exp2GO: improving prediction of functions in the Gene Ontology with expression data

Leandro Di Persia,Tiago Lopez, Agustin Arce,
Diego H. Milone and Georgina Stegmayer

April 18, 2022

Abstract

The computational methods for the prediction of gene function annotations aim to automatically find associations between a gene and a set of Gene Ontology (GO) terms describing its functions. Since the hand-made curation process of novel annotations and the corresponding wet experiments validations are very time-consuming and costly procedures, there is a need for computational tools that can reliably predict likely annotations and boost the discovery of new gene functions. This work proposes a novel method for predicting annotations based on the inference of GO similarities from expression similarities. The novel method was benchmarked against other methods on several public biological datasets, obtaining the best comparative results. exp2GO effectively improved the prediction of GO annotations in comparison to state-of-the-art methods. Furthermore, the proposal was validated with a full genome case where it was capable of predicting relevant and accurate biological functions. The repository of this project withh full data and code is available at https://github.com/sinc-lab/exp2GO

1 Introduction

In the current post-genomics era the inference of the functions associated with genes and their products, the proteins, is a necessary step for better understanding the development of living organisms [1]. As the number of sequenced genomes rapidly grows, the overwhelming amount of newly discovered genes can only be annotated initially by computational methods, which must provide a reasonable trade-off between precision and recall.

The functions of genes are generally described with terms or annotations of the Gene Ontology (GO) [2]. GO provides concepts organized in a structured set, which are systematically used to describe or annotate genes. Terms farther from the root node of the ontology describe more specific concepts, whereas terms closer to the root describe high-level abstract or generic concepts. The proximity between two terms in an ontology is named semantic similarity, and can be thought of as the extent to which they share information. This similarity information, which is contained in the level of specificity of the term that subsumes them (a common ancestor node), can be measured as the information content (IC) of this subsumer. As the specificity of the subsumer term raises, the IC is higher [3].

Computational methods for predicting candidate gene annotations are needed due to the fact that in-vitro biomolecular experiments to validate a gene function are costly, laborious and time consuming. In the last 15 years, several works have dealt with the functional annotation of genes. One of the first approaches was mining the literature to extract keywords that were then mapped to GO concepts [4]. Another early proposal was based on training a hierarchy of support vector machines for each gene, to obtain the most probable set of corresponding predictions [5]. An alternative model was a k-nearest neighbour (KNN) classifier, used for predicting annotations based exclusively on the existing GO terms of the nearest genes [6]. A more recent work [7] proposed learning more annotations according to patterns from available annotation profiles with decision trees and bayesian networks. Although these methods worked satisfactorily well in its own application domain, most of them did not take into account the structural organization of the ontology.

More recently, the ontology structure was taken into consideration for predicting the functions of partially annotated proteins by a random walk algorithm [8] and semantic kernels [9]. Alternatively, gene functions were extrapolated based on a singular value decomposition of a binary matrix built from the set of known annotations [10]. However, those methods were incapable of predicting the function of fully unannotated genes because they require an initial set of well-known annotations. Moreover, none of these methods considered expression data, which is a very valuable source of information for genes that share the same function. One method that considered expression data appeared in [11], where a data fusion technique based on matrix factorization was proposed to take advantage of multiple heterogeneous data sources, such as a binary matrix with GO terms and a matrix of expression data. Following this approach, a matrix factorization was proposed for the binary GO annotations matrix, adding terms to genes which are partially annotated and then using a KNN classifier for assigning GO terms to closer genes [12].

A widely cited study evaluated a very large number of methods representing the state-of-the-art in protein function prediction at that time with a benchmarking set of several organisms, named Critical Assessment of protein Function Annotation algorithms (CAFA challenge) [1, 13]. The study concluded that although the top methods performed well [14], there is still considerable room for improvement over currently available tools for such automatic annotation tasks. This can be due to several facts. On the one hand, existing approaches do not exploit the richness of the structural and hierarchical organization inherent to the ontology. That is, the GO terms location within this hierarchy, its position and distance in the ontology is not fully considered in the models. On the other hand, expression data being one of the most useful and specific information regarding the biological function is not used. The methods involved in the CAFA challenge provide annotation for proteins, where it is very difficult to have expression data. However, for genes there are lots of expression datasets available. That is why in this work we focus on the functional annotation at gene level, being therefore able to use at the same time, expression and semantic information for improving predictions.

The semantics of Gene Ontology can be effectively captured and taken into account through semantic measures. There are several semantic similarity measures available for comparing biological terms that annotate genes in an ontology [15]. Any of these measures can be used to build a semantic similarity matrix among all genes of an experiment in order to, for example, cluster genes. However, for the case of novel and unannotated genes, this matrix will have many empty rows and columns. Completing this missing data is essential in order to be able to infer a function for those genes, for which there is not any semantic information available. These empty spaces in the matrix could be completed from expression information related to all the genes. In this work, we propose a novel method to complete the semantic distance matrix with expresion distances by using non-negative matrix factorization (NMF) [16]. After semantic matrix completion, the GO terms can be inferred using a Bayesian approach.

The method proposed in this work, exp2GO, uses expression data of genes and takes into account the ontology hierarchy to provide predictions for completely unknown genes. Therefore, it can be useful in the case of: i) a model genome, when there is still a minority of unannotated genes; ii) a partially studied genome, when sequencing data becomes available but, in the meantime, annotation mostly based on homology with similar genomes has been done; iii) and when there is a de novo sequenced genome initially annotated by homology, and then these annotations are refined and extended when transcriptomics data become available.

2 The exp2GO method

We propose a novel pipeline for inferring GO terms for unannotated genes (Figure 1). Starting from expression data (Figure 1A), the matrix d_E with the pairwise expression distances among the expressions of all the genes under study is calculated (Figure 1B). This is a full matrix, without missing rows or columns. In the distance matrix d_E, different expression experiments could be considered. That is, if the prediction of a function related to a specific experimental condition for the organism under study is desired, then only expression data of this condition should be used. Instead, if a wider function prediction is desired, not just in the context of a specific experiment, all available expression data can be used. In any case, the method will use the distance between the expression data provided.

The GO annotations for well-known genes in the study are available; although there are some genes that are completely unknown (Figure 1C), that is, without GO terms associated. Among all these genes, a semantic distance matrix d_{GO} can be calculated (Figure 1D) using any of the already available

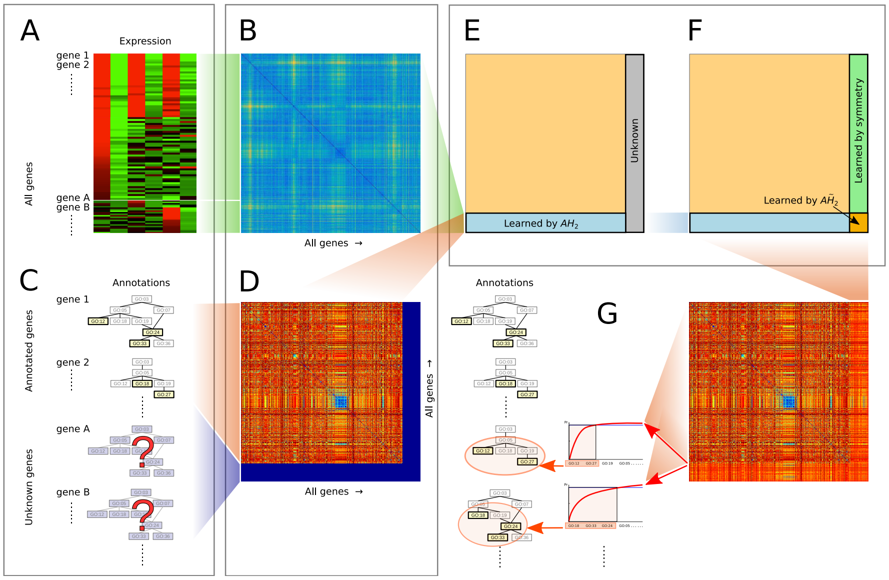

Figure 1: Pipeline of exp2GO for inferring GO labels. A) Expression data. B) Expression pairwise expression distance matrix d_E among all genes in the study. C) Semantic data: GO annotations for well-known genes (gene 1, gene 2, ...); some genes in the study are completely unknown (gene A, gene B, ...). D) Semantic distance matrix d_{GO} among all genes: with missing rows and columns because many genes are not semantically annotated. E) and F) exp2GO completes the missing distances in d_{GO}, using the information available in B) and D). G) Once the d_{GO} matrix is completed, GO annotations are assigned according to the reconstructed semantic space. From the closest set of genes with known GO terms, the potential terms (according to a Bayesian model) are sorted in descending order by their posterior probability. Finally, the candidate GO terms with the highest accumulated probability (in yellow) are assigned to each unannotated gene (gene A, gene B, ...).

semantic measures [15]. However, it should be noted that there will be missing rows and columns in this matrix because many genes are not semantically annotated, then a semantic distance among them cannot be calculated. With exp2GO it is possible to complete the missing distances in d_{GO}, by using information available in the full matrix d_E and in the partial matrix d_{GO} (Figure 1E-F). The proposed NMF approach completes just the last rows of the d_{GO} matrix. After its reconstruction, the symmetric part in the corresponding columns is completed by copying the information of the completed rows (and transposing it), thus forcing the symmetry. Once the d_{GO} matrix is completed, GO annotations are assigned according to the reconstructed semantic space. The potential labels are obtained from the set of genes with known GO labels according to a Bayesian model, by sorting the labels in descending order according to their posterior probability (Figure 1G). Finally, the candidate GO labels with the highest accumulated probability are assigned to each unannotated gene. The NMF reconstruction of d_{GO} and the GO labels assignment are explained in detail in the next subsections.

