GO term predictions in CATH: a machine learning approach

ICS5200 - Dissertation Progress Report

Kenneth Penza
Department of Artifical Intelligence
Faculty of ICT
University of Malta
kenneth.penza.16@um.edu.mt

ABSTRACT

Protein annotation is the process of describing the function or functions of a given sequence. Technological advances in sequencing gave rise to an unprecedented scale of unannotated proteins. Through annotations biologists can understand better, the workings of organisms. The high quality experimental annotations entail a laborious and costly process. Mismatch in sequencing and annotation rates highlights the need of an automated large-scale process to annotate proteins with high reliability. Protein ancestry (homology) is not a reliable indicator of the protein function. Homologues can have high sequence similarity and yet different functionality. Moonlight proteins are yet another challenge, whereby one protein performs multiple functions. This work aims to investigate protein features and employ bioinformatics tools to build a feature set for reliable function prediction. The performance of the proposed system will be evaluated against CAFA.

KEYWORDS

Bioinformatics, Artifical Intelligence, Machine Learning, Protein Function Prediction, Gene Ontology, Protein Annotation

1 INTRODUCTION

Proteins are composed of linear chains of amino acids connected by peptide bonds. Amino acids are composed of small molecules with a common backbone (C, H, O and N) and a side chain of up to 30 more atoms (C, H, O, N, and S) [22]. There are twenty naturally occurring amino acids that make up protein sequences. Protein sequences fold up into 3D structures that give the protein its functionality [13]. Lesk [18] details four structural levels of a protein, that are illustrated in Figure 1.

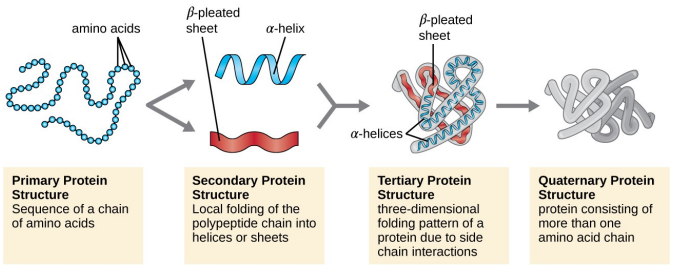

Figure 1: The four levels of proteins structure reproduced from [1]

[1]https://courses.lumenlearning.com/microbiology/chapter/proteins/

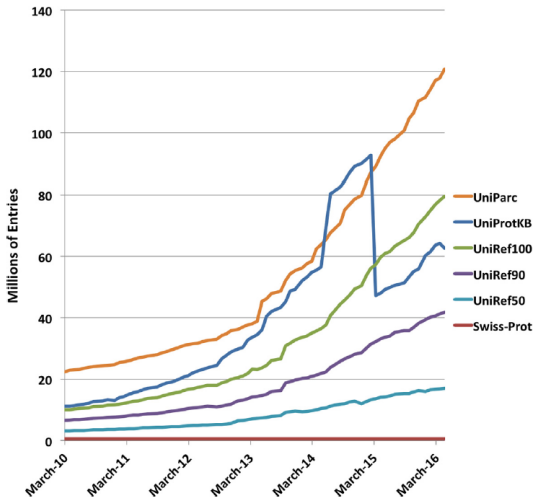

Figure 2: Growth of the number of protein sequences in UniProt databases. UniProt TrEMBL is illustrated using the blue line. The drop in TrEMBL growth is linked to the implementation of Proteome Redundancy Minimization (PRM) implemented on March 2015 [8]

A protein is composed of one or more structures called domains that form the functionality block of the protein. Proteins that perform multiple functions are called multi-domain proteins [10].

Advancements in sequencing technology is improving the efficiency, availability and affordability of genomic sequencing machines. Biological sequences are submitted to molecular databases to be catalogued and analysed. Laboratory experiments are required to determine the functionality of a protein within an organism. However this process is costly and time consuming [17]. The annotation process entails attaching experiment outcomes in biological databases.

The low throughput of experimental curation is evident with the growth experienced by biological databases. In 2013, the data repositories managed by European Bioinformatics Institute (EBI) required 18 petabytes, increasing more than two fold in a year to reach 40 petabytes [16]. This trend is also visible across UniProt databases. Uniprot is a collection of protein sequences and functional annotation. Figure 2, illustrates the growth experienced by UniProt databases between 2010 and 2016. The gradient of individual curves shows that the databases are growing at an increasing rate.

