Bioinformatics, 36(22-23), 2020, 5558–5560
doi: 10.1093/bioinformatics/btaa1052
Advance Access Publication Date: 21 December 2020
Letter to the Editor

Genome analysis

Comment on 'Hayai-Annotation Plants: an ultrafast and comprehensive functional gene annotation system in plants': the importance of taking the GO graph structure into account

Michiel Van Bel [1,2] and Klaas Vandepoele [1,2,*]

[1]Department of Plant Biotechnology and Bioinformatics, Ghent University, Ghent, Belgium and [2]Department of Plant Systems Biology, Vlaams Instituut voor Biotechnologie, Ghent, Belgium

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Contact: Klaas.vandepoele@ugent.vib.be

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on June 18, 2020; revised on September 24, 2020; editorial decision on December 7, 2020; Accepted December 9, 2020

The *Bioinformatics* article 'Hayai-Annotation Plants: an ultrafast and comprehensive functional gene annotation system in plants' (Ghelfi *et al.*, 2019), describes the Hayai-Annotation Plants tool for the functional annotation of plant genes (hereafter shortened to Hayai), together with a performance evaluation of the Hayai tool and similar software. The applied functional annotation is mostly based on the Gene Ontology (GO) framework, and GO-based performance evaluations of different functional annotation tools are reported. We agree with Ghelfi and coauthors that creating user-friendly tools and software for the functional annotation of nonmodel plant species is of prime importance in this era of ubiquitous transcriptome and genome sequencing. However, we are extremely alarmed at the evaluation methodology employed by the authors of the Hayai tool, as several critical flaws are present with regards to the assessment of (in)correct GO annotations, (not) incorporating the GO graph structure when estimating performance, and how the datasets for the evaluation were selected.

1 Introduction

GO is a structured ontology, providing a clear and unambiguous way to describe the function of gene products (Dessimoz and Škunca, 2017). The structure is provided by organizing the GO in a graph structure, where the top-level GO terms are generally quite general (e.g. '*reproduction*'), and the leaf GO terms are very specific (e.g. '*flower development*'). The implication is that the top-level GO terms are associated with many more gene products, than the bottom-level (leaf) GO terms.

The basis of various GO annotation software is fundamentally similar: (i) gene sequences are provided as input by the end-user, (ii) one or more algorithms are applied, often making use of reference databases and (iii) the software produces a list of gene identifiers annotated with GO annotations to the end-user. Given that the GO is organized in a complex graph structure, where child GO terms are more specific than their parental GO terms, the various GO annotation software packages can return the output per gene in different manners:

- Only the most specific GO terms are returned; parental GO terms are removed from the output where possible.
- For each GO term, all parental GO terms are also returned in the output.
- No effort is made to select the most specific terms, or to propagate the output to the parental terms. Thus, partial redundancy might be present.

Depending on the output per software tool, the end-user might thus need to perform additional postprocessing (filtering and/or propagation).

Importantly, when evaluating the performance of different GO annotation software tools, the graph structure needs to be used to homogenize the various outputs. This can be done either prior to the evaluation, or the homogenization needs to be an integral part of the evaluation procedure itself. The graph structure also plays a role in many of the used metrics to evaluate GO annotations: annotating a gene product with a parental GO term that is not as specific as the GO term used in the gold standard, should be seen as a lesser infraction than not annotating the gene product at all. To this end, many metrics make use of the Semantic Similarity between GO terms (Dessimoz and Škunca, 2017). Finally, an important point is that GO annotations are incomplete, and that absence of evidence is not equal to absence of function (Dessimoz and Škunca, 2017), the so-called 'Open World Assumption'. Thus, the assessment of false positives (FP) is quite difficult (Dessimoz *et al.*, 2013).

2 GO annotation evaluations

In the Hayai article (Ghelfi *et al.*, 2019), the performance of the Hayai tool is compared to that of TRAPID (Van Bel *et al.*, 2013) and BLAST2GO (Conessa and Götz, 2008), ignoring the graph structure of the GO and differences in the output of different tools.

