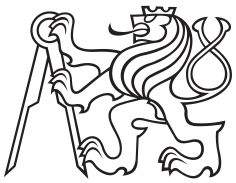
Bachelor Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Hierarchical Multi-Label Classification for Automated Protein Function Prediction

Martin Miadok

Supervisor: doc. Ing. Jiří Kléma, Ph.D.
May 2022

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Miadok Martin** Personal ID number: **492292**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence and Computer Science**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Hierarchical Multi-Label Classification for Automated Protein Function Prediction

Bachelor’s thesis title in Czech:

Automatická predikce funkce bílkovin jako hierarchická multi-label klasifikace

Guidelines:

Automatic prediction of protein function is a task where we try to predict a protein function based on a known protein sequence. The function is represented by a list of terms taken from a predefined gene ontology which corresponds to a hierarchical taxonomy of these terms. The function prediction is thus a sequence of binary decisions, each of the terms is either used or not used for a given sequence. As the terms are not independent, it is advisable to work with these dependencies and regularize the final list of terms.
Requirements:
1. Get acquainted with the current state of the art in hierarchical multi-label classification.
2. Get familiar with the successful solutions of the task of automatic prediction of protein function in the CAFA (The Critical Assessment of Protein Function Annotation) competition.
3. Carry out a literature search for methods ad 1 and 2.
4. Design your own algorithm / modification of an existing algorithm for automatic prediction of protein function.
5. Compare your solution with the basic benchmarks (BLAST + kNN, priors), or with the methods discussed in the search with available implementation, use the traditional measures of evaluating the quality of classifiers (precision, recall, F1).

Bibliography / sources:

[1] Hamp, T., et al. "Homology-based inference sets the bar high for protein function prediction." BMC Bioinformatics 14, Suppl 3 (2013), S7.
[2] Triguero, I., and Vens, C. "Labelling strategies for hierarchical multi-label classification techniques." Pattern Recognition 56 (2016): 170-183.
[3] Steinberg, E., and Liu, P. J. "Using ontologies to improve performance in massively multi-label prediction models." arXiv preprint arXiv:1905.12126 (2019).
[4] Makrodimitris, S. et al. "Improving protein function prediction using protein sequence and GO-term similarities." Bioinformatics 35.7 (2019): 1116-1124.

Name and workplace of bachelor’s thesis supervisor:

doc. Ing. Jiří Kléma, Ph.D. Intelligent Data Analysis FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **20.01.2022** Deadline for bachelor thesis submission: **20.05.2022**

Assignment valid until: **30.09.2023**

doc. Ing. Jiří Kléma, Ph.D.	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. _____ Date of assignment receipt

_____ Student's signature

Acknowledgements

Importantly, I would like to thank doc. Ing. Jiří Kléma, Ph.D., for his advice and help through all the stages of the project. Without his guidance the completition would not be possible.

I am grateful to my family and friends throughout for their support and motivation throughout my studies at the Czech Technical University in Prague to which I am grateful for the provided knowledge.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 20 May 2022 .....

Signature

Abstract

Protein function prediction using automatic classifiers may precede the actual function assignment. This thesis describes the hierarchical extension of the BLAST-KNN algorithm, which assigns gene ontology terms to proteins based on the amino acid sequences. The extension consists of the annotation of hierarchical ancestors of the terms and subsequent classification using a modified TPR algorithm. The results of the annotation by the terms from the biological process and molecular function ontologies are compared through precision and recall with the real annotations.

Keywords: protein, gene ontology, annotation

Supervisor: doc. Ing. Jiří Kléma, Ph.D. Intelligent Data Analysis FEE

Abstrakt

Priradzovaniu funkcie bielkovinám môže predchádzať predpovedanie ich funkcie pomocou automatických klasifikátorov. Táto práca sa venuje hierarchickej nadstavbe algoritmu BLAST-KNN, ktorý priradzuje bielkovinám atribúty z génovej ontológie na základe sekvencie aminokyselín. Nadstavba spočíva v ohodnotení hierarchických predkov atribútov a následnej klasifikácii pomocou modifikovaného algoritmu TPR. Anotácie sú vyhodnotené na ontológiách biologických procesov a molekulárnych funkcií pomocou precíznosti a senzitivity voči reálnym anotáciam.

Kľúčové slová: bielkovina, génová ontológia, anotácia

Preklad názvu: Automatická predikcia funkcie bielkovín ako hierarchická multi-label klasifikácia

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Aim of the Bachelor Thesis 1
1.3 Overview 1

2 Theoretical Part **3**
2.1 Protein Functions 3
2.2 Gene Ontology Annotation 3
2.3 Gene Ontology Graph 4
2.4 Automatic Function Prediction . . 5
2.4.1 Non-Hierarchical AFP 5
2.4.2 Hierarchical AFP 5

3 Proposed Solution **7**
3.1 BLAST-KNN 7
3.1.1 BLAST 7
3.1.2 KNN 8
3.2 Hierarchy Transformation 8
3.2.1 Minimal and Maximal Distance from the Root 8
3.2.2 Upward Propagation 9
3.3 Hierarchical Classifier 9
3.3.1 True Path Rule 9
3.3.2 Modification of TPR 11
3.4 Implementation 11

4 Evaluation **13**
4.1 Leave-One-Out Cross-Validation 14
4.2 Ancestral Cuts 14
4.3 Precision and Recall 14
4.4 F_1-Score 15
4.5 Parameter Settings 15
4.5.1 Threshold 15
4.5.2 Multiple Children Bonus 17
4.5.3 Root Distance 18
4.6 Classification Results 20

5 Conclusion **21**
5.1 Possible Improvements 21

Bibliography **23**

Figures

2.1 BPO Hierarchy 4

4.1 Precision-recall curve iterating threshold 16
4.2 Precision-recall curve iterating bonus for multiple children 18
4.3 Precision-recall curve iterating bonus for higher distance from root 19

5.1 Children nodes 22

Tables

4.1 Max F_1-score at different thresholds. 16
4.2 Max F_1-score at different MCB values. 17
4.3 Max F_1-score at different root distance values. 18
4.4 Max F_1-score by different classifiers. 20

Chapter 1
Introduction

1.1 Motivation

In fact, the benefit of the genetics department in medicine is evident. Gene scientists sequence genes more dynamically, but the function annotation of the protein sequences through expensive experiments is unable to keep pace with the sequencing [25]. Moreover, the prediction of protein function may be automated and thus help the classification of the protein sequences. Automatic function prediction assigns more labels to an input protein. The task of improving multi-label classifiers is challenging and yet encouraging for constant modifications and innovations.

Besides the compelling assignment, the possible contribution to various disease treatments may be valuable in the medical field. The COVID-19 pandemic emphasized the suffering of the many and highlighted the need for research. The annotations of genes may feasibly help in the research field and consequently in the disease response during health crises [2]. The positive impact of automatic protein function prediction may not be straightforward, but it indeed contributes to the research.

1.2 Aim of the Bachelor Thesis

The intention of this thesis is to analyze the hierarchical multi-label classification in the protein function prediction domain. This work delivers a new hierarchical approach to the algorithm which labels the protein sequences with the gene ontology terms. Finally, the capability of the modified hierarchical method is compared to the non-hierarchical methods.

