 2014).

F. MHC Immunopeptidomics

HIV-specific CD8 and CD4 immune responses have been identified in HIV-related cases, with peptide specificity, HLA restriction, immune response frequency, cytokine production, but the relationships between these factors are still not well defined (Porichis and Kaufmann, 2011). HIV-specific T cells play an extremely important role in controlling viral load in both HIV-positive patients and HIV controllers (Betts, 2006). Studies have tried to show correlations between viral load patterns in acute infection with T cell responses and HLA-restrictions, but have shown that T cell count alone does not clear infection. Non-protective immunodominant responses, immune pressure by T cell responses driving mutations in the virus and immune escape, antigen stimulation, and viral reservoirs establishment have all shown to contribute to some extent towards lack of T cell response efficacy (Kourjian, 2016).

Specific HLA and CD8 or CD4 T cell responses against Gag polyproteins are seen to be associated with lower viral load and spontaneous control, but the underlying mechanisms are not fully understood (Hancock et al., 2015). The research in detecting patterns within immune protection has focused on CD4, CD8 T or NK cell responses during infection, but less research has been done in the area of antigen processing and presentation to the antiviral capacity or immune responses (Dinter, 2015). As T cell responses are activated through MHC-peptide

recognition, there are various questions related to the nature and amount of HIV peptides displayed by HIV-infected cells across HLAs. It is also important to factor in the question of how efficient HIV-specific T cells are primed during infection to detect how HIV peptides are processed and presented by infected cellular subsets. Currently, there are no comparative datasets that exist for the MHC-peptidome displayed by CD4 T cells, macrophages, and dendritic cells of the same donor after HIV infection. With the variability in antigen processing activity and degradation patterns of HIV proteins between CD4 cells and other cellular subsets, this suggests the differences in peptide presentation as well. If this prediction were to be true to some degree, the assumption would be that the most efficient immune responses would be defined as those targeting commonly presented peptides within HIV-specific immune cells (Kourjian, 2016).

It will be important to study factors relating to HIV immunodominance during infection, which may be attributed to differences in antigen processing. Therefore, it will be useful to analyze HIV T cell patterns that occur during the degradation process of antigens, kinetics of epitope production in cell extracts or in dendritic cell endolysosomes, as well as MHC binding affinity (Le Gall, 2007). However, another aspect to consider would be mutations that occur that prevent proper peptide recognition by T cells, causing some immunodominant early responses effortless and broadening of immune responses during the chronic infection phase (Karlsson, 2007). With HIV being highly variable, it has developed many mechanisms to avoid or limit its presentation to immune cells, including the down-modulation of MHC-I and an adaption to the antigen processing machinery (Schwartz, 1996). With this, recent studies have been designed to analyze patterns defining antigen processing mutations during viral evolution. A study showing cleavable or non-cleavable residues by aminopeptidases showed that mutations towards poorly cleavable residues reduces epitopte production. Residues flanking HIV epitopes usually mutate due to HLA restriction around poorly cleavable residues (Zhang, 2012). HLA-restricted mutations within epitopes where people share one HLA usually have a reduction in intracellular peptide stability, amount of peptides available for CTL recognition, and immune escape. With the level of intracellular peptide stability is determined by specific motifs, immune escape patterns could be predicted in a population (Lázaro, 2011). Therefore, having a better understanding of antigen degradation and the development of computational tools that combine the diversity of the degradation machinery and sequence variability of HIV will be extremely important steps to identify immune escape patterns and to define motifs associated with efficient peptide presentation.

III. METHODOLOGY

A. Performing Peptide Degradation (Wet Lab) Experiments

To analyze peptide degradation patterns, wet lab experiments were performed and analyzed in the following manner under the next three sections.

1. HIV Peptide Degradation

Purified epitope-containing HIV peptides (1.5nmol) from Massachusetts General Hospital donors were digested with 15μg CD4+ T cell extracts at 37°C in degradation buffer (50 mM Tris-HCl, 137 mM potassium acetate, 1 mM MgCl2, and 1 mM ATP) at pH 4 and 7)). Aliquots were taken at a time point of 1 hour, and the reaction was stopped with 2μL 100% formic acid. Peptide fragments in the mix were purified by 10% TCA precipitation followed by centrifugation at 14,000 rpm at 4°C for 30 min to isolate the digestion products in the supernatant (Almazan, 2016)

2. Mass Spectrometry Analysis

The identity of the peptides in the digestion mix was determined by in-house mass spectrometry analyses. Equal amounts of peptide degradation samples at different time points were injected into a Nano-HPLC (Eksigent) in line with an Orbitrap mass

spectrometer (LTQ Orbitrap Discovery; Thermo) with a flow rate of 400 nL/min. A Nano cHiPLC trap column (200 μm × 0.5 mm ChromXP c18-CL 5 µm 120Å; Eksigent) was used to remove salts from samples, and peptides were separated on a Nano cHiPLC column (75 μm × 15 cm ChromXP c18-CL 5 µm 300Å; Eksigent) over a gradient of 2–40% buffer B (buffer A, 0.1% formic acid in water; buffer B, 0.1% formic acid in acetonitrile) and electrosprayed in the mass spectrometer. Mass spectra were recorded in the range of 370–2000 Da. In tandem mass spectrometry mode, the eight most intense peaks were selected with a window of 1 Da and fragmented. The collision gas was helium, and the collision voltage was 35 V. Tandem mass spectrometry spectra were searched against custom-made source peptide databases with Sequest and Proteome Discoverer (version 1.3; Thermo Scientific). The integrated area under a peak generated by a given peptide is proportional to the abundance of that peptide. Each degradation time point was run on the mass spectrometer at least twice (Almazan, 2016).

3. Statistical Analysis

Using our mass spectrometry data, we analyzed protein degradation through size, size intensity, size % intensity, N- and C-terminal peptide cuts, and coverage. We also quantified intracellular peptide presentation and surface peptide presentation (Almazan, 2016).

B. Organizing Peptide Degradation (Wet Lab) Experimental Data

To analyze peptide degradation patterns, wet lab experiments were performed and analyzed in the following manner under the next three sections.

1. MySQL Database: Peptide Degradation

Databases are a useful way to store large and structured amounts of information in a computer, especially one that is accessible in various ways. Most databases contain multiple tables, which may each include several different fields. Therefore, creating a database for experimental databases would be useful in organizing results and finding patterns amongst big data. MySQL is an open-source relational database management system. Its name is a combination of “My”, the name of one of the co-founder’s daughters, and “SQL”, which stands for Structured Query Language. Using the MySQL Workbench,

With the peptide degradation experiments, peptides are degraded in various pHs to mimic the various subcellular compartments during antigen processing. For the first database, tables were created to organize results obtained from peptide degradation experiments. Figure 4 depicts the entity relationship diagram for the tables and fields that this database stores. The ‘people’ table stores the information for experimenters’ names. The ‘projects’ table stores the information for the name of the project, the species of the project subject (species), and any additional information (notes). The ‘experiments’ table stores the information for the person that performed the type of project it is associated with (proj_id), experiment (people_id), the date of the experiment (date), the donor identity (donor), the type of experiment (Treatment), and the cell type (cell_type). The ‘original_peptides’ table stores the information for a reference to the peptide (ref), the peptide name (oripep_id), the type of protein (protein), the pathogenic disease the peptide is linked to (pathogen_disease), and any additional information (notes). The ‘experiment_data’ table stores the information for the type of project from the projects.csv file (proj_id), a person that performed the experiment, the experiment identification from the experiments.csv file (exp_id), the pH (pH), time point (time), peptide sequence (seq), the reference peptide information (ref), the intensity (intensity), and the total intensity of the peptides (total_intensity). The ‘input’ table inside the MySQL Workbench stores the information from the local input.csv file that is uploaded with the peptide degradation information, which provides information for the ‘experiments’, ‘experiment_data’ and ‘original_peptides’ tables.

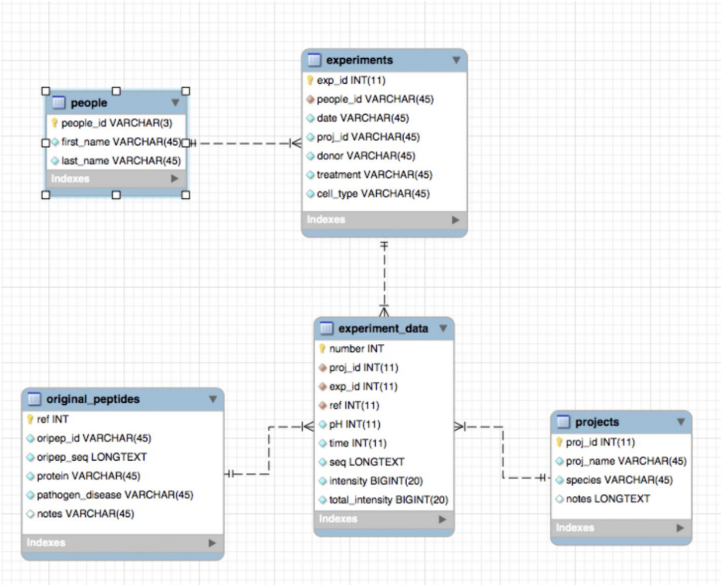


Figure 4. Entity relationship diagram for peptide degradation database.

The following figures are the .csv file templates with examples of what to put in each field in order for each corresponding table to have data. These .csv files are then imported to the MySQL Workbench.

input_example.csv

people_id	date	proj_id	ref	treatment	donor	cell_type	oripep_id	oripep_seseq	protein	pathogen	pH4 10	pH7 30
JB	########	1	4	none	27593766	mac	TB10.4-1-MGGKWS	RAEPAAD	Gag	HIV	1.77E+09	5.06E+09
JB	########	1	4	none	27593766	mac	TB10.4-1-MGGKWS	MRRAEPA	Gag	HIV	2E+09	5.92E+09
JB	########	1	4	none	27593766	mac	TB10.4-1-MGGKWS	MGGKWS	Gag	HIV	4764217	10205708
JB	########	1	4	none	27593766	mac	TB10.4-1-MGGKWS	MGGKWS	Gag	HIV	1.01E+08	2.72E+08
JB	########	1	4	none	27593766	mac	TB10.4-1-MGGKWS	MGGKWS	Gag	HIV	1.97E+11	3.07E+11
JB	########	1	4	none	27593766	mac	TB10.4-1-MGGKWS	GGKWSK	Gag	HIV	2.02E+10	2.99E+10

Figure 5. Template for what to put in input_example.csv file.

people_id	first_nam	last_name
AA	Anthony	Almazan
AL	Ann	Le
AV	Archana	Vaithiligam
CM	Cynthia	Masai
DW	Daniel	Wambuza
ED	Ellen	Duong
EO	Eniola	Oladipo
FR	Fabrice	Renaud
GK	Georgio	Kourjian
HP	Hugo	Poplimont
JB	Julie	Boucau
JD1	Jens	Dinter
JD2	Jeremy	Dufloo
JM	Julien	Madouasse

Figure 6. Template for what to put in the people.csv file.

proj_id	proj_nam	species	notes
1	CD4_activ	human	HIV
2	CD4_LRA	human	TB
3	Cancer	human	Bryostatin
4	Gag_pept	human	Panobinostat
5	peptidom	human	Ingenol
		human	Disulfiram
		human	CD3CD28
		human	PHA
		human	IFNg
		human	Ritonavir
		mouse	Nelfinavir
		mouse	Darunavir
			LPS
			CL097
			R848

Figure 7. Template for what to put in the projects.csv file.

