Formalization of Gene Ontology relationships with factor graph towards Biological Process prediction

F. Spetale*[1], P. Bulacio[1], F. Krsticevic[1], S. Ponce[2] and E. Tapia[1]

1 CIFASIS-Conicet Institute, Bv. 27 de Febrero 210 Bis, Rosario, Argentina.
2 Facultad Regional San Nicolás, Colón 332, Universidad Tecnolgica Nacional, Argentina.
* spetale@cifasis-conicet.gov.ar

Abstract— **Gene Ontology is a hierarchical controlled vocabulary for protein annotation. Its synergy with automatic classification methods, ensemble, has been widely used for the prediction of protein functions. Current classification methods use only the relation *is_a* and a few little *part_of* to generate prediction model. In this work we formalize the GO *part_of*, *regulates*; *negatively_regulates* and *positively_regulates* relationships through predicate logic. This formalization is incorporated within an ensemble method based on graph factor called *Factor Graph GO Annotation*. The proposed model is validated against four model organisms for GO Biological Process prediction.**

Keywords— **Gene Ontology, Factor Graph, Automatic function prediction**

I Introduction

The high-throughput of sequencing technologies provides huge amounts of data opening unlimited opportunities for better understanding of biological behavior of target organisms. The use of machine learning methods may achieve the initial approach for data analysis focalizing experiments, saving time and money. A central point of genomic research is to establish the biological functions of proteins, also called annotation. *Gene Ontology* (GO) provides a hierarchical architecture of biological functions [1] which may guide the automatic annotation of protein function. GO is composed of three sub-ontologies: Biological Process (BP), Molecular Function (MF) and Cellular Component (CC). Each of them is a Directed Acyclic Graph (DAG), where every node represents a GO-term (a biological function) and every edge represents a relationship between two GO-terms. The commonly used relationships in GO are: *is_a* (is a subtype of); *part_of*; *regulates*; *negatively regulates* and *positively regulates* [2]. Traditional ensemble methods for automatic function prediction based on GO consider the relationship *is_a* [3], [4], [5] and a few the relationship *part_of* [6].

In this paper, we propose the formalization of GO relationships beyond *is_a* for GO-BP prediction. Regarding inference process interpretability, a classification method based on factor graph [7] is considered. In particular, we use the *Factor Graph GO Annotation* (FGGA) [8] which models GO relationships with logical factor nodes. The formalization must consider TPG constraint, *"If the child GO-term describes the protein, then all its parent terms must also apply to that protein; and if a GO-term not describes a protein, then all its descendant GO-terms must not describe it"*, that governs the structure and inference within GO-DAG. The extension of logical factor nodes within FGGA model, hereafter FGGA$^+$, is able to infer functional predictions of proteins by using the adapted version of sum-product algorithm [8].

This paper is organized as follows. In Section II, GO relationships are formalized thought predicate logic to be included to FGGA$^+$. Section III discusses the results on *A. thaliana*, *D. melanogaster*, *D. rerio*, and *C. elegans* in BP-GO. In the last Section, conclusions are presented.

II Method

Given a GO subgraph, GO-terms *GO:i* are mapped to binary-valued latent variable nodes x_i of FGGA$^+$. Relationships between GO-terms are mapped to logical factor nodes f_k which describe valid *GO:i* configurations under the TPG constraint; and probabilistic factor nodes g_i which model statistical dependence between latent variable nodes (ideal) x_i and variable leaf nodes y_i modeling observable (real), i.e., uncertain in *GO:i* term predictions (see Fig. 1).

Practically, logical factor nodes f_k are implemented with truth tables of $2^{\#child+\#parents}$ entries. At each of these entries, the specific parent/child role and relationships of participating variable nodes are required to check the TPG constraint. As shown in Table 1, where 1/0 denotes positive/negative annotation, respectively. The logical factor f_4 in Fig. 1-b ensures that TPG constraint over variable nodes x_3, x_4 and x_5 is fulfilled whenever x_5 is a child node of x_3 (x_5 *regulates* x_4) and x_4 (x_5 *part_of* x_4), i.e., multiple inheritance over x_5.

