How to Cite:
Subhashini, U., Bhargavi, P., & Jyothi, S. (2022). Fusion protein functionality prediction using genetic algorithm. *International Journal of Health Sciences*, *6*(S9), 4180–4193.
https://doi.org/10.53730/ijhs.v6nS9.13664

Fusion protein functionality prediction using genetic algorithm

U. Subhashini
Research Scholar, Department of Computer Science, Sri Padmavati Mahila Visvavidyalayam Tirupati-517 502
Email: subhau7@gmail.com

P. Bhargavi
Assistant professor, Department of Computer Science, Sri Padmavati Mahila Visvavidyalayam Tirupati-517 502
Email: pbhargavi18@yahoo.co.in

S. Jyothi
Assistant professor, Department of Computer Science, Sri Padmavati Mahila Visvavidyalayam Tirupati-517 502
Email: jyothi.spmvv@gmail.com

***Abstract*---**A fusion gene, which is composed by fusing pieces of two distinct genes, gives rise to a protein. The body may naturally produce fusion genes by transferring DNA between chromosomes. For instance, the BCR-ABL gene, which is a fusion gene that creates the BCR-ABL fusion protein, is found in a few different types of leukemia. By combining genes or segments of genes from related or unrelated animals, fusion genes and proteins can be produced further. Real-time lab tests, however, are expensive and time-consuming for automated fusion protein functionality prediction. In order to address this issue, this research suggests a brand-new Fusion Protein Functionality Prediction (FPFP) approach based on the Genetic Algorithm (GA) technique. The results of the experiments showed that the FPFP method accurately predicts the functionality of fusion proteins.

***Keywords*---**Protein, Fusion protein, Protein Functionality, Genetic Algorithm.

1. Introduction

Protein sequences are naturally referred to as strings of letters, just like in human language. For instance, there are 20 common amino acids in the protein alphabet

(AAs). Furthermore, just like in natural language, naturally occurring proteins are typically made up of previously utilised modular components that exhibit minute differences and can be rearranged and constructed hierarchically. According to this comparison, common protein motifs and domains—the basic operational building blocks of proteins—are comparable to words, phrases, and sentences in human language. Data completeness is yet another important characteristic that both proteins and human language share. A protein is more than just a list of amino acids; it is also a three-dimensional machine with a predetermined shape and function. However, all of these additional characteristics are preset by the amino-acid sequence. Even while protein shape and function are dynamic and context-specific (e.g., cellular environment, other molecules), the full amino-acid sequence nonetheless defines them. According to data theory, the sequence of the protein contains the information about the protein, such as its structure.

1.1 Protein function

The concept of protein function is not clearly defined and very context-sensitive. Typically, this idea serves as a catch-all phrase for all types of protein-related activities, whether they are physiological, molecular, or cellular in nature. One such division of the various jobs a protein might perform [1]:

1) Molecular function: A protein's ability to perform biochemical tasks such as ligand binding, catalyzing biochemical reactions, and undergoing conformational changes.
2) Cellular function: Many proteins work together to carry out intricate physiological processes like metabolic pathways and signal transmission to improve the efficiency of various organ systems.
3) Phenotypic function: The phenotypic characteristics and activities of the organism are determined by the interaction of physiological subsystems, the execution of various proteins' cellular tasks by various proteins, and the communication of this integrated system with external stimuli.

These three groups are hierarchical rather than autonomous. Moreover, this is not the only classification that has been suggested. For instance, the Gene Ontology Classification system divides protein function into three categories: biological process, molecular function, and cellular component [2].

2. Functional Classification Schemes

From the discussion above, it appears that protein function is a somewhat subjective idea, and different researchers may have different understandings of how proteins operate. As a result, the initial method of labeling is assigning natural language labels to proteins after their functions are known. Naturally, this is the case; yet, sometimes a naming practice results in names that are significantly different, such as Yippee and Starry Night [3]. Furthermore, it is obvious that humans cannot examine the labeling system due to the system's substantial complexity and wide variety. As a result, the requirement for a uniform functional labeling system is crucial, and several initiatives have addressed this need by putting up extremely creative ideas. It is worth listing some of the desirable characteristics of such projects [4], [5].

1) Widespread coverage: This is a highly important characteristic because any functional system should encompass as many functional phenomena across as many organisms as is practical.
2) Standardized format: Adopting a standard data structure and ensuring that functional labels vary as little as possible makes the system easily readable by computer programmes and significantly increases their impact.
3) Hierarchical structure: Instead of forming a flat list, the possible functions are organised hierarchically at the conceptual level. Functional classes allow a researcher to choose the appropriate level(s) for his examination because they span from very specific functions to extremely general functional categories.
4) Disjoint categories: Functions can take many different forms, including those of biological processes, cellular components, and molecules. As a result, each sort should have its own hierarchy, with no ties between them. Additionally, it enables the selection of the appropriate type of function to be researched.
5) Many functions: An operational strategy should provide the designation of a single protein with multiple functions in order to model the biological potential involved in numerous biological processes depending on the environment.
6) Dynamic nature: Lastly, the system should not be static; rather, it should be changed as new functional information is found.