2.1 Reconstruction of d_{GO}

Both matrices (d_{GO} and d_E) are non-negative pairwise distances among genes, although measured in different spaces. The main idea for completing the d_{GO} matrix is to learn a basis that is good enough to represent jointly the d_E and the known part of d_{GO}. This basis learning can be done by using a weighted and coupled NMF algorithm, as follows.

In standard NMF, a data matrix X is decomposed in the product of two matrices A and H, such as [17]

$$X = AH + E, \tag{1}$$

where X of n-by-m contains the original non-negative matrix, A of n-by-p is a non-negative matrix that can be interpreted as a dictionary of atoms for synthesizing X, and H of p-by-m has the coefficients to reconstruct X by linear combination of the atoms of A. The inner dimension p is the dimension allowed for the dictionary. If this value is lower than the original data subspace dimension, there will be an error in the representation, which is measured in matrix E.

In our case we have two data matrices, d_E and d_{GO} that we want to express using the same dictionary A, but each will have its own coefficient matrix. Thus, we want to construct

$$\begin{aligned} d_E &= AH_1 + E_1, \qquad (2) \\ d_{GO} &= AH_2 + E_2. \qquad (3) \end{aligned}$$

In these equations, both d_E and d_{GO} are of n-by-n (as well as E_1 and E_2), with n the number of genes, thus A is of n-by-p and H_1 and H_2 are of p-by-n.

Other types of biological data could be used instead of, or additionally to, expression, in order to provide meaningful similarity matrices such as for example co-evolutionary data, phylogenetics, or co-occurrence in pathways. Replacing expression data is direct. Adding other data sources is very straightforward, it would just require to add equations like (3) in the cost function.

To learn the matrices A, H_1 and H_2, the following cost function will be minimized [18]:

$$J(A, H_1, H_2) = ||d_E - AH_1||_F^2 + \lambda ||d_{GO} - AH_2||_F^2, \tag{4}$$

where $|| \cdot ||_F^2 = \sum_i \sum_j \cdot_{ij}^2$ is the squared Frobenius norm. The λ parameter regulates the relative proportion of information that is used from each term. A large value of λ will produce a dictionary to approximate the d_{GO} matrix, but allowing for worst performance in approximation of d_E, and vice-versa.

To take into account that some entries of d_{GO} matrix are unknown we need to add a weight matrix W before calculating the Frobenius norm of the approximation semantic error, as follows [19, 18]

$$J(A, H_1, H_2) = ||d_E - AH_1||_F^2 + \lambda||W \odot (d_{GO} - AH_2)||_F^2, \tag{5}$$

where W is an n-by-n matrix defined by $[W]_{ij} = 1$ if $[d_{GO}]_{ij}$ is known, and $[W]_{ij} = 0$ otherwise; A is initialized to random positive numbers in $[0, 1]$; and $\odot$ represents an element-by-element product. This cost function is minimized by an iterative process.

Another alternatives to symmetric matrix decomposition in the NMF context can be considered. In the first place, we could use a symmetric factorization of the kind $d = UU^T$, with a cost function like $J = ||d_E - UU^T||_F^2 + \lambda||W \odot (d_{GO} - UU^T)||_F^2$ [20]. For this setup, given that d_E and d_{GO} may have different scaling ranges, the model would have poor capabilities to match both matrices at the same time, since a common approximation is required (i.e. UU^T). Therefore, we would need to use a tri-factorization of the kind $d = USU^T$, and a cost function like $J = ||d_E - US_1U^T||_F^2 + \lambda||W \odot (d_{GO} - US_2U^T)||_F^2$, with diagonal scalings S_1 and S_2 [21]. For this cost function, the information used for the reconstruction of the unknown rows in d_{GO} can only be extracted from the corresponding rows of d_E. That is, US_2U^T can take any value in the last rows because d_{GO} (and W) have zeros for the unknown genes. Then, the last rows of U can only be learned from the last rows of d_E and in these rows only the scaling factors in S_2 can model differences between d_{GO} and d_E. This way we would not be exploiting the information from the known part of d_{GO}, and no fusion would be done in the reconstructed rows. Conversely, the proposed model (5) takes full advantage of the known parts from both distance matrices. In this setup, the information of the common dictionary A is propagated to both H_1 and H_2 and thus the reconstructed part for the unknown rows of d_{GO} will have effectively contributions from both d_E and d_{GO}, producing a more effective fusion. To show this, an experiment comparing the symmetric tri-factorization with diagonal scalings and our approach was performed with real data (Supplementary Figure 1).

Assuming known H_1 and H_2, the update for A is

$$\begin{aligned} A \;&= A \;\; \odot \left(d_E H_1^T + \lambda(W \odot d_{GO})H_2^T\right) \\ &\qquad\;\; \oslash \left((AH_1)H_1^T + \lambda(W \odot (AH_2))H_2^T + \epsilon\right), \end{aligned} \tag{6}$$

where $\epsilon = 0.0000001$ is a small constant to avoid division by zero, and $\oslash$ represents element by element division. Once updated, each column of matrix A is normalized to have unitary 2-norm.

Then, with the updated A taken as fixed, H_1 and H_2 are updated by

$$H_1 = H_1 \odot \left((A^T d_E) \oslash (A^T(AH_1) + \epsilon)\right), \tag{7}$$

$$H_2 = H_2 \odot \left((A^T(W \odot d_{GO})) \oslash (A^T(W \odot (AH_2)) + \epsilon)\right), \tag{8}$$

where H_1, H_2 are initialized to random positive numbers in $[0, 1]$. This iteration (through (6), (7) and (8)) is repeated until convergence, detected by a threshold on the relative error reduction of both terms of the cost function:

$$2 - \frac{e_E(t)}{e_E(t-1)} - \frac{e_{GO}(t)}{e_{GO}(t-1)} < \tau, \tag{9}$$

where $e_E(t) = ||d_E - AH_1||_F^2$ at iteration t, $e_{GO}(t) = ||W \odot (d_{GO} - AH_2)||_F^2$ at iteration t, and τ is a small threshold (in our experiments we used $\tau = 0.001$). This is not a classical weighted NMF approach. We have adapted the approach proposed in [18] to our problem, where an NMF with two weighted terms is used, coupled with one common matrix for both terms. Our problem formulation is equivalent, but we use weights only in the second term. A more detailed analysis of convergence is provided in the Supplementary Material.

Once we have learnt a proper dictionary that expands jointly d_E and d_{GO}, we can use it to fill the unknown parts of d_{GO}. If gene i does not have labels in GO, the i-th row and the i-th column of d_{GO} will be unknown. Assume that we have grouped all unlabelled genes at the lowest part of the matrix (if this is not the case, it can be easily done by a preprocessing, saving the ordering information in order to be able to undo it). For example, genes 1 to q have known labels and genes $q + 1$ to n do not have GO labels. Thus rows and columns $q + 1$ to n of d_{GO} will have unknown values. The reconstruction of those missing rows and columns will be done in three steps:

1. Learning of the first q columns of the unknown rows of d_{GO}: the learned atoms in A have n known elements, thus they can be used to expand the missing rows. In this way, the product AH_2 will contain information for the first q columns of the unknown rows, as shown in the scheme of Fig. 1 B).

2. Learning of the first q rows of the unknown columns: as matrix d_{GO} is symmetric, this means that in step 1 we have also reconstructed the first q rows of the missing columns, by transposition of the reconstruction obtained in that step. This is shown in Fig. 1 C).

3. Learning of the last part of the missing row and columns: the only part of matrix d_{GO} that remains unknown is the small submatrix at the lower right part from rows $q + 1$ to n and columns $q + 1$ to n. To learn this unknown part a new iteration must be done. Using A again, fixed, a new $\tilde{H}_2$ is learnt but now changing the weight matrix $\tilde{W}$ to have 0 only in the unknown lower right part, and using the partially reconstructed matrix d_{GO}. This is shown in Fig. 1 D).

Then the new $\tilde{H}_2$ is updated as

$$\tilde{H}_2 = \tilde{H}_2 \odot ((A^T(\tilde{W} \odot \tilde{d_{GO}})) \oslash (A^T(\tilde{W} \odot (A\tilde{H}_2)) + \epsilon)), \tag{10}$$

and this is iterated until convergence when

$$\left(1 - \frac{e_{GO}(n)}{e_{GO}(n-1)}\right) < \tau. \tag{11}$$

After convergence, the reconstruction of the missing part of the d_{GO} matrix is given by the corresponding part of $A\tilde{H}_2$. It should be noticed here that, although we have included this last step of reconstruction for completeness, $\tilde{H}_2$ has the semantic distances among non-annotated genes and therefore, it will not be used afterwards in the inference.

Taking the sub-matrix from rows $q+1$ to n and columns $q+1$ to n as a matrix M we have an approximation of this missing part. But d_{GO} needs to have symmetry, and due to the approximation error the obtained sub-matrix M may not be perfectly symmetric. To correct this, a symmetric matrix M_1 is obtained as $M_1 = (M + M^T)/2$, and used to finish the matrix completion. After these 3 steps we obtain the reconstruction of the complete d_{GO} matrix.

Regarding computational complexity, summing up all the multiplications/divisions required for one iteration of updates, the algorithm is $O(n^2 p)$. This operations count assumes a direct implementation, but a version parallelized with CuPy [22] is also provided, which significantly reduces computation time.