1.3 Overview

The bachelor thesis consists of the three main parts:

- Theoretical Part

  The first chapter reports gene ontology graph structure as well as the automated function prediction algorithms. These algorithms include

non-hierarchical such as BLAST-KNN and PRIORS and also hierarchical for example true path rule algorithm.

- Proposed Solution

  The second chapter contains the BLAST-KNN algorithm whose output is then propagated within the gene ontology graph. The true path rule algorithm is further adjusted for the ontology graph traversal and is chained to the previous algorithm.

- Evaluation

  The final part contains the evaluation method through the leave-one-out cross-validation as well as the trimming of the higher nodes of the hierarchy. The ranking of the classifiers is done by the F_1-score which is calculated from the precision and recall. Different parameter settings are evaluated too. The setting of the resulting parameters is further verified through a reevaluation of an alike sample of proteins.

Chapter 2

Theoretical Part

"Traditional classification tasks deal with assigning instances to a single label" [1]. In contrast, multi-label classification obtains various labels for a given instance versus the single information assignment. The assignment of a function to a protein sequence can be automated by automatic function prediction (AFP). Critical Assessment of Function Annotations (CAFA) is a competition that focuses on the AFP of proteins. Predominantly, protein sequences are used as an input feature to the AFP methods, but less common features include protein-protein interaction or gene expression [12].

2.1 Protein Functions

The function prediction of protein sequences is beneficial for the medical and pharmacological industries. AFP can positively affect the diagnosis of diseases, new drug development, and also improvements in the treatments [17]. "Biologists strive to understand the function of a protein" [6]. Even though the predictions need to be verified through scientific experiments, the predictions are helpful for the experiment since they can save a lot of time and energy.

2.2 Gene Ontology Annotation

Gene Ontology (GO) is standardized and widely used database of functions annotating the proteins [27]. The GO database consists of three different ontologies: Molecular Function Ontology (MFO), Biological Process Ontology (BPO), and Cellular Component Ontology (CCO). Each protein can be marked with multiple labels from the three ontologies, but CCO is not used at CAFA meetings and we will focus on the other two ontologies too [8].

The AFP task of assigning GO terms to a protein can be computationally intensive since a single protein sequence can have associated many labels. "Human protein is labeled by around 71 GO terms on average" [3]. The number of labels per protein may quite vary and the number of proteins per label too. The majority of GO terms label only a small number of sequences, and less than 1% of MFO terms label more than 50 protein sequences [27].

2.3 Gene Ontology Graph

Biological Process Ontology and Molecular Function Ontology form oriented acyclic graphs with their terms as nodes of the graph and relations such as is a, part of, has part, etc. as the oriented edges [19]. These graphs are similar to the tree structures, although a node may have multiple parents. Starting from the leaves each parent level provides less specific information congregating in the broadest term like BPO or MFO [8]. By neglecting some relations edges of the acyclic ontology graph, it can be transformed into the tree graph [23].

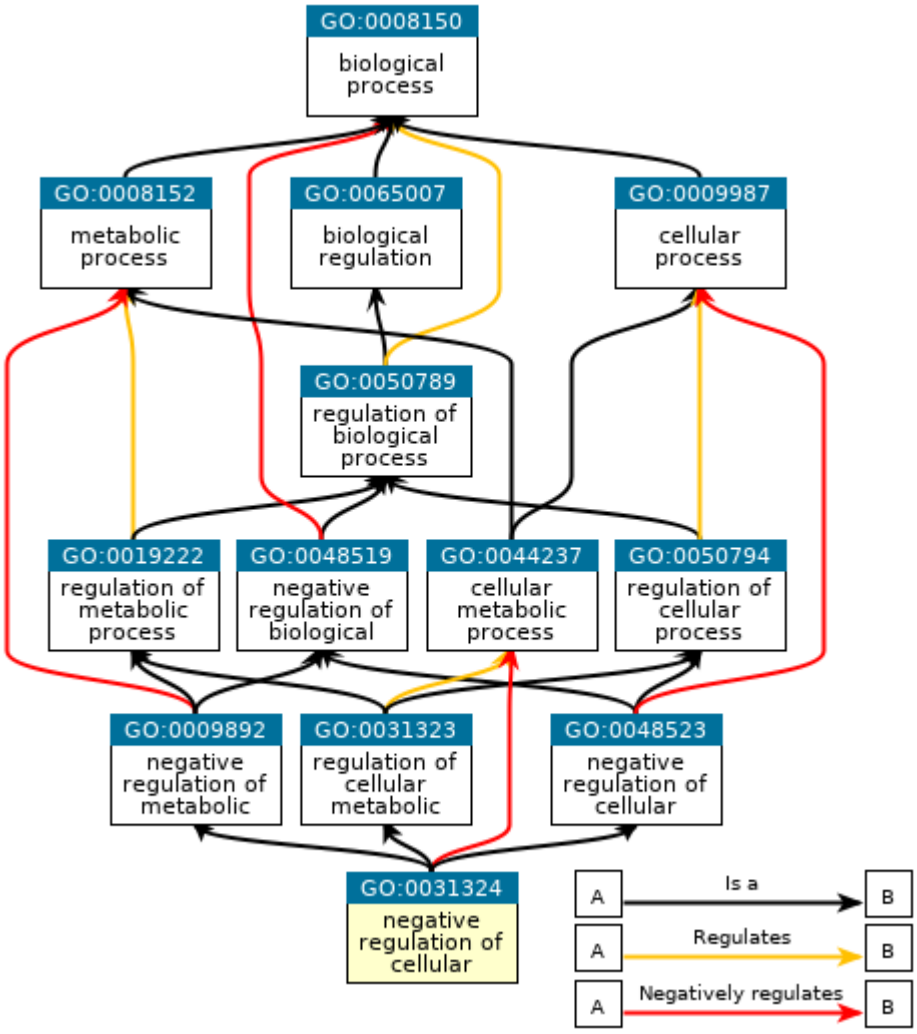


Figure 2.1: BPO Hierarchy [13]

2.4 Automatic Function Prediction

Automatic function prediction annotates proteins with multiple terms from the GO database. For many years it has been a growing field relevant to the protein sequences of numerous species [12]. With the respect to usage of the ontology graph, the AFP classifiers can be divided into non-hierarchical classifiers and their hierarchical counterparts.

2.4.1 Non-Hierarchical AFP

Non-hierarchical classification is independent of the GO graph, but it can be used as an input to the hierarchical classification. The standard AFP algorithms include PRIORS and BLAST-KNN [27]. PRIORS annotates every input sequence with the same most occurring labels. BLAST-KNN takes the similarities of other sequences into account and then assigns the most probable GO terms from the similar proteins. The incorrect outcomes of the BLAST-KNN are often positioned very close to the correct term within the ontology graph [6]. Both algorithms are non-hierarchical, and so they do not consider the position of the GO labels within the graph.