Formally, logical factor nodes f_k over subsets of variable nodes x_i ensure the local satisfiability of TPG constraint. With this aim, two logical rules are repeatedly evaluated. Specifically, if a child GO-term is annotated positive, then its parent GO-term(s) must also be annotated positive. On the other hand, if a parent GO-term is annotated negative,

I. Torres et al. (eds.), *VII Latin American Congress on Biomedical Engineering CLAIB 2016, Bucaramanga, Santander, Colombia, October 26th -28th, 2016*,
IFMBE Proceedings 60,
DOI: 10.1007/978-981-10-4086-3_15

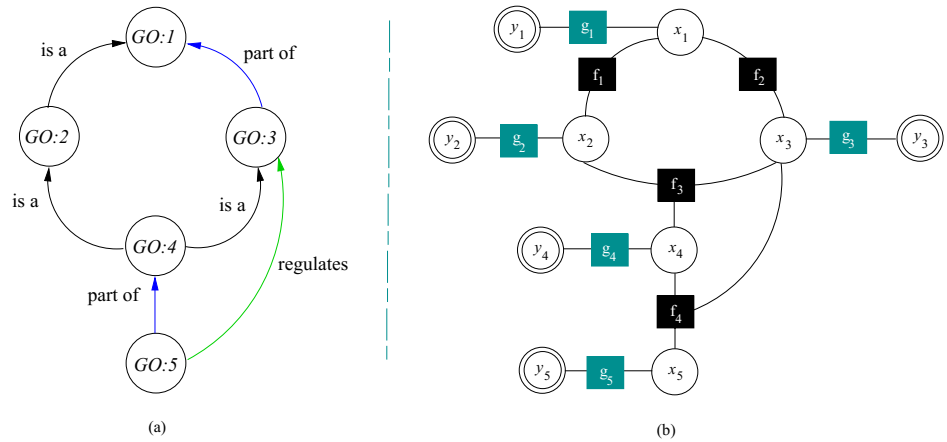

Fig. 1: (a) GO-DAG where *GO:i* nodes are GO-terms and edges are relationships (b) FGGA$^+$ model where x_i are latent variable nodes modeling actual positive/negative *GO:i* annotations and f_k are logical factor nodes modeling the TPG constraint over them, y_i are observable variable leaf nodes modeling real-valued *GO:i* predictions and g_i are probabilistic factor nodes modeling their statistical dependence on latent variable nodes x_i.

Table 1: The truth table of the logical factor node f_4. Positive/negative annotations of variable nodes x_3, x_3 and x_5 are depicted as 1/0. Parent variable nodes x_3 and x_4 are shown in blue.

x_3 [*regulates*]	x_4 [*part_of*]	x_5	$f_4(x_3,x_4,x_5)$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

then its children GO-term must also be annotated negative. In addition, they must fulfill a requirement of transitive inference on their grandparents. Using predicate logic [9], let $part\_of(GO{:}j, GO{:}i)$ denotes *GO:j* (child) is part of *GO:i* (parent) and $is\_a(GO{:}i, GO{:}z)$ denotes *GO:z* is parent of *GO:i* (child). Similarly, let $annotated(\cdot)$ denotes the positive annotation of the target protein with a GO-term. As a result, at least one of the following rules (Eq.1 or Eq.2) must be active and fullfilled by any pair of GO-terms involved within a *part_of* relationship:

$$r_1 : \forall i,j,z \;\; part\_of(GO{:}j, GO{:}i) \wedge annotated(GO{:}j) \wedge [is\_a(GO{:}i, GO{:}z) \vee part\_of(GO{:}i, GO{:}z)] \rightarrow annotated(GO{:}i) \quad (1)$$

$$r_2 : \forall i,j,z \;\; part\_of(GO{:}j, GO{:}i) \wedge \neg annotated(GO{:}i) \wedge [is\_a(GO{:}i, GO{:}z) \vee part\_of(GO{:}i, GO{:}z)] \rightarrow \neg annotated(GO{:}j) \quad (2)$$

