Original Article

A Hybrid Genetic Particle Swarm Optimization Algorithm Based Fusion Protein Functionality Prediction

U. Subhashini[1], P. Bhargavi[2], S. Jyothi[3]
[1]Research Scholar, Department of Computer Science
[2]Assistant professor, Department of Computer Science
[3]Professor, Department of Computer Science
Email: pbhargavi18@yahoo.co.in
DOI: 10.47750/pnr.2022.13.S06.149

Abstract

A fusion protein is a protein with at least two domains that are each encoded by a different gene and are combined into a single polypeptide by transcription and translation. For example, chromosomal rearrangement could result in the in vivo production of fusion proteins. One such fusion protein is the one responsible for chronic myelogenous leukaemia, the BCR-ABL protein. Recombinant DNA techniques could be used to create fusion proteins in vitro. By combining genes or portion of genes from similar or dissimilar organisms, fusion genes and proteins may be produced. But, real-time lab experiments for automated fusion protein functionality prediction are expensive and time-consuming. This paper proposes a novel Fusion Protein Functionality Prediction based on a Hybrid Genetic Particle Swarm Optimization (HybGPSO) algorithm to deal with this problem. The cellular component, biological process, and molecular function of an unannotated fusion protein by the GO consortium are the three functionalities predicted by this algorithm. The results of the experiments demonstrate that the proposed HybGPSO algorithm accurately predicts the function of fusion proteins.

1. INTRODUCTION

Fusion or chimeric proteins are proteins formed by combining two or more genes that initially coded for separate proteins [1]. By combining several proteins with the β-galactosidase enzyme in Escherichia coli, a few primary fusion proteins were created [2]. These fusions were initially used to calculate the protein of interest's expression level. Initially, only proteins from genes near the β-galactosidase gene were included in fusions. However, Malcolm Casadaban and colleagues later invented in vivo and in vitro methods that made it possible to fuse practically any protein.

Researchers were first shocked to find that when a protein's carboxy-terminus was fused to the amino terminus of β-galactosidase, both proteins retained activity; some of the fusions were functional [3]. Researchers started to create fusions to other proteins moreover β-galactosidase were more and more fusions to that protein were created and discovered to have activity. They discovered that fusing two domains frequently resulted in a fusion protein that kept the activity of both domains.

Other fusion partners have been added using the same method for making fusion proteins, and the fusion partner has been given new applications. The following are three of the most significant applications of fusion proteins: as tools for cloned gene purification, as reporters of expression level and histochemical tags to permit visualization of protein localization in a cell, tissue, or organism [4].

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. One such classification has various functions a protein might perform is provided by [5]:

1) Molecular function: The biochemical functions that a protein performs like ligand binding, catalyzing biochemical reactions and conformational changes.

2) Cellular function: To execute intricate physiological processes, including metabolic pathways and signal transmission and improve the functionality of various organ elements, several proteins interact together.

3) Phenotypic function: The phenotypic characteristics and activities of the organism are determined by the interaction of physiological subsystems, different proteins, and their connection with environmental stimuli.

These three groups are hierarchical rather than autonomous. Moreover, this is not only classification that is proposed. Protein function, for instance, is categorized by the Gene Ontology Classification system as a cellular component, molecular function, and biological process. [6].

1.1 Schemes for Functional Classification

From the discussion above, protein function appears somewhat subjective, and different researchers may have different understandings of how proteins operate. As a result, the initial method for labelling proteins is to assign natural language labels once their function has been established. Naturally, this is the case, yet, sometimes a naming practice results in significantly different names, such as Yippee and Starry Night [7]. Furthermore, humans cannot examine the labelling system due to the system's substantial complexity and a wide variety. As a result, the requirement for a uniform functional labelling system is crucial, and several initiatives have addressed this need by putting up extremely creative ideas. It is worthwhile to outline some of the ideal qualities of such projects [8], [9].

1) Extensive coverage: This is a significant property because any functional system must contain many functional events in many possible organisms.

2) Standardized format: Adopting minimal variation and static data structure in functional labels makes the system comprehensible by computer programs and significantly improves their impact.

3) Hierarchical structure: Possible functions do not form a flat list; instead, they are hierarchically organized at a theoretical level. Function classes ranges from precise function to the most general type of function.

4) Disjoint categories: Functions can be various types, for example, cellular components, molecular function and biological processes. Therefore, a distinct hierarchy would be created for each type with no correlation. Also, it allows the selection of suitable types of activity are explored.

5) Multiple functions: To model biological potential involved in multiple biological processes dependent on the environment, the functional structure should permit the designation of a protein with multiple functions.

6) Dynamic nature: Last, the system should not be static; however, it must be adapted when new functional awareness is discovered.

As stated, many operating systems have been suggested to deal with these problems, each by some degree of success and with different objectives. The first systematic system proposed in this region was Enzyme Classification (EC) [10]. By using their chemical components, this method classifies the class of proteins known as enzymes, which serve as the catalysts for metabolic reactions, into six types. The subsequent division of these classes into three hierarchical tiers, each representing a specific enzyme's precise reaction, ensues. However, this system was only useful to a certain extent because it classified reactions rather than different catalytic enzymes' features. This result is echoed by [9], who points out that the functional classification between systems is greater than that between the structural classifications. However, the variation is much higher than the former. Therefore, these studies justify the above assumptions that the evaluation of a function forecasting system is based on one of these systems, which, if implemented correctly, will yield strong results. But an effort should be made to use best available options. Nowadays, any criticism of functional classification techniques would be imperfect without discussing Gene Ontology (GO) and its voluminous attractive properties.

These properties are demonstrated by numerous studies that have applied GO to a wide variety of functional classes. Here we would like to present a detailed discussion of why GO is suitable for functional analysis of genes and proteins. Recognizing the capability to organize knowledge effectively is essential to biology, where investigation is disseminated, leading to the formation of GO [11]. GO is a functional classification scheme with three distinct functional ontologies equivalent

to cellular component, molecular function, and biological process. Each deals with a different aspect of a protein's function. Each of these ontologies is hierarchically structured and modelled like a Directed Acyclic Graph (DAG), in which each node resembles to a function label.