2.4.2 Hierarchical AFP

Hierarchical multi-label classification (HMC) is a classification task working with the hierarchy of the provided ontology. It cannot only result in the leaf terms but also the nodes higher in the hierarchy [19]. Specific examples of the HMC algorithms include hierarchical loss (H-loss) [4], normalized H-loss [19], top-down classifier [5], and the true path rule (TPR) [21].

H-loss and the normalized version function on the propagation principle. The root node is scored with the highest value, that would be 1, and every subsequent level of child nodes is assigned a coefficient divided by 2 [24]. For the normalized variant, the quotient of the parent node value and the number of the child nodes is assigned to every child of the given parent node. This practice is applied to the whole hierarchy [19].

The top-down approach iterates from the root too. If a given node does not belong to the classification result, then all the children nodes are excluded from the result too [5]. The true path rule algorithm does respect the descending pruning, moreover, TPR comprises upward progress of positive classification [21]. The propagation of both positive and negative decisions by TPR causes an enhanced balance of precision and recall [22].

Chapter 3

Proposed Solution

The proposed solution comes from the established algorithms. Initially, the annotations with corresponding values are computed by BLAST-KNN which is widely used at the CAFA meetings thanks to its performance [12]. The predictions are further used as the input data to the hierarchical true path rule algorithm. The preconditions of TPR are fulfilled to ensure correct computations. Those include ancestral nodes within the GO graph are scored along the leaf nodes, the comparison is able to produce dual outcomes at the leaf nodes and the nodes higher in the hierarchy as well.

3.1 BLAST-KNN

By chaining BLAST and KNN algorithms, the non-hierarchical BLAST-KNN is formed. The algorithm labels each sequence with GO terms according to the terms of similar sequences. The similar sequences are retrieved by NCBI BLAST which accepts proteins.

3.1.1 BLAST

Basic local alignment search tool (BLAST) is an algorithm that matches the protein sequence to the similar proteins from a given database [18]. BLAST accepts the protein sequence database additionally to the sequence that we try to assign GO labels. The query sequence is compared with the database to rank the proteins according to their local alignment identity [28]. The proteins do not necessarily need to match the whole sequence length with the alignment counterpart protein.

Similar proteins are outputs of the BLAST with corresponding similarity score (bit-score) and expect value (e-value) of matches in a large arbitrary database [10]. The number of returned proteins may vary, but always are only the top results returned, those with the highest bit-score respectively lowest e-value.

3.1.2 KNN

K nearest neighbors (KNN) is an algorithm that matches GO terms to the protein sequence. It is considered particularly simple yet computationally very efficient algorithm [11]. KNN uses the terms of the similar proteins provided by BLAST. The best matches of protein sequences are mapped to the GO terms from the provided annotations. Consequently, the probability score S is computed for the possible gene ontology labels, it is expressed by the equation 3.1.

$$S(G,\ P) = \frac{\sum_{p_i \in\ similar\ proteins(P)} Annotates(G,\ p_i) \times Bit\text{-}score(P,\ p_i)}{\sum_{p_i \in\ similar\ proteins(P)} Bit\text{-}score(P,\ p_i)} \tag{3.1}$$

Probability score S measures how the GO term G is related to the given protein P [26]. The outputs of KNN are BPO and MFO labels with the corresponding scores. The number of the labels differs and it depends on the search of BLAST and the annotations of the BLAST hits. Established score S is not generally higher for the proteins with higher bit-score or lower e-value, but takes into account only the proportion of GO terms annotating the similar proteins as can be seen in equation 3.1.

3.2 Hierarchy Transformation

The proteins annotated with the ontology terms do not include their ancestral nodes, therefore the hierarchical propagation is necessary [8]. Unlike the tree hierarchy, GO nodes may have multiple parent nodes thus multiple different paths between two nodes are possible. Moreover, two paths between two nodes may vary in length since one path may include more nodes and edges than the other one.

If we were to measure the distance from a node to the root, finding a single path does not guarantee that there is no shorter path possible. To save multiple future explorations and comparisons of different paths we will map the hierarchical distribution and use the outcomes in the upcoming computations.

3.2.1 Minimal and Maximal Distance from the Root

Starting from the root we measure the distance from the node by the breadth-first search (BFS) traversal. BFS ensures that the calculated distance is minimal since we respect the orientation of the edges and the sequential traversal. Every child node is explored before the lower successor except the case when the lower successor node was visited from another node and thus was omitted here.

Additionally, the same search may measure the maximal distances from the root as the graph does not contain cycles which would lead to infinite values. The traversal of the graph is analogous to the search for the minimal

distances with the difference in the keeping of the higher values in the visited nodes. The updates result in the visiting of the child nodes again as long as their value can be further increased too.

The minimal and maximal root distances gained from the graph exploration hint at the position within the hierarchy of the considered node as the position is not as evident as in the case of the tree graph.

3.2.2 Upward Propagation

The BLAST-KNN output consists of relatively arbitrary gene terms from the hierarchical point of view. It does not guarantee the ancestral value assignment thus we need to perform the upward propagation from every annotated node to the root.

Starting from the foundation nodes, their parents are initially annotated and consequently are annotated ancestors upwards. In every node, we sum the propagation value gained from the children set G. The propagation value is equivalent to the multiple children bonus variable MCB for every annotated child g except the child which has the highest value assigned. It is described in the equation 3.2.

$$propagation\ value = \begin{cases} g & g = \max(G) \\ MCB & g \neq \max(G) \end{cases}; \quad g \in G \tag{3.2}$$

As demonstrated, only the original nodes from the BLAST-KNN classification and their ancestors are evaluated, and the nodes with no prior annotation or relationship to the annotated terms are not visited during the upward propagation.

3.3 Hierarchical Classifier

The goal of the hierarchical classifier is to improve the classification of the proteins by using the information on the relative position of the annotating terms within the GO graph. It is not designed to reconstruct the whole graph, but to evaluate just the relevant nodes. The relevant nodes are the nodes that annotate the given protein or the nodes that may influence the annotation. Therefore it is supposed to further develop the annotations and yet be computationally adequate.

3.3.1 True Path Rule

The tree graph T represents a simplified GO structure. The tree edges are formed entirely by the is a relations while the other relations are omitted. True path rule iterates through the tree nodes, and make either positive or negative decision. The TPR core principle by Valentini [21] is shown in algorithm 1.