UniProtKB database is divided into two sections, the manually annotated section "Swiss-Prot" and the automatically annotated section "TrEMBL" [8]. In October 2017, for *Homo sapiens* organism, Swiss-prot contained 20,237 sequences whilst TrEMBL contained 140,126 sequences. Consequently only 14.4% of the *Homo sapiens* sequences are experimentally annotated. The difference in the annotation rate and the rapidly increasing number of unannotated sequences, means that most of the sequences will have a predicted role. This has made protein function prediction a central aspect within computational biology [23].

The efficiency in genomic sequencing is resulting in accumulation of unannotated sequences. The ability to map protein sequences to functionality would unlock better understanding of organisms. Within drug design and development, it would enable medicinals to target specific proteins reducing side effects [3].

The necessity to identify the function of a sequence gave rise to a number of approaches to identify protein function. The rationale being, to give the sequence a predicted role till it is laboratory verified or changed.

An approach to assign functionality, is to utilise conserved regions within the protein sequence. Conserved regions within protein sequences implies ancestry. The ancestry link between sequences can be used to transfer functionality between sequences. This technique leverages on annotations of similar sequences to determine the function of a sequence. However this technique is not reliable with low sequence similarity [27].

Machine learning (ML) techniques enable systems to learn from the data. The performance of ML techniques depends on the features and data available. ML has been utilised in bioinformatics, where the output is the protein function and in some cases the prediction confidence [1]. Predicted protein functions are annotated in biological databases clearly marking the annotation origin.

2 MOTIVATION

Protein function prediction is a central problem within bioinformatics. To date there is no protein function prediction technique that can replace high quality experimental annotations. Annotations are the mechanism used store protein sequence functionality and related evidence in biological databases. This research exploits bioinformatics computational methods and machine learning techniques to predict protein function. Generated predictions (annotations) will be utilised to hint researchers the functionality of a given sequence. Protein function prediction is a complex problem. Sequence similarity determines ancestry relationship between genes through conserved regions. Homology relationship cannot be used to transfer functionality, as in the case of paralogues and moonlighting proteins.

3 AIMS AND OBJECTIVES

The aim of this research is to perform protein function prediction of a given protein sequence. The defined aim will be achieved by fulfilling the following objectives:

(1) Develop a scalable web enabled platform through which a user can input protein sequences and visualise results
(2) Identify protein features that enable accurate protein function prediction
(3) Build a pipeline with the required bioinformatics tools to generate required features
(4) Utilise CAFA evaluation metrics to enable comparison with other solutions

4 BACKGROUND RESEARCH AND LITERATURE REVIEW

Two species can be linked together if they have common features, for example adaption to a specific environmental condition. In bioinformatics, gene homology is divided into two subclasses, paralogues and orthologues. Paralogues are genes linked through gene duplication whilst orthologues are genes related through speciation. Within the evolutionary context, these two subclasses have different endings. Paralogues have duplicate functionality and consequently in the long term they either diverge functionality or are lost. On the other hand orthologues tend to take the function of their precursor and thus are conserved [9, 24].

Moonlighting proteins are a class of multi functional proteins in which a single protein performs multiple functions. Essentially moonlighting proteins can be structurally described as proteins that has two biochemical functions in one peptide chain [14]. These proteins are found in different organisms including mammals, bacteria and archaea [12, 14]. MoonProt is a database of moonlighting proteins that contains information about 270 moonlighting proteins and related evidence [2]. Moonlighting complicates function prediction due to two aspects. First, there is evidence that homologues of moonlighting proteins may not exhibit moonlighting. Consequently function prediction techniques using homology might fail [12, 14]. Secondly, there is no single tool that identifies all functions of a given moonlighting proteins [14].

With the work being performed on sequences, biologists are acknowledging the fact that one gene universe exists [2]. Organism specific databases classify proteins using a classification or hierarchy oriented towards the target organism. Organism specificity in describing proteins hinders the ability to ask organism independent questions [5].

The Gene Ontology(GO) initiative by the Gene Ontology Consortium has three main aims [6]:

(1) Develop ontologies to describe molecular biology
(2) Apply GO ontologies to biological databases.
(3) Publish ontologies to enable universal access

GO defines three non-overlapping ontologies [2, 5, 6]:

(1) **Biological Process** describes the series of events or molecular functions.
(2) **Cellular Component** describe locations, at the level of subcellular structures and macromolecular complexes.
(3) **Molecular Function** describes the basic abilities at molecular level.

