Wei2GO: weighted sequence similarity-based protein function prediction

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

[1] Department of Ecology and Evolution, University of Lausanne, Lausanne, Switzerland
[2] Swiss Institute of Bioinformatics, Lausanne, Switzerland

ABSTRACT

Background: Protein function prediction is an important part of bioinformatics and genomics studies. There are many different predictors available, however most of these are in the form of web-servers instead of open-source locally installable versions. Such local versions are necessary to perform large scale genomics studies due to the presence of limitations imposed by web servers such as queues, prediction speed, and updatability of databases.

Methods: This paper describes Wei2GO: a weighted sequence similarity and python-based open-source protein function prediction software. It uses DIAMOND and HMMScan sequence alignment searches against the UniProtKB and Pfam databases respectively, transfers Gene Ontology terms from the reference protein to the query protein, and uses a weighing algorithm to calculate a score for the Gene Ontology annotations.

Results: Wei2GO is compared against the Argot2 and Argot2.5 web servers, which use a similar concept, and DeepGOPlus which acts as a reference. Wei2GO shows an increase in performance according to precision and recall curves, F_{max} scores, and S_{min} scores for biological process and molecular function ontologies. Computational time compared to Argot2 and Argot2.5 is decreased from several hours to several minutes.

Availability: Wei2GO is written in Python 3, and can be found at https://gitlab.com/mreijnders/Wei2GO.

INTRODUCTION

Predicting proteins functions computationally is an integral part of omics research, due to the laborious process of experimentally determining protein functions. This is most apparent in the size difference of Uniprot's TrEMBL and SwissProt databases, which contain predicted and manually curated proteins respectively (*The UniProt Consortium, 2019*). Because of the dependence on predicted protein functions, many software tools have been developed in an attempt to improve the accuracy of electronically annotating these proteins with Gene Ontology (GO) terms (*Ashburner et al., 2000*; *The Gene Ontology Consortium, 2019*).

Corresponding author
Maarten J.M.F. Reijnders,
mreijnders@live.nl

A comprehensive overview of protein function prediction methods is the Critical Assessment of Functional Annotation (CAFA) competition (*Zhou et al., 2019*). In its most recent edition, CAFA3, it compared the predictions of 68 participants employing a wide variety of techniques including sequence homology, machine learning, and text mining amongst others. The top-performing predictors in this challenge are considered the state-of-the-art for protein function prediction. With this exposure and proven performance, predictors participating in CAFA have the potential to act as a resource for any user wanting to predict protein functions of their organism of interest. But for CAFA participants to be such a resource, they have two other factors to consider: accessibility and speed. A look at the top performers of CAFA shows that nine out of fourteen teams which placed top ten in one of the GO ontologies, provide their method as a tool accessible through a web server. However, only four out of fourteen teams provide a working locally installable option. Web servers provide an excellent option for the annotation of a limited amount of proteins, but larger amounts are problematic due to computational limitations.

Contrary to web servers, locally installable software provides opportunities to speed up predictions without limitations usually imposed, such as limited protein sequence inputs or queues shared with multiple users. Local versions provide users the means to update the software databases, which is a necessity to provide the most up-to-date and accurate predictions. As a bonus, these versions are often open-source and provide flexibility in the usage of the tool by advanced users, *e.g.* by utilizing a meta-predictor (*Reijnders & Waterhouse, 2019*). As such, there is a need for accurate open-source and locally installable prediction tools.

This paper describes Wei2GO: a weighted sequence similarity and python-based open-source function prediction software. Wei2GO utilizes DIAMOND (*Buchfink, Xie & Huson, 2015*) and HMMER (*HMMER, 2021*) searchers against the UniProtKB (*The UniProt Consortium, 2019*) and Pfam (*El-Gebali et al., 2019*) databases respectively, acquires GO terms through these homology searches, and calculates several weighted scores to accurately associate probabilities to these GO terms. Wei2GO is similar in concept to the web server-based Argot2.5 (*Lavezzo et al., 2016*, p. 2) and its predecessor Argot2 (*Falda et al., 2012*, p. 2), which were top performers in CAFA3 and CAFA2 respectively (*Radivojac et al., 2013*; *Zhou et al., 2019*). Due to this shared niche, this paper provides comparisons between these tools and Wei2GO using precision recall curves, F_{max} scores, S_{min} scores, and a comparison of computational time for their predictions.

METHODS

Wei2GO is a function prediction tool based on sequence similarity with reference proteins, using adapted methods from Argot2 and Argot2.5 to provide an open-source alternative and to improve predictions where possible. It uses two of the most common sequence alignment algorithms, DIAMOND and HMMER, against the UniProtKB and Pfam database respectively. Each sequence alignment hit is associated with GO terms, taken from the Gene Ontology Annotation (GOA) database (*Huntley et al., 2015*) for UniProtKB, and pfam2go (*Blum et al., 2021*) for Pfam. All GO terms associated with a hit are given a weight relative to their e-value and associated annotation evidence. The weights

for each protein-GO term pair are summed across the DIAMOND and HMMER matches. Based on their semantic similarity, the weights of similar GO terms are obtained to get a group score. Finally the GO weight, its information content, and group score are used to calculate the final score associated with the prediction.