Algorithm 1 True Path Rule (TPR) hierarchical ensemble

Input:

- a test example x
- tree T of the m hierarchical classes
- set of m classifiers (one for each node) each predicting $\hat{p}_i(x), 1 \leq i \leq m$

1: **for all** levels k of T from bottom to top **do**
2: **for all** nodes i at level k **do**
3: **if** i is a leaf **then**
4: $p_i(x) \leftarrow \hat{p}_i(x)$
5: **if** $p_i(x) > t$ **then**
6: $d_i(x) \leftarrow 1$
7: **else**
8: $d_i(x) \leftarrow 0$
9: **end if**
10: **else**
11: $\phi(x) \leftarrow \{j | j \in child(i), d_j(x) = 1\}$
12: $p_i(x) \leftarrow \frac{1}{1+|\phi(x)|}(\hat{p}_i(x) + \sum_{j \in \phi(x)} p_j(x)$
13: **if** $p_i(x) > t$ **then**
14: $d_i(x) \leftarrow 1$
15: **else**
16: $d_i(x) \leftarrow 0$
17: **for all** $j \in$ subtree(i) **do**
18: $d_j(x) \leftarrow 0$
19: **if** $p_j(x) > t$ **then**
20: $p_j(x) \leftarrow t$
21: **end if**
22: **end for**
23: **end if**
24: **end if**
25: **end for**
26: **end for**

Output:

- the ensemble decisions $d_i(x) = \begin{cases} 1 & if\ x\ belongs\ to\ node\ i \\ 0 & otherwise \end{cases}$
- the probabilities $p_i(x)$ that x belongs to the node $i \in T$

The algorithm 1 starts making decisions from the leaves at bottom level (row 3), and continues in the higher levels (row 1). The decision process is based on the comparison of probability p with the threshold t (rows 5, and 13). If the considered node i is a leaf, the $p_i(x)$ value is equal to the predicted value $\hat{p}_i(x)$ (row 4). On the other hand, if the node i does not lay at the bottom level, the $p_i(x)$ value is an arithmetical mean of the predicted value $\hat{p}_i(x)$ of the node i and the positive children nodes (rows 11-12). This ensures positive upward propagation as the children nodes with positive decisions

can only positively correct the mean since their p value was compared before. Analogously, the negative downward propagation is integrated too. Within the concluding for cycle (row 17), the rejection of the node i will suppress the validity of all the descendant nodes j (row 18).

3.3.2 Modification of TPR

We adjusted the hierarchical TPR algorithm in a way to keep the core principles of the algorithm despite the transformation. First of all, the original algorithm worked with the simplified graph. Only the trees have distinguishable levels thanks to the solo parenthood of every node. In the GO acyclic graphs, the abbreviation of the levels in trees is substituted with the maximal distance from the root. The nodes that have the highest maximal distance are considered to be the lowest in the hierarchy, so they are the starting nodes. The propagation continues upwards with the difference as any parental node may come to the order much more later on. This is caused by the parents with distinct root distances. Although the count of the parent nodes varies among the nodes the valid traversal is achievable through the descending order by maximal root distance [9].

Both positive and negative effects of other nodes within the hierarchy are present throughout the value comparison. At the outset, accepted child nodes may add to the worth of their parent, so the parent node may be accepted too. The MCB gives the potential for a positive effect in the case of multiple children, otherwise, the parent node would be propagated consistently with the strongest child. This approach does supplement only the upward propagation. The downward pruning is adopted by the shifting of the threshold. The maximal threshold is present in the root and the nodes below have it lowered by their distance from the root.

$$t \leftarrow t - (min\ root\ distance \times root\ distance\ bonus) \tag{3.3}$$

The descendant nodes of the top node have their threshold reduced by the root distance bonus variable multiplied by their shortest distance from the root as in the assignment 3.3.

Modified TPR does consider ontology terms with lower values thanks to the decreased threshold in the terms further from the node, but it can cut out the whole subtrees too. The subtrees are removed as the threshold rises by getting closer to the root, but there are not enough successors to compensate through the multiple children bonus.

3.4 Implementation

The final algorithm is implemented in Python. Initially, the NCBI BLAST program makes the protein database out of fasta format. The fasta format stores the proteins with corresponding amino-acid sequences. BLAST iterates the query sequences. It accepts fasta format and computes similar proteins

out of the previously formed database. The sequence alignment is time demanding as the computations were divided into multiple parts and took altogether several days. Size of the merged files from the BLAST alignment is nearly 10 GB. Subsequently, the KNN classifier relates GO terms.

GO terms are comprised in the obo files with information such as id, name, namespace, definition, and the relationships between the term and other terms. As we concentrate only on molecular functions and biological processes, we separate out the two ontologies. Furthermore, some provided terms are labeled as obsolete. As potentially disused we further sort the obsolete occurrences out.

The oriented acyclic graph is formed by applying the relations between the ontology terms as edges, and the terms as the nodes of the graph. We work with Networkx graphs which are produced by the Obonet package. Obonet forms the hierarchical graph from the obo file where the database of gene ontology is stored.

After having the hierarchy with edges, we measure the minimal and maximal distance of every node to the root. Then we propagate the terms to their ancestors by taking the maximum value of the children nodes and adding the extra bonus value. The bonus value is a multiplication of newly introduced multiple children variable by the children nodes that had assigned value. After the ancestors of every node was a value assigned, the true path rule can be launched on these nodes starting with the nodes with the highest maximal distance to the root. The TPR algorithm iterates from the leaves and continues on the higher levels of a tree graph. This approach is not applicable to the GO graph without reducing the structure of the tree. We kept the GO graph with all the relations between the terms and rather adjusted the traversal of TPR. It starts from the nodes with the highest maximal distance to the root and continues with the nodes that have lower and lower maximal root distance. This restrains the access of a node before access to any parent of the node.

Finally, TPR is modified in order to assure the negative descent of the nodes to their children nodes. The negative propagation is a feature of TPR as well as the top-down algorithm. We substitute the stable threshold for a new adjusting one. An additional variable is proposed which is multiplied by the root distance. The root distance is derived from the calculations of minimal distance to the root of the given node. The product is consequently subtracted from the prior threshold.

The tuning of the parameters is computed for molecular function ontology and biological process ontology and roughly takes around 15 hours of computations as 3 variables are iterated through 5 different values.

Chapter 4

Evaluation

The Critical Assessment of Functional Annotation is a protein function prediction challenge that presents annotated sequences [14]. During the competition, participants are given the mapping function of known annotations and extra sequences to be classified. We use cross-validation to test our classifier and to tune the parameters. The CAFA database contains 47,314 proteins with biological process annotation and 32,272 proteins with molecular function annotation. For each ontology, we randomly chose and annotated 100 sequences using leave-one-out cross-validation in order to set the variables, and consequently another 100 as a control sample.

We compared the results with the actual annotations by calculation precision and recall and consequently the F_1-score. The hierarchical annotation performed better than the non-hierarchical, although the improvement was modest. Even though the quantified F measure may not seem to signify the improvement, hierarchical methods have a tendency to predict more specific terms while non-hierarchical methods annotate with less informative predictions [11]. The modified true path rule algorithm evaluates leaves differently than the nodes higher within the hierarchy. Lower levels have a higher acceptance interval, moreover, the densely concentrated nodes reward the subtree despite the prior score of the nodes might have been lower than the threshold at the higher levels. These clusters are not pruned with their subtrees but may lower the precision value despite increasing the recall. For instance, the real annotation may contain only a single node that is not assigned a higher value than the threshold by BLAST-KNN but may be returned as a result by the hierarchical annotator. This approach may result in the more incorrect classifications as the neighboring nodes from the cluster are marked positively too. Alternatively, the parent node may be correctly annotated, but would not count as success as the ancestral annotations are not scored with any points. The evaluation system is not empowered to rank the ancestral nodes [11], yet labeling the parent node does not imply it does not relate to the child node but may be beneficial because the child term may be experimentally annotated later [6].

