Journal for Research in Applied Sciences and Biotechnology

www.jrasb.com

ISSN: 2583-4053

Volume-3 Issue-1 || February 2024 || PP. 139-146

https://doi.org/10.55544/jrasb.3.1.23

Computational Efficacy of Artificial Intelligence Model for in Silico Vaccine Development

Renuka Anil Jojare[1], Mahadev Asaram Jadhav[2] and Dipak Pandit Chavan[3]

[1]PG Students, Dept. of Biotechnology & Bioinformatics, Deogiri College, Chhatrapati Sambhajinagar - 431005, INDIA.

[2,3]Assistant Professor, Dept. of Biotechnology & Bioinformatics, Deogiri College, Chhatrapati Sambhajinagar - 431005, INDIA.

[1]Corresponding Author: renukajojare5@gmail.com

https://orcid.org/0009-0008-2077-0930

www.jrasb.com || Vol. 3 No. 1 (2024): February Issue

Received: 11-02-2024	**Revised:** 14-02-2024	**Accepted:** 16-02-2024

ABSTRACT

Bioinformatics is an interdisciplinary branch of science that develops methods and software tools for understanding biological data. Bioinformatics include both the power of biological concept and computational method to solve biological problem. It also bridged biological field with speed and accuracy of computer. Pre-design of vaccines by using artificial intelligence model for future upcoming viruses. Using AI throughout the vaccine development process to ensure that virus/pathogen vaccine met the needs of individuals without spending much time. A piece of genetic code that is capable of copying itself and typically has a detrimental effect on body, the pre-design vaccines will be available on one click no need for direct trials on humans. The model gives the predicted information about the upcoming risks for transmitting the disease in future generations by using artificial intelligence. The model is based on artificial intelligences and bioinformatics filed, all data will be presented and analyze simultaneously by the model and will efficiently build the vaccine molecule against the virus. The model provides highest accuracy and speed to sort out the vaccine.

Keywords- bioinformatics**,** artificial intelligence, pathogen, vaccine, python.

I. INTRODUCTION

The innovative exploration at the intersection of Bioinformatics and Artificial Intelligence, this project endeavors to harness the synergies between biological concepts and computational methodologies, facilitating a profound understanding of biological data. In anticipation of future viral threats, our approach involves deploying AI throughout the vaccine development process, ensuring rapid and accurate responses to emerging pathogens. The objective is to streamline the creation of vaccines, negating the need for extensive time-consuming process. Through predictive modeling, our system allows for the instantaneous availability of pre-designed vaccines molecules, offering a one-click solution to combat potential health risks. The project aims to revolutionize the field by leveraging the synergy between biological principles and computational approach. By incorporating artificial intelligence into the pre-design of vaccines solutions, we intend to expedite the designing process while ensuring accuracy and efficacy. Central to our methodology is the incorporation of artificial intelligence in assessing genetic codes, allowing for the identification of upcoming risks in disease transmission across future generations. This predictive model, rooted in the realms of artificial intelligence and bioinformatics, not only analyzes data in real-time but also efficiently constructs vaccine molecules tailored to counteract specific viruses.

ISSN: 2583-4053
Volume-3 Issue-1 || February 2024 || PP. 139-146
https://doi.org/10.55544/jrasb.3.1.23

With an unwavering commitment to precision and speed, our model stands as a beacon for the rapid development and deployment of effective vaccines and antiviral solutions.

Python:

Python is a interpreted and object-oriented also called as high-level programming language with dynamic semantics and it was developed by Guido van Rossum. It was originally released in 1991. Designed to be easy to code as well as fun, the name "Python" is nod to the British comedy group Monty Python. Python has a aspect of beginner-friendly language, replacing Java as the most widely used introductory language because it handles much of the complexity for the user, allowing beginners to focuses on fully grasping programming concepts rather than focusing on minute details.

