**CIS 4900 – Research Project**

**Irenaeus Chan**

**0797453**

**Protein Structure Analysis**

**Introduction**

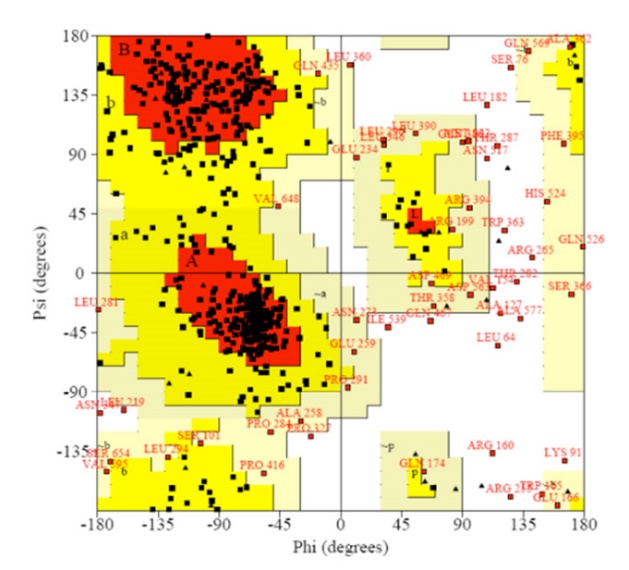
Protein structure prediction is the prediction of the three-dimensional structure of a protein from its amino acid sequence. Simply put, it is the prediction of its folding to determine the secondary, tertiary, and quaternary structure from its most simplistic form. The current field is fairly important in the scientific community due to the relation between structure and function. As the protein’s structure is closely linked with its functionality, understanding the structural genomics of a protein has the potential to infer knowledge of the protein which can lead to large leaps in understanding novel protein folds and discover potential targets for drugs and other medical research. As it stands today, accurate protein structure prediction is one of the most prized research with bi-annual competitions such as CASP causing entire groups of scientists to suspend current research to participate and improve their identification methods.

The large majority of protein structures are currently determined by atomic techniques such as X-ray crystallography and nuclear magnetic resonance (NMR). However, as with most techniques, there are limitations. X-ray crystallography sometimes distorts portions of the structure and current techniques still find it relatively difficult to differentiate whether a part of the protein belongs to the protein or is a bound water oxygen or metal ion. NMR is limited to smaller molecules of about 30 kD due to the unique limitation of requiring molecules to be perfectly crystallized. Even so, these experimental techniques are still more accurate than the computation methods that have been developed such as homology modelling, protein threading, and *de novo* protein structure prediction.

This forms the background for this project. With the importance of finding alternative and accurate ways to predict protein structures. As well as structural genomics being an integral part of the field of bioinformatics, I will attempt to investigate, utilizing various algorithms and computational metrics, patterns associated with proteins and attempt to categorize and determine repeating patterns that could potentially be used for further analysis. This report will specifically summarize the results of the findings as well as the steps that have been taken to reach the current point of this continued project.

**Analysis Techniques – Ramachandran Plot**

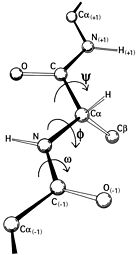
I have chosen to use Ramachandran plots as a form of examining possible patterns within proteins. Ramachandran plots were developed in 1963 by G. N. Ramachandran, *et al.* In a polypeptide chain the main chain N-Calpha­ and Calpha-C bonds are relatively free to rotate about. These rotations are represented by the torsion angles phi (φ) and psi (ψ) respectively (Fig. 2). Ramachandran *et al.* categorized these phi and psi angles by utilizing small polypeptides to systematically vary the phi and psi angles to look for stable conformations. For each conformation, the structure was examined for close contact between atoms. Each atom was treated as a hard sphere with dimensions corresponding to their van der Waals radii. Therefore, angles which caused spheres to collide resulted in disallowed conformations which ultimately gave us our present day plot showing the allowed conformations for the backbone chain (Fig. 1). The only amino acid that escapes this rule is glycine which lacks a side chain, therefore making it unique to the rule (Fig. 1).



**Fig 1.A Ramachandran plot generated from homology modelling.** White areas correspond to conformations where atoms in the polypeptide come closer than the sum of their van der Waals radii and red regions correspond to conformations where there are no steric clashes denoting alpha-helical and beta-sheet conformations.

