Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Manuscript Category

Subject Section

Protein function prediction using Gene Ontology terms

Jonas Schäfer[1,*], Michael Heinzinger[1] and Burkhard Rost[1]

[1]Department of Informatics, Technical University of Munich, Boltzmannstr. 3, 85748 Garching

*To whom correspondence should be addressed.

Received on 31.01.2020;

ABSTRACT

Proteins have an important role in living organisms. They catalyze reactions, transport molecules and act as messengers for information. In drug development and drug repurposing, knowing the multiplicity of functions of a protein can prevent side effects.

We predicted protein functions according to the Gene Ontology (GO), a standardized vocabulary to describe the function of genes and gene products as a hierarchical acyclic graph. As features, we used Sequence-to-Vector embeddings (SeqVec), a 1024 long vector that uses Language Processing on the sequence. In this paper, we compare a flat approach that predicts all classes independently, a hierarchical approach that starts at the top of GO and a Path Propagation Algorithm, which is a flat approach but propagates all probabilities until the top of the graph. The algorithms are compared on a data set consisting of 1300 proteins.

The balanced accuracy from the Path Propagation, which is the best of the three algorithms is 0.71 ± 0.14. The F1-score of Path Propagation is 0.4 ± 0.23.

Contact: jonas.schaefer@tum.de

1 Introduction

Gene Ontology consists of three ontologies: Molecular Function (MF), Biological Process and Cellular Component. These Ontologies offer a standardized vocabulary to describe Genes and Gene Products. Each of the three Ontologies is a hierarchical directed acyclic graph. In an ontology, terms have a relationship with each other. Exemplary relationships are “is a”, “part of “or “regulates”. At the top of the graph are general terms, whereas lower in the graph are specialized terms. Every term does also belong to all of his parents. Therefore a gene or gene product gets assigned to the most specific term. An annotation in the GO also gets an evidence code assigned that states where this annotation comes from, e.g. whether the annotation is experimentally, computationally or phylogenetically inferred. In the Full GO, there are connections between the three separate ontologies, but most tools rely on the filtered version, where there exist three separate ontologies without connections between each other [1, 2].

The lowering cost in high throughput sequencing led to an explosion in data size. In the TrEMBL database from Uniprot, there are 180,000,000 not manually curated proteins as of January 2020. SwissProt, which is the manually curated part of Uniprot, has 560,000 proteins. 47,000 proteins in SwissProt have a total of 80,000 experimental MF annotations [3]. A tool that can automatically infer annotations is needed, as databases will grow even more in the coming years and manual curation can not keep up. The importance of this issue even led to a competition that aims at motivating groups to work on protein function prediction. This competition is called the Critical Assessment of protein Function Annotation algorithms (CAFA). A large number of protein sequences without annotations are provided. After some time, some of these sequences will be experimentally verified and these are the sequences for which the tools, send in by researchers, will be validated [8].

Protein function prediction is a difficult task as it consists of two issues: First, there is no precise definition of what a function class is and the second issue is that there are many ways to represent proteins, but there is none that is superior above all. The first problem is partly solved with the GO. Only some classes of the GO need to be selected as there are more than 6000 MFs. For the second task, there are many possibilities to represent amino acid (AA) chains. ProtFun compares the similarity of AA chains where it is known they are signal peptides and transmembrane helices [4, 5]. FFPred3 is a state-of-the-art MF prediction tool using Gene Ontology. The tool scans proteins against SVM for biophysical attributes. While researchers trained the tool on human proteins, it also can infer annotations for other eukaryotic organisms [6].

Function prediction with proteins belongs to the class of hierarchical multi-label classification tasks, as each protein can have multiple functions annotated. We propose a new tool to solve this hierarchical multi-label classification issue. This tool uses SeqVec, a language model [7]. SeqVec learns the chain of AAs, similar to how ELMo learns the structure of sentences. On these features, we compare three different algorithms that are trained equally but predict positives differently: Flat Classification, Hierarchical Classification and Path Propagation Classification. Flat Classification predicts whether a protein belongs to the 38

classes. We developed classifiers for Hierarchical Classification only predicts if one of the immediate parent nodes is positive and Path Propagation, multiplies all probabilities of all parents together and assigns positives according to a threshold.