The GO is defined as a directed acyclic graph (DAG) whereby a vertex represents a term and an edge defines the relationship between the terms. In the DAG, a vertex representing a term can have multiple parents. GO terms are structured to support "is_a" (subtype of), "part_of" (subcomponents of parent),

[2] http://moonlightingproteins.org/

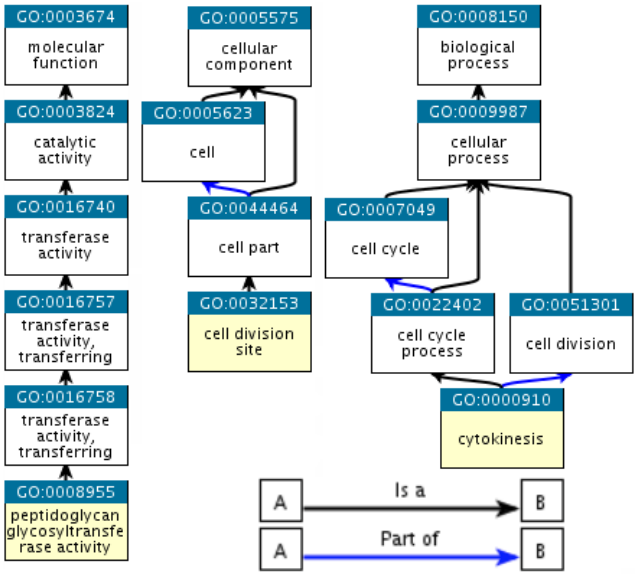

Figure 3: GO DAG of different terms, adapted from [3]

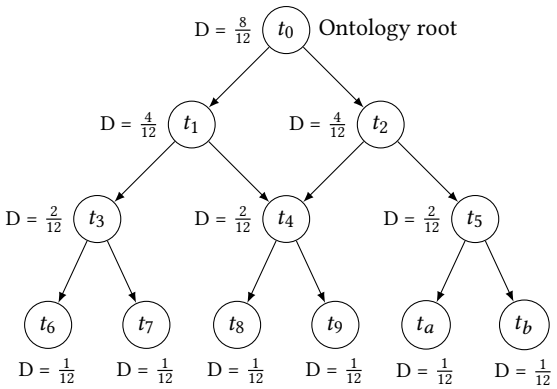

Figure 4: GO DAG with the D-Value computed for all nodes, using the methodology defined by Zhang et al. [26]

.

regulates (necessarily-regulates), "positively_regulates", "negatively_regulates" and "has_part" (necessarily part of) relationships. The GO follows the "True Path Rule" whereby the path between term and the ontology root must always be true [6, 7]. Figure 3, illustrates the GO DAGs' generated for three GO terms within different ontologies.

Annotation is the process of storing the results of a prediction algorithm or an experiment using the appropriate GO terms. For each attributed GO term, evidence supporting citations together with the evidence code must be provided. The evidence code, enables biologists to determine the annotation quality. High quality annotations originate from experimental evidence whilst annotation obtained through automatic methods are less reliable [6].

[3]https://www.ebi.ac.uk/QuickGO/

Within bioinformatics protein similarity is utilised for different reasons including finding proteins that have the same function and determining the performance of a predictor. For a given protein the true GO term is t_b whilst the predicted term is t_4. Figure 4 illustrates the DAG including terms t_4 and t_b. Term t_b is more specific than t_4 as it occurs at a deeper level in the DAG. The following subsection, tackles metrics proposed to quantify and compare the similarity of protein sequences.

4.1 Semantic Similarity

Protein function similarity metrics has been the motivation of various researchers that tackle the problem from different perspectives. Mazandu et al. [19], categorises these efforts into three main categories:

(1) **Information content** metrics computed on either annotations or ontology structure
(2) **Term Semantic Similarity** metrics computed on either terms, edges or both
(3) **Functional similarity** metrics computed using ontology structure or protein annotations

Information content (IC) measures computes similarity values using an annotation dataset. The IC value is dependent on the dataset utilised. Similarly, topology based IC computations are susceptible to GO structural changes as new terms are added and deprecated ones are removed [19].

Zhang et al. [26], proposed Gene functional similarity search tool (GFSST). Within GFSST, each node is assigned D-value (Distribution value), for each node it is computed as the sum of incoming links divided by the total number of nodes in the DAG. Comparison rules define the D-value of two genes having multiple terms as the mean D-value of the common terms.

Figure 4, illustrates the computed D-value for a DAG graph. For a given sequence the predicted term is t_4 whilst the actual term is t_b. Considering all the paths from the root node t_0 to t_4 (t_0, t_1, t_2 and t_4) and from t_0 to t_b (t_0, t_2, t_5 and t_b). The common terms from the two sets are extracted (t_0 and t_2) and the minimum D-Value ($\frac{4}{12}$) is the value of the metric.

