Protein function prediction for poorly annotated species

Abbreviations

MCMC: Markov chain Monte Carlo
PFP: Protein function prediction
BMRF: Bayesian Markov Random field
MRF: Markov Random field
GO: Gene Ontology
BP: Biological process
PU: Positive-unlabeled
PU-BMRF: Positive-unlabeled learning applied on a Bayesian Markov Random field
CAFA: Critical Assessment of protein Function Annotation (CAFA)
CC: Cellular Component (CC) or
MF: Molecular Function (MF)
AUC: Area under the curve
sd: standard deviation
RN: reliable negative

Introduction

Computational methods for PFP and network-based methods

Protein function prediction (PFP) is one of the most important aims of modern biology. In crop and livestock species, PFP is conventionally based on annotation transfer from the few well-studied species, such as Arabidopisis and human. While successful, these methods rely on the assumption that homologous proteins share function, which has been proved wrong in many cases [22]. Another disadvantage of homologous based methods and in general, of methods based methods on sequence, is that in many cases the biological context of the function cannot be inferred from sequence data. For instance, it is known that there is divergence in biological process annotation for proteins with similar sequences [5]. It is thus desirable to complement the orthology-based methods with other approaches.

Network-based methods, for instance, infer the function of proteins exploiting the principle of guilt-by-association. Based on this, proteins that interact are likely to have similar function [2]. The principle of guilt-by-association does not appply, to a large extend, to the molecular function GO (Gene Ontology) category, but it does holds for the biological processes [22]. Therefore network-based methods can be used to infer the biological process in which the protein is involved. Furthermore, network methods have a lot of potential because they can utilize the accumulating information generated by high-throughput biological experiments, such as co-expression [3] or protein-protein-interactions [4] to construct networks from which to infer function.

Sharan et al (2007) [16] distinguished two types of network methods methods (Figure 1). Direct methods infer the function of a protein based on its neighbors in the network. Module assisted methods, however, first identify modules of related proteins and then for each module, unannotated proteins are assigned a function that is unusually prevalent in the module. Direct methods proved to be slightly superior to the indirect ones.

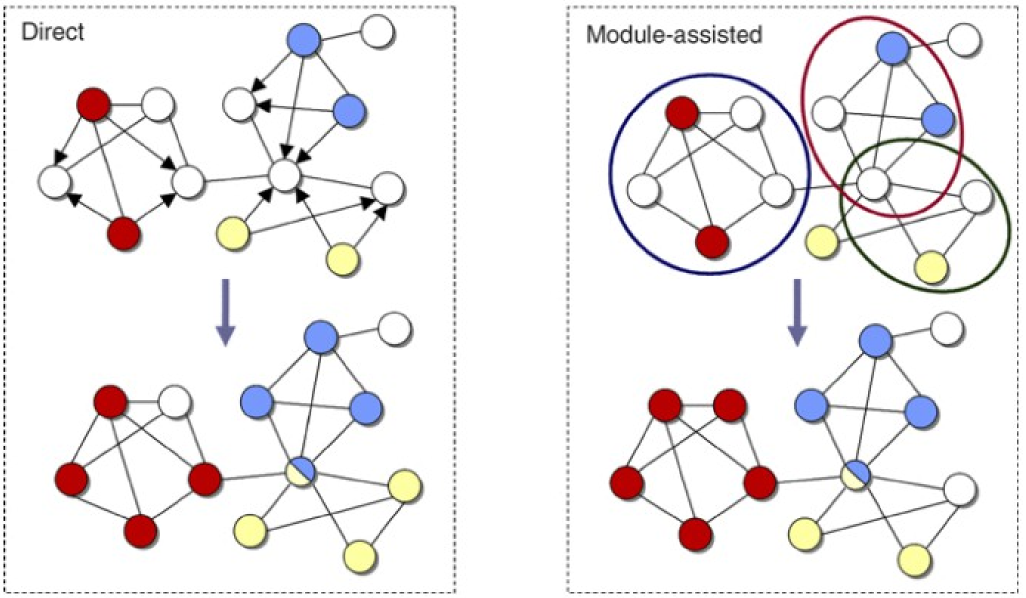


Figure 1: Direct versus Module-assisted network methods.

Source: Sharan et al (2007) [16]

In the network methods, proteins are represented as nodes and the edges may represent co-expression between proteins, or protein-protein-interactions. In the example, the colors represent the different functions and proteins without known functions appear as white. It is assumed that proteins can have more than one function. In the direct methods, the function of a method is inferred from its annotated neighbors (directed graph). In the module-assisted methods, first modules of proteins are identified and then the function of the proteins within a module is determined based on its members.

Networks in poorly annotated species

Because a wide range of data can be combined in these networks, the network approaches seem particularly relevant for poorly annotated species, such as agricultural species, where the validated data of a particular kind (i.e. co-expression, protein-protein-interactions…) is scarce [12]. Network approaches, however, are more challenging for these species because the data may be insufficient to carry a network analysis (Figure 2). A previous study, has shown that it is possible to develop network-based methods that can utilize the limited network resources of some crop species like rice, poplar, soybean and tomato, and achieve accurate PFP [5] by combining different data sources. In their approach, they used co-expression from these species, as well as information from other well annotated species, such as *Arabidopsis Thaliana*.

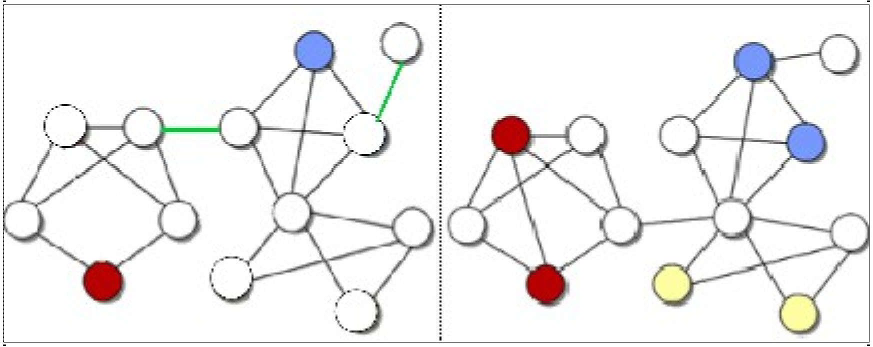


Figure 2: Example of network in poorly annotated species

Figures 2a and 2b correspond to the same hypothetical species. Figure 2a corresponds to a situation in which the species was poorly annotated, and Figure 2b when the annotation improved.

In the poorly annotated situation, there were fewer annotated proteins and the edges were less and less reliable, due to a lack of co-expression or ppi experiments. Edges in green (Figure 1a) were proven false when the annotation improved (Figure 1b).

In livestock species, there is increasing interest for functional annotation. Efforts such as the Functional Annotation of Animal Genomes consortium (FAANG) [6], are currently generating functional annotations regarding genotype to phenotype link for some relevant species such as pig and chicken. Although network data for livestock species may be even more limited than for the aforementioned crop species, there are some studies that have used network data from livestock species. Stanley et al., [12], for instance, used a co-expression networks in chickens to infer function via defining GO-enrichment-modules. In their approach, they stressed the importance of network methods to identify gene modules, as well as regulatory genes that are relevant for a set of functions or co-expression cascades. However, they used a module assisted method, which are slightly less accurate than the direct networks methods, as explained in [16]. Furthermore, the method used by [12] did not make use of statistical learning to identify combinations of features that correlate with certain functions. A review of PFP showed that methods that use statical learning are superior than those that do not [20]. An interesting question therefore is whether it would be possible to achieve accurate PFP in livestock species using direct statistical-learning-based methods. Another interesting question is whether the methodology used for PFP could provide some information regarding the co-expression cascades of genes that redeem biological processes at different levels of specificity.

BMRF

In order to develop a statistical-learning-based network method that is efficient for livestock species, a logical approach is to utilize one of the state-of-art methods used for crop species. Bayesian Markov Random Field (BMRF) [1] is a prediction method that was developed with the purpose of achieving accurate predictions when the data is far from complete. The method can be used as a classifier to distinguish genes that are associated with a certain function form genes that are not.

Markov random fields (MRF) are direct methods in that they exploit the 'guilt by associatuion principle'. This is, nodes in the network that are closer to one another are more likelly to have a similar function. MRF are also Graph theoretic methods [22] in that it seeks to compute annotations for all network nodes at once, while optimizing a global optimization criteria. This is possible only by assuming that the function of each gene is independent from all the other genes in the network except for its neighbors.

The prediction ability of BMRF was compared to other methods in the Critical Assessment of protein Function Annotation (CAFA) [20] experiment and its prediction performance was high for some species (Illustration 1). BMRF is particularly efficient for poorly annotated species for two reasons mainly: First, it can synthesize heterogeneous data into one network. For instance, it can integrate co-expression from the same species, as well as from related species; and second, though Gibbs sampling, BMRF can take into account unlabeled proteins to estimate the parameters of the model.

From a biological perspective, we would expect that the genes that are involved in several specific functions are more prone to be involved in co-expression cascades and therefore we would also expect that they have a larger number of neighbors in the network with whom they share a functional role. BMRF is a network method that directly exploits the information from the neighbors and therefore we would expect that predictions with BMRF will be more accurate for those genes that are co-expressed with genes that share a functional role. Consequently, it is interesting to investigate whether BMRF could be useful for the task of identifying relevant genes in the co-expression cascades. Questions to be addressed are whether the more specific GO terms have a higher degree of connection in the network and whether this is translated into a better accuracy of PFP with BMRF. Also, whether genes that are associated with a large number of GO terms of different levels of specificity are better predicted with BMRF.

Since the current methods for PFP in livestock species are not very effective in prediction biologicla processes, and given that network based methods, like BMRF, are particularly effective for these, in this study we aimed to assign genes to GO-terms from the Biological process category.

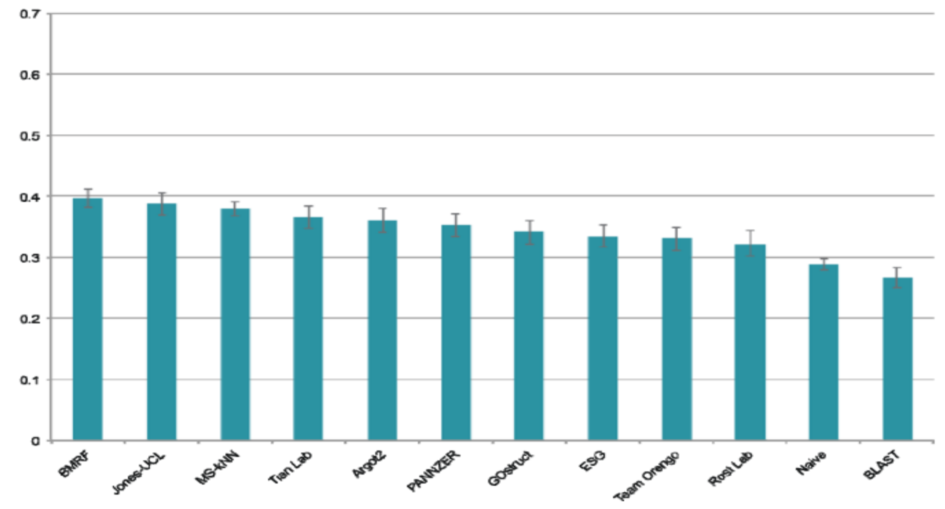


Figure 3: Comparison of BMRF with other PFP methods. Evaluation for the Biological process category in H. Sapiens.

Source: Radivojac et al (2013) [20].

Positive-unlabbeled learning

A common limitation of BMRF, and of learning methods in general, is that, in order to train a classifier, two differentiated classes of elements are required. For instance, in the PFP context, the classifiers expects that the input data consists of two classes of elements: proteins that have the function (function), and proteins that do not (negative). However, from a biological perspective, it is very difficult to find negative examples, so genes that do not have a certain function, because in biology the lack of evidence for a connection does not imply that such a connection does not exist. For this reason, the learning process in the PFP context may be biased because the classifiers attempt to solve a one class classification problem when in fact the annotated data consists solely on positive cases. As an example, for rice 415 proteins have experimental evidence for a biological process, but not a single protein has a validated proof of no-connection with a function [5]. Since only positive associations are reported, the negative set is composed of all unlabeled data. This leads to some bias in the prediction because the unlabeled data may contain some positive cases. This problem increases with increasing numbers of unannotated proteins, such as in the case of network data from livestock species. To overcome this problem, a new type of machine learning has emerged called Positive Unlabeled learning (PU). With PU it is possible to identify the proteins that are more unlikely to have a given function (Figure 4). Hence, the number of unlabeled cases can be minimized by extracting some “reliable negatives” proteins from the set of unlabeled. It has been proved theoretically that, by identifying sets of reliable negatives, PU improves the performance of machine learning algorithms in situations where only positive labels are known [7]. PU has been successfully applied to a variety of problems related to PFP [7-10,13]. In [7], for instance, the authors extracted a set of reliable-negatives proteins from the unlabeled dataset by defining a threshold of similarity based on the euclidean distance between a set of positive proteins and the set of unlabeled. Yang et al. [8] developed a multi-label version of PU learning to identify genes associated with diseases; [9] developed two novel approaches to identify reliable negatives that can be applied in different algorithms. Jiang et al. [10] applied PU on a support vector machine and outperformed all pre-existing methods for pupylation sites prediction, and then Nan et al. [13] improved their method by adding as a first step to the algorithm in [10], the method described in [7]. Lastly, in another recent study, Nusrath et al. [14] used a Self organizing map to extract reliable negatives from unlabeled data-set of drug-drug interactions. None of these studies, however, have applied PU learning on BMRF. BMRF was developed with the purpose of being able to handle networks with a large portion of unknown cases, and PU can extract reliable information from these unknowns. Our hypothesis therefore is that a PU implementation of BMRF will be particularly effective for PFP in species for which the portion of unannotated proteins is large, such as pig and chicken. Furthermore, the guilt-by-association principle that BMRF exploits could potentially be used to identify genes that play relevant roles in the co-expression cascades.

The aim of this study is to develop a PU implementation of an existing Bayesian Markov Random Field algorithm that can efficiently assign genes to biological processes using network data from chickens. The methodology could potentially be useful for identifying genes that play relevant roles in the co-expression cascades.

