

Genomic Data Science And COVID -19 Genome Analysis Using Bio Python

Mansi^{1*}, Dr Vishwajeet Goswami², Dr Aashima Bangia³

¹Department of Mathematics- University Institute of Sciences, Chandigarh University, Gharuan, Punjab, India

²Department of Mathematics- University Institute of Sciences, Chandigarh University, Gharuan, Punjab, India

³Department of Mathematics- University Institute of Sciences, Chandigarh University, Gharuan, Punjab, India

*Corresponding Author: Mansi

¹Department of Mathematics- University Institute of Sciences, Chandigarh University, Gharuan, Punjab, India

Doi: 10.47750/pnr.2022.13. S05.317

Abstract

The last few years have been amazing for biology and healthcare and the best thing that happens in this field was the human genome project. It was an international research project with the goal of mapping and understanding the entire human genome. Data science is the field in which we study data, understand the data to get valuable insight from that. Data science has already become an umbrella under which every industry and field comes. We can work in medical, marketing, Information technology, and every other field, even far before data science comes into the real world, we were using statistical and computational knowledge to get an outcome from our data. But today with the grace of the internet and social media our ability to decipher the information from that data is out of our range, where data science has come as a saviour. Because of the amount of genomic data being generated it becomes essential that the field of genomics or biology must be combined with modern technology and tools so that we can properly analyze such big data for precise and accurate prediction of disease and prevention mechanisms for that, which ultimately will result in improved human health. The data collected from a single week-long sequence today can create more data than whole genome research done a few years ago. "Bioinformatics", "computational genomics" and "genomic data science" are all very similar fields. To provide biological insights in these disciplines, we must be able to process and analyse huge genomics datasets, as well as validate the processed data's quality and transform it. Afterwards, depending on the nature of our issue, we must apply statistical or machine-learning models. The most likely scenario is to first perform some dimension reduction and clustering, then visualisation. In this paper, I will use python to pass my accountancies over genomic data science and genomic data analysis.

Keywords: Data science, genomics, python, data analysis, bioinformatics, bio python.

INTRODUCTION

As we are digging deeper into the field of healthcare, medical, biology, and genomics, the data generated from it is increasing enormously. One has to use the static approach and computational tool to analyze the diversified genomic data and then process it, store it and apply it to the algorithms. Genomic data science mainly involves the practical technical aspect of how one can deal with the data as the data getting generated from sequencing machines is in really huge amounts which we also term as big data. I will try to cover every basic step of genomic data science analysis using python. Genomics is all about DNA code and analysis of genomic data. So let's start with DNA.

DNA

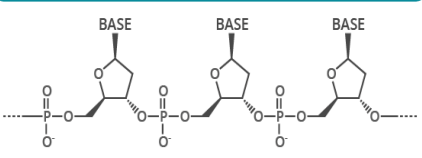
Deoxyribonucleic Acid is what it stands for. It is a molecularly distinct organic substance that is present in all prokaryotic and eukaryotic cells. It is a collection of molecules in charge of transferring genetic information or hereditary materials from one generation to the next. In the genomics analysis, we have to consider these two things, the first is DNA code, which is the information that we have already in hand and what we would like to ask or want to know from that data or code. Second is the sequencing, function, and structure which is what information is stored in the cell and how can we get insights from that for our use. DNA code is organized just like a book some letters form sentences which ultimately form paragraphs and the paragraphs are organized in the form of chapters, just as how helpful it can be if we know how to read that book's concepts apply them in questions and our real life, in the same manner, knowing how to read that codes can tell us the whole story about whom we are as a biological being and also defining the function of our fundamental constitution of a living cell.