4.1 Leave-One-Out Cross-Validation

The evaluation is calculated by the leave-one-out cross-validation when we take the whole database of protein sequences and the corresponding labels without a single sequence, and the AFP validation is computed. The leave-one-out cross-validation is methodically accurate in the protein function domain [7]. It does not only take the annotation about the query protein into account but in the neighbor search, the relevant proteins are guaranteed to be found. In contrast, separation of the larger test fold may lead to neglecting similar proteins and bias.

4.2 Ancestral Cuts

The annotations of the modified TPR algorithm may actually quite differ from the original non-hierarchical annotation. This is caused by the hierarchical propagation and consequently considering a much larger number of the GO terms. The ancestors were inserted into the annotations. Even though the ancestral annotations are not technically incorrectly annotated because a term is annotated by all its parents [15], we extract only the original terms and disregard the upper terms in the graph for a better evaluation. The comparison is evaluated with respect to the non-hierarchical BLAST-KNN results and the provided annotations as the control data.

The leaf measurements are judged at the CAFA meetings while the revision data are supported by experimental annotations [8]. Keeping only the labels from the lower levels of the hierarchy does not imply the error in the higher levels, but convenience in the evaluation as the ancestral information can be retrieved from the ontology graphs.

4.3 Precision and Recall

The retrieved results are compared against the real annotations gained from the mapping function. The positive label means that the BPO or MFO term was assigned to the sequence while the negative label the opposite. The classification is marked as true if it matches the provided annotations otherwise it is marked as false. The precision measure is gained by the ratio of the true classifications out of the all the positive classifications as seen below:

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \tag{4.1}$$

Even the low number of annotations may result in high precision as the number of the actual annotations is not considered. It quantifies the quality of the classified data while disregarding the quantity of positive results. In

contrast, the recall measure does take the number of positive terms into account. It compares the number of the positively labeled terms by both annotations to all the annotations gained from the control information as in the formula 4.2.

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{4.2}$$

The size of misclassified is not evaluated in the recall case as was in the precision case. On the contrary the recall takes the amount of the actual GO annotations into consideration.

4.4 F_1-Score

We calculated the F_1-score by the formula 4.3 which is also used in the Critical Assessment of Function Annotations meetings [8]. It aims to exploit the precision and recall while keeping them relatively balanced.

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \tag{4.3}$$

The score peaks as the precision and recall rise equally. The increase in just a single variable offsets the multiplication by the other measure.

Maximization of the F_1-score is an effort to increase the efficacy in the AFP field [20]. It means scoring a relatively high number of true positive annotations while keeping both false positives and negatives relatively low.

4.5 Parameter Settings

As in the CAFA challenge, both ontologies were evaluated. The local classification of the molecular functions scored higher through the F measure although individual precisions and recalls vary.

To achieve the highest F_1-score, different parameters were iterated. These include the multiple children bonus, the bonus for the root distance in the case of HMC, and the threshold t regardless of the hierarchy.

4.5.1 Threshold

The probability score S is paired with the GO terms as the output of BLAST-KNN. To use just the best-ranking values, a threshold filter is introduced. As the threshold decreases, fewer terms are returned and the precision tends to decrease while the recall tends to increase.

The hierarchical equivalent to the score S is adjusted in the ancestral nodes. The sum of the propagation values gained from the annotated children as in

the relation 3.2 is compared to the possibly lowered threshold by the root distance as in the relation 3.3.

The response in F_1-score to the lifting threshold is marked in the table below:

t	hierarchical		non-hierarchical	
	BPO	**MFO**	**BPO**	**MFO**
0.15	**0.2797**	**0.4155**	**0.2771**	**0.4075**
0.20	0.2737	0.3931	0.2657	0.3852
0.25	0.2736	0.3902	0.2496	0.3471
0.30	0.2551	0.3748	0.2502	0.3445
0.35	0.2453	0.3685	0.2102	0.2820

Table 4.1: Max F_1-score at different thresholds.

The table 4.1 displays the maximum of F_1-scores for the given ontology and threshold. The left part of the table shows the hierarchical annotations while t parameter is incremented. On the right side, the non-hierarchical counterpart is shown. Higher the threshold t, the higher the impact of the other two hierarchical variables may be achieved and the left f measure diverges from the BLAST-KNN flat classifier.

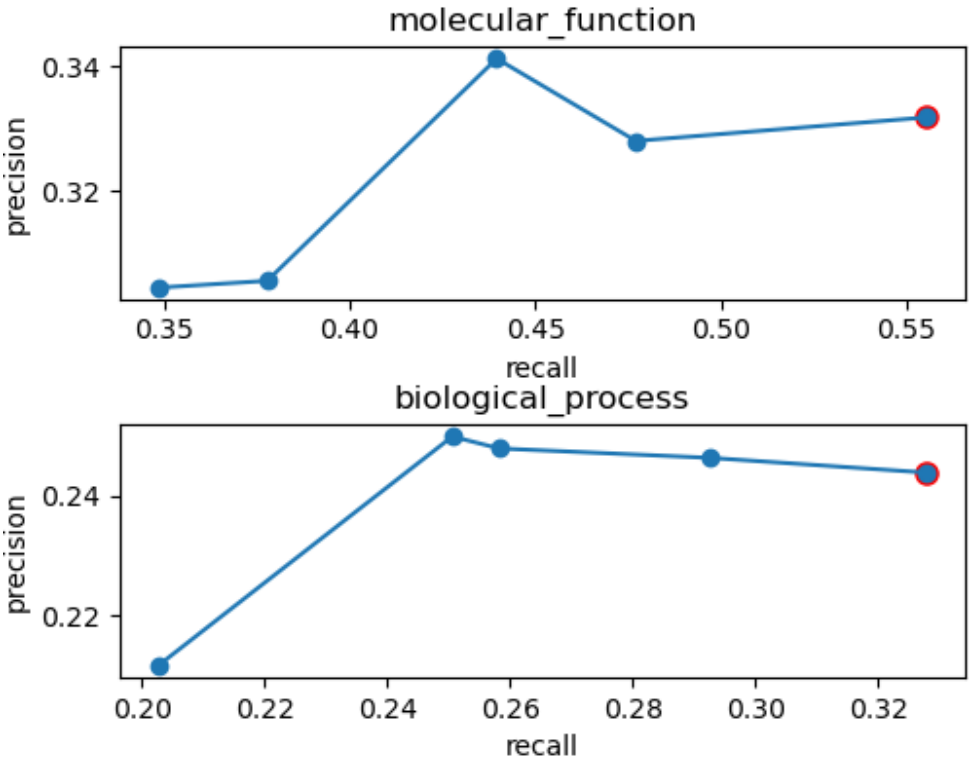


Figure 4.1: Precision-recall curve iterating threshold

As we can see at the precision-recall curves 4.1, as the threshold is lowered the recall increases and leads to the higher F_1-score. The peak in the F measure has a marked edge with a red color. The Multiple children bonus and the bonus for higher distance from the root are fixed at their best scoring annotation by F measure while the threshold iteratively decreases from 0.35 to the value of 0.15 at the right-most point.

The initial decrease of the threshold has a positive effect on the precision in both ontologies. Further decrease of the threshold negatively effects the precision value yet recall growth offsets the not so sharp fall in the precision and the F measure peaks at the right-most point with the minimal value of t.

