CrowdGO: a wisdom of the crowd-based Gene Ontology annotation tool

Maarten J.M.F. Reijnders[1,2]

[1]Department of Ecology and Evolution, University of Lausanne, and Swiss Institute of Bioinformatics, 1015 Lausanne, Switzerland
[2]Laboratory of Systems and Synthetic Biology, Department of Agrotechnology and Food Sciences, Wageningen University and Research, Wageningen, The Netherlands

Abstract

Motivation: Protein function prediction tools vary widely in their methodologies, resulting in different sets of GO terms being correctly predicted. Ideally, multiple tools are combined to achieve a higher recall of GO terms while increasing precision.

Results: CrowdGO combines input predictions from any number of tools and combines them based on the Gene Ontology Directed Acyclic Graph. Using each GO terms information content, the semantic similarity between GO predictions of different tools, and a Support Vector Machine model, it achieves improved precision and recall compared to each of the tools separately (Figure 1).

Availability: CrowdGO can be found at https://gitlab.com/mreijnders/CrowdGO

Intro

Protein function prediction is a hot topic in the omics era due to the slow and laborious process of validating Gene Ontology (GO) terms experimentally. In the latest Critical Assessment of Functional Annotation challenge [1], 68 different prediction methods were assessed for their ability to de novo predict GO terms. These methodologies use protein information such as sequence similarity, protein structure, and taxonomy, amongst others, and vary widely in their predictions. In an ideal case scenario, omics studies which require protein function prediction do so using multiple different tools to capture as much functional information as possible.

For this purpose, we designed CrowdGO: a wisdom of the crowd-based GO term annotation tool. It utilizes GO term semantic similarity in combination with machine learning to combine the predictions of multiple tools. CrowdGO achieves the GO term recall of combining multiple predictions, while improving precision.

Software description

CrowdGO modules

CrowdGO comes with two python scripts: CrowdGO_training.py and CrowdGO.py. The former is used to train a Support Vector Machine (SVM) model for specific combination of tools and their predictions, based on a training set of protein GO prediction data. The latter performs de novo predictions using a pre-trained SVM model.

CrowdGO input

Input consists of a tabular file containing the following columns: prediction tool name, protein, predicted GO term, and prediction probability. In the case of CrowdGO_training a column is added stating whether the prediction is true or false.

Calculating GO term information content

For each GO term the Information Content is calculated according to Lin's formula [2]:

$$IC(GOi) = \frac{|-log(GO: GOi \rightarrow GO)|}{|\{GO: GO \in GOA\}|}$$

The information content of GO_i is calculated by checking how many times the term is annotated in the UniProt database relative to the total amount of GO terms in the database.

Overlapping GO term predictions

For each tool, predictions for a protein are checked against the predictions of other tools. If the GO terms appear in the same GO Directed Acyclic Graph [3, 4], they are clustered into one group of predictions.

Semantic similarity

Clustered GO terms are compared using Lin's similarity approach [2]:

$$Sim(GOi, GOj) = \frac{2 * ICParent(GOi, GOj)}{IC(GOi) * IC(GOj)}$$

ICParent is the GO term with the lowest information content between GO_i and GO_j. The result is a similarity score between 0 and 1, with 0 meaning no similarity and 1 being the same GO term.

Training a Support Vector Machine model

CrowdGO now has the following information for each overlapping set of terms: GO term with the highest information content, amount of tools predicting the same term, the GO term predicted by each individual tool (if any), the score for each tool's prediction, information content for each tool's prediction, and semantic similarity scores between each overlapping prediction. The SVM model is trained with labeled training data using the numerical values mentioned above as features, and saved to a pickled output file.

Predicting GO terms with a Support Vector Machine model

A pre-trained SVM model is loaded into CrowdGO. This model is used to predict the probabilities of the GO term being correct or incorrect, resulting in a score between 0 and 1.

Output

The output format consists of the following columns: protein, predicted GO term, and probability score. Additionally, the following temporary files are saved in the process of running CrowdGO: SVM model input data, and the raw output data.

CrowdGO Demonstration

Training and test set preparation

To demonstrate the software, we provide a data set based on SwissProt proteins [5] created after 31/12/2016. Only proteins with at least three experimentally validated GO terms are kept, in an attempt to reduce an overestimation of false positive predictions as described by *Dessimoz et al* [6]. Initial predictions are done using Argot2.5 [7], InterProScan 5.21-60 [8], and DeepGOPlus [9], using only databases from before 2017 to ensure blind predictions. Half of the proteins are used to train an SVM model with CrowdGO_training, and half are used to predict GO terms using CrowdGO. The training set predictions are labeled by validating against the SwissProt GO terms with evidence codes indicating manual or experimental validation. To remove noise, Argot2.5 predictions with a score lower than 1, and DeepGOPlus predictions with a score lower than 0.2, are pruned from the training and test set. CrowdGO is run with two prediction methods (CrowdGO 2) and 3 prediction methods (CrowdGO 3). CrowdGO 2 is run with Argot2.5 and DeepGOPlus, and CrowdGO 3 adds InterProScan.

Precision and recall curve

Figure 1 shows the sensitivity and specificity of each method. The precision recall curve shows predictions made by CrowdGO 2, CrowdGO 3, Argot2.5, DeepGOPlus, and InterProScan. Additionally, we added CrowdGO unfiltered, which is CrowdGO 3 without removing noisy data from Argot2.5 and DeepGOPlus. As a baseline, we added a prediction curve based solely on a GO terms Information Content score.