DNA is made up of two strands, each of which is made up of nucleotides that twist around a central axis to form a structure known as a double helix. Three molecules make up a nucleotide: a sugar molecule, a phosphate group, and a nitrogenous base. Deoxyribose is the sugar found in DNA's nucleotides. The sugar-phosphate backbone of the DNA strand is formed when covalent connections are formed between the sugar of one nucleotide and the phosphate group of the following nucleotide. DNA contains nitrogenous bases, which are organic molecules with nitrogen and base-like chemical characteristics. DNA molecules contain the nitrogenous bases cytosine, guanine, adenine, and thymine, which are denoted by the letters C, G, A, and T, respectively. The nitrogenous base sequence on one strand of a double helix aligns precisely with the sequence on the other strand. Only cytosine and guanine pair with each other, while adenine and thymine do.

Each gene of the genome act as a recipe for a certain protein. A specific sequence of nitrogenous bases along the strand encodes particular RNA molecules. These sequences are called genes. mRNA molecules which are transcribed from genes through a process called transcription are then later translated into proteins through the process of translation. The image attached below shows the double helix structure of the DNA and the transcription and translation process. Hydrogen bond present in the DNA strand, how adenine is held together to thymine and guanine to cytosine. We can also see that in mRNA uracil is present in the place of thymine.

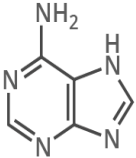
THE CHEMICAL STRUCTURE OF DNA

THE SUGAR PHOSPHATE 'BACKBONE'

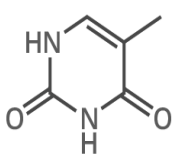


DNA is a polymer made up of units called nucleotides. The nucleotides are made of three different components: a sugar group, a phosphate group, and a base. There are four different bases: adenine, thymine, guanine and cytosine.

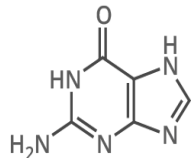
A ADENINE



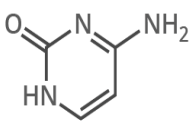
T THYMINE



G GUANINE

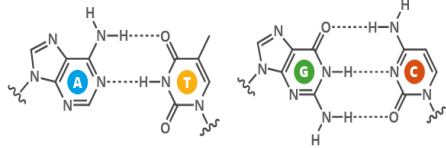


C CYTOSINE



WHAT HOLDS DNA STRANDS TOGETHER?

DNA strands are held together by hydrogen bonds between bases on adjacent strands. Adenine (A) always pairs with thymine (T), while guanine (G) always pairs with cytosine (C). Adenine pairs with uracil (U) in RNA.



FROM DNA TO PROTEINS

The bases on a single strand of DNA act as a code. The letters form three letter codons, which code for amino acids - the building blocks of proteins.

DNA

→

RNA

→


PROTEIN

TRANSCRIPTION TRANSLATION

An enzyme, RNA polymerase, transcribes DNA into mRNA (messenger ribonucleic acid). It splits apart the two strands that form the double helix, then reads a strand and copies the sequence of nucleotides. The only difference between the RNA and the original DNA is that in the place of thymine (T), another base with a similar structure is used: uracil (U).


DNA SEQUENCE	T	T	C	C	T	G	A	A	C	C	G	G	T	T	A
mRNA SEQUENCE	U	U	C	C	U	G	A	A	C	C	G	G	U	U	A
AMINO ACID	Phenylalanine		Leucine		Asparagine		Proline		Leucine						

In multicellular organisms, the mRNA carries genetic code out of the cell nucleus, to the cytoplasm. Here, protein synthesis takes place. 'Translation' is the process of turning the mRNA's 'code' into proteins. Molecules called ribosomes carry out this process, building up proteins from the amino acids coded for.



© Andy Brunning/Compound Interest 2018 - www.compoundchem.com | Twitter: @compoundchem | FB: www.facebook.com/compoundchem

This graphic is shared under a Creative Commons Attribution-NonCommercial-NoDerivatives licence.