The work proposed by Wang et al. [25] uses a different technique to compute the node S-Value (similarity value). The starting node is given S-Value of 1, for the child nodes, the weight of the node is the edge weights (0.8 for "is_a" and 0.6 for "part_of") multiplied by maximum node weight. The comparison between two genes is summation of the maximum S-Value of each GO term of G1 against G2 and each GO term of G2 against G1. Through this approach, all the nodes in the sub graph are considered.

Subsequently Mazandu and Mulder [20] proposed a topology based similarity metric based (GO-UNIVERSAL) aimed to address the issue of depth within the GO DAG. The similarity computation performed within GO-UNIVERSAL considers the depth of the term, whereby higher terms are generic whilst deeper terms are more specific. Within GO-UNIVERSAL for each node two values are computed, the topological position and topological information which in turn is a function of topological position. The information content score for each node is computed using the topological information. A GO term similarity score between two terms is computed using information content of the two terms. Gene similarity

functionality iterates over the GO terms of the gene and computes a similarity score.

The work by Clark and Radivojac [4], proposes a framework that models a Bayesian network with prior distributions on the GO DAG. The computation of prior distributions, references a database of annotated proteins such as Swiss-Prot. Using the annotations found in the dataset, marginal probabilities and information content for each term is computed using Equation 1.

The "information accretion" is utilised to compute two metrics, namely remaining uncertainty and misinformation in defined in Equations 2,3 respectively. The remaining uncertainty is information that is yet to be predicted when compared to the true positive set. Whilst the "misinformation" is terms that were incorrectly predicted. To facilitate ranking and evaluation of function prediction methods these metrics are combined into "Semantic distance" defined in Equation 4. "Semantic distance" is the minimum distance between the origin and the curve $(ru^k(\tau) + mi^k(\tau))_\tau$. The preferred distance metric is Euclidean distance attained by using $k = 2$.

$$ia(v) = \sum_{v \in T} log \frac{1}{Pr(v|\mathcal{P}(v))} \tag{1}$$

$$ru(T, P) = \sum_{v \in T-P} ia(v) \tag{2}$$

$$mi(T, P) = \sum_{v \in P-T} ia(v) \tag{3}$$

$$S_k = \min_\tau (ru^k(\tau) + mi^k(\tau))^{\frac{1}{k}} \tag{4}$$

where v is a node in the graph, $\mathcal{P}(v)$ is the set of parent nodes of v, T is the true positive sub graph and P is the predicted function sub graph.

4.2 CAFA evaluation

CAFA evaluation utilises set based and information theoretic metrics based on Equations 2,3,4 [15]. Set based metrics are computed using Equations 5,6,7, whilst information theoretic using Equations 8,9,10. The performance of CAFA submissions is evaluated using F_{max} and S_{min}.

$$pr(\tau) = \frac{1}{m(\tau)} \sum_{i=1}^{m(\tau)} \frac{\sum_f \mathbb{1}(f \in P_i(\tau) \wedge f \in T_i)}{\sum_f \mathbb{1}(f \in P_i(\tau))} \tag{5}$$

$$rc(\tau) = \frac{1}{n_e} \sum_{i=1}^{n_e} \frac{\sum_f \mathbb{1}(f \in P_i(\tau) \wedge f \in T_i(t))}{\sum_f \mathbb{1}(f \in T_i(t))} \tag{6}$$

$$F_{max} = \max_\tau \frac{2.pr(\tau).rc(\tau)}{pr(\tau) + rc(\tau)} \tag{7}$$

$$ru(\tau) = \frac{1}{n_e} \sum_{i=1}^{n_e} \sum_f ic(f).\mathbb{1}.(f \notin P_i(\tau) \wedge f \in T_i) \tag{8}$$

$$mi(\tau) = \frac{1}{n_e} \sum_{i=1}^{n_e} \sum_f ic(f).\mathbb{1}.(f \in P_i(\tau) \wedge f \notin T_i) \tag{9}$$

$$S_{min} = \min_\tau \sqrt{ru(\tau)^2 + mi(\tau)^2} \tag{10}$$

where $P_i(\tau)$ is the set predicted with a score that are greater than or equal to τ for a protein sequence, T_i is the true positive set the sequence, $m(\tau)$ is the number of sequences with at least one score greater than or equal to τ, $\mathbb{1}$ is the indicator function that returns 1 when the condition is true otherwise 0, n_e is the number of targets in evaluation, $ic(f)$ is the information content for term f and τ a value for 0.01 to 1.00 incremented by 0.01.