4.5.2 Multiple Children Bonus

The iteration of the newly introduced variable multiple children bonus starts from zero. That means that within the upward hierarchical propagation is the number of positively annotated successors disregarded and only the maximum of children values is taken. This way the pruning of the subtrees is neglected and the impact of the HMC is less significant. As the incrementation of MCB escalates it may overshadow the actual weighs of different nodes and may result in the high values of the clustered terms within a subtree. In between the two extremes, the optimum can be found through the iteration.

In the iterative table 4.2, we can find the highest F_1-score for every value of MCB.

MCB	BPO	MFO
0	0.2771	0.4100
0.01	**0.2797**	0.4148
0.02	0.2797	**0.4155**
0.03	0.2782	0.4114
0.04	0.2782	0.4127

Table 4.2: Max F_1-score at different MCB values.

The top performance for the BPO is at the value of 0.01, respectively at 0.02 for the MFO. Both namespaces show similar values in both peak values while the other values represent a slight drop in the F measure.

If we observe the change in precision and recall in the figure 4.2, we can see the high initial increase of precision by the iteration of the MCB variable as the threshold and root distance bonus are fixed. Before the center value of 0.02, the precision does not fluctuate much, but the follow-up incrementation results in the direct precision drop as well as F_1-score drop.

The recall increases too, but only throughout the first two iterations. The next iteration does not have a significant effect on the recall while the last one does not affect precision either, and it does blend at the graph with the second to last value for the ontologies.

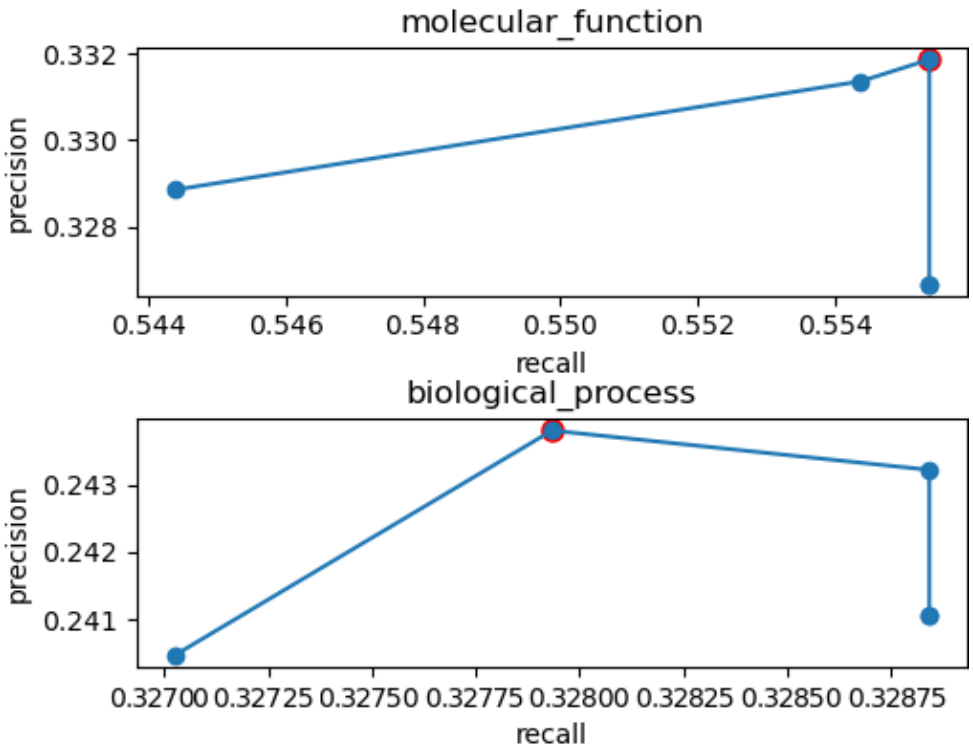


Figure 4.2: Precision-recall curve iterating bonus for multiple children

4.5.3 Root Distance

The root distance variable was established to allow the threshold to be progressive and to cut out the subtree later if needed. As the TPR starts its iteration from the leaf nodes the new variable would increment the threshold value. The variable is multiplied by the minimal number of edges that connect the current node to the root.

The table 4.3 shows the maximum of the F_1-score achieved at the given value of the root distance variable.

root distance	BPO	MFO
0	**0.2797**	0.4127
0.01	0.2755	**0.4155**
0.02	0.2742	0.3972
0.03	0.2736	0.3902
0.04	0.2565	0.3768

Table 4.3: Max F_1-score at different root distance values.

The annotations by biological process terms achieve the highest value at the initial iteration meaning the threshold remains the same for the whole

hierarchy. This may be explained by the low threshold values that would further decrease their weight thus too many annotations would be assigned to the given protein.

On the contrary, the MFO labels favor the slightly increased bonus for the root distance. The best performance is at the 0.01 value meaning the threshold decreases by a single percent for every level further from the root node. Whether the node remains in the set of annotations or is pruned away, the chances of the subtree trimming rise as the algorithm traverses closer to the node.

The F measure tries to balance the precision and recall and as the variable clearly increases the recall, the precision drops and it consequently brings down the F score. We can observe this at the precision-recall curves 4.3 with the maximum of F_1-score marked red. The other two parameters remain fixed at their peak performance while the value of the bonus for the root distance iterates from 0 to 0.04.

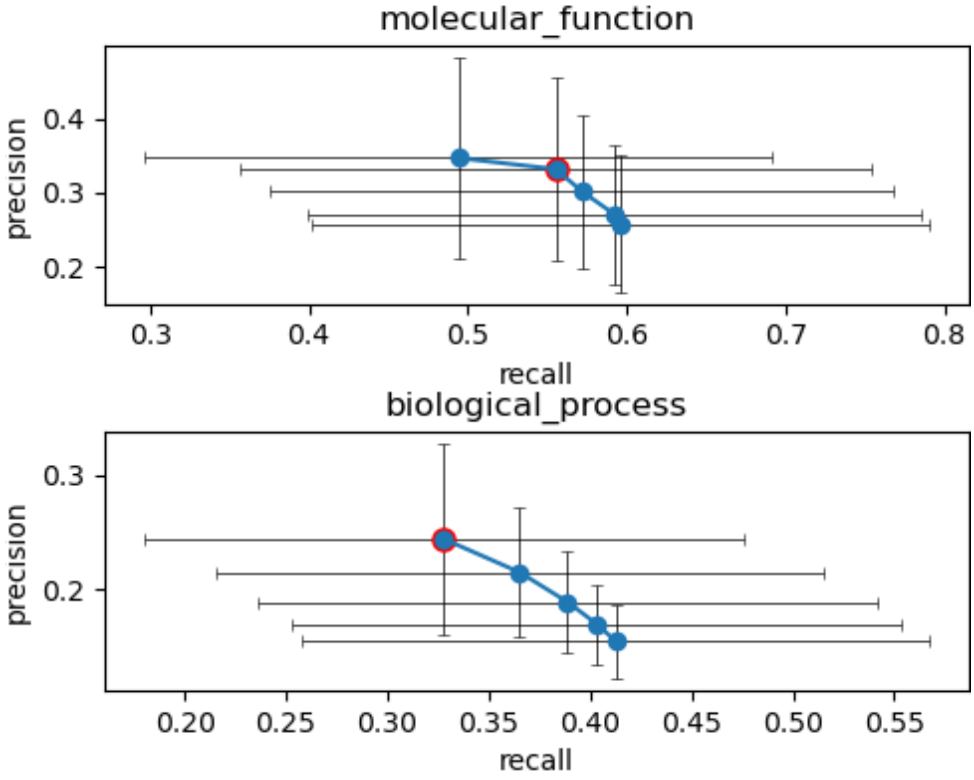


Figure 4.3: Precision-recall curve iterating bonus for higher distance from root

The blue mean precision values decrease and recall values increase as the root distance variable is incremented in the five iterations. The variance marked with the narrow black lines remains relatively high for the recall and quite stable for the precision with decreasing trend in the BPO graph.