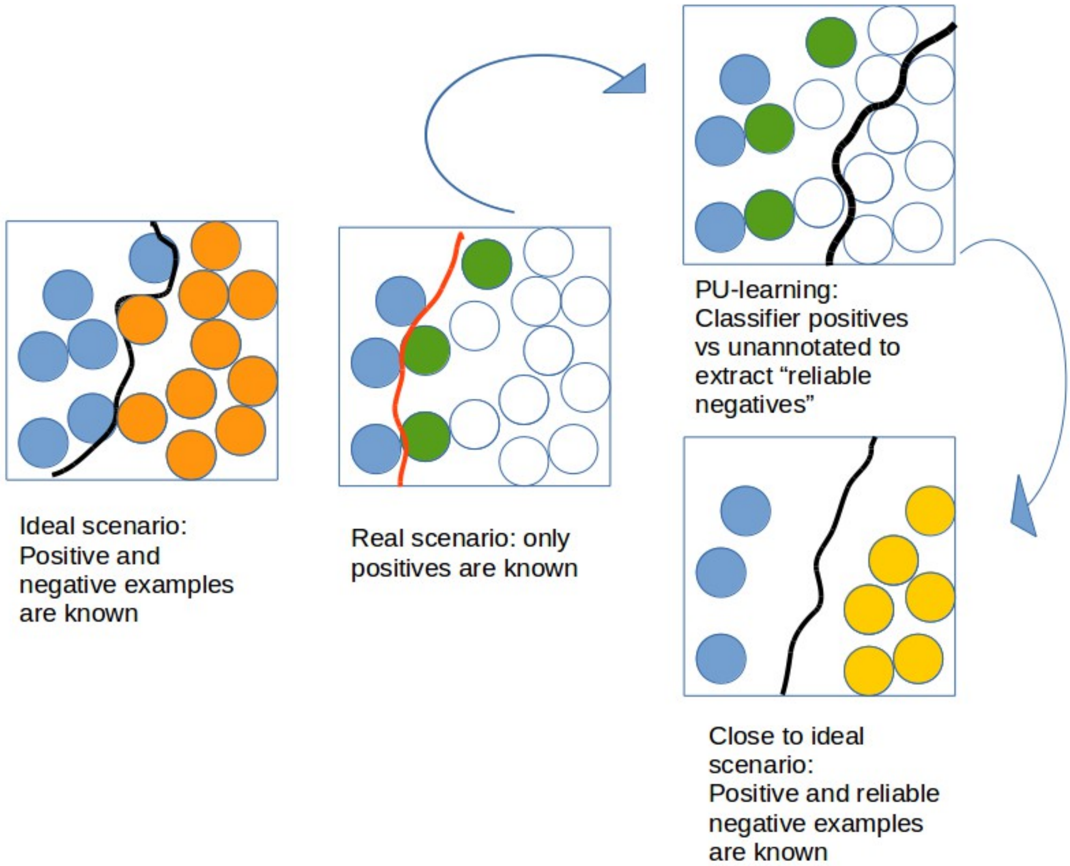


Figure 4: PU-learning diagram

In the ideal scenario (figure 4a), examples from two classes (positives and negatives) are known and it is possible to train a classifier. Such a classifier can then be used to infer whether the novel examples correspond to one or another class. In the PFP context, negatives examples are not known (Figure 4b). Examples in green are false negatives, so for instance, in the PFP context, they may be proteins that have the function of interest but whose association with the function has not been discovered yet. White examples are proteins that do not have the function, although it is not possible to prove it. Green and white examples compose a “non-positive class”. When we try to train a classifier positives versus non-positive class, the classifier will interpret that it has to differentiate between the positives and the “non positive class”, and as a consequence the boundary may be too strict. This is likely to cause false negatives when the classifier is used to assign novel examples to one of the two classes.

PU-learning tries to solve this problem by first extracting from the “non-positive class” a set of “reliable negatives” (Figure 4c) and then training a classifier positives versus reliable negatives (Figure 4d). The reliable negatives are a subset of non-positive example that is significantly different from the positive set and therefore it is expected that there are not false negatives in that subset. Thus, the reliable negatives can be used as a representative set of the negative class, and the classification scenario will very close to the ideal.

Thesis setup

The thesis consists of 5 parts. Part 1 includes some preliminary analysis to choose the values for the fixed parameters of BMRF, as well as the number of replicates that was required to achieve reproducible results. In part 1, we also carried analysis to choose the co-expression threshold for chickens. This step was required because for chickens the co-expression data is limited and using the convention co-expression threshold (Pearson Correlation:0.7) led to an excesevilly low number of edges.

Parts 2 and 3 aimed two objectives (both parts addressed both objectives). The first objective was to investigate for which species and GO-terms BMRF achieves accurate PFP. This information can be useful to get some insights about what is a “poorly annotated species” when it comes to PFP via networks and what is the minimum data that is required to achieve accurate PFP via network methods. The second aim was to investigate whether PU-learning is a good improvement strategy for PFP using BMRF. Although PU-learning is expected to help in situations suhc as in this study, where only positive examples are known, it may be that BMRF could be more greatly improved with other strategies like, for instance, accounting for non-validated data. Furthermore, it may be the case that the best way to improve PFP using network data is not by improving the method but by improving the quality of the data.

In Part 4, PU-BMRF was developed and its performance was evaluated.

In part 5 we aimed to identify genes that may be playing relevant roles in the co-exopression cascades that redeem biological processes at different levels of specificity. Here we addressed four questions, mainly:

(1) Whether the more specific GO terms have a higher degree of connection in the network. Also whether this is true when we look only within the genes that have a certain function. If this was the case, this could potentially mean that certain specific functions play relevant roles in the co-expression cascades.

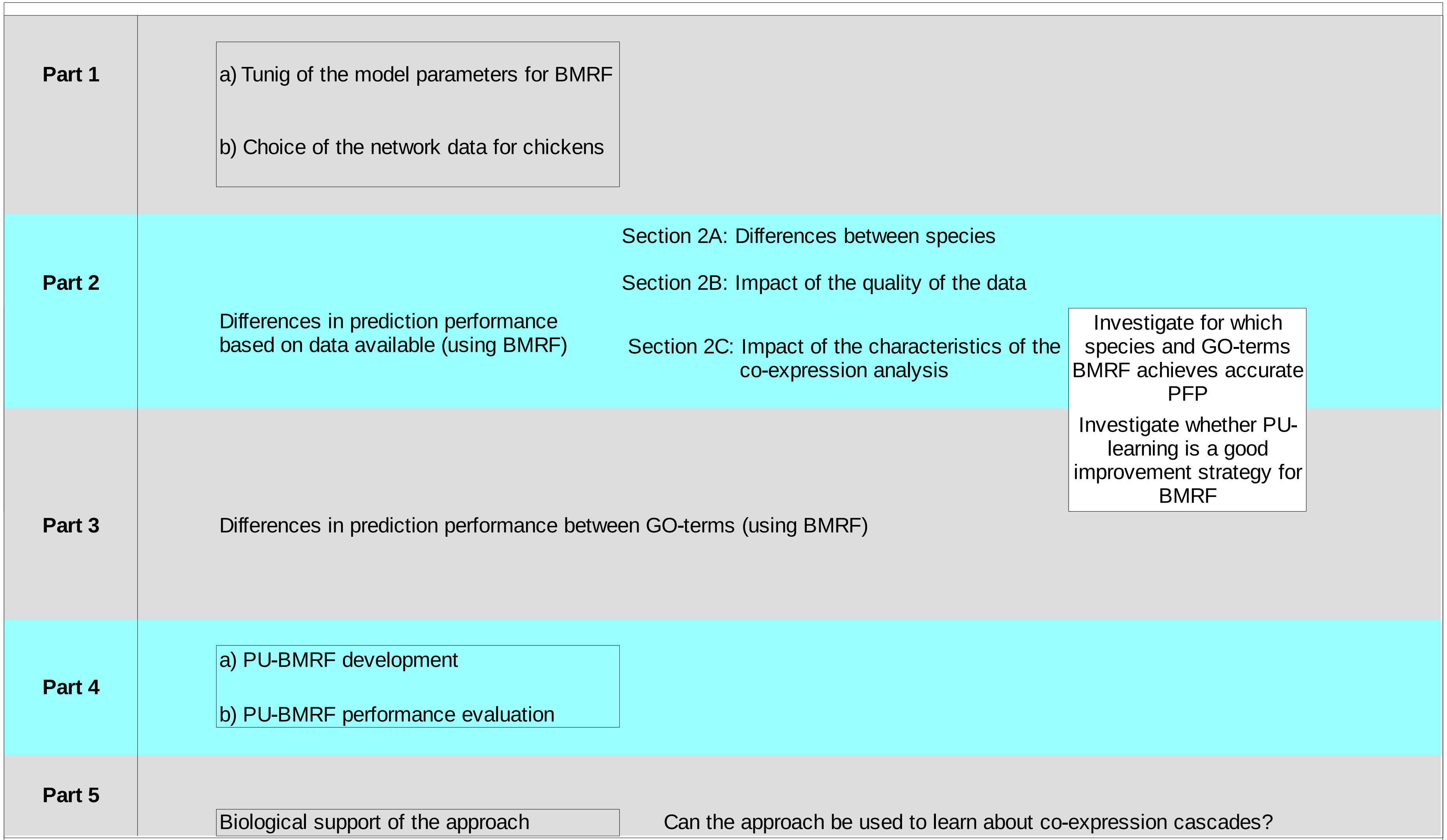
(2) Whether genes with a large number of edges are involved in more functions, and whether these functions are specific or general.

(2) whether genes that are highly connected are more accurately predicted with BMRF. BMRF directly exploits neighbor information, and therefore BMRF can be used as a short of validation that the co-expression that we observed in the data is actually consistent with having a similar function. In other words, if BMRF achieves accurate prediction in those particular cases, then we can expect that the principle of guilt by association works in those specific cases. And therefore, genes that are highly co-express actually are involved in more functions.

and (3) whether genes that are associated with a large number of GO terms of different levels of specificity are better predicted with BMRF.

The thesis contains 3 appendixes.

- Appendix I-Concepts. This appendix contains the definitions of most of the terms and concepts that were used in the thesis.
- Appendix II-Data overview. In this appendix we provide a more in-depth overview of the differences between the data of the species considered.
- Appendix III- Additional results. Here we included the results that were not relevant enough to be in the results but that still can provide some valuable information.



Part 1

a) Tunig of the model parameters for BMRF

b) Choice of the network data for chickens

Part 2

Differences in prediction performance based on data available (using BMRF)

Section 2A: Differences between species

Section 2B: Impact of the quality of the data

Section 2C: Impact of the characteristics of the co-expression analysis

Investigate for which species and GO-terms BMRF achieves accurate PFP

Investigate whether PU-learning is a good improvement strategy for BMRF

Part 3

Differences in prediction performance between GO-terms (using BMRF)

Part 4

a) PU-BMRF development

b) PU-BMRF performance evaluation

Part 5

Biological support of the approach

Can the approach be used to learn about co-expression cascades?

Figure 5: Thesis setup

BMRF (Bayesian Markov Random Field), GO-term (Gene-Ontology term), PFP (Protein Function Prediction), PU: Positive-Unlabeled learning

Material and methods

Basics of the network method

BMRF, same as other network methods, defines one network per GO term and for each network there are two classes of genes: those that are known to be associated with the GO term and those that are not known to do so. The genes are the nodes in the network, and they could be interpreted as having two different colors, one for each class. The “colors” or labels are the response variably (binary). Conventionally, genes that are not known to have the function are labeled as '0' (or white color), and the genes that are known to have the function have a label '1' (and a color other than white).
The edges represent co-expression between pairs of genes or protein-protein-interactions (ppi).

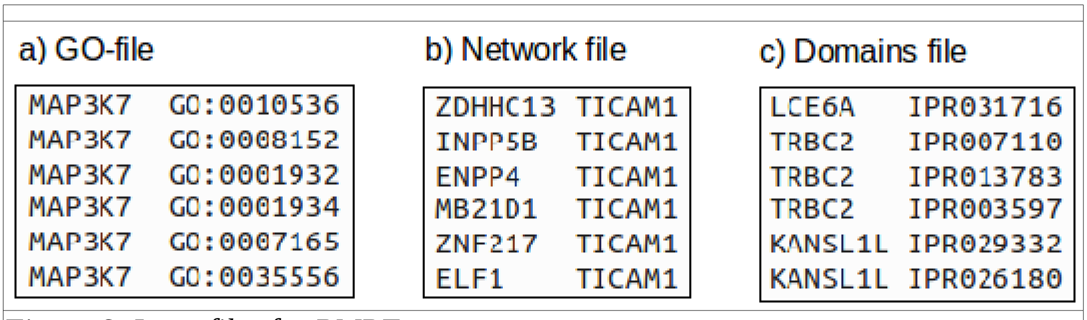
The analyzes are carried independently for each GO term and the prediction accuracy is given per GO term (instead of per gene). Thus, at the end of the analysis we will have a measure of accuracy of prediction AUC (Area Under the curve) for each Go term, which expresses how accurately BMRF predicted which genes have the function and which genes do not. GO terms to be predicted are those in the category of Biological process because these cannot be predicted with the sequence based method that are currently used for PFP in livestock.

Data preparation

For the networks, we used co-expression data from three species: yeast, human and chickens. In the case of yeast, ppi (protein-protein-interaction) data was used as well as co-expression data. Most of the analysis were carried in the three species and yeast ppi. However, some analysis were carried on yeast data because it was easily accessible, whereas some other analysis were carried on human as we though that chicken data would not become available and it resembles more the situation in chickens.

Chickens is considered as a poorly annotated species. Yeast and human data were used as an upper bound for the prediction performance, since for these species there is extensive annotation data available. Carrying the analysis with four different types of data (three species and yeast ppi) is advantageous in that the scope of the results can be more general. However, due to time constraints only chicken data was used for the development and evaluation of PU-BMRF, in part 4. Data sources for the these species are given in Table X in Appendix II-Data overview.

The BMRF code that was used for this thesis [1] takes three files as input (figure X).



a) GO-file

```
MAP3K7   GO:0010536
MAP3K7   GO:0008152
MAP3K7   GO:0001932
MAP3K7   GO:0001934
MAP3K7   GO:0007165
MAP3K7   GO:0035556
```

b) Network file

```
ZDHHC13  TICAM1
INPP5B   TICAM1
ENPP4    TICAM1
MB21D1   TICAM1
ZNF217   TICAM1
ELF1     TICAM1
```

c) Domains file

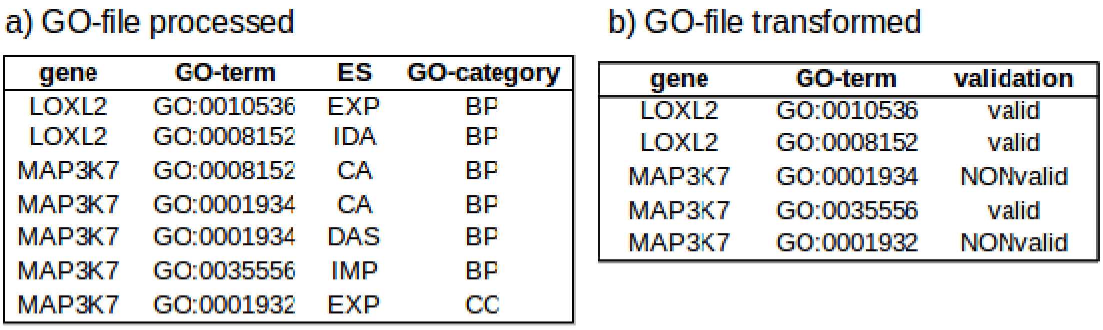
```
LCE6A    IPR031716
TRBC2    IPR007110
TRBC2    IPR013783
TRBC2    IPR003597
KANSL1L  IPR029332
KANSL1L  IPR026180
```

Figure 6: Input files for BMRF

The GO file (Figure 6a) consist of associations between genes and GO-terms. The network file (Figure 6b) consists of edges or connections between pairs of genes. The Domains file consists of associations between genes and domains.