2 Methods

2.1 Data set

For our data set, we used Swissprot, the manually curated part of Uniprot, downloaded on 19. August 2019. We downloaded the GO in the OBO file format on 19. August 2019. We selected all proteins with MF annotations and a maximum length of 1002 AAs without ambiguous Aas (unknown or multiple possible AAs for a position). To predict the function, we are only looking at the MF experimental evidence codes (EXP, IDA, IPI, IMP, IGI, IEP, TAS and IC), ignoring CC and BP [9]. To avoid having too similar proteins in our training and testing set, we performed a homology reduction with a sequence identity of 50% on the data set before splitting it into the four parts [10]. The data set now contains 21,000 proteins and 5,300 different MF GO terms.

2.2 Features

To represent the proteins in our data set, we used SeqVec. SeqVec uses the bidirectional language model ELMo to convert sequences to matrices of size 1024 x L. L is the length of this protein. To get the same length for all our features, we averaged SeqVec for each protein and ended up with a vector of size 1024 [7].

2.3 Training

We split the data set into four parts: Training set (60 %), Cross-Validation set (30 %), Testing set (7 %) and one set to determine independent path probabilities (3 %).

For our classes, we used the top 10 most appearing terms and backpropagated until the term MF, ending up with 38 classes. We went through all proteins and assigned a 1 to a vector of size 38 if the class or one of the children was in the annotation of the protein. Proteins that belonged only to classes which we are not predicting were removed.

2.4 Imbalance

The data set is negatively imbalanced. The smallest class appears 250 times, while the training set has 19,000 proteins. To solve this imbalance, we created 38 different data sets. We assigned each protein as positive if it includes the class we are creating the data set for or is a child of this class. To get the negative proteins, we chose all the proteins that belonged to the parent class but not the current class [10]. In case these data sets where positively imbalanced, we randomly chose negative proteins from the whole data set.

2.5 Baseline

For the baseline, we used our whole training set and just randomly assigned it positive and negative predictions for each of the 38 classes independently, according to their appearance in the original training set (stratified approach).

2.6 Algorithms

We implemented three different models. The first model is a flat classification, the second a hierarchical classification and the third a Path Propagation Algorithm. All algorithms use Random Forest classifiers for all 38 classes with the hyperparameters maximum depth, maximum features and bootstrapping. We used 3-fold stratified cross-validation [11].

The flat Classification predicts all 38 classes for the input separately. It is a binary prediction of whether a protein belongs to this class or not.

The hierarchical Classification starts at the top of the GO with Molecular Function (MF) and from there on looks at all the children of MF and predicts whether a protein belongs to any of the children's classes. If it does belong to that class, the classifier looks at all the children of this class and predicts for all of them again whether the protein belongs to any of these classes. Classification is done for all recursive paths until we either predict one path as negative or reach one of the ten leaf classes.

The Path Propagation Algorithm takes the probability of the current class and multiplies it with the probabilities of all its parents. Deeper classes have a lower probability than higher ones. Therefore 0.5 can not be used as a threshold anymore. To get thresholds for our classifiers, we used the third data set, which was not involved in training, hyperparameter tuning or final testing. We looked at the path probabilities of each class of the True Positives compared to the False Positives and maximized F1-score on this set to get higher precision. Classifiers usually predict a probability and if it is above 0.5, a class will be assigned as positive. Since GO is structured hierarchically, if there is one prediction slightly above 0.5 and in the next deeper level one just above 0.5, it might be because it is, in reality, only an FP.