Source: <http://www.embl.de/chemcore/chemcore_services/computational_chemistry/chemoinformatics/chemoinformatics_3l.jpg>

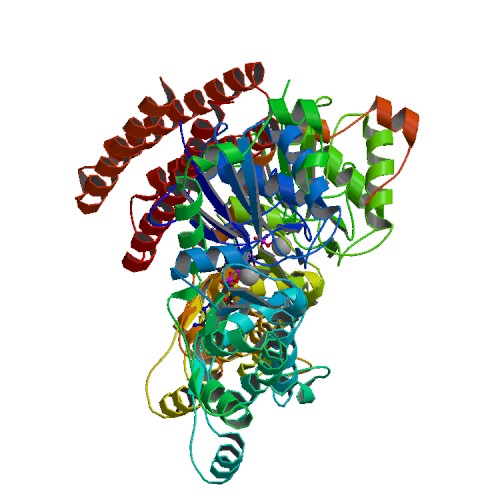
I chose this as my starting point as understanding torsion angles is an important step to understanding protein folding due to the nature of the plot itself being a good model for tertiary prediction. To explain, the torsion angles in the backbone provide the flexibility required for the polypeptide to adopt a certain fold. However, since the third possible torsion angle, C-N (ω) within the backbone is fixed to 180° as the partial double-bond character of the peptide bond restricts the rotation, it places the successive Calpha carbons as well as C, O, N, and H between them in a single plane that is easy to predict (Fig. 2). Which means the rotation of the protein chain can simply be described as the rotation of the peptide bond planes, phi and psi, relative to each other.



**Fig 2. The backbone dihedral angles phi (**φ**) and psi (**ψ**) as well as the omega angle (**ω**) in the backbone of an amino acid.** The phi angle lies between the N-Calpha bond, the psi angle lies between the Calpha­-C bond and the omega bond lies between the N-C bond.

Source: <https://upload.wikimedia.org/wikipedia/commons/thumb/c/c0/Protein_backbone_PhiPsiOmega_drawing.jpg/140px-Protein_backbone_PhiPsiOmega_drawing.jpg>

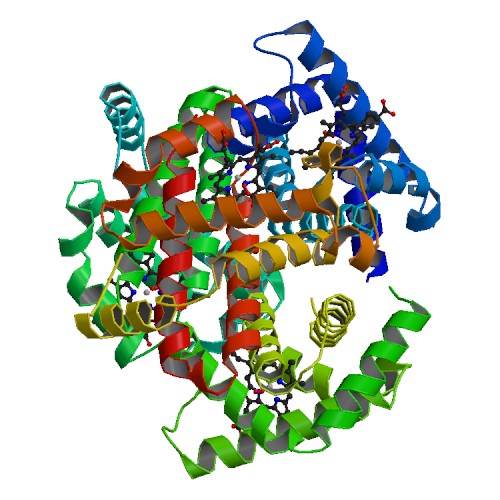
While there are many online Ramachandran plot generators that can be used, I have decided to create my own for this project. Instructions on how to use the program as well as any further information that is required will be addressed later on in this report under the “Code” section. Figure 3, shows an example of the output from the code for the 3JBI pdb file for an MDFF model of the vinculin tail domain bound to F-actin.



**Fig 3. Ramachandran plot generated using 4900.py combined with Excel 2013 graphing program for the PDB file, 3JBI of an MDFF model of the vinculin tail domain bound to F-actin.** A 3-D representation of the model is shown on the right and its corresponding phi and psi angles for the backbone is on the left with the corresponding amino acids in colour.

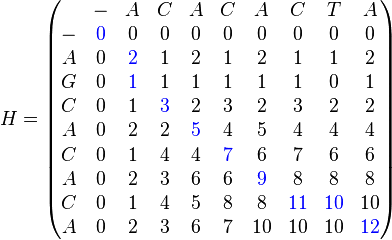
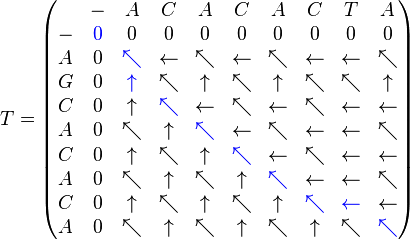
**Data Set Organization**

As a focus for the project I have specifically decided to focus on the patterns associated with mutations in the amino acid sequence. Specifically, I am interested in the possible conformational changes associated with indels and mutations that may affect local or global arrangements within the protein itself. The protein that I decided to explore and focus this project on was, hemoglobin. Hemoglobin is an important oxygen-transporting metalloprotein found in the red blood cells of all vertebrates and some invertebrates. Additionally, the structure of hemoglobin is fairly simple and lacks much variation, being primarily made of alpha-helices creating four heme subunits around a central iron ion. I specifically chose the hemoglobin 1GZX from the Protein Data Bank as my base hemoglobin which can be found at: <http://www.rcsb.org/pdb/explore.do?structureId=1gzx>.



**Fig 4. Oxy T state human haemoglobin: oxygen bound at all four haems.** Original Authors: Paoli, M., Liddington, R., Tame, J., Wilkinson, A., Dodson, G.