As was already said, many functional systems have been put out to address these issues, each of which is successful to varying degrees and has a unique scope. The Enzyme Classification (EC) was the first systematic approach to be suggested in this area [6]. Using their chemical makeup, this method divides the class of enzymes, which are essential proteins responsible for catalysing metabolic reactions, into six groups. The subsequent division of these classes into three hierarchical steps represents the precise response of a particular enzyme. However, this system was limited in its application because it was primarily a classification of reactions rather than the characteristics of different catalyst enzymes.

The overlap between functional categorization systems is far higher than the overlap between structural classifications, as noted by [5], which supports this conclusion. The variability is, however, far greater in the former than in the latter. These investigations thus provide support for the aforementioned statements that, if properly executed, the assessment of a function prediction technique created in accordance with any of these systems will show strong outcomes. However, a constant attempt should be made to use the best available alternative. Any examination of functional classification methods today would be incomplete without a mention of Gene Ontology (GO) and all of its appealing features. The vast number of research that have used GO for various types of functional classes have shown that these qualities exist. In this article, we set out to give a thorough explanation of why GO is appropriate for the functional study of genes and proteins. In biology, since research is decentralized, the ability to efficiently organize knowledge is crucial, which results in the establishment of GOs [7]. GO is a functional classification system made up of three separate functional ontologies that, at the highest level, correspond to cellular component, molecular

function, and biological activity. Each focuses on a distinct aspect of how a protein works. Every one of these ontologies has a hierarchically organized structure and is modeled after a directed acyclic graph (DAG), where each node denotes a functional label.

Several GO term-based protein activity prediction approaches have been put out in the last ten years to automate the examination of protein sequences using machine learning and statistical analysis methodologies [8, 9, 10]. It may be claimed that there is still opportunity for major development in this field given the predicted effectiveness of present approaches. The goal of the Critical Assessment of Protein Function Annotation (CAFA) programme is to evaluate protein function prediction methods on a large scale. The results of the first two CAFA challenges showed that protein function prediction is still a difficult area to study [11], [12]. Artificial neural networks (ANNs) and other machine learning methods have been used to predict protein function [13]. A subset of ANNs is called Deep Neural Network (DNN) algorithms include several hidden layers. DNNs create progressively complex features at each succeeding layer by starting with low-level characteristics as input. In the past, DNN-based algorithms in computer vision and natural language processing have established themselves as industry standards [14]. The scientific community is now able to use DNN-based algorithms on a variety of research topics that include the processing of biomedical data thanks to recent improvements in realistic computational capability. In the fields of bioinformatics and cheminformatics, DNN algorithms have been shown to perform better than traditional predictive algorithms [15].

2.1 Fusion protein functionalities

Fusion proteins have important roles in the movement of nutrients, the catalysis of biochemical events, and the recognition and transmission of signals in living things. Any specific fusion protein's "function" refers to the wide range of characteristics that make up that protein's purpose. However, the term "fusion protein function" is not clearly defined; rather, it refers to complex phenomena with multiple, mutually reinforcing levels, including biochemical, cellular, organism-mediated, developmental, and physiological. These overlapping layers are connected in a variety of ways; for example, fusion protein kinases may be involved in a variety of cellular processes (such as the cell cycle) and chemical processes (transferase). A related kinase could further "misfunction," leading to illness. The broad, operational idea that "function is anything that occurs to or by a fusion protein" is what we are using in this instance.

3. Fusion Protein Functionality Prediction Based on Genetic Algorithm

The fusion protein function could be predicted based on homology detection using the fusion protein sequences. At first, a massive number of un-annotated fusion protein sequences are available in the massive volume of data. Fusion protein sequences are the compilation of amino acids in which they predict the function of a fusion protein by discovering the general residues with similar functions. Therefore, automated fusion protein function prediction is vital aimed at annotating uncharacterized fusion protein sequences, where precise prediction techniques are still necessary. This work proposed the Fusion Protein

Functionality Prediction (FPFP) algorithm based on Genetic Algorithm (GA). Furthermore, this algorithm predicts the functionality of any fusion protein automatically. Figure 1.1 illustrations the flow diagram of proposed FPFP algorithm.