Equations outlined below are in agreement with those of Argot2 and Argot2.5 unless otherwise specified. The main algorithmic differences between Argot2, Argot2.5, and Wei2GO are:

1. The omission of a taxonomy filter used by Argot2.5.
2. The omission of supplementing Pfam sequence similarity hits with GO terms of non-Pfam origin.
3. The addition of a weight multiplier for GO terms derived from sequence similarity with UniProtKB proteins if these original terms are annotated with an evidence code other than IEA (Inferred from Electronical Annotation).

Wei2GO input preparation

Wei2GO requires two input files:

1. Diamond or BLASTP hits against the UniProtKB database in tabular format (–outfmt 6).
2. HMMScan hits against the Pfam-A database as a table output (–tblout).

These files can either be produced separately, or be incorporated with Wei2GO *via* its Snakemake pipeline (*Mölder et al., 2021*).

Wei2GO algorithm

Each protein matched with a UniProtKB reference *via* BLAST or DIAMOND is annotated with GO terms associated to these reference proteins in the GOA database. Consecutively, each protein matched with a Pfam domain *via* HMMScan is annotated with GO terms associated to this domain in the pfam2go database.

For every protein v, a GO term i is given a weight $W(v,i)$ by taking the sum of the -log10 e-value E of all UniProtKB and Pfam sequence similarity hits j associated with this GO term, where e-values above 1 are filtered out:

$$W(v,i) = \sum_{j \in Hits(v,i)} -log10(E_j),$$ (1)

where $Hits(v,i)$ is the set of BLAST and HMMScan sequence aligment hits to v annotated with GO term i, and E_j is the e-value of the sequence alignment j. Unique to Wei2GO compared to Argot is that terms annotated to a UniprotKB protein with an evidence code other than IEA (Inferred from Electronical Annotation) are given a weight multiplier:

$$W(v,i) = \sum_{j \in Hits(v,i)} -log10(E_j) \cdot log2(P),$$ (2)
where P is the ratio between the number of GO terms annotated with an IEA vs GO terms annotated with an other evidence code in the GOA database. Given that the amount of GO terms annotated with an IEA code is substantially higher than those with non-IEA codes, P is larger than 1 and results in higher weights when based on non-IEA instead of IEA evidence codes.

In accordance with the GO Directed Acyclic Graph (DAG) structure all weights associated to child terms of a GO term are added to its weight, with the rationale that if a protein is annotated with a certain GO term it by default includes all its parent terms. This has the effect of assigning more confidence in GO terms higher up the DAG, $i.e.$ annotations that are more generic and less prone to errors.

The total weight associated with each protein-GO term pair is used to calculate the Internal Confidence (InC) score by taking the relative weight of the GO term compared to the weight of the root node (3):

$$InC(v, i) = \frac{W(v, i)}{W(root)}.$$ (3)

The InC score is used to calculate a Group Score (GS) for each GO term i associated to a protein, by taking the sum of the InC of all GO terms j annotated to a protein v, if GO term j has a semantic similarity of 0.7 or higher with GO term i:

$$GS(v, i) = \sum\nolimits_{j \, \varepsilon \, annotations(v)} InC(v, j) \cdot I\left(Sim(i, j) \geq 0.7\right),$$ (4)

where $I(.)$ is the indicator function which is equal to 1 if its argument is true and 0 otherwise, and annotations(v) is the set of GO annotations for protein v. Note that each GO term can have a unique GS score, as opposed to Argot where one singular GS score is defined for a group of similar GO terms. The value of Sim (semantic similarity) is calculated using Lin's formula ($Lin, 1998$) as:

$$Sim(i, j) = \frac{2 \cdot IC(k)}{IC(i) + IC(j)},$$ (5)

where k is the first joint parent term between GO term i and j, and the IC (Information Content) is calculated as:

$$IC(i) = -\log10(P(i),$$ (6)

where $P(i)$ is the fraction of proteins annotated to term i. Finally, a total score (TS) for each GO term annotated to a protein v is calculated as:

$$TS(v, i) = IC(i) \cdot \frac{InC(v, i)}{GS(v, i)} \cdot W(v, i).$$ (7)

The TS score is similar to that of Argot2 and Argot2.5, which combines the scores calculated by the equations above. However, Argot introduces an additional non-cumulative InC score, $i.e.$ Eq. (3) without adding the weights of child terms, to remove bias of scores towards terms which have a higher probability of being predicted, $i.e.$ a lower

IC. This is done in an attempt to reduce bias towards predicting GO terms that are generic at the cost of GO terms that are highly specific, as reflected by their IC. Wei2GO instead operates under the assumption that GO terms which are easier to predict due to any reason, including having a lower IC, should have this reflected in their score. Any possible negative effects of this bias towards terms with lower IC affecting terms with higher IC will be reflected in the 'S_{min}' evaluation metric, which measures performance proportional to these IC's (See 'Test set evaluation' section).