In the GO- file, the associations were coded as "valid" if, for at least on of the association available in data, they correspond to Experimental evidence scores (precisely: 'EXP', 'IDA', 'IEP', 'IMP', 'IPI', 'IGI'), and as "NONvalid" otherwise (Figure 7). Also, since we are only interested in the category of biological process (BP), we include all the associations regarding the Cellular Component (CC) or Molecular Function (MF) categories in the group of

“NONvalid”. This way we got a single estimate regarding to which extend non-validated data from the BP category together with the validated and non-validated data from CC and MF, contribute to the accuracy of prediction of BP GO-terms. The reason why these were combined was to reduce the number of analysis. For some analysis with yeast data, however, we distinguished between (1) predictions with only BP-valid associations, (2) analysis with BP-valid associations and BP-non valid associations, (3) analysis with BP-valid association and MF and CC valid and non-valid associations, and (4) analysis with all types of associations.
Note that “valid” and “NONvalid” are just a level in a binary class that is defined for each associations and should not be misleaded with the labels that are asigned to the different gene for each GO term (0,1). These labels are explained in the section “Validation in BMRF”.



a) GO-file processed

gene	GO-term	ES	GO-category
LOXL2	GO:0010536	EXP	BP
LOXL2	GO:0008152	IDA	BP
MAP3K7	GO:0008152	CA	BP
MAP3K7	GO:0001934	CA	BP
MAP3K7	GO:0001934	DAS	BP
MAP3K7	GO:0035556	IMP	BP
MAP3K7	GO:0001932	EXP	CC

b) GO-file transformed

gene	GO-term	validation
LOXL2	GO:0010536	valid
LOXL2	GO:0008152	valid
MAP3K7	GO:0001934	NONvalid
MAP3K7	GO:0035556	valid
MAP3K7	GO:0001932	NONvalid

Figure 7: GO-files manipulation

Associations in the “GO-file processed” (Figure 7a) contain experimental scores, as well as information regarding the GO category. This information is simplified in Figure 7b by differentiating only between “valid” (associations with experimental evidence scores: 'EXP', 'IDA', 'IEP', 'IMP', 'IPI', 'IGI'; and GO-category:BP), and “NONvalid”: other experiment scores, and/or GO-category different than BP. In Figure 7b). The GO-file transformed is ready to be used as input for BMRF with the advantage that we can decide at a given moment whether we want to include or not also the NONvalid associations.

In order to maintain this distinction throughout the analysis between "valid" and "non-valid" associations these two groups of associations were up-propagated independently and then both sets of associations were combined into a single file, adding a third variable depending on whether the association corresponds to “valid or NONvalid” (table XX in Appendix II-Overview_data).

Domain and GO-terms files were pruned to exclude genes that are not available in the network file, as a requirement for the BMRF code. Then, the GO-size filter was applied to exclude the GO-terms that were too general or whose number of known associated genes was excessively low for the BMRF computations (Appendix I). The GO-size filter is based solely on the "valid" associations for two reasons: (1) the non-valid associations are not used in the validation and (2) the GO-size filter allows to make sure that there are enough number of genes in the validation. Analogous to the GO-size filter, BMRF uses another filter to exclude from the analysis the domains whose number of genes is below a certain threshold. This filter is named as 'DF-size filter' and is defined in table XX in Appendix I-Concepts.

Table XX in Appendix II-Data-overview shows one example of the first rows of each file type after data preparation.

Markov random fields

Markov random fields (MRF) are random field that satisfies the markov properties. In a MRF the most likely discrete class of an element can be predicted by the join probability distribution of its neighbors. Thus, in MRF first the priors and conditional probabilities of the annotations are computed followed by the joint likelihood of all target annotations [22].

A typical application of MRF is image restoration, where the value of a pixel (color) can be predicted based on the value of the neighbors pixels. The strength of MRF to infer the class of an element is that they allow for simultaneous predictions of many elements, and therefore they may achieve accurate predictions even when non of the neighbors are unknown. MRF are successful, in general, for prediction problems in which the principle of guilt-by-association holds, like in image restoration or PFP.

In a MRF the probability of a certain assignment of discrete states $x=x_1,...x_N$ is, as explained by Sharan et al, (2007) [21]:

$$P(x)=\frac{1}{Z}\exp(-H(x))=\frac{1}{Z}\exp(-\sum_{c\in C} H_c(X_c))$$

Equation 1: General formula of MRF for the probability of a certain assignment of discrete states

Where N is the total number of variables, Z is the normalizing constant, C is the set of all the cliques in the network, H_c is a potential function associated with clique c and X_c is the assignment of states to the members of c.

Computing this is hard and it is common to keep the equation to the second order and homogenize it by defining the same potential function for all cliques of the same size. Thus, we have:

$$H(x)=\sum_{v\in V} H_1(X_v)+\sum_{(u,v)\in E} H_2(X_{u,v})$$

Equation 2: Predictions with homogeneous second-order MRF

Deng et al, (2003) [3] adapted a MRF to the PFP problem and stated that the probability over the enter network is proportional to $\exp(\alpha N_{01}+\beta N_{11}+N_{00})$, where N_{01}, N_{11},N_{00}, correspond to the number of pairs of proteins that: while interacting, only one has the function; both of them have the function; and non of them has the function, respectively. And α and β are weighting the contribution of each of these classes of pairs of proteins. Then, by combining the a priory probability of an assignment with N_1 '1's, which depends on the frequency of the function and is proportional to $(f/(1 - f))^{N}{}_{,1}$ they obtained a homogeneous second order MRF and the following equation to estimate the probability that protein *v* is assigned with the function of interest given the annotations of its neighbors *N(v):*

$$P(X_{(v)}=1|X_{N(v)})=logit(\log\frac{f}{1-f}+\beta N(v,1)+\alpha(N(v,1)-N(v,0))-N(v,0))$$

Equation 3: Probability of a gene having a function given its neighbors, using MRF

where N(v,i) is the number of neighbors of v that are assigned with i∈{0,1} and logit is the logistic function logit(x)=1/(1+e−x). Deng et al (2003) [3] proposed to estimate the two parameters of the model using a quasi-likelihood method and then apply Gibbs sampling to infer the unknown functional annotations. The approach has two steps: first, the parameters are estimated and, second, the label is inferred using Gibbs sampling. The parameters are estimated by maximizing the pseudo-likelihood function with logistic regression.

For this, each protein is interpreted as a statistical unit, the predictors are N(v,1) and N(v,0), and the response is the assignment. However, because some proteins are not annotated, the response will be missing for these and there will be uncertainty within the predictors. Deng et al (2003) [3], overcame this by simply ignoring the unannotated proteins in the parameter estimation step. This can be problematic when the number of unknown proteins is large, like in the case of poorly annotated species, because by ignoring the unknowns, the neighbors are pruned and they may no longer express the complexity of the network.

Bayesian markov random fields

Kourmpetis et al, (2010) [1] developed a Bayesian Markov Random field (BMRF) to overcame the aforementioned problem. In BMRF, a Markov chain Monte Carlo (MCMC) algorithm is used to sample from the joint posterior density of α and β. Thus, the label of the proteins is iteratively updated conditionally on the parameters α and β though Gibbs sampling. Then, a candidate point θ'=(α', β') is obtained using the equation:

$$\theta' = \theta + \gamma(Z_{R1} - Z_{R2}) + \varepsilon$$

Equation 4: Parameter update in BMRF using the Differential Evolution Markov Chain algorithm

where θ denotes the current state of the parameter vector, γ~U(γ'/2,γ') is the scaling parameter and γ'=2.38/(√2d) is the optimal step size[41], where d is the parameter dimension (d=2, in this case). ZR1 and ZR2 are randomly selected form past samples of the Markov Chains stored in the matrix Z and e~MVN(0,10^{-4}). θ' is accepted using a Metropolis step, with probability:

$$r = min\left(1, \frac{PLF(x^{(t)}|\theta')}{PLF(x^{(t)}|\theta)}\right)$$

Equation 5: Probability at which θ' is accepted in BMRF using a Metropolis step

Then, the labeling vector x is initialized using the output of the MRF defined by Deng et al, (2003) [3], as explained by Kourmpetis et al, (2010) [1].

Validation in BMRF

BMRF [1] was used for PFP and learn about the impact of different method and network parameters on the prediction performance. In this framework, predictions are made individually for each GO-term that passes the GO-size filter (see Appendix I-Concepts). The predictions, however are not completely independent for each GO-term in the dataset because the genes that are not associated with any of the GO-terms (unknown genes) are treated differently. Note that the number of unknown genes depends on the number of GO-terms in the database, which is regulated through the GO-size filter. We investigated to which extend the GO-size filter and the number of unknowns affect the prediction performance.

Since predictions are made independently for each GO term, each gene will have one label per GO term. However, it should be noted that the unknown genes are always labeled as '-1'. This is because the unknown genes are classified as 'unknown' based on the data considered in the analysis, not on the GO-term.

A distinction should be taken into account between 'unlabeled genes', which are genes that are not known to have the function, and 'unknown genes', which are genes that are not known to have the function and that are not known to be associated with any of the GO terms in the dataset. These genes are labeled as '0' and '-1', respectively. In the validation process, genes in the test set will also be labeled with a '-1', similarly to the 'unknown genes'. In other words, the label '-1' can be used to hide the labels.

The reason why the label '-1' is considered in the BMRF code is that, since these genes have never been predicted as positives, they are less likely to be non-associated for a given function than those genes that have been found as positives for some function but not for the function of interest. This has to do with the fact that some function are more difficult to predict than others. Thus, if based on data, a gene has never been identified as positive for any function, it is more fair to assume that the function is particularly difficult to predict using experimental approaches, than assuming that a very low number of genes have the function. In other words, a large portion of unlabeled genes ("0") implies that the function is rare, (almost never observed), whereas a large portion of -1s implies that the function is difficult to predict. As explained before, this distinction between label '0' and label '-1' additionally, allows to do predictions in a fairer fashion, by assigning label '-1' to the genes in the test set, as explained before. One additional aspect to be considered is that if the portion of "-1" becomes excessively large with respect to the portion of "0", Gibbs sampling may fail in the relabeling, because it may expect that the portion of genes that have the function is very large.

Training and test sets

The training set consists of those genes that enter the BMRF code with a '1' (associated with the GO-term of interest), or a '0' (otherwise). Genes in the test set are labelled as '-1', as explained before. This way, the real label of the genes ramians hidden until the posteriors have been estimated using the BMRF code. The output of the code is vector of posteriors corresponding to the probability that each of the genes in the database has be associated with the function. Then, the AUC for the GO-term of interest will be computed by contrasting the posteriors of the genes in the test set with the label that was hidden (either a '1' if the gene is known to be associated with the function, or '0' otherwise).

The process of labeling in the different folds of the k-fold CV is illustrated in table 1:

Gene classes in BMRF	Gene class as defined in Appendix I-concepts	Label in BMRF input	Expected label in BMRF output	Fold
Associated with the function but the association is non-validated	Positive “NONvalid”	1*		
Not known to be associated with any function	unkown	-1		
Associated with the function. Assoc is validated but it is masked in this fold by labeling as '-1'	Positive-test	-1	1	a
Not associated with the function. This information is masked in this fold by labeling as '-1'	Unlabeled-test	-1	0	b
Associated with the function. Assoc is validated and it is not masked in this fold	Positive-train	1		a
Not associated with the function. This information is not masked in this fold	Unlabeled-train	0		b

Table 1: Labeling of genes at a given fold in the k-fold CV.

** The genes only takes label '1' if the parameter Only_EES is set to 'False', otherwise these genes are excluded from the analysis.*

a: k-fold CV for the positives. The positive genes (this includes all genes that are associated with the function and whose association is validated) are divided into k sets until the end of the analysis. In each fold, the genes in one of these k sets will be in Positive-test, the rest will be in Positive-train. Within the same analysis, this is repeated k times. Each time corresponds to a fold with a new labeling configuration such as represented in Table 1. Note that only the label of those gene classes that have 'a' or 'b' in the 'Fold column' change with each fold. For each fold, a new k set takes place as Positive-test and the rest are Positive-train, until each of the the k sets defined at the beginning of the analysis, have been assigned once to the Positive-test.

b: k-fold CV for the unlabeled. Same as 'a' but for unlabeled genes instead of positives.

As explained in figure 5, the thesis consist of 5 parts. The rest of the Materiel and Methods is dedicated to describe the methodology in each of these parts. The parameters and features that were dfined were also described in Appendix-I-Concepts.

Unless specified, all analysis were carried out with the network data and the values for model parameters chosen in part 1, except for analysis in Part 4 that were carried only with 4 replicates, due to time constraints. In order to estimate the reproducibility of the approach, the standard deviation across folds within the same replicate, as well as across replicates, were computed for each analysis.

Part 1- Tuning the model parameters and choosing the data for prediction performance with BMRF.

The BMRF code that was used in this thesis has four main fixed parameters whose impact on the prediction performance were investigated. These parameters are GO-size filter, Number of folds in k-validation, Number of iteration in Gibbs sampling.

However, the first analysis aimed to choose how many replicates were required to achieve reproducible results. Note that the assignations of genes to folds in the k-fold CV is a random process that may cause certain variability across runs. For this study it was considered that results were stable when the standard deviation (sd) across replicates of the analysis was below 0.02 AUC measures.

The GO-size filter regulates the GO-terms that are considered in the analysis. The filter consists of two values: 'minGOsize' and 'maxGOsize'. minGOsize specifies the minimum number of genes that the GO term should be associated with, and maxGOsize specifies the maximum. Both filters refer to the validated associations. In the case of minGOsize, the argument should be an integer larger than 0. An argument of 10, for instance, for minGOsize would exclude from the analysis all the GO terms whose number of known associated genes is less than 10. In the case of maxGOsize, the numeric should be a numeric between 0 and 1, since the value corresponds to the portion of genes in the network. Thus, if for instance, there are 10,000 genes in the network and the argument is 0.3, GO-terms that have more than 3000 associated genes will be excluded form the analysis.

Analogous to the GO-size filter, the BMRF has another filter for the domain size. This regulates the size of the domains that enter the analysis. The filter also consists of two values: minDFsize and maxDFsize. MinDFsize and maxGOszie specify the minimum and maximum number of genes that each domain should be associated with, respectively. Both argument should be an integer larger than 0.