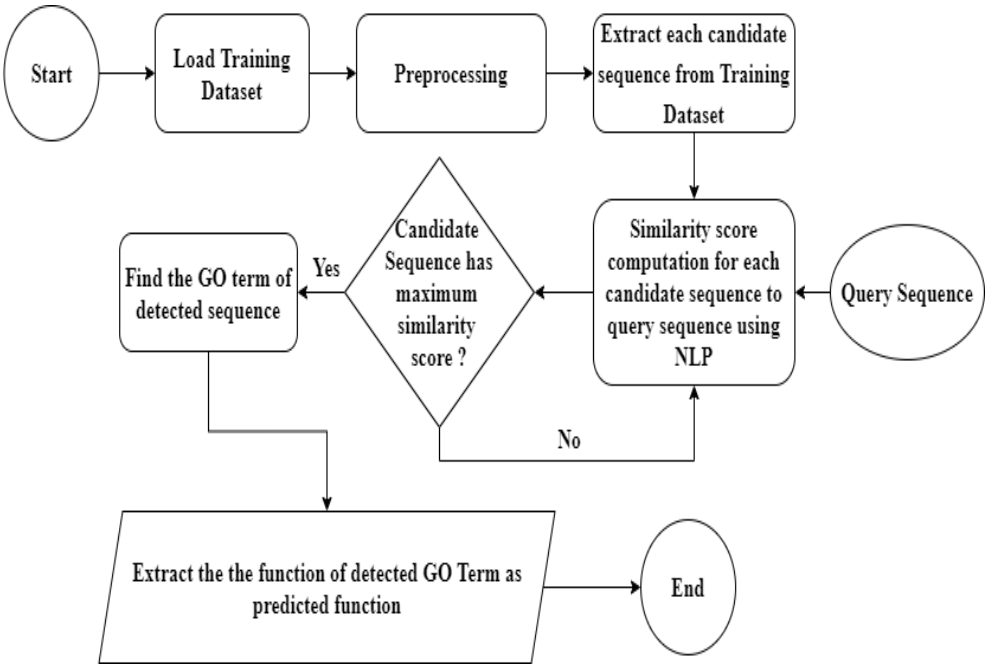

Figure 1.1: Proposed FPFP algorithm

Algorithm 1.1 shows the proposed FPFP algorithm based on the GA algorithm. Because fusion proteins sequences contain strings of amino-acid letters thus, the GA algorithm is a natural fit to predict the functions of fusion proteins. When a query protein is a feed to a prediction algorithm of FPFP, an individual fitness score is calculated for each GO term within that algorithm, representing the similarity of the query protein retains the function defined by the equivalent GO term. This algorithm first takes UniProtKB / SwissProt Training Dataset with each Gene Ontology (GO) category. Followed by, extracts each GO Term from the training dataset (Step 1). The Gene Ontology delivers a controlled vocabulary to classify the attributes of proteins built upon representative terms, referred as “GO terms”. The fusion protein function could be explained from numerous degrees, for example, physiological and phenotypical degrees. To get all different degrees features, the GO Consortium delivers three various functions such as biological process, cellular component and molecular function. The biological process obtains the functional definition of fusion protein function and allows denoting the gene processes in a cell. The cellular component explains the location of the structural component in which the gene activates. The molecular function explains the gene product involved in the cell. Each GO term signifies a unique functional attribute, and all terms are associated with each other in a Directed Acyclic Graph (DAG) structure based on inheritance relationships.

Furthermore, it extracts each GO Term's contents and function name (Step 4 - 5) and put all contents to a Population (P). Then it takes each Chromosome (C) from Population (P) and extracts each Gene (G) from Chromosome (C) and computes the cosine similarity score between each Genes to query sequence using Eq.(3.1) (Step 12).

$$\text{Similarity (P, Q)} = \frac{P \cdot Q}{\|P\| \times \|Q\|} = \frac{\sum_{i=1}^{n} P_i \times Q_i}{\sqrt{\sum_{i=1}^{n} P_i^2} \times \sqrt{\sum_{i=1}^{n} Q_i^2}} \qquad (3.1)$$

Followed by it computes the fitness score of each Chromosome (C) (Step 15) and find the Fitness Chromosome, which has maximum fitness score as predicted GO term (Step 32). Finally, this algorithm extracts the functionality of the predicted GO term as a predicted function in the GO hierarchy (Step 33). Figure 1.2 showing the hierarchy of sample GO terms.