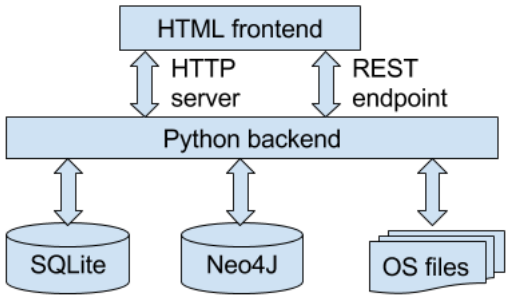

Figure 5: Proposed research architecture

The next section details the proposed solution and work performed.

5 PROPOSED SOLUTION

This section tackles the practical aspect of this research. It details the work carried out, high level solution, dataset, hardware and software requirements.

5.1 DataSet

In the initial phases, a dataset was required to understand the data and develop the require software. Dr. Lees [4] from University College London (UCL) made available a CAFA representative dataset.

The dataset contains 3,293,750 domains originating from 31,097 sequences. Within the dataset there are 6,388 unique GO terms. Each domain entry is described with sixty attributes including GO term, sequence name, protein disorder metrics, pFAM to GO term, prediction of cleavage sites and transmembrane helices, ancestry taxa, mmseq2 scores (many to many mapping) and CATH, PFam, PFam funfam metrics.

This research uses additional data sources including UniProt [8] and CATH [21] for verification and dataset enrichment purposes.

5.2 Method

The problem being tackled is a supervised multi-class classification machine learning problem. Whereby each instance in the "training" dataset is labelled with the respective GO term. The number of classes (GO terms) in the dataset is 6,388 classes, making the problem a multi-class one.

The development and prototyping will be carried out on a desktop system. However, processing the full dataset requires a larger infrastructure. The plan is to develop software in a modular architecture, to enable deployment on a distributed architecture. The main components illustrated in Figure 5 entail:

(1) a graph database back end to store the Gene Ontology (GO) structure
(2) python back end code that comprises of:
  (a) feature generation for input proteins

[4] https://github.com/jonglees

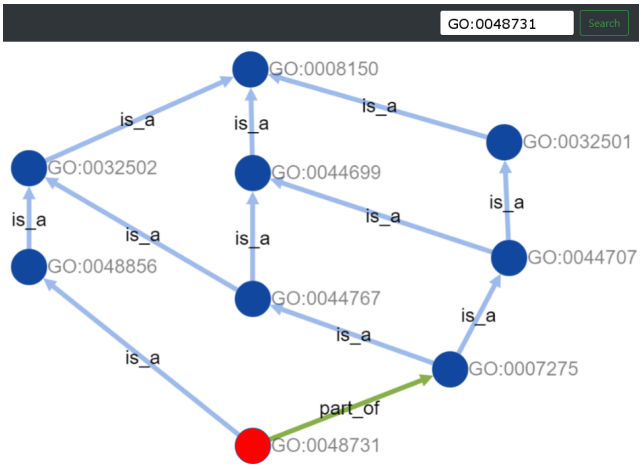

Figure 6: GO DAG for term "GO:0048731" in visualiser prototype developed

(b) machine learning core to perform predictions
(c) computation of information theoretic metrics
(3) front end to interact with user and visualise results

Python was selected as the main language for this work as it provides the required libraries and tools. The graph database was required to act as a repository for hierarchical relationships between the different GO terms. For this research *Neo4j* community edition was utilised as it full fills the requirements. SQLite will be used to persist Pandas dataframes. The research solution will be developed and deployed on Linux operating system running Ubuntu 16.04 distribution. All the software developed will be versioned in Git including LaTeX documents.

5.3 Ongoing work

Building on the knowledge acquired in Section 4, a number of tasks and experiments were performed.

The dataset provided by Dr. Lees from UCL was processed using a Python utility to generate the "training" and "testing" datasets. From the provided dataset, sequences were randomly selected until the number of domains exceeded 5,000. The selected entries were written to file. This process is repeated to generate the "training" and "testing" datasets.