A novel model parameter was defined in this study. 'only_EES" stands for "only experimental evidence scores (EES)". The argument is a logical that is true if we want to remove from the analysis all the association between genes and GO-terms that are not validated. If the logical is true, the non-validated associations will be included in the train set (they will never be part of the test set), as explained in Table 1. However, we also include in the NON-validated set all the associations (either validated or not validated that regartd the Cellular component and Molecular function GO categories), as explained in figure 7,

Details about which values were considered for each model parameter are given in section 1AB in Appendix III-Additional results.

Part 1 also includes the choice of the co-expression threshold for chickens given a conditionally independent co-expression network. Conventionally, a Pearson correlation of 0.7 is used as a threshold for co-expression. However, in the case of chickens, the number of co-expression experiments is limited and using a Pearson correlation of 0.7 would lead to a scarcity of network data. This is important because in BMRF the Gofile needs to pruned accordingly with the network file. This is because in the BMRF code genes in the GO file that are not in the network are not allowed. We computed AUC for datsetes with diffrenet Pearson correlations co-expression thresholds. Note that for each person correlation, the GO file needs to be up-propagted. Details about which Pearson correlation cutoffs were considered are given in section 1A in Appendix III-Additional results.

Part 2- Differences in prediction performance based on data available (using BMRF)

In part 2 we focus on how the network and the GO-files influence the prediction performance. Interets9i s in knowing which type of data is more suited for PFP via BMRF and what is the minimum data required. Also, whether the quality of the data is directly linked to the prediction performance or which species are better predicted, species with extensive datasets or species for which a large portion of the annotations are known. Part 3 consists of 3 sections:

Section 2A): Investigate the differences in prediction performance between species
Section 2B): Investigate the impact of the quality of the data on the prediction performance
Section 2C): Investigate the Impact of the nature of the network on the prediction performance

Section 2A) Investigate the differences in prediction performance between species

In this section we analyzed the differences in data between the species considered and we investigated how this differences are translated in a different prediction performance.

Differences in network data

A total of 10 parameters were defined to account of the differences in the network data between species. These parameters are a unique value for each data-set considered. For instance, a unique value for the yeast co-expression data-set, and a unique value for the chicken co-expression data when the Person correlation threshold was 0.35. However, with the exception of the first three parameter considered ('#te' and '#edges per gene', '#epp per GO'), for the other parameters considered the value corresponds to the sum of the values corresponding to the GO terms in the data-set. The 10 parameters considered were:

- #te (total edges): Number of edges in the network
- #edges per gene: This parameter corresponds to the distribution of the number of edges per gene in the data-set. Thus it is a vector of length equal to the number of genes in the data-set.
- #epp per GO
- #epp (edges-positive-positive): Number of edges that connect genes that are known to be associated for the same species. Thus, in principle, there is one value of #epp per GO trem, but in part 2 we refer to the sum of all the GO terms. The same holds for #epn and #enn, #epp*100/#te and #epp/tpepp.
- #epn (edges-positive-negative): Number of edges that connect genes that are known to have a given function with genes that are not know to have the same function.
- #enn (edges-negative-negative): Number of edges that connect two genes that are not known to be associated with a given function.
- #epp*100/#te (edges positive-positive divided by total edges): The ratio between the #epp and #te (the total number of edges of the network), to allow for a fairer comparison between species.
- #epp/tpepp (edges positive-positive divided by total possible edges positive positive). The ratio between #epp and tpepp. Tpepp is calcuated as: n*(n-1)/2, where n is the number of genes that are associated with the GO Term.
- #epp/tpepp standarized. The ratio between #epp and tpepp.

Differences in annotation data

Three parameters were considered to quantify the level of annotation (one individual value per data-set):

- #genes per GO-term. The average of the # genes associated with each GO term
- #GO-terms per gene. The average of the # GO-terms that are associated with each GO term.
- #assoc*1000/total possible assoc, number of gene-GOterm associations of a species divided by the total number of possible associations considering that each gene could potentially be associated with every GO-term. This parameter is a measure of the degree of “competitiveness” of the GO-file.

Differences across the species considered for the aforementioned parameters can be found in Appendix II-data overview, as well as in section 3a in Appendix III-Additional results. Appendix II provides a more graphical representation of the differences between the data of the different species, whereas Appendix III focuses on how these differences may be affecting the prediction performance. Formal definitions of these parameters can be found in Appendix I-concepts.

Section 2B) Investigate the Impact of the nature of the network on the prediction performance

Investigate the impact of the quality of the data on the prediction performance

In order to investigate the impact that the quality of the data has on the predictions, the AUC was computed after randomly removing associations and edges from the data. We distinguished between 2 types of associations: association of the GO-term of interest and associations of other GO-terms; and four types of edges: epp, epn, enn, te, as described in Section 2A.. Portions subtracted were 0, 10, 30, 50, 90 and 95% when yeast data was used; and 0, 5, 10, 20, 40, 60, 80, 99% when human data was used. Then the correlation between AUC and the percentage of edges (or associations removed) was computed in order to asses the impact of the extension of the data considered and the prediction performance.

Section 2C) Investigate the impact of the nature of the network on the prediction performance

In this section, we investigated how the conditions of the co-expression analysis influence the prediction performance. For this, we took different subsets of co-expression data and assessed the prediction performance in each case.

The source of expression data for yeast is organized based on experiments and a brief description of these is provided. Thus, we searched for key words, like “oxidation” or “stress” and created subsets of networks with data from those experiments that have those words in the description of the experiment. In the case of humans, data is organized by tissues. Thus, each tissue was a considered a different subset. We homogenized the subsets based on epp/tpepp instead of #te, because in Part 1 of the thesis we observed that the former parameter has barely any impact on the prediction performance. In order to homogenize the size of the subsets, we removed randomly edges from the network. Then, we computed AUC with the expression data from the different tissue experiments.

We investigated the global effect that the 'nature' of the network has on the prediction performance, and we searched for evidence of biological support. For instance, from a biological perspective, we would expect that a co-expression analysis carried for one specific tissue will allow for better predictions in those GO terms whose function is more relevant for that particular tissue.

Part 3- Differences in prediction performance between GO-terms (using BMRF)

We Investigated the impact of different GO-term-properties on the prediction performance
A total of 9 GO-term-properties were defined. Each of them corresponds to one value per GO-term. The value is also expected to be different across species for the same GO term:
epp/tpEpp: (edges positive-positive divided by total possible edges positive positive), described in part 2, but here it refers to one value per GO-term:

- eppA/tpEppA: (edges positive-positive divided by total possible edges positive positive, including also NONvalid associations). As epp/tpEpp but includes also gen-go associations coded as NONvalid. NONvalid associations are described in Figure 7.
- #genes: Number of genes that are associated with the GO term. Only validated associations were considered.
- spec (specificity): The inverse of the sum of all the validated genes from the 4 species considered.
- #epp/tpe (e#pp divided by total possible edges): #epp divided by the total number of edges

that connect any gene that is associated with the function of interest with any other gene of the network. Tpe is the sum of epp and epn.

- Depth: depth of the GO term in the GO hierarchy. Range of values are the integers from 1 to 15, 1 being the depth of the most general GO terms.
- AUC (Area under the curve): Prediction performance of the GO term
- sdAUC (standard deviation of AUC): mean of the standard deviation across replicates, for the GO term of interest.

Given the definition of depth, the most general GO terms have a lower depth than the most specific Go terms. However, it should be considered that depth is not a good estimate of the specificity of the GO terms, because in the GO hierarchy, different branches have different levels. Thus, GO terms with similar specificity can have different depth, depending on the branch. Moreover, some GO terms appear on more than one branch. For this reason the parameter "spec" was also defined.

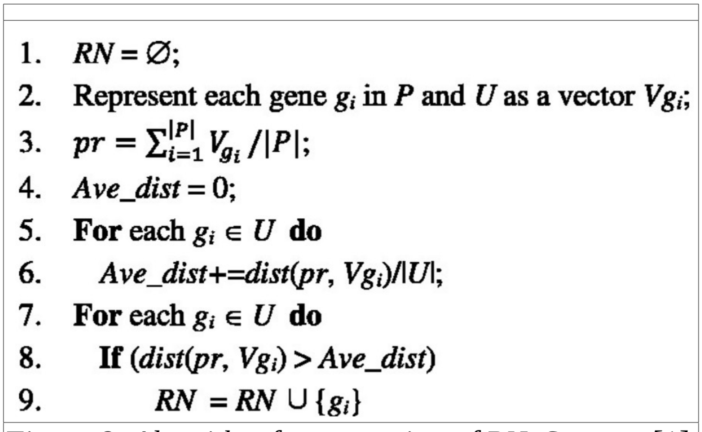
We computed the correlation between these GO-term properties in order to investigate how they may be affecting the prediction performance.

Part 4a- PU-BMRF development

Similarly to Part 1, in Part 2, the folds for the set of positives were created solely based on the validated associations, and the same holds for the unlabeled genes. 10fold -CV was also used, however, due to time constraints the analysis was performed with only 4 replicates (instead of 20 in the conventional approach of BMRF). Also, due to time constrains the analysis was carried only with chickens data. Moreover, for simplicity, in part 4 the non-validated positives associations were always excluded from the analysis. This can be done by simply setting the "only_EES" parameter to "True".

Steps 1 to 6 aim the computation 86 features including 70 non-GO-specific features and 16 GO-specific features These features were computed for each of the 1,714,512 total gene-associations combinations (138 GO-terms x 12,424 genes). Note that, for chickens, only 138 GO-terms passed the GO-size filter.

Based on these features, we computed the euclidean distance between each of the genes in the positive genes in the train set and unlabeled genes in the train set, and estimated the average euclidean distance between these two groups of genes was computed ("Ave_dist"). Then, each of the genes in the test set was classified as reliable negative (RN) or non reliable negative (nonRN) depending on whether its distance to the positive genes was larger or smaller than "Ave_dist", as explained in the algorithm introduced by Yang et al, (2012) [23] (Figure 8):



```
1. $RN = \varnothing$;
2. Represent each gene $g_i$ in $P$ and $U$ as a vector $Vg_i$;
3. $pr = \sum_{i=1}^{|P|} V_{g_i} / |P|$;
4. $Ave\_dist = 0$;
5. For each $g_i \in U$ do
6.     $Ave\_dist += dist(pr, Vg_i)/|U|$;
7. For each $g_i \in U$ do
8.     If $(dist(pr, Vg_i) > Ave\_dist)$
9.         $RN = RN \cup \{g_i\}$
```

Figure 8: Algorithm for extraction of RN. Source: [1]

Note that step 6 in the algorithm allows to introduce a constant to specify how strict we want to be in the extraction of RN genes. A constant of 1 means that the genes in the test whose distance to the positive set is more than “Ave_dist” will be classified as nonRN. However, a constant of 1.5 would mean that the algorithm is more strict in the process of extraction of RN and only genes that are 50% further the positive set than “Ave_dist” will be classified as RN.

The computation of the features that are used to calculate the euclidean distances involve 6 steps, as described below:

- **Step 1 – Similarity Matrix:**

A similarity matrix between the GO-terms was computed. The use of computing this matrix is two folded: First, it allows to extract a set of unrelated GO-terms from a bigger subset of GO-terms, in case it was not possible to carry the analysis for all the GO-terms that passed the GO-size filter, for instance, due to time constraints; and second, the matrix will be used in the computation of the features (step 5).

- **Step 2 – Defining the folds for the k-fold CV:**

The training and test-sets are created by randomly sampling genes among the positive associations for each GO-term. Then, for each fold, one GO_file was created, in which the associations in the test-set had been excluded. Also, the set of genes that, after “hidding” the test set, were associated with the GO-terms were stored, and also their neighbors. Thus, for each GO-term, two objects were stored: the set of positives 9gens associated with the GIO term of interest), and the neighbors of the positive genes. Then, another object was extracted with the neighbors of each gene. These objects are different for each fold and will be used in steps 4 and 5.

- **Step 3 – Network features:**

Transitivity, closeness and betweeness were computed for each gene-GO combination in three different networks:

- A network for all edges that are connected to at least one positive gene. This is expected to be useful for extracting RN because the positive genes are expected to be more interconnected in this network.
- A network of all edges that have at least one node in the set of Positives, and all the edges that have both their nodes in the set of neighbors of positives (step 2). In this network, it is also expected that the positive (either discovered or to be discovered) are more interconnected than the genes that are not associated with the GO-term (non-positive genes).
- A network of all edges except those that link two genes in the positive set. We expect that in this network, the genes that are positive are less interconnected.

- **Step 4 – non-GO-specific features:**

The following features are computed for each of the 12,424 genes:

- features f1-f4 refer to the number of GO-terms of the gene. It is expected that the genes that are associated with a large number of GO-terms are more likely to be associated with a novel GO-term. We expect this probability to be even higher for those genes that are associated with a large number of specific GO-terms before the GO-file up-propagating because this may imply that they are involved in unrelated functions and therefore they could potentially be regulatory genes. Genes that are associated with a large number of GO-terms only after up-propagating, however, may be associated with they are associated with a GO term whose branch in the GO-hierarchy is very large.

f1) The number of GO-terms the gene is associated with.
f2) The number of GO-terms the gene is associated with, including 'NONvalid' associations.
f3) The number of GO-terms the gene is associated with, in a GO file before up-propagating
f4) The number of GO-terms the gene is associated with, in a GO file after up-propagating

- features f5 and f6 refer to the number of GO-terms of the neighbors of the gene.

f5) The sum of the number GO-terms that are associated with the genes that are co-expressed with the gene to be annotated.
f6) The sum of the number of GO-terms that are associated with the genes that are co-expressed with the gene to be annotated in a database where NONvalid associations were also considered.

- features f7-f9 refer to the number of neighbor.

f7) The number of genes that are co-expressed with the gene of interest. BMRF accounts for this information, but only when the network data was extracted with a Pearson Correlation threshold of 0.35. In this step, we can provide additional information form other networks with other co-expression thresholds as well.
f8) The number of genes that are co-expressed with the gene of interest and are associated with at least 2 GO-terms.
f9) The number of genes that are co-expressed with the gene of interest and are associated with at least 5 GO-terms.

We expect that genes that are co-expressed with genes that have multiple functions are more likely to have multiple functions and therefore are more likely to be associated with novel GO-terms. The thresholds of 2 GO-terms and 5 GO-terms in f8 and f9 were chosen based on the variability within the data for that feature. We are interested in features that are highly, or at least moderately variable across the genes.

- features f10 to f70 are same as f1-f7 but for different features Pearson correlation thresholds. Mainly: 0.1, 0.2, 0.35, 0.5, 0.6, 0.7 and 0.8. Note that as the network database changes, so does the GO-file because BMRF does not allow for any gene that is not in the network. Due to the constrains in the data that come after the different correlation thresholds, it is expected that if a gene is associated with a very large number of GO-terms when the Pearson correlation was high (i.e. 0.6), it must be a gene involved in many different functions, whereas it may be that other genes are associated with more GO-terms when the Pearson correlation is lower, and this should also be considered.

