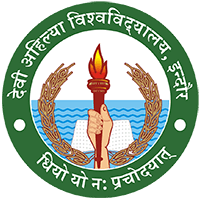
**“Writing a Python script for calculating conservation scores**

**for a multiple sequence alignment”**

**DEVI AHILYA VISHWAVIDYALAYA, INDORE**

**SCHOOL OF BIOTECHNOLOGY**



**In Partial Fulfillment of the Requirement for the Award**

**Of Degree Of**

**Master of Science in Bioinformatics**

**2019-2021**

**Guided by: Submitted by:**

**Dr. Venuka Goyal Sonali Patidar**

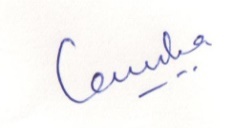
**( Visiting Faculty, School of Biotechnology DAVV) M.sc bioinformatics**

**Semester IV**

**Certificate**

**Date :- 30/06/2021**

This is certify that this dissertation entitled **“Writing a Python script for calculating conservation scores for a multiple sequence alignment”is** a bonafide record of the project work carried out by Ms. Sonali Patidar under my supervision **Dr. Venuka Goyal** for the partial fulfillment of the requirement for the degree of Master of Science in Bioinformatics from **SCHOOL OF BIOTECHNOLOGY Devi Ahilya Vishwavidyalaya Indore** and the conducted work is not presented by us for any Degree/Diploma before. However, it is noteworthy that the data presented are only preliminary results and need further repetition of the research work; as our research work was severely affected by the COVID-19 pandemic and associated lockdown of many months in the whole nation.



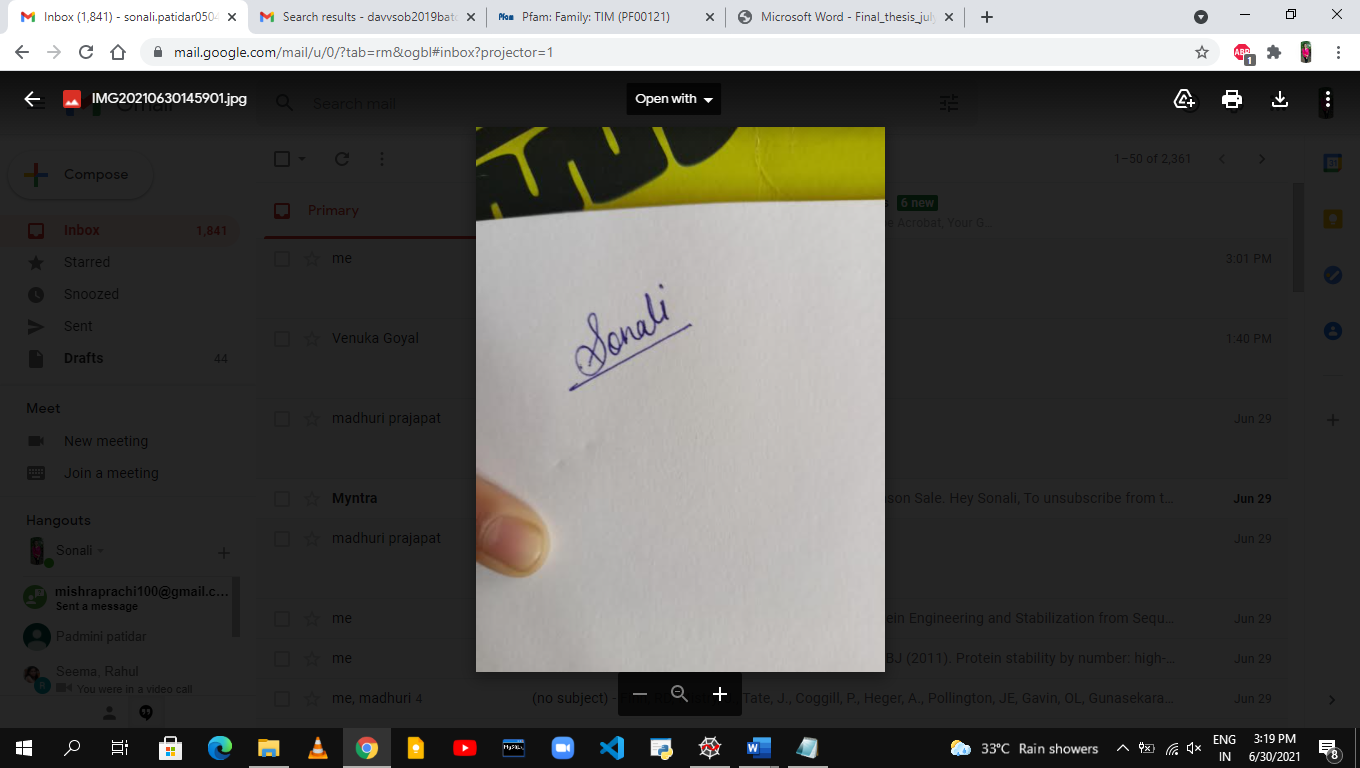
**(Dr. Venuka Goyal)**

**(Visiting Faculty, School of Biotechnology, DAVV)**

**DECLARATION**

I hereby declare that the Dissertation entitled “Writing a Python script for calculating conservation scores for a multiple sequence alignment”, embodies my original work carried out under the supervision of Dr Venuka Goyal, Visiting Faculty Of SCHOOL OF BIOTECHNOLOGY, DAVV Indore, for partial fulfillment of Master of Science in Bioinformatics during the period, Jan 2021 to June 2021. This work presented in this report is original and has not used for the award of any Degree/ Diploma before.

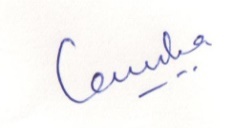
Signature



Sonali Patidar

Date 30/06/2021

Signature



Dr Venuka Goyal

(Visiting Faculty, School of Biotechnology, DAVV)

Date 30/6/2021

**Acknowledgement**

I would like to extend my sincere and heartfelt gratitude to my supervisor Dr. Venuka Goyal for her precious suggestion, immense encouragement, and continuous guidance. her guidance helped me a lot during the master's thesis work and writing of this thesis. I am grateful to you ma'am for all your favor upon me.

Besides my supervisor, I would like to thank Dr. Anjana Jajoo, head of the School of Biotechnology Devi Ahilya Vishwavidyalaya, Indore and entire DAVV community for encouragement and providing computational facilities and wi-fi connectivity organized for the benefit of the students.

I would like to express my sincere thanks to Dr. Anil Kumar, Dr. Hamendra Singh Parmar, Mr. Priyesh Hardia School of Biotechnology, Devi Ahilya University for their continuous guidance and teaching during my master's program.

I would like to thank my friends for continuous support and motivation at every stage of my project.

Last but not least I would love to thank my parents and brother who helped me a lot in finishing this project within the limited time.

**Contents**

**.**

|  |  |
| --- | --- |
| Summery | 1 |
| Introduction | 2 |
| Objectives | 6 |
| Review of Literature | 7 |
| Database and Tools | 9 |
| Materials and Method | 13 |
| Result | 16 |
| Conclusion | 33 |
| Reference | 34 |

**Summary**

in this study We have provided a step-by-step protocol for calculating relative entropy and describe how they can be used to predict mutations that have a high probability of stabilizing a protein. We used Biopython codefor calculating conservation scores for a multiple sequence alignment. We used Biopython code that can be applied to calculate each of the count Amino acid, sum function and frequency matrix. We estimated the conservation of positions in the MSAs used the relative entropy. We used MSA for predicted stability of protein. The standards of consensus and correlation in a couple of sequence alignments (MSAs) have been used in the previous to understand and engineer proteins. However, there are multiple ways of obtaining MSA databases. We used Pfam for the seed alignment. MSAs can be acquired or built and curated based on the availability of information. relative entropy is an records theoretic metric that gives the distance of a distribution from a reference distribution. It is logarithmically associated to the multinomial probability of looking a specific distribution if you assume the reference distribution. If a position is relatively conserved, its amino acid distribution would be very unique from that predicted at random, making its RE value very excessive relative to a impartial reference state.

**1**

**Introduction**

Biopython is a set of libraries to provide the capability to deal with “things” of interest to biologists working on the computer. Biopython is that it frequently presents more than one methods of “doing the same thing.”

The Biopython Project is an international association of developers of freely accessible Python ([https://www.python.org](https://www.python.org/)) tools for computational molecular biology.

Biopython is the largest and most famous bioinformatics package deal for Python. It consists of a quantity of different sub-modules for common bioinformatics tasks. It is developed by Chapman and Chang,. It runs on Windows, Linux, Mac OS X, etc. It is primarly written in python but some C code is there to resolve complicated optimization. Biopython is capable of a lot like it can do protein structure, sequence motifs, sequence alignment also machine learning.

Python ([www.python.org](http://www.python.org/)) and Biopython are freely accessible open source tools, available for all the most important operating systems. Python is a very high-level programming language, in large commercial and academic use.

It features an convenient to analyze syntax, object-oriented programming skills and a vast array of libraries. Python can interface to optimized code written in C, C++or even FORTRAN, and collectively with the Numerical Python task Numpy makes a proper preferencfor scientific programming.

Biopython is a collection of python modules that provide features to deal with DNA, RNA & protein sequence operations. Biopython requires very less code. Biopython Provides microarray data type used in clustering and Reads and writes Tree-View type files.

The aim of Biopython is to provide simple, popular and extensive access to bioinformatics via python language.

The Biopython web site ([http://www.biopython.org](http://www.biopython.org/)) offers an online useful resource for modules, scripts, and internet hyperlinks for developers of Python-based software program for

**2**

bioinformatics use and research. Basically, the purpose of Biopython is to make it as convenient as possible to use Python for bioinformatics via developing high-quality, reusable modules and classes.

Biopython has grown into a large collection of modules, described quickly below, meant  for computational biology or bioinformatics programmers to use in scripts or include into their own software.

Biopython is a large open-source application programming interface (API) used in both bioinformatics software program improvement and in day to day scripts for common bioinformatics tasks. The homepage ([www.biopython.org](http://www.biopython.org)) offers access to the source code, documentation and mailing lists.

Features of Biopython:-

* Biopython is portable, clear and has easy to study syntax, Interpreted , interactive and object oriented, Supports FASTA, PDB, GenBank, Blast, SCOP, PubMed/Medline, ExPASy-related formats.

The Pfam database is a broadly useful resource for classifying protein sequences into families and domains. It is extensively used to analyse novel genomes, metagenomes and to information experimental work on specific proteins and systems.

The Pfam database is a large collection of protein families, every represented by means of multiple sequence alignments and hidden Markov models (HMMs). Pfam **a**dditionally generates higher-level groupings of associated entries, known as clans. A clan is a collection of Pfam entries which are related by means of similarity of sequence, shape or profile-HMM.

Each Pfam family has a seed alignment that carries a representative set of sequences for the entry. A profile hidden Markov model (HMM) is automatically built from the seed alignment and searched towards a sequence database called pfamse*q* the usage of the HMMER software.

The common reason of the Pfam database is to provide a complete and correct classification of protein families and domains. The Pfam internet site approves users to submit protein or DNA

**3**

sequences to search for matches to families in the database. For each protein family in PFAM

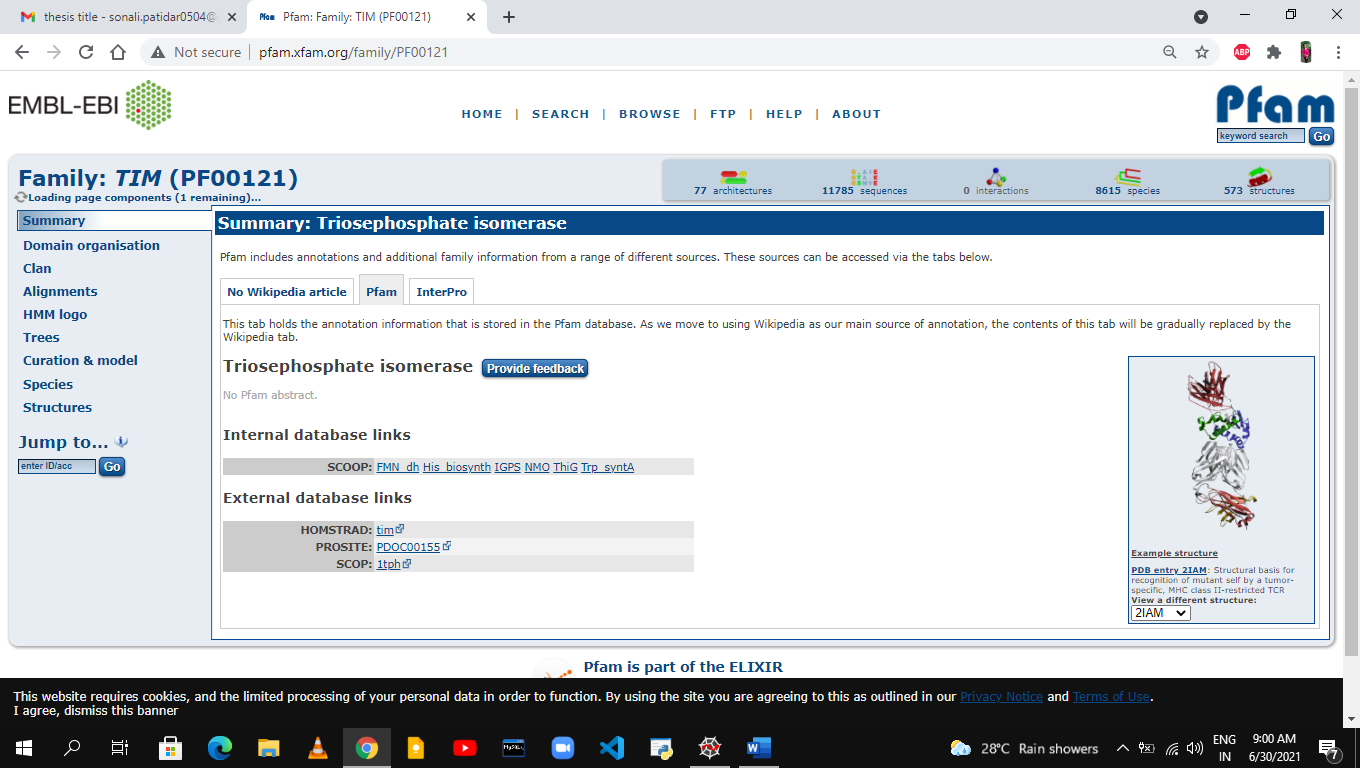
you can seem at multiple alignments, view protein domain architectures, and examine species distribution etc.