Test set generation

To assess the performance of Wei2GO it is compared to Argot2.5 and its predecessor Argot2, with the addition of DeepGOPlus (*Kulmanov & Hoehndorf, 2020*) which acts as a reference point. All proteins in the GOA database created between 01-01-2018 and 01-06-2020 were extracted. Only GO terms with the evidence code EXP (Inferred from Experiment), IDA (Inferred from Direct Assay), IPI (Inferred from Physical Interaction), IMP (Inferred from Mutant Phenotype), IGI (Inferred from Genetic Interaction), IEP (Inferred from Expression Pattern), TAS (Traceable Author Statement), and IC (Inferred by Curator), were retained. All molecular function terms 'GO:0005515' (protein binding) were removed from the annotation set if it was the only molecular function term annotated to the protein, to remove a large amount of bias due to many proteins having this generic term as their only molecular function term. This same approach was used in the CAFA3 test set generation.

The test set was expanded by including all terms according to their GO DAG structure using the 'is_a' and 'part_of' relationships for parent terms. The final test set consists of 4,067 proteins with 97,054 assigned GO terms of which 1,922 proteins have 59,697 biological process terms, 651 proteins have 4,785 molecular function terms, and 2,387 proteins have 32,572 cellular component terms.

Test set evaluation

Precision and recall is used for the assessment of Wei2GO. For a threshold t, the precision pr and recall rc are calculated per protein v, T_v is the set of true GO terms for a protein v, P_v is the set of predicted GO terms for a protein v, $m(t)$ is the number of proteins with at least one GO term predicted with a score above the threshold, and n is the total number of proteins:

$$pr_v(t) = \frac{|P_v \cap T_v|}{|P_v|},$$

(8)

$$rc_v(t) = \frac{|P_v \cap T_v|}{|T_v|},$$

(9)

$$avgPr(t) = \frac{1}{m(t)} \cdot \sum_{v=1}^{m(t)} pr_v(t),$$

(10)

$$avgRc(t) \ = \ \frac{1}{n} \cdot \sum_{v=1}^{n} rc_v(t). \tag{11}$$

The optimal balance between precision and recall is calculated using the maximum F1-score, which for this study is calculated by calculating F1-scores with 0.01 step-sizes after transforming the Total Scores to a 0–1 range:

$$F_{max} = \max_t \left\{ \frac{2 \cdot avgPr(t) \cdot avgRc(t)}{avgPr(t) + avgRc(t)} \right\}. \tag{12}$$

The S_{min} score is calculated based on the remaining uncertainty ru and missing information mi, which is the IC of all false negatives and false positives, respectively. A threshold t step-size of 0.01 is used after transforming the Total Scores to a 0–1 range.

$$S_{min} = \min_t \sqrt{ru(t)^2 + mi(t)^2}, \tag{13}$$

with ru as:

$$ru(t) = \frac{1}{n} \sum_{v=1}^{n} \sum_{GO \in T_v - P_v(t)} IC(i), \tag{14}$$

and mi as:

$$mi(t) = \frac{1}{n} \sum_{v=1}^{n} \sum_{GO \in P_v(t) - T_v} IC(GO). \tag{15}$$

MATERIALS

The following datasets and software versions were used:

- UniProt Knowledgebase release 2016-11 for DIAMOND searches
- Pfam-A release 30 for HMMER searches
- DIAMOND version 0.9.11.112
- HMMER version 3.3
- GOA UniProt data release 161 for creating the Wei2GO required data files
- GOA UniProt data release of 13-06-2020 for generating the test set
- go.obo release 02-12-2016
- pfam2go generated based on interpro2go release 31-10-2016
- Argot2 and Argot2.5 were last accessed on 08-04-2021
- DeepGOPlus was downloaded from https://github.com/bio-ontology-research-group/deepgoplus on 30-10-2020.

All datasets used by Wei2GO correspond to the datasets used by Argot2 and Argot2.5.

Table 1 An overview of the major algorithmic differences between Argot2, Argot2.5, and Wei2GO.