Through the process of transcription and translation, the information is passed on from the gene or DNA to the protein. At each step of this transformation, we can see how the DNA code is being used to get the information. First of all, we transcribe the source code in the working repository of RNA, where some code is modified and released to produce the protein. The gene sequence encodes the instruction from which a protein can be made which is also called a codon. We have learnt enough about DNA till now so now let's move toward the composition of our genome data because that is what matters to us being data scientists.

GENOME COMPOSITION:

Genome consists of many elements other than genes one can see in the image attached below that even less than 1% of the whole genome composition is constituted of a transcribable exonic element gene. In humans, these genes can vary from a few hundred DNA bases to more than 2 million bases. this size of genes varies between organisms too for example influenza virus has only 11 genes whereas the grape genome has 30,000 genes and a whole human genome has about three billion base pairs of DNA that are precisely organised to give us our basic anatomy as well as our distinctive traits like height and eye colour, among other things. these differences between genes present organism form the basis of the difference between all life forms we see around us.

Data processing: In this stage, we transform our data into a format that is appropriate for the exploratory analysis and modelling steps that follow. The data frequently arrives unprepared for analysis. Therefore, we must transform data points to translate data into different formats.

Exploratory Data Analysis and Modeling: In this stage, we use machine learning and statistical techniques to explore the data and examine the relationships between the variables. We start with the processed data from the previous step. The following stage is modelling. Using genomics, we might be attempting to infer the patients' disease condition from the expression of their genes. Regression machine learning techniques can be used to resolve this kind of methodology, which is referred to as predictive modelling.

Visualization and reporting: While nearly every phase benefits from visualisation, the final step requires tables, texts, and figures that explain the findings of our investigation.

Let's jump to our jupyter notebook

First of all, we need to install the packages like biopython and squiggle which make our dealing with the biological sequence data easier. Then we need to import the dependencies.

```
In [1]: import numpy as np
import pandas as pd
pd.plotting.register_matplotlib_converters()
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import os

In [2]: import Bio.SeqIO
```

The bio.seqIO module from biopython will be used to parse the DNA sequence data. It offers a straightforward, uniform interface for importing and exporting several sequence file formats. It will use the sequence and the length of the sequence

```
In [41]: for sequence in Bio.SeqIO.parse('MN908947.fna', "fasta"):
print('Id: ' + sequence.id + '\nSize: ' + str(len(sequence))+' nucleotides')
print(sequence.seq)
print(len(sequence), 'nucleotides')
```

```
from Bio.SeqRecord import SeqRecord
from Bio import SeqIO
DNAsequence = SeqIO.read('MN908947.fna', "fasta")
```

```
DNAsequence

SeqRecord(seq=Seq('ATTAAAGGTTTATACCTTCCAGGTAACAAACCAACCACTTTCGATCTTTGT...AAA'), id='MN908947.3', name='MN908947.3', description='MN908947.3 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome', dbxrefs=[])
```