2. MySQL Database: Epitopes

With the epitope database, the 'surfacepep_papers' table stores the information from the local surfacepep_input.csv file. This table stores the information for the identification number of the paper (ref), the author of the reference (first_author), the journal of the paper (journal), the year of the

paper (year), the cells used in the experiment (cells), the assay (assay), the sequencing method (sequencing), the number of peptides (num_peps), the data (data), the HLAs identified in the paper (HLAs), the HLA assignment (HLA_assignment), and any additional comments (comments), the data origin (data_origin), the search engine (search_engine), the search parameters such as precursors or fragments masses (search_param), the false discovery rate (false_dscvr_rate), and any post-translational modifications (posttrans_mod). The 'paper_library' stores the information from the local surfacepep_input.csv file. This table stores the information for the reference of the paper (ref), the first author of the paper (author), the amount of peptides mentioned in the paper (num_pep), and the number of unique peptides in that paper (num_uniquepep). The 'peptide_library' table stores the information from the local peptidelib_input.csv file. This table stores the information for the identification number of the paper (id), the peptide sequence (seq), the length of the peptide (length), the accession number of the peptide (access_num), the protein name of the peptide (prot_name), the gene name of the peptide (gene_name), the HLA identity (HLA_identity), the HLA class (HLA_class), the post-translational modifications of the peptide (posttrans_mod), any extra information for that peptide (comments), and the sample that the peptide came from (sample). The user will have to input information into the following .csv files to upload the data onto the MySQL Workbench.

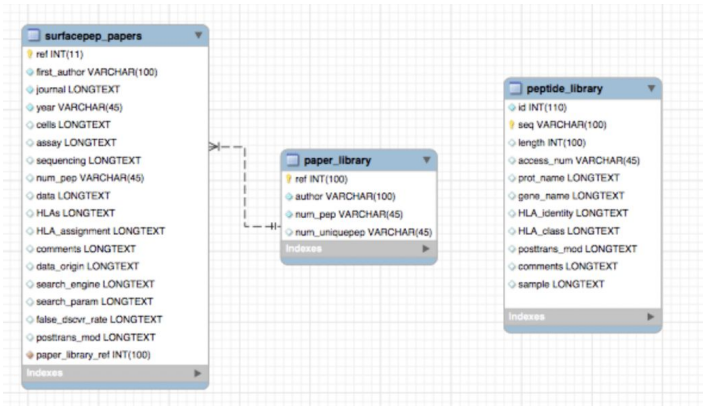


Figure 8. Entity relationship diagram for epitope database.

Ref	Sequence	Length	Accession	Protein_n	Gene_na	HLA_iden	HLA_class	Post_tran	Comment	Sample
279	KESTLHLV	9	æP02248	Ubiquitin	name	A29	I	oxi	platelet	human
279	ASRPGELF	9	æNP_002	Platelet f	name	B57	I	oxi	platelet	human
279	KEVPDAC	9	æNP_003	Thrombos	name	A29	I	oxi	platelet	human
279	ALNELLQH	9	æAAD131	Talin	name	A2	I	oxi	platelet	human

Figure 9. Template for what to put in the peptidelib_input.csv file.

ref	first_auth	journal	year	Cells	HLAs	Assay	Sequenci	num_uniq	num_pep	data	HLA	HLA_assig	comment	Data_orig	Search_e	Search_p	False_Dis	Post_translational_Modificatio
1	Milner	MCP	2013	MCF	A*0201;	Bimmunoaffinity			1776	yes	HLA	not assigned			MaxQuan	6ppm, 0.55%	FDR	Met oxidation, n-acetylation
2	Scull	Mol. Imm.	2012	Jesthym	A*0201,	Bsecreted HLA; immunoaffinity			1978	yes	HLA	not assigned;	A2 only?		Paragon	In Protein P	1% FDR	no cys alkylation
3	Hassan	MCP	2013	B-LCL HH	A*0201,	Bimmunoaffinity			15,736	yes	HLA		PTMs		Mascot	2PPM , 0.5DA		Met oxidation, phosphorylatio
4	Gebreselt	Human Im	2006	U937	A*0301	acid elution and immunopreci			85	yes	HLA		MALDI		GPS system			
5	Weinzierl	Eur. J. Im	2008	LCL721.4	B*0201	aimmunoprecitpitation			49	yes	HLA		TAP-dependent vs.		Mascot	0.2Da, 0.2	manual annotation	1%FDR
6	Vogel	MTNA	2013	HEK293,	B*02/02,	immunoprecipitation			54	yes	HLA		rapamycin treated		Mascot, manual annotation			
7	Nijveen	Immunog	2011	B-LCL JY	A*0201	immunoaffinity			9	yes	HLA	not assigned	HSVPdb		Mascot	2ppm, 0.8Da		
8	Milner	MCP	2006	UCI-107	A*0201	immunoaffinity			36	yes	HLA				Sequest, Pepminer			
9	Hickman	JI	2004	B-LCL 721	B*1801	soluble HLA			200	yes	HLA				Mascot, Bio-Multiview			
10	Granados	Blood	2012	B-LCL fro	HLA-A*01	acid elution			2,273	yes	HLA	not assigned			Mascot	0.02Da, 0.5Da		oxidation (Met) and deamidati
11	Granados	Nat. com	2014	cells from	A*03:01,*	acid elution			6480	yes	HLA	not assigned						
12	Juncker	Plos ONE	2013						2,062	yes	HLA	not assigned	SYFPEITHY data					
13	Garcia-Ma	MCP	2012	C1R	A*6801	immunoaffinity			816	yes	HLA		also MALDI					
14	Escobar	JI	2008	K562, B-LC	B*3501,	Bimmunoaffinity			927	yes	HLA							

Figure 10. Template for what to put in the surfacepep_input.csv file.

C. Web Scraping and Analyzing Secondary Structure Information of Degraded Intracellular Peptides

Web scraping from UniProt and fuzzy matching were utilized to analyze secondary structure of peptide degradation experiments. To begin, a text file of peptide sequences with their corresponding protein accession numbers is needed. However, in the case that the accession numbers were unknown for the peptide sequences, Basic Local Alignment Search Tool (BLAST) protein-protein (BLASTp) in batches was utilized to retrieve that information. BLASTp is an alignment tool that finds regions of similarity between biological sequences, in this case protein sequences, and compares them to sequences databases to calculate statistical significance (Altschul, 1990). With this, the Figure 25 in the appendix shows the protocol for how to utilize BLASTp to retrieve the original protein accession number with an input of a FASTA sequence of peptide sequences. However, in the case that the file was not in FASTA format, Figure 26 in the appendix shows the protocol for how to make an appropriate FASTA file through Excel. A FASTA file is a text-written file that begins with a single-line description with a greater-than (">") symbol at the

beginning, followed by lines of sequence data (Altschul, 1990).

Once the protein accession numbers were identified, UniProt was used as an online database to scrape secondary structure information using these accession numbers. UniProt is an open source database of protein sequence and functional information (The UniProt Consortium UniProt, 2019). Figure 27 in the appendix shows the protocol of how to webscrape secondary structure information from UniProt with the corresponding protein accession numbers. In said protocol from previously mentioned Figure 27, a Python script named "accession_numbers_into_uniprot_url.py" was used to output a .txt file of UniProt URLs from an input of protein accession numbers (shown in Figure 28 in the appendix). After the UniProt URLs are created into another text file, a Python script named "secondary_structure_through_multiple_uniprot_urls .py" was created to scrape the information from all of the URLs and put that information into a new .csv file.

For each peptide derived from specific proteins, secondary structure information (alpha helices, beta strands, turns) was parsed by a Python script using the BeautifulSoup library. The BeautifulSoup library is a Python library for scraping data from HTML files (Richardson, 2020). Figure 29 in the appendix shows the Python code that was used to parse secondary structure information from UniProt with an input of a .txt file full of UniProt website URLs. This gives an output of secondary structure information on the specific website URLs for each protein accession number.

On the secondary structure information was all inside a .csv file, the frequency was analyzed by matching the peptide ranges with the corresponding secondary structure and totalling the count of alpha helices, beta strands, and turns. With this,

D. Analyzing Amino Acid Motifs

To study amino acid motifs from where peptides were being degraded, Figure 30 shows a Python script created to scrape further protein information from a list of UniProt URLs, which was also utilized in the previous section of parsing secondary structure information. With this, each degraded peptide sequence was matched in rows with their length, corresponding protein accession number, and the full length origin source protein sequence. From there, Excel formulas were created to locate their position within the source protein to analyze six and amino acids before and after the N-terminal and C-terminal of the degraded peptide region. From there, each single amino acid was put into its own separate single Excel cell, similar to what is shown in Figure 11.

	K	L	M	N	O	P	Q	R	S	T	U	V
r pi	N-6	N-5	N-4	N-3	N-2	N-1	N1	N2	N3	N4	N5	N6
G	A	S	G	T	V	V	A	A	A	A	G	G
L	V	L	T	E	H	V	A	A	A	E	L	G
	A	L	K	K	A	L	A	A	A	G	Y	D
	A	L	K	K	A	L	A	A	A	G	Y	D
	A	L	K	K	A	L	A	A	A	G	Y	D
	A	L	K	K	A	L	A	A	A	G	Y	D
	R	K	E	E	K	K	A	A	A	P	A	P
	D	A	I	H	V	G	A	A	A	P	V	V
	Q	K	K	V	K	L	A	A	D	E	D	D
	E	A	I	N	Y	M	A	A	D	G	D	F
	E	A	I	N	Y	M	A	A	D	G	D	F
V	I	V	G	K	K	C	A	A	D	L	G	L
E	I	V	G	K	K	C	A	A	D	L	G	L
E	I	V	G	K	K	C	A	A	D	L	G	L
D	I	V	G	K	K	C	A	A	D	L	G	L
V	I	V	G	K	K	C	A	A	D	L	G	L
	I	V	G	K	K	C	A	A	D	L	G	L
V	M	I	G	Q	A	T	A	A	D	Q	E	K
	R	A	L	A	A	P	A	A	E	E	K	E
G	R	A	L	A	A	P	A	A	E	E	K	E

Figure 11. Example of amino acids flanking motifs six amino acids before the N-terminal split into each separate cell on Excel.