2.2 GO terms assignment

Let us define $G = \{\boldsymbol{g}_j\}$, $j = 1, \ldots, m$ as the set of genes with known GO labels in one specific GO sub-ontology. $L_j = \{\ell_{jk}\}$ is the set of all labels in gene $\boldsymbol{g}_j$ and $L = \cup L_j$ the set of all labels associated to G. Then, using a Bayesian estimation, for $\ell \in L$

$$\begin{aligned} p(\ell|\boldsymbol{g}_i) &\propto p(\ell) \cdot p(\boldsymbol{g}_i|\ell) && (12) \\ &= \frac{1}{C} \sum_{\boldsymbol{g}_j} I(\ell, \boldsymbol{g}_j) \cdot \sum_{\boldsymbol{g}_j / \ell \in L_j} S(\boldsymbol{g}_i, \boldsymbol{g}_j)^{\gamma}, && (13) \end{aligned}$$

where $I(\ell, \boldsymbol{g}_j)$ is an indicator function with value 1 if gene $\boldsymbol{g}_j$ was labelled with label ℓ and 0 otherwise; $S(\boldsymbol{g}_i, \boldsymbol{g}_j)$ is a similarity measure among genes $\boldsymbol{g}_i$ y $\boldsymbol{g}_j$, and C is a normalization constant. The exponent γ allows considering only the terms of the closest genes. This is particularly important when there is a large number of genes involved. In this work we used the similarity

$$S(\boldsymbol{g}_i, \boldsymbol{g}_j) = \frac{2}{1 + d(\boldsymbol{g}_i, \boldsymbol{g}_j)} - 1, \qquad (14)$$

which is always in the interval $[0, 1]$. Then L is sorted in descending order by $p(\ell|\boldsymbol{g}_i)$, and those labels with the highest probability are assigned to the gene $\boldsymbol{g}_i$ up to a maximum of accumulated probability μ. In this Bayesian inference, the GO tree structure is taken into account by combining the a-priori probabilities of the GO terms from the GO, including their ancestors propagated in the tree, together with their semantic distance, which takes into account the GO terms location in the tree and their hierarchical relationships.

3 Data and performance measures

We tested our proposal with three organisms: *Saccharomyces cerevisiae*, *Arabidopsis thaliana* and *Dictyostelium discoideum.* These organisms have been chosen because of their comprehensive set of functional annotations, that is, most genes are annotated with GO. This provides a precise way of measuring performance.

YEAST dataset: Gene expression data generated in *Saccharomyces cerevisiae.* In [23] several characteristics were collected in order to study cluster analysis of expression patterns. From an original dataset of 2,467 genes, only those with no missing values were considered, that is, a total of 605 genes with 79 microarray expression values. GO annotations with evidence codes EXP, IDA, IMP, IGI, IEP, TAS, and IC, with relation type "is-a" and "part-of", were considered following the recommendations of the CAFA challenge [13]. For the first experiment (Section 4.1) only GO annotations for Biological Process (BP) were considered, making a total of 583 genes. For the second experiment (Section 4.2), in order to make a fair comparison with a related method that uses NMF, GO terms annotated to fewer than 3 genes were excluded, leaving 422 genes for BP, 386 genes for Molecular Function (MF) and 442 genes for Cellular Component (CC).

ARA dataset: Gene expression measured in *Arabidopsis thaliana* leaves. The original work aimed to study the effects of cold temperatures on circadian-regulated genes in this plant [24]. Genes under light-dark cycles at two control temperatures (20ºC and 4ºC) and also involved in diurnal cycle and cold stress responses were selected for the study. From a total of 1,546 genes with 32 microarray expression values, only those annotated with evidence codes and relation types recommended by CAFA were considered. For the first experiment (Section 4.1) 656 genes remained with BP annotations. For the second experiment (Section 4.2) after the filter, 521 genes remained for BP, 315 genes for MF and 740 genes for CC.

DICTY dataset: Gene expression data of 1,219 genes of the amoeba *Dictyostelium discoideum* from dictyBase [25]. This dataset has been included in the experiments for fair comparison with a closely related state-of-the-art work on data fusion, [11], on exactly the same conditions. The gene expression measurements include different time-points of a 24-hour development cycle for 14 experiments, annotated with evidence codes and relation types recommended by CAFA. For the experiment in Section 4.1, 707 genes remained with BP annotations. For the experiment in Section 4.2, after the occurrence filter, 652 genes remained for BP, 366 genes for MF and 630 genes for CC.

ARA2 dataset: It involves genome-wide transcriptomic data from *A. thaliana.* RNAseq data were retrieved from the ATTED-II coexpression database [26]. The gene expression table used included 2,120 RNAseq transcriptomes from 22,761 genes and data were already normalized using the ComBat method [27]. To select sufficiently expressed samples, 90% of the highly expressed genes were selected. In summary, there were 20,842 genes with sufficient expression level. Among them, only 12,013 genes had BP annotations as well [28].

The prediction quality of each model was assessed with cross validation using a leave-one-out scheme, where we consider one gene at a time as unlabelled and predict the labels for that gene using the information from the rest. Genes with less than 10 GO terms were not considered in the performance measures. The classical classification measures of sensitivity (s^+), precision (p), and harmonic mean of sensitivity and precision (F_1) were used:

$$s^+ = \frac{TP}{TP+FN}, \tag{15}$$

$$p = \frac{TP}{TP+FP}, \tag{16}$$

$$F_1 = 2\frac{s^+ p}{s^+ + p}, \tag{17}$$

where TP is the number of true positives, FP the number of false positives and FN the number of false negatives, respectively. The s^+ measures how good is a classification method for not missing the true positives. The precision measures the relation between true positives and false positives. To take into account both sensitivity and false positives, F_1 is used as a global comparative measure among the prediction methods.

The optimal hyperparameters were found with a grid-search in the yeast dataset over $\lambda \in (0.001, 2.0)$, atom size $p \in (20, 100)$, maximum iterations number $t_{max} \in (100, 10,000)$, and $\gamma \in (0.50, 15)$. Alternative distances for expression data were tested, including: euclidean, correlation and cosine distance. Several semantic distances were tested such as cosine, Lin and Relevance with combination strategy BMA, minimum and average [15]. The optimum hyperparameters found were: cosine for expression distance and minimum Relevance for semantic distance, $p = 80$, $\lambda = 0.001$, $\gamma = 6$ and $t_{max} = 500$.

Friedman test and critical difference diagram [29] with post-hoc Nemenyi test were used to assess the statistical significance of differences in the performance achieved by each model.

4 Results and discussion

4.1 Comparative results with matrix factorization methods for data fusion

In this section, a leave-one-out cross validation was used for comparison in the same conditions with a related method in data fusion. For each testing gene, the original GO annotations were artificially removed. Then, the semantic distances related to these annotations were reconstructed by exp2GO. In this way, in each iteration we train our model in all the genes less one, and predict the functions, assumed unknown, for the remaining testing gene. In particular, *D. discoideum* was included in this study for fair comparison, on exactly the same conditions, with a closely related state-of-the-art proposal [11], which also uses gene expression data and it is based on penalized matrix tri-factorization [30]. This

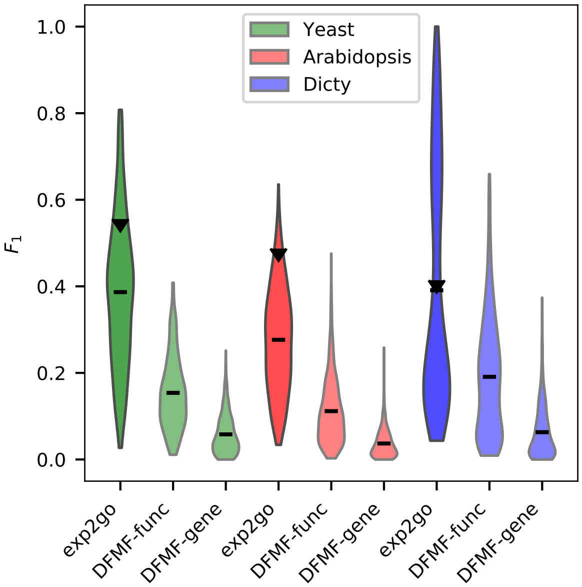

Figure 2: Violin plot with the comparative performance in several species for the BP sub-ontology. Average cross-validated F-score (F1) and standard deviation is reported for exp2GO; matrix tri-factorization function-centric (DFMF-func), matrix tri-factorization gene-centric (DFMF-gene). Black triangle: reference results using the original d_{GO}, instead of the reconstructed one.

proposal, named data fusion by matrix factorization (DFMF), performs data fusion by directly considering any data that can be represented in a matrix, including for example feature-based representations, ontologies, associations or networks. In [11], it is mentioned that classical data integration can be achieved by fusing input data (early integration) or by fusing predictions (late integration) but not by directly combining heterogeneous representation of objects of different types. We have considered here the two versions of DFMF as proposed by the authors: gene-centric (DFMF-gene), where the threshold for function assignment is based on the average of the gene function labels; and function-centric (DFMF-func), where the threshold is set according to the mean number of genes having the function.

Figure 2 shows a violin plot with the comparative results in several species for the BP sub-ontology. The values reported are average F_1 for the methods exp2GO, DFMF-func and DFMF-gene. The black triangle on top of exp2GO indicates the F_1 that could be achieved if the original d_{GO} were used instead of the reconstructed matrix. This value has been included as reference, in order to show the maximum possible performance that could be obtained only by the Bayesian inference if the d_{GO} reconstruction were perfect.

It can be seen in Figure 2 that for YEAST (green violins), exp2GO obtained the best $F_1 = 0.3867$, largely outperforming the DFMF methods (0.1540 and 0.0560). In the ARA dataset (red violins), exp2GO obtained again the best $F_1 = 0.2765$ while the DFMF methods provided very low performance (0.1090 and 0.0350). Finally, for the DICTY dataset (blue violins), exp2GO, again, outperformed the comparative methods, achieving $F_1 = 0.3908$, more than twice the DFMF best value (0.1910). In this case, it should be highlighted that exp2GO achieved the same F_1 than if the original d_{GO} were used. In summary, the performance of exp2GO in comparison to DFMF is 2.5 times higher in YEAST (0.3867 vs. 0.1540), and twice better in ARA (0.2765 vs. 0.1120) and dicty (0.3908 vs. 0.1910). This is a remarkable result and a strong indication that the expression data used in the d_{GO} reconstruction by exp2GO has added important information for the correct prediction of GO terms.