SeqIO will give us all the basic information of the sequence given. Our input sequence which is FASTA is a DNA sequence and coronavirus is an RNA type of virus so we need to transcribe our DNA sequence to the RNA sequence and then translate RNA to the amino acid sequence. So, in DNA the bases thymine(T) is present which gets replaced by base Uracil (U) in the RNA. Using transcribe (), the DNA will be changed into an mRNA sequence. The next step is to use the Translate () method to convert the mRNA sequence to the amino acid sequence.

```
DNA = DNAsequence.seq#Convert DNA into mRNA Sequence
mRNA = DNA.transcribe() #Transcribe a DNA sequence into RNA.
print(mRNA)
print('Size : ',len(mRNA))
```

```
AUUAAAGGUUUUUAUACCUCCAGGUUAACAAACC AACCAUUUCGAUCUUCUUGAGAUUCUGUUCUUAACGAUCUUUAAAUCUGUGUGGUGUCACUCGGCUGAUGCUUAGUGCACUCACGC
AGUUAUUUUUAUUAUUAUACUUGUCUGUAGCAGGACAGAGUUAUCUUCUUCUUCGAGGUGUCUUAUGGUGUUCGUCGUGUUGGACGCAUCUAGGUGUUCGUCGGGGUG
UGACGGAAGGUUAAGUAGGAGAGCCUUGUCCUGGUUUUUAACGAGAAAACACACGUCUUAUCAGUUUUCUGUUUUUACAGGUUUCGACGUGUCUUAUGGUGUUCGUGGAGACUCCGUGGAGGA
GGUUCUUAUCAGAGGCACGUCUUAUCUUAAGAUGGACUUGUGGCUUAGUAGAAGUUGAAAAGGCGUUUUUGCCUUAUCUUAAGCAGCCUUAUGUGUUAUCUUAAGUUCGGAUGUCGGAACUG
CACCUCUAGGUGUUAUUGGUUGAGCUGGUUAGCAGAUCUCGAAGGCAUUCAGUACGGUUCUAGUGGUGAGACACUUGGUGUCUUGUCUUAUCUUGGUGGAGAAUUAUCAGUUGGCUUACCGCAAG
GUUCUUCUUAUUAAGAACGUAUUAAGGAGCUGGUGGCCAUAGUUAUCGGCCGCAUCUUAAGUUAUUGAGUUAAGGCGACGAGCUUGGACUUAUCUUAUUAAGAUUUUAAGAAAACUGGAA
CACUAAACUUAAGCAGUGGUGUUAUCCGUGAACUUAUCGUGAGCUUAACGGAGGGGCAUACACUCGCUUAUGUGUAUUAACUUCUUGUGGCCUGAUGGCUUACCUUUGAGUGCAUUAAGAAC
```

```
Amino Acid = mRNA.translate(table=1, cds=False)
print('Amino Acid', Amino Acid)
print("Length of Protein:",len(Amino Acid))
print("Length of Original mRNA:",len(mRNA))
```

```
Amino Acid IKGLYLPR*QTNQLSISCRSVL*TNFKICVAVTRLHA*CTHAV*LITNYCR*QDTSNSIFCRLLTVSSVLQPIISTSRFRPGVTER*DGEPCPNWFQRENTPTQFACFTGSRRAR
TNLWLRGGGLIRGTSTS*RWHLWLSRS*KRRFAST*TALCVHQTFGCSNCTSWSCY*AGSRTRRHVSRS*W*DTNCPSPCGRNTSGLPQGSSES*ER**RSWMP*LRRRSKVI*LRRRAWH*SL*
RFSRKLH*TW*QWCP*THA*A*RRGIHSLCR*QLLWP*WLP*VH*RPSS*TW*SFMHFVRTTGLY*H*EGCILLP*TA*INCLVHGT*F*KEL*ITADTF*N*IGKEI*HLQNGMSKFCISLKFHNQ
DYSTKG*KEKA*WLYG*NSICLSSCVTK*MQPNVFPNSHEV*SLW*NFMDGRFC*SHLRILWH*EFD*RRCHYLWLTPKCCC*NLSSMSQFERSR*A*SCRIP**IWLENHSS*GWSHYCLWRL
CVLLCWL*P*QVCLLGT*CR*HRL*PYRCNRRFRS**QPS*NTPKRESQHQYCW*L***RDRHYF*GFFCFHCKFCGNCERFGL*SIQTN*ILW*F*SYKRKS*KRCLEYW*TEINTESSLCI
CTRGCSCCTINFLPHS*NCCKFCACFTTEGRYNNRWNFTVFTETH*CYDVHI*FGY*QSSCNGLHYRWCCSDVFAVAN*HLWHCL*KTQTRP*LA*REV*GRCRVS*RRLGN*CYLNLCL*NCRWT
NCHLCKGN*GECSDIL*ACK*IFGFVC*LHYHMS*TSLEFR*NICHALKGIVQKVC*IQRNWP*THASKSPKRNVLRRGRNTSHRSVNRGSCLENN*FTTIRTTY**SS*SSIGWYTSLY*RAYV
ΔNRNRHAKVI CPT*VNGKQVI HTRRRTKKGFV**HGRSΔRI OFCFEYH**TKD**CT**FVI C I V S*TRVRSK*VRI CCGRCCHNFATST*TTVTG*FR*VEGVYI I T**VU*V*TG
```

