# PROTEIN FUNCTION PREDICTION FOR POORLY ANNOTATED SPECIES

**FERNANDO BUENO GUTIÉRREZ**
**890605-143-090**

**Major thesis Bioinformatics: BIF-80336, November 2017**

**Examiners:**

Dr. Aalt-Jan van Dijk

Prof. Dr. Ir. Dick de Ridder

Dr. Ir. Hendrik-Jan Megens

**Thesis: Bioinformatics**

# PROTEIN FUNCTION PREDICTION FOR POORLY ANNOTATED SPECIES

Major thesis Bioinformatics: BIF-80336, November 2017

FERNANDO BUENO GUTIÉRREZ
890605-143-090
MSC OF BIOINFORMATICS, WAGENINGEN UR

November 2017

Supervisors:

Dr. Aalt-Jan van Dijk, Bioinformatics, Wageningen University

Dr. Ir. Hendrik-Jan Megens, Animal Breeding and Genomics Centre, Wageningen University

Table of Contents

## Figures Index



## Index of Tables

## Equations Index

## Glossary

AUC: Area under the curve

BMRF: Bayesian Markov Random Field

BP: Biological process

CAFA: Critical Assessment of protein Function Annotation

CC: Cellular Component

EES: Experimental evidence scores

GO: Gene Ontology

MCMC: Markov chain Monte Carlo

MF: Molecular Function

MRF: Markov Random Field

PFP: Protein function prediction

PU-BMRF: Positive-unlabeled version of a Bayesian Markov Random field

PU: Positive-unlabeled

RN: reliable negative

sd: standard deviation

## Abstract

Protein function prediction (PFP) is an important goal in current biology. In poorly annotated species, PFP is conventionally based on annotation transfer from the few well-studied species, such as Arabidopsis and humans. Although these methods are successful for inferring molecular function, they are not effective in predicting the biological process in which genes are involved. Biological processes, however, can be predicted using network methods. In this thesis, we expanded upon an existing network method (Bayesian Markov Random Field – BMRF) and developed a method that is efficient for PFP in poorly annotated species, such as chickens. We hypothesized that BMRF does not perform at its best in the PFP context because BMRF requires two classes of examples, and in PFP only positive examples are known. We used Positive unlabeled learning (PU-learning) to overcome this problem. After applying PU-learning to the BMRF using chicken data, the area under the curve (AUC) for the prediction performance significantly increased, from 0.706 (0.026) to 0.758(0.084). In this thesis, we also provide an overview of which type of data adjusts better to the method used, as well as some aspects regarding the biological context of the approach.

## Introduction

### 1. 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 Arabidopsis and humans. While successful, these methods rely on the assumption that homologous proteins share functions, which has been proved wrong in many cases [1]. Another disadvantage, both of homologous-based methods and general methods based 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 [3]. 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 principle, proteins that interact are likely to have similar function [3]. The principle of guilt-by-association does not apply, to a large extent, to the molecular function GO (Gene Ontology) category, but it does hold for the biological processes [1]. Therefore network-based methods can be used to infer the biological process in which the proteins are involved. Furthermore, network methods have great potential because they can utilize the accumulating information generated by high-throughput biological experiments, such as co-expression [4] or protein-protein-interactions [5] to construct networks from which to infer function.

Network methods can be of different types. Sharan et al. (2007) [6] distinguished two main types of network 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, the unannotated proteins are assigned a function that is unusually prevalent in the module. Direct methods proved to be slightly superior to the indirect ones.

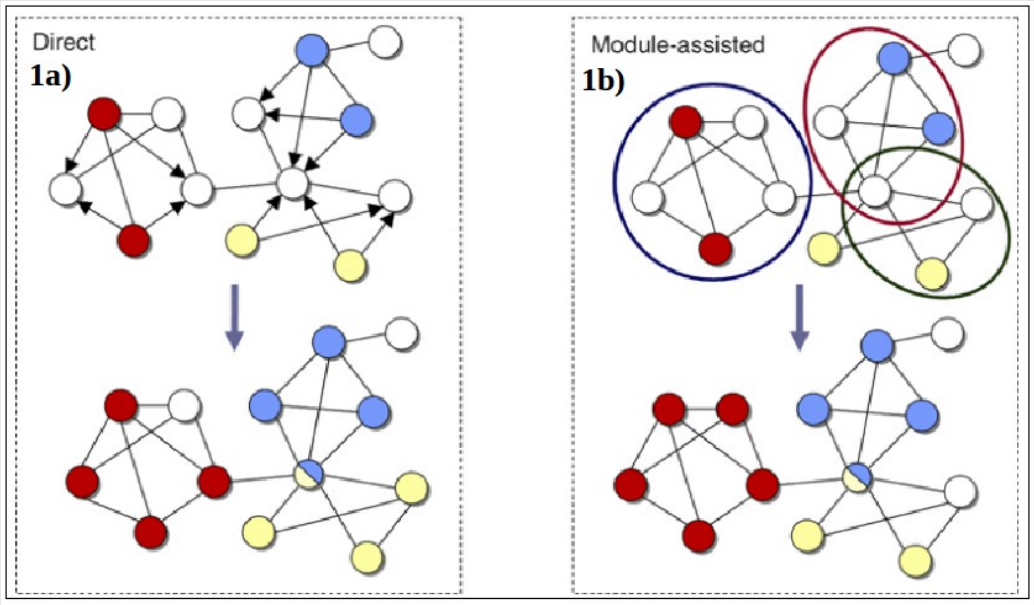



*Figure 1: Direct versus Module-assisted network methods.*

*Source: Sharan et al. (2007) [6].*

*In the network methods, proteins are represented as nodes, and the edges represent co-expression between proteins or protein-protein-interactions. In the example, the colors represent the different functions and proteins without any known functions appear as white. It is assumed that proteins can have more than one function. In the direct methods (Figure 1a), the function of a protein is inferred from its annotated neighbors (directed graph). In the module-assisted methods (Figure 1b), first modules of proteins are identified, and then the function of the proteins within a module is determined based on its members.*

## 2. Networks in poorly annotated species

Since 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 (for instance, co-expression or protein-protein-interaction) is limited [7]. On the other hand, PFP via networks is more challenging for these species because the data may be insufficient (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 [8] 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*.