In summary, for all datasets evaluated here it can be seen that the best performance was clearly obtained by exp2GO. Regarding the computational resources needed and the stability of convergence, it should be noted that in spite of exp2GO being an algorithmically complex method, its convergence is fast for this application. For example, for the YEAST dataset, running on 4 Intel Xeon processors and 1 Tesla P100 GPU, exp2GO parallelized with CuPy took on average 5.09 s to converge (using the maximum number or iterations t_{max}). Furthermore, variance of the method in label assignment under random initialization was very low. For example, for one gene in 100 independent runs, it has a standard deviation of 0.016 for F_1.

4.2 Comparative results in CAFA-like challenge setup

To quantitatively study the performance of exp2GO in a more challenging scenario, we evaluated it in the setup like the proposed by the CAFA challenge [1, 13]. According to the CAFA rules, the function prediction challenge involves different times used as training and testing datasets. For CAFA3, the first time is when the challenge is released (2016) and a second reference time is the deadline for the submissions of the predictions (2017). In the challenge, two classes of genes/proteins are considered. On the one hand, the no-knowledge (NK) proteins are those that do not have experimental annotations in any of the GO sub-ontologies, initially, but have accumulated at least one GO term with an experimental evidence code during the prediction period. On the other hand, the limited-knowledge (LK) proteins are those which already had one or more GO terms experimentally annotated in at least one of the three GO sub-ontologies at the initial time.

Two releases of GO annotation (GOA) files of the three species (*S.cerevisiae*, *A. thaliana* and *D. discoideum*) were downloaded in different dates: the first one on 2016-06-01, and the other one on 2017-02-01. Since the GO structure changes with time (new terms are added), only GO terms shared in both times were considered. This information was downloaded for the 3 sub-ontologies: BP, MF and CC. The GOA file of each species gathers the available associations between genes and GO terms. These associations were expanded according

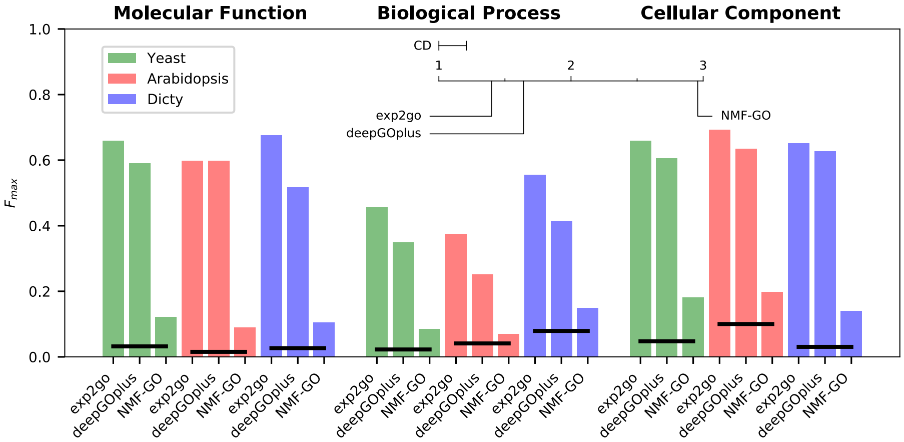

Figure 3: The F_{max} for the comparative methods in the CAFA challenge setup for several species: YEAST (green bars), ARA (red bars) and DICTY (blue bars) in the MF, BP and CC sub-ontologies. The baseline method BLAST is indicated with a black line. On top: statistical significance of results is presented with the critical difference (CD) diagram.

to the GO hierarchy and the True Path Rule [15], which indicates that if a term is annotated to a gene, then all ancestors of this term are also inherently annotated to the same gene. Results are reported according to the CAFA rules, with the maximum F_1-measure (F_{max}), which considers predictions across the full spectrum from high to low sensitivity.

In this setup, we compared exp2GO with one baseline method and two state-of-the-art methods: a method based on the sequence similarity (measured by aligment with BLAST [31]), which has been used as baseline in the CAFA challenge [13], NMF-GO [12] and deepGOplus [14]. NMF-GO is a very recent method that completes the gene-function association matrix using NMF, imposing constraints with the semantic distance between functions terms calculated with the Lin measure [15]. This method has already been compared to another 6 methods, clearly outperforming all. We followed the indications of the NMF-GO authors on the experimental setup: training on the historical GOA files (released in 2016) and validating predictions on the more recent GOA file (released in 2017). DeepGOplus was also proposed very recently and it is based on deep learning. This method has outperfomed all the winners of the CAFA3 challenge.

Figure 3 shows the comparative results in the CAFA challenge setup for exp2GO, BLAST, NMF-GO and deepGOplus. The baseline method BLAST is indicated with a black line. In the MF sub-ontology, exp2GO is clearly superior to NMF-GO, deepGOplus and BLAST in YEAST (green bars) and DICTY (blue bars), with almost the same result that deepGOplus in ARA (red bars).

The exp2GO scores here are $F_{max} = 0.6587$ for YEAST, $F_{max} = 0.5975$ for ARA and $F_{max} = 0.6760$ for DICTY. In the BP sub-ontology, the hardest one, exp2GO is superior to all other methods and with important differences among them. For example in DICTY (blue bars) exp2GO has $F_{max} = 0.5547$ while deepGOplus reached $F_{max} = 0.4123$ and NMF-GO reached only $F_{max} = 0.1487$. In YEAST and ARA, again exp2GO is superior, up to 5 times better than NMF-GO. In the CC sub-ontology, the exp2GO scores are the highest for all species. The same trend than with the previous results can be observed and analogous conclusions can be achieved: exp2GO is far better than the baseline BLAST, with F_{max} higher than the other methods.

In order to provide a statistical analysis of results, a Friedman test was done, showing that differences in the F_{max} results are statistically significant ($p <$ 6.6E-20). Critical difference diagram (Figure 3, top) shows that exp2GO is the best method for gene function prediction. It should be noticed that the gain in performance of exp2GO in comparison to other methods is significant, as this statistical analysis indicates. The performance for exp2GO in comparison to NMF-GO in BP (the hardest sub-ontology) is 5 times more in YEAST (0.456 vs. 0.085), 6 times higher in ARA (0.374 vs. 0.069) and 5 times better in DICTY (0.555 vs. 0.149). These are very large gains in performance for exp2GO in comparison to another NMF-based method.

Notably, these gains are achieved by exp2GO without a downside, since the other methods are more complex. From a computational point of view, in the comparison between exp2GO and NMF-GO it should be noticed that the last one needs extremely large matrices, of size $m \times m$, while exp2GO only needs two matrices of size $n \times n$. This is an important difference because NMF-GO operates with matrices of thousands of terms, whereas exp2GO only requires working with matrices that involve hundreds of genes for the NMF fusion. In the case of deepGOplus, it needs each sequence in order to build an embedding and feed it into a convolutional neural network, with a very high computational cost. Therefore, in any case, exp2GO requires simpler and smaller inputs, and thus, it has lower computational cost than the other methods.

All the experiments in this section used distance matrices. However, NMF methods usually tend to underestimate near zero values (i.e. the most similar genes). Since distance matrices are normalized, $d \in [0, 1]$, we replicated the experiments by defining similarity matrices as $s = 1 - d$. The results were slightly worse, with drops in the F_{max} for all species and sub-ontologies. This might be because, in this prediction task, it is more important to distinguish between the non-similar cases than to better detect the similar cases. That is, the efficient detection of non similar genes annotations has more impact on the final results, since it avoids the assignment of many incorrect labels.

4.3 Genome-wide data

In order to get a better sense of the type of results obtained with exp2GO in the context of a complete genome, we performed an experiment with 12,013 genes of the ARA2 dataset, which included 2,120 transcriptomic experiments (features) for each gene.

To verify whether exp2GO can give preference to GO terms related to a specific condition, we first tested the performance of the algorithm on 10 genes annotated to the GO category "defense response" (GO:0006952). These genes were randomly selected, but preserving a similar distribution of the number of annotations with respect to all genes in this category. The results (Supplementary Table 1) indicate that exp2GO gave preference to defense-response GO terms for those test genes, since it annotated all 10 test genes with "response to stress" (GO:0006950), which is a direct ancestor term of "defense response", and also to the more generic ancestor "response to stimulus" (GO:0050896).

Then, we selected extensively studied genes with important roles in *Arabidopsis* development and/or responses to external stimulus and performed a manual inspection of the categories predicted by exp2GO with respect to the set of true annotations. For these genes, their original GO annotations were artificially removed and the semantic distances related to these annotations (the d_{GO} missing parts) were then reconstructed by exp2GO. After that, GO labels were assigned to the test genes according to the statistical procedure described in Section 2.2. The details of the GO labels inferred for those genes can be found in Supplementary Table 2, where the first column shows the GO terms inferred by exp2GO alone, and the third column shows the GO terms inferred by exp2GO that coincide with the original GO terms for each gene.