The supplementary material of the Hayai article makes it clear that only Unix commands such as *grep* and *awk* are used to check the absence and presence of GO annotations, which will result in erroneous counts. This in turn leads to wrong estimates of both the true positives (TP), false positives (FP), and false negative (FN) rates, and subsequently also provides wrong results for the overall specificity and sensitivity of the evaluated tools.

We will demonstrate the issues in the evaluation procedure, by showcasing an example for which the GO annotation from UniProt Consortium (2018) will be considered the *gold standard*, which we compare to the GO annotation produced by any other tool. Note that this approach is in itself a simplification due to the Open World Assumption.

We can distinguish between four comparisons with regards to the propagation of GO terms to their respective parental terms in GO annotations:

I. The gold standard is not propagated, and neither is the prediction
II. The gold standard is propagated, the prediction not
III. The gold standard is not propagated, but the prediction is
IV. The gold standard is propagated, and so is the prediction

As example we use the *Arabidopsis thaliana* gene AT5G35770, which is annotated in UniProt with, among others, the GO term '*flower development*' (UniProt Consortium, 2018). Due to the graph structure, it is also annotated with all its parental terms that have the '*is_a*' relationship (Table 1).

Now, assume that the tool that we wish to evaluate annotates the example gene with the GO term '*reproductive structure development*' (GO:0048608). This GO term is clearly correct, just not the most precise and informative.

Using the Hayai tool methodology for evaluating GO annotations (see Supplementary Materials S1–S5), we can generate a table which showcases the results that would have been obtained when either propagating or not propagating the GO annotations to the respective parental terms (Methods I–IV in Table 2). When the graph structure is fully not taken into account (I), it is clear that the predicted GO annotation is flagged as wrong, while the expected result is flagged as missing. When the graph structure is taken into account for the gold standard, but not for the GO predictions (II), it is clear that the number of FN is very high due to the increased number of GO annotations in the gold standard. When the graph is not taken into account for the gold standard, but it is for the GO predictions (III), it is clear that the number of FP is very high due to the many predicted GO annotations, as mirrored in (II). When the DAG is taken into account for both the gold standard and the GO predictions (IV), then artificially high numbers of TP and FN are achieved, with the balance depending on the depth of the GO term reported by the tool annotation.

Table 1. GO annotations for gene AT5G35770

GO accession	GO name
GO:0008150*	Biological process
GO:0000003*	Reproduction
GO:0032502*	Developmental process
GO:0022414*	Reproductive process
GO:0032501*	Multicellular organismal process
GO:0048856*	Anatomical structure development
GO:0007275*	Multicellular organismal development
GO:0003006*	Developmental process involved in reproduction
GO:0048731*	System development
GO:0061458*	Reproductive system development
GO:0048367*	Shoot system development
GO:0048608*	Reproductive structure development
GO:0009791*	Postembryonic development
GO:0090567*	Reproductive shoot system development
GO:0009908	Flower development

Note: GO annotations for gene AT5G35770. The order of the GO terms is determined by the depth of the GO term in the DAG. Propagated terms, when starting from the GO accession GO:0009908, are indicated with a *. Source data: UniProt.

Table 2. Results when using Hayai methodology

Method	Gold standard	Tool	TP	FP	FN
I	Not propagated	Not propagated	0	1	1
II	Propagated	Not propagated	1	0	14
III	Not propagated	Propagated	0	11	1
IV	Propagated	Propagated	12	0	3

Note: Results when using the Hayai methodology for evaluating GO annotations, with the gold dataset containing the single annotation GO:0009908, and the test dataset containing the single annotation GO:0048608.

TP, true positives; FP, false positives; FN, false negatives.

This makes it very clear that not taking the graph structure into account leads to erroneous conclusions, but simply propagating GO terms is also not an optimal solution when solely using *awk* and *grep* for testing the absence and presence of GO annotations. The core issue is that these Unix commands can only be used to perform string-matching operations on a line-by-line basis, forgoing the complexity of the GO structure. Various solutions have been proposed for the problem of evaluating GO annotations (Dessimoz *et al.*, 2013), with the usage of GO semantic similarity metrics being one of the more prominent (Dessimoz and Škunca, 2017).