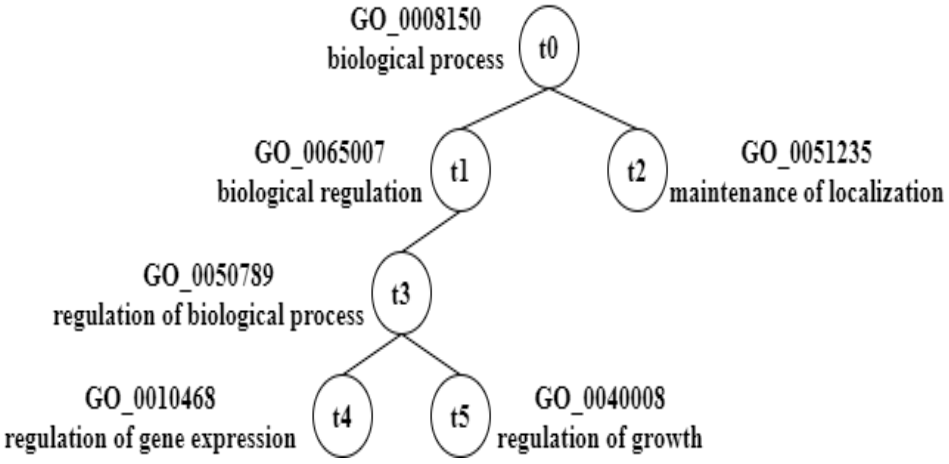

Figure 1.2: An example showing the hierarchy of sample GO terms.

Algorithm 1: Fusion Protein Functionality Prediction (FPFP) algorithm based on Genetic Algorithm

Input : UniProtKB / SwissProt Training Dataset (TD) with each Gene Ontology (GO) Category, Fusion Protein Query Sequence (FPQuerySeq)
Output : Predicted Function (PF), Predicted Score (PS) and Predicted Term (PT)
Step 1 : GT[] = Extract each GO Term from TD
Step 2 : P[] = "", FN[] = "", i = 0 //P-Population
Step 3 : For each GO Term G from GT
Step 4 : P[i] = Extract the contents of G //P[i] – ith chromosome in Population P
Step 5 : FN[i] = Extract the function name of G
Step 6 : i++
Step 7 : End For
Step 8 : SR[] = "", SI[] = "", Iteration = 0
Step 9 : For each Chromosome C from P
Step 10 : TS = 0
Step 11 : For each Gene G from C
Step 12 : score = CosineSimilarity(G, FPQuerySeq) // Eq.(3.1)
Step 13 : TS = TS + score
Step 14 : End For
Step 15 : FS = TS / NG //FS – Fitness Score and NG – Number of Genes in Chromosome C
Step 16 : SR[Iteration] = FS
Step 17 : R <-- Put FS and Iteration
Step 18 : SI[Iteration] = R
Step 19 : Iteration++
Step 20 : End For
Step 21 : Sort SR based on descending order for fittest
Step 22 : PS = "", i=0
Step 23 : maximumScore = SR[0]
Step 24 : For each Result from SI
Step 25 : Extract FS and Iteration from R
Step 26 : If maximumScore is equal to FS, then
Step 27 : i = Iteration
Step 28 : PS = FS
Step 29 : Break
Step 30 : End If
Step 31 : End For
Step 32 : PT = Extract the GO term at the ith position in GT[]
Step 33 : PF = Extract the name of the function at the ith position in FN[]
Step 34 : Return PF, PS and PT

Algorithm 1: Fusion Protein Functionality Prediction

Algorithm 2 explains similarity score computation between candidate sequences to query fusion protein sequences based on cosine similarity. This algorithm first extracts each amino acid from the Gene and fusion protein query sequence (Step 1 - 2). Then, it converts the Gene to numerical array vector-A and Fusion Protein Query Sequence to numerical array vector-B (Step 3 - 43). Furthermore, it computes similarity scores based on Eq. (3.1).