The first gene studied was *Phytochrome B* (*PhyB*, AT2G18790), a well-characterized red light sensor. The categories correctly identified by our method capture some of the most important aspects of *PhyB* known function, for instance, "photomorphogenesis" (GO:0009640), "circadian rhythm" (GO:0007623) and "regulation of transcription, DNA templated" (GO:0006355) [32]. Interestingly, also "protein autophosphorylation" (GO:0046777) was correctly predicted, which could intuitively be considered a category with low probabilities of being inferred from expression data. Additionally, exp2GO predicted for *PhyB* several categories which, despite not being true annotations, are closely related to them (Supplementary Figure 2). exp2GO found terms such as "positive regulation of transcription, DNA-templated" (GO:0045893) and "negative regulation of transcription, DNA-templated" (GO:0045892) that are direct children of the true annotation "regulation of transcription, DNA-templated" (GO:0006355), also found by the algorithm. Moreover, the term "transcription, DNA templated" (GO:0006351), a direct ancestor of the same category, was also found. Analogously, "protein phosphorylation" (GO:0006468) was found by exp2GO together with its true positive children category "protein autophosphorylation" (GO:0046777), previously mentioned. Another interesting result was obtained in relation to the true annotation "chromatin organization" (GO:0006325). exp2GO did not predict this term, but predicted the children category "chromatin silencing" (GO:0006342) and the related category "gene silencing" (GO:0016458). These more informative categories are valid since there are published results supporting the involvement of *PhyB* in these processes [33].

The annotation of another well-studied gene was inspected. *Pathogenesis-*

related 1 (*PR1*, AT2G14610) is a gene involved in the salicylic acid-mediated response to pathogens (Supplementary Table 2 and Supplementary Figure 3). As with *PhyB*, among the categories identified by the method there were many related to true annotations as well as ancestor GO terms (Supplementary Figure 3). For example, exp2GO predicted the term "response to bacterium" (GO:0009617), ancestor of the true annotation "defense response to bacterium" (GO:0042742); the term "salicylic acid mediated signaling pathway" (GO:0009863), ancestor of the true annotation "systemic acquired resistance, salicylic acid mediated signaling pathway" (GO:0009862); and the term "response to bacterium" (GO:0009617), ancestor of the true annotation "defense response to bacterium" (GO:0042742).

Lastly, the results for the gene *AGAMOUS* (*AG*, AT4G18960) were inspected. *AG* is a transcription factor essential for flower development as it is responsible for floral organ identity determination [34]. Some of the main aspects of *AG* function were recovered by the GO terms inferred by exp2GO, for instance, "regulation of flower development" (GO:0009909), "stamen development" (GO:0048443) and "plant ovule development" (GO:0048481). As in previous cases, there were many categories closely related to true annotations (Supplementary Figure 4). For example, exp2GO predicted "meristem maintenance" (GO:0010073), children of the true annotation "meristem development" (GO:0048507); and "flower development" (GO:0009908), indirect ancestor of "stamen development" (GO:0048443) and "petal development" (GO:0048441).

Altogether, these observations indicate that exp2GO was capable of inferring key characteristics of the function of well-studied genes. The careful inspection of these terms shows that many were highly related to true annotations, being either more (children categories) or less (ancestor categories) informative. Furthermore, ancestor categories could arguably be considered true positives, since they describe the function at a more general level; thus there is clearly some level of underestimation of the performance of the algorithm by using standard measures. This is probably further impacted by any biologically meaningful category predicted but not annotated or known yet.

Another interesting observation that came from the inspection of the results for these genes is that many true annotations had in fact a very small number of genes annotated to them in our dataset. For example, for *PhyB* "entrainment of circadian clock" (GO:0009649) was annotated to five genes, "transpiration" (GO:0010148) was annotated to four genes, "circadian regulation of calcium ion oscillation" (GO:0010617) was annotated to three genes and "response to low influence red light stimulus" (GO:0010202) was annotated only to *PhyB*. Similarly, "response to vitamin B1" (GO:0010266), the only category which was not identified for *PR1* by exp2GO, was annotated to two genes only. In fact, these terms with only one or very few annotated genes are very difficult to be predicted for any method, in particular one relying on known annotations of other genes of the same species.

5 Conclusions

In this work we have proposed exp2GO, a novel method for inferring GO annotations for genes with unknown function. Using the expression distance and the semantic distances among known genes, the semantic distance between unknown genes was reconstructed by using non-negative matrix factorization. With the reconstructed semantic distance, a Bayesian algorithm was used to predict the GO annotations.

The proposal was compared against state of-the-art methods on public datasets. It can be stated that exp2GO has shown the best results in all experiments, not missing true GO terms and not assigning, either, a large number of false positives to unannotated genes. Moreover, in the very challenging CAFA setup, exp2GO has clearly outperformed other methods. Finally, we have validated and tested the proposal with a case involving genome-wide data, where exp2GO was capable of effectively predicting relevant biological functions.

Acknowledgments

This work was supported by ANPCyT (PICT 2018 #3384) and UNL CAID 2020 115. In the present work we used computational resources form the Pirayu high-performance computer cluster which was acquired with funding from the Agencia Santafesina de Ciencia, Tecnologia e Innovacion, government of the Provincia de Santa Fe, Project AC-00010-18, R No117/14 (ASACTEI).

References

[1] P. Radivojac, W. Clark, T. Oron, A. Schnoes, and T. Wittkop, "A large-scale evaluation of computational protein function prediction," *Nature Methods*, vol. 10, no. 3, pp. 221–227, 2013.

[2] J. Blake, M. Dolan, H. Drabkin, D. Hill, N. Li, D. Sitnikov, S. Bridges, S. Burgess, and T. Buza, "Gene ontology annotations and resources," *Nucleic Acids Research*, vol. 41, no. 1, pp. 530–535, 2013.

[3] P. Resnik, "Semantic similarity in a taxonomy: An information based measure and its application to problems of ambiguity in natural language," *Journal of Artificial Intelligence Research*, vol. 11, no. 1, pp. 95–130, 1999.

[4] A. Perez, C. Perez-Iratxeta, P. Bork, G. Thode, and M. Andrade, "Gene annotation from scientific literature using mappings between keyword systems," *Bioinformatics*, vol. 20, no. 13, pp. 2084–2091, 2004.

[5] Z. Barutcuoglu, R. Schapire, and O. Troyanskaya, "Hierarchical multi-label prediction of gene function," *Bioinformatics*, vol. 22, no. 7, pp. 830–836, 2006.

[6] Y. Tao, L. Sam, J. Li, C. Friedman, and Y. Lussier, "Information theory applied to the sparse gene ontology annotation network to predict novel gene function," *Bioinformatics*, vol. 23, no. 13, pp. i529–i538, 2007.

[7] O. King, R. Foulger, S. Dwight, J. White, and F. Roth, "Predicting gene function from patterns of annotation," *Genome Research*, vol. 13, no. 5, pp. 896–904, 2013.

[8] G. Yu, H. Zhu, C. Domeniconi, and J. Liu, "Predicting protein function via downward random walks on a gene ontology," *BMC Bioinformatics*, vol. 16, pp. 271–281, 2015.

[9] G. Yu, G. Fu, J. Wang, and H. Zhu, "Predicting protein function via semantic integration of multiple networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 2, pp. 220–232, 2016.

[10] P. Pinoli, D. Chicco, and M. Masseroli, "Computational algorithms to predict gene ontology annotations," *BMC Bioinformatics*, vol. 16, no. 6, p. S4, 2015.

[11] M. Zitnik and B. Zupan, "Data fusion by matrix factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 41–53, 2015.

[12] G. Yu, K. Wang, G. Fu, M. Guo, and J. Wang, "NMFGO: Gene Function Prediction via Nonnegative Matrix Factorization with Gene Ontology," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 1, pp. 238–249, 2020.

[13] N. Zhou, Y. Jiang, T. R. Bergquist, A. J. Lee, B. Z. Kacsoh, A. W. Crocker, K. A. Lewis, G. Georghiou, H. N. Nguyen, M. N. Hamid, L. Davis, T. Dogan, V. Atalay, A. S. Rifaioglu, A. Dalkıran, R. C. Atalay, C. Zhang, R. L. Hurto, P. L. Freddolino, Y. Zhang, P. Bhat, F. Supek, J. M. Fernández, and B. Gemovic, "The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens," *Genome Biology*, vol. 20, no. 1, pp. 244–250, 2019.

[14] M. Kulmanov and R. Hoehndorf, "DeepGOPlus: improved protein function prediction from sequence," *Bioinformatics*, vol. 36, no. 2, pp. 422–429, 07 2019.

[15] C. Pesquita, D. Faria, A. Falcao, P. Lord, and F. Couto, "Semantic similarity in biomedical ontologies," *PLoS Computational Biology*, vol. 5, no. 7, p. e1000443, 2009.

[16] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 1, pp. 788–791, 1999.

[17] A. Cichocki, R. Zdunek, A. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations. Applications to exploratory multi-way data analysis and blind source separation.* John Wiley & Sons, 2009.

[18] H. Lee, J. Yoo, and S. Choi, "Semi-supervised nonnegative matrix factorization," *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 4–7, 2010.

[19] Y. Kim and S. Choi, "Weighted nonnegative matrix factorization," in *IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2009, pp. 1541–1544.

[20] D. Kuang, C. Ding, and H. Park, *Symmetric Nonnegative Matrix Factorization for Graph Clustering*, 2012, ch. 1, pp. 106–117.

[21] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 126–135.

[22] R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis, "CuPy: A NumPy-compatible library for NVIDIA GPU calculations," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017. [Online]. Available: https://cupy.dev

[23] M. Eisen, P. Spellman, P. Brown, , and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Nat. Acad. Sciences*, vol. 95, no. 5, pp. 14 863–14 868, 1998.

[24] C. Espinoza, T. Degenkolbe, C. Caldana, E. Zuther, A. Leisse, and L. Willmitzer, "Interaction with diurnal and circadian regulation results in dynamic metabolic and transcriptional changes during cold acclimation in arabidopsis," *PLOS ONE*, vol. 5, no. 11, p. e14101, 2010.