For each amino acid position before and after the N- and C-termini, a count was taken for each amino acid in that specific position. Therefore, a color-coded heat map was created to show the frequency by count and intensity by color of each amino acid at each position. On a color scale, the closer to red it was, the higher the frequency and the closer to green it was, the lower the frequency.

From these frequencies, a naive approach was first taken to identify the top five flanking amino acid motif sequences before and after the N- and C-termini. However, a motif sequence identifier for short sequences called MEME suite was later utilized to try and identify amino acid motif sequences before and after the N- and C- termini respectively.

IV. RESULTS

A. Antigen Degradation Patterns

From a subset of 7107 non-duplicate intracellular peptides that derived from 1568 proteins, peptide size was first analyzed as a pattern within the peptide degradation wet lab experiments. For each degraded peptide, the length was analyzed in terms of amino acids, as shown in Figures 12 and 13.

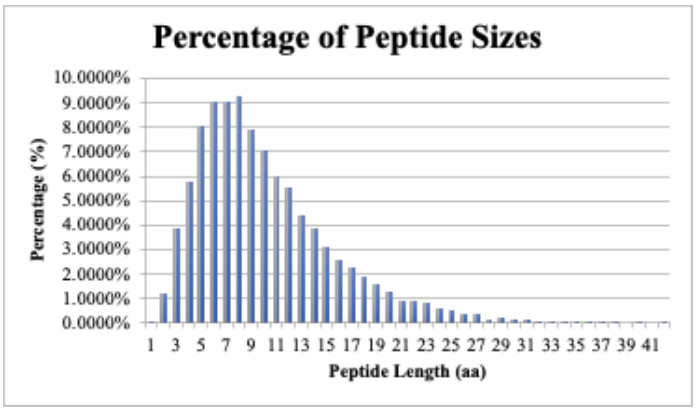


Figure 12. Percentage of peptide sizes in amino acids.

From Figure 12, the size of degraded peptides is right-adjusted on the graph, with a high percentage around the 6 to 9 amino acids in length.

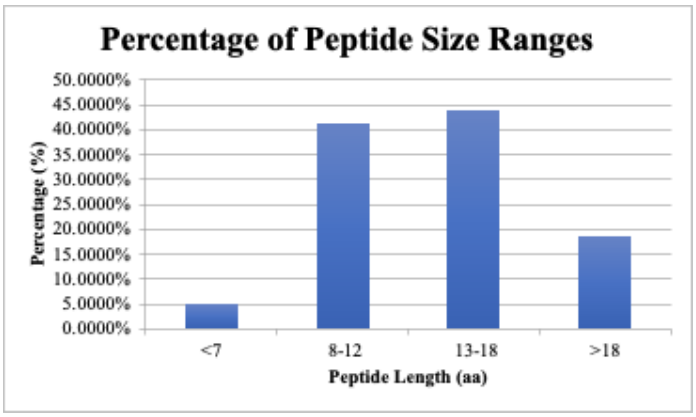


Figure 13. Percentage of peptide size ranges in amino acids.

To group the peptide sizes as ranges, almost half of the peptides were either 8-12 amino acids or 13-18 amino acids in length. This "preference" in peptide sizes when being degraded in the wet lab experiments are a confirmation that the experiments are designed in a way that mimics the antigen processing environment such as endosomes or lysosomes.

Another pattern analyzed from peptide degradation was the amount of peptides that came from each protein.

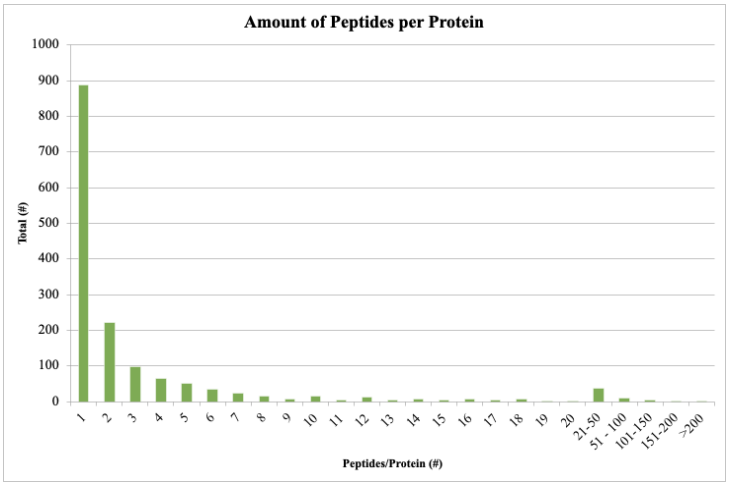


Figure 14. Amount of peptides per protein.

From a dataset full of 1538 proteins, Figure 14 shows a right-adjusted graph, with 887 peptide sequences corresponding to one protein. Then, there is a sharp decline to 222 proteins that had 2 peptide sequences corresponding to it. It then falls quickly down to 98 proteins having 3 peptides each. Overall, most proteins had 1 corresponding peptide sequence Therefore, this can suggest that there will be high variability in patterns seen across the peptides that derive from the large number of different proteins.

Another pattern that was analyzed was the molecular weight (Mw) and theoretical pI (isoelectric point) of the source proteins in which the peptides are derived from. This data was taken from an Expasy tool to compute pI/Mw (Bjellqvist, 1994). There were four categories that the peptides were put in: small acidic, small basic, large acidic, and large basic. To be considered small, the peptide was less than 500,000 g/mol in molecular weight and big is

greater than 500,000 g/mol in molecular weight. An acidic peptide would have pI less than 7 in pH and basic if pI was greater than 7 in pH.

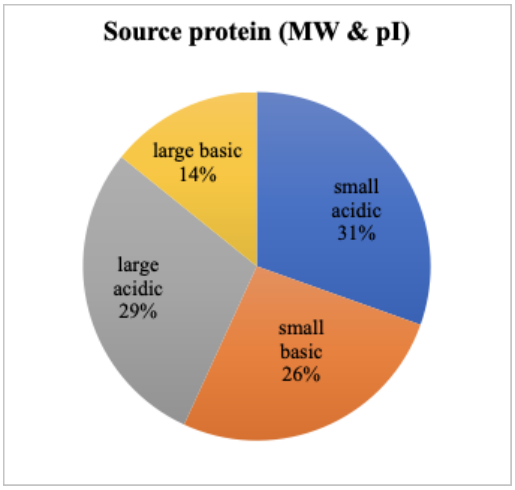


Figure 15. Molecular weight and isoelectric point distribution of degraded peptides from source protein.

From the peptides in our dataset, most peptides came from small acidic source proteins at 31%, with it coming at a close probability of 29% in large acidic proteins. Small basic source proteins (26%) made up a good majority of the proteins, with the large basic proteins being least likely (14%), as depicted in Figure 15.

The chemical properties of the N-terminal and C-terminal of each degraded peptide was also analyzed. The amino acid frequency at the N- and C-termini calculated from the total (not adjusted calculations) within the dataset and from Lenntech (Dyer, 1971) (adjusted calculations) was analyzed.

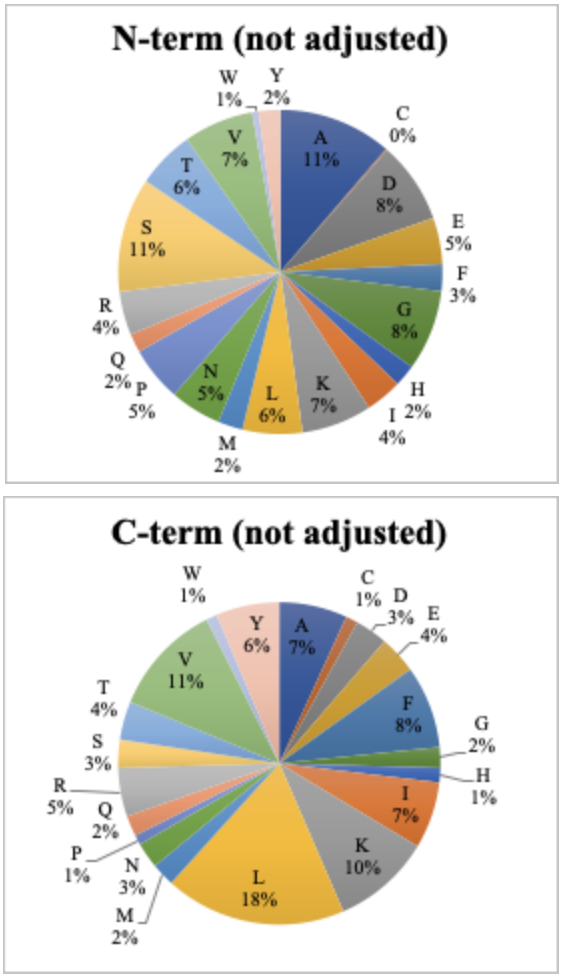


Figure 16. Amino acid frequency for the N- and C-termini (not adjusted).

Figure 16 shows that the N-terminal has the highest frequencies at around 11% for amino acids A and S and the lowest frequencies at around 1% for amino acids C and W. At the C-terminal, amino acids L (18%) and K (10%) had the highest frequencies and amino acids C, H, P, and W all had the lowest frequency at around 1%.

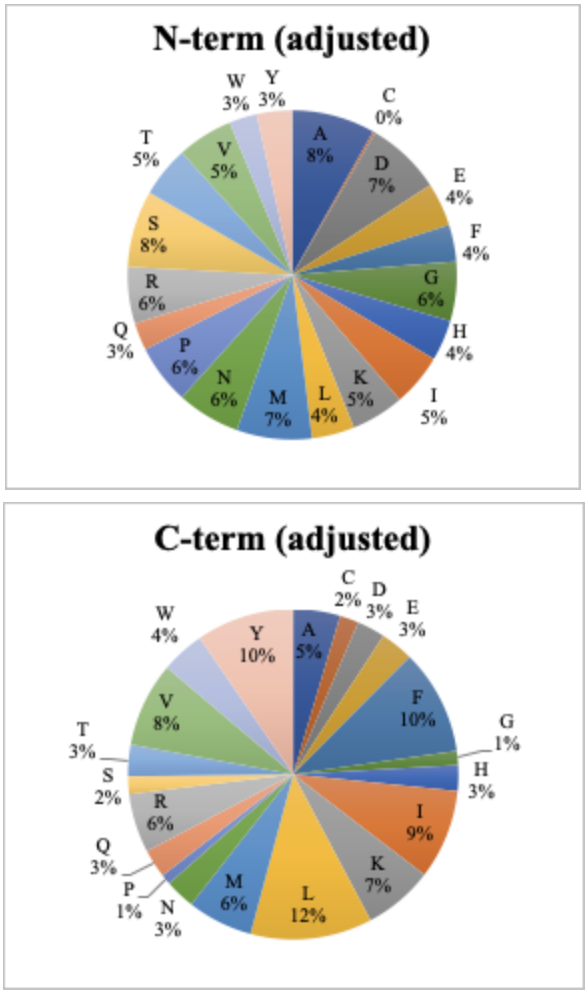


Figure 17. Amino acid frequency for the N- and C-termini (adjusted).