Using the BLAST algorithm hosted by the National Center for Biotechnology Information (NCBI) found at: <http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastHome> I selected 30 hemoglobin proteins on the PDB records. Of the 30 hemoglobin, 10 proteins that had a 98% or higher identical score, 10 proteins that had a 65% to 72% identical score, and 10 proteins that had less than 50% identical score were chosen. The reason for doing so was to examine the changes in the protein as the sequence differed more and more from the original base hemoglobin. Of the proteins that had a 98% or higher identical score, there were maximum four indels and some mutations in the sequence compared to the proteins that had a 65% to 72% identical score having a maximum of 40 indels and even greater mutations. In order to make sure the comparison of the Ramachandran plots made between the two PDB files were accurate in their representation as discussed later, a local alignment algorithm (smith-waterman) was used between the base 1GZX hemoglobin and the resulting 30 hemoglobin obtained through BLAST (Fig. 5). The smith-waterman algorithm was created by myself and instructions on how it works will be discussed in the “Code” section below.

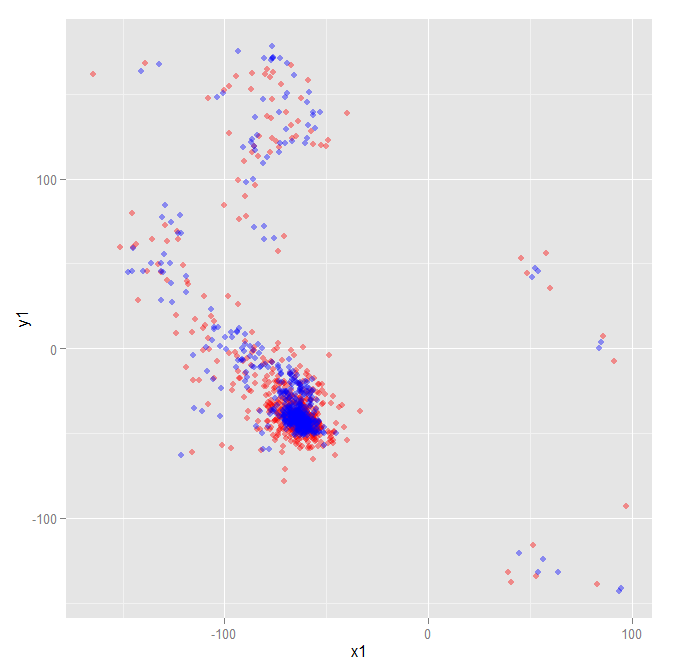
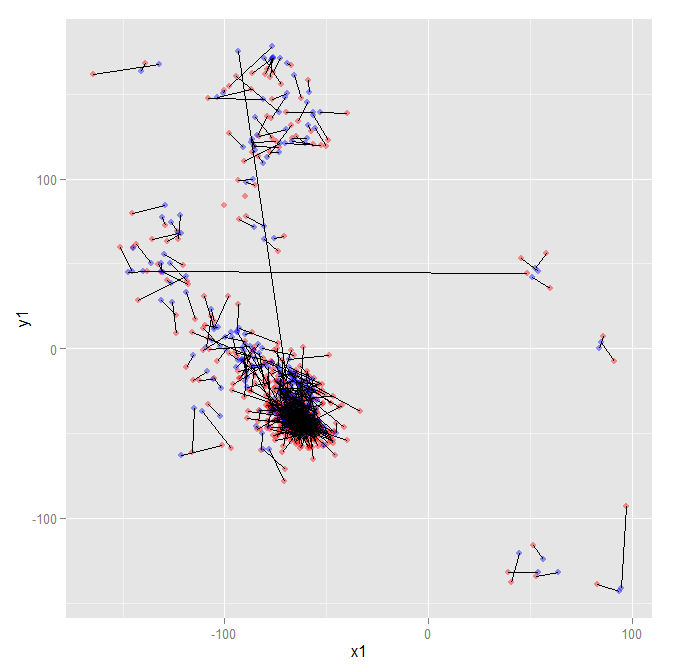
 

**Fig 5. Visual representation of the Smith-waterman algorithm used for the local alignment algorithm.** Source: <https://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm>