In the same way, we can extend the predicate logic to *regulates* relationships:

$$r_3 : \forall i,j,z \;\; reg\_GO(GO{:}j, GO{:}i) \wedge annotated(GO{:}j) \wedge [is\_a(GO{:}i, GO{:}z) \vee part\_of(GO{:}i, GO{:}z)] \rightarrow annotated(GO{:}i) \quad (3)$$

$$r_4 : \forall i,j,z \;\; reg\_GO(GO{:}j, GO{:}i) \wedge \neg annotated(GO{:}i) \wedge [is\_a(GO{:}i, GO{:}z) \vee part\_of(GO{:}i, GO{:}z)] \rightarrow \neg annotated(GO{:}j) \quad (4)$$

where $reg\_GO(GO{:}j, GO{:}i)$ can be just *regulates* or *positive/negative regulation*.

When multiple inheritance exists, multiple relationships must be considered in both, GO and FGGA$^+$ sides. For instance, Table 1 shows that "x_5 is the child of x_3" and "x_5 is also child of x_4", considering the *regulates* relation between *GO:3* and *GO:4*, and the *part_of* relation between *GO:4* and *GO:5*. For instance, row 1 shows the fulfillment of both relationships: *part_of* and *regulates*, by rule 2 and rule 4 activation, hence, f_4 is true. On the other hand, row 4 shows for these relationships, rule 1 and rule 3 are active but only rule 1 is fulfilled, hence, f_4 is false. Note that the modeling of GO relationships by predicate logic requires a detailed examination of cascade GO relationships to accomplish transitiveness.

III Results and Discussion

A Experimental Protocol

Four models organisms, *D. rerio* [10], *A. thaliana* [11], *C. elegans* [12] and *D. melanogaster* [13] are considered. For

Table 2: Datasets in the GO-BP

Organism	# GO-terms	# Samples
D. rerio	44	1002
A. thaliana	97	6032
C. elegants	112	3223
D. melanogaster	156	4189

Table 3: GO-BP prediction performance, Hierarchical Precision (HP), Hierarchical Recall (HR), Hierarchical F-score (HF)

Organism	HP	HR	HF
D. rerio	0.66	0.72	0.66
A. thaliana	0.52	0.68	0.57
C. elegants	0.56	0.76	0.63
D. melanogaster	0.59	0.75	0.64

each organism, GO-BP annotation datasets (see Table 2) are generated with experimental GO evidence codes[1]: inferred from mutant phenotype (IMP), inferred from genetic interaction (IGI), inferred from physical interaction (IPI), inferred from expression pattern (IEP) and inferred from direct assay (IDA), considering GO-terms with at least 300 positively annotated proteins. To balance the training dataset [14] of each GO-term, the number of positive and negative samples must be the same. The negative annotated samples are selected by the *inclusive* separation policy [15]. The protein characterization to a fixed number of input features is done by 457 physicochemical/secondary structure properties, Physicochemical$^+$, 453 of the physicochemical type [16] and 4 of the secondary structure [17]. Practically, protein characterization is implemented with the Bio.SeqsUtils [18] package. FGGA$^+$ method is built from GO-term classifiers implemented with SVM default constant complexity C=1. The Gaussian assumption in FGGA$^+$ is attained by real valued predictions of SVM soft-margin outputs (implemented with e-1071 R package [19]).

The FGGA$^+$ is evaluated with 5-fold cross-validation test, computing per GO-term the AUC average scores [20]. Taking into account that GO annotation gets harder as deeper levels of the hierarchy [21], prediction performance was measured by the hierarchical precision (HP), the hierarchical recall (HR), and the hierarchical balanced F-score (HF) reflecting their trade-off.

B Prediction performance on model organisms

Whatever the organism, FGGA$^+$ improves the SVM baseline classifiers. This is particularly evident in the annotation of *D. melanogaster* and *C. elegants*, see Fig. 2.

All relationships modeled in this paper are presented in the Fig. 3 and shows the GO-DAG the annotated sequence "ENSDARP00000061793" of the *NR1H4* gene in *D. rerio*. This gene is related to the hormone nuclear receptor family members and encodes a nuclear receptor for bile acids ENSDARP00000061793 protein which regulates the expression of genes involved in bile acid synthesis. The *is_a* consideration in the GO-BP activate two novel and specific terms, GO:0050794, regulation of cellular process, and GO:0044700, single organism signaling. By including relations *part_of* and *regulates* within GO-BP (see Fig. 4) allow the annotation of three new terms, the more specific term GO:0007165, signal transduction, which is part of terms GO:0007154, cellular communication, and GO:0051716, cellular response to stimulus.