[25] L. Kreppel, "dictyBase: a new dictyostelium discoideum genome database," *Nucleic Acids Research*, vol. 32, no. 90001, pp. 332D–333, Jan. 2004. [Online]. Available: http://dictybase.org

[26] T. Obayashi, Y. Aoki, S. Tadaka, Y. Kagaya, and K. Kinoshita, "ATTED-II in 2018: A Plant Coexpression Database Based on Investigation of the Statistical Property of the Mutual Rank Index," *Plant and Cell Physiology*, no. July, pp. 1–7, 2017. [Online]. Available: http://atted.jp

[27] W. E. Johnson, C. Li, and A. Rabinovic, "Adjusting batch effects in microarray expression data using empirical Bayes methods," *Biostatistics*, vol. 8, no. 1, pp. 118–127, 2007.

[28] M. Carlson, S. Falcon, H. Pages, and N. Li, "org.At.tair.db: Genome wide annotation for Arabidopsis," 2017. [Online]. Available: https://doi.org/doi:10.18129/B9.bioc.org.At.tair.db

[29] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, p. 1–30, dec 2006.

[30] F. Wang, T. Li, and C. Zhang, "Semisupervised clustering via matrix factorization," *Proc. of SIAM International Conference on Data Mining*, vol. 1, no. 1, pp. 1–12, 2008.

[31] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, Oct. 1990. [Online]. Available: https://blast.ncbi.nlm.nih.gov

[32] K. A. Franklin and P. H. Quail, "Phytochrome functions in arabidopsis development," *Journal of Experimental Botany*, vol. 61, no. 1, pp. 11–24, 2009.

[33] F. Tessadori, "Phytochrome b and histone deacetylase 6 control light-induced chromatin compaction in arabidopsis thaliana," *PLoS Genetics*, vol. 5, no. 9, p. e1000638, 2009.

[34] D. Weigel and E. M. Meyerowitz, "The abcs of floral homeotic genes," *Cell*, vol. 78, no. 2, pp. 203–209, 1994.

Supplementary Material

exp2GO: improving prediction of functions in the Gene Ontology with expression data

Leandro Di Persia, Tiago Lopez, Agustin Arce, Diego H. Milone and Georgina Stegmayer

1 Convergence analysis

In this kind of updates, the cost function is not convex in all variables and the non-negativity restrictions make difficult to prove the convergence. The usual way to show convergence is based in the method of auxiliary function, which consist of proposing a convex function that works as an upper bound for the cost function, tight at the present value of the variable, and minimize this one in each iteration instead of the original cost function. By using the auxiliary function, one can show that the cost function is non-increasing with the given updates. To formalize this, the following definition is needed [1].

Definition 1. *Given the current estimates of the parameters Θ^t, let $\mathcal{O}(\Theta|\Theta^t)$ be defined in $\mathbb{R}_+^{m\times n} \times \mathbb{R}_+^{m\times n} \to \mathbb{R}_+$. This mapping is called an Auxiliary Function for the cost function $J(\Theta)$ if and only if:*

$$\forall \Theta \in \mathbb{R}_+^{m\times n}, \mathcal{O}(\Theta|\Theta) = J(\Theta) \quad \text{and} \quad \forall (\Theta, \Theta^t) \in \mathbb{R}_+^{m\times n} \times \mathbb{R}_+^{m\times n}, \mathcal{O}(\Theta|\Theta^t) \geq J(\Theta).$$

Lemma 1. *If $\mathcal{O}(\Theta|\Theta^t)$ is an Auxiliary Function for $J(\Theta)$, then $J(\Theta)$ is non-increasing under the update*

$$\Theta^{t+1} = \arg\min_{\Theta \geq 0} \mathcal{O}(\Theta|\Theta^t).$$

The demonstration is straightforward, as $J(\Theta^{t+1}) \leq \mathcal{O}(\Theta^{t+1}|\Theta^t) \leq \mathcal{O}(\Theta^t|\Theta^t) \leq J(\Theta^t)$.

1.1 Convergence of updates for A

In this case H_1 and H_2 are kept fixed, so the cost in Eq. 5 of the manuscript, $J(A, H_1, H_2)$, is reduced to

$$\begin{aligned} J(A) &= ||d_E - AH_1||_F^2 + \lambda ||W \odot (d_{GO} - AH_2)||_F^2 \\ &= \sum_i \sum_j \left([d_E]_{ij} - \sum_k [A]_{ik}[H_1]_{kj} \right)^2 + \lambda \sum_i \sum_j [W]_{ij} \left([d_{GO}]_{ij} - \sum_k [A]_{ik}[H_2]_{kj} \right)^2, \end{aligned}$$

where $[W]_{ij}^2 = [W]_{ij}$ since $[W]_{ij} \in \{0, 1\}$ by definition. For this cost, an auxiliary function is proposed as

$$\mathcal{O}(A|A^t) = \sum_i \sum_j \sum_k \delta_{ijk} \left([d_E]_{ij} - \frac{[A]_{ik}[H_1]_{kj}}{\delta_{ijk}} \right)^2 + \sum_i \sum_j [W]_{ij} \sum_k \nu_{ijk} \left([d_{GO}]_{ij} - \frac{[A]_{ik}[H_2]_{kj}}{\nu_{ijk}} \right)^2,$$

where $\delta_{ijk} = \frac{[A^t]_{ik}[H_1]_{kj}}{\sum_\ell [A^t]_{i\ell}[H_1]_{\ell j}}$ satisfies $\sum_k \delta_{ijk} = 1$, $\nu_{ijk} = \frac{[A^t]_{ik}[H_2]_{kj}}{\sum_\ell [A^t]_{i\ell}[H_2]_{\ell j}}$ satisfies $\sum_k \nu_{ijk} = 1$, and they do not depend on A. By replacement, it is straightforward to show that $\mathcal{O}(A|A) = J(A)$. Besides, the Jensen inequality can be used to show that it is an upper bound for the cost

$$\begin{aligned} \mathcal{O}(A|A^t) &\geq \sum_i \sum_j \left([d_E]_{ij} - \sum_k \delta_{ijk} \frac{[A]_{ik}[H_1]_{kj}}{\delta_{ijk}} \right)^2 + \lambda \sum_i \sum_j [W]_{ij} \left([d_{GO}]_{ij} - \sum_k \nu_{ijk} \frac{[A]_{ik}[H_2]_{kj}}{\nu_{ijk}} \right)^2 \\ &= \sum_i \sum_j \left([d_E]_{ij} - \sum_k [A]_{ik}[H_1]_{kj} \right)^2 + \lambda \sum_i \sum_j [W]_{ij} \left([d_{GO}]_{ij} - \sum_k [A]_{ik}[H_2]_{kj} \right)^2 = J(A), \end{aligned}$$

and then $\mathcal{O}(A|A^t)$ is an auxiliary function for $J(A)$.

At this point, as $\mathcal{O}(A|A^t)$ is a convex function, to apply Lemma 1 we need to get the derivative, equate it to zero and solve for the update, resulting in the update in Eq. 6 of the manuscript.

1.2 Convergence of updates for H_1

When updating for H_1, with A and H_2 fixed, the cost function is reduced to

$$J(H_1) = ||d_E - AH_1||_F^2 = \sum_i \sum_j \left([d_E]_{ij} - \sum_k [A]_{ik}[H_1]_{kj} \right)^2 .$$

It must be noted that this problem is a standard one and it was already proven that it is non-increasing under the multiplicative update used in this work [1]. Nevertheless, it will be proven here for completeness, using a slightly different auxiliary function, similar to the one used in [2]

$$\mathcal{O}(H_1|H_1^t) = \sum_i \sum_j \sum_k \lambda_{ijk} \left([d_E]_{ij} - \frac{[A]_{ik}[H_1]_{kj}}{\lambda_{ijk}} \right)^2 ,$$

where $\lambda_{ijk} = \frac{[A]_{ik}[H_1^t]_{kj}}{\sum_\ell [A]_{i\ell}[H_1^t]_{\ell j}}$ satisfies $\sum_k \lambda_{ijk} = 1$ and does not depends on H_1. It is straightforward to show that $\mathcal{O}(H_1|H_1) = J(H_1)$. On the other side, using the Jensen inequality we have

$$\mathcal{O}(H_1|H_1^t) \geq \sum_i \sum_j \left([d_E]_{ij} - \sum_k \lambda_{ijk} \frac{[A]_{ik}[H_1]_{kj}}{\lambda_{ijk}} \right)^2 = \sum_i \sum_j \left([d_E]_{ij} - \sum_k [A]_{ik}[H_1]_{kj} \right)^2 = J(H_1).$$

In this way, $\mathcal{O}(H_1|H_1^t)$ is an auxiliary function for $J(H_1)$.

As $\mathcal{O}(H_1|H_1^t)$ is convex and quadratic in H_1, only the first order condition is needed to find its maximum. Taking partial derivative with respect to $[H_1]_{p,q}$, equating it to zero and solving for $[H_1]_{p,q}$ the update in Eq. 7 of the manuscript is obtained.