The initial efforts focused on understanding the data set. For this purpose RapidMiner [5] was utilised to prototype the machine learning pipeline. The following steps were performed:

(1) Read the training and testing datasets
(2) Train RandomForest Classifier using entropy as split criteria
(3) For n between 1 and number of features in the dataset
  (a) Select n top important features from classifier
  (b) Amend datasets to include only selected features
  (c) Train RandomForest Classifier using k-folds cross validation (k=10)
  (d) Measure accuracy using test dataset

[5] https://rapidminer.com/

Table 1: GO ontology statistics

Ontology	No. of terms	No. of edges	No. of paths
Biological Process	28,431	69,222	4,617,763
Cellular Component	3,895	7,318	208,144
Molecular Function	10,015	12,227	23,477

Table 2: CAFA aligned dataset from UCL

Taxa	No. of families	No. of sequences
Archaea	2	2,457
Bacteria	6	107,711
Eukaryote	12	3,183,582

Table 3: CAFA2 dataset details

	CAFA2	
Taxa	No. of families	No. of sequences
Archaea	7	3,291
Bacteria	10	15,451
Eukaryote	10	82,074

The RapidMiner prototype attained the best accuracy of 71% when using two features. Following prototyping, the ML pipeline was implemented in Python. The accuracy attained from the Python implementation was 71.4% which is inline with the RapidMiner prototype. In preparation for the evaluation, information theoretic metrics were implemented. Further work is ongoing to understand and select the appropriate features from the dataset.

Gene Ontology Consortium provides ontology downloads in two formats, OBO and MySQL dump. For this research the MySQL dump was utilised. A Python utility was developed to read the relational data from *MySQL* and populate the *Neo4j* graph database. A random sample of GO terms were selected and verified against QuickGO [6] to validate the loading process. The *Neo4j* database was analysed to determine some basic statistics of the ontologies, that are reported in Table 1.

The GO visualiser has two main components. The front end is a website developed using Bootstrap, Cytoscape and jQuery. The back end is a web server and REST service developed in Python. From the web page, the user can search for GO terms. For a particular GO search, a REST call is performed to the back end service via AJAX and in turn the back end responds with a JSON document representing the DAG for the GO term. Figure 6 illustrates the GO map for GO term "GO:0048731" rendered in the developed prototype. The visualisation uses different edge colours to highlight edge types ("is_a" and "part_of") that connect GO terms.

The dataset was enriched with sequence information from UniProt to include sequence name and organism information. This additional information enabled classification per taxa as per Table 2 and comparison of taxa with CAFA2 dataset as per Table 3. The GO terms in the dataset were queried in *Neo4j* to determine the

[6] https://www.ebi.ac.uk/QuickGO/

Table 4: CAFA2 best performance metrics

Ontology	F_{max}	S_{min}
Biological Process	0.38	17.5
Cellular Component	0.46	4.5
Molecular Function	0.60	6.2

ontology. The 6,388 GO terms present in the test dataset originate from the Molecular Function ontology.

6 EVALUATION PLAN

Evaluation of protein function prediction tools is not a trivial task. Given an unknown protein, the system will perform function predictions, one for each domain in the sequence. These annotations must be compared with the "gold standard" to determine system performance. "Gold standard" annotations are obtained from laborious expensive laboratory experimentation. In case a protein has just been discovered, "gold standard" annotation will be unavailable. Consequently, the performance of the system cannot be measured on this new protein.

The proposed work will be evaluated using the Critical Assessment of protein Function Annotation (CAFA) methodologies. CAFA is a competition to evaluate the current state of computation protein function prediction. Two CAFA challenges have been completed, namely CAFA1 (2010-2011) [22] and CAFA2 (2013-2014) [15] whilst CAFA3 is currently in progress. CAFA is organised every three years. The competition has three phases, the first phase a number of unannotated sequences are given the community. In the prediction phase, the community has 4 months to analyse the data and submit predictions. During the second phase, the competition waits for nine months for proteins to be experientially annotated. After the nine months, the assessment phase is performed. The protein annotations are extract from Swiss-Prot and submissions performance is assessed [11].

This work will be evaluated using CAFA2 methodology and through literature comparison with other solutions. The CAFA2 dataset will be utilised and is publicly available from [7]. Utilising CAFA dataset and methodology will provide a robust evaluation platform. Table 4, reports the performance metrics of the best performing CAFA2 submission per ontology [15].

7 CONCLUSION

The number of protein sequences is accumulating at an ever-increasing rate in biological databases, such as UniProt. Determining the function of a given protein is a central aspect of understanding organisms and drug design. Experimental laboratory curation is required to determine protein function however, this process is laborious and costly. The protein functions of a given sequence are stored in biological databases as annotations with related evidence. A stop-gap solution is the utilisation of computational techniques to determine protein function. Predicted functions are annotated in databases using the GO terms and supporting evidence.

[7] http://biofunctionprediction.org/cafa/

This work leverages bioinformatics tools and machine learning techniques to perform protein function prediction. This work will be evaluated against CAFA submissions in terms of performance, whereby the best results per ontology are reported in Table 4. This work will be scheduled as per schedule available in Figure 7.