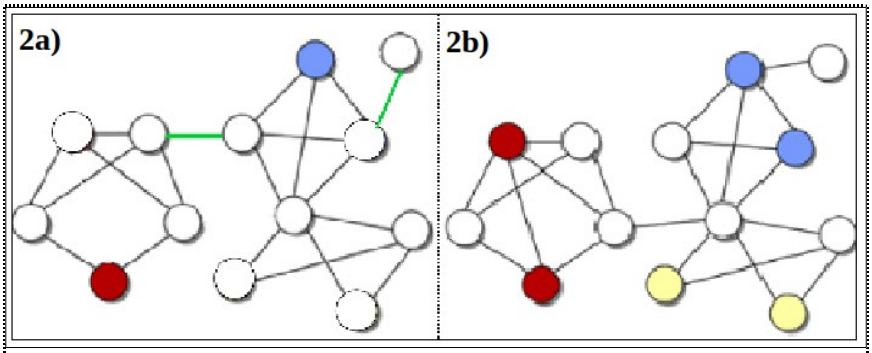



*Figure 2: Example of network in poorly annotated species.*

*Source: Adapted from Sharan et al. (2007) [6].*

*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 to an improved annotation.*

*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 2a) were proven false when the annotation improved (Figure 2b).*

In livestock species, there is increasing interest in functional annotation. Efforts such as the Functional Annotation of Animal Genomes consortium (FAANG) [9] are currently generating functional annotations regarding genotype to phenotype links for relevant species such as pigs and chickens. Although network data for livestock species may be even more limited than for the aforementioned crop species, some studies have used network data from livestock species. Stanley et al. 2013 [7], for instance, used a co-expression network in chickens to infer function via defining GO-enrichment-modules. 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. In their approach, however, they used a module-assisted method, which proved to be less effective than the direct methods [6]. Furthermore, the method used by Stanley et al. 2013 [7] did not make use of statistical learning to identify combinations of features that correlate with certain functions. It has been shown that statistical learning approaches can be very helpful for PFP via networks[10]. An interesting question, therefore, is whether it is possible to achieve accurate PFP in livestock species using direct statistical learning-based methods.

## 3. Bayesian Markov Random Fields

In order to develop a statistical learning-based network method that is efficient for livestock species, a good starting point is to use one of the methods that have been used for PFP in crop species. Bayesian Markov Random Field (BMRF) is a prediction method that was developed with the purpose of achieving accurate predictions when the data is far from complete [4]. The method can be used as a classifier to distinguish genes that are associated with a certain function from genes that are not.

Markov Random Fields (MRF) are direct methods in that they exploit the guilt-by association-principle. Based on this, adjacent nodes in the network are likely to have a similar function. MRF is part of a group of methods called “graph-theoretic methods”. The main characteristic of the graph-theoretic methods is that they seek to compute annotations for all network nodes at once while optimizing global criteria [1].

In order to do this simultaneous computation, these methods assume that the function of each gene is independent of all 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) [10] experiment, and its prediction performance was high for some species (Figure 1). BMRF is particularly efficient for poorly annotated species for two reasons: 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.

Since BMRF exploits the principle of guilt-by-association, we expect that by testing this approach on biological data it should be possible to gain insights into how biological networks work. 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. Moreover, genes that are involved in co-expression cascades are expected to have a larger number of neighbors in the co-expression network in which they share a functional role. In BMRF, predictions are expected to 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.

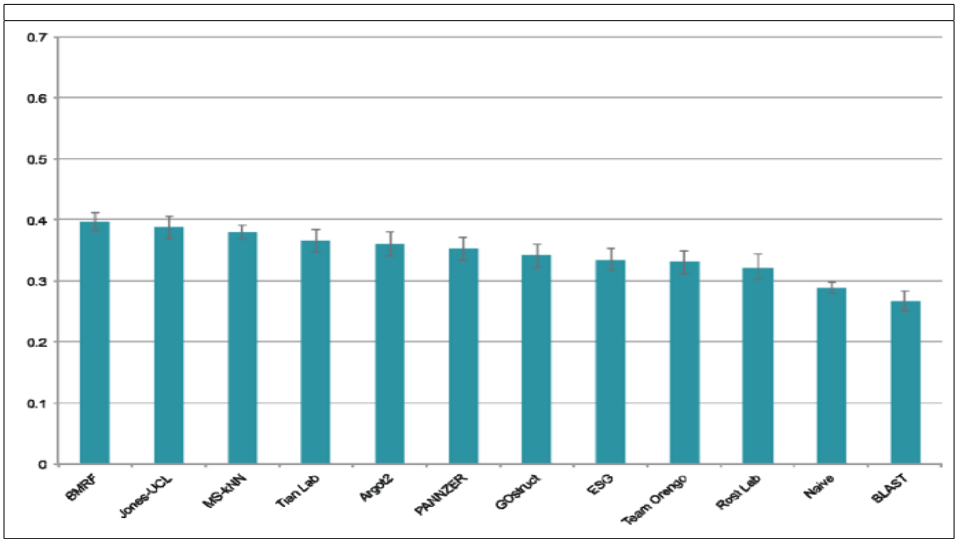



*Figure 3: Comparison of BMRF with other PFP methods.*

*Evaluation for the Biological process category in H. Sapiens.*

*Source: Radivojac et al (2013) [10].*

## 4. Positive-unlabeled 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 classifier expects that the input data consists of two classes of examples: proteins that have the function and proteins that do not have the function. However, from a biological perspective, it is very difficult to find negative examples (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 classifiers attempt to solve a one-class classification problem when in fact the annotated data consists solely of positive cases. In rice for example, 415 proteins have experimental evidence for a biological process, but not a single protein has a validated proof of no-connection with a function [8]. 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. Moreover, we would expect that this problem becomes more severe as the numbers of unannotated proteins increases, as in the case of poorly annotated 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 less likely to have a given function (Figure 4). Hence, the number of unlabeled cases can be minimized by extracting some “reliable negative” proteins from the unlabeled set. It has been proven theoretically that, by identifying sets of reliable negatives, PU improves the performance of machine learning algorithms in situations where only positive labels are known [11].