Pfam families are divided into two categories, Pfam-A and Pfam-B.

Pfam-A is the manually curated component of the database.

Pfam-B entries are automatically generated from the ProDom database and are represented by using a single alignment.

Pfam is a database of curated protein families, every of which is described through two alignments and a profile hidden Markov model (HMM). In Pfam, the profile HMM is searched against a large sequence collection, based totally on UniProt Knowledgebase (UniProtKB) , to locate all situation of the family.



The relative entropy, also known as the Kullback-Leibler divergence. It is a measure of how

**4**

one [probability distribution](https://en.wikipedia.org/wiki/Probability_distribution) is special from a second, reference probability distribution. In the easy case, a relative entropy of zero suggests that the two distributions in question have equal portions of information. Relative entropy is an information theoretic metric that gives the distance of a distribution from a reference distribution. It is logarithmically associated to the multinomial probability of looking a specific distribution if you assume the reference distribution. Relative entropy is calculated by the following formula.

RE (D) is calculated as:

**D= Σ p(xi)\*(log p(xi)/q(xi))**

p(xi) is the probability of observing residues i at position x in the MSA and q(xi) is the probability of observing residues i in a reference distribution.

**5**

**Objective**

* Write a Python code for calculating conservation scores for a MSA.

**6**

**Review of literature**

Some of research paper which has already published in the area of Protein Engineering and Stabilization from MSA. Consensus design, the decision of mutations based totally on the most common amino acid in every function of a multiple sequence alignment, has validated to be an efficient way to engineer stabilized mutants and even to format entire proteins. However, its software has been limited to small motifs or small families of especially associated proteins. Also, we have little concept of how information that specifies a protein's properties  is  distributed between positional results (consensus) and interactions between positions (correlated occurrences of amino acids). Here, we designed a number of consensus variants of triosephosphate isomerase (TIM), a large, various family of complex enzymes. The first variant used to be only weakly active, had molten globular characteristics, and was once monomeric at 25 °C in spite of being primarily based on nearly all dimeric enzymes. A intently related variant from curation of the sequence database resulted in a native-like dimeric TIM with near-diffusioncontrolled kinetics.

Understanding the determinants of protein steadiness stays one of protein  
science's biggest challenges. There are still no computational options that calculate the balance consequences of even factor mutations with sufficient reliability for realistic use. Amino acid substitutions not often extend the stability of native proteins; hence, large libraries and high-throughput screens or options are needed to stabilize proteins the use of directed  
evolution. Consensus mutations have verified effective for increasing stability, however these mutations are profitable only about 1/2 the time. We set out to recognize why some consensus mutations fail to stabilize, and what standards would possibly be beneficial to predict stabilization greater accurately. Overall, consensus mutations at extra conserved positions had been extra likely to be stabilizing in our model, triosephosphate isomerase (TIM) from  
Saccharomyces cerevisiae.

Pfam is a extensively used database of protein families and domains. This article describes a set of important updates that we have applied in the latest release (version 24.0). The most essential exchange is that we now use HMMER3, the latest version of the popular profile hidden Markov model

**7**

package. This software is approximately 100 times faster than HMMER2 and is more movements due to the routine use of the forward algorithm. The cross to HMMER3 has necessitated numerous changes to Pfam that are described in detail. Pfam release 24.0 consist 11,912 families.

Most proteins are only barely stable, which impedes research, complicates therapeutic applications, and makes proteins susceptible to pathologically destabilizing mutations. Our ability to predict the thermodynamic consequences of even single point mutations is still surprisingly limited,and set up techniques of measuring steadiness are slow. Recent advances are bringing protein steadiness research into the high-throughput realm. Some strategies are primary based on inferential read-outs such as activity, proteolytic resistance or split-protein fragment reassembly. Other techniques use miniaturization of direct measurements, such as intrinsic fluorescence, H/D exchange, cysteine reactivity, aggregation and hydrophobic dye binding (DSF). Protein engineering based totally on statistical analysis (consensus and correlated occurrences of amino acids) is promising, but much work remains to understand and implement these methods.

Information principle was once used to discover nonconserved coevolving positions in multiple sequence alignments from a range of protein families. Coevolving positions in these alignments fall into two common categories. One set is composed of positions that coevolve with only one or two different positions. These positions regularly show direct amino acid side-chain interactions with their coevolving partner. The different set contain positions that coevolve with many others and are regularly placed in areas intergal for protein function, such as active sites and surfaces worried in intermolecular interactions and recognition. We locate that coevolving positions are extra in all likelihood to exchange protein feature when mutated than are positions displaying little coevolution. These consequences mean that facts idea may additionally be utilized commonly to discover coevolving, nonconserved positions that are phase of purposeful websites in uncharacterized protein families. We recommend that these coevolving positions compose an important subset of the positions in an alignment, and may be as important to the shape and feature of the protein family as are highly conserved positions.

**8**

**Databases & Tools used**

In whole process we have used Databases and Bioinformatics tools.

1. **Anaconda**
2. **Spyder**
3. **Pfam**
4. **Anaconda :-** Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands.

Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

Anaconda is a Python distribution for large-scale data processing, predictive analytics, and scientific computing. It contains 1000+ of the most popular Python packages for science, math, engineering, data analysis.

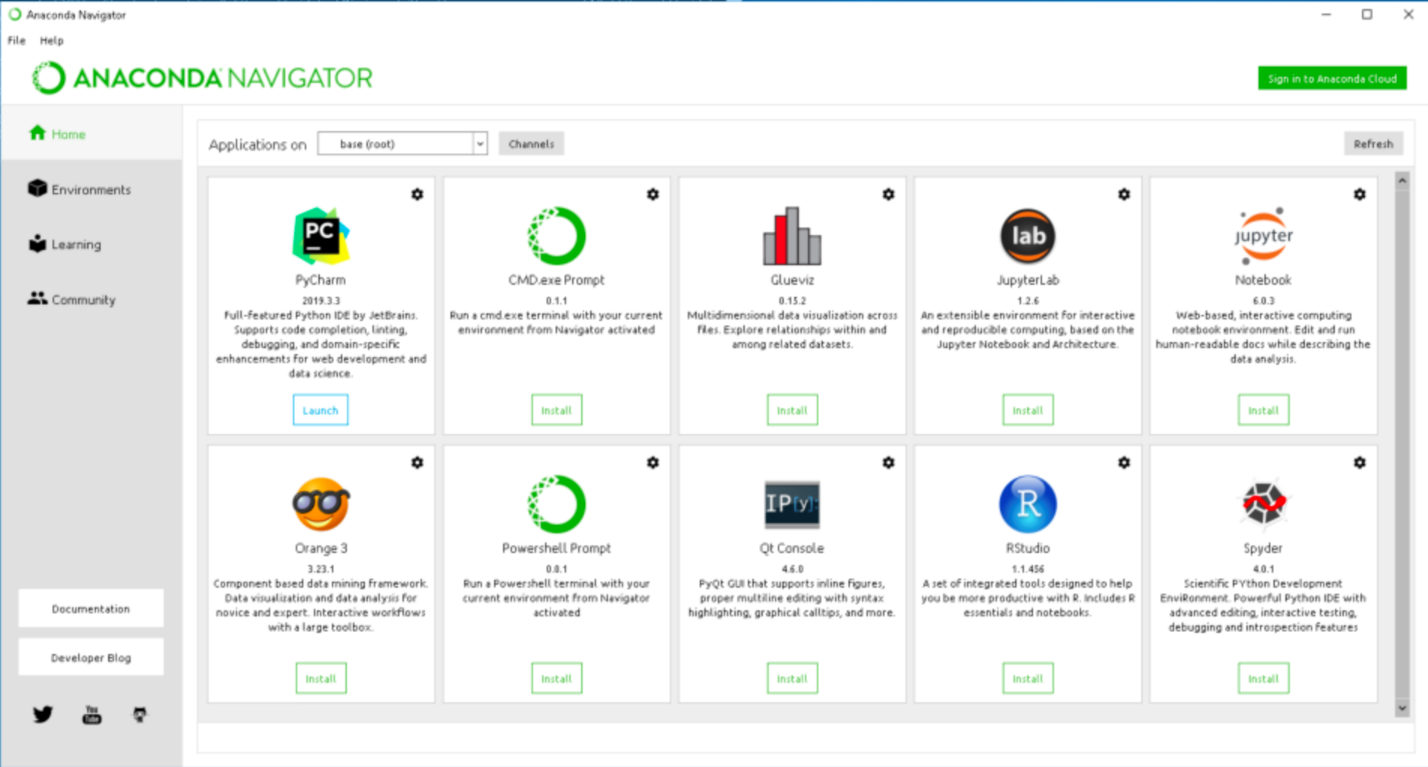
The following applications are available by default in Navigator:

[Jupyter Notebook](https://jupyter.readthedocs.io/en/latest/) ,[Spyder](https://www.spyder-ide.org/), [PyCharm](https://www.jetbrains.com/pycharm/documentation/), [VSCode](https://code.visualstudio.com/docs), [Orange 3 App](http://orange.biolab.si/docs/), [RStudio](http://docs.rstudio.com/), Anaconda Prompt (Windows only), Anaconda PowerShell (Windows only)

### **Benefits of Using Python Anaconda**

* It is free and open-source.
* It has more than 1500 Python.
* Anaconda simplifies package management and deployment
* Anaconda is the industry standard for developing, testing and training on one machine.

**9**



**Fig. Anaconda**

**2.Spyder :-** Spyder stands for SCIENTIFIC PYTHON DEVELOPMENT ENVIRONMENT.

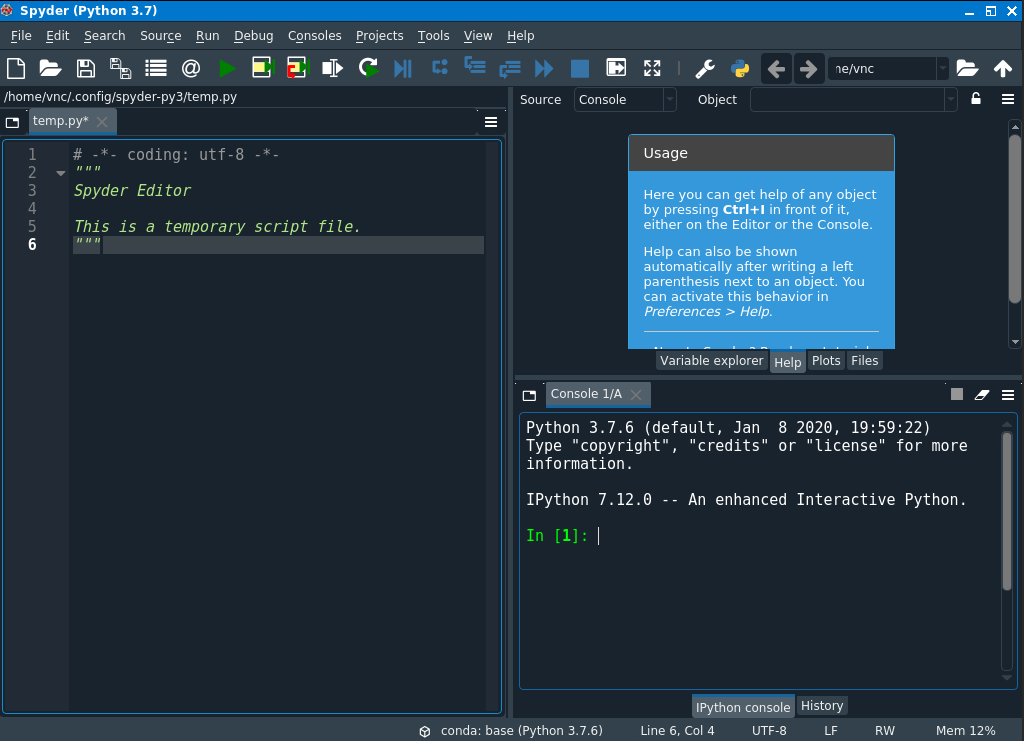
Spyder is a simple and lightweight, powerful interactive development environment for scientific programming in the Python language. This software is open source and cross-platform.

Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda. It is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. The Spyder IDE is written completely in Python. It is designed by scientists and is exclusively for scientists, [data analysts](https://www.edureka.co/blog/data-analyst-roles-and-responsibilities/), and engineers.