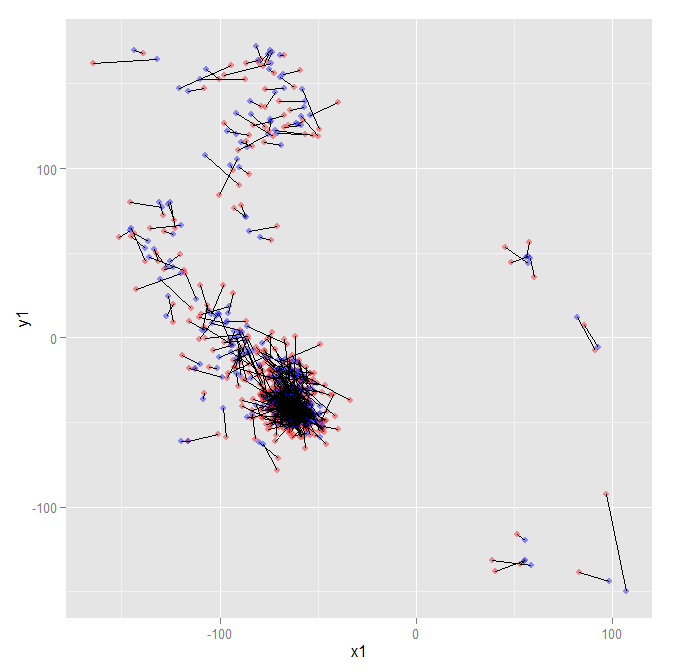
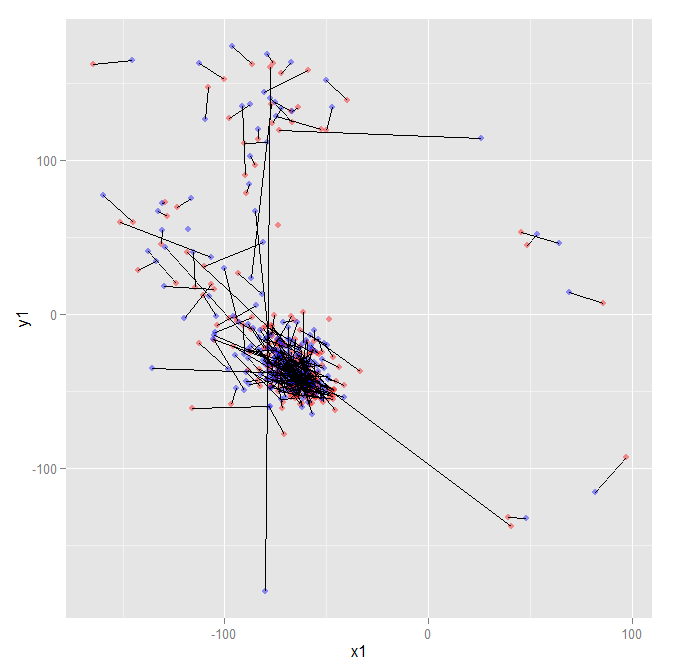
Later into the project it was decided that the 10 proteins that had less than a 50% identical score was removed from the data set due to the identical score being low caused by the global alignment algorithm registering longer sequences causing for possible variables that could not be accounted for with the level of analysis that was currently being done.

**Dihedral Angle Shifts**

In order to analyze the conformation changes caused by indels and mutations, Ramachandran plots were generated for both base hemoglobin protein (1GZX) and the target hemoglobin proteins. To compare the changes, the two generated Ramachandran plots were overlayed using R statistical computing software and corresponding amino acids were followed to see the magnitude and direction of the changes (Fig 6 and 7).

**Fig 6. R generated overlay of Ramachandran plots for PDB file 1GZX (red) and 1A3O (blue) (left) and the corresponding magnitude change of every amino acid from 1GZX (red) to 1A3O (blue) (right).**

sdfkjsdlfsd

**Fig 7. R generated overlay of Ramachandran plots for PDB file 1BAB, a 99% identical score (left) and for 1A4F, a 71% identical score (right) and their corresponding magnitude changes for each amino acid.**

From the resulting figures it can be seen that there are clear large changes in the dihedral angles for some of the amino acids resulting in large shifts in orientation and conformation (Fig. 6B and 7). These conformational changes are large enough to change their rotation completely from some amino acids exhibiting dihedral angles similar to that of alpha-helices to become beta-sheets. However, the overall protein itself still retains its function which suggests these large conformations aren’t detrimental to the functionality of the protein. Further investigation into these conformational shifts revealed no specific patterns as to which amino acids were more likely to shift or if these shifts were a result of any other amino acid properties. Furthermore, there were no immediately reportable trends suggesting that beta-sheets would always become alpha-helices or vice versa. Although, no pattern could be distinguished in the 20 proteins examined, this could be due to a small sample size used for this project.

Additionally, the same pattern was seen in both higher identical score protein comparisons as in the lower identical score protein comparisons (Fig. 7). While there were some proteins that didn’t exhibit large shifts (Fig. 7A), the average result suggest that this shift was not by pure chance and proteins were just as likely to have some amino acids exhibit these conformational shifts whether they were more identical or not to the original 1GZX hemoglobin.