Python is mostly known for using in server-side web development, software development, mathematics, as well as in system scripting, it is popular for Rapid Application Development and acts as a scripting or glue language to tie existing components because of the high-level, built-in data structures, dynamic typing, as well as for dynamic binding. Program maintenance costs are reduced with Python due to the easily learnable syntax and emphasis on readability. Additionally, Python can support the modules and packages facilitates modular programs and reuse of code. The Python is an open-source community language, which means numerous independent programmers are continually building libraries and functionality for it.

II. LANGUAGE FEATURES

Interpreted:

- No need to use the separate compilation like C and C++.
- We can directly run the program from the source code.
- Internally, Python converts the source code into its convenience language an intermediate form called bytecodes which is then translated into native language of specific computer to run it.
- No need to link and load with libraries, etc.
- Platform Independent:
- Python programs can be executed and run on multiple operating system platforms.
- Python can be used on Linux, Windows, Macintosh, Solaris and many more operating systems.

Free and Open Source:

- Python is open source hence anyone can download and run it.
- Python has the public license (GPL).

High-level Language:

- In Python, we don't have to worry about low-level details such as managing the memory used by the program.

Simple:

- Python is closer to English language; Easy to Learn
- More emphasis on the solution to the problem rather than the syntax.

Robust and Embeddable:

- It has Exceptional handling features
- It has Memory management techniques in built
- Python can be used within C language program to give scripting capabilities for the program's users.

Rich Library Support:

- Rich is a Python library for writing rich text (with colour and style) to the terminal, and for displaying advanced content such as tables, markdown, and syntax highlight code.

III. MATERIALS AND METHODS

Biopython: It is a set of freely available tools for biological computation written in Python by an international team of developer's. It is a distributed collaborative effort to develop Python libraries and applications which address to the needs of current and future work in bioinformatics. The source code is made available under the Biopython License, which is extremely liberal and compatible with almost every license in the whole world. Basically, the Biopython is a collection of python modules that provide functions to deal with DNA, RNA & protein sequence operations such as reverse complementing of a DNA string as well as finding motifs in protein sequences, etc.

The Biopython have lots of functionality including the capability to parse bioinformatics files into Python utilizable data structures, including support for the following formats:

- Blast output – both from standalone and WWW Blast.
- Clustalw (used for aligning multiple nucleotide or protein sequences in efficient manner.)
- FASTA (FASTA format is a text-based format for representing either nucleotide sequences or amino acid sequences)
- GenBank (annotation of all publicly available nucleotide sequences and their protein translations.)
- PubMed and Medline (Journal citation database)
- ExPASy files, like Enzyme and Prosite
- SCOP files(scope database files for sequences)
- UniGene (Each entry is a set of transcripts that appear to stem from the same transcription locus (i.e. gene or expressed pseudogene).)
- SwissProt (SWISS-PROT is a curated protein sequence database which strives to provide a high level of annotations (such as the description of the function of a protein, its domain structure, post-translational modifications, variants, etc) a

ISSN: 2583-4053
Volume-3 Issue-1 || February 2024 || PP. 139-146
https://doi.org/10.55544/jrasb.3.1.23

minimal level of redundancy and the higher level of integration with other databases.

Following module have used in python:

Matplotlib:

Matplotlib is a comprehensive library for creating static, animated, as well as interactive visualizations in Python. Matplotlib makes things easy and hard things possible. Matplotlib is a plotting library for the Python language and it have numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, and GTK. There is also a procedural "pylab" interface based on a state machine designed to closely resemble that of MATLAB, it use is discouraged. SciPy can make use of Matplotlib.

Matplotlib was originally written by John D. Hunter. Since then it has an active speedly developing community and is distributed under a BSD-style license. Matplotlib is a NumFOCUS fiscally sponsored project.

Matplotlib 2.0.x supports Python versions 2.7 through 3.10. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the latest version which supports Python 2.6. Matplotlib has pledged not to support Python 2 past 2020 but by signing the Python 3 versions Statement.

