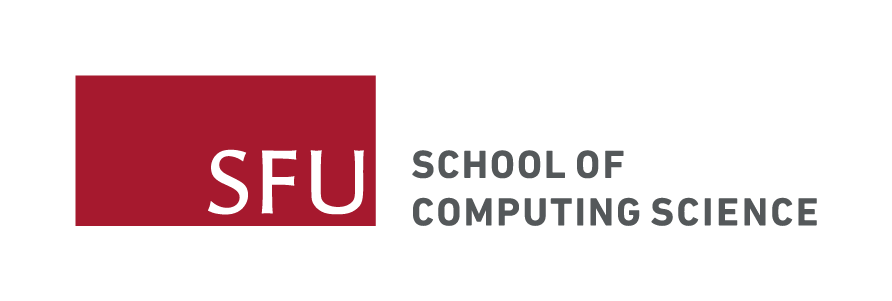
Bioinformatics

(Computational analysis of Post translational modifications in Human Proteins)

Document Number: 1

Revision: 1.0, Date: June 6, 2020



Sun’s Lab, Simon Fraser University

British Columbia, Canada

This document is the sole property of Sun’s Lab and may not be reproduced or distributed without written consent of Dr. Bingyun Sun

**Abstract:**

Abstract Proteins are large biomolecules, or macromolecules, consisting of one or several long chains of amino acids, i.e. polypeptides. Proteins have complex shapes that include various alpha helixes, beta sheets, loops, and disordered regions. Functions of Human proteins are determined by their structures and many of these structures are still unknown and difficult to derive experimentally. To solve this problem, computer-aided data modelling has been widely applied. Our project focuses on identifications of patterns between two specific post-translational modifications ─ sulfide pairs (i.e. disulfide bond) and N-glycosylation, in human proteins and examine how the positions of these post-translational modifications affect the structures thereof functions of proteins. Human protein data is download from UniProt, and python programming is used to compute the relative positions and relationships between disulfide bonds and N-glycosylation. Proteins with unique patterns of these modifications are identified, and their functions are further analyzed. Our study provides useful insights for structures and functions of proteins and particularly to proteins with unknown structures

**TABLE OF CONTENTS**

**1.0 System Requirements**

**2.0 Background Information**

**3.0 Executing Code**

**4.0 Functions and their working**

**5.0 References**

1. System Requirement:

To run the code, the following are the minimum requirements which must be satisfied.

**Python:**

The Python version supports for this code is 3.0+. The code does not support Python 2 version.

**System:**

Supports Windows and Linux as long as Python 3+ is compatible with it.

**Python Libraries:**

In most of the cases the libraries are installed during the Python Installation but if somehow, they did not get installed then to run the code it requires to install Pandas, NumPy, Re (Regular Expression operators), Matplotlib, Sys (System-specific parameters and functions) libraries.

1. Background Information:

The main strategy is to work with the sulfide bond positions and glycosylation positions and to figure out how the glycosylation position are inside sulfide bonds, outside sulfide bonds, N-terminus distance, C-terminus distance and so on. So, the first stage is to manage data and the second stage is to do data analysis.

**Downloading Data from UniProt Website**

Down below are the following steps to download data in correct form from Uniport website.

* Go to <https://www.uniprot.org/>
* In the search bar type Humans, then click on reviewed articles.
* Customize results table and select Entry, Entry name, Protein names, Genes names, Length, Organism, Mass, Status, Disulfide bond, Glycosylation.
* Make sure to select all these columns if not considered then program will give error. If interested in more fields then add them along with the above columns.
* Download the file in either excel format or tab-separated format.

1. Executing Code:

The code requires some steps to follow to get the results.

**Running the File: -**

* Open Command Prompt in Windows or Terminal in Linux.
* Setup the directory in the terminal where the code and UniProt files are present.