	Supplementing pfam2go annotations	Evidence-based weights	Taxonomy-based filtering	Limited number of input sequences
Argot2	Yes	No	No	Yes
Argot2.5	Yes	No	Yes	Yes
Wei2GO	No	Yes	No	No

RESULTS

Wei2GO is made to facilitate open-source protein function prediction, and the ability to do this independent of web-servers. Two of these web-servers, Argot2.5 and its predecessor Argot2, utilize sequence similarity searches against the UniProtKB and Pfam databases, and apply a weighing algorithm to come to a final prediction score. Wei2GO takes that idea and puts this in an open-source and locally installable concept, which opens up the possibility to update data used by the algorithm independent of the original authors, and the ability to perform protein function prediction on big datasets.

Algorithm comparisons between Wei2GO and Argot

Because Wei2GO's concept is based on that of Argot, it is important to highlight major algorithmic differences between the different software (Table 1). Both Argot2 and Argot2.5 supplement pfam2go annotations with the GOA GO terms of all proteins belonging to each Pfam entry, which Wei2GO does not. Wei2GO weighs GO annotations differently based on the associated evidence codes in the GOA database, which both Argot's do not. Argot2.5 is the only software that performs taxonomy-based filtering of GO terms. And finally, Argot2 and Argot2.5 can only handle 5,000 and 10,000 proteins at a time, respectively.

To get an indication of the effect of the algorithmic differences between the software, Table 2 shows an overview of the correctly and incorrectly predicted protein-GO term pairs, irrespective of their prediction scores. The main difference between the software is the amount of false positive (FP) predictions. Argot2, which supplements pfam2go annotations, and to a lesser extend Argot2.5, which supplements pfam2go annotations and applies a taxonomic filter, show a substantially larger amount of FP predictions compared to Wei2GO whilst not showing a large increase in true positive (TP) predictions. Exceptions are Biological Processes (BPO) where Argot2.5 shows both a lower TP and FP count compared to Wei2GO, and Cellular Components (CCO) where Argot2 and Argot2.5 show an increase in TP count. This general trend suggests that supplementing Pfam annotations by adding GOA GO annotations from proteins belonging to the Pfam entry, which is not present in Wei2GO, adds quantity but not quality to the predictions.

Performance comparisons between Wei2GO and Argot

Several performance metrics are used to compare Wei2GO and Argot: precision-recall curves, F_{max} scores, S_{min} scores, and computational time. DeepGOPlus is added as a part of all these analyses. This deep learning-based and open source protein function prediction

Table 2 Comparison of Argot2, Argot2.5, and Wei2GO raw prediction counts. All predicted protein-GO term pairs present in the test set are classified as a True Positive (TP) and all predicted protein-GO term pairs not present in the test set are classified as False Positive (FP). Shown are the numbers for Biological Process ontology (BPO), Molecular Function ontology (MFO), and Cellular Component ontology (CCO).

	BPO TP	BPO FP	MFO TP	MFO FP	CCO TP	CCO FP
Argot2	39,781	1,766,545	3,325	122,286	19,325	238,918
Argot2.5	29,291	104,322	3,677	177,220	17,212	48,230
Wei2GO	39,460	119,702	3,467	33,815	10,903	33,748
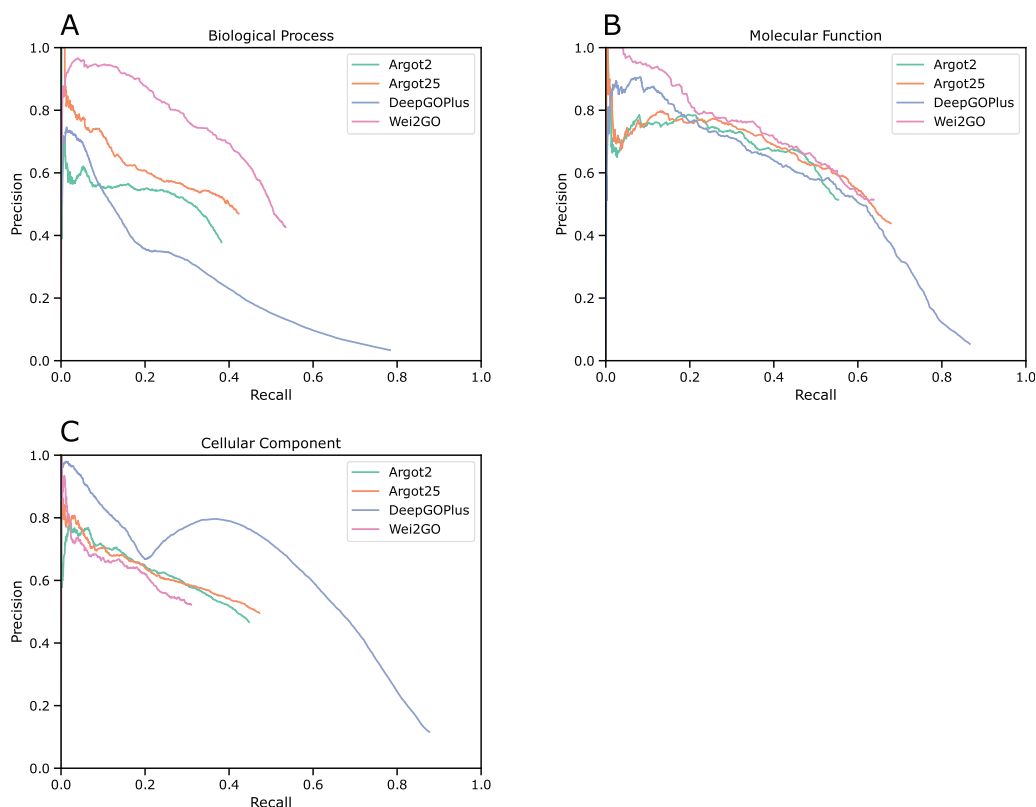
Figure 1 Precision recall curves for (A) biological process, (B) molecular function, and (C) cellular component ontologies.