Over the past decade, several GO term-based protein activity prediction methods has been proposed to automate protein sequencing using machine learning and statistical analysis methods [12], [13], [14]. Considering the forecast performance of the current methods, it can be said that there is still room for significant improvement in this area. The Critical Assessment of Protein Activity (CAFA) is an initiative that is a large-scale evaluation of protein activity predictive techniques, and the results of the first two CAFA challenges proved that protein activity prognosis is still a challenging area [15] [16]. Numerous machine learning algorithms have been used to predict protein function, such as Artificial Neural Networks (ANNs) [17]. Deep Neural Network (DNN) algorithms a subset of ANNs consist of several hidden layers. DNNs input fewer features and create more advanced features in each successive layer.

DNN based algorithms have previously become industry standards in computer vision and natural language processing [18]. Current developments in computational power have allowed the scientific community to use DNN based algorithms in many research areas, together with biomedical data analysis. DNN algorithms have been demonstrated to outperform conventional predictive algorithms in bioinformatics and cheminformatics [19].

1.2 Fusion Protein Functionalities:

Fusion proteins perform significant functions in organisms like transporting nutrients, catalyzing biochemical reactions and recognizing and transmitting signals. The majority of the features of the role of any particular fusion protein are called its 'function'. However, fusion protein function is not a well-defined term; alternatively, the function is a multifaceted phenomenon associated with many equally interrelated states: biochemical, cellular, bio-mediated, developmental, and physiological. These interconnected levels are intertwined in various ways; For example, fusion protein kinases may be associated with various cellular functions (for example, the cell cycle) and a chemical function (transferase). Similar kinase can also be 'misused', leading to disease. Here we use the general, functional concept that 'function is a fusion protein or all that occurs through it'.

But, real-time lab experiments for automated fusion protein functionality prediction are expensive and time-consuming. This paper proposes a novel Fusion Protein Functionality Prediction based on a Hybrid Genetic Particle Swarm Optimization (HybGPSO) algorithm to deal with this problem.

2. Fusion Protein Functionality Prediction Based on a Hybrid Genetic Particle Swarm Optimization (HybGPSO) Algorithm:

The fusion protein function could be predicted based on homology detection using the fusion protein sequences. At first, a massive number of unannotated 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 for annotating uncharacterized fusion protein sequences, where precise prediction techniques are still necessary. This work proposed the Fusion Protein Functionality Prediction technique based on a Hybrid Genetic Particle Swarm Optimization (HybGPSO) Algorithm. Furthermore, this algorithm predicts the functionality of any fusion protein automatically. Figure 1 shows the flow diagram of the proposed HybGPSO algorithm.

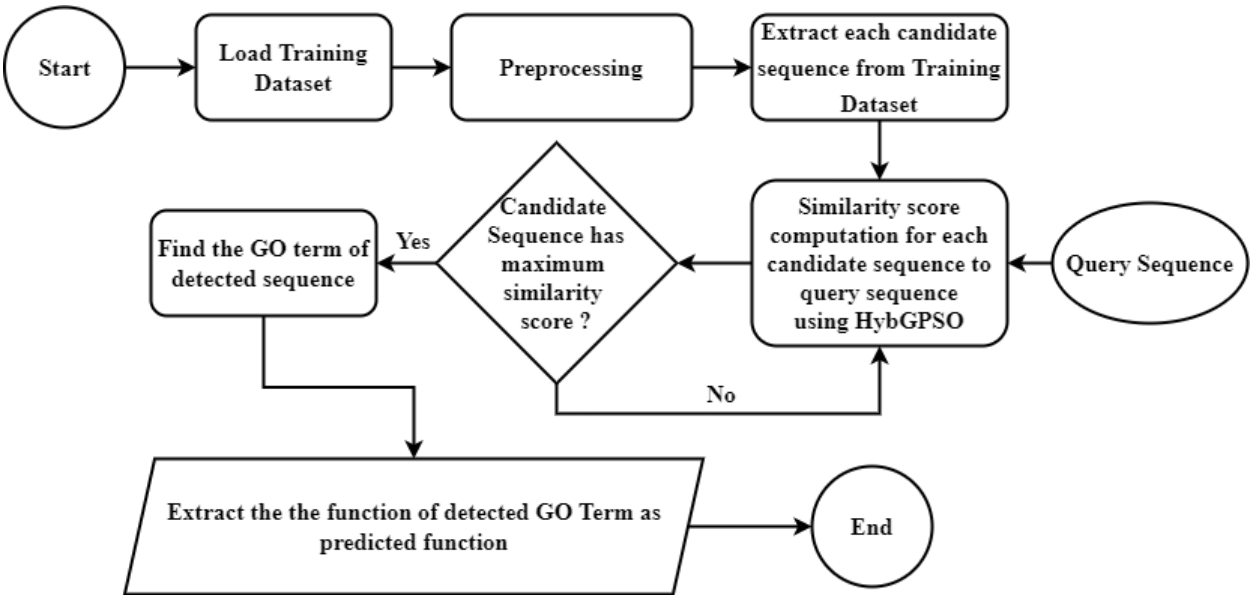

Figure 1: Proposed HybGPSO Algorithm

Algorithm 1 shows the proposed fusion protein functionality prediction based on the HybGPSO algorithm. Because fusion protein sequences contain strings of amino-acid letters thus, the HybGPSO algorithm is a natural fit to predict the functions of fusion proteins. When a query protein is fed to a prediction algorithm of HybGPSO, an individual fitness score is calculated for each GO term within that algorithm, representing the similarity of the query protein possessing the function defined by the equivalent GO term. This algorithm first takes UniProtKB / SwissProt Training Dataset with each Gene Ontology (GO) category. Followed by it predicting the functionality of fusion protein using genetic algorithm in Algorithm 2 and also predicting the functionality of fusion protein using particle swarm optimization algorithm in Algorithm 4. Algorithm 1 then compares the predicted score of the two algorithms and chooses the function of the algorithm with the higher predicted score as the final function.