PU has been successfully applied to a variety of problems related to PFP [11-16]. Bhardwaj et al. 2010 [11], for instance, extracted a set of reliable-negative proteins from an unlabeled set by defining a threshold of similarity based on the Euclidean distance between a set of positive proteins and the unlabeled set. Yang et al. 2012 [12] developed a multi-label version of PU learning to identify genes associated with diseases;

Youngs et al. 2014 [13] developed two novel approaches to identify reliable negatives that can be applied in different algorithms. Jiang et al. 2016 [14] applied PU on a support vector machine and outperformed all pre-existing methods for pupylation site prediction. Nan et al. 2017 [15] improved the method developed by Jiang et al. 2016 [14] by adding, as a previous step, the method described in Bhardwaj 2011 [11]. Lastly, in another recent study, Nusrath et al. 2017 [16] used a self-organizing map to extract reliable negatives from unlabeled data sets of drug-drug interactions. None of these studies, however, have applied PU learning to BMRF. BMRF was developed specifically for networks with large portions of unlabeled examples, and PU can extract reliable information from the unlabeled data. 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, as in chickens. Additionally, since BMRF exploits the principle of the guilt-by-association, the approach may increase understanding of how the co-expression cascades are reflected in the network data.

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 groups of genes that play relevant roles in co-expression cascades.

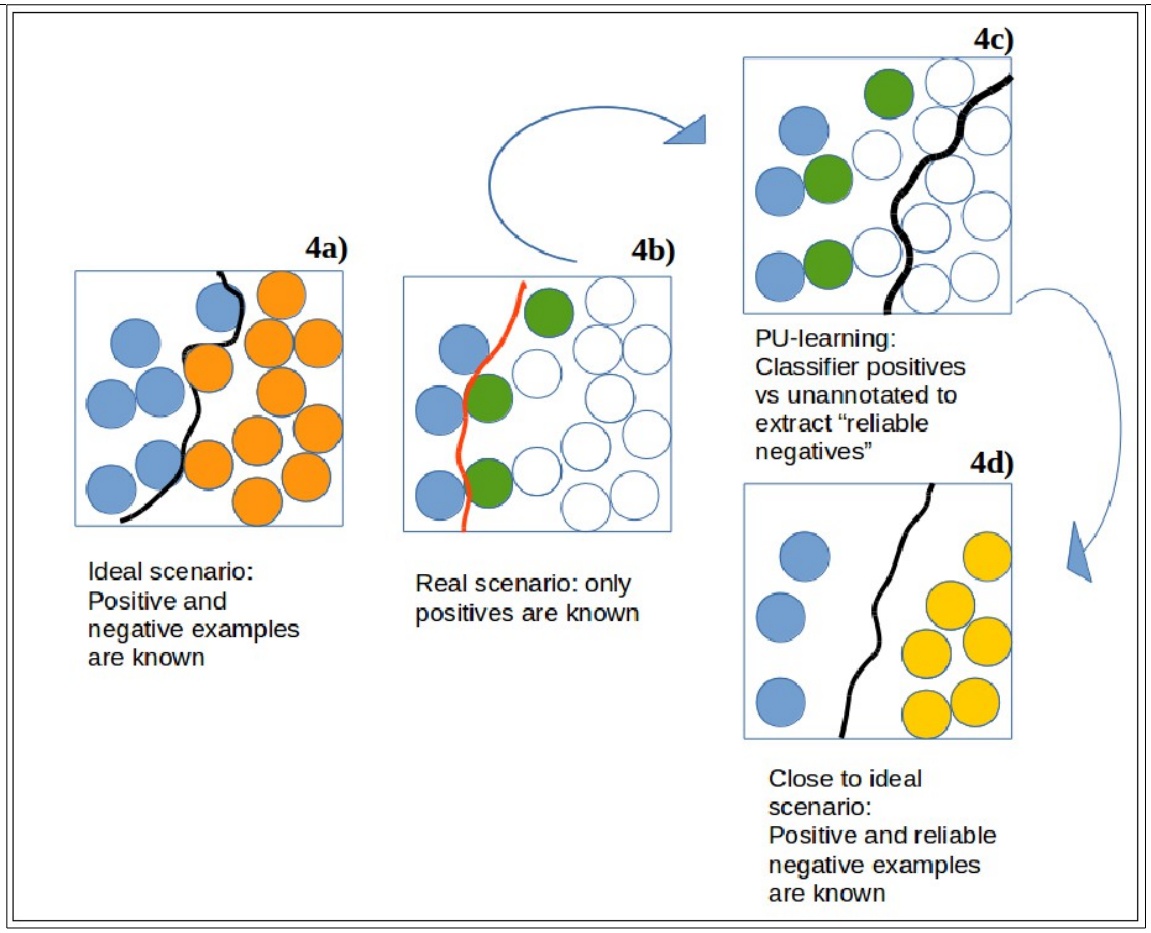



*Figure 4: PU-learning diagram.*

*In the ideal scenario (Figure 4a), examples from two classes ("positive" and "negative") are known, and a conventional classifier can be used. Such a classifier can then be used to infer whether the novel examples correspond to the positive or the negative class. In the PFP context, negative examples are not known. For instance, in Figure 4b, there are not two differentiated classes: instead, what we observe is that balls are either blue or not, and we have reasons to assume that some of the "non-blue" balls are blue balls that are not known to be blue. Blue balls that are not known to be blue are represented as green in Figures 4b and 4c. So, in the PFP context, the green balls represent false negatives, so proteins that have the function of interest but whose association with the function has not yet been discovered. 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 interprets that it has to differentiate between the positives and the "non-positive class". Subsequently, the classifier may choose a boundary that is excessively strict. Such a boundary may cause false negatives in the assignation of 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 no 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 resemble more the ideal scenario.*

## 5. 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 were required to achieve reproducible results. In Part 1, we also performed an 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 excessively low number of edges.