2.7 Evaluation matrices

We evaluated our models with three different metrics: Balanced accuracy, F1-Score and Matthews Correlation Coefficient (MCC).

$$balanced\,accuracy = \frac{1}{N}\sum_{class}^{N}\frac{TP_{class}}{TP_{class}+FN_{class}}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Table 1. Average scores for the three implemented algorithms

	Flat	Hierarchical	Path Propagation
Balanced Accuracy	0.67 ± 0.12	0.69 ± 0.12	0.71 ± 0.14
F1-score	0.26 ± 0.20	0.33 ± 0.22	0.40 ± 0.23
Matthews Correlation-Coefficient	0.21 ± 0.17	0.29 ± 0.20	0.35 ± 0.23

Predicting average scores for the three algorithms (Flat Classification, Hierarchical Classification and Path Propagation): Balanced Accuracy (average recall for every class), F1-score (weighted average of precision and recall) and Matthews Correlation Coefficient (MCC, a balanced measure that uses true and false positives and negatives). The Balanced Accuracy is very similar in the three algorithms whereas F1-score and MCC differ between the Flat and Path Propagation Classification.

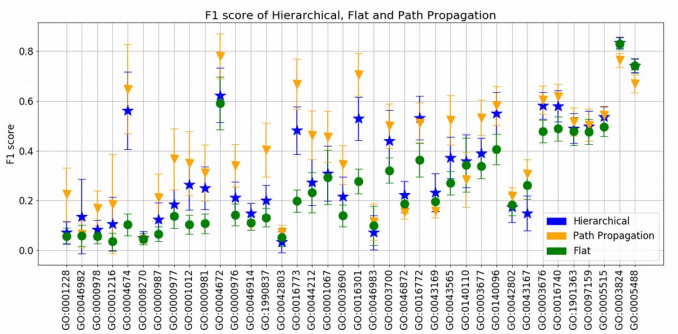

Fig. 1. F1-score with the standard error of every class for the three algorithms (Flat Classification, Hierarchical Classification and Path Propagation) on the test set. The Path Propagation Algorithm is, in most cases, higher than the Hierarchical or Flat Classification, though only some cases are statistically significant. The classes are ordered according to their data set size, increasing from left to right. The Flat Algorithm shows higher F1-scores with increasing size of the training set.

Fig. 2. F1-score for the three algorithms. All classes with the same minimal distance (number of steps until the top-level node Molecular Function is reached) are averaged together. Except for the smallest distance, the Path Propagation algorithm performs better than the other two algorithms. Similarly, the Hierarchical Algorithm reaches higher F1 scores than the Flat Classification. Figure 6 shows the same result with the MCC.

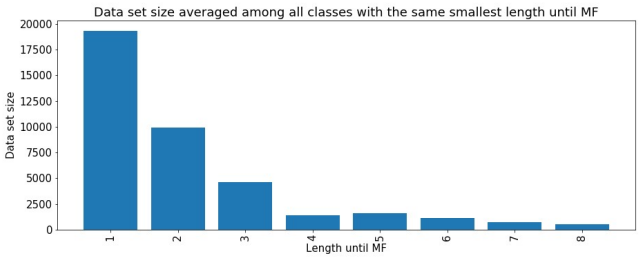

Fig. 3. The size of the data set, used for training, is averaged by the same minimal length until the root term Molecular Function. The further away from the root, a class is, the smaller the data set for this class is since all special classes are also all general classes. The bar of length fife is higher than length four since the Gene Ontology has multiple paths and is not built like a tree.

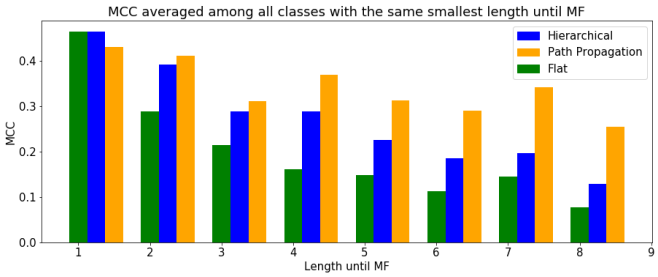

Fig. 6. Matthews Correlation Coefficient for the three algorithms. All classes with the same minimal distance (number of steps until the top-level node Molecular Function is reached) are averaged together. Except for the smallest distance, the Path Propagation algorithm performs better than the other two algorithms. Similarly, the Hierarchical Algorithm reaches higher F1 scores than the Flat Classification. Figure shows the same result with the F1-score.