A stop codon, also known as a separator for proteins, will appear as IKGLYLPR*QTNQLSISCRSVL*T in the output, which I will attach below. The amino acids are separated by a "*" in this output. Where each letter denotes a single amino acid, IKGLYLPR stands for the first protein, QTNQLSISCRSVL for the second protein, and so on. We also need to be aware that there are fewer sequences in proteins than in mRNA. This is because three mRNA molecules, known as amino acids, are used to create a single protein component, which can be demonstrated with the aid of a codon table, which we will create next. The asterisk (*) designates a stop codon; in these areas, the protein has achieved its maximum length and many of these stop codons are frequent, resulting in short proteins that have a very limited biological function, therefore we omit them from the remaining analyses.

```
from Bio.Data import CodonTable
print(CodonTable.unambiguous_rna_by_name['Standard'])
```

Table 1 Standard, SGC0

	U	C	A	G	
U	UUU F	UCU S	UAU Y	UGU C	U
U	UUC F	UCC S	UAC Y	UGC C	C
U	UUA L	UCA S	UAA Stop	UGA Stop	A
U	UUG L(s)	UCG S	UAG Stop	UGG W	G
C	CUU L	CCU P	CAU H	CGU R	U
C	CUC L	CCC P	CAC H	CGC R	C
C	CUA L	CCA P	CAA Q	CGA R	A
C	CUG L(s)	CCG P	CAG Q	CGG R	G
A	AUU I	ACU T	AAU N	AGU S	U
A	AUC I	ACC T	AAC N	AGC S	C
A	AUA I	ACA T	AAA K	AGA R	A
A	AUG M(s)	ACG T	AAG K	AGG R	G
G	GUU V	GCU A	GAU D	GGU G	U
G	GUC V	GCC A	GAC D	GGC G	C
G	GUA V	GCA A	GAA E	GGA G	A
G	GUG V	GCG A	GAG E	GGG G	G

Since the smallest functional protein that has been discovered so far is 20 amino acids in length, let's start by identifying all the proteins that are separated by an asterisk (*) and then subtract any sequences that are under that length.

```
Proteins = Amino_Acid.split('*') # * is translated stop codon
for i in Proteins[:]:
    if len(i) < 20:
        Proteins.remove(i)
df = pd.DataFrame(Proteins)
df.describe()
print('Total proteins:', len(df))
def conv(item):
    return len(item)
def to_str(item):
    return str(item)
df['sequence_str'] = df[0].apply(to_str)
df['length'] = df[0].apply(conv)
df.rename(columns={0: "sequence"}, inplace=True)
df.head()# Take only Longer than 20
functional_proteins = df.loc[df['length'] >= 20]
print('Total functional proteins:', len(functional_proteins))
functional_proteins.describe()
```