Another pattern that was seen between the higher identical score proteins compared to the lower score proteins were the magnitude of their shifts (Fig. 7). The average magnitude for the higher identical score protein comparison was significantly lower at 11.908 units compared to the 19.204 seen in lower identical score proteins (p<0.01). An ANOVA and a post-hoc Tukey HSD determined no variation between sample populations for both the high identical score proteins and the low identical score proteins suggesting a trend for proteins with greater variation in their amino acid sequence to have a small but measurable conformation shift.

**Flexibility and Rigidity**

Continued exploration into the shifts resulted in the analysis of indels and SNPs between the proteins. In order to do this, the Python modelling software known as PyMOL was used for a better visualization of the perceived metric data. Both proteins were loaded into PyMOL and an alignment algorithm that existed already in the PyMOL software was used to align both molecules to one another. In the following images, the green protein is the base 1GZX hemoglobin. Other coloured ribbon or line representation of the protein are the other proteins.

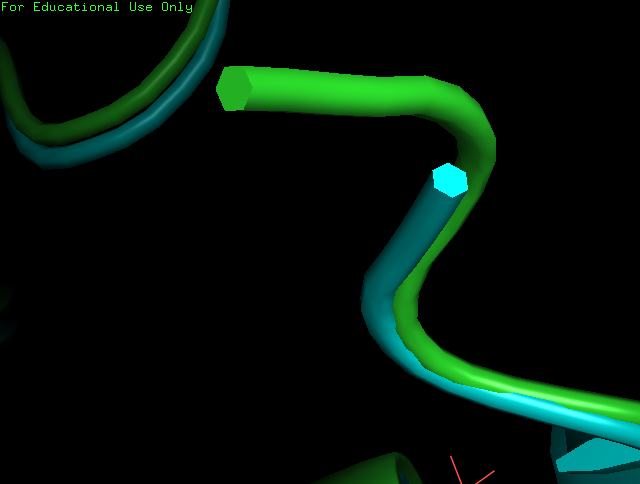
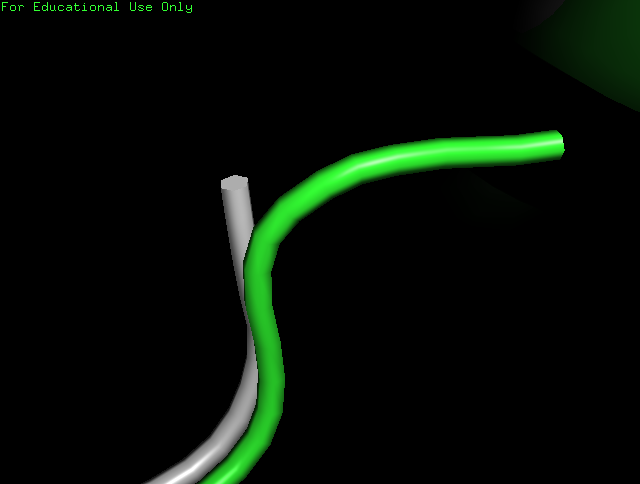
**Table 1. Summary of indels found in the 10 high identical score proteins aligned to 1GZX.**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HISP | 1A3O | 1BAB | 1BZI | 1BZZ | 1COH | 1O1O | 1YDZ | 4MQC | 1AJ9 | 1Y0D |
| # of indels | 2 Del  2 Ins | 1 Ins | 1 Ins | None | None | None | 1 Del | None | None | 2 Del  2 Ins |

**Table 2. Summary of indels found in the 10 low identical score proteins aligned to 1GZX.**

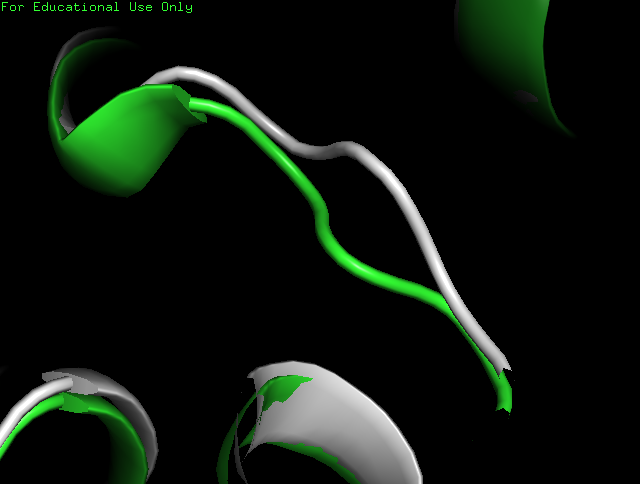
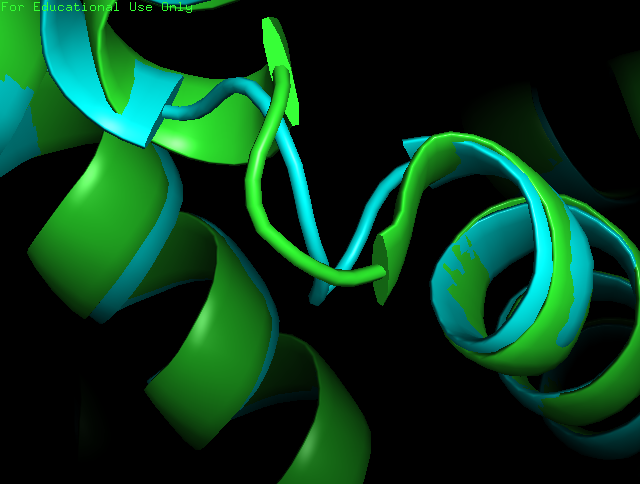
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LISP | 1A4F | 1FAW | 2QMB | 2R80 | 3DHR | 3EOK | 3FSA | 3K8B | 3MJP | 3WTG |
| # of indels | 13 D  13 Ins | 19 D  19 Ins | 28 D  28 Ins | 30 D  30 Ins | 32 D  30 Ins | 9 Del  9 Ins | 26 D  26 Ins | 27 D  27 Ins | 22 D  22 Ins | 38 D  32 Ins |