Features:

- Create publication quality plots.
- It makes interactive figures that can zoom, pan and update.
- Customize visual style and layout.
- Export to many file formats.
- Embed in JupyterLab and Graphical User Interfaces.
- Use a rich array of third-party packages built on Matplotlib.

Keras:

Keras means horn in Greek. It is a reference to a literary image from ancient Greek and Latin literature, first found in the Odyssey, where dream spirits) are divided between those who deceive dreamers with false visions, which arrive to Earth through a gate of ivory, and those who announce a future that will pass, which arrive through a gate of horn. However, Keras is highly-flexible framework suitable to iterate on state of the art research ideas. It follows the principle of progressive disclosure of complexity. It makes an easy to get started, yet it makes it possible to handle arbitrarily advanced use case, only need incremental Learning at each step. In much the same way that you were able to train & evaluate a simple neural network above in a few lines, you can use Keras to quickly develop new training procedures or exotic model architectures.

Keras features:

- It is Simple but not simplistic. Keras reduces developer cognitive load to free to focus on the parts of the problem that really matter.
- It is Flexible adopts the principle of progressive disclosure of complexity: simple workflows is quick and easy, while arbitrarily advanced workflows possible via a clear builds path.
- It is Powerful provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, Google.

Keras was initially developed as part of the research effort of project Based on Open-ended NeuroElectronic Intelligent Robot Operating System. Keras is an Open-Source Neural Network library written in Python language that runs on top of TensorFlow. It is designed to be modular, fast and easy to use. It was developed by François Chollet, a who is Google engineer. Keras doesn't handle low-level computation, keras uses another library to handle it, called the "Backend".

TensorFlow:

TensorFlow is an end-to-end open-source platform for design for machine learning. It has a comprehensive, flexible ecosystem tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers simply build and deploy ML powered applications. TensorFlow offers multiple levels of abstraction so you will choose the right one for your needs. Build and train models by using the high-level Keras API that makes getting started with TensorFlow and machine learning easy.

If require more flexibility, eager execution allows for immediate iteration and intuitive debugging. For large ML training tasks, we can use the Distribution Strategy API for distributed training on different hardware configurations without changing the model definition.

Keras vs TensorFlow:

TensorFlow is an end-to-end, open-source machine learning platform. we can think of it as an infrastructure layer for differentiable programming. It combines four key abilities:

- Efficiently executing lower-level tensor operations on CPU, GPU, or TPU.
- Computing the gradient of arbitrary differentiable expressions.
- Scaling computation to many devices, such as clusters of hundreds of GPUs.
- Exporting programs (graphs) to the external runtimes such as servers, browsers, mobile and embedded devices.

Keras is the high-level API of TensorFlow 2: It is approachable, highly-productive interface for solving the machine learning problems with a focus on modern deep learning. It provides the abstractions as well as building blocks for developing and shipping machine learning solutions with high iteration velocity.

Keras empowers engineers, researchers and students to take full advantage of the scalability and cross platform

Journal for Research in Applied Sciences and Biotechnology

www.jrasb.com

ISSN: 2583-4053

Volume-3 Issue-1 || February 2024 || PP. 139-146

https://doi.org/10.55544/jrasb.3.1.23

capabilities of TensorFlow 2: we can run Keras on TPU or on large clusters of GPUs, as well as we can export the Keras models to run in the browser or on a mobile device.

Sklearn:

It is the most useful & robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction as well as consistence interface in Python. It features various algorithms & support vector machine, random forests, and neighbors. keras supports Python numerical and scientific libraries like NumPy & SciPy.

Coalescent theory is the model to study how alleles sampled from a population may have originated from a common ancestor. In the simplest case, coalescent theory assumes no recombination, no natural selection, and no gene flow or population structure, meaning that each variant is equally likely to passed from one generation to the next generation. The model looks backward in the time, merging alleles into a single copy according to a random process in coalescence events. Under the model, the expected time between successive coalescence events increases almost exponentially back in the time, Variance is in the model.