Algorithm 2: Similarity Score Computation based on Cosine Similarity

```
Input   : Gene(G), Fusion Protein Query Sequence (FPQuerySeq)
Output  : Similarity score (SS)
Step 1  : sp1[] = Extract amino acid from G
Step 2  : sp2[] = Extract amino acid from FPQuerySeq
Step 3  : NR[] = "", forstr1[] = "", forstr2[] = ""
Step 4  : i = 0
Step 5  : For each amino acid AA from sp1
Step 6  :         NR[i] = AA
Step 7  :         forstr1[i] = AA
Step 8  :         i++
Step 9  : End For
Step 10 : j=0
Step 11 : For each amino acid AA from sp2
Step 12 :         NR[i] = AA
Step 13 :         forstr2[j] = AA
Step 14 :         i++
Step 15 :         j++
Step 16 : End For
Step 17 : p[] = "", q[] = ""
Step 18 : For each amino acid AA from NR
Step 19 :         If forstr1 contains AA Then
Step 20 :                 p[i] = "1"
Step 21 :                 index = Get the index value Of AA in forstr1
Step 22 :                 Remove the index of forstr1
Step 23 :         Else
Step 24 :                 p[i] = "0"
Step 25 :         If forstr2 contains AA Then
Step 26 :                 q[i] = "1"
Step 27 :                 index = Get the index value Of AA in forstr2
Step 28 :         Remove the index of forstr2
Step 29 :         Else
Step 30 :                 q[i] = "0"
Step 31 :         End If
Step 32 :         i++
Step 33 : End For
Step 34 : vectorA[] = "", vectorB[] = "", i = 0
Step 35 : For each letter L from p
Step 36 :         vectorA[i] = L
Step 37 :         i++
Step 38 : End For
Step 39 : i = 0
Step 40 : For each letter L from q
Step 41 :         vectorB[i] = L
Step 42 :         i++
Step 43 : End For
Step 44 : dotProduct = 0, normA = 0, normB = 0
Step 45 : For (i = 0; i < vectorA.length; i++) Then
Step 46 :         dotProduct += vectorA[i] * vectorB[i]
Step 47 :         normA += Math.pow(vectorA[i], 2)
Step 48 :         normB += Math.pow(vectorB[i], 2)
Step 49 : End For
Step 50 : SS = dotProduct / (Math.sqrt(normA) * Math.sqrt(normB))
Step 51 : Return SS
```

Algorithm 2: Similarity Score Computation based on Cosine Similarity

4. Results & Discussions

The training dataset of the FPFP algorithm was created using the UniProtKB/SwissProt database protein entries. In this paper, we utilized annotations with manual curation or experimental evidence, which are extremely dependable. To create the training dataset, the equivalent annotations were extracting from the UniProt-GOA database, propagated to their parent terms according to the “true path rule”, which explains the inheritance relationship between GO terms. Using this dataset, a positive training dataset was created for each GO term. Briefly, proteins annotated either with the equivalent GO term or with one of its children terms were incorporated in the positive training dataset of the equivalent GO term. A set of structured vocabulary terms presented by the Gene Ontology training dataset explain operational data for a specific gene product. At present, ~40,000 GO terms exist. The Gene Ontology training dataset presents a helpful classification of functions, using a dictionary of definite terms separated into three main groups like molecular function, biological process, and cellular component. Figure 5.3 shows the number of terms for each category.

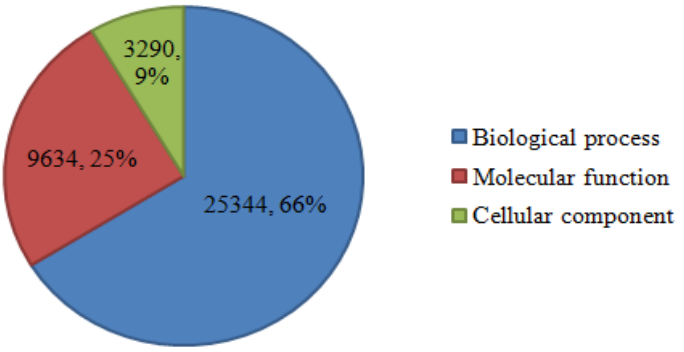

Figure 1.3: Number of terms for each category in the training dataset

Any user can query this dataset with a fusion protein sequence; the proposed FPFP algorithm retrieves related GO terms using GA. Over 4,00,000 species have been allocated to GO annotations in the training dataset. The distribution of annotations from the biological process, molecular function, and cellular component ontologies per species for proteins in UniProtKB is shown in Figures 1.4, 1.5, and 1.6, respectively. The ten species with the most annotations are shown for each ontology and annotations for all other species are shown in the 'rest' group.

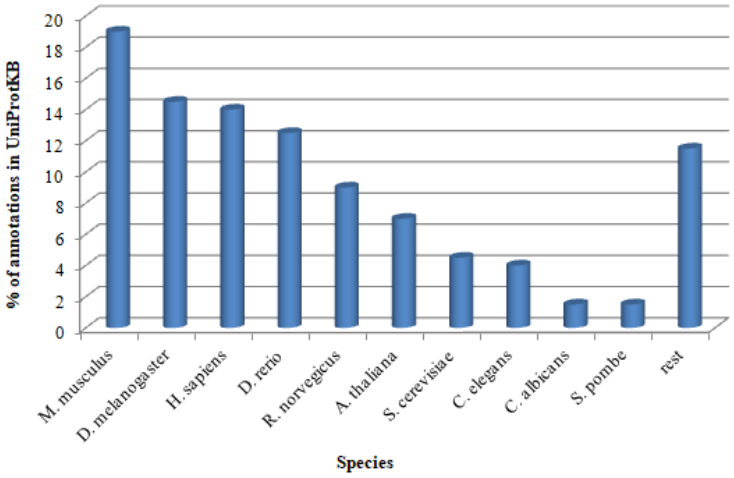

Figure 1.4: Distribution of annotations from the Biological Process ontologies per species for proteins in UniProtKB