software is an excellent example of both predictive performance and usability. Adding this software in the assessments provides perspective as to any performance differences between Wei2GO and Argot. For the purpose of this analysis, all databases used by Wei2GO are the same versions used by Argot2 and Argot2.5. Each species was given to Argot2.5 as a separate input with a species level taxonomic code.

Precision recall curves were generated for biological process (BPO), molecular function (MFO), and cellular component (CCO) ontologies (Fig. 1). This analysis shows that Wei2GO is the best performer in both the BPO and MFO assessment, with most notably BPO having a substantial increase in precision. For CCO, Wei2GO is slightly outperformed by Argot2, while DeepGOPlus performs substantially better than both.

Table 3 Comparison of the maximum F1-scores (Fmax) and minimum S-scores (Smin) between methods for biological processes (BPO), molecular functions (MFO), and cellular components (CCO). Best results displayed in bold.

Method	Fmax BPO	Fmax MFO	Fmax CCO	Smin BPO	Smin MFO	Smin CCO
DeepGOPlus	0.32	0.56	**0.63**	90.31	11.71	22.71
Argot2	0.39	0.55	0.46	73.42	10.00	22.54
Argot2.5	0.45	**0.58**	0.48	69.01	9.72	**22.21**
Wei2GO	**0.52**	**0.58**	0.39	**53.08**	**8.87**	23.03

Table 4 Comparison of the maximum F1-scores (Fmax) between the top three CAFA3 performers, Argot2.5, and Wei2GO. Given are the scores for biological processes (BPO), molecular functions (MFO), and cellular components (CCO). Argot2.5 scores are taken from the CAFA. Best results displayed in bold.

Method	Fmax BPO	Fmax MFO	Fmax CCO
CAFA #1	**0.40**	**0.62**	**0.61**
CAFA #2	0.39	0.54	**0.61**
CAFA #3	0.38	0.53	0.60
DeepGOPlus	0.39	0.56	**0.61**
Argot2.5	0.38	0.52	0.59
Wei2GO	0.39	0.52	0.57

F_{max} scores were calculated based on the highest F1-score at any given threshold, and S_{min} scores were calculated based on the lowest S-score at any given threshold (Table 3). The F_{max} score is highest in Wei2GO for BPO, similar between Wei2GO and Argot2.5 for MFO, and is outperformed by both Argot2 and DeepGOPlus for CCO. Wei2GO shows the lowest S_{min} score for both BPO and MFO, and is outperformed by both Argot2 and DeepGOPlus for CCO.

Wei2GO was compared to Argot2.5 on the CAFA3 data using the F_{max} score, relative to the top three performers of the challenge (Table 4). Both Wei2GO and Argot2.5 are competitive, and show similar results to those in Table 3. One notable difference is a discrepancy in performance for the BPO category. Protein function annotation is an 'open world' problem, *i.e.* a GO term annotation can be incorrectly assigned as a false positive due to an as-of-yet unidentified function of a protein (*Dessimoz, Škunca & Thomas, 2013*). The test set used for Table 3 is more complete for BPO terms than the CAFA3 challenge set, with 6.5 average leaf terms per protein compared to 3.6 respectively, resulting in a larger F_{max} difference between Wei2GO and Argot2.5. These differences in GO terms per protein were marginal for MFO and CCO terms.

Computational times were compared in Table 5 based on the 4,067 proteins of the test set. The Argot2 and Argot2.5 web-servers were run without queue times, and all analyses were run on one CPU.

Table 5 A comparison of the computational time required for the prediction of the test set used for the analysis in this paper. Based on 4,226 proteins. DIAMOND and HMMScan calculations required for the input of Argot2, Argot2.5, and Wei2GO, were excluded from the computational times. No queue times for Argot2 and Argot2.5 were reported.

Method	Computational time (min)
DeepGOPlus	2
Argot2	173
Argot2.5	303
Wei2GO	4

DISCUSSION

This paper describes Wei2GO: a weighted sequence similarity and python-based open source function prediction software. It utilizes a combination of DIAMOND and HMMER searches, and combines these to extract a probability score associated with GO terms. It is compared against the web-based tools Argot2 and Argot2.5, which share a similar concept, and DeepGOPlus, which acts as a reference point for the various metrics used.