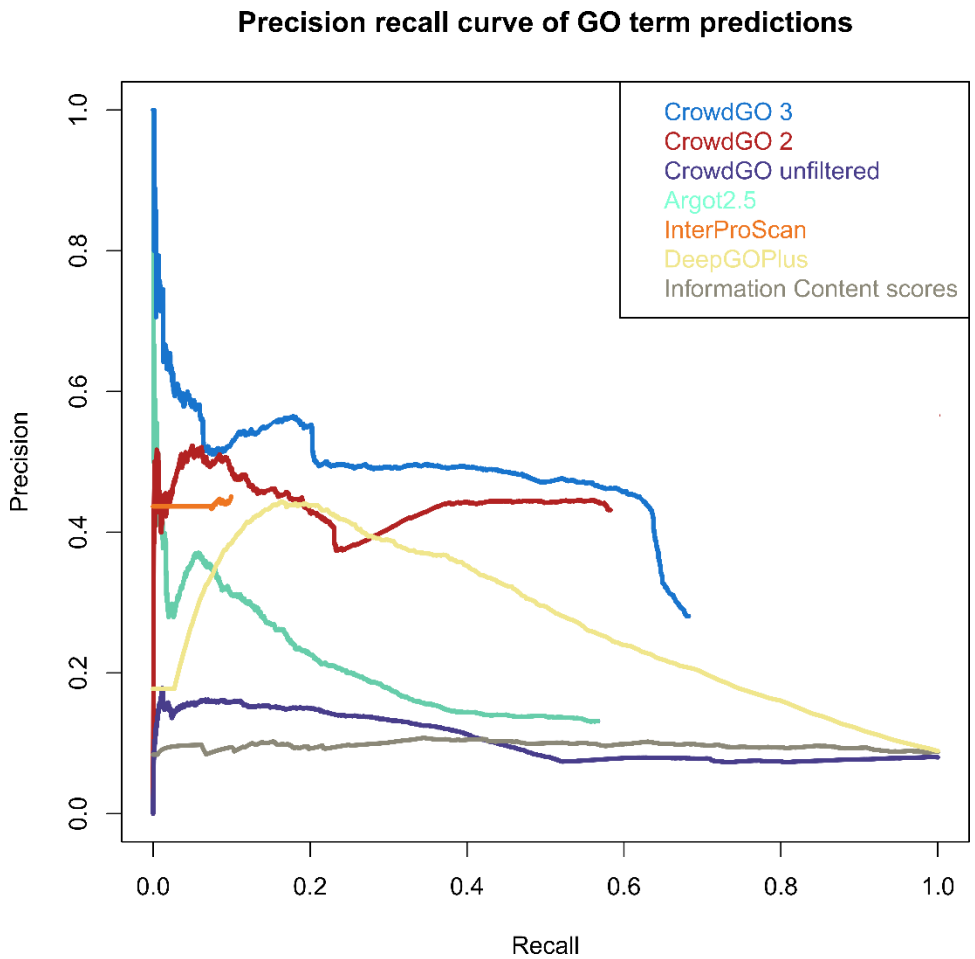

Figure 1: precision recall curve of the predictions for each CrowdGO input tool, CrowdGO with two input methods, CrowdGO with three input methods, and CrowdGO with three input methods and no pre-processing. The Information Content score is used as a baseline predictor. CrowdGO training input is pre-processed by removing all Argot2.5 predictions with a score below 1, and all DeepGOPlus predictions with a score below 0.2 to remove noise

The maximum f1-scores were calculated for CrowdGO 3, Argot2.5, InterProScan, and DeepGOPlus: 0.52, 0.22, 0.16, and 0.41 respectively. The probability score cutoff for CrowdGO coinciding with its maximum f1-score is 0.47, and should be used when predicting GO terms using the supplied SVM model.

Conclusion

We have successfully improved GO term predictions of existing methods by combining their results using CrowdGO. In particular, CrowdGO manages a high recall while substantially improving accuracy compared to existing predictions (Figure 1). Adding more input methods to CrowdGO increases its performance. However, performance is heavily affected by noise in the data as seen in the performance drop when not pre-processing the input data. Therefore, when training their own SVM model users are encouraged to test its performance using the training and test set data provided with CrowdGO. The training and testing data used in this paper, the SVM model, as well as the python scripts used for pre-processing and post-processing the data, are available on the CrowdGO gitlab page.

Acknowledgements

This study is funded in part by the VLAG Graduate School, and the Swiss National Science Foundation grant PP00P3_170664.

References

1. Zhou, N., et al., *The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens.* bioRxiv, 2019: p. 653105.
2. Lin, D. *An information-theoretic definition of similarity*. in *Icml*. 1998. Citeseer.
3. Ashburner, M., et al., *Gene Ontology: tool for the unification of biology.* Nature genetics, 2000. **25**(1): p. 25.
4. Gene Ontology, C., *The gene ontology resource: 20 years and still GOing strong.* Nucleic acids research, 2018. **47**(D1): p. D330-D338.
5. Consortium, U. *UniProt expert biocuration*. 2018; Available from: https://www.uniprot.org/help/?fil=section:biocuration.
6. Dessimoz, C., N. Škunca, and P.D. Thomas, *CAFA and the open world of protein function predictions.* Trends in Genetics, 2013. **29**(11): p. 609-610.
7. Lavezzo, E., et al., *Enhancing protein function prediction with taxonomic constraints--The Argot2.5 web server.* Methods, 2016. **93**: p. 15-23.
8. Jones, P., et al., *InterProScan 5: genome-scale protein function classification.* Bioinformatics, 2014. **30**(9): p. 1236-1240.
9. Kulmanov, M. and R. Hoehndorf, *DeepGOPlus: Improved protein function prediction from sequence.* bioRxiv, 2019: p. 615260.