**Running the code for Excel File:**

* Run this command in terminal:
  + python3 Bioinformatics\_excel.py input.xlsx \*

**Running the code for Tab-Separated File:**

* Run this command in terminal

$ python3 Bioinformatics\_tab.py input.tab \*

**\*** Input.tab and Input.xlsx are the downloaded tab and excel files respectively from uniport website.

1. Functions and their working:

**Part (1): Data Cleaning and Exploration of Sulfide bonds and Glycosylation**

In the Bioinformatics file includes data cleaning and ETL (Extract-Transform-Load) is done. The code has comments which help user to understand code and is divided into parts to make it efficient and clear.

**Each Part Explanation and Functionality**

(1.1) Reading File and Data frame

The code functionality is to read file from terminal using panda’s library. A new data frame has been created to store data values and all statistical properties will be saved in that data frame.

(1.2) Storing data

The data copied from input file to our new data frame. The organism data is filtered with a condition that contains only ‘Homo Sapiens’ as our interest is in human proteins. The data which contain empty values are dropped from data.

(1.3) Functions for Integer and Float Values

* The function get\_sulfide\_value() used to get the integer pair values from data. The data is in the form of ‘X..X’ where X is a integer.
* The function get\_int\_value() used to get integer value from data.
* The function get\_float\_value() used to get the float values form data.
* The function get\_Nlinked() helps separate the integer values form the Glycosylation data column.

(1.4) Glycosylation and Sulfide bond data cleaning

Data cleaning is done to get the integer position pair value for the disulfide bond and integer position of glycosylation using the functions described in part (1.3). The data is cleaned on the pattern found in the glycosylation and disulfide positions. The disulfide data pattern is ‘X..X’ where X is the position. The glycosylation data pattern is ‘X; note=\"N-linked’ where X is the position.

(1.5) Finding Positions of Glycosylation in Disulfide Bonds

This part is complex as it contains the code to find the glycosylation positions are inside or outside the sulfide bonds. There are new temporary data frame are used such as glycol\_data and sulfide\_data which helps in finding length and other properties. The code uses a simple algorithm with conditions specified to check the position of glycosylation, if the glycosylation is found inside sulfide pair then it will have ‘i’ attached with it otherwise it has ‘o’ attached with it. The code outputs the relative distance of glycosylation position with respect to sulfide positions and will show that they are inside or outside.

(1.6) Defining Score for Disulfide pairs

The score is defined based on overlapping of soldier pairs.

* One pair is totally inside other pair then score will be 1.
* One pair is partially inside other pair then score will be 0.5.
* One pair is totally outside other pair then score will be 0.

The code uses the above scoring method to calculate the score for each sulfide pair. After calculating the score, then uses the above score the code calculates the sum of score for each Protein.

(1.7) Interbond Distance

The code manipulates the interbond distance between pairs which is the distance from one pair end point to other pair starting point.

(1.8) Intrabond Distance

The code manipulates the intrabond distance which is the length of each sulfide pair.

(1.9) Count Sulfide Bonds

Counting the number of sulfide bonds pair for each protein.

(2.0) Count Glycosylation Positions

Counting the number of glycosylation positions for each protein.

(2.1) First Disulfide pair to N-terminus and Last Disulfide pair to C-terminus

The first part finds the find the first position where the sulfide pair occur. The second part finds the last pair position of sulfide bond.

(2.2) Average Intrabond Distance

The average intrabond distance is the sum of intrabond distance for each protein divided by number of intrabonds.

(2.3) Average Intrabond Distance with Glycosylation Inside Sulfide Pairs

The average intrabond distance calculated for those sulfide pairs which has glycosylation inside them.

(2.4) Sulfide Pair with No Glycosylation Inside

The code uses a unique technique to find sulfide pair with no glycosylation position inside. A new data frame called glycol\_sulfide\_frame which has glycosylation and disulfide bond series. The code use ‘nil’ to separate those pairs which has positions inside from those which have no position inside.

The next part calculates the distance between pairs for those which does not have any glycosylation position inside sulfide pairs.

(2.5) Average Length for Sulfide Pair with No Glycosylation Inside

The average distance calculated of each protein sulfide pair for part (2.4).

(2.6) Output File

The output file format is excel and can be changed according to user need.

1. References:
2. Uniprot, <https://www.uniprot.org/>
3. David Bioinformatics Resources, https://david.ncifcrf.gov/