The possibilities of restricting the data-set based on the Pearson correlation cutoff greatly increases the information available. This is because based on different correlation cutoffs we may be be able to observe different patterns in data. Furthermore, condition-independent networks, such as used in this study can benefit particularly from this approach.

- **Step 5 – GO-specific features:**

For each gene, we computed up to 16 GO-specific features. First, we defined four intersection for each gene and most of the features defined in step 5 will be computed for each of these intersection (for each gene).

Intersections of genes:

(1) Whether the gene of interest is in the set of positives
(2) Genes that are found in the neighbors of the gene of interest and in the set of positives
(3) Genes that are found in the gene of interest and the neighbors of the genes in the set of positives
(4) Genes that are found in the neighbors of the gene of interest and the neighbors of the genes in the set of positives.

F1-f4) Following from step 3, for each gene, we compute the sum of the betweeness, transitivity and closeness of the GO-terms that are in the interactions 1-4.

f5-f9) The number of genes in intersections 1-4 and the portions of neighbors of the genes of interest that are in the intersection s1-4.

F10-f13) The number of domains of the genes in intersections 1-4 the number of genes that share domains in intersections 1-4, and the number or unique domains that are sheared between genes in intersections 1-4.

f14-f16) The number of genes in interaction 1-2 (neighbors of the genes in the positive set are not considered here) weighted by the degree of similarity between the GO-terms they have in common and the GO-term we are interested in.

Note that these fatures vary depending on the train and test set, and therefor they need to be computed k-tie4ms per replicate.

- **Step 6 – extraction of RN:**

The databases with features information obtained in steps 4 and 5 were combined and the values of the features were scaled by dividing each value by the square root of the squared sum of all the values. Thus, for each GO-term we have a database with 86 features per gene. Checking the label of the genes that are in the training set, we apply the algorithm in illustration XX:

We check whether any of the RN is in the set of non-validated positive cases and we removed from the analysis those that did so. We tried different thresholds in step 6 of the algorithm in Illustration 1 (default value is 1) and through an iterative process we adjusted the threshold to the highest value that allows to extract a maximum number of RN. We gradually increased the threshold by 0.05 if the criteria was not satisfied (thus, if the number of RN was excessively high for our purpose). We proceeded the analysis for different values of “maximum number of RN”, mainly, for 1000, 2000… and 8000.

In step 6, we also extracted the same amount of RN but through random extraction and stored them separately.
Two approaches were also used to extracted RN. Instead of defining a fixed number of RN, we allowed the threshold to change according to a desired value of AUC in the process of extraction of RN. For instance, we can specify that we want to extract as many RN as possible as long as the AUC is equal to 1 in the process of extraction. The genes that are used to evaluate the performance of extraction are those in the test-set. These genes were excluded from the analysis in step 1 of the PU-BMRF and therefore were not considered to define the threshold in step 6 of the algorithm in illustration 1.

Finally, in order to be more reliable about the RN not being positives, wee excluded from the set of RN those genes that while being in the set of RN they were and also in the set of positive-NONvalid.

Note that a different set of RN will be extracted per fold, per replicate and per GP term.

After the extraction of RN we train the BMRF classifier (RN vs positives), instead of (unlabeled vs positives) in the conventional BMRF approach.

Part 4b- PU-BMRF performance evaluation

- Evaluation of the process of extraction of RN

We computed the accuracy of extractyion of the RN. Note that, in oprinciple, only unlabelled genes should be classified as RN. For AUC in BMRF, the expected label for the positives-train is 1, and the unlabeled-test as 0. And the predicted label was '0' if the gene was classified as RN, and '1', otehrwosie. This means that a gene whose expected label is '1' and its predicted label is a '0' is a false positive, because it is a gene that is positive and therefore it should not have been classified as RN. Similarly, a gene whose expected label is 0 and whose predicted labels is 1, may hve been prperly classified (it is and unkwon gene that was not extracted as RN); a gene whose expected label is '1' and whsoe predicted labels is '1', has been surelly properly classified because it was a positive gene and it has not been classified as RN; and lastly, a gene whose expected label is '0' and prediction label is '0', has been properly classified as well, because it is an unlablled gene that was classified as RN.

Since the fouc was on the accuracy of extracion of RN we extracted from the computation of AUC those genes fiow which wee expect a '0; and we observed a '1. So the unlabeld genes that were not classdifed as RN. Note that if these were included in the compuattion of AUC, AUC could be inflated based on this very large subset of genes when in fact we cannot tell if they were properly classified. This is an important aspect to ciobnsider, especially since we want to caompare the accuracy of prediction for differnet number of RN and this class of genes may beconme very large when the number of RN is very low (for instance setting a maximum of 1000).

- Reproducibility of the process of extraction of RN

In order to asses the eproducibility of the extraction of RN across replicates, we investigated which portionof the RN exctraed were cmmon in all the replciateds of the analysis. Furrther, we assesed the reproducibility across folds within the same replicate. For tis, we calculated how many differnet RN were extracted per repllicate (combining the extraction of each fold).

- Prediction perfoamnce using the RN

The positive genes and the genes in the set of RN were used to the train the BMRF classifier. It ios expected that the classification will be more accurate now that the set of unlabellled gens have been replaced by a smaller set of RN. Apart from that, the RN were terated in a simlar way than the unlabelled gen sin the conventional approach of BMRF.

Part5-Biolgical support of the approach

In principle, PU-learning should contribute to improve the epp/tpepp by removing epn edges from the data. From a biological perspective, we would expect that the genes that are involved in more specific functions have a higher epp/tpepp. The aim of this part is to investigate whether that principle holds in the data, whether the ratio improves more in genes involved in specific function or in general functions, and finally how this related to the increase in accuracy when PU-BMRF is used instead of BMRF, in both types of genes.

In order to investigate whether the principle of more specific-better connected holds in the data, we followed two approaches. First, we computed the correlation between specificity and epp/tpepp, and second, we compared epp./tpepp in two groups of genes depending on whether these genes are associated with specific or general GO terms.

To compare epp/tpepp in these two groups, we looked into genes that are exclusive form general or specific Go terms. For this, we subseted GO terms with a specificity larger than 5 and less than 3000 associated genes. The first requirement is to make sure that the GO terms considered have at least a minimum of edges. Otherwise, we

GO terms will be considered for which almost none of the associated genes are coexpressed. The second requirement is to guarantee that there are some gene in the specific GO term that are not in the general GO terms. For instance, if we choose the GO-term 'Biological process', which is associated with XXXX genes, it would be very difficult to find genes that are associated with a more specific GO term but not with 'Biological process'.

Then we distinguish between two groups of GO terms: one specific and one of general GO terms. We decided that the specific GO terms should have a maximum of 7 genes whereas the general should have 2700. The choice of these values is trivial because we want a large difference between the two group, at the same time that enough number of representatives in both groups. Furthermore, if the groups are two large, then it will be very difficult to find genes that are exclusive only of one of the groups. Then we sampled 200 genes that are exclusive from each group and we compared the number of edges that connect the genes of each of these sets.

Results

The approach aimed rthe predicon of the BP ctegory because this category is more difficult to predict weith the ocnventional homologouse-based PFP methods

Part 1- Choice of the network data for chickens and tune of the model parameters.

The preeiction performance was higher for chcieksn when a Pearson correlation of 0.35 was used as a co-expression thershodl, as it can be obbserve din Illustration 4. We therfore carried the next analysis on chickens with this co-expression cutoff.

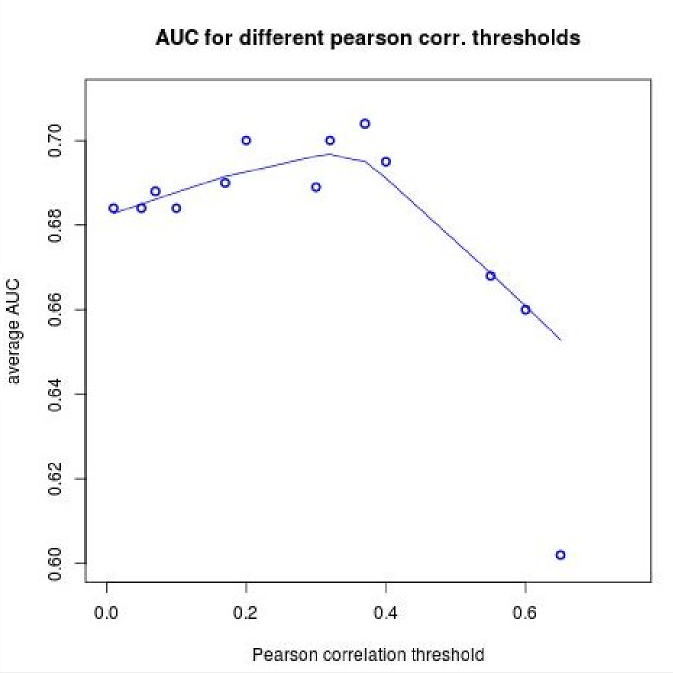


Illustration 1: AUC for different Pearson correlation thresholds for the chickens' co-expression network.

The Y-axis corresponds to the average AUC for all GO-terms considered in the analysis. The line corresponds to the lowness line. Values corresponding to the plot are in table 1.

After tuning the model parameters we come with the following conclusions about how to carry the next analyses:

- 10-fold CV will be used, because the accuracy of prediction is significantly larger than for 5-fold CV.
- Each analysis will be carried with 20 replicates, because the sd across replicates is low (<0.02) on human data.
-The number of iterations for the Gibbs sampling in BMRF will be set to 30, since this number is enough even when the number of unlabeled genes is large
- A GO-size filter of (20 , 0.1), for minGOsize and maxGOsize, respectively. This filter

specifies that no predictions will be made for the most general GO terms (since they are less relevant), and it guarantees that there will be enough number of positive cases in each fold when BMRF is used.
- We will use domain information, as well as non-validated associations because they seem to improve the performance.

Table 2: Concussions from tuning the model parameters.

Part 2- Differenes in prediction performance based on data available (using BMRF)

The conclusions form this part are:

- Although overall predictions are better for chickens when we look at specific GO terms, predictions are better for humans than for chickens, which highlights the difficulty of doing PFP in poorly annotated species

- A lower quality of the data (less annotations and/or less reliable edges) is translated into a lower reproducibility in the prediction performance
- The size of the network does not seem to have a strong effect on the prediction performance
- Reducing enn and epn greatly improved the prediction performance .

- There was no difference between using the co-expression data from one tissue or another

Section 2.a) Impact data difference between the species on the prediction performance

Table 5 shows the prediction performance for the species considered using BMRF.

	yeast	yeast_ppi	humans	Chicken
# GO terms	1,102	1,019	1,982	138
mean AUC (sd AUC)	0.764 (0.081)	0.714(0.09)	0.7 (0.077)	0.726 (0.08)
median AUC	0.762	0.711	0.701	0.721
mean sd across replicates	0.016	0.02	0.017	0.03

Table 3: Overall prediction performance for the different species using BMRF

Predictions were more accurate for yeast, then for chickens, yeast_ppi and finally, for humans.
The fact that we achieve better prediction performance for chickens than for yeast_ppi, is most probably linked to the fact that a much lower number of GO terms are predicted when the chicken data was used, than in the other cases. It may be the case that the GO terms that were predicted with chickens data are more easy to predict than the overall set of GO terms that are predicted in yeast, yeast_ppi and humans.

Therefore, we aimed to compare the prediction performance of the GO terms that were predicted in the 4 cases. We observed that of the 138 GO terms predicted with chicken data, 20, were also predicted in the other 3 cases. Illustration 7 shows, a comparison of the prediction performance using data from the different species:

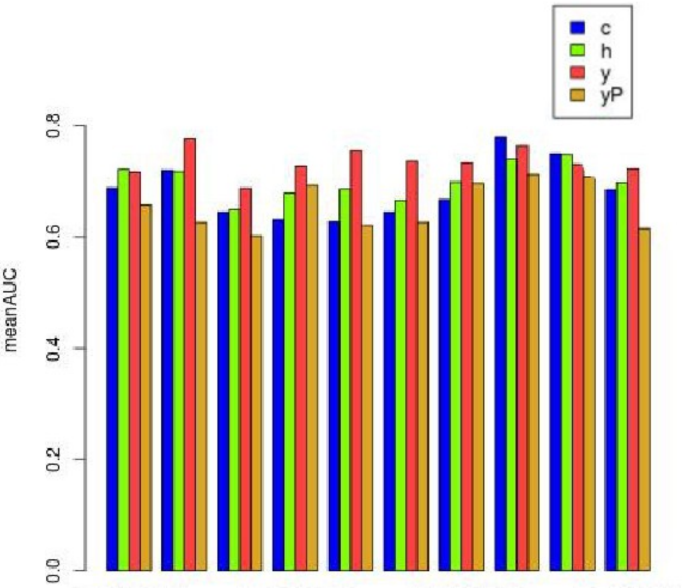


Illustration 2: GO_0009605,GO_0009653,GO_0009719,GO_0009892,GO_0009966,GO_0010033,GO_0010605,GO_0010628,GO_0010629,GO_0030154

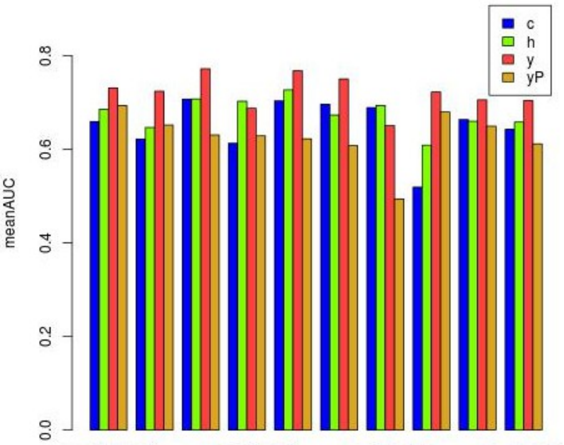


Illustration 3: GO_0031324,GO_0032879,GO_0048468, GO_0048584,GO_0048646,GO_0048869, GO_0050793,GO_0051128,GO_0070887, GO_1901700

Illustration 7 revealed that the prediction performance when we look at the same GO term in the 4 species is overall larger for yeast, then for humans, the chickens and finally yeast_ppi. The mean AUC (and sd) were, 0.729(0.03), 0.688(0.03), 0.668(0.05) and 0.642(0.05), respectively for the 4 species.

The fact that predictions are still high for chickens even though a Pearson correlation of 0.35 was used instead of 0.7 in the other 3 cases, suggests that BMRF has better prediction performance when the network has more edges even at the expenses of a lower reliability of the edges. Furthermore we can conclude that BMRF had a high prediction performance on chickens, which, to our knowl3dge is the most poorly annotated species ever considered for PFP via networks.