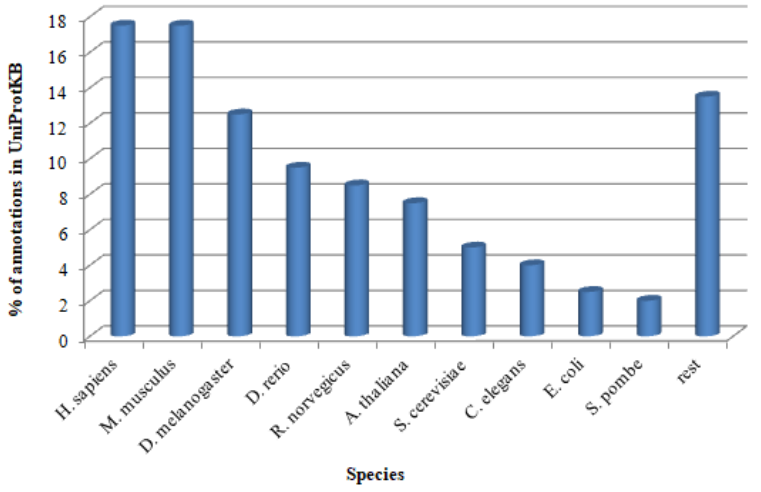

Figure 1.5: Distribution of annotations from the Molecular Function ontologies per species for proteins in UniProtKB

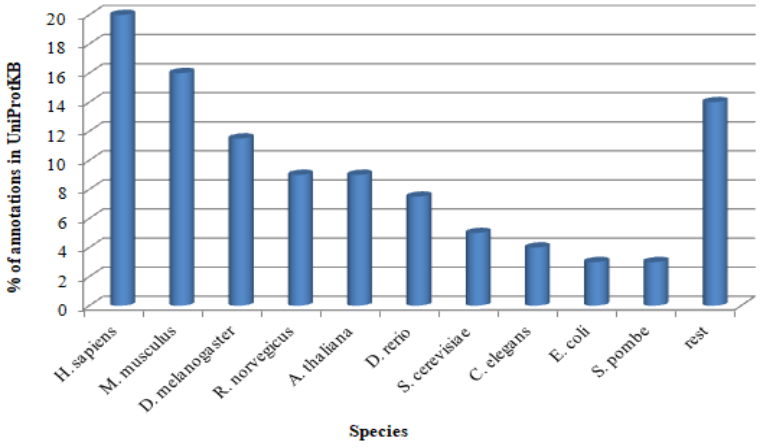

Figure 1.6: Distribution of annotations from the Cellular Component ontologies per species for proteins in UniProtKB

4.1 Case study 1

To predict the functionality of the following fusion protein,

Fusion Protein Sequence:
YEHDFHHIREWGNHWKNFLAVMGFFTALSTVMSLLTEVETPIRNEWGCRCNDSS

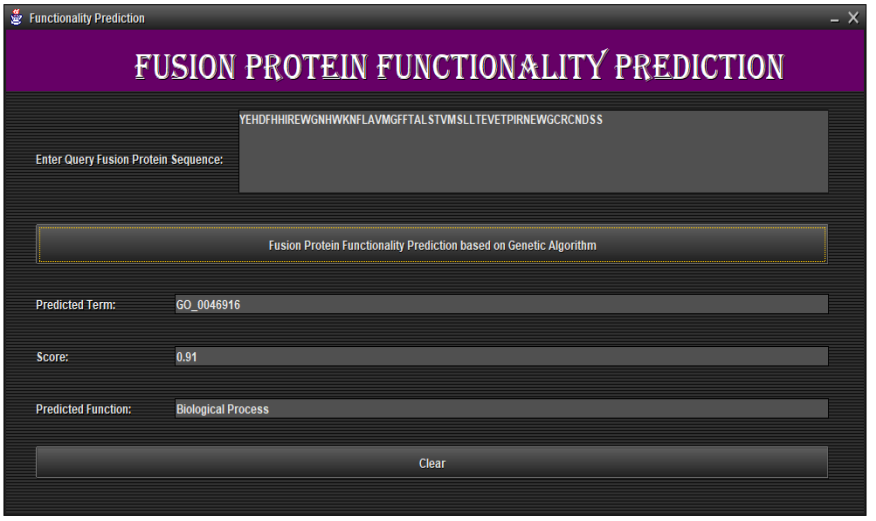

Figure1.1: Case study 1

Figure 1.7 shows the FPFP algorithm predicted GO Term is GO_0046916 for the above fusion protein sequence. In addition, its predicted score is 0.91, and also its predicted function is biological process.