Parts 2 and 3 aimed at 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 gain insight into what constitutes a "poorly annotated" species when it comes to PFP via networks and what represents the minimum data 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. It is expected that PU-learning will help in situations as occur in this study where only positive examples are known. However, it may be the case that BMRF could be greatly improved with other strategies, such as 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 prediction method but the quality of the data.

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

In Part 5, we aimed to identify groups of genes that may be playing relevant roles in co-expression cascades.

The thesis contains three appendices.

- Appendix I-Concepts. This appendix contains the definitions of most of the terms and concepts that were used in the thesis.
- Appendix II-Data. In this appendix, we provide the sources for the data used in this thesis, as well as a summary of the data available.
- Appendix III- Additional results. In this appendix, we included the results that were not relevant enough to be placed in the main results.

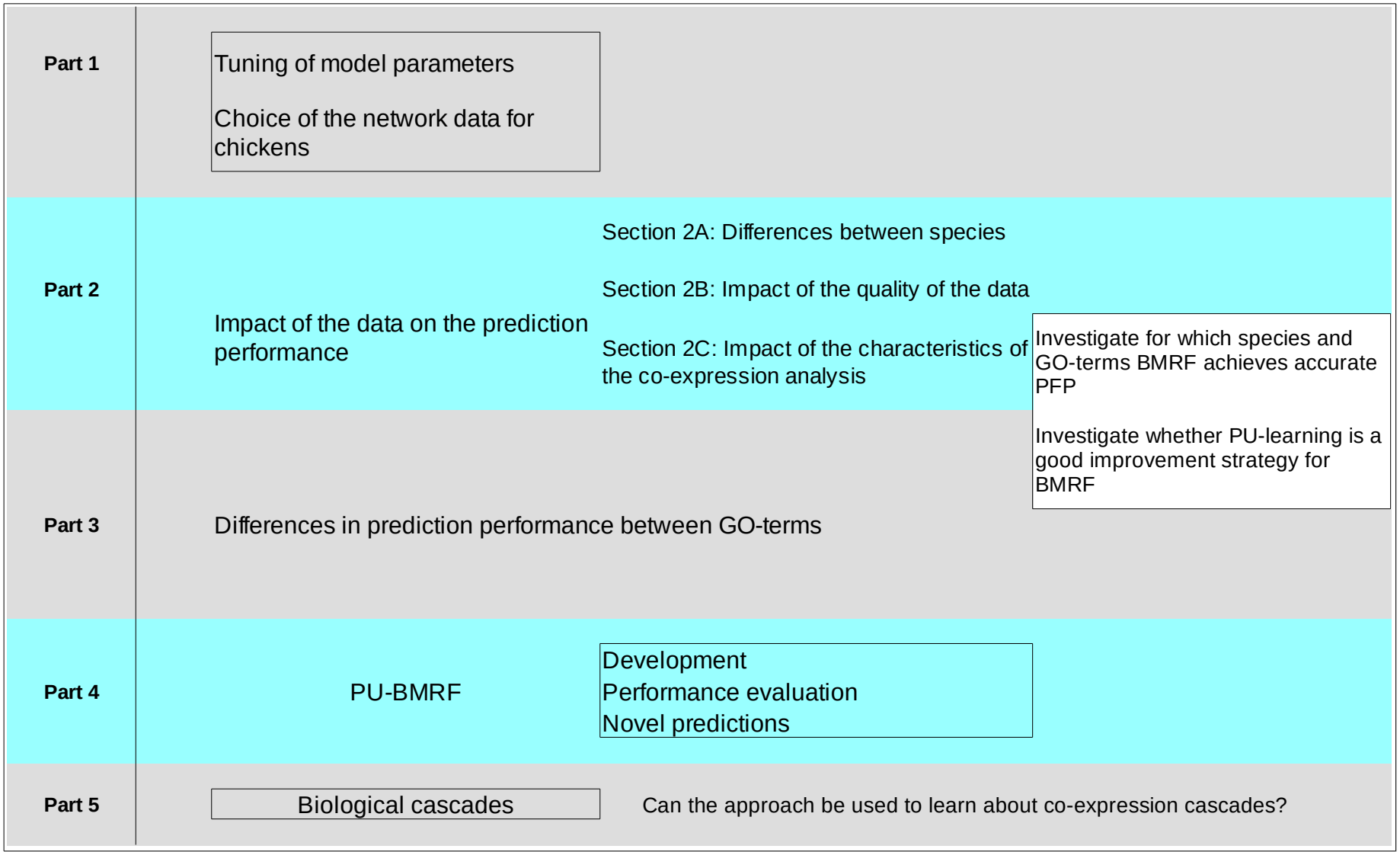



*Figure 5: Thesis setup.*

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

# Material and methods

## 1. Basics of the network method

In the Bayesian Markov Random Field (BMRF) framework, the aim is to assign genes to GO terms, where the GO terms represent the function of the genes. In particular, in this thesis, we aim to assign genes to the biological process (BP) GO category. Predictions are made for each GO term individually. The “target GO term” is the GO term whose prediction we are interested in at a given moment.
For each gene in the data, we will predict whether it is associated with the target GO term or not. Thus, after having considered all the genes, we will get one estimate of accuracy of prediction per GO term. This estimate represents how accurately BMRF predicted which genes are associated with the target GO term and which are not. Or in other words, which genes have the function and which genes do not.