Algorithm 1: Fusion Protein Functionality Prediction (FPFP) based on a Hybrid Genetic Particle Swarm Optimization (HybGPSO) 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	**:**	R1[] <-- FPFP_GA(TD,GO,FPQuerySeq) // **Algorithm 2**
Step 2	**:**	R2[] <-- FPFP_PSO(TD,GO,FPQuerySeq) // **Algorithm 4**
Step 3	**:**	PF1 = R1[0], PS1 = R1[1], PT1 = R1[2]
Step 4	**:**	PF2 = R2[0], PS2 = R2[1], PT2 = R2[2]
Step 5	**:**	If(PS1 >= PS2) Then
Step 6	**:**	PF <-- PF1
Step 7	**:**	PS <-- PS1
Step 8	**:**	PT <-- PT1
Step 9	**:**	Else
Step 10	**:**	PF <-- PF2
Step 11	**:**	PS <-- PS2
Step 12	**:**	PT <-- PT2
Step 13	**:**	End If

Algorithm 2 shows the proposed fusion protein functionality prediction based on the GA algorithm. Because fusion protein 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 GA, an individual fitness score is calculated for each GO term within that algorithm, representing the similarity of the query protein possessing the function defined by the equivalent GO term. This algorithm first takes UniProtKB / SwissProt Training Dataset with each Gene Ontology (GO) category. Followed by extracting each GO Term from the training dataset (Step 1). The Gene Ontology provides a controlled vocabulary to classify the attributes

of proteins based upon representative terms, referred to 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 represents a unique functional attribute, and all terms are associated 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 puts all contents into 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 gene to query sequence using Eq. (1) (Step 12).

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

Followed by it computes the fitness score of each Chromosome (C) (Step 15) and finds the Fittest Chromosome, which has the 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 2 shows the hierarchy of sample GO terms.

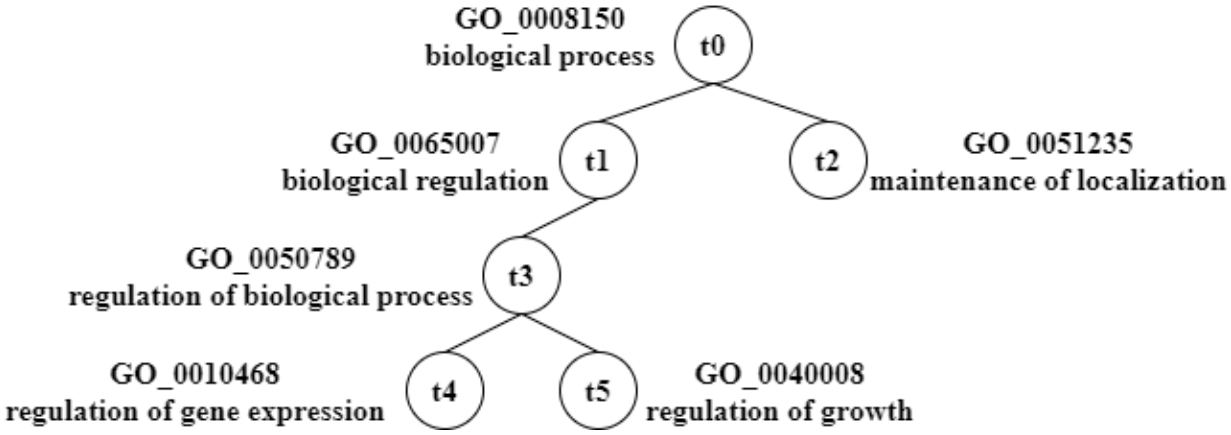

Figure 2: An example showing the hierarchy of sample GO terms.Algorithm 2: Fusion Protein Functionality Prediction based on Genetic Algorithm (FPFP_GA)

Input	**:**	UniProtKB / SwissProt Training Dataset (TD) with each Gene Ontology (GO) Category, Fusion Protein Query Sequence (FPQuerySeq)
Output	**:**	R1[]
Step 1	**:**	GT[] = Extract each GO_Term from TD
Step 2	**:**	R1[] = "", 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] - i^{th} 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 = Cos_Dis(G, FPQuerySeq) **// Eq. (1) and Algorithm 3**
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 chromosome selection
Step 22	**:**	PS1 = "", i=0
Step 23	**:**	maximumScore = SR[0]
Step 24	**:**	For each Result R 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	**:**	PS1 = FS
Step 29	**:**	break
Step 30	**:**	End If
Step 31	**:**	End For
Step 32	**:**	PT1 = Extract the GO_Term at the ith position in GT[] **// Fittest Chromosome**
Step 33	**:**	PF1 = Extract the name of the function at the ith position in FN[]
Step 34	**:**	R1[0] = PF1, R1[1] = PS1, R1[2] = PT1
Step 35	**:**	Return R1[]

Algorithm 3 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 vectorA and Fusion Protein Query Sequence to numerical array vectorB (Step 3 - 43). Furthermore, it computes similarity scores based on Eq. (1).