## **Features of Spyder :-**

* Python interpreter
* running Python script
* Support for multiple [IPython](https://en.wikipedia.org/wiki/IPython) [consoles](https://en.wikipedia.org/wiki/Command-line_interface)

**10**

* Also provides features such as help, file explorer, find files

**Fig. Spyder**

1. **Pfam:-** The Pfam database is a large collection of protein families, each represented by multiple sequence alignments and hidden Markov models (HMMs). The Pfam website allows users to submit protein or DNA sequences to search for matches to families in the database.

Pfam is a database of curated protein families, each of which is defined by two alignments and a profile hidden Markov model (HMM).

The general purpose of the Pfam database is to provide a complete and accurate classification of protein families and domains. For each protein family in PFAM you can look at multiple alignments, view protein domain architectures, and examine species distribution etc.

Pfam database contains information about protein domains and families. Pfam families are divided into two categories, Pfam-A and Pfam-B.

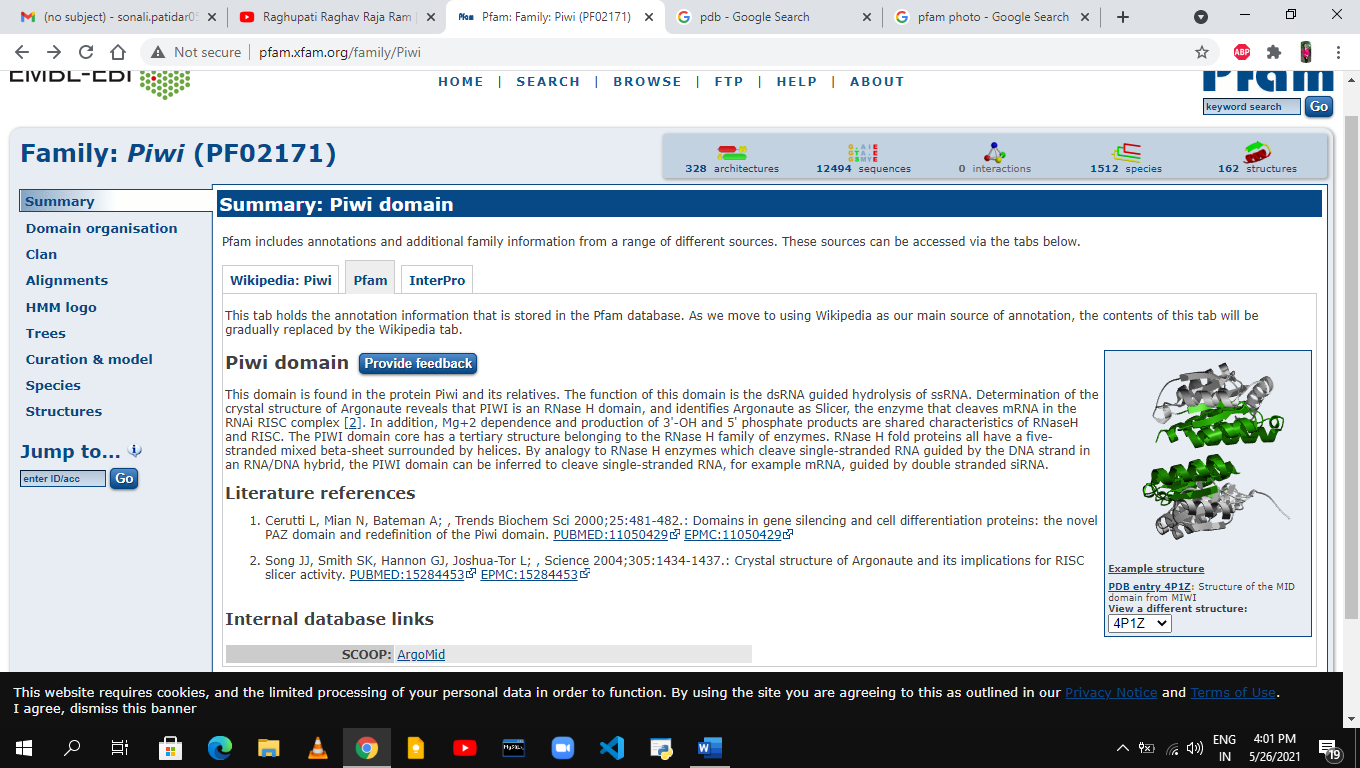
Pfam-A is the manually curated portion of the database.

Pfam-B entries are automatically generated from the ProDom database and are represented by a single alignment.

**11**

**Features of Pfam:-**

* View a description of the family
* Look at multiple alignments
* View protein domain architectures
* Examine species distribution
* Follow links to other databases
* View known protein structures



**Fig. Pfam**

**12**

**Materials and Methods**

1.We have Installed Anaconda python 3.8 version (<https://www.anaconda.com/products/individual>). It creates an environment that’s easily manageable for deploying any project.

2. Learned to code using Spyder, which is an integrated development environment for coding in python.

* When we have installed anaconda, opened anaconda navigator.
* From the home tab of anaconda navigator, launched Spyder.
* When we have launched Spyder, our workspace will have a console in the bottom right corner. use this console to test simple codes and learn python.
* Type print ('hello world') and press enter.
* we had wrote small programs to make ourself familiar with coding in python.

3.After that we have opened Pfam database and downloaded seed alignment of triosephosphate isomerase (PF00121).

* When we have downloaded the seed alignment of triosephosphate isomerase (PF00121), we wrote a small python code in Spyder to read this alignment, found the length of this alignment and also the number of sequences in the alignment.
* We found the lenghth of each sequence.

4.for the frequency matrix, we counted amino acid frequency for each column

**13**

* we created 2d array and after that we counted every column amino acid and extend same for each amino acid.
* we created a new file for the result of each amino acid frequency.

5. After that we summed all amino acid in each row.

6.We have divided each column amino acid by their row sum and calculated frequency calculation.

7. We have calculated Relative Entropy and we used frequency of amino acid in yeast proteome is used as reference.

**14**

**Flowchart**

Pfam

(Seed alignment)

Spyder

(Uploaded file)

Uploaded file

(PF00121)

Calculated the length

Created 2Darray

Count, sum Amino Acid,

Frequency calculated Frequencycalculated

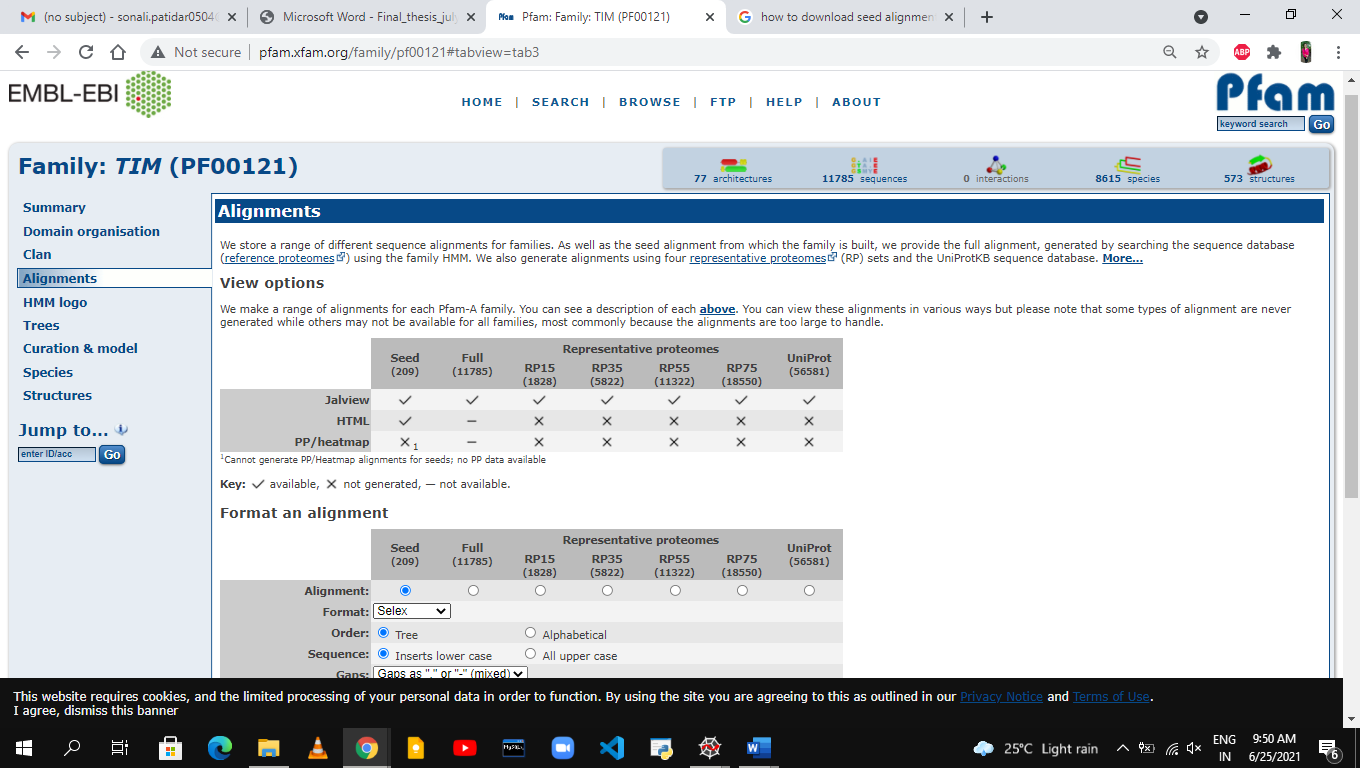
Relative Entropy calculated

**15**

Results

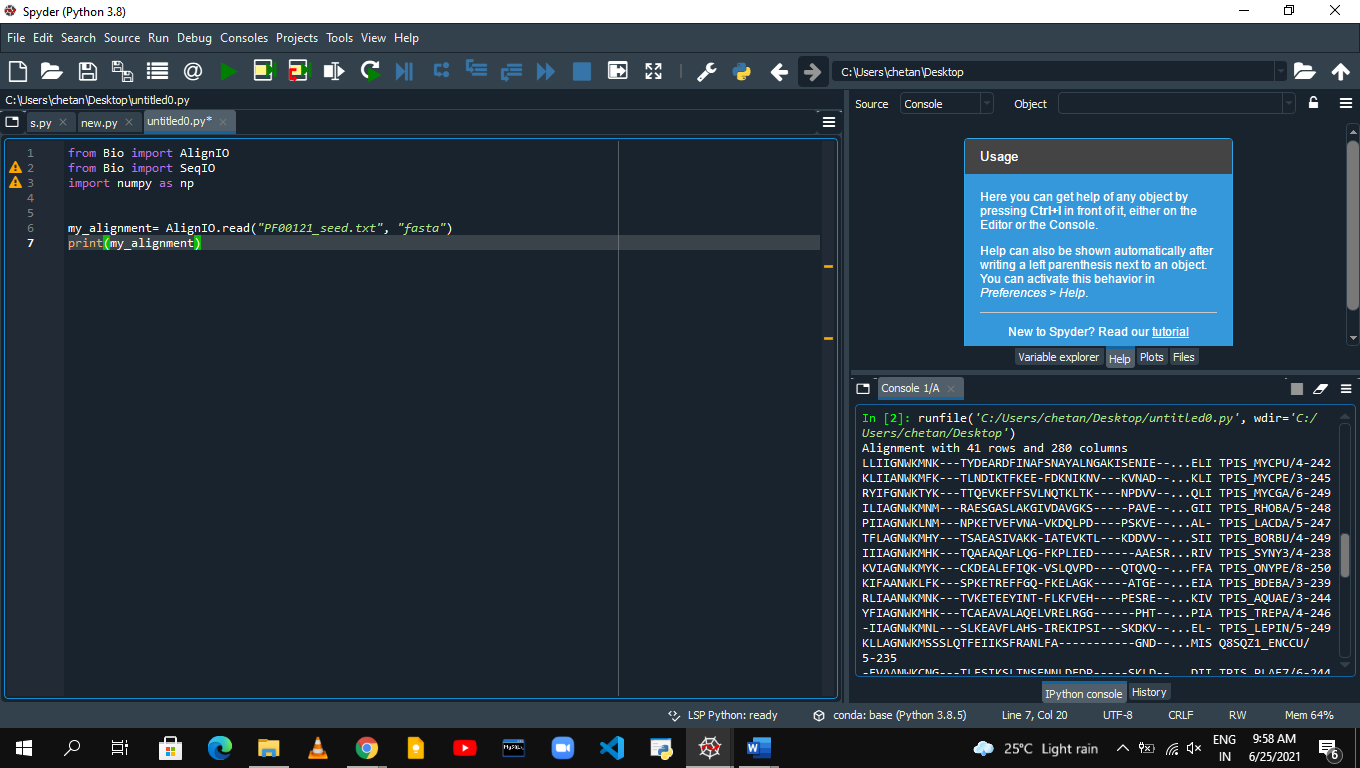
**Result**

1. we have opened Pfam database and downloaded seed alignment of triosephosphate isomerase (PF00121).