The genes are represented as nodes in the network and the edges represent co-expression between two genes.
The relationship between a GO term and a gene is represented by a label with two possible levels: the gene is associated with the target GO term, or the gene is not known to be associated with the GO term. The label is the variable that we aim to predict, and it is known in the training set.

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 are labeled "1" (and a color other than white). The edges are the same for all the GO terms, but the network is not exactly same for all GO terms because the configuration of the labels of the genes is different depending on the GO term.

## 2. 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 analysis were carried in the three species and yeast PPI. However, some analysis were carried only on yeast data because this data was made available earlier, whereas some other analysis were carried on human, as we thought that chicken data would not become available and humans resembles more the situation in chickens.

Chickens are 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 is 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 these species are given in Appendix II-Data.

We used an existing BMRF code for protein function prediction (PFP) [4]. The code takes three files as input (Figure 6).

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 one 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) and Molecular Function (MF) in the group of “NONvalid”. We used the NONvalid associations to get an estimate corresponding to the extend to which the 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 the non-validated associations from the BP category was combined with the associations from CC and MC into a single “NONvalid” group of associations was to limit the number of analysis.

Note that “valid” and “NONvalid” are just a level in a binary class that is defined for each association and should not be misled with the labels that are assigned 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 it can be decided at a given moment whether we want to include or not the “NONvalid” associations in the train set.*

When an association between a gene and a GO term is known, it is assumed that the gene is associated with the GO terms that while being in the same branch, are more specific. The information about the more general GO terms is useful and was obtained by up-propagating the GO-file. To maintain the distinction between "valid" and "NONvalid" associations throughout the analysis, these two groups of associations were up-propagated separately, and then both sets of associations were combined into a single file, with the advantage that the variable “validation” can be added (Figure 7b).

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. 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 is 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'.

## 3. Markov random fields

Markov random fields (MRF) are random fields that satisfy 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 the target annotations [19].

A typical application of MRF is image restoration, where the value of a pixel (color) can be predicted based on the value of the neighbor's pixels. The strength of MRF to infer the class of an element is that they allow for simultaneous predictions of many elements, and therefore MRF can potentially achieve accurate predictions even when none of the neighbors are unknown. MRF is successful, in general, for prediction problems in which the principle of guilt-by-association holds, like image restoration or the prediction of BP from network data.

In a MRF, the probability of a certain assignment of discrete states x=x1,...xN is, as explained by Sharan et al. (2007) [6]:

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

*Equation 1: General formula of MRF*

*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 simplify 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) [5] adapted a MRF to the PFP problem and stated that the probability over the enter network is proportional to exp(αN01+βN11+N00), where N01, N11,N00, correspond to the number of pairs of proteins that: while interacting, only one has the function; both of them have the function; and none 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 prior probability of an assignment with N1 "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 (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) [5] 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) [5], overcame this problem by simply ignoring the unannotated proteins in the parameter estimation step. This can be particularly problematic when the number of unknown proteins is large, like in the case of poorly annotated species. Note that by ignoring the unknowns, the neighbors are pruned and the network may no longer express the co-expression relationship between the genes.

## 4. Bayesian Markov random fields (BMRF)

Kourmpetis et al. (2010) [4] 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.*

*A Differential Evolution Markov Chain algorithm is used.*

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, where d is the parameter dimension (d=2, in this case). ZR1 and ZR2 are randomly selected from 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:

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

*Equation 5: Probability at which θ' is accepted in BMRF.*

*A Metropolis step is used.*

Where PLF is the pseudo-likelihood function, which is an approximation to the joint probability distribution of a collection of random variables. Then, the labeling vector x is initialized using the output of the MRF defined by Deng et al. (2003) [5], as explained by Kourmpetis et al, (2010) [4].

## 5. Validation in BMRF

BMRF [4] was used for PFP to 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. The predictions, however, are not completely independent for each GO term in the data set 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 extent the GO-size filter and the number of unknowns affects 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, and are independent of the GO term considered.

A distinction should be made 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", similarl to the

'unknown genes'. In other words, the label "-1" can be used to hide the labels of the genes in the test set.

The reason why the label “-1’ is considered in the BMRF code is that the genes that have never been predicted as positives are less likely to be non-associated for a given function than the genes that have been found as positives for other functions. This has to do with the fact that it is easier to experiment with some genes than with others. Thus, if based on data a gene has never been identified as positive for any given function, it is fairer to assume that it is difficult to prove the function of the gene than to assume that the gene has no function. This difficulty in the predictions can also be observed at the level of GO terms. Thus, if for a given GO term, the portion of unlabeled genes (labeled as “0”) is large, we expect that the function is rare, (almost never observed), whereas if for a GO term a large portion of the genes are labeled as “-1s” we should assume that the function is difficult to predict. As explained before, this distinction between label “0” and label “-1” additionally allows for more accurate predictions by assigning label “-1” to the genes in the test set. One additional aspect to be considered is that if the portion of genes labeled as “-1” becomes excessively large with respect to the portion of genes labeled as “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 labeled as “-1”. Assigning label “-1” makes it possible to hide the “real label” of the genes in the test set until the posteriors have been estimated with BMRF. The output of the code is a vector of posteriors corresponding to the probability that each of the genes in the database is associated with the function. Then, the Area under the curve (AUC) for the GO term of interest is computed by contrasting the posteriors of the genes in the test set with the “real label” that was hidden (either a “1” if the gene is known to be associated with the function, or otherwise a “0”). In this thesis, AUC refers to the Area Under a Receiver operating characteristic (ROC) Curve.