1.3 Convergence of updates for H_2

In a similar way to the previous case, now A and H_1 are fixed and the cost function is reduced to

$$J(H_2) = ||W \odot (d_E - AH_2)||_F^2 = \sum_i \sum_j [W]_{ij} \left([d_{GO}]_{ij} - \sum_k [A]_{ik}[H_2]_{kj} \right)^2 .$$

It should be noted that as W acts as a row and column selection matrix, it makes zero all elements of some rows and columns. This optimization would be equivalent to first reduce all matrices to the selected row and columns, and the problem would be then reduced to a standard unweighted problem equivalent to the one presented in the update of H_1. Additionally, in [3] the problem for more general W matrices was analyzed. It was proven that $J(H_2)$ is non-increasing under the update equation of our work. Nevertheless, a demonstration following the same ideas as in the previous section is presented here for completeness. The proposed auxiliary function for this problem is

$$\mathcal{O}(H_2|H_2^t) = \sum_i \sum_j [W]_{ij} \sum_k \lambda_{ijk} \left([d_{GO}]_{ij} - \frac{[A]_{ik}[H_2]_{kj}}{\lambda_{ijk}} \right)^2$$

where $\lambda_{ijk} = \frac{[A]_{ik}[H_2^t]_{kj}}{\sum_\ell [A]_{i\ell}[H_2^t]_{\ell j}}$ satisfies $\sum_k \lambda_{ijk} = 1$ and does not depends on H_2. By direct replacement it is shown that $\mathcal{O}(H_2|H_2) = J(H_2)$. To complete the verification that it is an auxiliary function, Jensen inequality is used again

$$\mathcal{O}(H_2|H_2^t) \geq \sum_i \sum_j [W]_{ij} \left([d_{GO}]_{ij} - \sum_k \lambda_{ijk} \frac{[A]_{ik}[H_2]_{kj}}{\lambda_{ijk}} \right)^2 = \sum_i \sum_j [W]_{ij} \left([d_{GO}]_{ij} - \sum_k [A]_{ik}[H_2]_{kj} \right)^2 = J(H_2)$$

and in this way, $\mathcal{O}(H_2|H_2^t)$ is an Auxiliary Function for $J(H_2)$.

As previously, the auxiliary function is a convex quadratic, thus only the first order condition is needed to minimize it, resulting in the update of Eq. 8 in the manuscript.

1.4 Convergence of updates for $\tilde{H}_2$

For the second stage of the algorithm, the basis matrix A is kept constant and the update is only performed for matrix $\tilde{H}_2$. For this case also the weight matrix is changed to $\tilde{W}$. This problem is exactly the same as the update for H_2, and for this problem it was already shown that the cost is non-increasing after the given update.

References

[1] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13. MIT Press, 2001.

[2] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the beta-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[3] N.-D. Ho, "Nonnegative matrix factorization algorithms and applications," Ph.D. dissertation, Citeseer, 2008.

2 Supplementary figures

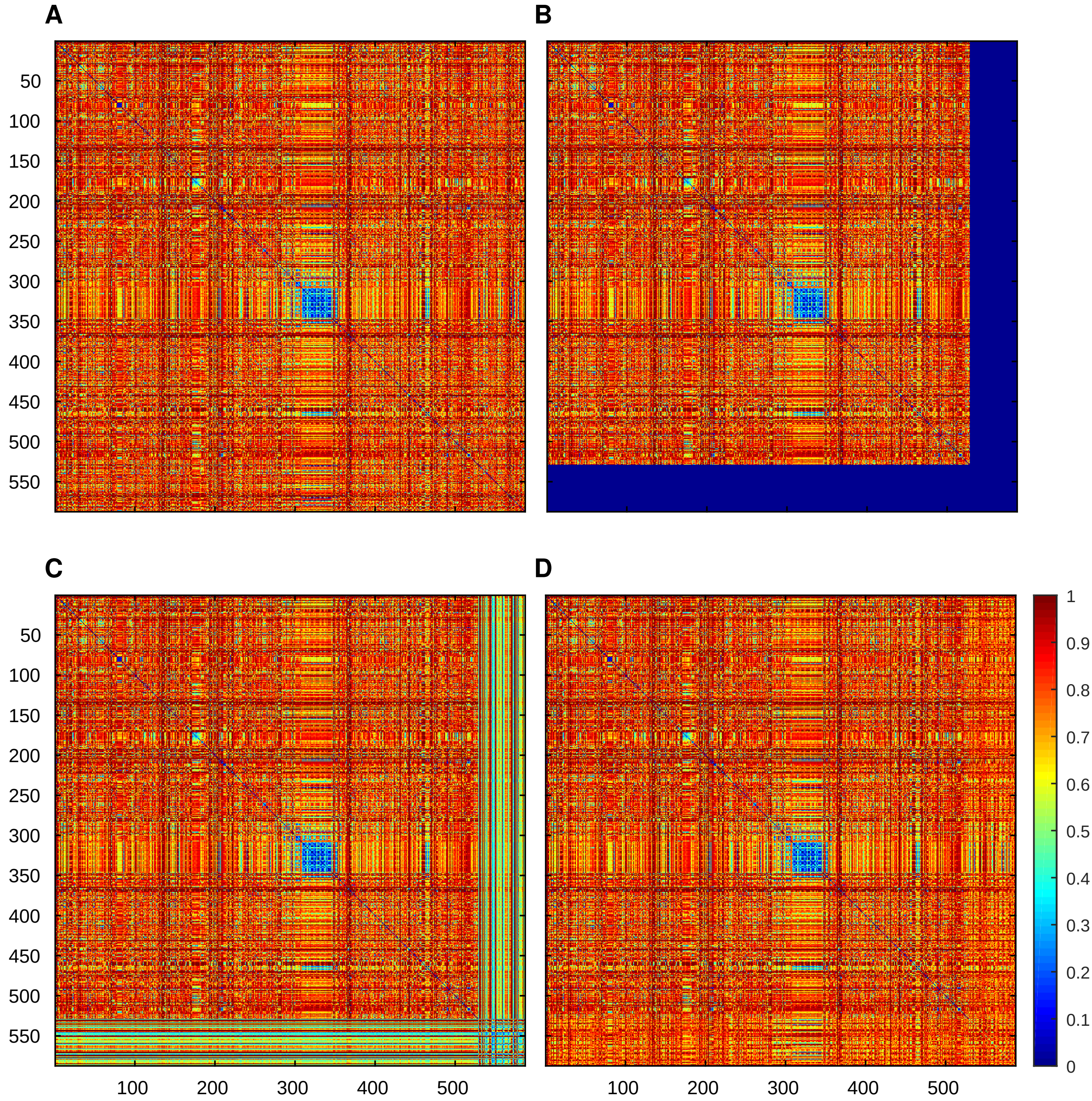

Fig. 1. Alternatives to matrix decomposition with NMF using the YEAST dataset and GO annotations for Biological Process. A) Semantic distances among all genes. B) Semantic distance matrix with missing rows and columns for (simulated) non-annotated genes. C) Semantic distance matrix reconstructed by symmetric tri-factorization, with $J = ||d_E - US_1U^T||_F^2 + \lambda||W \odot (d_{GO} - US_2U^T)||_F^2$. For this case the reconstruction error was $||d_{GO} - \hat{d}_{GO}||_F^2 = 82.2994$. D) Semantic distance matrix reconstructed by the proposed method, with $J(A, H_1, H_2) = ||d_E - AH_1||_F^2 + \lambda||W \odot (d_{GO} - AH_2)||_F^2$. For exp2GO the reconstruction error in this example was $||d_{GO} - \hat{d}_{GO}||_F^2 = 50.3659$.

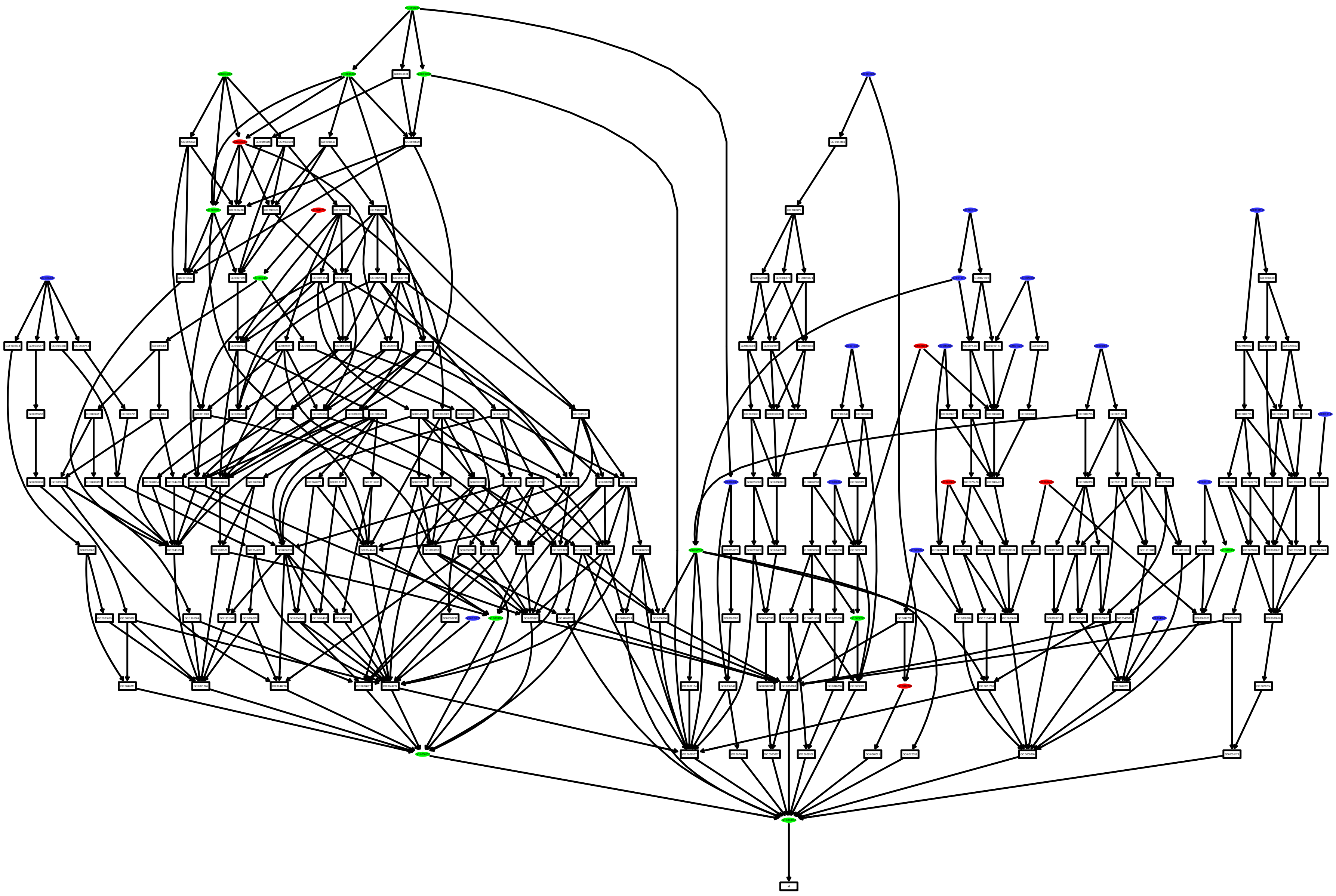

Fig. 2. Gene Ontology (GO) terms and corresponding GO directed acyclical graphs (DAG) for AT2G18790-*PhyB* induced from the full GO-BP DAG using the set of true annotations plus their direct children that were also predicted by exp2GO. Node color code is as follows: in red, true annotations predicted by exp2GO; in blue, true annotations not found by exp2GO, and in green are exp2GO annotations which are children or ancestors of true annotations.