Algorithm 3: Similarity Score Computation based on Cosine Distance (Cos_Dis)

```
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 4 shows the proposed fusion protein functionality prediction based on the PSO algorithm. When a query protein is a feed to a prediction algorithm of PSO, an individual fitness score is calculated for each GO term within that algorithm,

representing the similarity of the query protein possessing the function defined by the equivalent GO term. This algorithm first takes UniProtKB / SwissProt Training Dataset with each Gene Ontology (GO) category. Followed by extracting each GO Term from the training dataset (Step 1). The Gene Ontology provides a controlled vocabulary to classify the attributes of proteins based upon representative terms, referred to as "GO terms". Each GO term represents a unique functional attribute, and all terms are associated 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 puts all contents into a Swarm (S). Then it takes each Particle (P) from Swarm (S) and extracts each Gene (G) from Particle (P), and computes the Levenshtein Distance score between each gene to query sequence using Eq. (2) (Step 12).

The Levenshtein distance is a string metric used to compare two sequences. The Levenshtein distance between two phrases is the smallest number of single-character modifications (insertions, deletions, or substitutions) needed to transform one word into another.

For two instances, I and C, their lengths are |I| and |C|, and their Levenshtein Distance LevDist(I, C) are described as:

$$\begin{aligned} LevDist(I,C) = \; & \{|I| \; if \; |C| = 0, |C| \; if \; |I| = 0, LevDist(tail(I), tail(C)) \; if \; I[0] \\ & = C[0], 1 \\ & + \{LevDist(tail(I), C) \; LevDist(I, tail(C)) \; LevDist(tail(I), tail(C)) \quad otherwise \end{aligned} \tag{2}$$

Where y[n] is the nth character in the string y, counting from 0, and the tail of a string y is a string of all but the first character of y. Note that at least the first element corresponds to deletion (I to C), the second to insertion, and the third to substitution.

Followed by it computes the fitness score of each Particle (P) (Step 15) and finds the Fittest Particle, which has the 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).

Algorithm 4: Fusion Protein Functionality Prediction based on Particle Swarm Optimization (FPFP_PSO)

Input : UniProtKB / SwissProt Training Dataset (TD) with each Gene Ontology (GO) Category, Fusion Protein Query Sequence (FPQuerySeq)