3 Results and Discussion

3.1 Algorithm Comparison

The Flat Classification has an average balanced accuracy of 0.67 ± 0.12, but since the f1-score is at 0.26 ± 0.20 (Table 1), the model is not very good. The training data of the model is designed to differentiate between a parent class and the current class. The Flat classification tests on the whole test set. The model for class “GO:0003690 double-stranded DNA binding” was trained only on proteins belonging to “GO:0003677 DNA binding” and therefore has difficulties differentiating between “GO:0003690 double-stranded DNA binding” and for example “GO:0003824 catalytic activity”.

Hierarchical Classification solves this, with only testing proteins that were positively assigned to the parent class. The average balanced accuracy is 0.69 ± 0.12, the average F1-score is 0.33 ± 0.22. However, if a class is wrongly positively assigned, the children of this class continue to make mistakes. This is why the Hierarchical algorithm is just better than the Flat Classification.

To tackle this issue, we used the Path Propagation Algorithm. This algorithm multiplies all the probabilities of all parents with the current class and, depending on a threshold, assigns a protein to a class. The threshold is calculated by the third data set. Therefore if the probability is just above 50% more than once, the protein falls under the threshold and will not be predicted positively. This is the reason the Path Propagation Algorithm is the best of the algorithms (Table 1, Figure 1). This algorithm has an average balanced accuracy of 0.71 ± 0.14 and an F1-score of 0.40 ± 0.23.

3.2 Distance to Root

As mistakes accumulate while traversing the GO graph, more detailed annotations are predicted poorer. The Path Propagation does not solve this completely but is better than the Flat or Hierarchical Classification (Figure 2, Figure 6). There are several reasons why mistakes continue to be made. One reason is that we trained the classifiers to differentiate between a parent and a child node. If a protein that does not belong to one branch and is positively assigned once, often the whole branch predicts it as positive, as it never trained on a protein of this class. Besides this, there is also the fact that deeper in the GO, the functions of proteins are similar. The tool LocTree, which predicts cellular location in a hierarchical structure, also has the issue with a falling score, the further away the class is from the root [12]. Another reason for this trend is that with a larger distance from the root, the training data set gets smaller (Figure 3). The reason for this is the fact that every class always belongs to a more general class, besides “GO:0003674 Molecular Function”, which is the most general class.

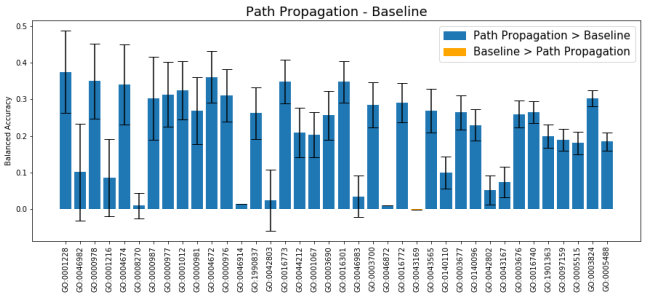

Fig. 4. Balanced Accuracy of the baseline subtracted from the Balanced Accuracy of the Path Propagation Algorithm. The balanced accuracy baseline outperforms some of the terms that had a significantly higher F1-score.