After looking for the frequencies of certain amino acids, Figure 17 shows that the N-terminal has the highest frequencies around amino acids A and S still (8%) and the lowest frequencies around amino acids W, X, Y, and Q (3%). For the C-terminal, the lowest frequencies are around amino acids P and G at around 1%.

From the different frequencies of certain amino acids, the reasoning behind these degradation preferences are still unknown. Therefore, a more extensive look into these patterns are analyzed through amino acid motifs in part C of the results section in this paper.

B. Antigen Degradation Preferences around Specific Secondary Structures

After running the Python scripts and Excel macros to gather information from UniProt from a dataset of 7107 intracellular peptides from 1568 proteins, it was found that of the 1568 proteins, 868 (55.36%) proteins had no defined structure in the UniProt database, 3770 peptides were unassigned (53%), and 3337 peptides (47%) had some form of assigned secondary structure. Of the 3337 peptides that had an assigned secondary structure, 1940 (58.13%) peptides had 1 secondary structure motif, 1216 (36.43%) peptides had 2 secondary structure motifs, and 180 (5.4%) peptides had 3 secondary structure motifs. Of the same 3337 peptides that had an assigned secondary structure, 2421 (72.55%) peptides had an alpha helix, 1876 (56.22%) peptides had beta strands, and 615 (18.42%) peptides had turns.

It is interesting to see the highest preference in alpha helices being the secondary structure where degradation occurs. However, it can be seen that there are multiple areas within a peptide sequence where there are more than one secondary structure, causing a duplicated count to occur in the calculations. Therefore, a future consideration to look into would be a way for the Excel macros to not double-count any secondary-structure that is seen on a similar section of the same peptide sequence. That way, the assigned secondary structure would determine the actual percentage out of the total unrepeated secondary sequence motif total.

C. Amino Acid Motif Identification

From the information that was further scraped from UniProt using the Python scripts, the frequencies of certain amino acids at the end and beginning of the N- and C-termini were calculated and visually represented by heat maps.

Figure 18. Colored heat maps with a corresponding color scale above that shows frequency of specific amino acids from 6aa before and after the N- and C-termini.

The heat maps show the amino acids on the left-side column labels with the position labels. The first 2 sections shows before (i.e. N-6, N-5) and after (i.e. N1, N2) the N-terminal and the last 2 sections shows before (i.e. N-6, N-5) and after (i.e. N1, N2) the C-terminal. In each box, there is a count of how many peptides have that specific amino acid in the corresponding position at both sides of the terminal. With the counts, a heat map with a colored scale from green, indicating lowest frequency, to red, indicating highest frequency, was created.

N-6	N-5	N-4	N-3	N-2	N-1	N1	N2	N3	N4	N5	N6	C-6	C-5	C-4	C-3	C-2	C-1	C1	C2	C3	C4	C5	C6
E	M	M	M	M	M	A	E	E	E	E	E	E	I	E	E	L	R	L	R	E	R	R	
P	R	I	I	A	F	S	D	D	D	D	D	P	I	P	Q	A	F	A	E	A	P	P	I
R	G	V	R	Q	L	M	L	P	P	P	I	D	A	V	D	Q	Y	M	M	E	K	K	K
M	A	L	Q	V	Y	D	P	G	I	I	P	I	G	L	V	K	I	K	D	L	G	I	E
I	P	A	E	R	R	N	G	I	L	A	A	L	D	E	I	L	V	S	I	P	Q	E	G

Figure 19. Top 5 amino acid motifs for the N- and C-termini.

From the color coded frequency that was displayed, a naive approach was taken to determine the top five amino acid motifs for each terminal side. From Figure 19, the top highest frequencies displayed by the number, which was also more easily viewed by the color, was taken for each position before and after the N- and C- termini. However, a bioinformatics tool called Multiple Em for Motif Elicitation (MEME) was used to more accurately identify novel-ungapped motifs within the peptide sequences (Timothy, 1994). Instead of 6 amino acids before and after the N- and C-termini, two sequences of what occurred at the first 12 amino acids and the last 12 amino acids at the N- and C- termini were analyzed for motifs.

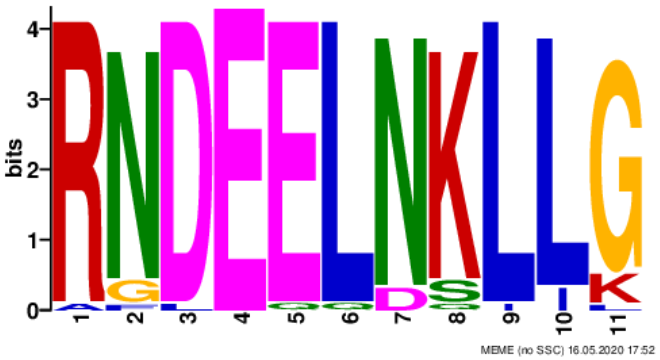


Figure 20. Top motif identified by MEME for the first 12 amino acids at the N-terminal.

Compared to Figure 19 that shows the top motifs identified by using a manual method, MEME shows very small similarities. From the MEME output, the top motif for the first 12 amino acids at the N-terminal, depicted by Figure 20, had an e-value of 3.9e-070 with 33 sites from the input of 4466 peptide sequences that had a length of at least 12 amino acids. There are some overlapping E's in both the naive identification version in Figure 19 and the MEME output in Figure 20.

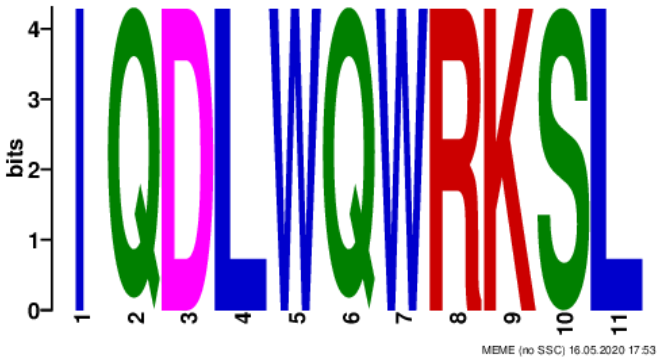


Figure 21. Top motif identified by MEME for the last 12 amino acids at the C-terminal.

From the MEME output, the top motif for the last 12 amino acids at the C-terminal, depicted by Figure 21, had an e-value of 7.7e-024 and 6 sites from the input of 4466 peptide sequences that had a length of at least 12 amino acids. There are some overlapping L's in both the naive identification version in Figure 19 and the MEME output in Figure 21.

The complexity of HLA class prediction is linked to the antigen processing pathway across the cytosol and endoplasmic reticulum, and thus these motifs shed light upon what occurs within the source-protein sequences flanking the ligands. One such study performed a comprehensive analysis on flanking motifs and discovered a decreased frequency of proline at amino acid positions 1-3 upstream of the ligands, which was the strongest among all the upstream and downstream profiles. This result from their study highlighted an important role and position of proline for inhibiting downstream epitope presentation (Hongo, 2019). With both methods of trying to identify amino acid motifs around the N- and C- termini, this knowledge is key in understanding the biology behind how T-cells recognize peptides displayed by HLA class molecules on cell surfaces. Therefore, the specific positions at which certain flanking amino acids are enriched or depleted suggests a potential bias during degradation. For the results from the naive approach in Figure 19 compared to the results from MEME in Figures 20 and 21, there does not seem to be an obvious duplicated pattern. This may be due to the fact that the first method observed 6 amino acids compared to the 12 amino acids the second time. Although there are some similar specific amino acids, there is not one exact overlap pattern-wise. Therefore, a future step could be to use a larger dataset using the same source protein, which may result in more similarities in the flanking amino acid motifs to analyze the enrichment or depletion of specific amino acids in certain positions.

non-polar, aliphatic	L, G, A, V, M, I
polar, uncharged	S, T, C, P, Q, N
positivley charged	K, R, H
negatively charged	D, E
non-polar aromatic	F, Y, W

Figure 22. Kinetic property grouping of amino acids based on Sigma Aldrich (Millapore Sigma, n.d.).

After identifying the amino acid motifs, the understanding of certain amino acid motifs, the kinetic properties were analyzed throughout the N-terminal and C-terminal.

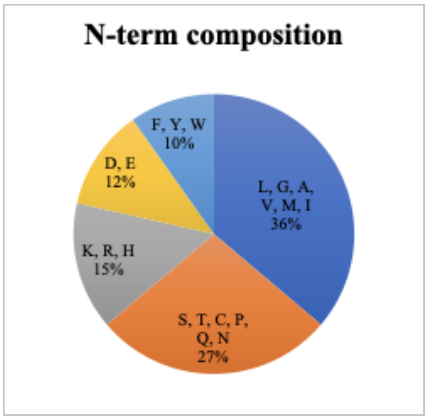


Figure 23. Composition of amino acid frequency based on kinetic properties for the N-terminal.

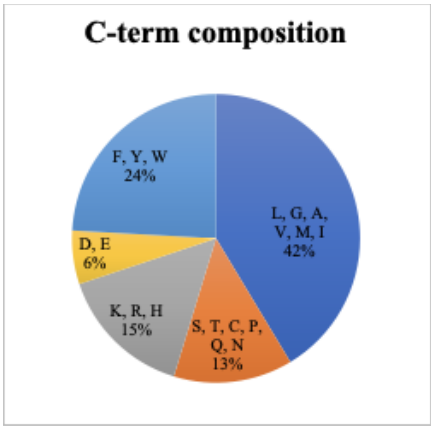


Figure 24. Composition of amino acid frequency based on kinetic properties for the C-terminal.

The two pie graphs above show the kinetic property frequency of each amino acid grouping. For the N-terminal, the composition shown in Figure 23 mainly consists of non-polar, aliphatic amino acids (L, G, A, V, M, I) with a frequency of 36% and the next highest frequency with polar, uncharged amino acids (S, T, C, P, Q, N) at 27%. For the C-terminal, the composition shown in Figure 24 mainly consists of non-polar, aliphatic amino acids (L, G, A, V, M, I) as well with a frequency of 42% and the next highest frequency with non-polar, aromatic amino acids (F, Y, W) at 24%. Further studies into the frequency of preference in amino acids' kinetic properties would be useful in identifying further degradation patterns.