Enrichment through this three new nodes indicates that probably the *NR1H4* gene is involved in the regulation of a cellular process, in this case a bile acid synthesis. It function is also related to the response to a stimulus, in this case the presence of bile, and to transduction signal within the cell, in this case expression of genes involved in bile production. The new prediction enriched of the GO term GO:0044700 results in the biological sense acquisition.

The performance of GO-BP prediction by FGGA$^+$ is presented in Table 3. The results show a good F-score independent of the number of GO-terms and organism complexity.

IV Conclusions

The formalization of the GO relationships within FGGA$^+$ allows a hierarchical and consistent prediction of GO-terms within any of the three sub-ontology GO (BP, MF or CC) achieving deeper, broader, and more jumping edges of predicted DAGs. This approach may be extended to another types of no transitive relationships which are in development, such as *capable of* and *occurs in*[2].

Conflict of Interest

"The authors declare that they have no conflict of interest."

Acknowledgements

The authors were supported by the project PID INI 3600 - Facultad Regional de San Nicolás, Argentina.

References

1. Consortium Gene Ontololy. Creating the gene ontology resource: design and implementation *Genome Res.*. 2001;11:1425-1433.
2. Consortium The Gene Ontology. The Gene Ontology in 2010: extensions and refinements *Nucleic Acids Research.* 2010;38:D331-D335.

[1] http://geneontology.org/page/guide-go-evidence-codes

[2] ftp://ftp.geneontology.org/pub/go/www/GO.draft-page.shtml

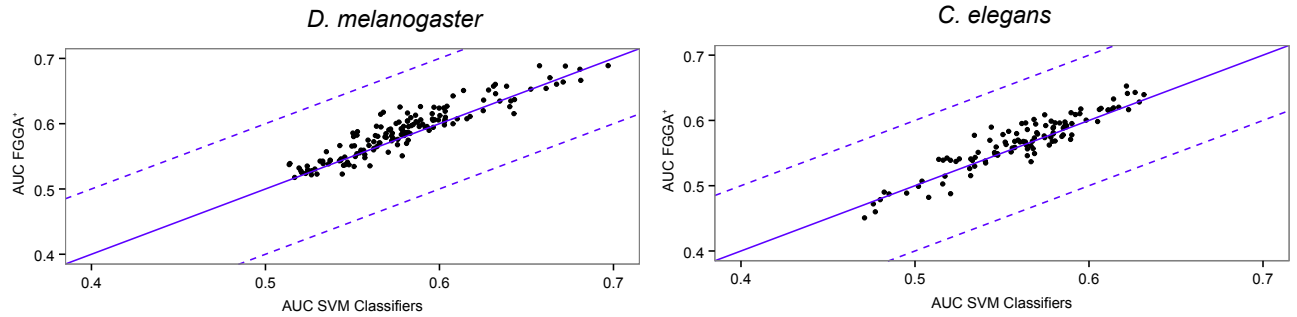

Fig. 2: Scatter-plot of the average AUC of base SVM vs. FGGA$^+$ GO-BP predictions on *D. melanogaster* (left) and *C. elegans* (right) with Physicochemical$^+$ characterization.

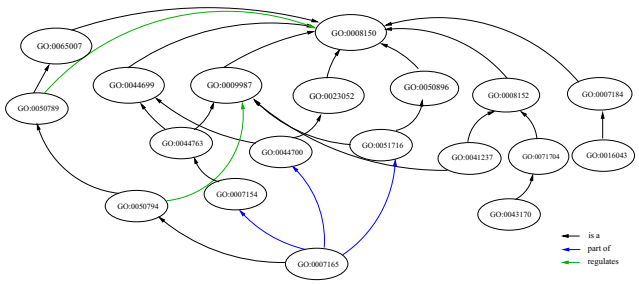

Fig. 3: GO-DAG of a *D. rerio* annotated sequence "ENSDARP00000061793"