Output : R2[]
Step 1 : GT[] = Extract each GO_Term from TD
Step 2 : R2[] = "", S[] = "", FN[] = "", i = 0 **// S - Swarm (Population)**
Step 3 : For each GO_Term G from GT
Step 4 : S[i] = Extract the contents of G **// S[i] - ith Particle in Swarm S**
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 Particle P from S
Step 10 : TS = 0
Step 11 : For each Gene G from P
Step 12 : score = Lev_Dis(G, FPQuerySeq) **// Eq. (2) and Function 1**
Step 13 : TS = TS + score
Step 14 : End For
Step 15 : FS = TS / NG **// FS - Fitness Score and NG - Number of Genes in Particle P**
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 Particle selection
Step 22 : PS2 = "", i=0
Step 23 : MS = SR[0] **// MS = Maximum Score**
Step 24 : For each Result R from SI
Step 25 : Extract FS and Iteration from R
Step 26 : If MS is equal to FS, then
Step 27 : i = Iteration
Step 28 : PS2 = FS
Step 29 : break
Step 30 : End If
Step 31 : End For
Step 32 : PT2 = Extract the GO_Term at the ith position in GT[] **// Fittest Particle**
Step 33 : PF2 = Extract the name of the function at the ith position in FN[]
Step 34 : R2[0] = PF2, R2[1] = PS2, R2[2] = PT2
Step 35 : Return R2[]

Step 36 : **Function Lev_Dis(G,FPQuerySeq)**
Step 37 : SS = 0 **// Similarity Score**
Step 38 : If length of G == 0 then
Step 39 : SS = length of FPQuerySeq
Step 40 : End If
Step 41 : If length of FPQuerySeq == 0 then
Step 42 : SS = length of G
Step 43 : End If
Step 44 : Indicator <-- G[0] != FPQuerySeq[0] ? 1 : 0
Step 45 : SS = Minimum(LevDist(G.substr(1), FPQuerySeq) + 1, **// deletion**
LevDist(FPQuerySeq.substr(1), G) + 1, **// insertion**
LevDist(G.substr(1), FPQuerySeq.substr(1)) + Indicator **// substitution**
)
Step 46 : return SS
Step 47 : **End Function**

3. Results & Discussions:

The training dataset of the HybGPSO algorithm was created using the UniProtKB/SwissProt database protein entries. In this work, we utilized annotations with manual curation or experimental evidence, which are extremely dependable. To create the training dataset, the equivalent annotations were extracted from the UniProt-GOA database and 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's 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 well-defined terms separated into three main categories: molecular function, biological process, and cellular component. Figure 3 shows the number of terms for each category.

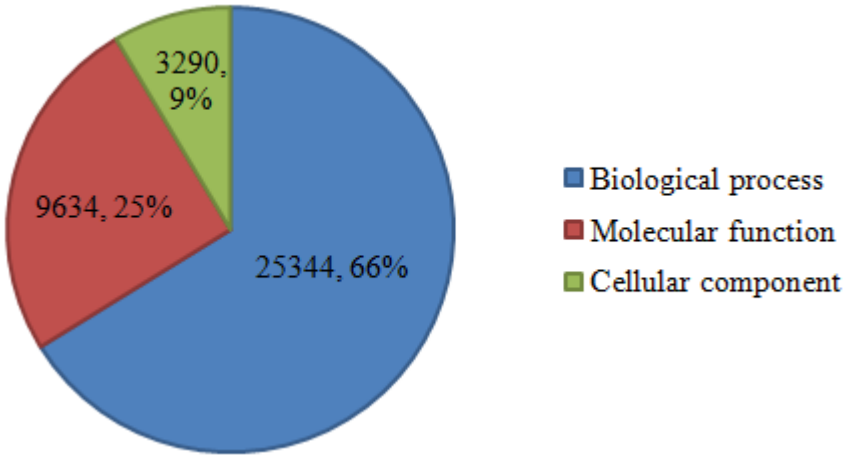

Figure 3: Number of terms for each category in the training dataset