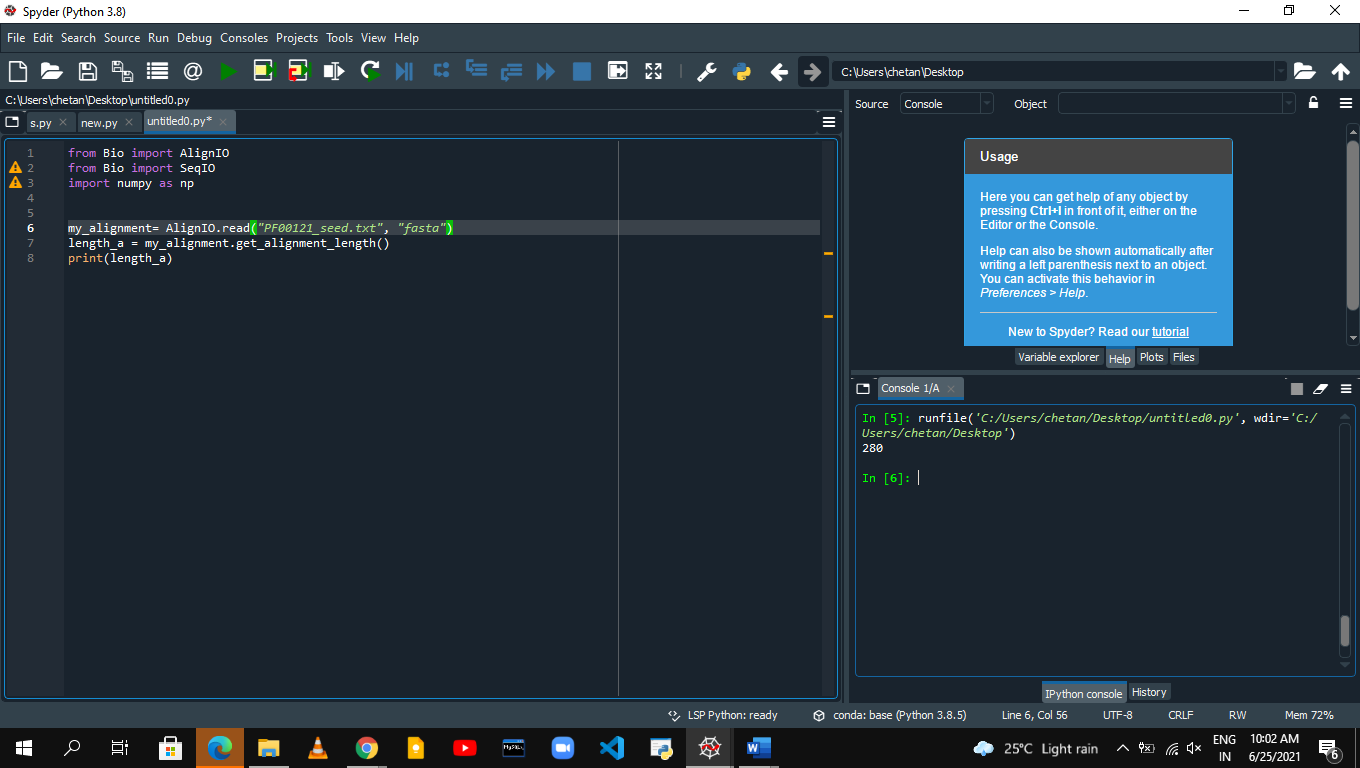


2., we wrote a small python code in Spyder to read this alignment.

**16**

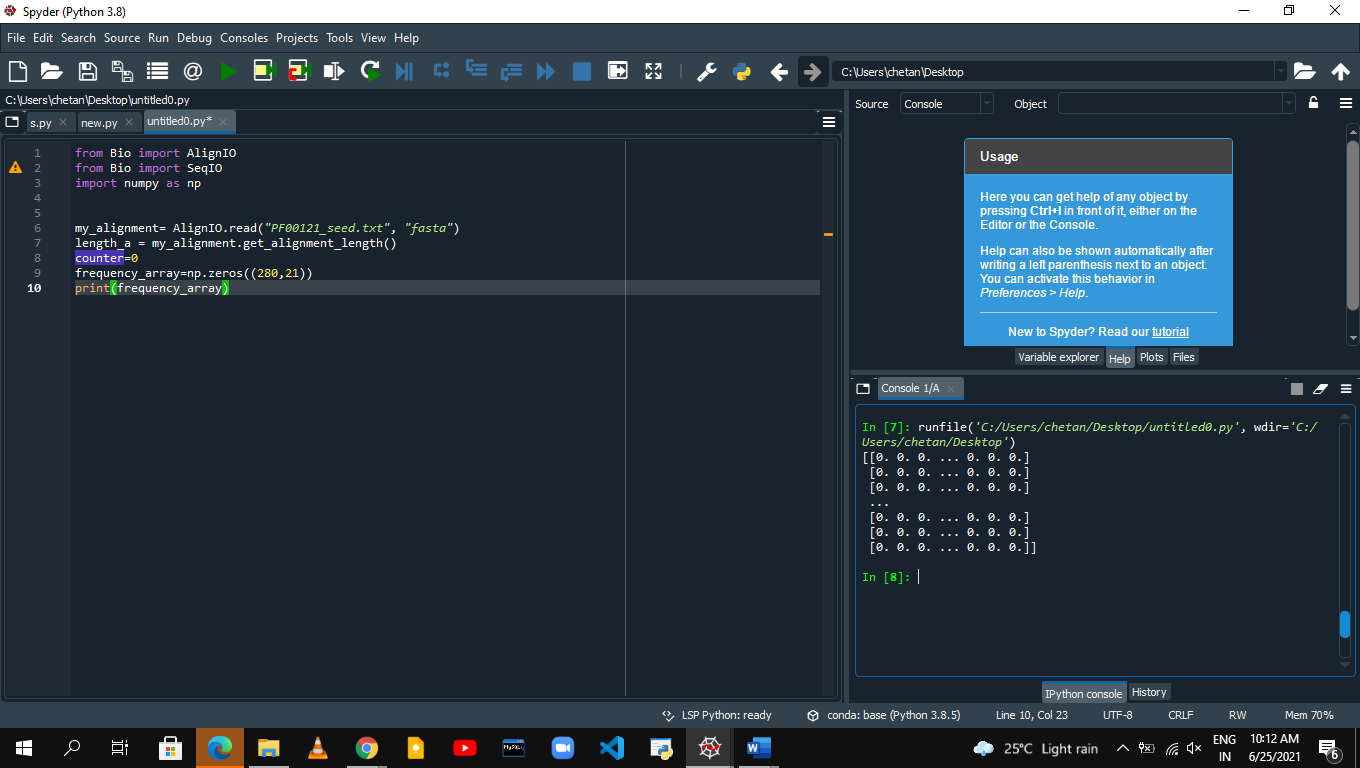


3.length of this alignment and also the number of sequences in the alignment.



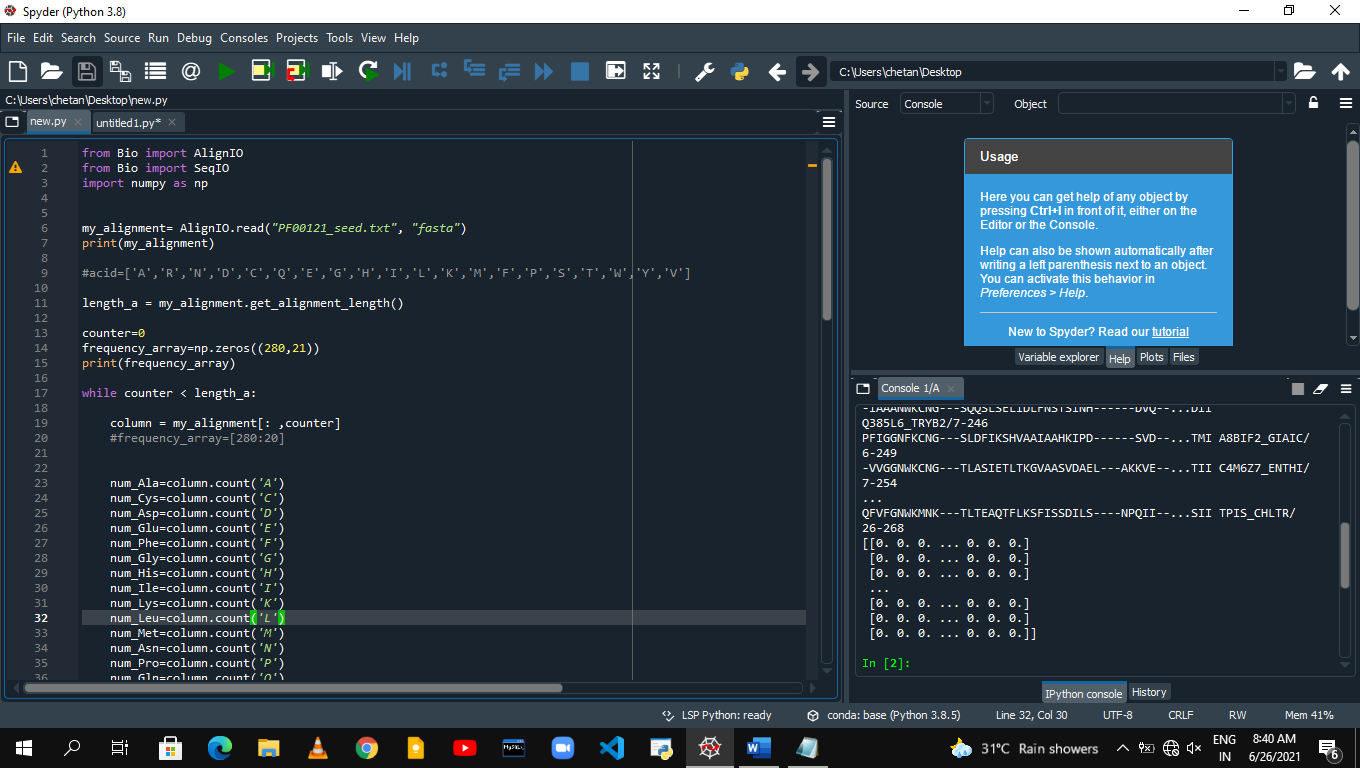
**17**

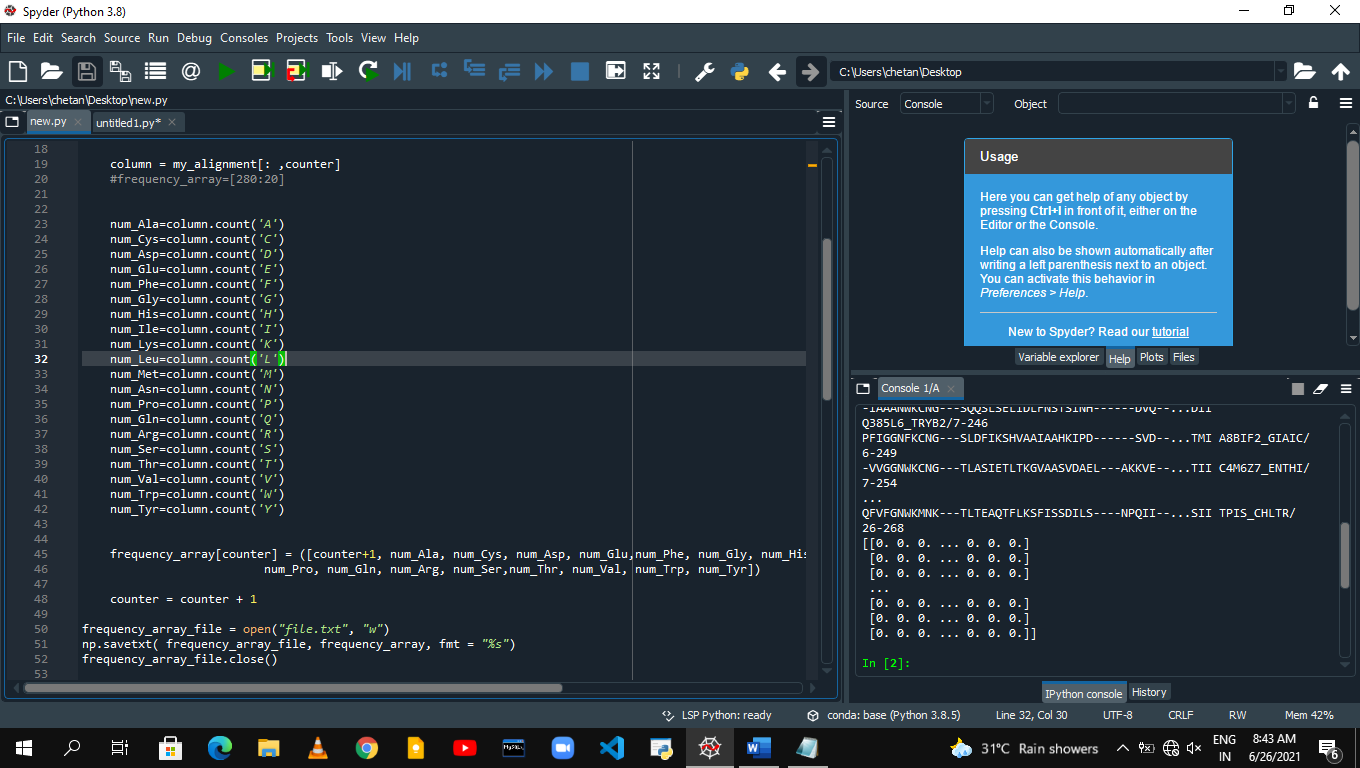
4.For the frequency matrix, we created 2d array.