The initial precision of the biological process annotation was already low and the increase in the recall is not large enough to balance out the precision

descent. Unlike the process annotation, the sharp improvement of recall still boosts the F measure in the case of molecular function. Further growth in the number of annotations is not sustainable and results in the F_1-score downturn.

4.6 Classification Results

Altogether, the resulting modified true path rule algorithm on the propagated annotations of BLAST-KNN within the GO hierarchy was compared with the non-hierarchical algorithms of PRIORS and BLAST-KNN. The comparison comprises of a random sample of 100 protein sequences and a subsequent sample of 100 protein sequences for validation of the results. The PRIORS method does not score too high in efficacy as it labels all the proteins with the same most-frequent terms. The frequent reference point of the BLAST-KNN clearly outperformed the less sophisticated classifier in both domains [16].

	first sample		control sample	
classifier	**BPO**	**MFO**	**BPO**	**MFO**
PRIORS	0.0452	0.0808	0.0423	0.0530
BLAST-KNN	0.2771	0.4075	0.2004	0.3780
modified TPR	0.2797	0.4155	0.2004	0.3827

Table 4.4: Max F_1-score by different classifiers.

As the table 4.4 suggests, the BPO classifications were less efficient than the MFO classifications regardless of the use of the hierarchy. Even though the first sample of 100 proteins showed minimal improvement by the hierarchical classification of the biological process labels, the control sample did not come to the same conclusion. The second sample produced the same results for BLAST-KNN and the modified TPR with the variables set to the values from the previous sample. The multiple children bonus variable was fixed at the value of 0.01 and the root distance bonus at the value of 0. The BPO precision and recall did not change throughout the control computations thus the F_1-score remained the same even after the computations by the hierarchical algorithm.

Nevertheless, the F_1-score computed from the precision and recall product for the molecular function hints the improvement although it is not vast. The hierarchical approach improved the F measure in both samples. The reason behind the better performance in the molecular function domain may be caused by the higher values of the parameters. MBC was set to 0.02 and the bonus for the distance from the root node was set to 0.01 during the iterations on the first sample. These values were higher than during the BPO control computations and led to an opportunity for a hierarchical adjustment in annotations. The verification calculations support the assumption that the modified TPR improves the F_1-score, but the scale may not be the same as in the initial calculations.

Chapter 5

Conclusion

In summary, the multi-label classification for the automated function prediction indicates the possible annotations of proteins by the gene ontology terms. The thesis illustrates the mechanisms of these classifiers. In particular, it explains how the hierarchical relationships between ontology terms help to maintain the individual annotations consistent and also increase the annotation accuracy.

A new approach to the true path rule algorithm is proposed. The raw probabilities that score assignments among proteins and gene ontology terms are derived from BLAST-KNN, and they serve as inputs to the algorithm. Their values are propagated to their ancestral terms on the maximum principle with the adjustment for multiple children nodes. During the comparison, the modified TPR contains a variable threshold that can promote the cut out of the nodes if they do not accumulate large enough value from the descendant nodes.

The annotations for every AFP of the three prediction methods are contrasted with the real annotations. PRIORS was by far the least accurate out of them as it assigns the most straightforward labels. BLAST-KNN and the modified hierarchical method produce similarly accurate predictions. The improvement by the hierarchical TPR algorithm is not evident for the BPO as the hierarchical distribution is not measurable by the standard approaches like precision and recall. In the case of MFO, the improvement was observable through the computation of the F_1-score, the annotations have to be propagated downwards and restricted to the annotations without their ancestral terms that do indeed validly annotate.

5.1 Possible Improvements

Including the broader variety of information per protein may help us better understand the proteins and potentially improve the protein function prediction. BLAST computes the sequence alignment but input features such as protein-protein interactions may add later weight to the annotation decisions. Furthermore, some data have a really low number of annotations and thus may not be particularly relevant and may skew the results. Annotating with more input features on the more informative proteins may have a positive

effect on the result.

A part of the BPO graph is shown in figure 5.1 with the term highlighted in yellow having four child terms. If the outputs of BLAST-KNN are three terms of GO:1901632 through GO:1901634 with all the values just below the threshold, but only single of them truly annotates the protein, the modified TPR may pick all as positive. On the other hand, the non-hierarchical approach may easily discard all as they do not reach the threshold score. The standard evaluation measures such as precision and recall do not cope with the issue well. The possible improvement in the annotation scoring may be with the focus on the position within the hierarchy.

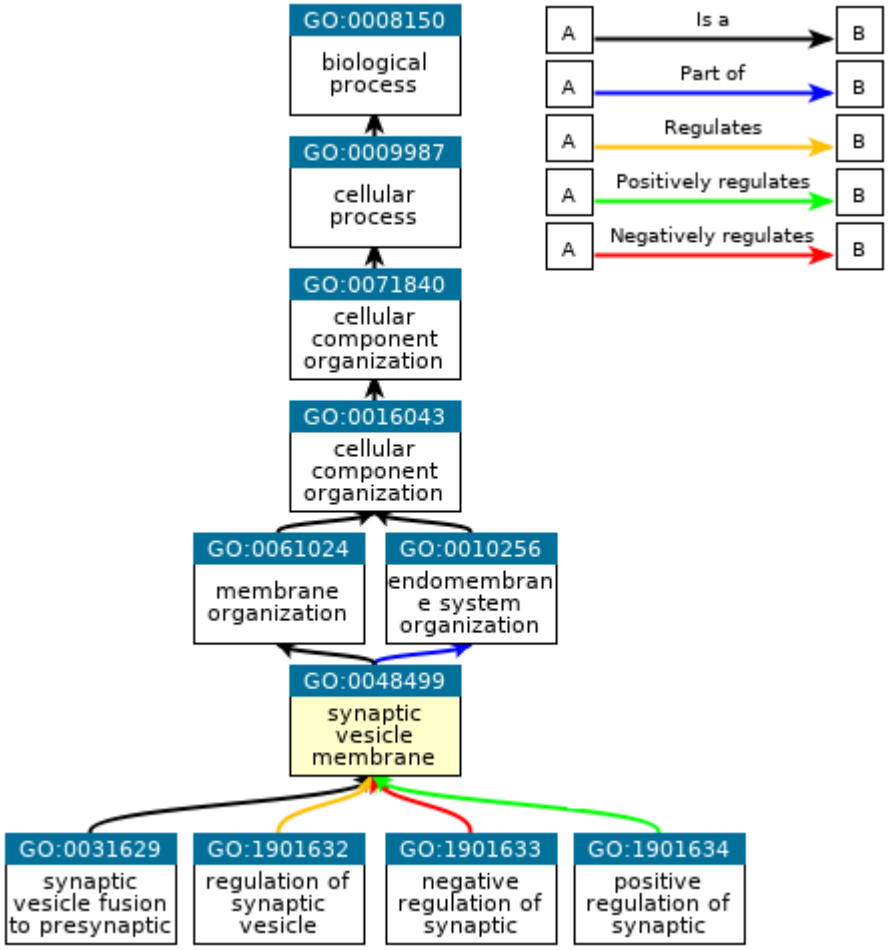


Figure 5.1: Children nodes [13]