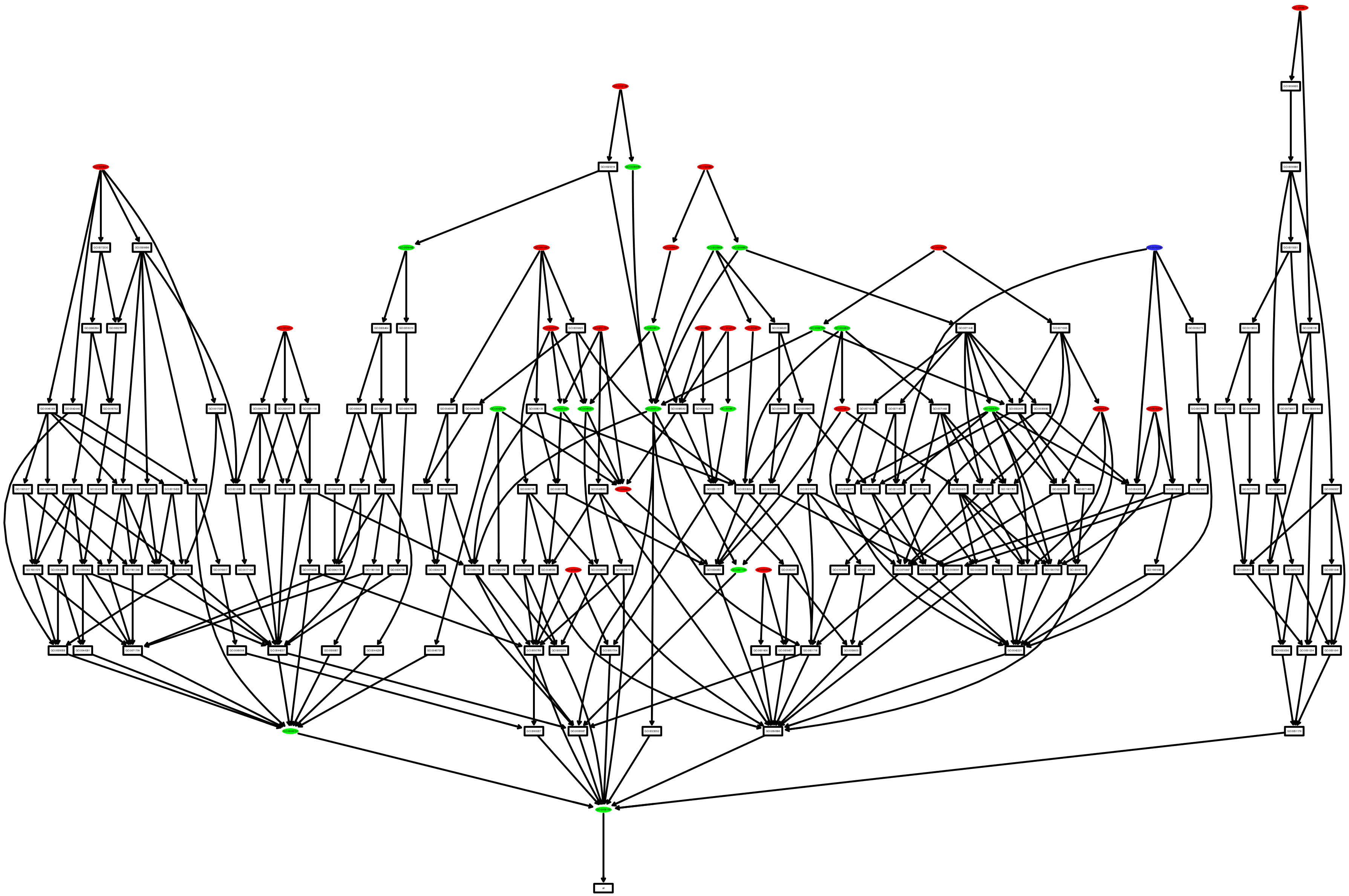

Fig. 3. Gene Ontology (GO) terms and corresponding GO directed acyclical graphs (DAG) for AT2G14610-*PR1* induced from the full GO-BP DAG using the set of true annotations plus their direct children that were also predicted by exp2GO. Node color code is as follows: in red, true annotations predicted by exp2GO; in blue, true annotations not found by exp2GO and in green are exp2GO annotations which are children or ancestors of true annotations.

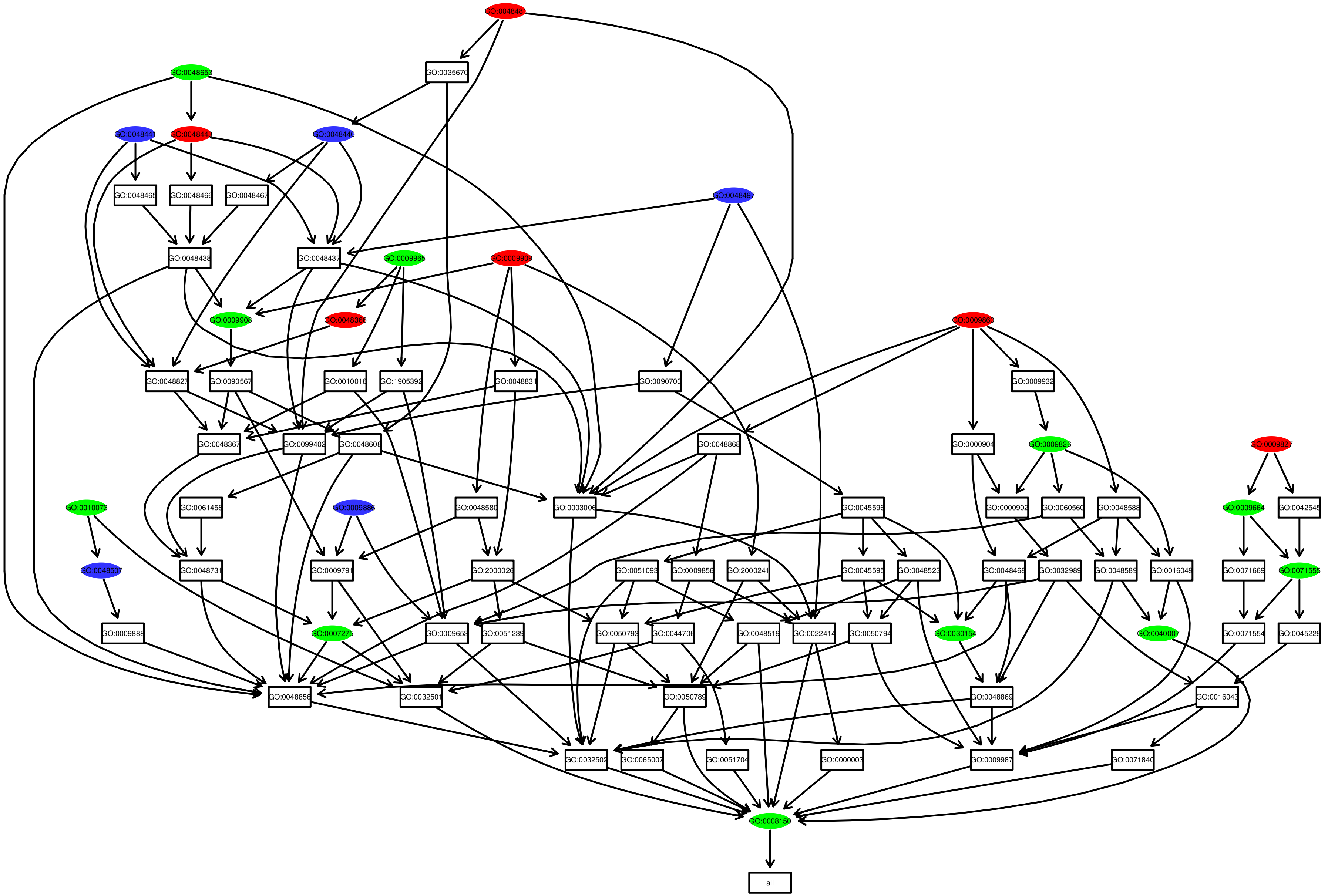

Fig. 4. Gene Ontology (GO) terms and corresponding GO directed acyclical graphs (DAG) for AT2G18960-*AG* induced from the full GO-BP DAG using the set of true annotations plus their direct children that were also predicted by exp2GO. Node color code is as follows: in red, true annotations predicted by exp2GO; in blue, true annotations not found by exp2GO, and in green are exp2GO annotations which are children or ancestors of true annotations.