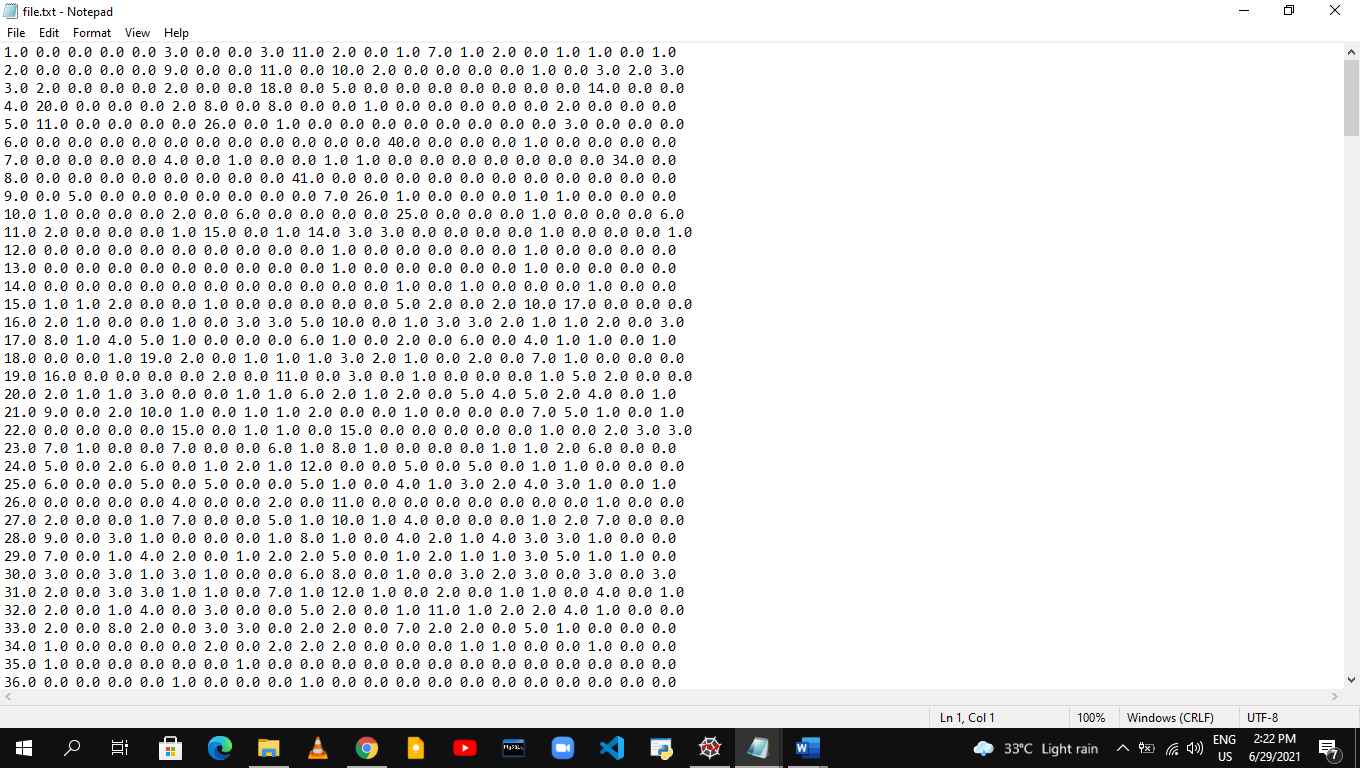
5. we counted amino acid frequency for each column.