Table 2. F1-scores averaged by Binding, Catalytic and Transcription Regulator Activity

	Binding Activity	Catalytic Activity	Transcription Regulator Activity
Flat	0.23 ± 0.18	0.41 ± 0.22	0.17 ± 0.13
Hierarchical	0.27 ± 0.18	0.59 ± 0.10	0.25 ± 0.14
Path Propagation	0.33 ± 0.18	0.66 ± 0.09	0.31 ± 0.19

Average F1-score of all classes belonging to Binding Activity, Catalytic Activity and Transcription Regulator Activity for the three algorithms (Flat Classification, Hierarchical Classification and Path Propagation). Binding Activity has 25 classes, Catalytic Activity has 8 classes and Transcription Regulator Activity has 5 classes. Between Binding Activity and Transcription Regulator Activity are no statistically significant differences. Catalytic Activity is statistically significantly higher than Binding Activity and Transcription Regulator Activity in the Hierarchical and Path Propagation Algorithm.

3.3 Comparison with Baseline and State-of-the-Art

The best algorithm (Path Propagation) is except for the class GO:0046983 better than the randomly stratified baseline in the F1 measurement. For the balanced accuracy, there are multiple classes that are not statistically significantly better than the baseline (Figure 4, 5, 7). Due to technical difficulties, a state of the art comparison is not possible.

3.4 Differences between all classes belonging to 3 top-level classes

In the Hierarchical and the Path Propagation Algorithm, the classes belonging to "GO:0003824 Catalytic Activity" have a statistically significant higher F1 score (Table 2). The classifiers do not capture attributes that are involved in classes that belong to "GO:0005488 Binding" and "GO:0140110 Transcription Regulator Activity" as easily as attributes that belong to "GO:0003824 Catalytic Activity" because the children of Catalytic Activity have an average of 780 proteins to train on, while the children of Transcription Regulator Activity and Binding have an average of 280 proteins. The conclusion is that we need more data for some classes. One way to solve this would be not only to take experimental annotation codes but also proteins with evidence codes such as computationally inferred annotations with a review and not filter out proteins with a length of more than 1002 AA.

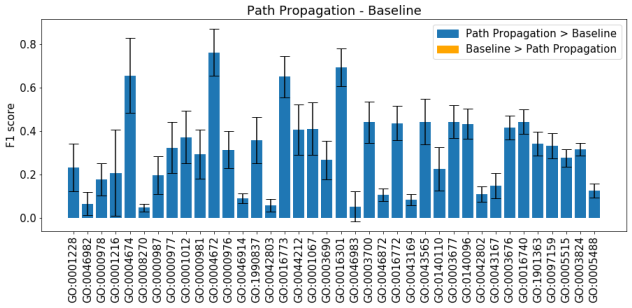

Fig. 5. F1-score of the baseline subtracted from the F1-score of the Path Propagation Algorithm. Besides the GO term "GO:0046983", all terms are statistically significantly better than the random stratified baseline.

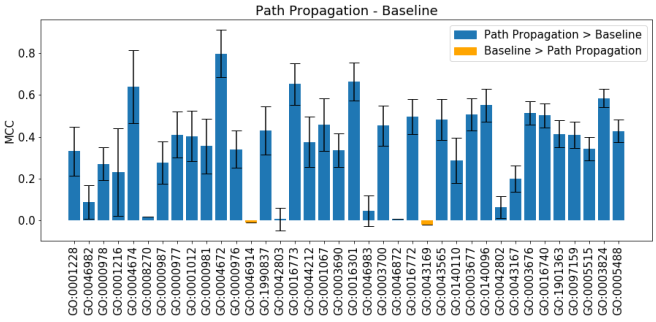

Fig. 7. MCC of the baseline subtracted from the MCC of the Path Propagation Algorithm. The performance of some classifiers are not better than the baseline performance.

4 Conclusion

The Hierarchical structure makes most classes besides the most general ones, that do not hold much information, very imbalanced. The way we solved this imbalance made our models very prone to errors, because, for our positive and negative training data, we only used proteins that belong to the parent class. The next step is to look at other ways to deal with the imbalance. Another solution is developing an algorithm that does the Path Propagation differently; for example, using logic not to annotate two opposing biological classes to the same protein.

The other conclusion from the analysis is that we need to increase the data set size. One way to do this is to take lower quality data (for example, computationally inferred with a review) and use this data for training but not for testing as we only used experimentally inferred annotations.