Section 2.B) Impact of the quality of the data-set

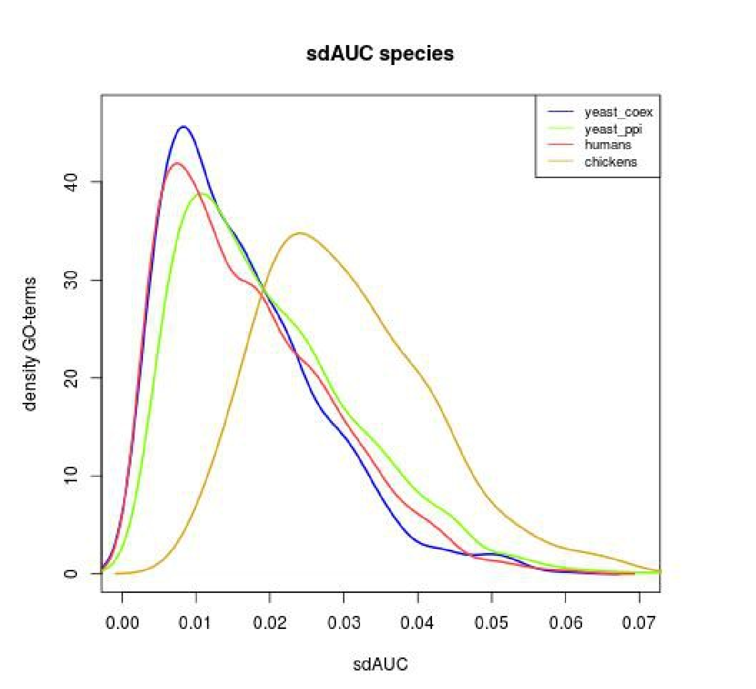


Illustration 4: Distribution of the standard deviation of AUC across replicates for the different GO-terms in the different species.

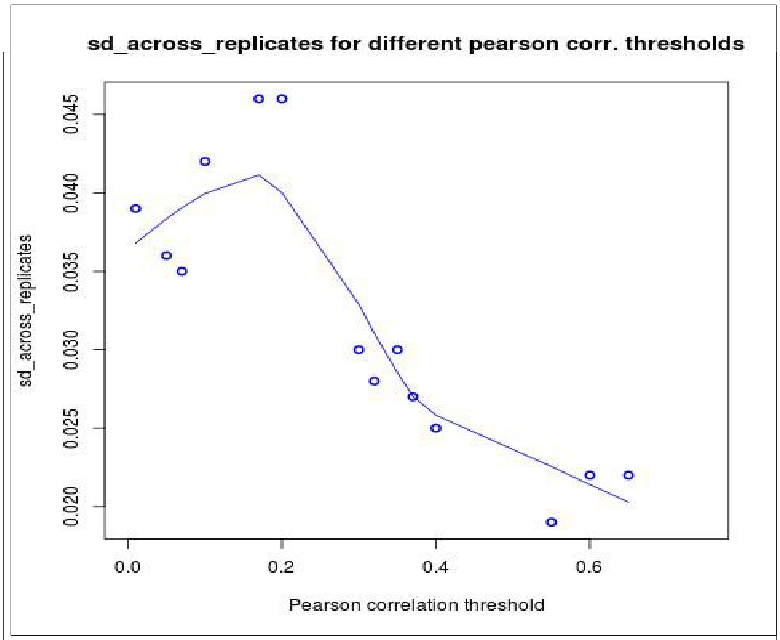


Illustration 5: Standard deviation across replicates for different co-expression thresholds.

Values corresponding to the plot are in table 1.

In illustration 3 we observe that the standard deviation across replicated decreases also as we become more strict in the co-expression threshold. Thus, choosing a high co-expression threshold allows for better reproducibility. On the left hand side of the illustration, we observe that the standard deviation was lower for Pearson correlation thresholds (0.01, 0.02 and 0.015), than for a Person correlation of 0.17 and 0.2. This effect is counter-intuitive, but does not seem to be significant and may be an artifact of the analysis. Values for this plot are in Table XX of Appendix III-Additional results

We randomly extracted from the network a known percentage of the edges and calculated the prediction performance. Table X shows the results of this analysis.

Portion of edges extracted from data	Mean AUC
0% (all network data used)	0.744
10%	0.738
30%	0.733
50%	0.738
90%	0.719
95%	0.719

Table 6: Impact of number of edges in the prediction performance, using yeast data.

From table X, we observed that **r**emoving random edges from the data did not seem to affect much the prediction performance. We observed a larger impact after removing 10% of the edges and after removing more than 50% of the edges. Also, this decrease in the AUC can be caused by the fact that less GO terms were considered in the anlaysis. This is because in BMRF the GO-term annotation is pruned based on the network.

	Correlation
AUC_reduceEpn	0.98
AUC_reduceEnn	0.95
AUC_reduceAmg	0.67
AUC_reduceOa	-0.47
AUC_reduceEpp	-0.26

Table 4: Correlation between AUC and data quality, for human data

Section 2.C) Impact of the characteristics of the expression analysis

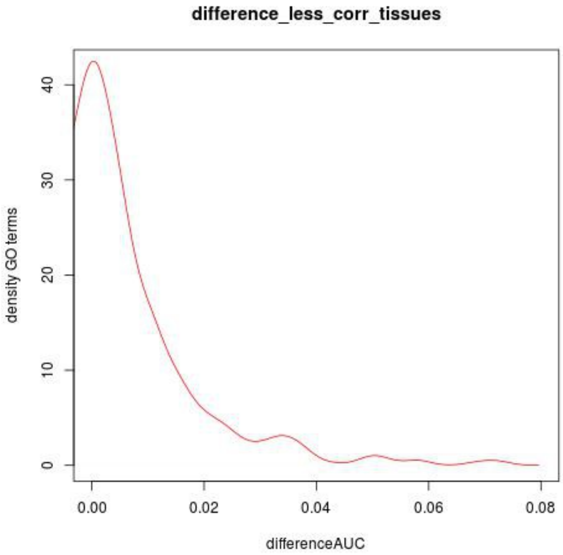


Illustration 7: Difference in AUC for the different GO terms

There was no difference between using the co-expression data from one tissue or another

Part3) Differences in prediction performance between GO-terms (using BMRF)

With the purpose of identifying which GO-term-properties have a more direct effect on the predictions performance, we computed the correlation between AUC and the different GO-term properties, Th results are shown in Table 4.

	yeast	yeast_PPI	humans	chickens
epp_V/tpEppV	0.62	0.373	0.462	.
epp/tpEpp	0.582	0.34	0.378	.
sdAUC	-0.408	.	-0.337	.
teV/tpeV	0.394	0.241	-0.13	-0.263
depth	0.265	0.104	.	0.478
spec	0.095	.	.	0.518
#genesV	.	.	.	-0.518

Table 5: Correlations with AUC_comparison species

Only significant correlation s (pvalue<0.05) were added to the table.

From table 4 we learn that epp/tpepp and sd are the parameters that affect more the prediction performance. Epp/tpepp shows a favorable correlation with AUC, meaning that for GO terms whose associated genes are interconnected in the network (coexpressed), the method has more chances to

distinguish genes that are associated associated with the GO term from genes that are not.
In order to achieve high PFP accuracy, epp/tpepp should be as large as possible. Epp depends on data available and cannot be increased with methods, however tpepp could be reduced by PU-learning. Note that, from theory, PU-learning improves two ratios: (1) it reduces the portion of epn within enn, and (2) it reduces the epn, as some unknown genes are droped from the anlaysis. By improving the seond ratio, PU-learning would be increasing the epp/tpepp ratio.

Sd shows a negative correlation with AUC, meaning that for those GO terms whose AUC fluctuates more from replicate to replicate are overall worse predicted. A possible explanation for this is that the sd is high when epp/tpepp is low. Thus, indirectly, high sd means low overall AUC. This is because if only a few of the associated genes are interconnected with each other, the results will depend on whether these interconnected lebeled genes enter the training or the test set in the cross-validation.

In Material and methods it was explained that depth is not a reliable estimate of the specificity of the GO terms. Thus, its is not very surprising that we we observed positive correlations between AUC and depth.

Illustration X shows a graphical rep[resnetation of how these correlation change across the species considred.

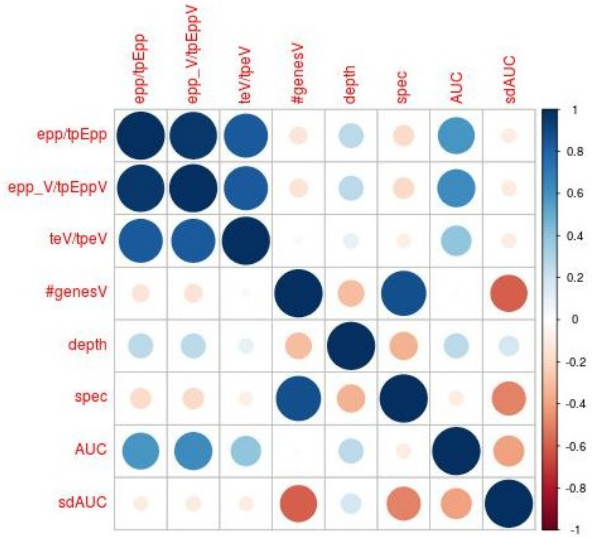
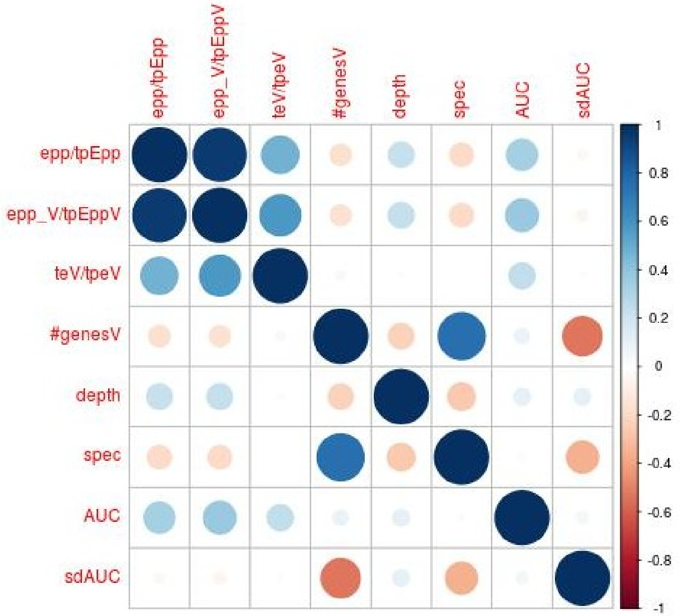


Illustration 9: Correlations yeast



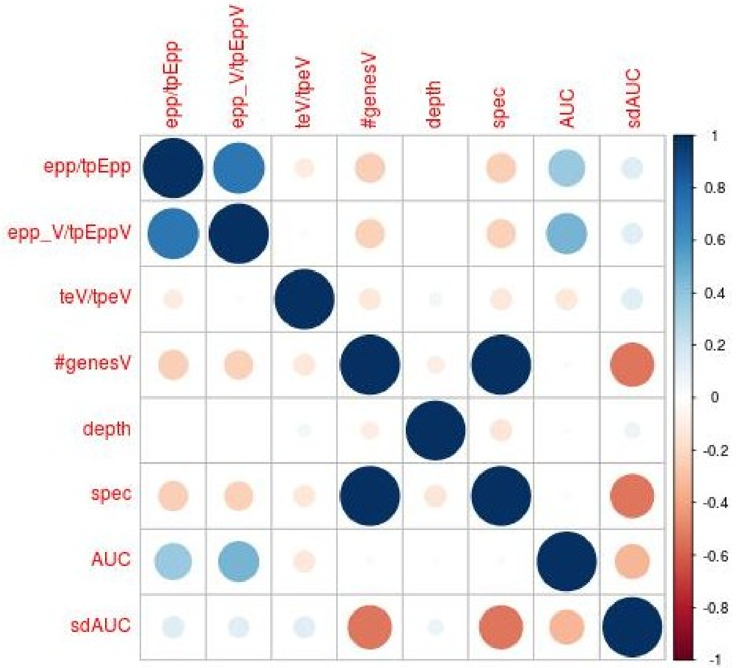


Illustration 11: Correlations humans

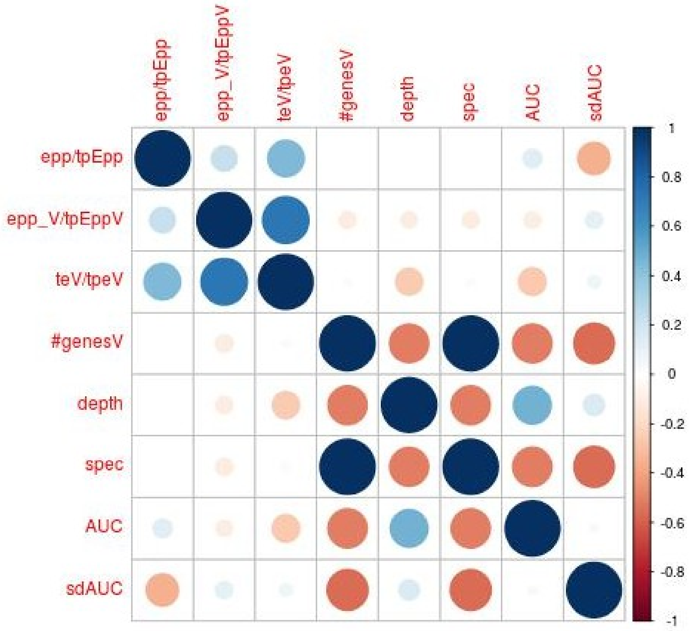


Illustration 10: Correlations chickens

The anlaysis with chickens data showed a diffrenet patern of correlations. The strong corrrelation betwwen specificty and AUC is surprisng, as well as the correlation between genesV and AUC (-0.52).

ompare the ratio epp/tpepp at the level of individual GO terms. (Illustration X)for 30 randomly chose GO terms that are predicted in the four cases

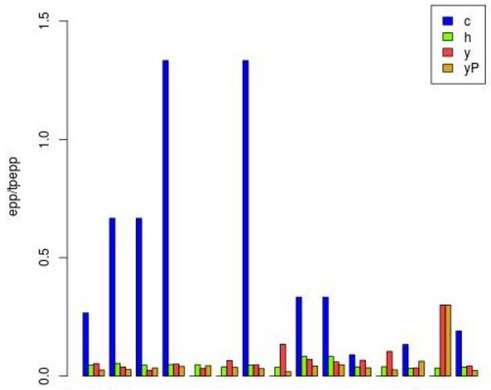


Illustration 13: GO_0000003,GO_0000075,GO_0000077,GO_0000086,GO_0000122,GO_0000165,GO_0000278,GO_0000302,GO_0000724,GO_0000725,GO_0000902,GO_0001101,GO_0001558,GO_0001676,GO_0001932