REFERENCES

[1] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. 2016. Deep learning for computational biology. *Molecular systems biology* 12, 7 (2016), 878.

[2] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. 2000. Gene Ontology: tool for the unification of biology. *Nature genetics* 25, 1 (2000), 25.

[3] Seth I Berger and Ravi Iyengar. 2009. Network analyses in systems pharmacology. *Bioinformatics* 25, 19 (2009), 2466–2472.

[4] Wyatt T Clark and Predrag Radivojac. 2013. Information-theoretic evaluation of predicted ontological annotations. *Bioinformatics* 29, 13 (2013), i53–i61.

[5] Gene Ontology Consortium et al. 2001. Creating the gene ontology resource: design and implementation. *Genome research* 11, 8 (2001), 1425–1433.

[6] Gene Ontology Consortium et al. 2004. The Gene Ontology (GO) database and informatics resource. *Nucleic acids research* 32, suppl 1 (2004), D258–D261.

[7] Gene Ontology Consortium et al. 2010. The Gene Ontology in 2010: extensions and refinements. *Nucleic acids research* 38, suppl 1 (2010), D331–D335.

[8] UniProt Consortium et al. 2017. UniProt: the universal protein knowledgebase. *Nucleic acids research* 45, D1 (2017), D158–D169.

[9] Sayoni Das, Natalie L Dawson, and Christine A Orengo. 2015. Diversity in protein domain superfamilies. *Current opinion in genetics & development* 35 (2015), 40–49.

[10] Sayoni Das and Christine A Orengo. 2016. Protein function annotation using protein domain family resources. *Methods* 93 (2016), 24–34.

[11] Iddo Friedberg and Predrag Radivojac. 2017. Community-wide evaluation of computational function prediction. *The Gene Ontology Handbook* (2017), 133–146.

[12] Daphne HEW Huberts and Ida J van der Klei. 2010. Moonlighting proteins: an intriguing mode of multitasking. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research* 1803, 4 (2010), 520–525.

[13] Lawrence Hunter. 1993. Molecular biology for computer scientists. *Artificial intelligence and molecular biology* (1993), 1–46.

[14] Constance J Jeffery. 2015. Why study moonlighting proteins? *Frontiers in genetics* 6 (2015).

[15] Yuxiang Jiang, Tal Ronnen Oron, Wyatt T Clark, Asma R Bankapur, Daniel D'Andrea, Rosalba Lepore, Christopher S Funk, Indika Kahanda, Karin M Verspoor, Asa Ben-Hur, et al. 2016. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome biology* 17, 1 (2016), 184.

[16] Hirak Kashyap, Hasin Afzal Ahmed, Nazrul Hoque, Swarup Roy, and Dhruba Kumar Bhattacharyya. 2015. Big data analytics in bioinformatics: A machine learning perspective. *arXiv preprint arXiv:1506.05101* (2015).

[17] David Lee, Oliver Redfern, and Christine Orengo. 2007. Predicting protein function from sequence and structure. *Nature reviews. Molecular cell biology* 8, 12 (2007), 995.

[18] Arthur Lesk. 2013. *Introduction to bioinformatics.* Oxford University Press.

[19] Gaston K Mazandu, Emile R Chimusa, and Nicola J Mulder. 2016. Gene Ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery. *Briefings in bioinformatics* (2016), bbw067.

[20] Gaston K Mazandu and Nicola J Mulder. 2012. A topology-based metric for measuring term similarity in the gene ontology. *Advances in bioinformatics* 2012 (2012).

[21] Christine A Orengo, AD Michie, S Jones, David T Jones, MB Swindells, and Janet M Thornton. 1997. CATH–a hierarchic classification of protein domain structures. *Structure* 5, 8 (1997), 1093–1109.

[22] Predrag Radivojac. 2013. A (not so) quick introduction to protein function prediction. (2013).

[23] Andrea Scaiewicz and Michael Levitt. 2015. The language of the protein universe. *Current opinion in genetics & development* 35 (2015), 50–56.

[24] Günter Theißen. 2002. Orthology: secret life of genes. *Nature* 415, 6873 (2002), 741–741.

[25] James Z Wang, Zhidian Du, Rapeeporn Payattakool, Philip S Yu, and Chin-Fu Chen. 2007. A new method to measure the semantic similarity of GO terms. *Bioinformatics* 23, 10 (2007), 1274–1281.