We compared three different algorithmic approaches to predict the hierarchical directed acyclic graph of GO. The Flat Classification performs the worst, due to the way we handled the imbalance. The Hierarchical algorithm performs better and makes biological more logical predictions, e.g., it is not possible for a protein to belong to the child but not the parent node. Propagating the possibilities of all parents, was the best algorithm as it eliminates paths where all probabilities are just above 0.5.

Acknowledgements

The authors of this paper acknowledge the help and useful discussions after the presentations with Michael Bernhofer, Christian Dallago, Michael Heinzinger and Maria Schelling.

References

1. Ashburner, Michael, u. a. „Gene Ontology: tool for the unification of biology“. Nature Genetics, Bd. 25, Nr. 1, Mai 2000, S. 25. EBSCOhost, doi:10.1038/75556.

2. du Plessis, L., u. a. „The What, Where, How and Why of Gene Ontology--a Primer for Bioinformaticians“. Briefings in Bioinformatics, Bd. 12, Nr. 6, November 2011, S. 723–35. DOI.org (Crossref), doi:10.1093/bib/bbr002.

3. The UniProt Consortium. „UniProt: A Worldwide Hub of Protein Knowledge“. Nucleic Acids Research, Bd. 47, Nr. D1, Januar 2019, S. D506–15. DOI.org (Crossref), doi:10.1093/nar/gky1049.

4. Jensen, L. J., u. a. „Prediction of Human Protein Function from Post-Translational Modifications and Localization Features“. Journal of Molecular Biology, Bd. 319, Nr. 5, Juni 2002, S. 1257–65. DOI.org (Crossref), doi:10.1016/S0022-2836(02)00379-0.

5. Jensen, L. J., u. a. „Prediction of Human Protein Function According to Gene Ontology Categories“. Bioinformatics, Bd. 19, Nr. 5, März 2003, S. 635–42. academic.oup.com, doi:10.1093/bioinformatics/btg036.

6. Cozzetto, Domenico, u. a. „FFPred 3: Feature-Based Function Prediction for All Gene Ontology Domains“. Scientific Reports, Bd. 6, Nr. 1, August 2016, S. 1–11. www.nature.com, doi:10.1038/srep31865.

7. Heinzinger, Michael, u. a. „Modeling the Language of Life – Deep Learning Protein Sequences“. BioRxiv, Mai 2019, S. 614313. www.biorxiv.org, doi:10.1101/614313.

8. Radivojac, Predrag, u. a. „A Large-Scale Evaluation of Computational Protein Function Prediction“. Nature Methods, Bd. 10, Nr. 3, März 2013, S. 221–27. www.nature.com, doi:10.1038/nmeth.2340.

9. Kulmanov, Maxat, u. a. „DeepGO: Predicting Protein Functions from Sequence and Interactions Using a Deep Ontology-Aware Classifier“. Bioinformatics, herausgegeben von Jonathan Wren, Bd. 34, Nr. 4, Februar 2018, S. 660–68. DOI.org (Crossref), doi:10.1093/bioinformatics/btx624.

10. Cheng, Liangxi, u. a. „Gene Function Prediction Based on the Gene Ontology Hierarchical Structure“. PLoS ONE, herausgegeben von Yi-Hsiang Hsu, Bd. 9, Nr. 9, September 2014, S. e107187. DOI.org (Crossref), doi:10.1371/journal.pone.0107187.

11. Pedregosa, Fabian, u. a. „Scikit-learn: Machine Learning in Python“. arXiv:1201.0490 [cs], Juni 2018. arXiv.org, http://arxiv.org/abs/1201.0490.

12. Nair, Rajesh, und Burkhard Rost. „Mimicking Cellular Sorting Improves Prediction of Subcellular Localization“. Journal of Molecular Biology, Bd. 348, Nr. 1, April 2005, S. 85–100. DOI.org (Crossref), doi:10.1016/j.jmb.2005.02.025.