**18**

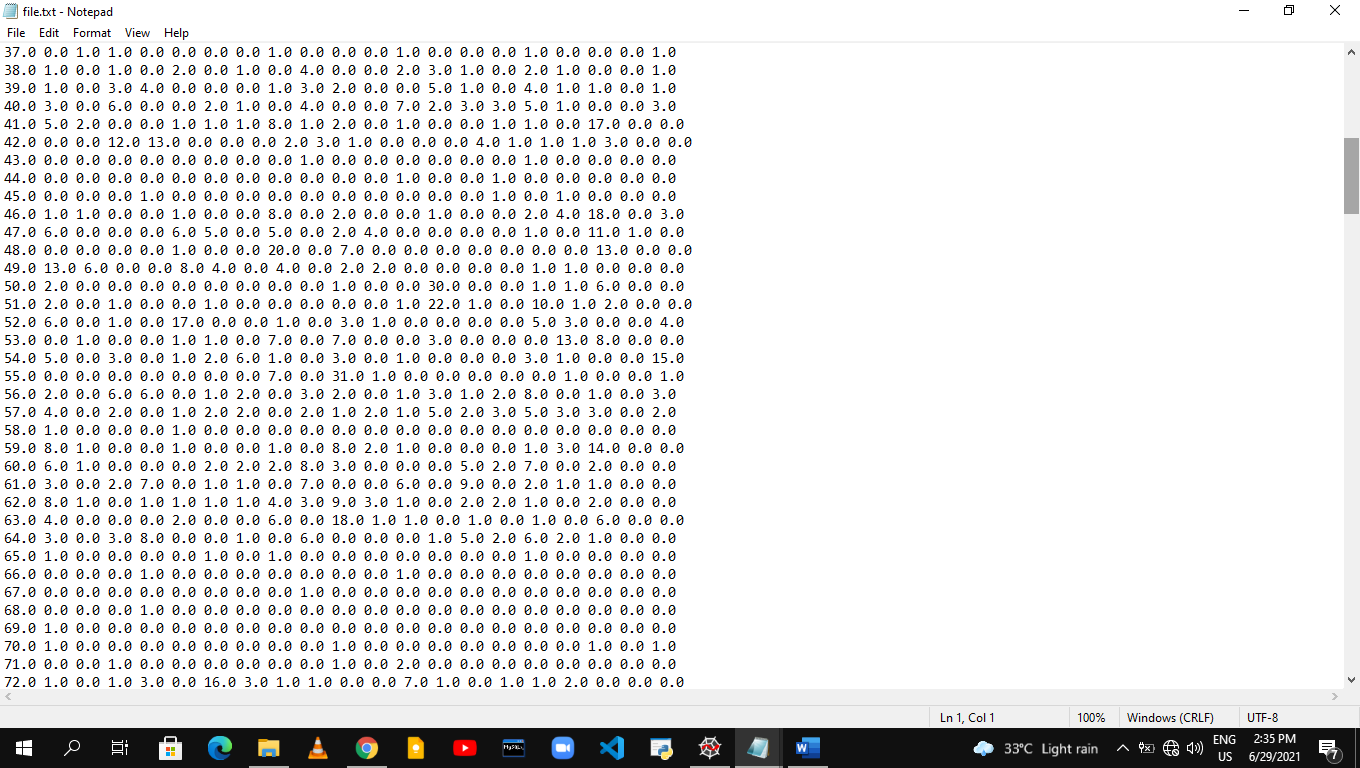




**19**

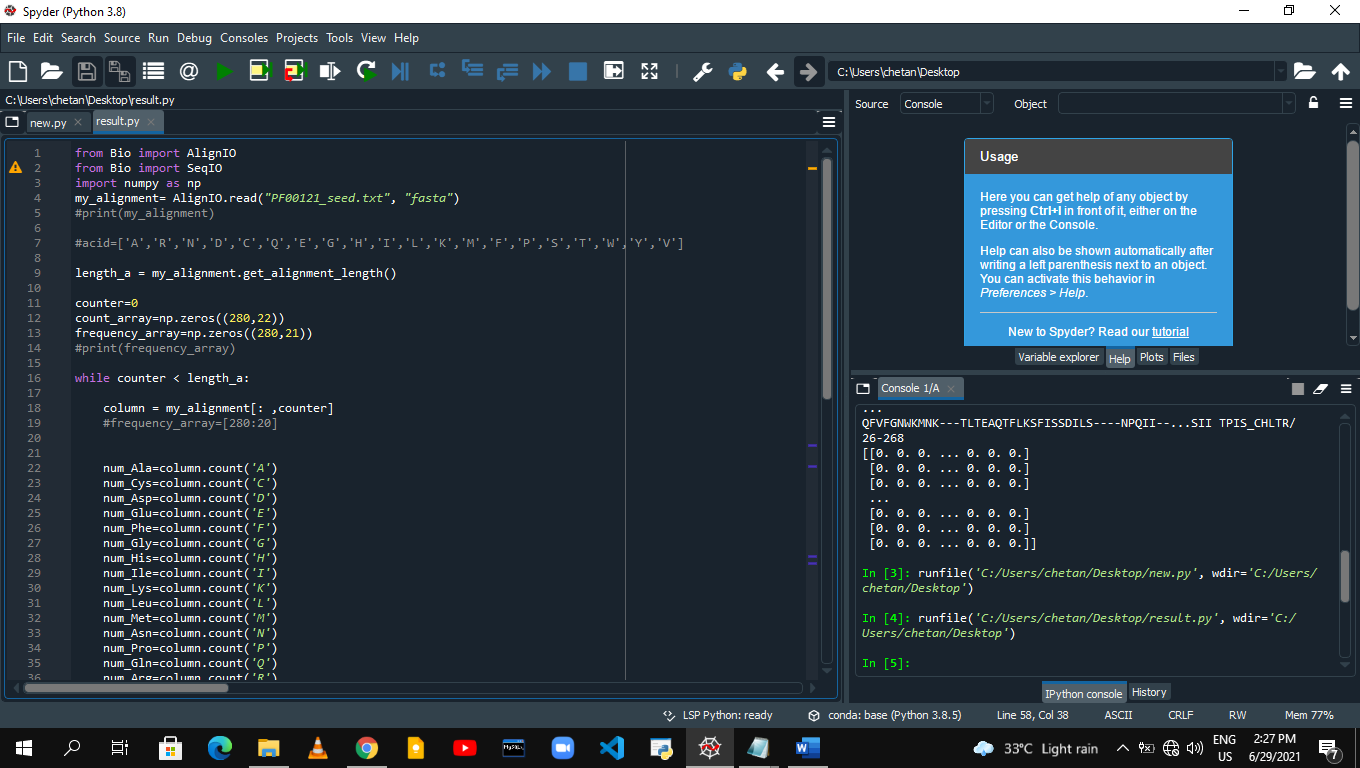


**20**

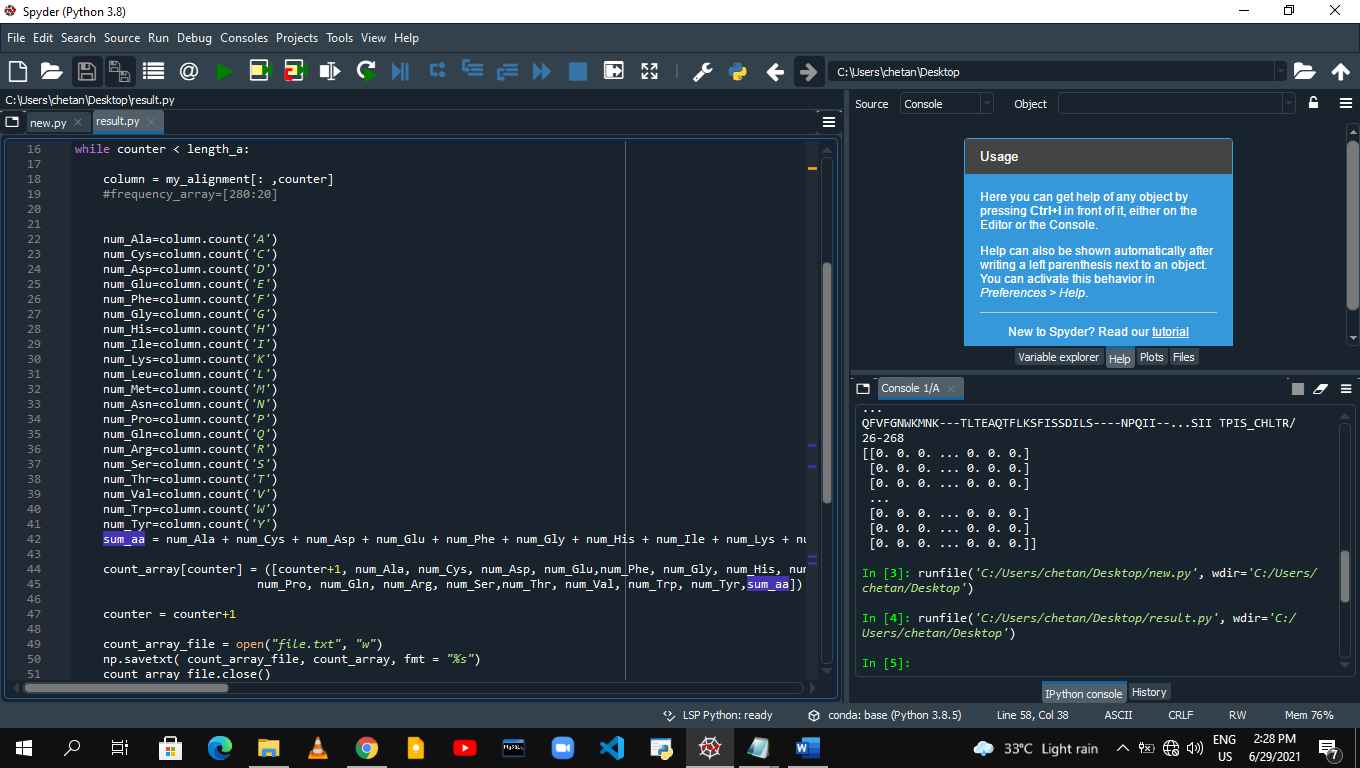


**21**

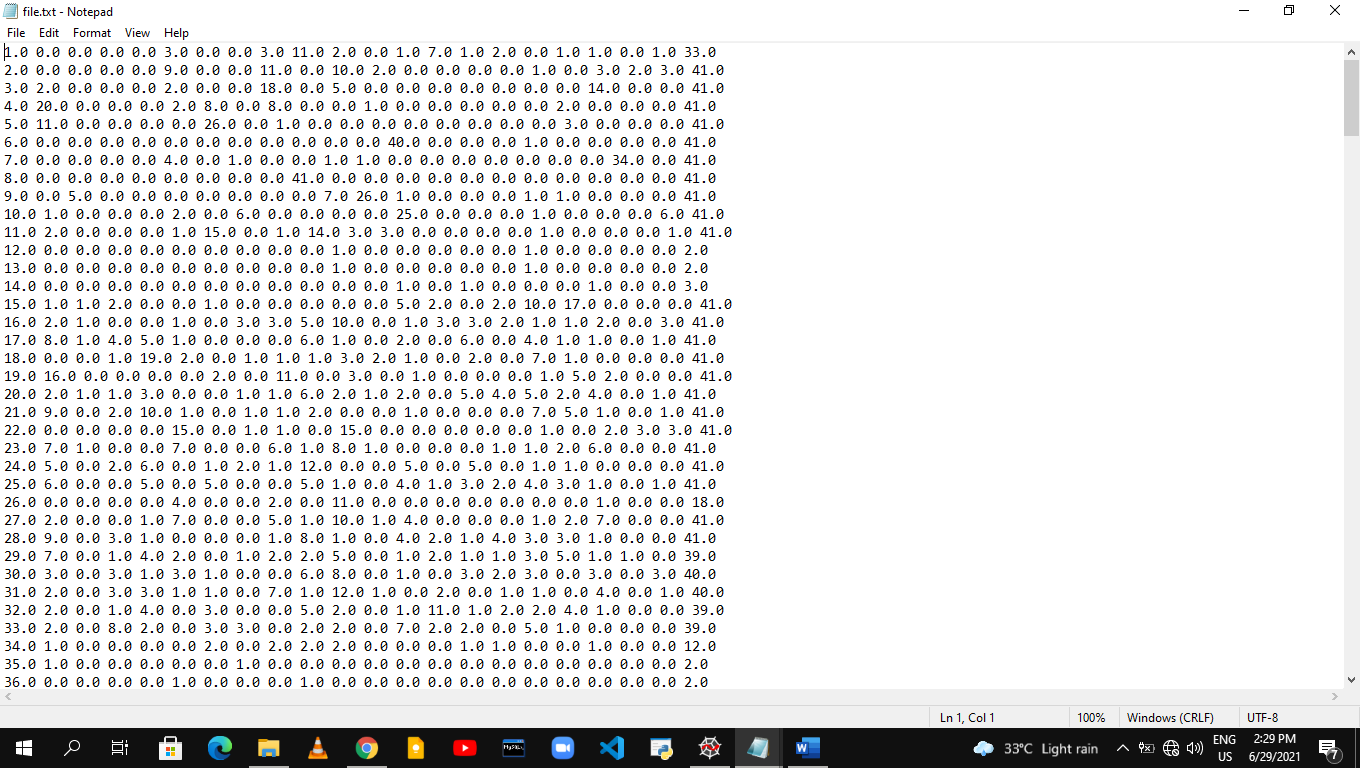
6. After that we sumed all amino acid in each row .