At first, I tried to determine where the mutations were occurring. It was during this time where the difference between the high identical scoring proteins and the low identical scoring proteins became evident. Initially when examining the metrics provided by the code, the first pattern that was evident was the balance of the insertion and deletions. In all proteins examined, there was always the same number of insertions as deletions (Table 1 and 2). While there were some exceptions, 3WTG and 3DHR, the majority of the proteins examined always had a balanced number of insertions and deletion (Table 1 and 2). This prompted me to examine this trend further. For the high identical scoring proteins, most of the deletions or insertions that occurred in the amino acid sequence occurred at the beginning or ends of segments (Fig. 8). This pattern was seen for all the indels for the 10 proteins surveyed using PyMOL

****  ****

**Fig 8. Examination into the indels for the high identical scoring proteins revealed every deletion (left) and insertion (right) were either at the beginning (left) or the end (right) of the segment sequence, with no indels occurring anywhere in the middle of the segment.**

The next interest was to compare the large shifts previously seen before. Where were they located? What does that conformational change look like? Even though it is such a large conformational shift, how does the molecule still remain the same? The answer became very evident once I investigated further. As shown in Figure 9, when both proteins were aligned with one another, their polypeptide chains are generally very close together. This is evident in Fig. 9B where the Helix appears to be both blue and green on the bottom right hand corner suggesting a very tight or similar fit between the two proteins. However, it can be seen that there are areas where there are large distortions in the pattern. These areas shown below are the conformational shifts seen in the Ramachandran plots. They aren’t as large or major as one would assume, but they have major implications. Firstly looking at the one on the left, it can be seen that the protein is starting to come apart and contort in a different direction. On the protein on the right, the contortion seems to be twisting about, moving in a completely different direction. However, even with these large conformational changes, the polypeptide chains seemingly still maintain structure and continue to come together at the end to fulfill their rigidity (Fig. 9). Furthermore, upon examining all 20 proteins, every single conformational change occurred outside of the alpha-helices and were only evident in the connecting chains between alpha-helices.

** **

**Fig. 9 Conformational shifts recorded by Ramachandran plots visualized using 3D software.** While the majority of the protein seems to be aligned quite well, there are areas that have major conformational differences between the base protein (green) and the aligned protein (grey & and blue).

These patterns agree with previous literature which states that proteins are very flexible but still maintain rigidity. With how evolution occurs, proteins need to maintain variation in order to potentially combat species specific diseases and allow for further evolution. However, even with their variability, the original function of the protein must remain stable. These conformations show this extremely well (Fig 9). By preserving the structural integrity of the alpha-helices which are the primary tertiary structures responsible for the function of the hemoglobin, the variability and potential for these large shifts falls on the connecting segments which are less important in terms of function and clearly have a large degree of flexibility. Further proof of this rigidity in the alpha-helices can be seen when examining areas of indels in the low identical scoring proteins.

LYS – ALA – ALA – TRP – GLY – \_\_\_\_ – \_\_\_\_ – \_\_\_\_ – LYS (1GZX - Indel)

LYS – \_\_\_\_ – \_\_\_\_ – \_\_\_\_ – GLY – VAL – PHE – SER – LYS (1A4F - Indel)

LYS – ALA – ALA – TRP – GLY – LYS (1GZX - Indel)