Features*:*

Simple and efficient tools for predictive data analysis.

Accessible to everybody, and reusable in various contexts.

Built on NumPy, SciPy, and matplotlib

Role of Artificial Intelligence In Drug discovery And Vaccine

- Application of AI to drug discovery and vaccine Production is an exciting area that is rapidly developing and is primed to revolutionize analysis for target disease causing sequence.
- The potential that AI holds for understanding genome biology, and extending to all aspects of genomics research.
- AI utilizes the latest advances in biology and computing to develop state-of-the-art algorithms for drug & vaccine discovery, With the rapid increase in processing power and reduction in processing cost, the AI has a great potential to level the playing field in drug development and designing.
- AI obtained greater speed, and promising targets can be shortlisted without the need for extensive experimental input and manpower hours.

IV. PROCESS

Install TensorFlow and Keras using Anaconda Navigator: Keras and TensorFlow are open-source Python libraries for working with neural networks, creating machine learning models and performing deep learning. Because Keras is a high-level API for TensorFlow, they are installed together. We played around with pip install for multiple configurations and several hours, tried to figure how to properly set my python environment for TensorFlow and Keras. TensorFlow Requirements TensorFlow and Keras require Python 3.6+ (Python 3.8 requires TensorFlow 2.2+), and the latest version of pip.

Keras is the high-level API of TensorFlow: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing as well as for shipping machine learning solutions with high iteration velocity.

Keras empowers the user to take full advantage of the scalability and cross-platform. One key benefit of installing TensorFlow using conda rather than pip is a result of conda package management system. When TensorFlow is installed using conda, then conda installs all the necessary and compatible dependencies for the packages as well.

Following is the way we created environment for tensorflow and keras to combinely run:

Each virtual environment has its own Python binary (which matches the version of the binary that was used to create its environment) and will have its own independent set of installed Python packages in its site directories.

1. Launch Anaconda Navigator. Go to Environments tab and click ‘Create’.

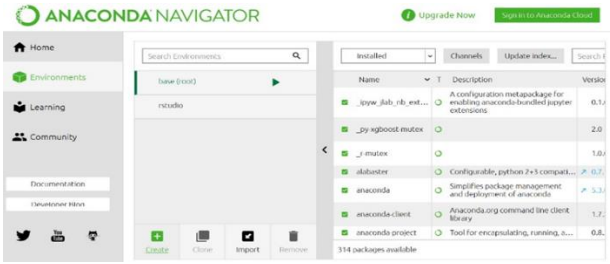

Figure 1: Creating the environment

Here we are creating the environment in anaconda navigator to run our packages successfully. this environment will provide platform to our module to run and build a model for analysis of viral genetic sequence.

2. Input new environment name, I put ‘tensorflow_env’. Make sure to select Python 3.6 Then click on ‘Create’, this may take few minutes.

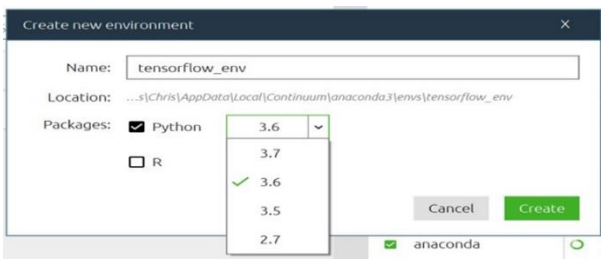

Figure 2: Selection of python version

ISSN: 2583-4053
Volume-3 Issue-1 || February 2024 || PP. 139-146
https://doi.org/10.55544/jrasb.3.1.23

3. At our new 'tensorflow_env' environment. Select 'Not installed', type in 'tensorflow'. Then, tick 'TensorFlow' and 'Apply'. The pop-up window have appear, going ahead and apply. This takes several minutes.