4.2 Case Study 2

Fusion Protein Sequence:

To predict the functionality of fusion protein,
ADAAAGAQVFAANCAACHAGGNNAVMPTKTLKADALKTYLAGYKDGSKSLEEAVAYQ
VTNGQGAMPAFGGRLSDADIANVAAYIADQAENNKWIVVLNRAETPLPLDPTGKVKAE
LDTRMLYLVRMTVNLPRNLDPREEERLKASEKARSRTLQEQGQWRYLWRTTGKYGNI
SVFDVNSHDELHEILWSLPFFPYLTIDVEPLSHHPARVGKD

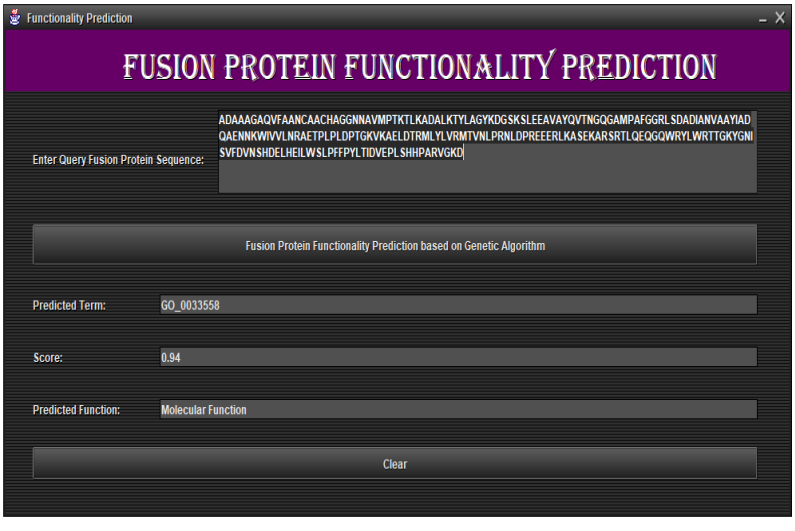

Figure1.2: Case study 2

Figure 1.8 shows the FPFP algorithm predicted GO Term is GO_0033558 for the above fusion protein sequence. In addition, its predicted score is 0.94 and also its predicted function is a molecular function.

4.3 Case Study 3

Fusion Protein Sequence:
To predict the functionality of fusion protein,
GSHMNTEVSSEIYQWVRDELKRAGISQAVFARVAFNRTQGLLSEILRKEEDPKTASQS
LLVNLRAMQNFLQLPEAERDRIYQDERERSLRKRKGVTPSTTALPDIVNLSTNYLDKNT
REDRIHSIKDFSNADEVENLYTQVADNEYLVQGRMLIDEFNEVFETDLHMSDVDTMA
GYLITALGTIPDEGEKPSFEVGNIKLTAEEMEGTRLLVLRVHFYDEE

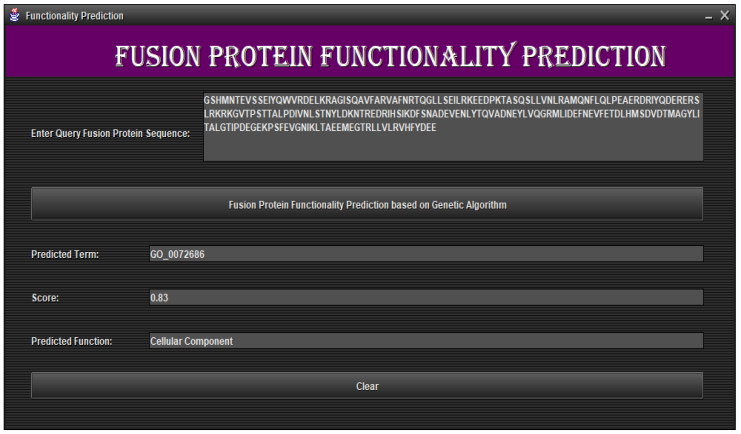

Figure1.3: Case study 3

Figure 1.9 shows the FPFP algorithm predicted GO Term is GO_0072686 for the above fusion protein sequence. In addition, it's predicted score is 0.83, and also its predicted function is a cellular component. Furthermore, function prediction time comparisons of the above case studies are showed in Figure 1.10.