[26] Peisen Zhang, Jinghui Zhang, Huitao Sheng, James J Russo, Brian Osborne, and Kenneth Buetow. 2006. Gene functional similarity search tool (GFSST). *BMC bioinformatics* 7, 1 (2006), 135.

[27] Xing-Ming Zhao, Luonan Chen, and Kazuyuki Aihara. 2008. Protein function prediction with high-throughput data. *Amino Acids* 35, 3 (2008), 517.

A PROJECT TIME LINE

The project plan for this research is detailed in Figure 7. The work is being carried out on part-time basis for a period of a year. On completion, the research will be presented as the dissertation for a masters level in Artificial Intelligence. Work will be tracked via the regular meetings that will be held with my supervisor and co-supervisor.

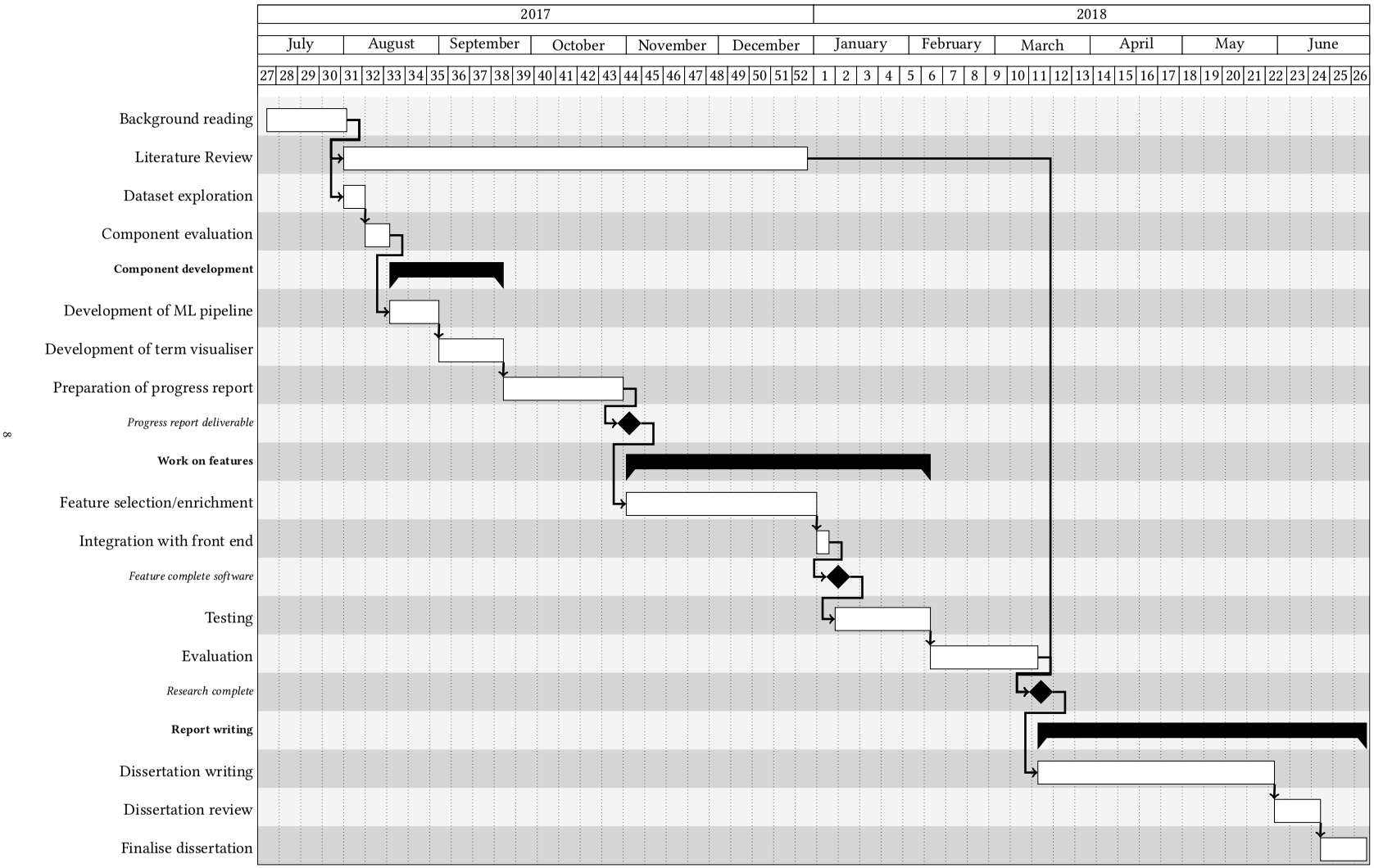

Figure 7: Research Gantt chart