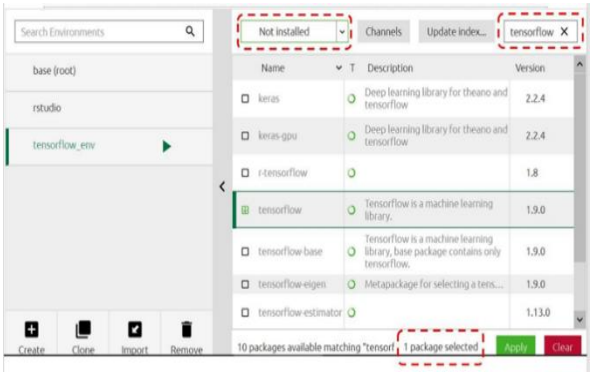

Figure 3: Package selection

In our new Environment 'tensorflow_env' Select 'Not installed', type in 'keras'. Then, tick 'keras' and 'Apply'. The pop-up window have appear, going ahead and apply. The same process we have done for this also take several minutes. Now we have imported both the packages in our anaconda environment.

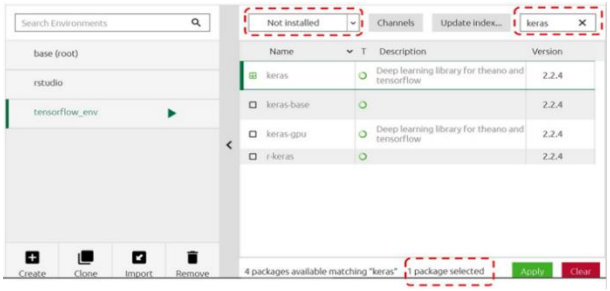

Figure 4: Keras Package Selection

For Checking our installation by importing the packages. If everything is set perfectly, the command will return nothing. If the installation was unsuccessful, we will get an error.

```
Python 3.6.8 |Anaconda, Inc.| (default, Feb 21 2019, 18:30:04) [MSC v.1
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> import keras
Using TensorFlow backend.
>>>
```

Figure 5: Commands to check installation

Here no error pop up, hence our environment is successfully created with the both modules Keras as well as with TensorFlow. Hence, we can work with both the modules.

Creating the model:

Here I have expanded the ideas about using NLP for expression of protein to demonstrate the superiority of LSTMs for analysis of sequencing data. Here I have implemented an LSTM model that reaches 99% accuracy of detecting sequence stretches from viral gene. model started with reading the sequences, chopping them into 200 nucleotides long sub sequences, each of them represents a sentence so we can further split each sentence into k-mers / words of the sentences by importing Biopython modules. These genetic sequences will convert into its expressing protein sequences.

The next step is to one-hot-encode the sentences via converting the k-mers / words into integers using the Tokenizer class in Python. Note that there is no need of padding in our case as all sentences are of the same length, but I include it here for generality.

The vocabulary size is 3910 in our case implying that not all possible 10-mers built out of 4 characters are present within the sequences. Finally, we define a Sequential model starting with the Keras Embedding Layer by implementing Keras and NumPy module that learns Word Embedding's while classifying the sequences, followed by a Bidirectional LSTM and a Dense layer.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 10)          3910
_________________________________________________________________
bidirectional_1 (Bidirection (None, 20)                1680
_________________________________________________________________
dense_1 (Dense)              (None, 10)                210
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 11
=================================================================
Total params: 5,811
Trainable params: 5,811
Non-trainable params: 0
_________________________________________________________________
```

Figure 6: vocabulary size output

The advantage of using Word Embedding's at the first layer is the dimension reduction from 3910 down to 10 dimensions only. This reduces overfitting and improves generalizability of the model by forcing similar words to share fitting parameters/weights, so we start training our model by importing matplotlib module.