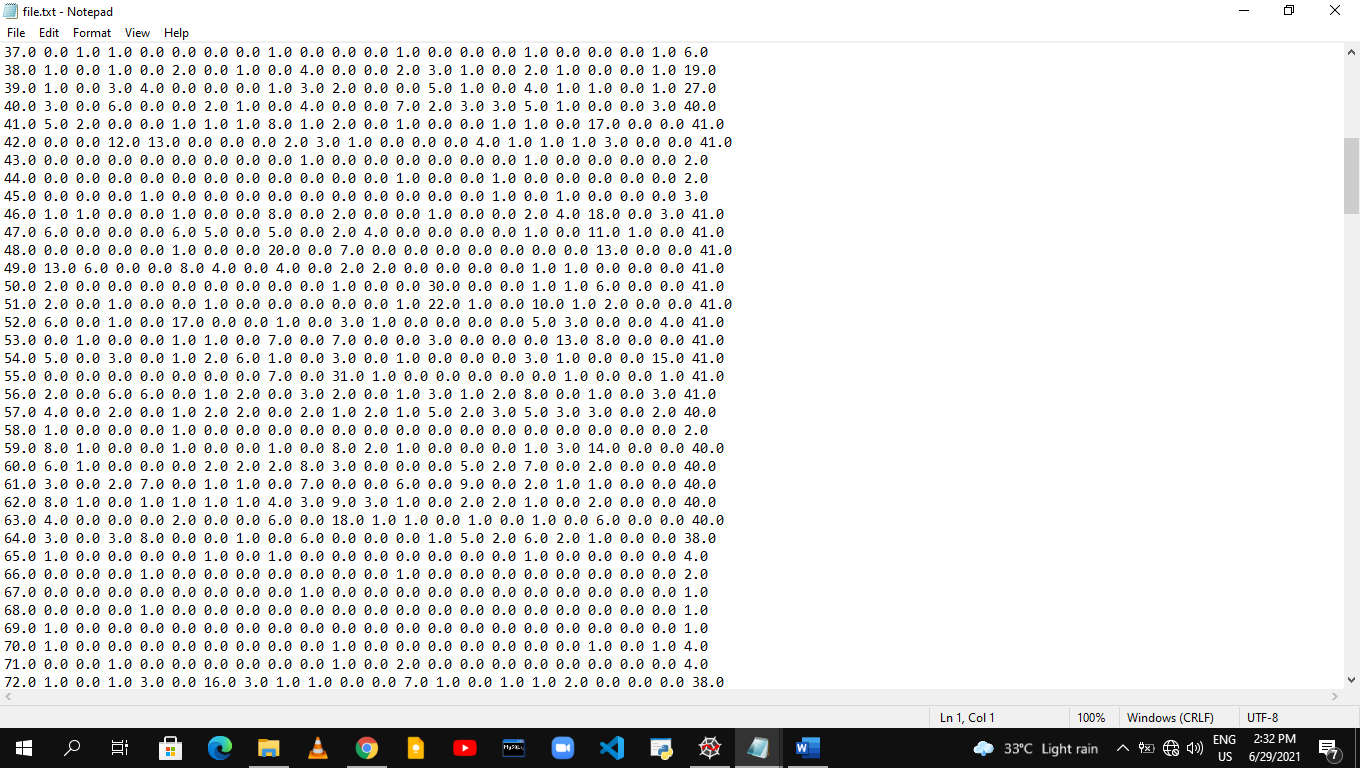
**22**

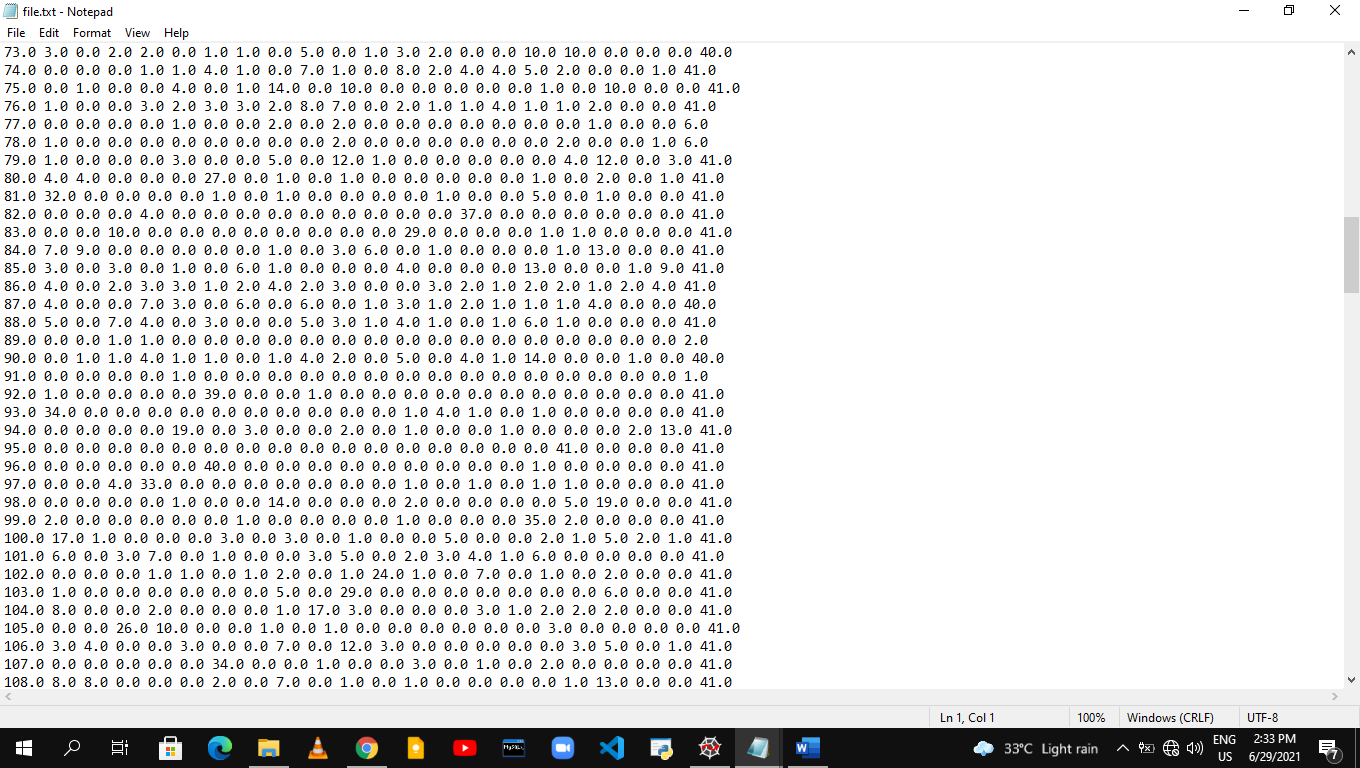


**23**



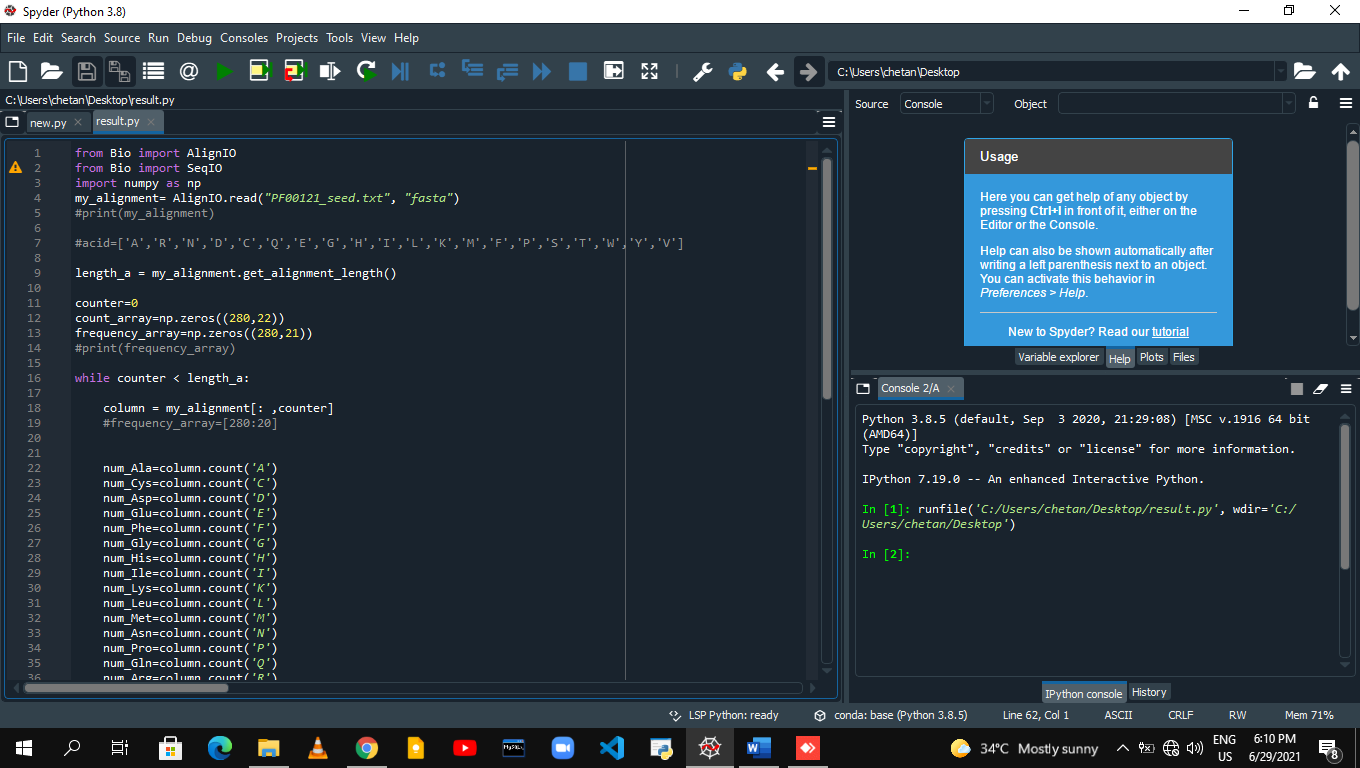
**24**



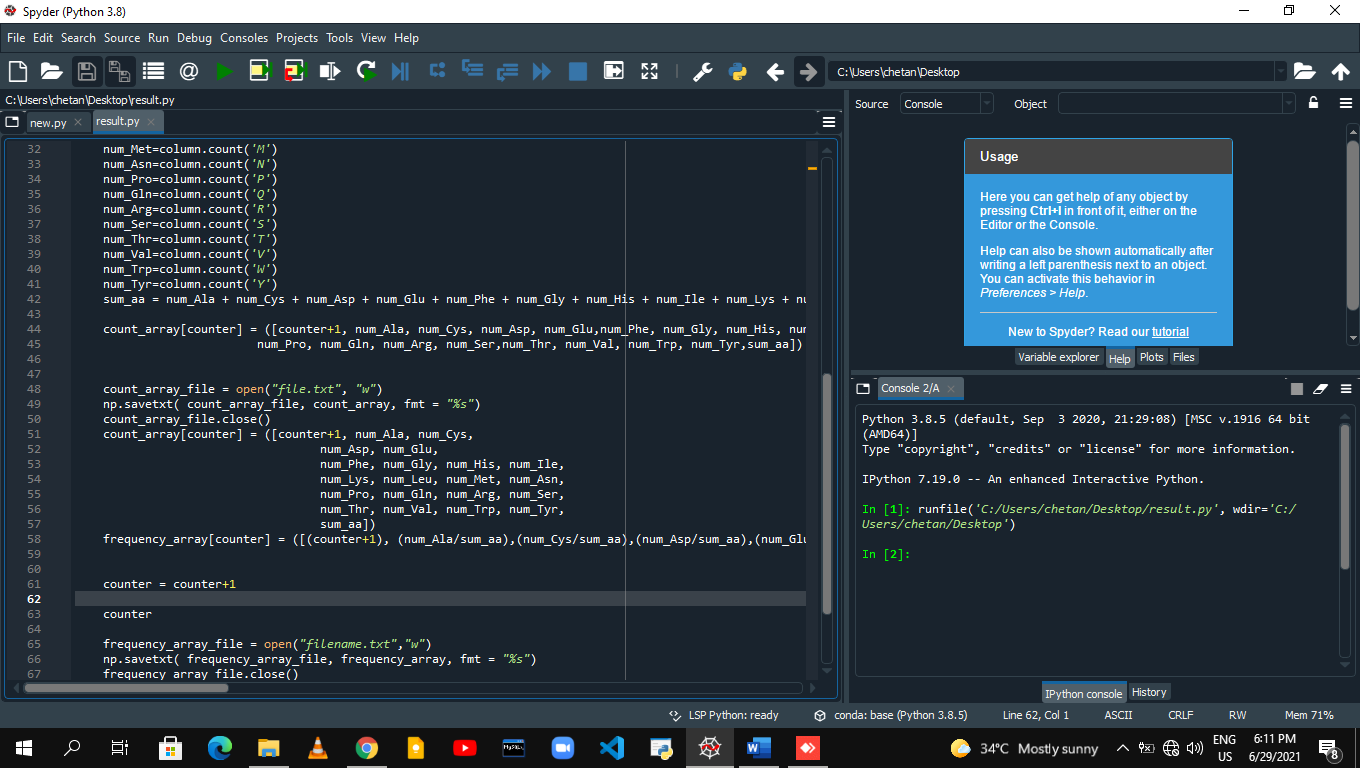


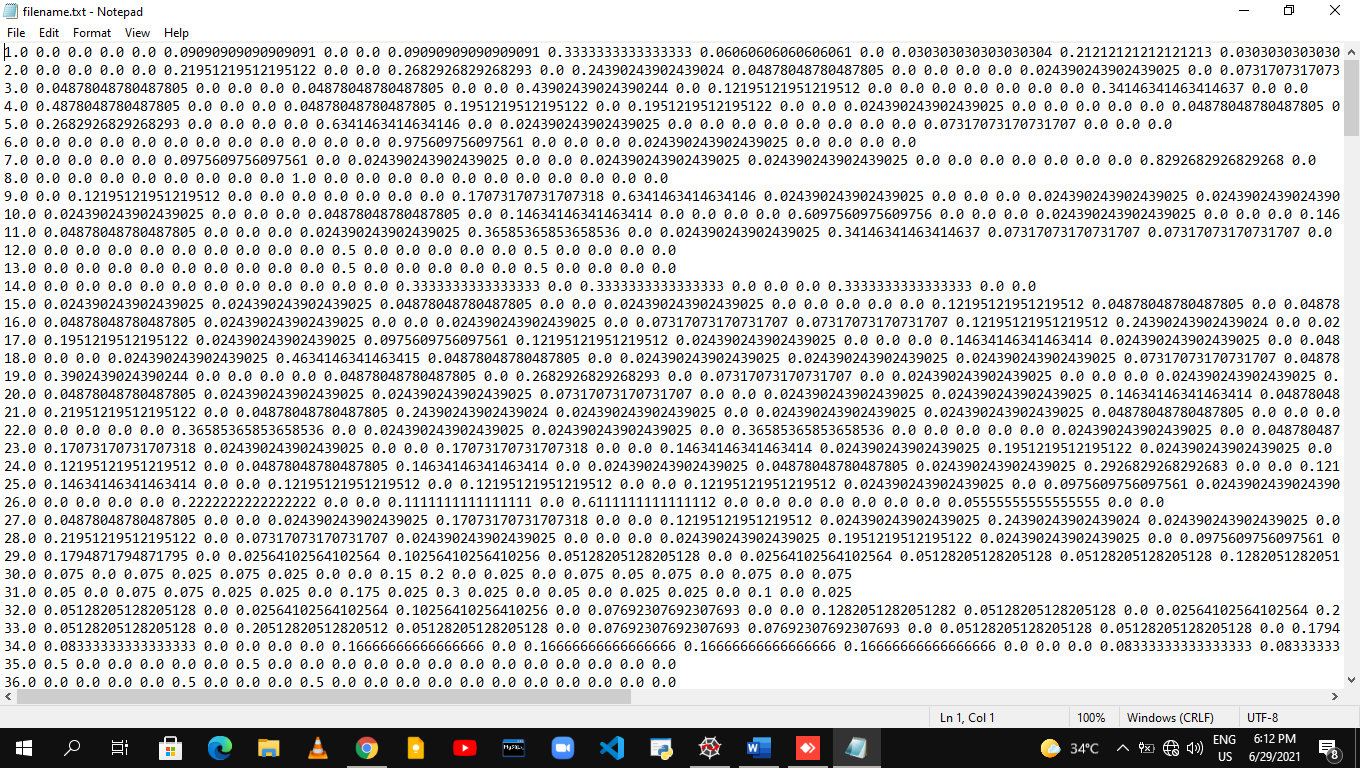
**25**

7. We have divided each column amino acid by their row sum and calculated frequency calculation.

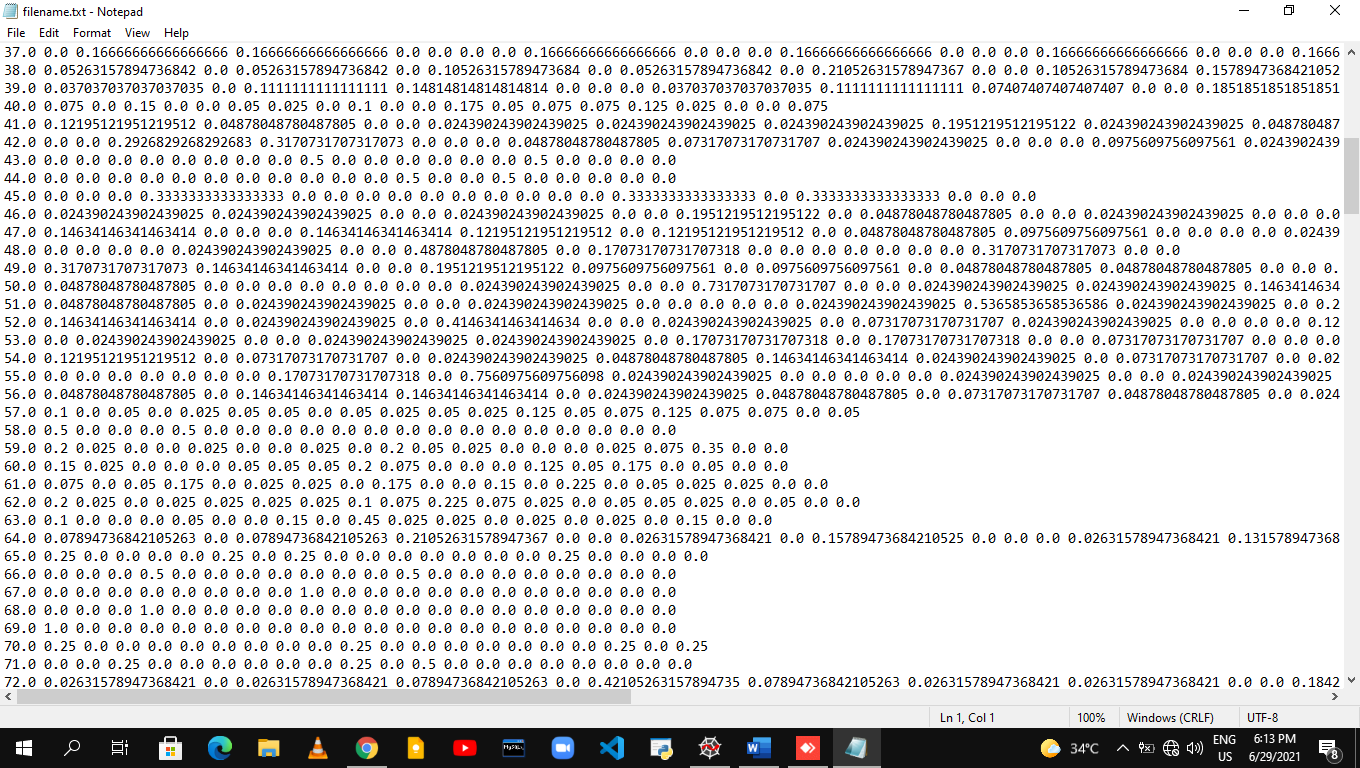


**26**



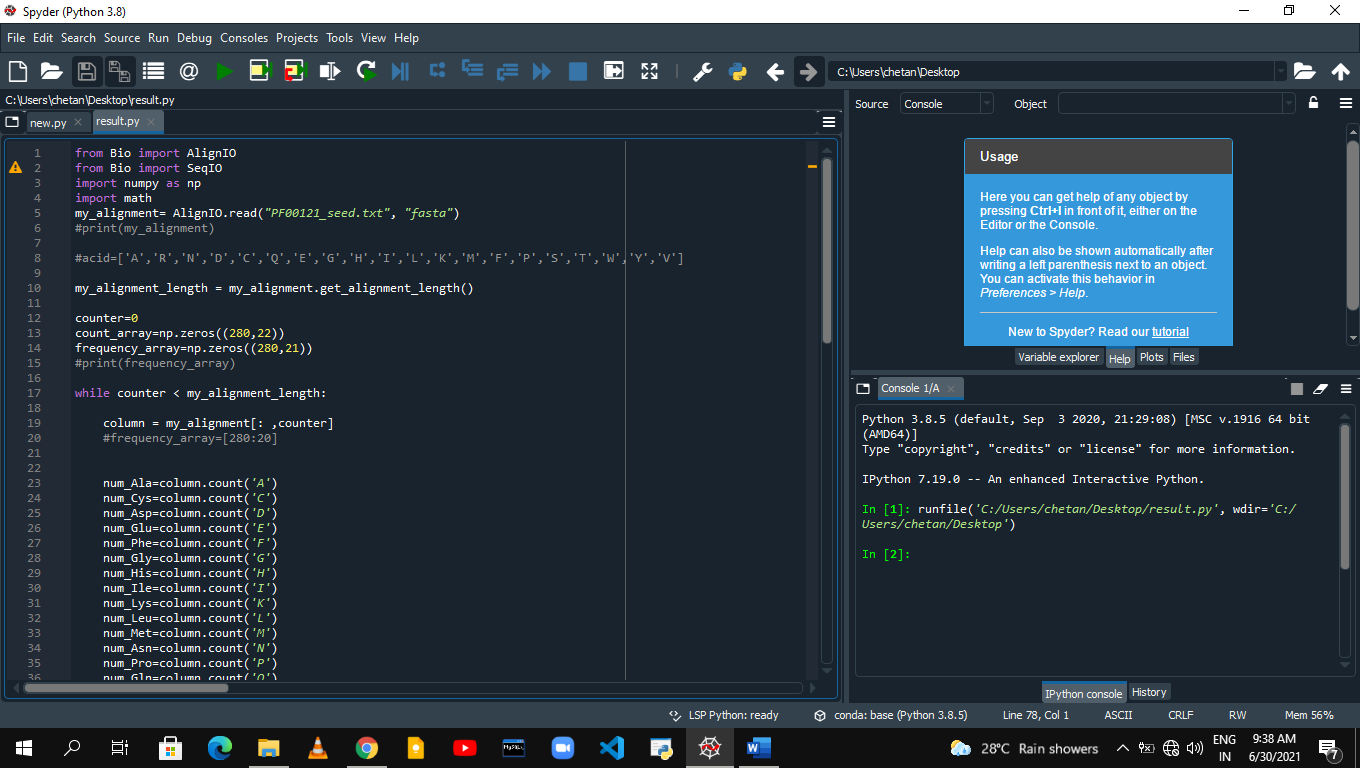


**28**

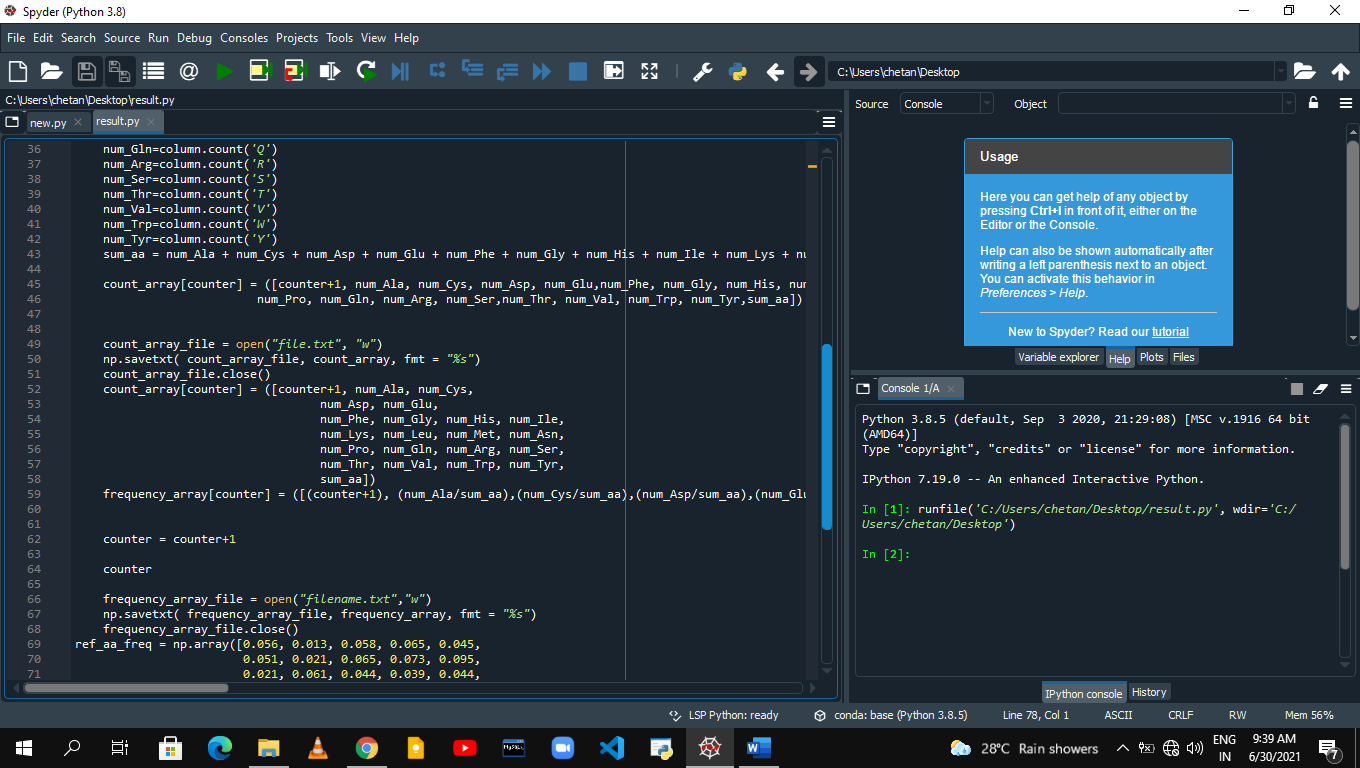


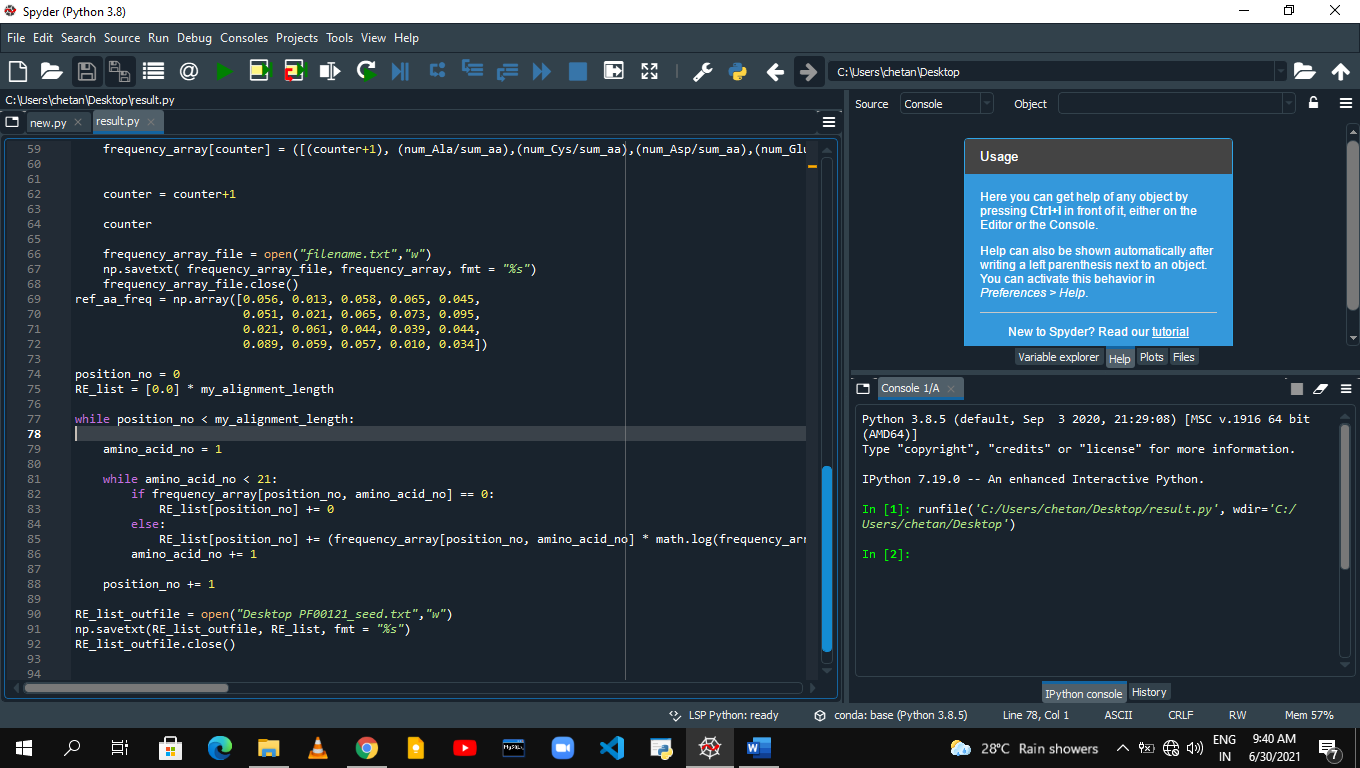
**29**

8**.** We have calculated Relative Entropy and we used frequency of amino acid in yeast proteome is used as reference.

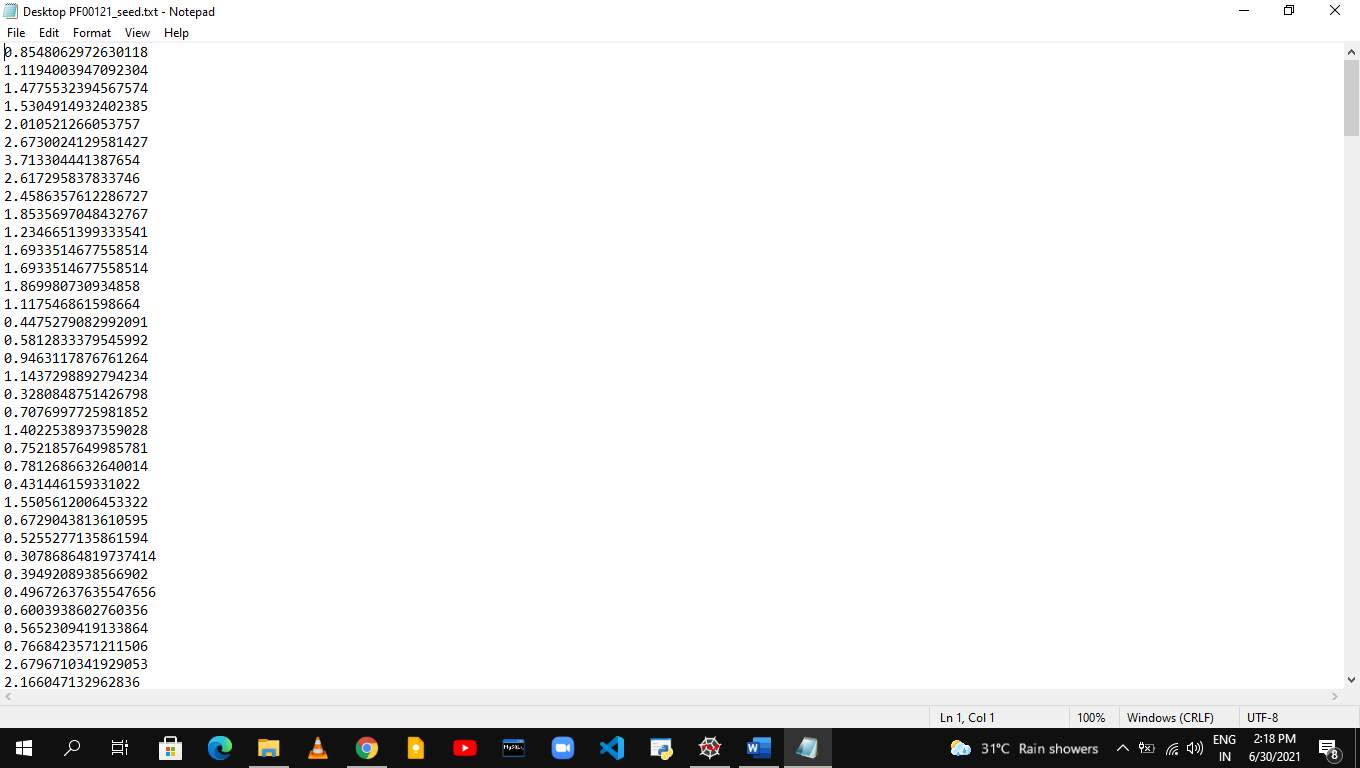


30





**31**



**32**

**Conclusion**

We have provided a protocol for calculating relative entropy and describe how they can be used to predict stability of a protein. Biopython is proposed and successfully used to codefor calculating conservation scores for a multiple sequence alignment. the MultipleSeqAlignment holds the data, and the Bio.AlignIO module for reading and writing them as various file formats. we used MSA for predicting stability of protein. Relative entropy is a measure of conservation, We estimated the conservation of positions in the MSAs using the relative entropy.

**33**

**References**

Sullivan, B. J., Durani, V., & Magliery, T. J. (2011). Triosephosphate Isomerase by Consensus Design: Dramatic Differences in Physical Properties and Activity of Related Variants. *Journal of Molecular Biology*, *413*(1), 195–208. https://doi.org/10.1016/j.jmb.2011.08.001