Users can query this dataset with a fusion protein sequence; the proposed HybGPSO 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 4, 5, and 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.

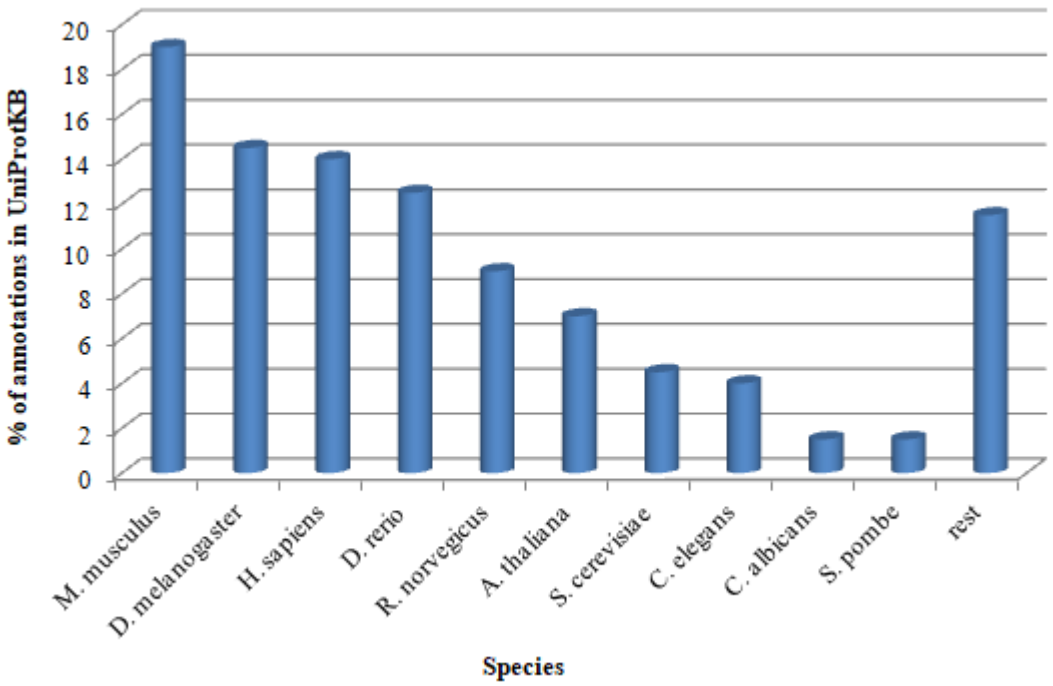

Figure 4: Distribution of annotations from the Biological Process ontologies per species for proteins in UniProtKB

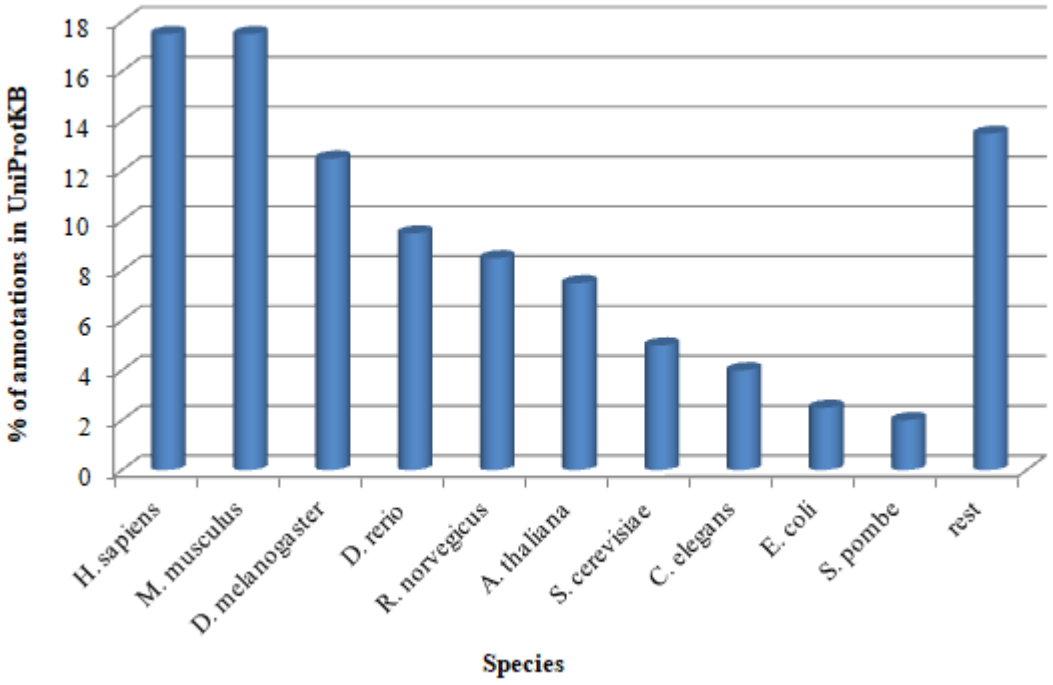

Figure 5: Distribution of annotations from the Molecular Function ontologies per species for proteins in UniProtKB

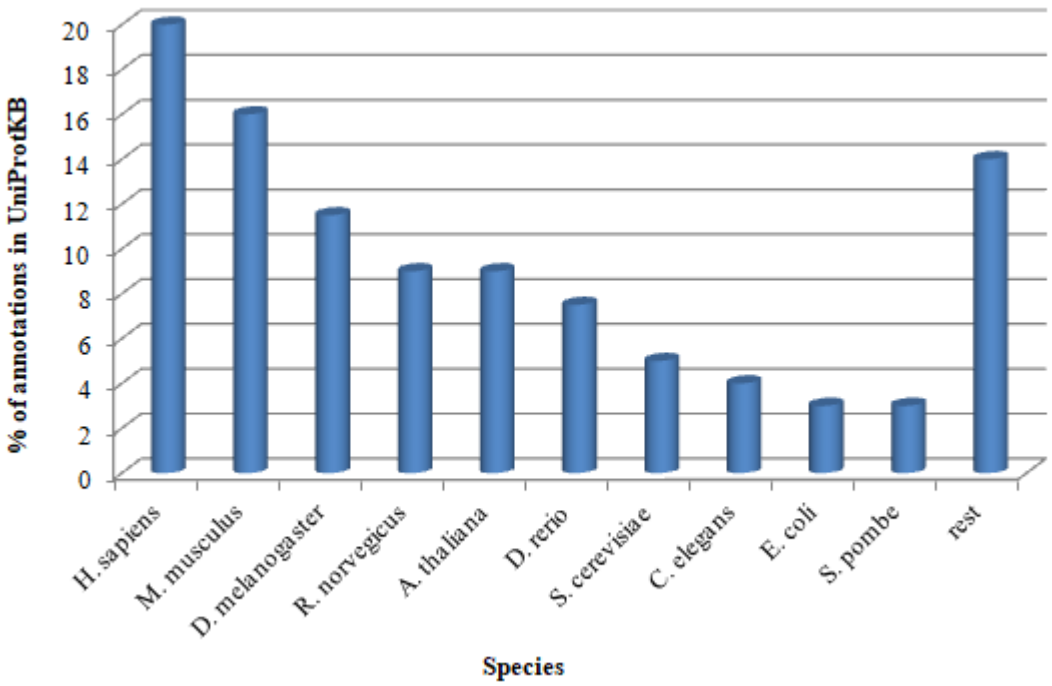

Figure 6: Distribution of annotations from the Cellular Component ontologies per species for proteins in UniProtKB

3.1 Case study 1:

To predict the functionality of the following fusion protein,

Fusion Protein Sequence:

YEHDFHHIREWGNHWKNFLAVMGFFTALSTVMSLLTEVETPIRNEWGCRCNDSS

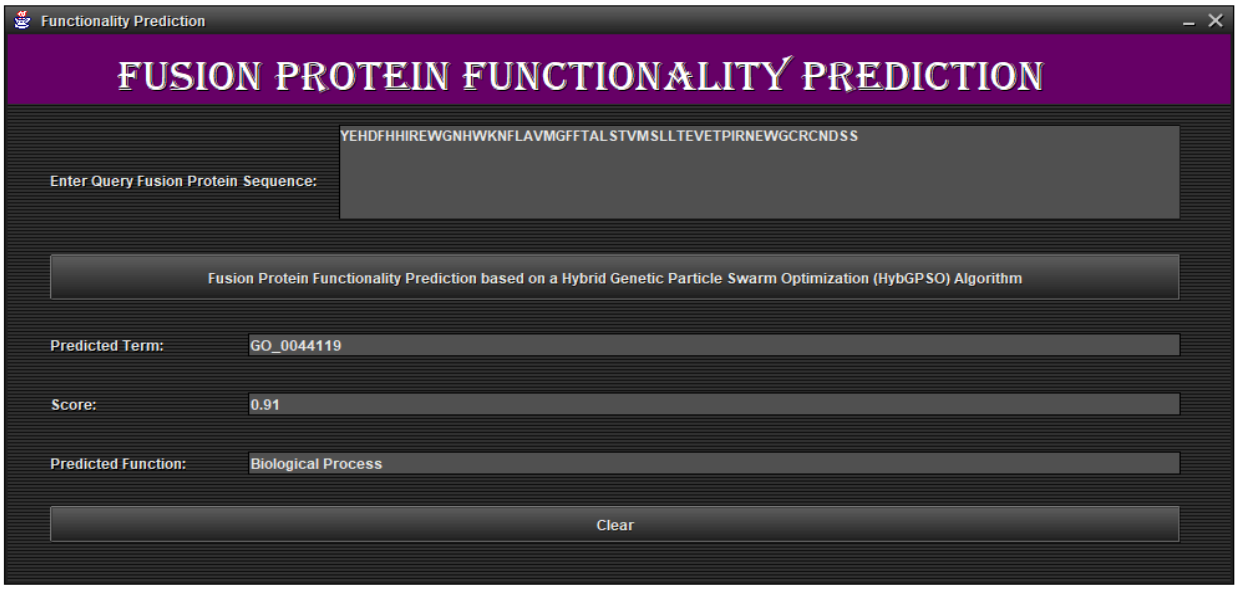

Figure 7: Case study 1

Figure 7 shows the HybGPSO algorithm predicted GO Term is GO_0044119 for the above fusion protein sequence. In addition, its predicted score is 0.91, and its predicted function is a biological process.

3.2 Case Study 2:

Fusion Protein Sequence:

To predict the functionality of fusion protein,

ADAAAGAQVFAANCAACHAGGNNAVMPTKTLKADALKTYLAGYKDGSKSLEEAVAYQVTNGQGAMPAFGGRLS
DADIANVAAYIADQAENNKWIVVLNRAETPLPLDPTGKVKAELDTRMLYLVRMTVNLPRNLDPREEERLKASEKA
RSRTLQEQGQWRYLWRTTGKYGNISVFDVNSHDELHEILWSLPFFPYLTIDVEPLSHHPARVGKD

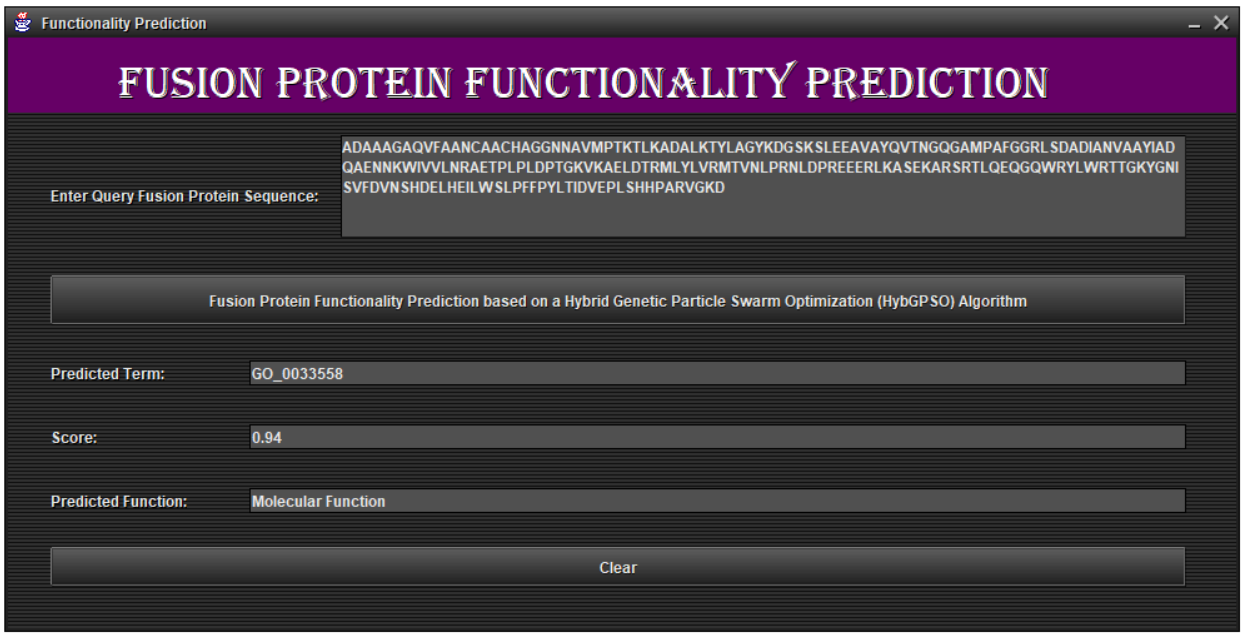

Figure 8: Case study 2

Figure 8 shows the HybGPSO algorithm predicted GO Term is GO_0033558 for the above fusion protein sequence. In addition, its predicted score is 0.94, and its predicted function is a molecular function.

3.3 Case Study 3:

Fusion Protein Sequence:

To predict the functionality of fusion protein,

GSHMNTEVSSEIYQWVRDELKRAGISQAVFARVAFNRTQGLLSEILRKEEDPKTASQSLLVNLRAMQNFLQLPEAER
DRIYQDERERSLRKRKGVTPSTTALPDIVNLSTNYLDKNTREDRIHSIKDFSNADEVENLYTQVADNEYLVQGRMLI
DEFNEVFETDLHMSDVDTMAGYLITALGTIPDEGEKPSFEVGNIKLTAEEMEGTRLLVLRVHFYDEE

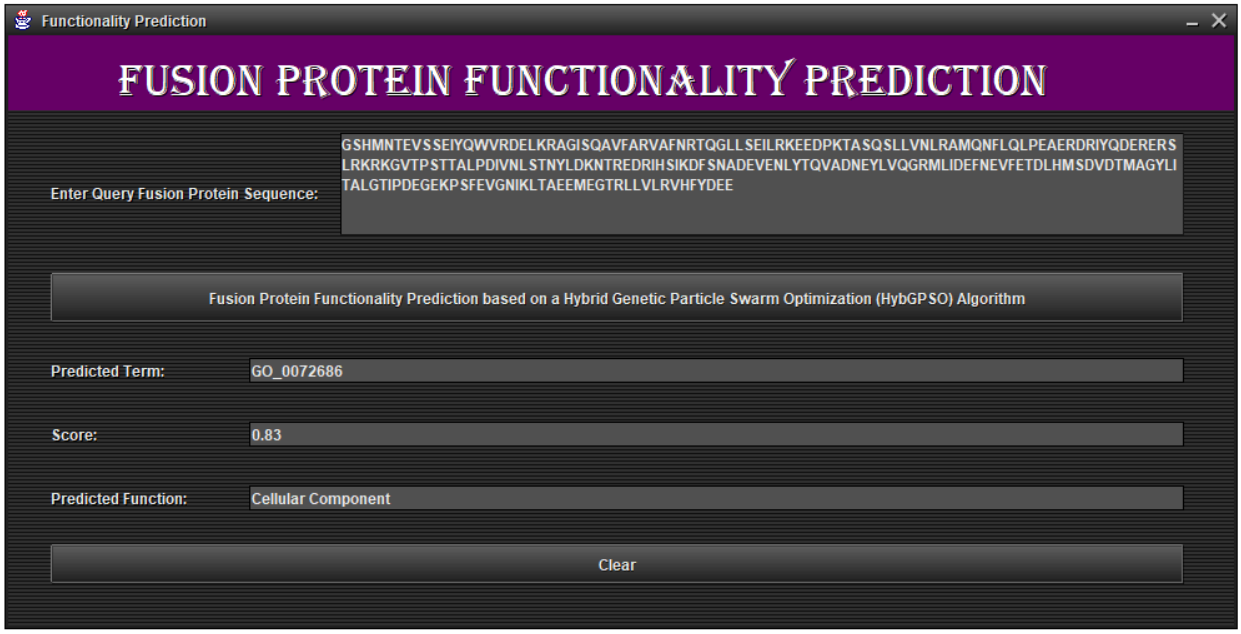

Figure 9: Case study 3

Figure 9 shows the HybGPSO algorithm predicted GO Term is GO_0072686 for the above fusion protein sequence. In addition, its predicted score is 0.83, and its predicted function is a cellular component. Furthermore, function prediction time comparisons of the above case studies are shown in Figure 10.

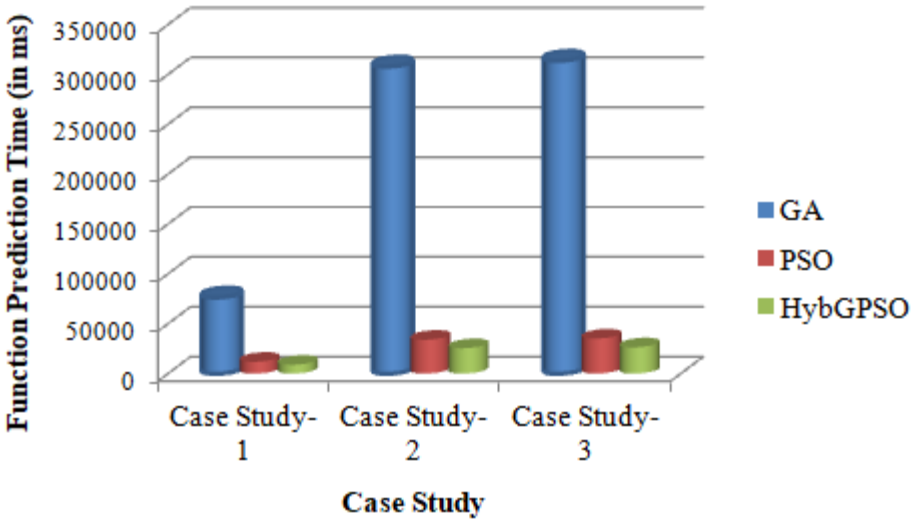

Figure 10: Function Prediction Time comparison