```
Train on 6 samples, validate on 2 samples
Epoch 1/5
6/6 [==============================] - 7s 1s/step - loss: 0.6919 - accuracy: 0.8333 - val_loss: 0.6848 - val_accuracy: 1.0000
Epoch 2/5
6/6 [==============================] - 0s 81ms/step - loss: 0.6869 - accuracy: 1.0000 - val_loss: 0.6793 - val_accuracy: 1.0000
Epoch 3/5
6/6 [==============================] - 1s 110ms/step - loss: 0.6822 - accuracy: 1.0000 - val_loss: 0.6709 - val_accuracy: 1.000
0
Epoch 4/5
6/6 [==============================] - 1s 107ms/step - loss: 0.6739 - accuracy: 1.0000 - val_loss: 0.6628 - val_accuracy: 1.000
0
Epoch 5/5
6/6 [==============================] - 1s 101ms/step - loss: 0.6659 - accuracy: 1.0000 - val_loss: 0.6549 - val_accuracy: 1.000
0
```

Figure 7: training the sample

The advantage of using Word Embedding's at the first layer is the dimension reduction from 3910 down to 10 dimensions only. This reduces overfitting and improves generalizability of the model by forcing similar words to share fitting parameters/weights, so we start training our model by importing matplotlib module.

Journal for Research in Applied Sciences and Biotechnology

www.jrasb.com

ISSN: 2583-4053

Volume-3 Issue-1 || February 2024 || PP. 139-146

https://doi.org/10.55544/jrasb.3.1.23

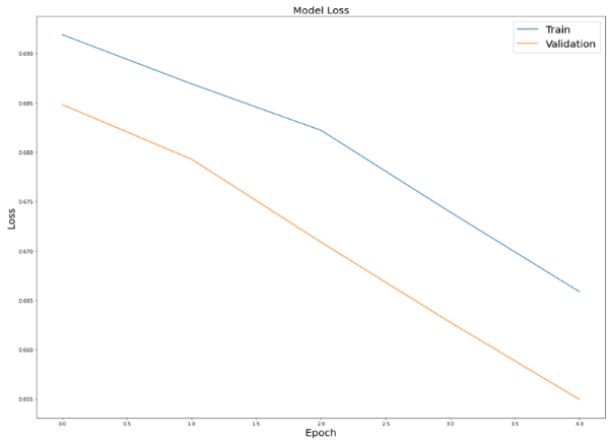

Figure 8: Train epochs and validation

Now we have a trained LSTM model that we will use later for predicting gene sequences from virus protein sequences. Now it is time for interpretation of the model which includes visualization of the vocabulary and detecting the virulent site / mutated sequence site which lead to infection and Disease by comparing with its non-mutated/non-virulent sequence predictive k-mers (protein sequences) that drive the classification.

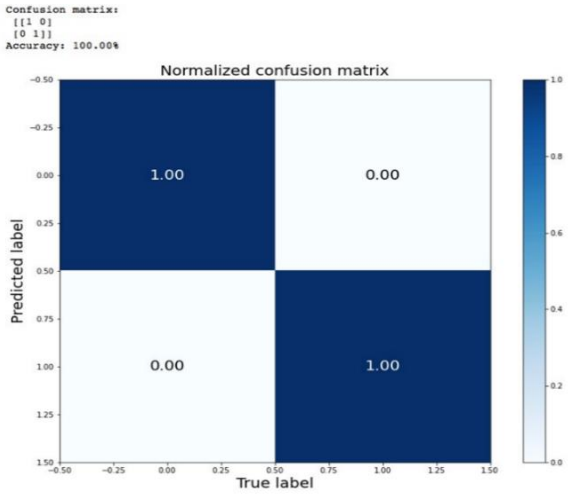

Figure 9: Normalization confusion matrix

One way to build feature importance for a neural network is to impose some perturbation on the input data and monitor the variation in the accuracy of prediction at the output of the network. Here, we would like to find the most informative k-mers, however in our case the input matrix X had dimensions, where the first dimension represented the number of training examples for the neural network, and the second dimension corresponded to the number of words in each sentence / sequence. This implies that an index or position of each word / k-mer was the feature for the input matrix X. Therefore, if we shuffle each of the 191 features and check the decrease in accuracy compared to the un-perturbed input matrix, we can rank the features by their importance for the final prediction. In our case, ranking features is equivalent to determining the most important positions of words / k-mers across all sentences or sequences. We observe some enrichment of important words at the beginning of sentences. Now we simply import the pandas module to check what are most common words / k-mers across all sentences at the positions, As by the comparing both input sequences this model reveals the mutant sequence (k-mers) which lead to disease and will recognize as disease causing sequence and the process of annotation will perform. Annotation is the process of deriving the structural and functional information of a protein or gene from a raw data set using different analysis, comparison, estimation, precision as well as other mining techniques. the motifs as shown in above output with highest accuracy for detecting common sub sequences. Genomics and proteomics represent a truly Big Data resource that delivers millions and billions of sequences that can be used as training examples / statistical observations for training the model.

As soon as model detects the disease-causing sequence after the process of comparing with the AI modules and by training the model according to data, The information of functional abnormality of disease-causing gene will attach along with the sequence so that our module will recognize it in future. Now the actual core work will start, to build the vaccine molecule along with the bond formation conditions that we provided to our model. Now we can see the vaccine molecule that is builds comparative to the disease-causing sequence. This vaccine molecule will bind to the mutant diseased sequence and will inhibit the activity which may lead to viral infection and disease. The model will also perform analysis and will show the basic molecules which are present in the sequences and will determine the composition to bind and isolate the toxic functional group of mutant sequence.