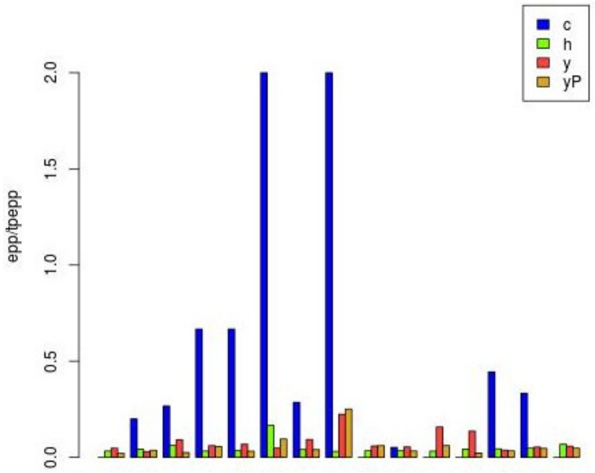


Illustration 12: GO_0001933,GO_0001934,GO_0003006,GO_0005975,GO_0006066,GO_0006071,GO_0006082,GO_0006090,GO_0006109,GO_0006139,GO_0006163,GO_0006164,GO_0006259,GO_0006260,GO_0006261

The ratio is considerably larger for chickens, although this could be an artifact of the scarce data that is available, as explained in Table XX of Appendix_II-Data_overview. After that, the GO terms seem to have a higher ratio for yeast than for the other species, which is i

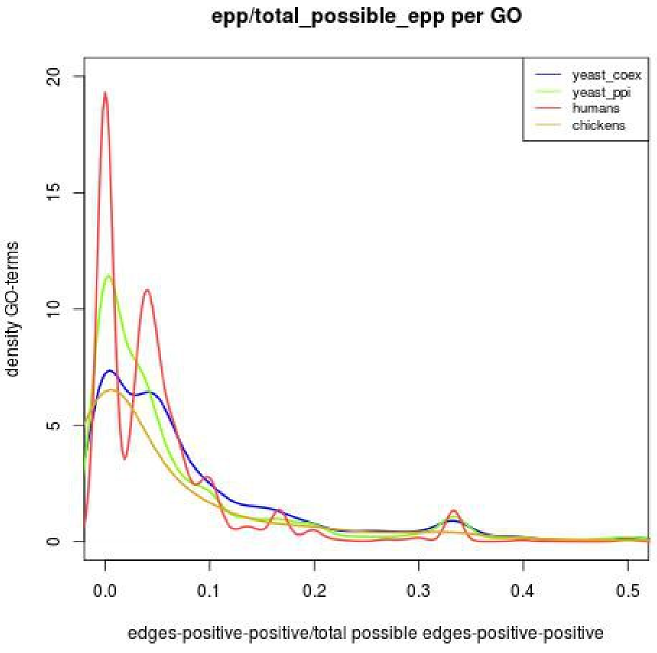


Illustration 14: epp/tpepp. Value sin table x appendix I

	Average epp/tpepp (sd)
yeast	0.122 (0.204)
yeast_PPI	0.106 (0.204)
humans	0.066 (0.138)
chickens	0.255 (0.55)

Table 6: average epp/tpepp per GO-term for the different species

epp/tpepp seem to be the network parameter with a larger impact in the prediction performance.

In part 2 we have seen that prediction sin indiviudal GO terms were higher when human data was used than when chiocken data was used, because human data is

we conlsude that a method the predcition perfomance with BMRF could be iomproved with PU-learning because it will allow to improv ethe ratioepp/tpepp. More

Part 4- PU-BMRF performance evaluation.

Table 5 illustrates the accuracy of prediction sin the two steps of PU-BMRF, first the RN are extracted from the set of unlabeled genes, and second BMRF is trained on the set of positives and the set of RN.

	Max # of RN extracted							
	1000	**2000**	**3000**	**4000**	**5000**	**6000**	**7000**	**8000**
Accuracy of extraction of RN	0.981 (0.01)	0.973 (0.015)	0.966 (0.019)	0.965 (0.02)	0.966 (0.021)	0.961 (0.023)	0.959 (0.023)	0.958 (0.024)
PFP AUC using PU-BMRF (sd)	0.723 (0.08)	0.75 (0.072)	0.758 (0.084)	0.751 (0.086)	0.725 (0.094)	0.728 (0.089)	0.716 (0.092)	0.701 (0.095)

Table 7: Accuracy in the two steps of PU-BMRF, for different choices of max #RN extracted. Results from 30 GO-terms

The prediction accuracy for these GO terms using BMRF was 0.706 (0.026). Thus, except when a maximum of 8000 RN were extracted, PU-BMRF outperformed BMRF. The maximum improvement (+0.052 AUC) was when a maximum of RN was set to 3000. This is also illustrated in Illustration 7.

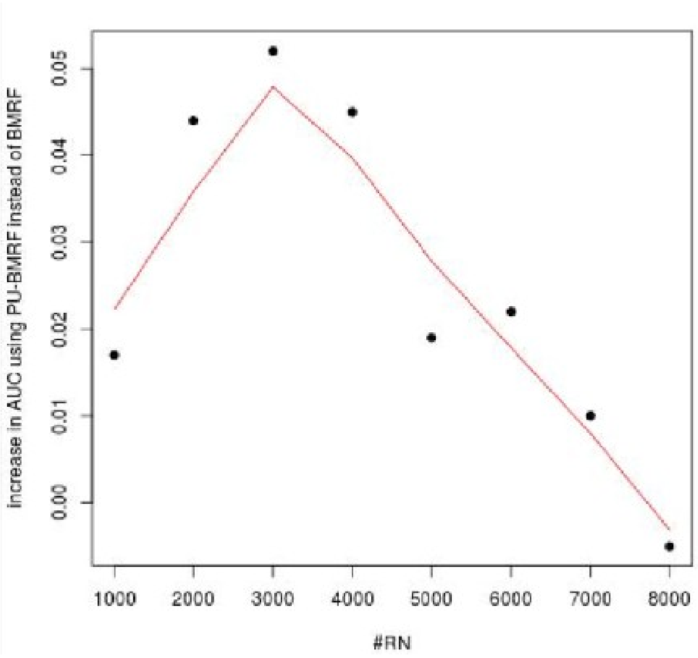


Illustration 15: Increase in AUC PU-BMRF vs BMRF, for different choices of max #RN extracted

A disappointing results was the increase in standard deviation across replicates (0.03 with BMRF vs 0.076 with PU-BMRF). This increase is surprising considering that, as shown below, the reproducibility in the process of extraction of RN was high. The most likely explanation for this is that the network used was smaller in the case of PU-BMRF. In fact, it was shown that the sd was even higher when a set of 3000 RN was extracted randomly (sd was 0.1).

The average AUC in the process of extraction of 3000 RN for the 30 GO-terms was 0.966 (0.019), and the average sd across folds was (0.0434). However, in order to test the reproducibility in the process of extraction of RN, we calculated which portion of the RN were extracted in the 4 replicates. We carried the analysis in the situation where a maximum of 3000 were chosen. We observed than one average 96.7% (0.0159) of the RN were extracted in the 4 replicates. Results for 30 GO terms are given in table XX in Appendix III-Additional_results and in illustration XX, part 3. Furthermore, we observed that On average, 3033.583 (104.67) different RNN were extracted in the 1- folds of each replicate, which is very close to the 3000 minimum RN per fold, indicating that the large majority of the RN were common in the 10 folds.

Another important result is that, as expected, the accuracy of prediction was lower when the RN were extracted randomly than when BMRF, or PU-BMRF was used, as it is shown illustration 9

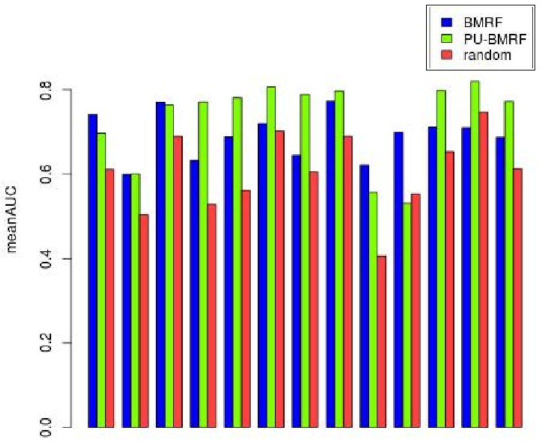


Illustration 16: GO_0006928,GO_0006950,GO_0007423, GO_0008283,GO_0009605,GO_0009653, GO_0009719,GO_0009887,GO_0032879, GO_0033554,GO_0040011,GO_0044699, GO_0044700

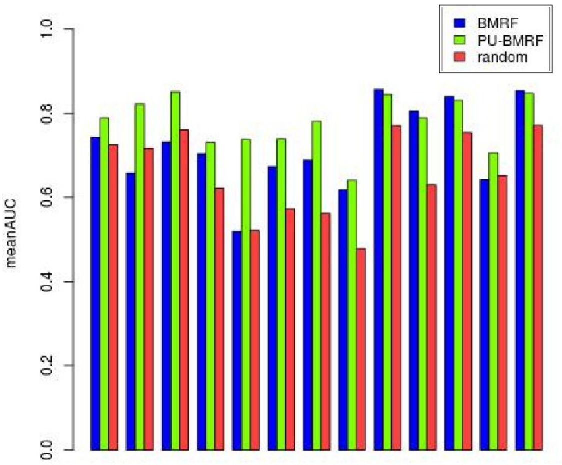
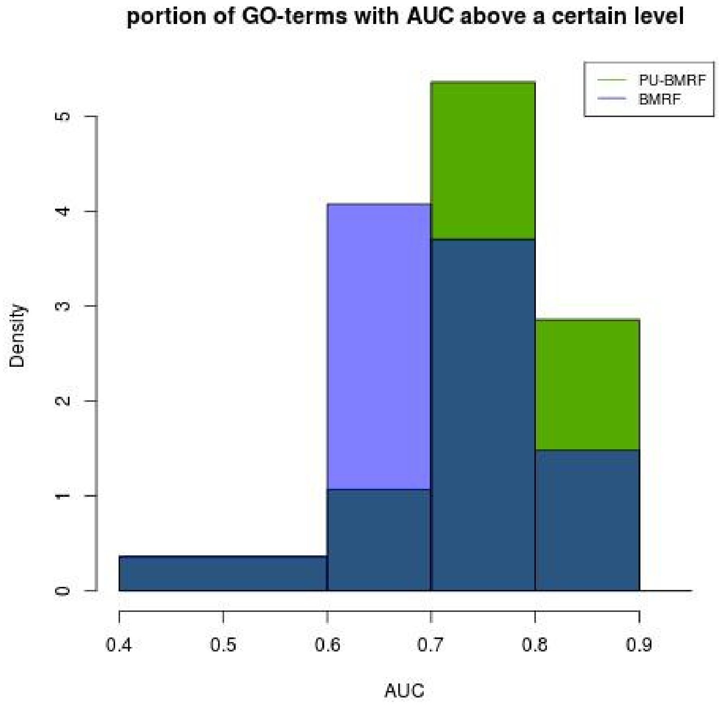


Illustration 17: GO_0044710,GO_0044763,GO_0044767, GO_0048646,GO_0051128,GO_0051240, GO_0060485,GO_0065008,GO_0097659, GO_1901360,GO_1901362,GO_1901700, GO_1903506

Illustration XX shows to which extend PU-BMRF changed the distribution of AUC across the 30 GO terms used. It is shown that the portion of GO terms for which AUC higher than 70 or 80% increases, whereas there are less GO terms with an AUC of 60-70% when PU-BMRF was used. Further, even when PU-BMRF was used, the same portion of GO terms with AUC<60% reminded.



We then investigated whether the increase in AUC with PU-BMRRF vs BMRF was correlated with any of the GO-term properties seen in part 2. We observed that non of the GO-term properties was significantly correlated with the increase in AUC. However, the strongest correlation was with epp/tpepp (0.25), with a p-value of 0.21. We would expect that this p-value will be lower if more than 30 GO terms were used to compute the correlation.

Part 5- Biological support of the approach

Questions to be addressed are:

1.whether the more specific GO terms have a higher degree of connection in the network.

2.whether this is translated into a better accuracy of PFP with BMRF.

3.whether genes that are associated with a large number of GO terms of different levels of specificity are better predicted with BMRF.

We are basically interested in the relatiosnhip between 4 variables (it would be good to provide the partial corrs as well as the corrs:

AUC
increase in AUC (PU-BMRF vs BMRF)
specificity
degree of connection

However, there are a few ways to compute degree of connection. And AUC should concern individual genes rather than GO-terms

1. The answer is “yes, to some extend”. It would be good to do other experimnet and test for siginifcance. As it can be seen below I tried to link to PU.

2. 3. I will try to do before saturday. It requires changing the code from “predictions accuracy per GO-term” to “prediction accuracy per gene”

Another thin that I want to is to provide some information about the GO terms that are predicted with PU-BMRF, and contrast with literature

In part 3 we observe that PU contributes more in GO terms whose associated genes show a high degree of connection (high epp/tpepp). Moreover, in part 2 we observed that and epp/tpepp shows a negatoive corr with 'specificity' (-0.23 in humans). Thus, PU may contribute more in specific GO terms. Furthermore, since PU contributes more when the ratio epp/tpepp, we would expect that PU contributes more in species for which the overall ratio epp/tpepp is high. From table XX in Appendix II, we learnt that the overall ratio epp/tpepp was higher for chickens than for the other more annotated species. Thus, it would be interesting to check whether the ratio epp/tpepp is generally larger for poorly annotated species and if PU contributes more to increase the prediction performance in poorly annotated species than in other more annotated species. Table XX shows the correlation between 'specificity' and epp/tpepp.

Genes that are exclusive	corr(spec-epp/tpepp) (p_value)
yeast	0.192 (0)
yeast_PPI	0.199 (0)
humans	0.235**(0)**
chickens	0.108 (0.21)

In table XX we observe that the correlation between 'specificity' and epp/tpepp ranges between -0.1 and -0.24 depending on the species, and is lower for poorly annotated species. In addition, further analysis revealed that this correlation changes depending on whether we include in the computation of the correlation Go-trems with a minimum number of associated genes Table XX). This can be due to the fact that epp/tpepp tends to be larger when more genes are considered. The correlation reaches its highest value when only Go terms with a minim of 2000 associated genes. In other words, we observed that within a group of very general GO terms, the principle that “genes of specific GO terms are highly connected” holds to a larger extend. This makes, sense since when the number of positive genes is very large it is more difficult that they all are co-expressed at the same time. Whereas, when the number of genes is low, the genes within the group have higher chances to be co-expressed with other genes that are not in the group. More so I f we take into account that some of the genes that while being outside the group of positives, are co-expressed with positive genes, are actually positive genes not yest discovered. And we would expect that the chances of making false positives decreases as we look into more general GO terms. Table XX shows the effect of minGOsize on the correlation between generality and epp/tpepp.