Wei2GO shows a large improvement in predicting BPO terms according to the precision and recall, F_{max}, and S_{min} evaluation metrics. Similarly, MFO shows a small improvement in the precision recall curve and S_{min} scores, but no difference in the F_{max} score compared to Argot2.5. CCO shows a slight decrease in performance over Argot2 and Argot2.5, and a substantial decrease in comparison to the reference DeepGOPlus.

In comparison to the reference, DeepGOPlus, Argot2 and Argot2.5 similarly show an improvement in BPO term prediction, which implies a sequence similarity-based approach is preferable over a deep learning-based approach. Since Argot and Wei2GO share a similar approach and used the same databases for their predictions, performance difference between them implies key algorithmic differences leading to a large improvement in BPO prediction for Wei2GO over Argot2 and Argot2.5. No definitive conclusion can be made due to the closed-source concept of Argot, however Table 2 shows Argot2 predicts many more GO terms than Wei2GO whilst not increasing the amount of correct predictions, which are then reduced by Argot2.5 due to presumed taxonomic filtering. This large increase is likely due to the Argot supplementing pfam2go annotations with the GOA annotations of all proteins sharing a Pfam entry.

MFO predictions are similar between Argot2, Argot2.5, Wei2GO, and the reference DeepGOPlus. While the same trend is shown in the number of correct and incorrect GO terms predicted by Argot2, Argot2.5, and Wei2GO as for BPO (Table 2), the precision and recall curve (Fig. 1), F_{max}, and S_{min} scores (Table 3) show Argot2 and Argot2.5 are better at sorting out the incorrect predictions, leading to a much smaller increase in performance for Wei2GO.

Wei2GO performs worse than Argot2 and DeepGOPlus for CCO predictions. The big difference in precision between DeepGOPlus and the other methods indicates deep learning-based prediction is preferable for this ontology compared to sequence similarity-based prediction. A decrease in performance for Wei2GO compared to Argot2

and Argot2.5 could be due to the relative simple nature of the GO DAG for CCO compared to the other ontologies. Enhancing Pfam domains with the GOA GO terms of its protein members could in such a case improve the predictions. While the F_{max} difference between Argot2, Argot2.5, and Wei2GO is substantial, Fig. 1C indicates precision is similar and the decrease in performance is almost entirely due to a decrease in recall.

As observed in the precision recall curves (Fig. 1), Wei2GO offers a smoother curve for BPO and MFO predictions compared to Argot2 and Argot2.5. This is preferable, as it suggests better sorting of high-confidence and low-confidence predictions. While no definitive conclusion can be made due to the closed-source nature of Argot, this is likely a result of the evidence-based weighing of Wei2GO, as it is more reliable to transfer GO terms through sequence similarity when these come from experimental or curated evidence.

Finally, the computational time for Wei2GO is orders of magnitude lower than that of Argot2 and Argot2.5 (Table 5), while allowing more proteins to be predicted simultaneously. This opens up the possibility of high-throughput annotating large amounts of omics data with GO terms, as is often needed in the study of comparative genomics and big data studies.

Future implementations of Wei2GO will address two aspects. Firstly, to assess the impact of DIAMOND and HMMScan on Wei2GO, Fig. S1 shows a precision recall curve of Wei2GO using both DIAMOND and HMMScan, only DIAMOND, and only HMMScan input. Currently the HMMScan input provides recall but not precision to the Wei2GO predictions, indicating that threating HMMScan input different than DIAMOND input could be beneficial to final predictions, or that an alternative to HMMScan can provide improved results. Secondly, future versions of Wei2GO will allow for user-adjustable parameters, namely choosing between multiple semantic similarity algorithms and choosing between semantic similarity cutoffs used in the Wei2GO algorithm, as different ontologies or species could potentially benefit from a different parameter selection.

CONCLUSION

Wei2GO offers an open-source, locally installable alternative of a proven concept for protein function prediction in the name of Argot2 and Argot2.5. Algorithmic differences in Wei2GO lead to a large performance increase of BPO predictions, while showing similar performance in MFO or CCO predictions. Computational time is decreased from hours down to minutes on a test set of 4,193 proteins, and contrary to Argot, Wei2GO is able to predict an unlimited amount of proteins at a time. In conclusion, Wei2GO not only provides a faster, more flexible, and open-source alternative to Argot2 and Argot2.5, but is able to substantially improve BPO predictions while producing similar performance for MFO and CCO.

Implementation and availability

Wei2GO requires Python 3.6 or higher, and the NumPy package. Supplemental Files used to create the results in this paper, instructions on how to install the software, and

instructions on how to run the software can be found on https://gitlab.com/mreijnders/Wei2GO. Additionally, Snakemake scripts are provided to run the full annotation pipeline, and update the databases to their latest versions.