```
C is present ,'4HYDROGEN WILL bind to it'
hydrogen is present,'OH GROUP WILL PRESENT for condensation'
oxygen is present, '2nitrogen WILL PRESENT'
sulfer is preset, 'another SULFER WILL BIND TO FORM DI-SULFHIDE BOND'
nitrogen is present ,3hydrogen WILL PRESENT
```

Figure 10: Shows Conditions For building vaccine molecule

The model will state the functional group of virulent disease-causing/mutant sequence.

```
disease causing notation : CHOSNSOSCHS
disease inhibiting sequence for vaccine : CH3OHH2SNH3SH2SCH3OHS
```

Figure 11: Molecular annotation

The notation of toxic functional group will show's in the output in its molecular formula notation and according to it the molecule which can inhibits its toxicity after binding with toxic group will be appear as output. The molecular formula for vaccine molecule is shown in the output.

ISSN: 2583-4053
Volume-3 Issue-1 || February 2024 || PP. 139-146
https://doi.org/10.55544/jrasb.3.1.23

V. ADVANTAGES

Wide range of problems in biological sequence analysis have also benefited from the different problem-solving algorithms. This sequence analysis Model is useful for protein, RNA and gene comparison where large amount of sequence data is handled and compare with 99% data accuracy. The Model gives various advantages. AI utilizes the latest advances in biology and computing to develop state-of-the-art algorithms for drug & vaccine discovery, With the rapid increase in processing power and reduction in processing cost, AI has the great potential to level the playing field in drug development. This model is fast as well as highly accurate.it can handle or manage huge amount of data and compare it without any complexity or confusion. This model reduces paper work as well as manpower. It is totally automated model and manage all data with the high accuracy the result is determine. We can use this model to compare the gene sequence of disease gene with normal one and can state the mutated gene sequence which cause the disease with in few minutes. We can predict the repeated sequence of gene or proteins.

VI. LIMITATIONS

As such the project doesn't have any limitations but yes it does have certain

Shortcomings

- Deeply analysis information is not present.
- Its only work on python-based environment systems.
- Without module installation it may not work properly.

VII. DISCUSSION

Various Models are available online which compare the sequences after taking it as input but there is only 50-70% accuracy obtain in result but this model has 99% accuracy obtaining model in result.

Being offline there is no fear of internet connectivity once you imported this model. It has simple and understandable interface so, it could be used efficiently even by any new user. Just need to input the sequence file and run this model in your PC the analysis can predict the Mutant or virulent sequence segments present in the whole virus sequence which may led to disease. The model with high accuracy can build the vaccine molecule which can bind to toxic group of disease-causing sequence segment and will inhibit its toxicity. The user may not be an acknowledged of deep learning concept or may not be skilled person but using this model for analysis of sequences and building the vaccine molecule become less complex.