Bibliography

[1] Noor Alaydie, Chandan K Reddy, and Farshad Fotouhi. Exploiting label dependency for hierarchical multi-label classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 294–305. Springer, 2012.

[2] Shane Babcock, John Beverley, Lindsay G Cowell, and Barry Smith. The infectious disease ontology in the age of covid-19. *Journal of biomedical semantics*, 12(1):1–20, 2021.

[3] Emmanuel Boutet, Damien Lieberherr, Michael Tognolli, Michel Schneider, Parit Bansal, Alan J Bridge, Sylvain Poux, Lydie Bougueleret, and Ioannis Xenarios. Uniprotkb/swiss-prot, the manually annotated section of the uniprot knowledgebase: how to use the entry view. In *Plant Bioinformatics*, pages 23–54. Springer, 2016.

[4] Nicolo Cesa-Bianchi, Claudio Gentile, Andrea Tironi, and Luca Zaniboni. Incremental algorithms for hierarchical classification. *Advances in neural information processing systems*, 17, 2004.

[5] Liangxi Cheng, Hongfei Lin, Yuncui Hu, Jian Wang, and Zhihao Yang. Gene function prediction based on the gene ontology hierarchical structure. *PLoS One*, 9(9):e107187, 2014.

[6] Roman Eisner, Brett Poulin, Duane Szafron, Paul Lu, and Russell Greiner. Improving protein function prediction using the hierarchical structure of the gene ontology. In *2005 IEEE symposium on computational intelligence in bioinformatics and computational biology*, pages 1–10. IEEE, 2005.

[7] Jian Guo, Xian Pu, Yuanlie Lin, and Howard Leung. Protein subcellular localization based on psi-blast and machine learning. *Journal of Bioinformatics and Computational Biology*, 4(06):1181–1195, 2006.

[8] Tobias Hamp, Rebecca Kassner, Stefan Seemayer, Esmeralda Vicedo, Christian Schaefer, Dominik Achten, Florian Auer, Ariane Boehm, Tatjana Braun, Maximilian Hecht, et al. Homology-based inference sets the bar high for protein function prediction. In *BMC bioinformatics*, volume 14, pages 1–10. Springer, 2013.

[9] DV Klopfenstein, Liangsheng Zhang, Brent S Pedersen, Fidel Ramírez, Alex Warwick Vesztrocy, Aurélien Naldi, Christopher J Mungall, Jeffrey M Yunes, Olga Botvinnik, Mark Weigel, et al. Goatools: A python library for gene ontology analyses. *Scientific reports*, 8(1):1–17, 2018.

[10] I Lobo. Basic local alignment search tool (blast). *Nat Educ*, 1(1), 2008.

[11] Stavros Makrodimitris, Roeland CHJ van Ham, and Marcel JT Reinders. Improving protein function prediction using protein sequence and go-term similarities. *Bioinformatics*, 35(7):1116–1124, 2019.

[12] Stavros Makrodimitris, Roeland CHJ Van Ham, and Marcel JT Reinders. Automatic gene function prediction in the 2020's. *Genes*, 11(11):1264, 2020.

[13] QuickGO. Ancestor chart for GO:0031324. `https://www.ebi.ac.uk/QuickGO/term/GO:0031324`, 2022. Online; accessed 11 May 2022.

[14] Predrag Radivojac. A (not so) quick introduction to protein function prediction. *Indiana University, USA*, 2013.

[15] Seyyede Fatemeh Seyyedsalehi, Mahdieh Soleymani, Hamid R Rabiee, and Mohammad RK Mofrad. Pfp-wgan: Protein function prediction by discovering gene ontology term correlations with generative adversarial networks. *PloS one*, 16(2):e0244430, 2021.

[16] Amir Shanehsazzadeh, David Belanger, and David Dohan. Fixed-length protein embeddings using contextual lenses. *arXiv preprint arXiv:2010.15065*, 2020.

[17] Freitas Silla. Silla cn, freitas aa. *A survey of hierarchical classification across different application domains, Data Mining and Knowledge Discovery*, 22(1):31–72, 2011.

[18] Ethan Steinberg and Peter J Liu. Using ontologies to improve performance in massively multi-label prediction models. *arXiv preprint arXiv:1905.12126*, 2019.

[19] Isaac Triguero and Celine Vens. Labelling strategies for hierarchical multi-label classification techniques. *Pattern Recognition*, 56:170–183, 2016.

[20] Muhammad Usman, Shujaat Khan, Seongyong Park, and Abdul Wahab. Afp-src: identification of antifreeze proteins using sparse representation classifier. *Neural Computing and Applications*, 34(3):2275–2285, 2022.

[21] Giorgio Valentini. True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 8:832–47, 05 2011.

[22] Giorgio Valentini. Hierarchical ensemble methods for protein function prediction. *International Scholarly Research Notices*, 2014, 2014.

[23] Cen Wan and Alex A Freitas. Hierarchical dependency constrained tree augmented naive bayes classifiers for hierarchical feature spaces. *arXiv preprint arXiv:2202.04105*, 2022.

[24] Cinna Wu, Mark Tygert, and Yann LeCun. A hierarchical loss and its problems when classifying non-hierarchically. *Plos one*, 14(12):e0226222, 2019.

[25] Shuwei Yao, Ronghui You, Shaojun Wang, Yi Xiong, Xiaodi Huang, and Shanfeng Zhu. 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, 2021.

[26] Ronghui You, Shuwei Yao, Yi Xiong, Xiaodi Huang, Fengzhu Sun, Hiroshi Mamitsuka, and Shanfeng Zhu. NetGO: improving large-scale protein function prediction with massive network information. *Nucleic Acids Research*, 47(W1):W379–W387, 05 2019.

[27] Ronghui You, Zihan Zhang, Yi Xiong, Fengzhu Sun, Hiroshi Mamitsuka, and Shanfeng Zhu. Golabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics*, 34(14):2465–2473, 2018.

[28] Naihui Zhou, Yuxiang Jiang, Timothy R Bergquist, Alexandra J Lee, Balint Z Kacsoh, Alex W Crocker, Kimberley A Lewis, George Georghiou, Huy N Nguyen, Md Nafiz Hamid, et al. 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, 2019.