From Figure 10, we know the smallest fusion protein sequence takes less time for function prediction, and the largest fusion protein sequence takes more time. Compared with the existing GA and PSO algorithms, the proposed HybGPSO algorithm takes less time for function prediction.

4. 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 technique based on a Hybrid Genetic Particle Swarm Optimization (HybGPSO) algorithm to deal with this problem. This algorithm predicts the function of fusion protein efficiently. It predicts the cellular component, biological process and molecular function of an unannotated fusion protein by the GO consortium. The experimental results showed the proposed HybGPSO algorithm predicts the function of fusion protein efficiently.

REFERENCES

[1] Pandya, M., Jani, S., Dave, V., & Rawal, R. (2020). Nanoinformatics: An emerging trend in cancer therapeutics. Nanobiotechnology, 135-162.
[2] Huang, Y., Zhang, Y., Li, S., Lin, T., Wu, J., & Lin, Y. (2019). Screening for functional IRESes using an α-complementation system of β-galactosidase in Pichia pastoris. Biotechnology for biofuels, 12(1), 1-12.
[3] Przybilla, M. J., Stewart, C., Carlson, T. W., Ou, L., Koniar, B. L., Sidhu, R., ... & Whitley, C. B. (2021). Examination of a blood-brain barrier targeting β-galactosidase-monoclonal antibody fusion protein in a murine model of GM1-gangliosidosis. Molecular Genetics and Metabolism Reports, 27, 100748.
[4] Hani, S., Cuyas, L., David, P., Secco, D., Whelan, J., Thibaud, M. C., ... & Nussaume, L. (2021). Live single-cell transcriptional dynamics via RNA labelling during the phosphate response in plants. Nature Plants, 7(8), 1050-1064.
[5] 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.
[6] Gligorijević, V., Barot, M., & Bonneau, R. (2018). deepNF: deep network fusion for protein function prediction. Bioinformatics, 34(22), 3873-3881.
[7] 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.
[8] Rison, S. C., Hodgman, T. C., & Thornton, J. M. (2000). Comparison of functional annotation schemes for genomes. Functional & integrative genomics, 1(1), 56-69.
[9] 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.
[10] 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.
[11] Gene Ontology Consortium. (2019). The gene ontology resource: 20 years and still GOing strong. Nucleic acids research, 47(D1), D330-D338.
[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] Jain, A., & Kihara, D. (2019). NNTox: gene ontology-based protein toxicity prediction using neural network. Scientific reports, 9(1), 1-10.