ACKNOWLEDGEMENTS

The author would like to thank Dr. Robert M. Waterhouse for offering suggestions and editing parts of the manuscript, and Dr. Bastian V.H. Hornung for proofreading the manuscript, as well as Reviewer 1, Reviewer 3, and the Editor for providing useful suggestions that have improved the manuscript.

Funding

This study is funded by Swiss National Science Foundation grant PP00P3_170664. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Swiss National Science Foundation: PP00P3_170664.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Maarten J.M.F. Reijnders conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability

The software and its Supplemental Files are available at GitLab: https://gitlab.com/mreijnders/wei2go.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj.12931#supplemental-information.

REFERENCES

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G. 2000. Gene ontology: tool for the unification of biology. *Nature Genetics* **25(1)**:25–29 DOI 10.1038/75556.

Blum M, Chang H-Y, Chuguransky S, Grego T, Kandasaamy S, Mitchell A, Nuka G, Paysan-Lafosse T, Qureshi M, Raj S, Richardson L, Salazar GA, Williams L, Bork P,

Bridge A, Gough J, Haft DH, Letunic I, Marchler-Bauer A, Mi H, Natale DA, Necci M, Orengo CA, Pandurangan AP, Rivoire C, Sigrist CJA, Sillitoe I, Thanki N, Thomas PD, Tosatto SCE, Wu CH, Bateman A, Finn RD. 2021. The InterPro protein families and domains database: 20 years on. *Nucleic Acids Research* **49(D1)**:D344–D354 DOI 10.1093/nar/gkaa977.

Buchfink B, Xie C, Huson DH. 2015. Fast and sensitive protein alignment using DIAMOND. *Nature Methods* **12(1)**:59–60 DOI 10.1038/nmeth.3176.

Dessimoz C, Škunca N, Thomas PD. 2013. CAFA and the Open World of protein function predictions. *Trends in Genetics* **29(11)**:609–610 DOI 10.1016/j.tig.2013.09.005.

El-Gebali S, Mistry J, Bateman A, Eddy SR, Luciani A, Potter SC, Qureshi M, Richardson LJ, Salazar GA, Smart A, Sonnhammer ELL, Hirsh L, Paladin L, Piovesan D, Tosatto SCE, Finn RD. 2019. The Pfam protein families database in 2019. *Nucleic Acids Research* **47(D1)**:D427–D432 DOI 10.1093/nar/gky995.

Falda M, Toppo S, Pescarolo A, Lavezzo E, Di Camillo B, Facchinetti A, Cilia E, Velasco R, Fontana P. 2012. Argot2: a large scale function prediction tool relying on semantic similarity of weighted gene ontology terms. *BMC Bioinformatics* **13(S4)**:S14 DOI 10.1186/1471-2105-13-S4-S14.

HMMER. 2021. HMMER: biosequence analysis using profile hidden Markov models. *Available at* http://hmmer.org/ (accessed 4 November 2020).

Huntley RP, Sawford T, Mutowo-Meullenet P, Shypitsyna A, Bonilla C, Martin MJ, O'Donovan C. 2015. The GOA database: gene ontology annotation updates for 2015. *Nucleic Acids Research* **43(D1)**:D1057–D1063 DOI 10.1093/nar/gku1113.

Kulmanov M, Hoehndorf R. 2020. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics* **36**:422–429 DOI 10.1093/bioinformatics/btz595.

Lavezzo E, Falda M, Fontana P, Bianco L, Toppo S. 2016. Enhancing protein function prediction with taxonomic constraints—the Argot2.5 web server. *Methods* **93(Suppl. 5)**:15–23 DOI 10.1016/j.ymeth.2015.08.021.

Lin D. 1998. An information-theoretic definition of similarity. In: *Proceedings of the 15th International Conference on Machine Learning*. 296–304.

Mölder F, Jablonski KP, Letcher B, Hall MB, Tomkins-Tinch CH, Sochat V, Forster J, Lee S, Twardziok SO, Kanitz A, Wilm A, Holtgrewe M, Rahmann S, Nahnsen S, Köster J. 2021. Sustainable data analysis with Snakemake. *F1000Research* **10(10)**:33 DOI 10.12688/f1000research.29032.1.