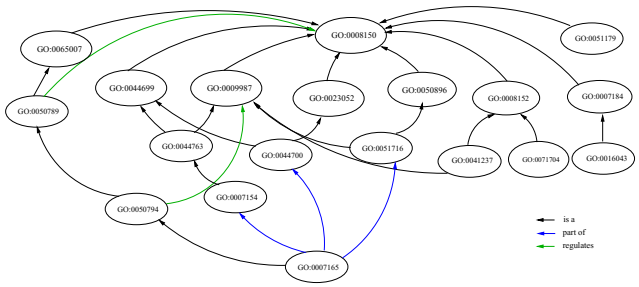

Fig. 4: GO-DAG of a *D. rerio* predicted sequence using FGGA$^+$

3. Barutcuoglu Zafer, Schapire Robert E., Troyanskaya Olga G.. Hierarchical multi-label prediction of gene function *Bioinformatics.* 2006;22:830-836.
4. Valentini Giorgio. True Path Rule Hierarchical Ensembles for Genome-Wide Gene Function Prediction *Computational Biology and Bioinformatics, IEEE/ACM Transactions on.* 2011;8:832-847.
5. Sykacek Peter. Bayesian assignment of gene ontology terms to gene expression experiments *Bioinformatics.* 2012;28:i603-i610.
6. Cheng Liangxi, Lin Hongfei, Hu Yuncui, Wang Jian, Yang Zhihao. Gene Function Prediction Based on the Gene Ontology Hierarchical Structure *PLoS ONE.* 2014;9:e107187.
7. Kschischang Frank R., Frey Brendan J., Loeliger Hans-Andrea. Factor Graphs and the Sum-product Algorithm *IEEE Trans. Inf. Theor.*. 2001;47:498-519.
8. Spetale F.E., Tapia E., Krsticevic F., Roda F., Bulacio P.. A Factor Graph Approach to Automated GO Annotation *PLoS ONE.* 2016;11:1-16.
9. Burger Albert, Davidson Duncan, Baldock Richard A.. Formalization of mouse embryo anatomy *Bioinformatics.* 2004;20:259-267.
10. Carlson Marc. Genome wide annotation for Zebrafish 2016. Version: 3.0.0, Accessed: 2016-04-06.
11. Carlson Marc. Genome wide annotation for Arabidopsis 2016. Version: 3.0.0, Accessed: 2016-04-06.
12. Carlson Marc. Genome wide annotation for Worm 2016. Version: 3.0.0, Accessed: 2016-04-06.
13. Carlson Marc. Genome wide annotation for Fly 2016. Version: 3.0.0, Accessed: 2016-04-06.
14. Wei Qiong, Dunbrack Roland L.. The role of balanced training and testing data sets for binary classifiers in bioinformatics. *PloS one.* 2013;8.
15. Eisner Roman, Poulin Brett, Szafron Duane, Lu Paul, Greiner Russ. Improving protein function prediction using the hierarchical structure of the Gene Ontology in *Proc. IEEE CIBCB*:1-10 2005.
16. Lee Bum, Shin Moon, Oh Young, Oh Hae, Ryu Keun. Identification of protein functions using a machine-learning approach based on sequence-derived properties *Proteome Science.* 2009;7:27.
17. Chou Peter Y., Fasman Gerald D.. Prediction of protein conformation *Biochemistry.* 1974;13:222-245.
18. Sicheritz-Ponten Thomas, Alsmark Cecilia. Package SeqUtils 2002. Second Version, Accessed: 2015-09-02.
19. Meyer David, Dimitriadou Evgenia, Hornik Kurt, Weingessel Andreas, Leisch Friedrich. Misc Functions of the Department of Statistics (e1071), TU Wien 2014. Version: 1.6-4, Accessed: 2015-09-02.
20. Fawcett Tom. An Introduction to ROC Analysis *Pattern Recogn. Lett.*. 2006;27:861-874.
21. Verspoor Karin, Cohn Judith, Mnizewski Susan, Joslyn Cliff. A categorization approach to automated ontological function annotation *Protein Science.* 2006;15:1544-1549.