Sullivan, B. J., Nguyen, T., Durani, V., Mathur, D., Rojas, S., Thomas, M., Syu, T., & Magliery, T. J. (2012). Stabilizing Proteins from Sequence Statistics: The Interplay of Conservation and Correlation in Triosephosphate Isomerase Stability. *Journal of Molecular Biology*, *420*(4-5), 384–399. <https://doi.org/10.1016/j.jmb.2012.04.025>

Goyal, V. D., & Magliery, T. J. (2018). Phylogenetic spread of sequence data affects fitness of SOD1 consensus enzymes: Insights from sequence statistics and structural analyses. *Proteins: Structure, Function, and Bioinformatics*, *86*(6), 609–620. <https://doi.org/10.1002/prot.25486>

Finn, R. D., Mistry, J., Tate, J., Coggill, P., Heger, A., Pollington, J. E., Gavin, O. L., Gunasekaran, P., Ceric, G., Forslund, K., Holm, L., Sonnhammer, E. L., Eddy, S. R., & Bateman, A. (2009). The Pfam protein families database. *Nucleic Acids Research*, *38*(suppl\_1). <https://doi.org/10.1093/nar/gkp985>

Gloor, G. B., Martin, L. C., Wahl, L. M., & Dunn, S. D. (2005). Mutual Information in Protein Multiple Sequence Alignments Reveals Two Classes of Coevolving Positions†. *Biochemistry*, *44*(19), 7156–7165. <https://doi.org/10.1021/bi050293e>

Magliery, T. J., Lavinder, J. J., &amp; Sullivan, B. J. (2011). Protein stability by number: high-throughput and statistical approaches to one of protein science's most difficult problems. Current Opinion in Chemical Biology, 15(3), 443 -451. <https://doi.org/10.1016/j.cbpa.2011.03.015>

Durani, V., &amp; Magliery, T. J. (2013). Protein Engineering and Stabilization from Sequence Statistics. Methods in Enzymology, 237–256. <https://doi.org/10.1016/b978-0-12-394292-0.00011-4>

**34**