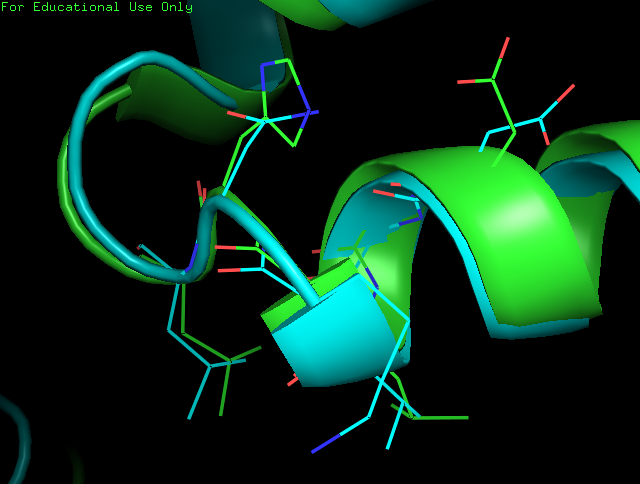
LYS – GLY – VAL – PHE – SER – LYS (1A4F - Indel)

**Fig 10. Small sequence of code pulled from the alignment algorithm showing the balance between insertions and deletions between the base, 1GZX protein and the corresponding aligned, 1A4F protein.** Even though there are deletions or insertions that exist in the code, these are generally “fixed” and replaced immediately with a corresponding insertion or deletion to match the lost or changed amino acids.

As mentioned before, the indels that were seen in the high identical scoring proteins weren’t very interesting. They only occurred on the beginning or ending segments and did not affect the overall structure in a meaningful way (Fig. 8). Furthermore, there were very limited mutations that occurred between the proteins (Table 1). However, the pattern seen for the indels in the low identical scoring proteins was far more interesting. A general pattern that started to present itself that agreed with the balanced number of insertion and deletions was the trend that every insertion or deletion that occurred in the protein would be followed with an immediate insertion or deletion in order to attempt to balance out the previous indel (Table 2 and Fig. 10).

The figure above (Fig. 10) is a small sequence of an amino acids showing a deletion in the 1A4F from the original 1GZX hemoglobin protein, but clearly after the glycine, there is an immediate insertion of three amino acids which when aligned, perfectly replace small amino acid with small amino acid and large amino acid with large amino acid.

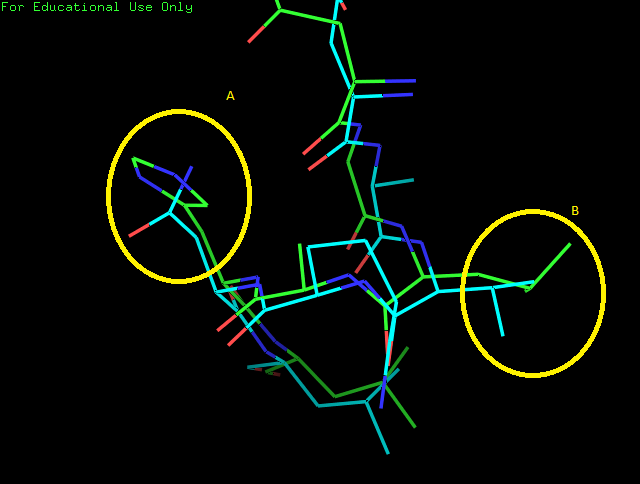
This further supports the previous statement of the rigidity in proteins as well as showing the flexibility of the genetic sequence. Even though the genetic sequence changes, the overall shape remains the same due to the replacement of the mutations with similar functioning or size amino acids (Fig. 10 and Fig. 12). To make this further intriguing, these changes were unlike the large conformational shifts seen before. While those conformational shifts were exclusive to the connecting segments, the mutations were highly variable, appearing in helices, in the connecting segments and generally did not have an apparent distinguishable pattern (Fig. 11).

****

**Fig 11. Mutations of insertions and deletions occurred everywhere, sometimes they occurred in the flexible connecting segments, and other times they occurred in the alpha-helices**

In order to visualize the pattern seen in Figure 10, the figure below was generated to attempt to show the rigidity of the protein. The image shown below is the same as figure 11 except it has the general ribbon structure removed in order to show the polypeptide sequence identified in figure 10. There are two highlighted areas, A and B. In both these areas, the amino acids that are aligned are different. A is an alignment of phenolphthalein and tryptophan while the amino acids identified in B are valine and isoleucine. However, it is apparent that the overall structure of the polypeptide maintains its shape. Each of the mutations that has occurred has only replaced the original amino acid with another amino acid of similar size, and functional properties. This pattern was seen for all 10 proteins that were examined. Every time there was a deletion or an insertion, a corresponding insertion and deletion would always occur somewhere immediately downstream and the replacing amino acids would always be similar.

As the tertiary structure of a protein is usually an interaction between its larger amino acids and their larger side chain, it would make sense that to keep the original functionality you would require something that could maintain the original form, therefore, by that inference, this pattern does not seem too farfetched, but actually proves what has been known about the potential for evolution where sequences can be fairly different from one another, but their overall functionality always seems to stay the same.