Coming back to the Hayai article (Ghelfi *et al.*, 2019), Table 1 in its manuscript details the TP, FP, and FN counts for Hayai, Blast2GO, and TRAPID. The extremely high FP counts for TRAPID should be seen in the light of case III of Table 2: the UniProt-GOA data are not propagated to the parental terms, while the TRAPID data are. Additionally, the relatively (compared to Blast2GO and Hayai) lower TP rate of TRAPID compared to Hayai is a side effect of TRAPID often reporting a parental term of the annotation from the UniProt-GOA gold standard, rather than the most specific. While this can indeed be construed as TRAPID having a lower sensitivity, an optimal scoring system should not report single Boolean values for GO annotations. Chapter 10 in the GO handbook (Dessimoz and Škunca, 2017) titled '*Community-Wide Evaluation of Computational Function Prediction*' goes more in-depth in how to perform the proper evaluation.

Another important take-home message from the CAFA challenge (Radivojac *et al.*, 2013) is the need for the performance evaluations for computational predictions to be split up over the three different aspects that make up the GO: molecular function, cellular component, biological process. Each of the aspects represent different challenges and performance. For example, in the recent CAFA3 challenge (Zhou et al., 2019), the performance for predicting molecular function and cellular component GO terms is much better than the one for predicting the biological process GO terms. Averaging the data over all three aspects will give rise to drawing incorrect conclusions.

3 Separation of test versus training data

Another issue that is present in the 'Hayai-Annotation Plants' article is that the data used to build the models used in the software, contains the same information as the data used to evaluate the tool. As training data for the Hayai tool the entirety of UniProt Consortium (2018), which includes *A.thaliana*, is used. Subsequently the tool is evaluated by looking at the annotation of *A.thaliana* genes.

An optimal solution would be to have independent test datasets, or make use of a data freeze, such as during the CAFA challenge (Radivojac *et al.*, 2013). If this is not a feasible approach, then cross-validation can be applied to train the software on a subset of the data, and evaluate the software using another subset of the data (Dangeti, 2017).

4 Conclusion

We find major issues with how the evaluation of the Hayai tool was performed. Not separating the testing and evaluation data, or using solutions such as cross-validation, is only a minor problem in the evaluation procedure. The much greater problem seems to be an incorrect interpretation of how the GO is organized, and how GO predictions should be evaluated. We acknowledge being a party of interest, as we are the authors of the TRAPID tool which was compared to the Hayai tool. However, the issues we bring forward are generic enough for them to be accounted for in all functional prediction software.

Acknowledgements

We thank François Bucchini for helpful discussions.

Funding

This work has been supported by Ghent University and VIB.

Conflict of Interest: The authors of the article are also authors of the TRAPID tool, which was compared to the Hayai tool.

References

Conesa,A. and Götz,S. (2008) Blast2GO: a comprehensive suite for functional analysis in plant genomics. *Int. J. Plant Genomics*, **2008**, 1–12.

Dangeti,P. (2017) *Statistics for Machine Learning*. O'Reilly, Sebastopol, CA, USA.

Dessimoz,C. and Škunca,N. (2017) *The Gene Ontology Handbook*. Humana Press, New York, NY, USA.

Dessimoz,C. *et al.* (2013) CAFA and the Open World of protein function predictions. *Trends Genet.*, **29**, 609–610.

Ghelfi,A. *et al.* (2019) Hayai-Annotation Plants: an ultra-fast and comprehensive functional gene annotation system in plant. *Bioinformatics*, **35**, 4427–4429.

Radivojac,P. *et al.* (2013) A large-scale evaluation of computational protein function prediction. *Nat. Methods*, **10**, 221–227.

UniProt Consortium (2018) UniProt: the universal protein knowledgebase. *Nucleic Acids Res.*, **46**, 2699.

Van Bel,M. *et al.* (2013) TRAPID: an efficient online tool for the functional and comparative analysis of de novo RNA-Seq transcriptomes. *Genome Biol.*, **14**, R134.

Zhou,N. *et al.* (2019) The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol.*, **20**, R244.