V. CONCLUSION

HIV research is still being continued today due to the fact that it is still affecting people globally (Bhatti, Usman, & Kandi, 2016). Due to the importance of a better understanding of the adaptive immune system and how the body responds to viral diseases, this analysis focused on HIV-specific antigen processing and presentation machinery. Previous studies on this specific topic have suggested some type of patterns in how the antigen processing and presentation machinery works before, during, and after viral infection. However, due to the variability of HIV and its effects on the immune system, much is still not fully understood (Le Gall, 2019). Therefore, this project expanded upon the identification of patterns found in wet-lab peptide degradation experiments and developing supplementary bioinformatics tools to further study these patterns.

To understand the antigen processing machinery, wet lab experimental data was given in the form of Excel sheets. This raw data was processed and organized into databases using the MySQL Workbench platform. With this, the database was used to not only organize all of the data, but it was also used to query information, such as peptide sequences with specific parameters. From a dataset of 7107 peptide sequences derived from 1568 source proteins, Python scripts and Excel macros were utilized to identify antigen degradation patterns.

One of the first antigen degradation patterns found was the percentage of peptide sizes in amino acids, mainly ranging from 8-12 amino acids in length. This was also a confirmation of the peptide degradation lab experiments mimicking certain cellular sub-compartmental environments such as endosomes or lysosomes. Another pattern analyzed was the amount of peptides that came from each protein. Overall, most proteins had 1 corresponding peptide sequence. This suggests that there will be high variability in patterns seen across the peptides that derive from the large number of different proteins. Another pattern analyzed was the molecular weight (Mw) and theoretical pI (isoelectric point) of the source protein in which the peptides are derived from. From the peptides in the dataset, most peptides came from small acidic source proteins at 31% with a close 29% from large acidic proteins. The chemical properties of the N-terminal and C-terminal of each degraded peptide was also analyzed. From the peptide dataset, the N-terminal had the highest frequencies at around 11% for amino acids A and S and lowest frequencies at around 1% for amino acids C and W. At the C-terminal, amino acids L (18%) and K (10%) had the highest frequencies and amino acids C, H, P, and W had the lowest frequency at around 1%. After adjusting the frequencies based on the Lenntech (Dyer, 1971) amino acid frequency, it

was found that the N-terminal had the highest frequencies with amino acids A and S at around 8% and the lowest frequencies around amino acids W, X, Y, and Q (3%). For the C-terminal, the lowest frequencies were around amino acids P and G at around 1%.

The next antigen degradation pattern that was analyzed was the preference of secondary structures: alpha helices, beta strands, and turns. Overall, the peptides were mainly being degraded around alpha helices with a frequency of 72.55%. However, this percentage was not counting non-duplicated areas of a similar peptide. Therefore, a future step would be to improve the Python scripts and Excel macros so that it would take into account areas of duplicated peptide sequences where the secondary structure was already counted.

Flanking amino acid motifs were also analyzed through the creation of colored heat maps. From these heat maps, a naive approach was initially taken to identify the top 5 amino acid motifs for the N- and C- termini. However, MEME was used to more accurately identify novel-ungapped motifs within the peptide sequences. From there, motif logos were created for the first 12 amino acids at the N- and C-termini, which showed some similarity to the naive approach. With these similarities, a closer look at the kinetic properties of specific amino acids was important in trying to explain the identified amino acid patterns. Overall, the N-terminal had the highest frequency of non-polar, aliphatic amino acids at 36%. For the C-terminal, the highest frequency was from non-polar, aliphatic amino acids at 42%. This suggests that there may be certain kinetic properties that the antigen machinery prefers to degrade the peptides at. These results may expand knowledge for how to define peptide-HLA formation and T-cell responses.

In order to best understand the antigen processing and presentation machinery and how it works together with immune cells to contribute to disease recognition, other approaches may want to be considered. In the future, more advanced computational approaches and the development of further tools will be incredibly useful to more accurately detect patterns within the protein degradation experimental data. Adding more depth to the understanding of the immunopeptidome may help in generating more insightful and conclusive results regarding HIV disease targets for efficient immune clearance and further vaccine immunogen design.

VI. REFERENCES

Alberts, B. (2002). The Adaptive Immune System. In A. Johnson (Ed.), *Molecular Biology of the Cell* (4th ed.). New York: Garland Science.

Almazan, Anthony. Investigating Associations Between Antigen Processing Efficiency and HIV Control. Diss. Harvard College, 2016. N.p.: n.p., n.d. Print.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. (1990) "Basic local alignment search tool." J. Mol. Biol. 215:403-410.

Avert. (2019, October 10). The science of HIV and AIDS - overview. Retrieved May 16, 2020, from https://www.avert.org/professionals/hiv-science/overview

Berger, C. T., Llano, A., Carlson, J. M., Brumme, Z. L., Brockman, M. A., Cedeño, S., ... & Brander, C. (2015). Immune screening identifies novel T cell targets encoded by antisense reading frames of HIV-1. *Journal of virology*, *89*(7), 4015-4019.

Betts, M. R., Nason, M. C., West, S. M., De Rosa, S. C., Migueles, S. A., Abraham, J., ... & Roederer, M. (2006). HIV nonprogressors preferentially maintain highly functional HIV-specific CD8+ T cells. *Blood*, *107*(12), 4781-4789.

Bhatti, A. B., Usman, M., & Kandi, V. (2016). Current scenario of HIV/AIDS, treatment options,

and major challenges with compliance to antiretroviral therapy. *Cureus*, *8*(3).

Bjellqvist, B., Basse, B., Olsen, E. and Celis, J.E. Reference points for comparisons of two-dimensional maps of proteins from different human cell types defined in a pH scale where isoelectric points correlate with polypeptide compositions. Electrophoresis 1994, 15, 529-539.

Blum, J. S., Wearsch, P. A., & Cresswell, P. (2013). Pathways of antigen processing. *Annual review of immunology*, *31*, 443-473.

Boucau, J., & Le Gall, S. (2019). Antigen processing and presentation in HIV infection. *Molecular immunology*, *113*, 67-74.

Centers for Disease Control and Prevention. (2020, January 16). U.S. Statistics. Retrieved from https://www.hiv.gov/hiv-basics/overview/data-and-trends/statistics

Chauveau, L., Donahue, D. A., Monel, B., Porrot, F., Bruel, T., Richard, L., ... & Schwartz, O. (2017). HIV fusion in Dendritic cells mainly occurs at the surface and is limited by low CD4 levels. *Journal of Virology*.

Churchill, M. J., Deeks, S. G., Margolis, D. M., Siliciano, R. F., & Swanstrom, R. (2016). HIV reservoirs: what, where and how to target them. *Nature Reviews Microbiology*, *14*(1), 55.

Deeks, S. G., & Walker, B. D. (2007). Human immunodeficiency virus controllers: mechanisms of durable virus control in the absence of antiretroviral therapy. *Immunity*, *27*(3), 406-416.

Dinter, J., Duong, E., Lai, N. Y., Berberich, M. J., Kourjian, G., Bracho-Sanchez, E., ... & Le Gall, S. (2015). Variable processing and cross-presentation of HIV by dendritic cells and macrophages shapes CTL immunodominance and immune escape. *PLoS pathogens*, *11*(3).

Dyer, K. F. 1971. The quiet revolution: A new synthesis of biological knowledge. Journal of Biological Education 5:15-24

Goldwich, A., Hahn, S. S., Schreiber, S., Meier, S., Kämpgen, E., Wagner, R., ... & Schubert, U. (2008). Targeting HIV-1 Gag into the defective ribosomal product pathway enhances MHC class I antigen presentation and CD8+ T cell activation. *The Journal of Immunology*, *180*(1), 372-382.

Hongo, A., Kanaseki, T., Tokita, S., Kochin, V., Miyamoto, S., Hashino, Y., ... & Sato, N. (2019). Upstream Position of Proline Defines Peptide–HLA Class I Repertoire Formation and CD8+ T Cell Responses. *The Journal of Immunology*, *202*(10), 2849-2855.

History.com Editors. (2017, July 13). History of AIDS. Retrieved May 16, 2020, from https://www.history.com/topics/1980s/history-of-aids

Karlsson, A. C., Iversen, A. K., Chapman, J. M., De Oliveira, T., Spotts, G., McMichael, A. J., ... & Nixon, D. F. (2007). Sequential broadening of CTL responses in early HIV-1 infection is associated with viral escape. *PloS one*, *2*(2).

Kourjian, G., Xu, Y., Mondesire-Crump, I., Shimada, M., Gourdain, P., & Le Gall, S. (2014). Sequence-specific alterations of epitope production by HIV protease inhibitors. *The Journal of immunology*, *192*(8), 3496-3506.

Lazaro, E., Kadie, C., Stamegna, P., Zhang, S. C., Gourdain, P., Lai, N. Y., ... & Le Gall, S. (2011). Variable HIV peptide stability in human cytosol is critical to epitope presentation and immune escape. *The Journal of clinical investigation*, *121*(6), 2480-2492.

Lázaro, S., Gamarra, D., & Del Val, M. (2015). Proteolytic enzymes involved in MHC class I antigen processing: A guerrilla army that partners with the proteasome. *Molecular immunology*, *68*(2), 72-76.

Le Gall, S., Stamegna, P., & Walker, B. D. (2007). Portable flanking sequences modulate CTL epitope processing. *The Journal of clinical investigation*, *117*(11), 3563-3575.

Levy, J. A. (2001). The importance of the innate immune system in controlling HIV infection and disease. *Trends in immunology*, *22*(6), 312-316.

Llano, A., Williams, A., Olvera, A., Silva-Arrieta, S., & Brander, C. (2013). Best-characterized HIV-1 CTL epitopes: the 2013 update. *HIV molecular immunology*, *2013*, 3-25.

Malmstrøm, M., Jentoft, S., Gregers, T. F., & Jakobsen, K. S. (2013). Unraveling the evolution of the Atlantic cod's (Gadus morhua L.) alternative immune strategy. *PloS one*, *8*(9).

Millapore Sigma. (n.d.). Amino Acids Reference Charts. Retrieved May 17, 2020, from https://www.sigmaaldrich.com/life-science/metabolomics/learning-center/amino-acid-reference-chart.html

Porichis, F., & Kaufmann, D. E. (2011). HIV-specific CD4 T cells and immune control of viral replication. *Current Opinion in HIV and AIDS*, *6*(3), 174.

Rucevic, M., Kourjian, G., Boucau, J., Blatnik, R., Bertran, W. G., Berberich, M. J., ... & Le Gall, S. (2016). MHC-bound HIV peptides identified from various cell types reveal common nested peptides and novel T cell responses. *Journal of Virology*, JVI-00599.