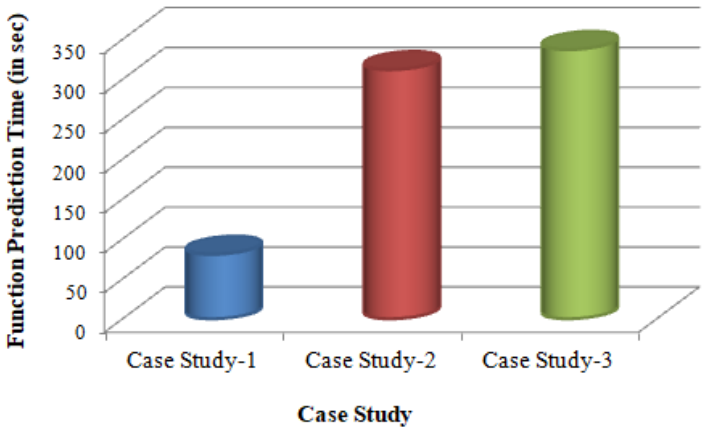

Figure1.4: Function Prediction Time comparison

From Figure 1.10, we know the smallest fusion protein sequence takes less time for function prediction, and the largest fusion protein sequence takes more time.

5. Conclusion

Fusion proteins could be created by merging proteins or parts of proteins from similar or dissimilar organisms. However, real-time lab experiments for automated fusion protein functionality prediction are luxurious and take more time. This paper proposed a novel Fusion Protein Functionality Prediction (FPFP) algorithm based on Genetic Algorithm to deal with this problem. This algorithm predicts the function of fusion protein efficiently. It presents the cellular component, biological process and molecular function of an unannotated fusion protein by the GO consortium. The experimental results showed the proposed FPFP algorithm predicts the function of fusion protein efficiently.

References

1. Cao, R., Freitas, C., Chan, L., Sun, M., Jiang, H., & Chen, Z. (2017). ProLanGO: protein function prediction using neural machine translation based on a recurrent neural network. Molecules, 22(10), 1732.
2. de Souza Vandenberghe, L. P., Karp, S. G., Pagnoncelli, M. G. B., von Linsingen Tavares, M., Junior, N. L., Diestra, K. V., ... & Soccol, C. R. (2020). Classification of enzymes and catalytic properties. In Biomass, Biofuels, Biochemicals (pp. 11-30). Elsevier.
3. Gene Ontology Consortium. (2019). The gene ontology resource: 20 years and still GOing strong. Nucleic acids research, 47(D1), D330-D338.
4. Gligorijević, V., Barot, M., & Bonneau, R. (2018). deepNF: deep network fusion for protein function prediction. Bioinformatics, 34(22), 3873-3881.
5. Gligorijević, V., Barot, M., & Bonneau, R. (2018). deepNF: deep network fusion for protein function prediction. Bioinformatics, 34(22), 3873-3881.
6. Irby, S. M., Pelaez, N. J., & Anderson, T. R. (2018). Anticipated learning outcomes for a biochemistry course-based undergraduate research experience to predict protein function from structure: Implications for assessment design. Biochemistry and Molecular Biology Education, 46(5), 478-492.
7. Jain, A., & Kihara, D. (2019). NNTox: gene ontology-based protein toxicity prediction using neural network. Scientific reports, 9(1), 1-10.
8. Lan, N., Jansen, R., & Gerstein, M. (2002). Toward a systematic definition of protein function that scales to the genome level: Defining a function in terms of interactions. Proceedings of the IEEE, 90(12), 1848-1858.
9. Liu, X. (2017). Deep recurrent neural network for protein function prediction from the sequence. arXiv preprint arXiv:1701.08318.
10. Ouzounis, C. A., Coulson, R. M., Enright, A. J., Kunin, V., & Pereira-Leal, J. B. (2003). Classification schemes for protein structure and function. Nature Reviews Genetics, 4(7), 508-519.
11. Rison, S. C., Hodgman, T. C., & Thornton, J. M. (2000). Comparison of functional annotation schemes for genomes. Functional & integrative genomics, 1(1), 56-69.

12. Seyyedsalehi, S. F., Soleymani, M., Rabiee, H. R., & Mofrad, M. R. (2021). PFP-WGAN: Protein function prediction by discovering Gene Ontology term correlations with generative adversarial networks. Plos one, 16(2), e0244430.
13. Wan, C., & Jones, D. T. (2020). Protein function prediction is improved by creating synthetic feature samples with generative adversarial networks. Nature Machine Intelligence, 2(9), 540-550.
14. Zhao, Z., Zhang, H., Hu, M., Yang, N., Wang, H., Wang, C., ... & Gu, L. (2021). Protein Function Prediction with Deep Neural Learning.
15. Zhou, N., Jiang, Y., Bergquist, T. R., Lee, A. J., Kacsoh, B. Z., Crocker, A. W., ... & Kihara, D. (2019). The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. Genome Biology, 20(1), 1-23.