VIII. CONCLUSION

We showed that with the help of artificial intelligence we achieved impressive accuracy and outperformed models detecting disease causing regions of viruses and the vaccine molecule for the future approach for development of peptide vaccine. After comparing both the sequence we come to the conclusion that the mutation will be detected and the attachment of its expressed functional information is done. Although AI based methods have been developed relatively recently, they offer significant improvements in computational speed, especially for sequence study and analysis or comparing the gene sequence regions by training the model to handle the biological generated vast data with accuracy.

Interpretation of the model revealed k-mers to be the key of Vaccine designing and development. Our model can be used in biological sequence analysis In this representation is the sequence with particular mutation is occurring with toxic functional group (that is, that the sequences share a particular character which lead to infection and malfunction at a particular position) is coded as a single node with as many outgoing connections as there are possible characters of the alignment. In the terms of a model the result state that position of sequence which causes the disease or infection by the viruses as well as the vaccine molecule which will bind to it and inhibits its malfunction.

REFERENCES

[1] Thomas S, Abraham A, Baldwin J, Piplani S, Petrovsky N. Artificial Intelligence in Vaccine and Drug Design. Methods Mol Biol. 2022;2410:131-146. doi: 10.1007/978-1-0716-1884-4_6. PMID: 34914045

[2] McCaffrey P. Artificial Intelligence for Vaccine Design. Methods Mol Biol. 2022;2412:3-13. doi: 10.1007/978-1-0716-1892-9_1. PMID: 34918238..

[3] Pedregosa, F. et al. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011).

[4] Carrasco-Ramiro F, Peiró-Pastor R, Aguado B: Human genomics projects and precision medicine. Gene Ther. 2017; 24(9): 551–561.PubMed Abstract |Publisher Full Text

[5] Fourment M, Gillings MR: A comparison of common programming languages used in bioinformatics. BMC Bioinformatics. 2008; 9: 82. PubMed Abstract |Publisher Full Text.

[6] Tambonis T, Boareto M, Leite VBP: Differential Expression Analysis in RNA-seq Data Using a Geometric Approach. J Comput Biol. 2018; 25(11): 1257–1265.

[7] Bystrykh L: LeonidBystrykh/PY4GE: Python for gene expression (Version v0.0.1). Zenodo. 2021, June 30. Publisher Full Text

[8] Levy, S. E. & Myers, R. M. Advancements in next-generation sequencing. Annu. Rev. Genomics Hum. Genet. 17, 95–115 (2016).

ISSN: 2583-4053

Volume-3 Issue-1 || February 2024 || PP. 139-146

https://doi.org/10.55544/jrasb.3.1.23

[9] Subramanian, A. et al. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. Proc. Natl. Acad. Sci. 102, 15545–15550 (2005).

[10] Mathur, R., Rotroff, D., Ma, J., Shojaie, A. & Motsinger-Reif, A. Gene set analysis methods: A systematic comparison. BioData Min. 11, 8 (2018).

[11] Sun, L. et al. WebGIVI: A web-based gene enrichment analysis and visualization tool. BMC Bioinform. 18, 237 (2017).

[12] Raudvere, U. et al. g:Pr

[13] 3.ofiler: A web server for functional enrichment analysis and conversions of gene lists (2019 update). Nucleic Acids Res. 47, W191–W198 (2019).