Radivojac P, Clark WT, Oron TR, Schnoes AM, Wittkop T, Sokolov A, Graim K, Funk C, Verspoor K, Ben-Hur A, Pandey G, Yunes JM, Talwalkar AS, Repo S, Souza ML, Piovesan D, Casadio R, Wang Z, Cheng J, Fang H, Gough J, Koskinen P, Törönen P, Nokso-Koivisto J, Holm L, Cozzetto D, Buchan DWA, Bryson K, Jones DT, Limaye B, Inamdar H, Datta A, Manjari SK, Joshi R, Chitale M, Kihara D, Lisewski AM, Erdin S, Venner E, Lichtarge O, Rentzsch R, Yang H, Romero AE, Bhat P, Paccanaro A, Hamp T, Kaßner R, Seemayer S, Vicedo E, Schaefer C, Achten D, Auer F, Boehm A, Braun T, Hecht M, Heron M, Hönigschmid P, Hopf TA, Kaufmann S, Kiening M, Krompass D, Landerer C, Mahlich Y, Roos M, Björne J, Salakoski T, Wong A, Shatkay H, Gatzmann F, Sommer I, Wass MN, Sternberg MJE, Škunca N, Supek F, Bošnjak M, Panov P, Džeroski S, Šmuc T, Kourmpetis YAI, van Dijk ADJ, ter Braak CJF, Zhou Y, Gong Q, Dong X, Tian W, Falda M, Fontana P, Lavezzo E, Di Camillo B, Toppo S, Lan L, Djuric N, Guo Y, Vucetic S, Bairoch A, Linial M, Babbitt PC, Brenner SE, Orengo C, Rost B, Mooney SD, Friedberg I. 2013. A large-scale evaluation of computational protein function prediction. *Nature Methods* **10(3)**:221–227 DOI 10.1038/nmeth.2340.

Reijnders MJMF, Waterhouse RM. 2019. CrowdGO: machine learning and semantic similarity guided consensus gene ontology annotation. *Bioinformatics* **34**:660 DOI 10.1101/731596.

The Gene Ontology Consortium. 2019. The gene ontology resource: 20 years and still GOing strong. *Nucleic Acids Research* **47(D1)**:D330–D338 DOI 10.1093/nar/gky1055.

The UniProt Consortium. 2019. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research* **47(D1)**:D506–D515 DOI 10.1093/nar/gky1049.

Zhou N, Jiang Y, Bergquist TR, Lee AJ, Kacsoh BZ, Crocker AW, Lewis KA, Georghiou G, Nguyen HN, Hamid MN, Davis L, Dogan T, Atalay V, Rifaioglu AS, Dalkıran A, Cetin Atalay R, Zhang C, Hurto RL, Freddolino PL, Zhang Y, Bhat P, Supek F, Fernández JM, Gemovic B, Perovic VR, Davidović RS, Sumonja N, Veljkovic N, Asgari E, Mofrad MRK, Profiti G, Savojardo C, Martelli PL, Casadio R, Boecker F, Schoof H, Kahanda I, Thurlby N, McHardy AC, Renaux A, Saidi R, Gough J, Freitas AA, Antczak M, Fabris F, Wass MN, Hou J, Cheng J, Wang Z, Romero AE, Paccanaro A, Yang H, Goldberg T, Zhao C, Holm L, Törönen P, Medlar AJ, Zosa E, Borukhov I, Novikov I, Wilkins A, Lichtarge O, Chi P-H, Tseng W-C, Linial M, Rose PW, Dessimoz C, Vidulin V, Dzeroski S, Sillitoe I, Das S, Lees JG, Jones DT, Wan C, Cozzetto D, Fa R, Torres M, Warwick Vesztrocy A, Rodriguez JM, Tress ML, Frasca M, Notaro M, Grossi G, Petrini A, Re M, Valentini G, Mesiti M, Roche DB, Reeb J, Ritchie DW, Aridhi S, Alborzi SZ, Devignes M-D, Koo DCE, Bonneau R, Gligorijević V, Barot M, Fang H, Toppo S, Lavezzo E, Falda M, Berselli M, Tosatto SCE, Carraro M, Piovesan D, Ur Rehman H, Mao Q, Zhang S, Vucetic S, Black GS, Jo D, Suh E, Dayton JB, Larsen DJ, Omdahl AR, McGuffin LJ, Brackenridge DA, Babbitt PC, Yunes JM, Fontana P, Zhang F, Zhu S, You R, Zhang Z, Dai S, Yao S, Tian W, Cao R, Chandler C, Amezola M, Johnson D, Chang J-M, Liao W-H, Liu Y-W, Pascarelli S, Frank Y, Hoehndorf R, Kulmanov M, Boudellioua I, Politano G, Di Carlo S, Benso A, Hakala K, Ginter F, Mehryary F, Kaewphan S, Björne J, Moen H, Tolvanen MEE, Salakoski T, Kihara D, Jain A, Šmuc T, Altenhoff A, Ben-Hur A, Rost B, Brenner SE, Orengo CA, Jeffery CJ, Bosco G, Hogan DA, Martin MJ, O'Donovan C, Mooney SD, Greene CS, Radivojac P, Friedberg I. 2019. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biology* **20(1)**:244 DOI 10.1186/s13059-019-1835-8.