Richardson, L. (2020). Beautifulsoup4. Retrieved May 16, 2020, from https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Santoro, M. M., & Perno, C. F. (2013). HIV-1 genetic variability and clinical implications. *ISRN microbiology*, *2013*.

Schwartz, O., Maréchal, V., Le Gall, S., Lemonnier, F., & Heard, J. M. (1996). Endocytosis of major histocompatibility complex class I molecules is induced by the HIV–1 Nef protein. *Nature medicine*, *2*(3), 338-342.

Serwold, T., Gonzalez, F., Kim, J., Jacob, R., & Shastri, N. (2002). ERAAP customizes peptides for MHC class I molecules in the endoplasmic reticulum. *Nature*, *419*(6906), 480-483.

The UniProt Consortium UniProt: a worldwide hub of protein knowledge Nucleic Acids Res. 47: D506-515 (2019)

Thompson, A. E. (2015). The immune system. *Jama*, *313*(16), 1686-1686.

Timothy L. Bailey and Charles Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers", *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pp. 28-36, AAAI Press, Menlo Park, California, 1994.

US. Department of Health and Human Services. (2019, June 27). What is a Preventive HIV Vaccine? Understanding HIV/AIDS. Retrieved May 16, 2020, from https://aidsinfo.nih.gov/understanding-hiv-aids/fact-sheets/19/96/what-is-a-preventive-hiv-vaccine-

World Health Organization. (2020, April 16). HIV/AIDS. Retrieved May 16, 2020, from https://www.who.int/gho/hiv/en/

Yaciuk, J. C., Skaley, M., Bardet, W., Schafer, F., Mojsilovic, D., Cate, S., ... & Olvera, A. (2014). Direct interrogation of viral peptides presented by the class I HLA of HIV-infected T cells. *Journal of virology*, *88*(22), 12992-13004.

Zhang, S. C., Martin, E., Shimada, M., Godfrey, S. B., Fricke, J., Locastro, S., ... & Ogbechie, O. A. (2012). Aminopeptidase substrate preference affects HIV epitope presentation and predicts immune escape patterns in HIV-infected individuals. *The Journal of Immunology*, *188*(12), 5924-5934.

VII. APPENDICES

Handling FASTA File
Convert the Word Excel file of your local sequences that you wish to study with identifiers into a FASTA file. Create a new folder in USER/blastdb and name it sequences. Save the FASTA file into this folder.

Building a BLAST database with local sequences
https://www.ncbi.nlm.nih.gov/books/NBK279688/
The makeblastdb application produces BLAST databases from FASTA files The FASTA definition lines are not parsed by makeblastdb and may be completely unstructured. The text in the definition line will be stored in the BLAST database and displayed in the BLAST report, but it will not be possible to fetch individual sequences using blastdbcmd or to limit the search with the –seqidlist option. Use the –parse_seqids flag when invoking makeblastdb to enable retrieval of sequences based upon sequence identifiers. Each sequence must have a unique identifier, and that identifier must have a specific format. It should begin right after the ">" sign on the definition line and contain no spaces.

```
$ cat sequences/sequence2.fasta
>1
AIDWLTG
>2
AQTYSPS
>3
AFRKFLPL
```

Makeblastdb can be invoked for this file as below.

```
$ makeblastdb -in sequences/sequence2.fasta -parse_seqids -dbtype prot
Building a new DB, current time: 05/12/2017 14:08:59
New DB name:   /Users/legalllab/blastdb/sequence2.fasta
New DB title:  sequence2.fasta
Sequence type: Protein
Deleted existing Protein BLAST database named /Users/legalllab/blastdb/sequence2.fasta
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 19 sequences in 0.00102186 seconds.
dhcp-172-21-66-152:blastdb legalllab$ cat sequence2.fasta
$
```

Execution
To execute the protein-protein comparison, invoke the following command into the terminal app.

```
$ blastp -task blastp-short -db NCBI/swissprot -query sequences/sequence2.fasta -out results/sequence2results.xls -outfmt "7 qseqid qseq qlen sseqid sacc slen score ppos"
```

```
$ blastp -task blastp-short -db swissprot –remote –entrez_query "Homo sapiens [Organism]" -query sequences/sequence2.fasta -out results/sequence2results.xls -outfmt "7 qseqid qseq qlen sseqid sacc slen score ppos"
```

*Sometimes, it is suggested to type this out instead of copying and pasting because the Terminal might reformat the next line and it might read it incorrectly. This is if you get an error saying "Too many positional arguments (1)" after copying and pasting. This is just a side note.

Explanation of command line

-task An option with blastp-short optimized for query sequences shorter than 30 residues
-db Refers to the NCBI database, swissprot in this experiment
-query Refers to the subject that will be compared to NCBI database = FASTA file with local sequences
-out Refers to the output and in this case, it will output as an Excel file that will show up in the blastdb folder
-outfmt Refers to customized format of output; 7 is output as tabular with comment line, qseqid is query sequence id, qseq is query sequence, qlen is query length, sacc is subject accession, score is raw score, ppos is percentage of positive-scoring matches"

Results
Insert a column between columns B and C. Type in the formula "=len(cell)" and for cell, replace that with the referenced cell with the query sequence.
This will allow you to see whether the query sequence is the same length as column D, that outputs query length.
It is also useful to look at sequences that output 100% for positive-scoring matches.
These are some of the ways that you can analyze the data you are given in order to calculate for accuracy in accession numbers.

Figure 25. Protocol on how to use BLASTp in batches for local sequences.

1. Format the Excel File in the A and B columns (unique identifier and query sequence) as follows:

	A	B
1	identifier	sequence
2	1	AVIDKYV
3	2	ASGPKRT
4	3	APSSFVF
5	4	APGLPEL
6	5	ALQAIVE

2. In column C, use the concatenate command as follows, making sure the highlighted boxes are corresponding to the correct cell.
=CONCATENATE(">",A2, " ", CHAR(13),B2)
***Use CHAR(10) for Windows and CHAR(13) for Mac devices.
After entering and producing a value, hover over bottom right corner of cell and drag down to apply the formula to all sequences.

	A	B	C
1	identifier	sequence	
2	1	AVIDKYV	=CONCATENATE(">",A2," ",CHAR(13),B2)
3	2	ASGPKRT	>CONCATENATE(text1, [text2], [text3], [text4], [text5], [text6], ...) ASGPKRT

3. Highlight all of column C and under the Home tab, click on "Wrap Text" which should make the output go from a single line of text to what is formatted as follows:

	A	B	C
1	identifier	sequence	
2	1	AVIDKYV	>1 AVIDKYV
3	2	ASGPKRT	>2 ASGPKRT
4	3	APSSFVF	>3 APSSFVF
5	4	APGLPEL	>4 APGLPEL
6	5	ALQAIVE	>5 ALQAIVE

4. Highlight all of column C and copy and paste it into a Word document. Save this file with an appropriate name and save it as Plain text (.txt).

If you wish to use these sequences on the BLAST+ app to BLAST all sequences in a batch, save it into a created folder called sequences on the blastdb folder in your home directory (USER/blastdb/sequences).

This step of copying and pasting into a Word document is necessary because if the column is copy and pasted straight into a text file, there are unnecessary quotes that show up in the format. This step eliminates the quotes. This is an example of what the Word document should look like.

>1
AVIDKYV
>2
ASGPKRT
>3
APSSFVF
>4
APGLPEL
>5
ALQAIVE

5. Locate it on the computer and replace .txt with .fasta. Save changes. Open the file if you wish to check for correct formatting. It should look like the following, which is correct FASTA formatting.

6. You are ready to use the BLAST+ app with your FASTA file now!

Figure 26. Protocol for creating FASTA file out of peptide sequences in Excel.

Creating Text File of UniProt URLs

1. From the Excel file with the list of accession numbers, highlight the column ("cmd-shift-downarrow") of non-duplicated accession numbers grouped by decreasing number of peptides per protein. Copy and paste that into a Word document.
2. Delete column title "Accession number."

3. Scroll all the way down to the last accession number. Notice that there may be many empty rows after that. If there are, add a page break after the very last accession by going to Layout → Break → Page. The rest of the column will show up on the following page. Go to that page, click on the 4-way arrow drag icon and delete it.
4. Save document as accessions.txt (change file name according to what you want) and check "Insert line breaks", "End lines with 'CR/LF'", and "Allow character substitution". Go to text file and delete any empty space in the beginning of the file and any extra UniProt URLs without accession numbers.

5. Go on Python script **accession_numbers_into_uniprot_url.py**.
6. On line 5, change name and path of file 'accessions.txt' with the name of the text file with the accession numbers you just created in the previous steps.

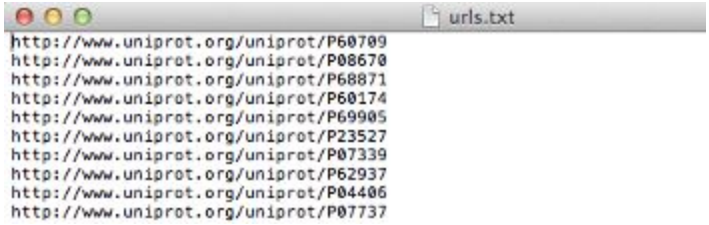
```
    #change 'accessions' to name of created file with accession numbers and change os path (optional)
with open('/Users/legalllab/Documents/CODE/accessions.txt', 'r') as f:
```

7. On line 13, change 'urls.txt' to a name that you want the new text file of UniProt URLs to be. You can save it in a specific place on your device is you specify the path.

```
    #change 'urls' to whatever you want to name your file and change os path (optional)
with open('/Users/legalllab/Documents/urls.txt', 'w') as p:
```

8. Run script with F5 or click Run → Run Module on the taskbar.
9. A file with the name you just replaced 'urls.txt' with will be created in the Documents folder. You can save it in a specific place on your device is you specify the path.

Your text file should look like the following:



Querying UniProt URLs for Secondary Structure Information

1. Go on Python script **secondary_structure_through_multiple_uniprot_urls.py**.
2. On line 5, change name of file 'accessions.txt' with the name of the text file with the accession numbers you just created in the previous steps.

```
    #change 'urls' to the name of your created file of all UniProt URLs
quote_page = open('urls.txt')
```

3. (OPTIONAL STEP) If you want to check and see that the script is reading the correct file of URLs, you can run with F5 or Run → Run Module on the taskbar. After the first couple of proteins run successfully, you can kill the program.