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

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

| Gene classes in BMRF | Gene class as defined in Appendix I-concepts | Label in BMRF input | "Real label" (expected output) | Fold |
|---|---|---|---|---|
| Associated but the association is non-validated | Positive "NONvalid" | 1* | | |
| Not known to be associated with any GO term | Unkown | -1 | | |
| Associated and validated but the "real label" is hidden in this fold by labeling as '-1' | Positive-test | -1 | 1 | a |
| Not associated. The "real label" is hidden in this fold by labeling as '-1' | Unlabeled-test | -1 | 0 | b |
| Associated and validated. The label is not hidden in this fold | Positive-train | 1 | | a |
| Not associated. The label is not hidden in this fold | Unlabeled-train | 0 | | b |

*The process is independent for each GO term. By association we refer to association with the target GO term (the GO term of interest).*

** The genes only take 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 new fold and the label of a different group of genes is hidden, 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. In each fold, a new k set takes place as Positive-test and the rest are Positive-train until each of the k sets defined at the beginning of the analysis has been assigned exactly 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 consists of 5 parts. The rest of the Materiel and Methods is dedicated to describing the methodology in each of these parts.

Unless specified, all analysis was carried 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.

## 6. Part 1- Tuning of model parameters and choice of the co-expression data

**Tuning of model parameters**

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

Previous to the tuning of the model parameters, we aimed to choose the number of replicates that 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 replicates. For this study, it was considered that results were reproducible if the standard deviation (sd) across replicates was lower than 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, would mean that the GO terms whose number of known associated genes is less than 10 are excluded from the analysis. In the case of "maxGOsize", the argument should be a numeric between 0 and 1, since the value in this case corresponds to the portion of genes in the network. Thus if, for instance, there are 10,000 genes in the network and the argument for "maxGOsize" is 0.3, the GO terms that are associated with more than 3,000 genes are excluded from 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 arguments should be an integer larger than 0. The BMRF code does not allow for values of minGOsize lower than 8.

A novel model parameter that was defined in this study is 'only_EES”, which stands for “only experimental evidence scores (EES)”. The argument is a logical that is true if interest is in removing from the analysis all the associations between genes and GO terms that are not validated. If the logical is false, the “NONvalid” associations (Figure 7) will be included in the train set (they will never be included in the test set), as explained in Table 1.

**Choice of the co-expression data**

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 GO-file is pruned based on the network file. The code does not allow for genes that are in the GO file but not in the network file. We computed AUC for datasets with different Pearson co-expression thresholds. Note that for each Pearson correlation, the GO file needs to be up-propagated, because the GO-file is pruned based on the network.

## 7. Part 2- Impact of the data on the prediction performance

In part 2, we focus on how the network and the GO-files influence the prediction performance. Interest is 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, as well as which species are better predicted. For instance, BMRF is more efficient for species with extensive datasets or for species for which a large portion of the annotations is known. Part 3 consists of 3 sections:

2A) Differences in prediction performance between species

2B) Impact of the quality of the data on the prediction performance

2C) Impact of the nature of the network on the prediction performance

### 2A) Differences in prediction performance between species

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

#### Differences in network data

A total of 8 parameters were defined to account for the differences in the network data between the species. These parameters are unique for each dataset. For instance, a unique value for the yeast co-expression data-set, and a unique value for the chicken co-expression data when the Pearson correlation threshold was 0.35. Nevertheless, with the exception of the first two parameters 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 8 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 (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 term, 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 known 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 calculated as: n*(n-1)/2, where n is the number of genes that are associated with the GO Term.
- #epp/tpepp standardized. The ratio between #epp and tpepp standarized.

**Differences in annotation data**

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

- #genes per GO term: average of the # genes associated with each GO term
- #GO terms per gene: average of the # GO terms that are associated with each GO term.
- #assoc*1000/total possible assoc: # of associations between genes and GO terms for a given 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.

## 2B) Impact of the quality of the data on the prediction performance

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 assess the impact of the extension of the data considered and the prediction performance.

## 2C) 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 some key words, like "oxidation" or "stress" and created subsets of networks with data from the experiments whose description of the experiment includes the key words. 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, since in Part 1 we observed that '#te' has barely any impact on the prediction performance. In order to homogenize the size of the subsets by randomly removing edges from the networks. 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.

## 8. Part 3- Differences in prediction performance between GO terms

We investigated the impact of different GO term-properties on the prediction performance

A total of 9 GO term-properties were defined. Values for these parameters are specific for each GO term. However, 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), also described in part 2, but here it is GO-term-specific rather than species-specific
- eppA/tpEppA: (edges positive-positive divided by total possible edges positive positive, including also NONvalid associations). As epp/tpEpp but includes also associations coded as “NONvalid” (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. The 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. More specifically, it is the Area Under a Receiver operating characteristic (ROC) Curve.
- 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.

## 9. Part 4- PU-BMRF

### PU-BMRF development

In Part 2, (same as in Part1) the folds for the sets of test and train were created solely based on the validated associations. 10fold -CV was used, however, due to time constraints, the analysis was performed with only 4 replicates (instead of 20 replicates 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 1-6 aim the computation of 77 features including 63 non-GO-specific features and 14 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 train-positive set and train-unlabeled-set ("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) [12] (Figure 8):

1. $RN = \emptyset$;
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. $\quad Ave\_dist += dist(pr, Vg_i)/|U|$;
7. **For each** $g_i \in U$ **do**
8. $\quad$ **If** $(dist(pr, Vg_i) > Ave\_dist)$
9. $\quad\quad RN = RN \cup \{g_i\}$

*Figure 8: Algorithm for extraction of RN.*

*Source: Yang et al. 2012 [12]*

In step 1, an empty set of reliable negatives is created. Then a set of positives is created with the information form the gene sin the positive-train-set, and this is normalized. Then, the average Euclidean distance between the genes in the positive-train-set and the unlabeled-train-set is computed. Finally, for each gene in the unlabeled set, the Euclidean distance to the positive-train-set is computed, and this

distance is compared with the average Euclidean distance computed in steps [4-6]. This way, genes that are very distant from the set of positives can be extracted as RN.