[15] 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.
[16] 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.
[17] Gligorijević, V., Barot, M., & Bonneau, R. (2018). deepNF: deep network fusion for protein function prediction. Bioinformatics, 34(22), 3873-3881.
[18] Liu, X. (2017). Deep recurrent neural network for protein function prediction from the sequence. arXiv preprint arXiv:1701.08318.
[19] Zhao, Z., Zhang, H., Hu, M., Yang, N., Wang, H., Wang, C., ... & Gu, L. (2021). Protein Function Prediction with Deep Neural Learning.
[20] Gligorijević, V., Barot, M., & Bonneau, R. (2018). deepNF: deep network fusion for protein function prediction. Bioinformatics, 34(22), 3873-3881.
[21] Si, Z., Zhang, J., Shivakoti, S., Atanasov, I., Tao, C. L., Hui, W. H., ... & Zhou, Z. H. (2018). Different functional states of fusion protein gB were revealed on human cytomegalovirus by cryo electron tomography with a Volta phase plate. PLoS pathogens, 14(12), e1007452.
[22] Gligorijević, V., Renfrew, P. D., Kosciolek, T., Leman, J. K., Berenberg, D., Vatanen, T., ... & Bonneau, R. (2021). Structure-based protein function prediction using graph convolutional networks. Nature communications, 12(1), 1-14.
[23] Sureyya Rifaioglu, A., Doğan, T., Jesus Martin, M., Cetin-Atalay, R., & Atalay, V. (2019). DEEPred: automated protein function prediction with multi-task feed-forward deep neural networks. Scientific reports, 9(1), 1-16.
[24] Zhang, F., Song, H., Zeng, M., Li, Y., Kurgan, L., & Li, M. (2019). DeepFunc: a deep learning framework for accurately predicting protein functions from protein sequences and interactions. Proteomics, 19(12), 1900019.
[25] Kulmanov, M., & Hoehndorf, R. (2020). DeepGOPlus: improved protein function prediction from the sequence. Bioinformatics, 36(2), 422-429.
[26] 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.
[27] Cai, Y., Wang, J., & Deng, L. (2020). SDN2GO: an integrated deep learning model for protein function prediction. Frontiers in bioengineering and biotechnology, 8, 391.
[28] Yao, S., You, R., Wang, S., Xiong, Y., Huang, X., & Zhu, S. (2021). NetGO 2.0: improving large-scale protein function prediction with massive sequence, text, domain, family and network information. Nucleic Acids Research, 49(W1), W469-W475.
[29] Makrodimitris, S., van Ham, R. C., & Reinders, M. J. (2019). Improving protein function prediction using protein sequence and GO-term similarities. Bioinformatics, 35(7), 1116-1124.