Transferring Python Script Information Onto Excel File

1. On Terminal, change path to the location of **secondary_structure_through_multiple_uniprot_urls.py** like the following:

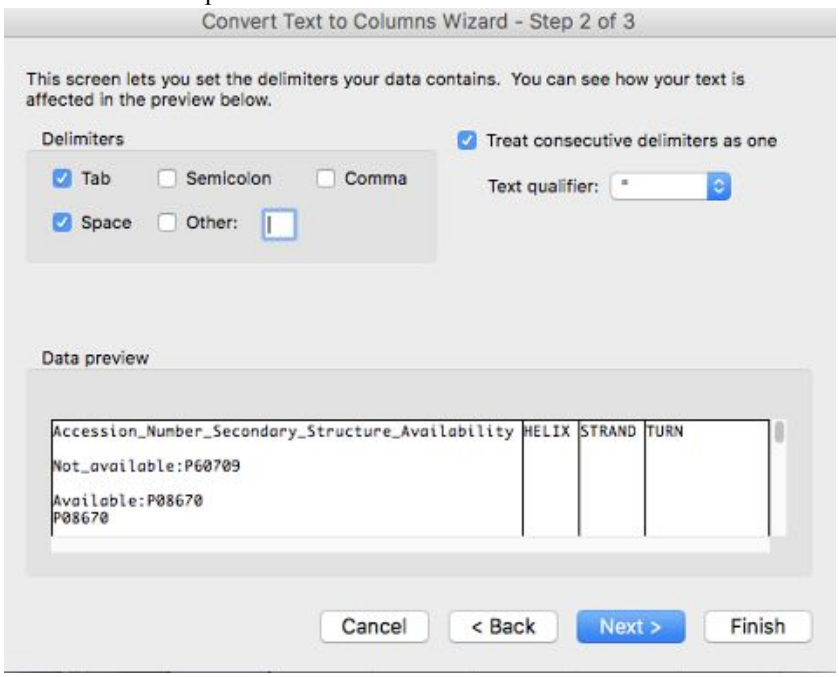
$ cd $HOME/PycharmProjects

2. Type into the command line the following where you can change the name of your Excel file to a name you think is appropriate to easily identify later in your files.

$ python PDB/secondary_structure_through_multiple_uniprot_urls.py > PDBResults/secstruc_name_date.xls

3. You may need to wait a while (and by while I mean like about 35 minutes for 1500 peptides so the more peptides, the longer amount of time unfortunately) for this to run (until Terminal requests for another command denoted by $) before opening the Excel file with all the secondary structure information.

4. On the Excel file, separate the information into separate columns by selecting the column and going to Data → Text to Columns → check Delimited → check Space and Treat consecutive delimiters as one → Next → Finish.



5. To reformat the columns, delete one empty row from rows B through D in order to shift the cells up.

6. Highlight all of Column C, D, and E, right-click the mouse and select “Format Cells…” Under “Number”, choose “Custom” and type in “mm-dd”. The Excel sheet should look something like the following right now.

	A	B	C	D
1	Accession_Number_Secondary_S	HELIX	STRAND	TURN
2				
3	Not_available:P60709			
4				
5	Available:P08670			
6	P08670	102-135	238-241	244-248
7				
8	P08670	167-235	None	None
9				
10	P08670	266-334	None	None
11				
12	P08670	385-405	None	None
13				
14				
15	Available:P68871			
16	P68871	6-17	None	78-80
17				
18	P68871	21-35	None	None
19				
20	P68871	37-42	None	None
21				
22	P68871	44-46	None	None

7. Highlight columns A through D, click the “Filter” button under the Data section, and filter by checking “Select All” first, then scrolling to the bottom of this list and unchecking “Blanks” and “Successful!”

8. Open a new sheet and copy and paste columns A-D. Highlight columns A through D, click the “Filter” button under the Data section, and filter with “Does Not Contain: Available” and “Does not contain: Not_available” like the following.

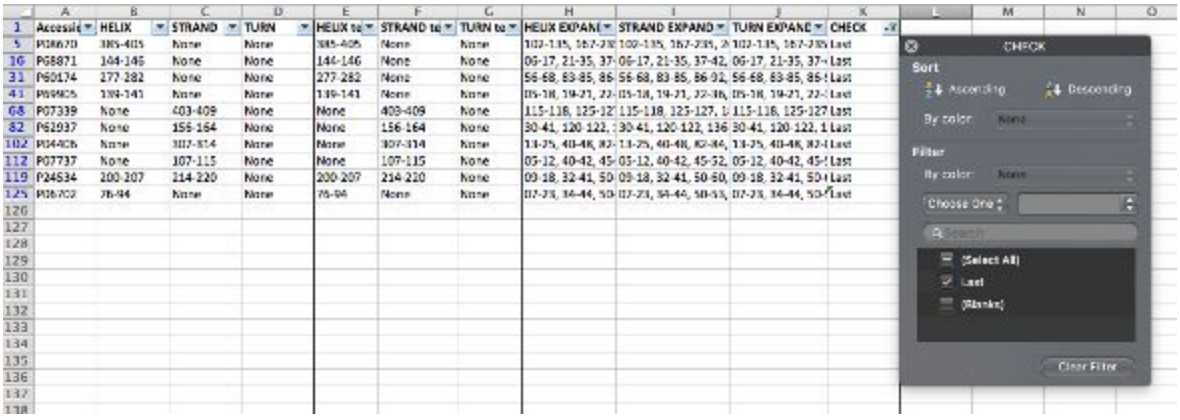
Using “Matching Secondary Structure.xlsm”

1. Open the Excel sheet named “Matching Secondary Structure” and copy and paste columns A-D into the sheet named “edit_output”.

2. In the following cells, these formulas are already entered in so just highlight across B2-K2 and double-click on the right-bottom corner of the cell to copy the formula down the column.

E2: =TEXT(B2,"mm-dd"), **F2**: =TEXT(C2,"mm-dd"), **G2**: =TEXT(D2,"mm-dd"), **H2**: =IF(A2=A1,H1&", "&E2, E2)
I2: =IF(A2=A1,I1&", "&F2, F2), **J2**: =IF(A2=A1,J1&", "&G2, G2), **K2**: =IF(A2<>A3, "Last", "")

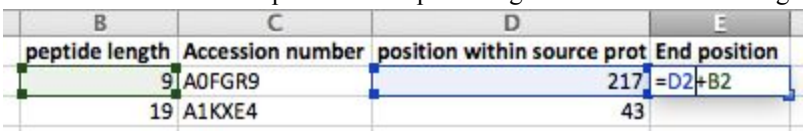
3. Highlight all columns. Under the “Data” tab, select “Filter”. Under the “CHECK” or K column, uncheck “Blanks” in the drop down menu.



4. Highlight columns H through K, press “command-F” and replace all “None” with blanks. Copy columns A(Accession_Number_Secondary_Structure_Availability), H(HELIX RANGE), I(STRAND RANGE), and J(TURN RANGE) into columns A-D in the “ref” sheet. Select entire columns B through D.

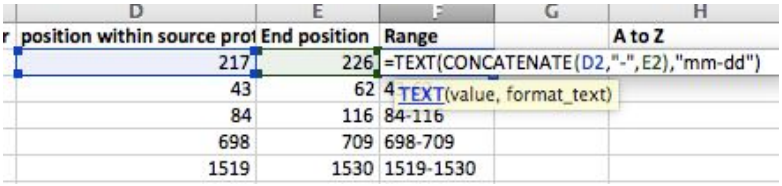
Useful trick: Whenever you want to select information down a column until the very last cell with information inside, click on the column title or drag across the column titles you want to copy and then press "Command-shift-down arrow".

5. From the Excel sheet with all the given data, copy the "Peptide sequences" and "Accession number" information into the columns E and F. In that same Excel sheet, make a column named "End position" and type in a formula that adds numbers in the columns "Position within source prot" and "Peptide length" such as the following.



B	C	D	E
peptide length	Accession number	position within source prot	End position
9	A0FGR9	217	=D2+B2
19	A1KXE4	43	

6. Make another column named "Range" and type in a formula that concatenates the numbers in the columns "Position within source prot" and "End position" such as the following. Copy and paste special as "Values only" the range into columns G and H of the "ref" sheet in the "Matching Secondary Structure" Excel sheet.



D	E	F	G	H
position within source prot	End position	Range		A to Z
217	226	=TEXT(CONCATENATE(D2,"-",E2),"mm-dd")		
43	62	4 TEXT(value, format_text)		
84	116	84-116		
698	709	698-709		
1519	1530	1519-1530		

7. *(This step is already completed for the user but this is to explain where the formula comes from)*
On the sheet, "ref", click the button labeled "Find H,S,T". This is what happens:
Under the Developer tab, open Editor. Right click "Sheet 6(ref)" → Insert Module. Copy and paste the following VBA code.

```
Function NumRange(v)
    Dim arrC, arr, x As Long, rv As String, sep As String, e

    arrC = Split(v, ",")
    rv = ""

    For Each e In arrC
    If InStr(e, "-") Then
    arr = Split(e, "-")
    arr(0) = Trim(arr(0))
    arr(1) = Trim(arr(1))
    If IsNumeric(arr(0)) And IsNumeric(arr(1)) Then
    For x = CLng(arr(0)) To CLng(arr(1))
        rv = rv & sep & x
        sep = ","
    Next x
    End If
    ElseIf IsNumeric(e) Then
    rv = rv & sep & CLng(e)
    sep = ","
    End If
    Next e
    NumRange = rv
End Function
```

8. In the "RANGE EXPAND" column, type in "NumRange(cell)". For cell inside the formula, select the adjacent cell under the "Range" column. The formula will already be entered in H2, so just double-click the bottom-right corner of the cell. The result will turn out to look like the following:
In the following cells, these formulas are already entered in so just highlight I2-N2 and double-click on the right-bottom corner of the last cell to copy the formula down the columns.

I2 (Find Helix): =IFERROR(VLOOKUP(F2,$A:$D,2,FALSE),""), **J2 (HELIX EXPAND)**: =NumRange(I2)
K2 (Find Strand): =IFERROR(VLOOKUP(F2,$A:$D,3,FALSE),""), **L2 (STRAND EXPAND)**: =NumRange(K2)
M2 (Find Turn): =IFERROR(VLOOKUP(F2,$A:$D,4,FALSE),""), **N2 (TURN EXPAND)**: =NumRange(M2)

9. On the sheet, "range", click on "Copy Peptides and Expand" which copies the peptide sequences accession numbers, and column "RANGE EXPAND" from the "ref" sheet.
10. On the sheet, "ss", click on "Copy Columns A and B" which copies the peptide sequences accession numbers into the "ss", "match" and "results" sheet.
11. On the sheet, "ss", click on "Copy H" which copies the column "HELIX EXPAND" from the "ref" sheet. It will expand the helix ranges so that each number is in a separate cell across rows. When you come back to this step for strands and turns, click on "Clear H,S,T" to delete the secondary information and replace by clicking on "Copy S" or "Copy T", depending on which structure you are concentrating on.
12. On the sheet, "results", the following formula is in C7 (H, S or T). Double-click the bottom-right corner of the cell to apply the formula down the column. Copy and paste the "H,S,or T" column into the corresponding structure column.