Total proteins: 80
Total functional proteins: 80

The Protein Analysis Using the ProtParam Module in Biopython is what we'll conduct next.

```
# 1. Protein Analysis With The Protparam Module In Biopython
from __future__ import division
poi_list = []
MW_list = []
from Bio.SeqUtils import ProtParam
for record in Proteins[:]:
    print("\n")
    X = ProtParam.ProteinAnalysis(str(record))
    POI = X.count_amino_acids()
    poi_list.append(POI)
    MW = X.molecular_weight()
    MW_list.append(MW)
    print("Protein of Interest = ", POI)
    print("Amino acids percent = ", str(X.get_amino_acids_percent()))
    print("Molecular weight = ", MW)
    print("Aromaticity = ", X.aromaticity())
    print("Flexibility = ", X.flexibility())
    print("Isoelectric point = ", X.isoelectric_point())
    print("Secondary structure fraction = ", X.secondary_structure_fraction())
```

```

Protein of Interest = {'A': 0, 'C': 1, 'D': 1, 'E': 1, 'F': 2, 'G': 1, 'H': 0, 'I': 3, 'K': 0, 'L': 3, 'M': 0, 'N': 1, 'P':
2, 'Q': 2, 'R': 4, 'S': 7, 'T': 4, 'V': 3, 'W': 0, 'Y': 0}
Amino acids percent = {'A': 0.0, 'C': 0.02857142857142857, 'D': 0.02857142857142857, 'E': 0.02857142857142857, 'F': 0.057142
85714285714, 'G': 0.02857142857142857, 'H': 0.0, 'I': 0.08571428571428572, 'K': 0.0, 'L': 0.08571428571428572, 'M': 0.0, 'N':
0.02857142857142857, 'P': 0.05714285714285714, 'Q': 0.05714285714285714, 'R': 0.11428571428571428, 'S': 0.2, 'T': 0.114285714
28571428, 'V': 0.08571428571428572, 'W': 0.0, 'Y': 0.0}
Molecular weight = 3896.388
Aromaticity = 0.05714285714285714
Flexibility = [1.0254166666666666, 1.0115238095238097, 0.9737499999999999, 0.9758214285714286, 0.9602857142857143, 0.9736428
571428574, 0.9377261904761905, 0.9531309523809526, 0.9757857142857143, 0.9605952380952383, 0.9992738095238096, 0.984464285714
2858, 0.9694166666666666, 0.9892738095238094, 1.004392857142857, 0.9920833333333334, 0.9642261904761905, 0.9855000000000002,
1.0142619047619048, 0.9869166666666667, 1.004809523809524, 0.991095238095238, 0.9842857142857142, 1.016547619047619, 1.006202
3809523808, 1.000690476190476]
Isoelectric point = 10.196142387390136
Secondary structure fraction = (0.3142857142857143, 0.3142857142857143, 0.11428571428571428)

```

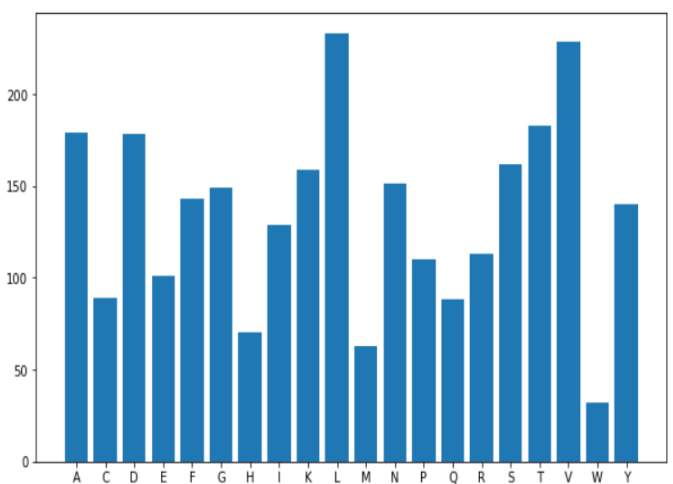
The protparam contains a variety of tools that we can use to learn more about our proteins. I've used count amino acids, which will only tally how many times a certain amino acid appears in a given protein sequence. Get amino acid per cent () will return the number representing the overall sequence's percentage of amino acids. Molecular weight () This gives back the individual protein's molecular weight. Secondary structure fraction will provide us with a list of the fraction of amino acids that tend to be in helix, turn, or sheet, and aromaticity () will compute and give us the aromaticity value of the protein according to Lobry and Guatier[3].

```

#plot POI
poi_list = poi_list[48]
plt.figure(figsize=(10,6));
plt.bar(poi_list.keys(), list(poi_list.values()), align='center')

```

<BarContainer object of 20 artists>



Helix's amino acids are V, I, Y, F, W, and L.