Genes that are exclusive	cor(corr(spec-epp/tpepp) – minGOsize)
yeast	0.1503 (0.006)
yeast_PPI	0.87 (0)
humans	0.901 (0)
chickens	0.402 (0.06)

In order to correct for the size of the group off genes, we carried another experiment. We compared the degree of connection between a group of gens that are only involved in general GO term, and a group of genes that are only involved in general GO terms. We homogenized the size of the two groups and we observed that in out of 10,000 replicates of genes sampling for the two groups, in 8394 replicates the genes in the group of specific GO terms appeared to be more interconnected than the genes in the set of general GO terms. On average, these genes appeared to be 6.36% more highly connected than the genes exclusive of the general GO terms.

Discussion

Development of computational methods for PFP based on network data is a challenging problem in poorly annotated species. Here, we expand upon an existing BMRF and develop a PU implementation (PU-BMRF) that is more accurate than predecessor for PFP in poorly annotated species such as chicken. The efficiency of BMRF to infer function of proteins in poorly annotated had been previously noted [5,6]. Nevertheless, the efficiency of BMRF is hampered because the algorithm attempts to solve a two class classification problem when in fact the annotated data is from one single class (positive class). PU-BMRF tackles this problem by adding a previous step to the BMRF algorithm, in which a set of reliable negative are extracted. Subsequently, the BMRF classifier can be trained with a representative set of genes of each class (negative and positives) and prediction become more accurate.

The basis of PU is in extracting the genes within the set of unlabeled, that show strong differences with respect to the set of positives. It is a must that the features that are used to investigate these differences are as unrelated as possible to the features that the classifier, BMRF in this case, uses afterward. BMRF uses neighborhood information but neglects other important features like, for instance, whether the gene is associated with a related GO term, or the degree of connection between the neighbors of the gene and the genes that are associated with the GO-term of interest. We developed a PU implementation in which 64 features were taken into account to identify a set of reliable negatives (RN) for each GO term.

Lack of reprodubility

Overall, 76% of the GO-terms were more accurately predicted when PU-BMRF was used with respect to when only BMRF was used. PU-BMRF, however, have some limitations. We observed that the extraction of RN has low reproducibility. To our knowledge this lack of consistency in the extraction of RN can be explained by the low number of folds that were used in the cross validation (only two folds). In fact, probably the main drawback of PU-BMRF is that the computational time is around 200 time larger than for BMRF and this may force the user to choose a low value of k-folds in the validation. The increase in computational time was, to some extend, expected given that PU-BMRF requires the computation for 64 features for each gene within each GO term; and some of which are complex like, for instance, the sum of the closeness of a set of gene in the network. As a consequence, one may have to choose a low value of k, at the expenses of a lack of reproducibility in the extraction of RN.

The lack of reproducibility, had been previously noted in PU methods. [13], for instance extracted 20,099 and 4066 RN with two different approaches, respectively and only 589 of the RN were common between the two approaches. [7] explained that in some cases the information from GO alone is not enough to predict a good set of negative examples, because some proteins defy the conventional annotation patterns.

In line with this, it should be considered that the main goal of PU applied to PFP is to eventually predict the function of genes that do not fall in the set of positives or reliable negatives. These predictions, however, cannot be evaluated with the same method that was used to do the predictions because in most PU approaches, the genes that do not fall in the set of positives or reliable negatives are extracted form the analysis. It could be, thus, argued that the lack of reproducibility is advantageous in that the predictions can be evaluated for a larger number of genes. Since in PU-BMRF, the set of RN that are extracted differs slightly from run to run, we would expect that given a sufficient number of runs, most of the genes will be included in the set of RN at least once. Thus, due to the low reproducibility, we may be able to evaluate the predictions for nearly every gene in the database.

Both, the reproducibility and the number of genes for which the predictions are made, can to some extend, be regulated in PU-BMRF by specifying the number of RN that we want to extract in each run. The user can choose to extract a very large number of RN, although it would come at the cost of a lower increase in accuracy. In [1], for instance, they set a value of 1 as cutoff in equation XX that regulates how many RN will be extracted, whereas in [2] they chose 1.05. In our case, we changed this value according to the GO term as we aimed to extract a fixed number of RN. We decided that this was a reasonable option since we observed that the accuracy did not increase when the number of RN was very large or very small. Another common threshold to separate RN from the unlabeled data is that specificity is equal to one. Thus, the cutoff in equation XX or in any other equation that aims to separate RN form the unlabeled data, could be be defined in such a way that non of the known positives would fall in the set of RN.

Factors that determine the prediction perfromance

Overall, since PU approaches consist on adding previous step to the classification algorithms, one may say that several PU approaches can be considered (either combining or applying one after another), in order to extract a more reliable set of RN. [2], for instance, extracted a set of likely negatives (LN) based on the approached introduced by [1] and then they extracted a set of RN using the approach developed in [3]. [13], for instance extracted RN based in two sets of features, one that consider each features individually and one that considered only the mean of the features of each group of features, and then extracted those RN that were extracted with both approaches. An important aspect to consider is that the quality of the method will not depend so much on the number of features that are defined or on how many PU approaches are integrated, but rather on two main factors: (1) How different the Positives are from the negatives for the features considered (data properties and quality of the features) and (2) Which portion of the unlabeled genes will be extracted as RN. Regarding to the fist factor, [7] referred to the so-called "moonlighting" proteins to those proteins that having a unique combination of features cannot be predicted from the conventional annotations.

In addition to this, there is another important factor that determined not only the accuracy with which the RN are extracted, but also the accuracy with which the final predictions are made via the final classifier. This factor is the extend to which the principle of gilt by association holds. It was previously shown that this "guilt by association" heuristic is universal and preserved beyond organism boundaries [12, 13]. However, the same greatly depends on the quality of the data. In particular, in the case of the co-expression networks, it should be considered that even if the quality of the data is good, the phenotypic variation is controlled at many levels, some of which are independent of transcript abundance.

AUC in one class classification problems

One aspect to be considered is that, has noted by [13], the AUC may not be the best way to estimate the prediction performance situations where only examples from one class are kown, because a gene in the test set whose 'hidden' label is a '0', may have been predicted as positive be actually positive. Thus, we would falsely claim a false positive when in fact is a true positive. [13] proposed that recall and specificity are better accuracy measures in these situations. Notwithstanding, in this thesis we computed AUC because it is not expected that the number of 'claimed false positives' will be high given that for the majority of the GO-terms the portion of genes that are expected to be associated is very low. Furthermore, using AUC enables to do better comparison with other methods, since AUC is the most commonly choice to express the accuracy of prediction.

Avenues of improvemnet

The current method can integrate protein-protein-interaction (ppi) data with the co-expression data, as well as data from some well-annotated related species, like, *Mus musculus* or *Homo sapiens* in the case of chicken, and this could lead to and improvement ion accuracy of prediction as explained in [5,6]. The main aspect to be improved regarding PFP in poorly annotated species like chicken, however, is not the accuracy of prediction but the number of GO-terms for which the predictions can be made. This number is very low even when the GO-size filters were at its minimum (minGOsize:9, 321 GO-terms). This problem, in-fact, may be even more sever in PU-BMRF than in BMRF, because since the computational time is much larger for PU-BMRF, the value of k may be lower, and subsequently a larger number of positives cases is required.

The number of GO terms for which make predictions can be made, could be improved, for instance, by expanding upon the existing R function "glmnet" that BMRF uses to train the classifier, in such a ways that the matrix are less sparse or that sparser matrix can be solved. In fact, probably the best avenue of improvement for PU-BMRF is be to extend the "glmnet" function to more than two classes. This would allow to do predictions within the set of unlabeled genes, taking simultaneously into account the information from the set pf positives and the set of negatives. More importantly, the matrix would become less sparse and predictions could be made for a lower number of positives.

In order to reduce the computational time of PU-BMRF, a first step would be to identify and discard the features that are weakly contributing to extraction of RN. Also, to discard those features whose values change depending on which genes are in the training set. Were these features discarded, the computation of features would need to be computed only once. This could lead to a significant decrease of the running time given that the computation of features accounts for nearly 82% of the running time of PU-BMRF.

Further aspects to be improved are, for instance, a more accurate extraction of RN, for instance, using Self organizing maps as explained in [13], or the Rochio technique [1]; accounting for the magnitude of coexpression; or including an option to make predictions for individual genes rather than for each GO-term.

Conclusions

PFP via networks for poorly annotated species

Prediction performance will be more accurate for those species for which the network data has low enn and epn. The portion of gene-Go associations that have been annotated among all the possible annotations (given all the possible GO-gene combinations). seemed to be also very relevant. More than the total amount of gene-Go associations that are known (better prediction sin yeast than in humans. Non-validaed associations ere not of much help.

The size of the network, however, does not seem to be important, as well as the characteristics of the co-expression network. In other words, conditionally independent network seem to be more effective for PFP using co-expression data..

Moreover, prediction swill be more stable (lower sd across replicates) as the reliability of the level of co-expression is larger and the number of annotated gene-Go associations is high.

PU-BMRF as a PFP method

1. PU-BMRF outperformed BMRF using chicken data: 0.758 vs 0.706 AUC. However, the standard deviation across replicates was more than double (0.078 vs 0.026)

- The best performance of PU-BMRF was when a maximum of 3000 RN were extracted
- The reproducibility in the process of extraction of RN was high, with an average of 96.7% of the RN extr5acted being extracted in the four replicates considered.
- When the RN were extracted randomly, PU-BMRF achieved a low prediction performance (0.652 using chicken data), lower than when the conventional BMRF was used (0.706).
- The increase in accuracy when PU-BMRF was used instead of BMRF was higher for the most specific GO terms (0.126, pvalue=0.53*)
- The computational time is considerably larger than for BMRF (about 200 times more). This is because the computation of features is computationally expensive given the large amount of gene and GO-term combinations.

BMRF and PU-BMRF for poorly annotated species

- Although overall predictions are better for chickens when we look at specific GO terms, predictions are better for humans than for chickens, which highlights the difficulty of doing PFP in poorly annotated species
- Both, the conventional BMRF and PU-BMRF achieved a good performance on chickens. However, predictions can only be made by a maximum of 364 GO terms due to the GO-size-filters that BMRF uses.
- the most important limitation of BMRF and PU-MBRF for poorly annotated species is that a minimum of genes annotated with the GO terms is required (minimum 8) genes.
- The increase in accuracy when PU-BMRF was used instead of BMRF was higher for those GO terms with a high epp/tpepp ratio (correlation of 0.22, pvalue 0.21*). We observed that the ratio epp/tpepp for a given GO terms was considerably higher in the chickens data, therefore we would expect that PU-learning is particularly effective in poorly annotated species where the portion of unlabeled genes is large.

Data properties that determine the prediction performance

- In the trade-off between the extension of the network and the quality of it, BMRF seemed to perform better when a the co-expression threshold was set to 0.35.
- Although using a low co-expression threshold lead to a higher prediction performance, the prediction becomes less reliable (high standard deviation across replicates).

- The size of the network does not seem to have a strong effect on the prediction performance
- The GO term property most strongly correlated with the prediction performance was epp/tpepp
- Decreasing the number of epn and enn in the network seemed to improve the prediction performance considerably
- The size of the network did not seem to have a strong impact on the accuracy of prediction.

Biology behind co-expression networks

- The characteristics of the co-expression analysis (tissue, experimental; conditions…) had a very low impact on the prediction performance.
- The genes from the most specific GO termed seemed to be more interconnected, with a correlation between epp/tpepp and specificity of 0.235 in humans. Furthermore these correlation increased as we carried the analysis only with sets of the most general GO terms general GO terms.
- Genes that are exclusive from specific GO terms seem to have a higher degree of connection (6.3% more edges) than the genes that are exclusive form general GO terms

Others

- An in-depth overview of the differences in data between the species considered ,a s well as the current state of annotation is provided in Appendix II-Data-overview.

* This correlation was estimated with only 30 GO terms. This could explain the high p_value.

References

1.KOURMPETIS, Y. A. I., VAN DIJK, A. D. J., BINK, M., VAN HAM, R. & TER BRAAK, C. J. F. 2010. Bayesian Markov Random Field Analysis for Protein Function Prediction Based on Network Data. Plos One, 5.
2.SCHWIKOWSKI, B., UETZ, P. AND FIELDS, S.. 2000. A network of protein-protein interactions in yeast. Nature Biotechnology 18:1257 − 1261.
3.MINGHUA DENG, KUI ZHANG, SHIPRA MEHTA, TING CHEN & SUN., F. 2004. Prediction of Protein Function Using Protein–Protein Interaction Data. Journal of Computational Biology, 10, 6.
4.YAN, K. K., WANG, D. F., ROZOWSKY, J., ZHENG, H., CHENG, C. & GERSTEIN, M. 2014. OrthoClust: an orthology-based network framework for clustering data across multiple species. Genome Biology, 15.
5.JOACHIM W. BARGSTENA, EDOUARD I. SEVERINGB, J.-P. N., GABINO F. SANCHEZ-PEREZA & DIJK, A. D. J. V. 2014. Biological process annotation of proteins across the plant kingdom. Current Plant Biology, 1.
6.Consortium TF, Andersson L, Archibald AL, et al. Coordinated international action to accelerate genome-to-phenome with FAANG , the Functional Annotation of Animal Genomes project. Genome Biol. 2015:4-9. doi:10.1186/s13059-015-0622-4.
7.BHARDWAJ, N., GERSTEIN, M. & LU, H. 2010. Genome-wide sequence-based prediction of peripheral proteins using a novel semi-supervised learning technique. Bmc Bioinformatics, 11.
8.YANG, S. K. 2012. Positive-unlabeled learning for disease gene identification. Bioinformatics, 28, 2640-2647.
9.YOUNGS, N., PENFOLD-BROWN, D., BONNEAU, R. & SHASHA, D. 2014. Negative Example Selection for Protein Function Prediction: The NoGO Database. Plos Computational Biology, 10.10.YANG, P., LI, X. L., MEI, J. P., KWOH, C. K. &
10.JIANG, M. & CAO, J. Z. 2016. Positive-Unlabeled Learning for Pupylation Sites Prediction. Biomed Research International.
11.XIAOLI, LI & BING, LIU. 2003. IJCAI'03 Proceedings of the 18th international joint conference on Artificial intelligence. 587-592.
[12]**https://bmcgenomics.biomedcentral.com/articles/10.1186/1471-2164-14-13**
[13]DDI https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1546-7
16. Sharan R, Ulitsky I, Shamir R (2007) Network-based prediction of protein function. Molecular Systems Biology 3: 1–13.R. SharanI. UlitskyR. Shamir2007Network-based prediction of protein function.Molecular Systems Biology3113

19. https://www.frontiersin.org/articles/10.3389/fpls.2016.00444/full

20. CAFA https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3584181

21. http://msb.embopress.org/content/3/1/88.long
22. https://www.nature.com/articles/s41598-017-00465-5
23.EUC_dist. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3467748/