=match!C7&IF(match!D7<>"","&match!D7,"")&IF(match!E7<>"","&match!E7,"")&IF(match!F7<>"","&match!F7,"")&IF(match!G7<>"","&match!G7,"")&IF(match!H7<>"","&match!H7,"")&IF(match!I7<>"","&match!I7,"")&IF(match!J7<>"","&match!J7,"")&IF(match!K7<>"","&match!K7,"")&IF(match!L7<>"","&match!L7,"")&IF(match!M7<>"","&match!M7,"")&IF(match!N7<>"","&match!N7,"")&IF(match!O7<>"","&match!O7,"")&IF(match!P7<>"","&match!P7,"")&IF(match!Q7<>"","&match!Q7,"")&IF(match!R7<>"","&match!R7,"")&IF(match!S7<>"","&match!S7,"")&IF(match!T7<>"","&match!T7,"")&IF(match!U7<>"","&match!U7,"")&IF(match!V7<>"","&match!V7,"")&IF(match!W7<>"","&match!W7,"")&IF(match!X7<>"","&match!X7,"")&IF(match!Y7<>"","&match!Y7,"")&IF(match!Z7<>"","&match!Z7,"")&IF(match!AA7<>"","&match!AA7,"")&IF(match!AB7<>"","&match!AB7,"")&IF(match!AC7<>"","&match!AC7,"")&IF(match!AD7<>"","&match!AD7,"")&IF(match!AE7<>"","&match!AE7,"")&IF(match!AF7<>"","&match!AF7,"")&IF(match!AG7<>"","&match!AG7,"")&IF(match!AH7<>"","&match!AH7,"")&IF(match!AI7<>"","&match!AI7,"")&IF(match!AJ7<>"","&match!AJ7,"")&IF(match!AK7<>"","&match!AK7,"")&IF(match!AL7<>"","&match!AL7,"")&IF(match!AM7<>"","&match!AM7,"")&IF(match!AN7<>"","&match!AN7,"")&IF(match!AO7<>"","&match!AO7,"")&IF(match!AP7<>"","&match!AP7,"")&IF(match!AQ7<>"","&match!AQ7,"")&IF(match!AR7<>"","&match!AR7,"")&IF(match!AS7<>"","&match!AS7,"")&IF(match!AT7<>"","&match!AT7,"")&IF(match!AU7<>"","&match!AU7,"")&IF(match!AV7<>"","&match!AV7,"")&IF(match!AW7<>"","&match!AW7,"")&IF(match!AX7<>"","&match!AX7,"")&IF(match!AY7<>"","&match!AY7,"")&IF(match!AZ7<>"","&match!AZ7,"")&IF(match!BA7<>"","&match!BA7,"")&IF(match!BB7<>"","&match!BB7,"")&IF(match!BC7<>"","&match!BC7,"")&IF(match!BD7<>"","&match!BD7,"")&IF(match!BE7<>"","&match!BE7,"")&IF(match!BF7<>"","&match!BF7,"")&IF(match!BG7<>"","&match!BG7,"")&IF(match!BH7<>"","&match!BH7,"")&IF(match!BI7<>"","&match!BI7,"")&IF(match!BJ7<>"","&match!BJ7,"")&IF(match!BK7<>"","&match!BK7,"")&IF(match!BL7<>"","&match!BL7,"")&IF(match!BM7<>"","&match!BM7,"")&IF(match!BN7<>"","&match!BN7,"")&IF(match!BO7<>"","&match!BO7,"")&IF(match!BP7<>"","&match!BP7,"")&IF(match!BQ7<>"","&match!BQ7,"")&IF(match!BR7<>"","&match!BR7,"")&IF(match!BS7<>"","&match!BS7,"")&IF(match!BT7<>"","&match!BT7,"")&IF(match!BU7<>"","&match!BU7,"")&IF(match!BV7<>"","&match!BV7,"")&IF(match!BW7<>"","&match!BW7,"")&IF(match!BX7<>"","&match!BX7,"")&IF(match!BY7<>"","&match!BY7,"")&IF(match!BZ7<>"","&match!BZ7,"")

Note: DO NOT DELETE OR ALTER ANY CELL FORMULAS FROM THE "MATCH" OR "RESULTS" SHEET.
These are the formulas in the "match" sheet for reference and future applications.

13. Follow steps 10-12 for "Strands" and "Turns". For a repeat for strands and turns, you do not need to change any formulas for step 12. All you need to do it copy and paste the column into the corresponding "Strands" or "Turns" column in the "results" sheet.

Figure 27. Protocol for analyzing secondary structure using .txt file of Uniprot URLs.

```
####
# Le Gall Lab at Ragon Institute of MIT, Harvard, and MGH
# Ann Le
# June 2017
# Last Updated: June 2017
# Protocol Name: Protein Information Through Multiple UniProt URLs
# This code opens the text file with the PDB accession numbers and then creates
# another text file with uniprot urls for all of the accession numbers
####
```

```
# open text file with all the accession numbers
# change 'accessions' to name and path of created file with accession numbers
with open('/Users/legalllab/Documents/PDB/Example Files/quentin.txt', 'r') as f:
  accession = f.readlines()  # read lines of accession file

# add the accession number to the end of the url
  website = ['http://www.uniprot.org/uniprot/' + line for line in accession]
  print(website)

# open text file called urls
# change name and path of 'urls' to whatever you want to name your file with the uniprot urls
with open('/Users/Desktop/HLAA.txt', 'w') as p:
  p.writelines(website) # write websites into the new file

f.close() # close accessions.txt
p.close() # close urls.txt
```

Figure 28. Python code for how to open a text file of PDB accession numbers and creates another file of UniProt URLs for each accession number.

```
####
# Le Gall Lab at Ragon Institute of MIT, Harvard, and MGH
# Ann Le
# June 2017
# Last Updated: June 2017
# Protocol Name: Protein Information Through Multiple UniProt URLs
####

from bs4 import BeautifulSoup
import urllib2
import re
import csv

####open text file of all urls for uniprot
# change 'urls' to the name of your created file of all UniProt URLs
quote_page = open('/Users/Ann Le/Documents/R_at_Ragon/urls.txt')

####print column titles
print ('Accession_Number_Secondary_Structure_Availability ' + \
     'HELIX ' + 'HSEQ ' + 'STRAND ' + 'TURN')

####for loop to put all information into dictionary
for pg in quote_page:
   print() # have a break between each protein
   ####query website and return the html to the  variable 'site'
   page = urllib2.urlopen(pg).read()
   ####parse html using beautiful soup and store in variable 'soup'
   soup = BeautifulSoup(page, 'html.parser')

   ####get the peptide information
   span_id = soup.findAll('span', {'id': 'entrySequence'})
   ####get the secondary structure
```

```
    second_structure = soup.find('div', {'id': 'secondarystructure'})
    if second_structure:  # when you find secondarystrucutre
        for accessions in span_id:
            print ('Available:' + accessions.getText()[4:10])  # print message
    else:  # if you can't find secondarystructure
        for accessions in span_id:
            print ('Not_available:' + accessions.getText()[4:10])  # print message


     ####print secondary structure information in 6 separate columns
    list1 = []
    list2 = []
    list3 = []
    list4 = []

    for helix in soup.findAll('rect', attrs={'class': 'HELIX'}):
        helices = helix['title']
        list1.append(helices[6:])  # add helices info into list1

    for strand in soup.findAll('rect', attrs={'class': 'STRAND'}):
        beta_sheets = strand['title']
        list3.append(beta_sheets[7:])  # add betasheet/strand info into list2

    for turn in soup.findAll('rect', attrs={'class': 'TURN'}):
        turns = turn['title']
        list4.append(turns[5:])  # add turns info into list3

    for a, b, c, d in map(None, list1, list2, list3, list4):
        for accessions in span_id:
            print (accessions.getText()[4:10])
            print (' ', a, b, c, d)

print ("Successful!")  # will print at the bottom if everything is printed out
```

Figure 29. Python code for how to scrape secondary structure information into a .csv file from a text file of UniProt URLs

```
####
# Le Gall Lab at Ragon Institute of MIT, Harvard, and MGH
# Ann Le
# June 2017
# Last Updated: July 2018
# Protocol Name: Protein Information Through Multiple UniProt URLs
# This code opens the text file with the Uniprot URLs and then creates
# another text file with information about the peptide.
####

from bs4 import BeautifulSoup
import urllib
import urllib2
import re
import csv
```

```
def get_protein_information (uniprot_urls_txt):

    # open text file of all urls for unipro
    quote_page = open('/Users/legalllab/Documents/PDB/PeptideIDresults.txt')

    # print column titles
    print 'Accession_number ' + 'Protein_ID ' + 'Sequence ' + 'Length ' + 'Mass(Daltons)'

    # for loop to put all information into dictionary
    for pg in quote_page:
        print  # have a break between each protein
    # query website and return the html to the  variable 'site'
        page = urllib2.urlopen(pg).read()
    # parse html using beautiful soup and store in variable 'soup'
        soup = BeautifulSoup(page, 'html.parser')

        list1 = []
        list2 = []
        list3 = []
        list4 = []
        list5 = []

    # get the peptide information
        span_id = soup.find('span', {'id': 'entrySequence'}).getText()
        accessions = span_id[4:10]
        # print
        #  accessions
        list1.append(accessions)

    # get the protein information
        id_name = soup.find('title').getText()
        protein = id_name[:id_name.find('-')]
        # print protein
        list2.append(protein)

    # get the sequence information
        seq = soup.find('span', {'id': 'entrySequence'}).getText()
        seqs = seq.split('\n')
        seqss = seqs[1:]
        seqsss = ''.join(seqss)
        # print seqsss
        list3.append(seqsss)

    # get length and mass information
        len_and_mass = soup.find('div', attrs={'class': 'sequence-isoform-rightcol'})
        spans = len_and_mass.findAll('span')
        for length in spans[1]:
            # print length
            list4.append(length)
        for mass in spans[3]:
            # print mass
            list5.append(mass)
```

```
        for a, b, c, d, e in zip(list1, list2, list3, list4, list5):
            print a, b, c, d, e

    print
    print "Successful!"  # will print if everything is successful


# change 'uniprot_urls_txt' to the name of your created file of all UniProt URLs
get_protein_information ('/Users/legalllab/Documents/PDB/UniprotURLS/.txt')
```

Figure 30. Python code for how to open a text file with UniProt URLs and creates another text file with information about the peptide.