Note that step 6 in the algorithm allows introducing 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 RN here extracted are used as a representative class of negative examples (figure 4d). Thus, once the RN are extracted BMRF is trained to differentiate between two classes of examples (positives and RN).

In the process of extraction of RN, interest is in utilizing features that are different from the features that BMRF uses. Otherwise, the process would be redundant and no progress would be made. BMRF uses network information, however, there is plenty of neighbor information that BMRF does not use. For instance, neighbor information for different co-expression thresholds. Note that although in Part 1 we computed AUC for different co-expression thresholds, only information from using one of these thresholds was used at a time. Furthermore, since BMRF makes predictions for each GO term individually, it neglects some information that may be useful for separating genes that have a given function from gene that do not have it. For instance, the number of GO terms that the gene is associated with may be relevant to extract RN, as well as knowing whether the gene is associated with any related GO term.

The computation of the features that are used to calculate the Euclidean distances in steps 6 and 8 of figure 8, consists of 5 steps:

- **Step 1 – Similarity Matrix:**

A similarity matrix between the GO terms was computed. The computation of this matrix serves two purposes: first, it allows to extract a set of unrelated GO terms from a bigger subset of GO terms. This allows making a selection of very unrelated GO terms in case that, for instance, due to time constrains, it was not possible to carry the analysis for all the GO terms; and second, the matrix will be used in the computation of the GO specific 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, we extracted subsets of genes that will be used in step 5- Computation of GO specific features.

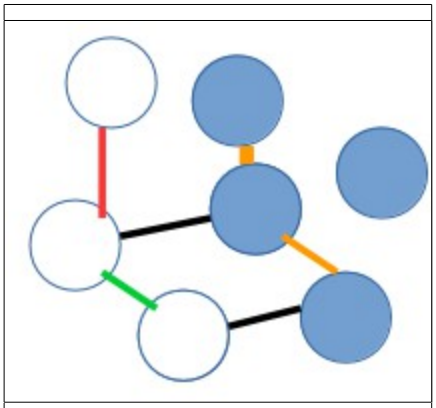
For instance, we stored the sets of genes that, after the creation of the test set, were associated with the GO terms. Furthermore, the neighbors of these were also stored. Additionally, we subseted the neighbors

of each gene.

Thus, for each GO term, three sets were extracted: the set of positives genes associated with the GO term of interest, the neighbors of the positive genes, and the neighbors of the gene. These sets differ between folds and are used several times in Steps 4 and 5. Thus, storing these sets at the beginning of the analysis for each fold saves computational time.

- **Step 3 – Network features:**

Three different sub-networks are created. Then, in step 5, these sub-networks can be used to compute the transitivity, closeness and betweenness of each gene in these sub-networks. Figure 9 helps to illustrate these sub-networks.



*Drawing 1: Dummy diagram to illustrate the sub-networks in step 3.*

The sub-network are:

- A sub-network with only those edges that are connected, at least, to one positive gene. Thus, the edges black and orange in Drawing 1. This is expected to be useful for extracting RN because the positive genes (and the positives yet to be discovered) are expected to be more interconnected in this sub-network.

- A sub-network with only those 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. Thus, the edges black and orange in Drawing 1, and the green edge, since both of its nodes are neighbors of positives. We expect that in this sub-network the genes in the positive set are more interconnected than the genes that are not associated with the GO term (non-positive genes).

- A sub-network with all the edges except those that link two genes in the positive set. Thus, the edges red, green and black in Drawing 1. We expect that in this sub-network, the genes that are positive are less interconnected.

- Step 4 – non-GO-specific features:

Here we computed 70 features with information regarding gene properties that may be useful for extracting RNs. The features computed in this step are not GO specific and therefore they are computed only once for all the GO predictions. For each gene, the following features are computed:

- Features f1-f4 refer to the number of GO terms that the gene is associated with. 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 was up-propagated because these genes may be involved in different unrelated functions and therefore they could potentially be regulatory genes. By accounting for this information, putative regulatory genes will have fewer chances of being extracted as RN for any given GO term. Genes that are associated with a large number of GO terms only after up-propagating, however, may be associated with fewer specific GO terms. For instance, it may be the case that they are associated with a GO term whose branch in the GO-hierarchy tree is very large and therefore, after up-propagating, these genes appear to be associated with a large number of GO terms, when in fact they play only one functional role.

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 before up-propagating, including NONvalid associations.

The “NONvalid” associations may be useful in (f4) because it also includes the associations from the molecular function and cellular component GO categories. A gene that is associated with a large number of molecular functions and cellular components should have fewer chances to be extracted as a RN for any given GO term.

- 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 when the NONvalid associations were also included.

- Features f7-f9 refer to the number of neighbors.

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 include information from other networks with other co-expression thresholds.

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 the genes that are co-expressed with the genes that have multiple functions are more likely to have multiple functions as well, and therefore these genes 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 in the data. Note that we are only interested in features that have a minimum of variability across genes.

- Features f10 to f63 are same as f1-f9 but for different Pearson correlation thresholds. Pearson correlation cutoffs used were: 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 genes in the GO-file that is not in the network. We expect 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 may be a gene involved in many different functions. Furthermore, 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 observe different patterns of co-expression in the network.

- **Step 5 – GO-specific features:**

For each gene, we computed up to 13 GO-specific features. These features need to be computed once for each gene-GO term combination. Moreover, they need to be computed once per fold, because the information changes from fold to fold. For instance, the number of neighbors of the gene of interest that are positive may change from fold to fold, as different positive cases are assigned to the test set.

