**An Outline of my Programming Approach**

User can retrieve information about genes using my programme such as gene names, its sequence, isoelectric point, its interaction partners with quality score not more than 900 and PDB IDs of proteins with more than 10 structures available. It also provides the user with a column chart o amino acid composition and frequency, a network graph of its interaction partners and a bar chart of proteins with more than 10 structures available. A further explanation can be found below in this document.

**Initiating Perl** : I have used **“#!/usr/bin/perl”** to tell the command line interpreter that I am going to work with Perl. And LWP simple library to connect with the huge amount of data that is available online libraries such as uniport, string database, ensemble, etc. I had to let the user input the gene name as he can retrieve whatever the information he needs. Hence why I used “<STDIN>”.

**Step 1:** I have used two different URLs in order to retrieve to gene name and sequence in this step and the user has to make sure he has relevant version of Perl and the libraries within the working directory. And got rid of any white spaces as they are interrupting my programmes further down the line. And I printed both gene name and the protein sequence.

**Step 2:** Calculating the isoelectric point is very important as this is the point where an amino acid does not carry any negative charge, which helps it not to migrate in electric field. I have used “Bio::Seq” to turn the sequence into an object and “Bio::Tools::pICalculator” to calculate the isoelectric point of any given protein/gene. This programme returns the isoelectric point of the given input.

**Step 3:** In this step, I have turned already retrieved sequence into an array by splitting it and added a counter for every single amino acid and put it through “foreach” loop to count the amino acid composition and frequency and printed out the results. I have also used the Java script to create a HTML script/link to display the results in a user-friendly column chart. User can find the html link in a separate file within this zipped folder which can be opened from any browser.

Bar Chart displays what amino acids are there in the sequence on x-axis and how many times they’re appeared on y-axis respectively.

**Step 4:** This step retrieves up to 20 interacting proteins of BRCA1 and limited by quality score only above 900. This gives us the accurate results as we are going for highest quality score.

To get this information I have used String database to retrieve the required information by specifically requesting for whatever information needed in the URL and got rid of unnecessary information with substrings. Later, printed it out after splitting it at required locations of the array. This gave me the information of 20 interacting proteins with quality score over 900.

**Step 5:** This step shows the interacting partners in a network graph which is included in combined graph. To create this network graph, I have used string database to get the interaction partners and made a protein list by sing “grep” and specifically told my programme to highlight the input protein in RED. I added the java script to create HTML script for Network graph.

Network graph shows the given protein in red colour as it’s easier to spot. It shows it’s other 20 interaction partners and how they are connected. It highlights the connections of every single protein when user clicks on any protein in the network. As we can see in the graph, it tells us that BRCA1 works with many other proteins to repair the DNA efficiently and quickly. This further clarifies the importance of interactions between proteins within the cell.

HTML file is included within the combined file and can be opened using any browser given the user has the required files in the running directory.

**Step 6:** This step gives the user protein structure identifiers from uniport using specific uniport URL.

These IDs can be used to retrieve protein structures. To get them, I used string IDs from string database and turned them into an array before I got rid of substrings and any white spaces left. Later I have used these string IDS to get the PDB IDs from uniport and printed them out.

Protein databases on line have specific IDs to find the structures of that proteins. These IDs are assigned permanently. Protein structures are required to map the protein and to find out what is the protein made of, how they are aligned and how it interacts with other proteins in the network.

**Step 7:** After getting PDB IDs from uniport, I have calculated the structure counts available for each protein and separated them by proteins with only 10 structure counts available as this gives the user a more accurate result.

I have created a bar chart consisting of proteins with more than 10 structure counts available. User can see the proteins on y-axis and the structure counts on x-axis. User can find out how many structures counts available for each protein. This gives the user accurate information about structure counts.

**Conclusion:** It is important for the user to get all the required information. It gives the user much needed information such as gene name, sequence, amino acid composition and frequency, 20 interacting proteins of input protein, PDB structure identifiers and structure counts. It also prints html script for a column chart of composition and frequency, a network graph of given proteins interaction partners, a bar chart of structure identifiers and structure counts. This html link helps the user further understand the output in an easier and clear way.

**References:**

Book 1: Beginning Perl for Bioinformatics by **James Tisdall**

Book 2: Mastering Perl for Bioinformatics by **James Tisdall**

Module material from Cranfield university library services by Dr Fady Mohareb.

* <https://info.gbiosciences.com/blog/bid/149959/what-is-the-role-of-the-isoelectric-point-of-a-protein-in-its-purification>
* <http://proteopedia.org/wiki/index.php/PDB_identification_code>
* <https://www.uniprot.org/uniprot/Q3B891>
* <https://www.uniprot.org/uniprot/?query=BRCA1&sort=score>
* <https://www.uniprot.org/uniprot/?query=organism:9606+AND+accession:Q3B891&format=tab&columns=genes(PREFERRED)>
* <https://www.uniprot.org/uniprot/?query=organism:9606+AND+accession:Q3B891&format=tab&columns=sequence>
* <http://string-db.org/api/tsv-no-header/interaction_partners?identifier=BRCA1&species=9606&limit=20&required_score=90>
* <http://string-db.org/api/psi-mi-tab/interactions?identifier=BRCA1&species=9606&required_score=900&limit=20>
* <http://string-db.org/api/psi-mi-tab/interaction_partners?identifier=BRCA1&species=9606&limit=20&required_score=900>