****

**Fig 12. A visual representation of the polypeptide sequence identified in Figure 10 with two distinguishing groups highlighted as A and B.**

**Code**

As mentioned before, a large majority of the analysis done in this report was done using code created and written by myself. All code is primarily written in python and uses objects and libraries created by myself. There are some functionalities such as analyzing helices and sheets that are currently not updated, but the majority of the code remains functional.

To generate the phi and psi angles for Ramachandran plots, simply use the command: *python 4900.py valid\_pdb\_file.pdb* and the code should create a *name\_of\_valid\_pdb\_file­.txt* that outputs something similar to:

**Residue Sequence Number | Amino Acid | Phi | Psi | Distance from Center**

2 LEU -86.9199524082 121.663292073 21.3849995069

3 SER -76.2210055178 171.134822774 24.4887606323

4 PRO -59.6996415098 -40.6502638334 27.8152472225

5 ALA -67.4581643516 -37.6316116506 26.7488192941

6 ASP -62.5102906578 -42.4955902714 23.0363722521

…

It is possible to calculate the dihedral angles for multiple files at once. In order to do this, simply use the command: *python 4900.py all folder\_name\_with\_pdb\_files­* and the program will calculate the dihedral angles for every single PDB file located within the folder and create the same *output.txt* as the one seen above. Note that the program will not create individual dihedral angles for every single PDB file, but will combine every single file into one large text file. It is possible to set a set amount of files for the program to calculate.

On line 31, there is a variable called: “only\_100” which is currently set at 100, by changing the numerical value of this variable, the program will calculate the number of files indicated by this numerical value whether it be 1000 or 10.

In addition to generating the dihedral angles for the creation of a Ramachandran plot, the program can also align two sequences either given the original PDB file or the produced *output.txt* that was created containing the dihedral angles. To align the two sequences using the Smith-Waterman algorithm, simply use the command: *python 4900.py align name\_of\_1st\_file.pdb.txt name\_of\_2nd\_file.pdb.txt*. This will produce an output that is exactly the same as the output shown above, however there will be dashes wherever the algorithm determines there is an insertion or deletion event that has occurred. Similar to:

**Residue Sequence Number | Amino Acid | Phi | Psi | Distance from Center**

8 THR -76.76965996 -43.2136435723 21.7505985422

9 ASN -60.5833946488 -48.1678156214 18.0376890921

10 VAL -56.819949344 -46.8358462565 17.0042297568

11 LYS -57.0985494695 -47.6622868463 21.5732900989

-

-

-

12 GLY -52.322497643 -54.8021946285 19.6792601157

13 VAL -76.0858751628 -24.2011010745 16.4231124152

There are other functionalities of the program as well including the ability to calculate the x, y, and z coordinates of the “center” of the protein based on molecular mass and the distance that each amino acid within the protein is from that calculated center. The command for both of these functions are: *python 4900.py PDB\_file.pdb center* and *python 4900.py PDB\_file.pdb distance* respectively. The output for the center functionality will be a single sentence stating the x, y, and z coordinates of the proteins center. The output for the distance functionality will be a text file called, *distance.txt* that will contain the amino acid name and their relative distance away from the center of the protein.

**File Navigation**

1GZX BLAST Contains the 30 proteins identified by BLAST that was used

Library Contains the object files and class files used for 4900.py

PDB Files Six proteins used to test and create the dihedral angle script

4900.py The main program

Magnitudes.xlsx The ANOVA results as well as post-hoc Tukey HSD on magnitude

Ramachandran Plots.xlsx The result from the six proteins and rough work involving 4900.py

**Sources of Information**

Protein Structure Information

<https://en.wikipedia.org/wiki/Protein_structure_prediction>

<http://www.rcsb.org/pdb/static.do?p=general_information/about_pdb/nature_of_3d_structural_data.html>

Ramachandran Plot Information

<https://en.wikipedia.org/wiki/Ramachandran_plot>

<http://www.cryst.bbk.ac.uk/PPS95/course/3_geometry/rama.html>

<http://www.proteinstructures.com/Structure/Structure/Ramachandran-plot.html>

<http://stackoverflow.com/questions/1011938/python-previous-and-next-values-inside-a-loop>

Hemoglobin Information + 1GZX Model Page <https://en.wikipedia.org/wiki/Hemoglobin#Structure>

<http://www.rcsb.org/pdb/explore.do?structureId=1gzx>

BLAST

<http://blast.ncbi.nlm.nih.gov/Blast.cgi>

Smith-Waterman Algorithm Information

<https://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm>

<http://puriney.github.io/2013/08/22/smith-waterman-algorithm/>

PDB File Information

<http://deposit.rcsb.org/adit/docs/pdb_atom_format.html>

Other

<https://www.r-project.org/>

<https://www.pymol.org/>