Turning amino acids: N, P, G, and S.

E, M, A, and L are the amino acids in Sheet[2].

We can therefore observe from our plot that this protein has a high number of lysines (L) and valines (V), which indicates the presence of a significant number of alpha-helices, and that the frequency of adenine (A) and thymine (T) is higher than that of guanine (G) and cytosine (C). Understanding this genetic information is the key to finding cures and vaccines so, Next question which should occupy our mind is how can we extract that information from this long sequence of letters this process is called gene expression. Gene expression is the process of extracting information from a gene and then using that in the synthesis of a functional gene product which is often proteins [1]. The first step of gene expression is the transcription and then comes the translation and then we study the amino acids for further analysis and gain more information from our gene sequence.

Conclusion: More accurately and individually than ever, disease prediction, diagnosis, and treatment are now achievable because of genomics. We are aware that the likelihood of successful treatment is increased by early disease diagnosis. Long before symptoms appear, genomics can detect disease. Changes in the DNA can lead to diseases like cancer as well, and this is where genomics can help because it can see these changes and look for them utilising developing technologies. We might be able to take simple steps to slow down or even stop the onset of disease if one's genome test results indicate susceptibility to any genomic disorder. Therefore, we can conclude that genetics is a key enabler in determining the specific healthcare steps a person should or should not take. The cancer genome atlas has identified the significant genetic alterations in over 30 different forms of cancer in the United States. Such a database can quickly provide the correct diagnosis and even suggest specific treatments based on the DNA of the patient and the ailment. The genetic sequencing

of cancer tumours aids in not only identifying specific cancers but also in understanding their causes and potential killers. this is also something which we all were able to witness during the pandemic only because of genomics we were able to check the mutations happening in the virus and that we were able to find the vaccination in such a short time.

REFERENCES

1. Gennady Evtugyn. "Chapter 3 Biosensor Signal Transducers" , Springer Science and Business Media LLC, 2014
2. Sujun Li, Aditi Dabir, Santosh A. Misal, Haixu Tang, Predrag Radivojac, James P. Reilly. "Impact of Amidination on Peptide Fragmentation and Identification in Shotgun Proteomics" , Journal of Proteome Research, 2016
3. homolog.us
4. www.coursehero.com
5. Karthikeyan, S., Jose, D.V. (2021). Role of Data Science in the Field of Genomics and Basic Analysis of Raw Genomic Data Using Python. In: Shukla, S., Unal, A., Kureethara, J.V., Mishra, D.K., Han, D.S. (eds) Data Science and Security. Lecture Notes in Networks and Systems, vol 290. Springer, Singapore. https://doi.org/10.1007/978-981-16-4486-3_19.
6. A Brief Guide to Genomics (2019) 15 August 2015. <https://www.genome.gov/about-genomics/fact-sheets/A-Brief-Guide-to-Genomics>.
7. Quilez Oliete J: A step-by-step guide to DNA sequencing data analysis, Kolabtree Blog, 23 March 2020. <https://www.kolabtree.com/blog/a-step-by-step-guide-to-dna-sequencing-data-analysis/>.
8. Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, Michiel J. L. de Hoon, Biopython: freely available Python tools for computational molecular biology and bioinformatics, *Bioinformatics*, Volume 25, Issue 11, 1 June 2009, Pages 1422–1423, <https://doi.org/10.1093/bioinformatics/btp163>.
9. John P A Ioannidis, Muin J Khoury, Evidence-based medicine and big genomic data, *Human Molecular Genetics*, Volume 27, Issue R1, 01 May 2018, Pages R2–R7, <https://doi.org/10.1093/hmg/ddy065>.
10. <https://www.kdnuggets.com/>.
11. <https://www.toptal.com/>.
12. <https://www.nationalgeographic.com/>.