First, for each gene to be annotated, we defined 4 sets of genes. Then, the features defined in this step, are computed using this information on sets.

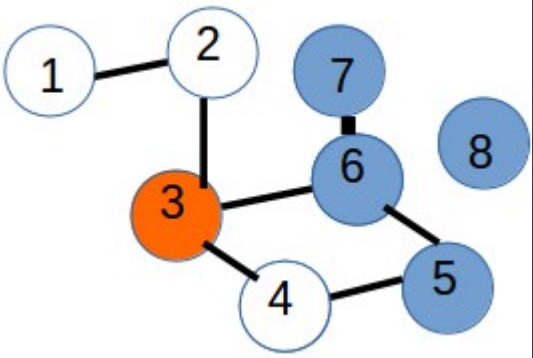
The sets are composed of the following genes:

(1) The gene that we want to annotate, if it is in the training-positive-set. This set is composed of 0 or 1 genes.

(2) The neighbors of the gene that we want to annotate, if they are in the set of training-positive-set.

(3) The gene that we want to annotate if it is in the set of neighbors of the training-positive-set. This set is composed of 0 or 1 genes.

(4) The neighbors of the gene that we want to annotate that are in the set of neighbors of the training-positive-set




*Drawing 2: Dummy diagram to illustrate the sets defined in step 5.*

*Blue circles represent genes in the training-positive-test.*

Drawing 2 may allow for a more clear understanding of the four sets of genes. The orange circle represents the gene to annotate and the blue circles represent the positive-train-set.

In the dummy example, the sets would be composed of the following genes:

Set (1): would be empty because the gene is not a positive gene

Set (2): would be composed of gene 6

Set (3): would be the gene 3 because it is neighbor of the positive 6

Set (4): would be genes 3 and 4, since gene 4 is both a neighbor of the gene of interest and a neighbor of a positive gene.

f1-f4) Following from step 2, for each gene, we compute the sum of the betweenness, transitivity and closeness of the GO terms that are in the sets 1-4.

f5) The number of genes in set 2 corrected by the size of the sets. Two ratios are considered here: the first ratio is the number of genes that are both, neighbors of the gene of interest and neighbors of the genes in the positive set, divided by the total number of neighbors of the gene of interest. The second ratio is the same numerator as in the first ratio but divided by the total number of neighbors of the genes in the

positive set.

Note that the higher these ratios are, the higher the chances that the gene to annotate is associated with the target GO term. This is because if, for instance, the gene to annotate has two neighbors and both of them are also neighbors of positive genes, the chance is greater that the gene is associated with the GO term. However, if for instance the gene to annotate has a large number of neighbors but only two of them are also neighbors of the positives we would infer that the changes of the gene associated with the GO term are less.

These ratios are not computed for the sets 1-3 because this information is already accounted for in BMRF.

F6-f10) The number of domains of the genes in the sets 1-4, the number of genes that share domains in 1-4, and the number or unique domains that are shared between genes in the set 1-4.

f11-f14) The number of genes in sets 1-4 (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.

- **Step 6 – extraction of RN:**

The databases with features information obtained in steps 4 and five were combined, and the values of the features were scaled by dividing the value of each gene by the square root of the squared sum of the values of all the genes. 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 applied the algorithm in Figure 15:

We checked whether any of the RN was in the set of non-validated positive cases and if so, we removed those genes from the set of RN. A constant was placed on the right-hand side of the equation in step 6 of the algorithm described in figure 15, to regulate the tolerance in the process of extraction of RN. Thus, the constant allows to choose between being very strict in the process of extraction of RN and extract few RN but very reliable, or extract a larger number of RN at the cost of a lower reliability. Through an iterative process, we adjusted the threshold to the highest value that allows extracting a maximum number of RN. We gradually increased the threshold by 0.05 if the criteria were 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 randomly 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 the 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 Figure 1.

Note that a different set of RN will be extracted per fold, per replicate and per GO term. After the extraction of RN, we train the BMRF classifier (RN vs positives), instead of (unlabeled vs positives) as in the conventional BMRF approach..

**PU-BMRF performance evaluation**

**Evaluation of the process of extraction of RN**

We computed the accuracy of extraction of the RN. Note that, in principle, only unlabelled genes should be classified as RN. In BMRF, the expected label for the positives-test-set is "1", and for the genes in the unlabeled-test-set is "0". In the process of extraction of RN, the predicted label was "0" if the gene was classified as RN, and "1", otherwise. This means that there are four types of genes based on the prediction performance in the process of extraction of RN:

a) Genes whose expected label is "1" and its predicted label is "0". These genes are positive genes, and therefore they should not be classified as RN. Therefore, these genes they are false positive.

b) Genes whose expected label is "0" and whose predicted labels is "1". These genes are unknown genes that were not extracted as RN. They may have been properly classified.

c) Genes whose expected label is "1" and whose predicted labels is "1". These genes were well predicted since they are positive genes that have not been classified as RN.

d) Genes whose expected label is "0" and the prediction label is "0". These genes were also well predicted since they are unlabeled genes that were classified as RN.

This is also illustrated in Table 16.

*Table 2: Accuracy of prediction in the process of extraction of RN.*

| Accuracy of prediction for the extraction of RN | | | |
|---|---|---|---|
| **gene type** | **Predicted label** | **Expected label** | **Well-classified?** |
| a | 0 | 1 | No |
| b | 1 | 0 | Maybe |
| c | 1 | 1 | Yes |
| d | 0 | 0 | Yes |

Since, the interest in this case is to estimate the accuracy of extraction of RN, we did not take into account the genes of “type b” for the computation of the AUC. Note that if these genes were included in the computation of the AUC, the AUC could be inflated based on these genes. More so, considering that it is expected that a large portion of the genes in the network belong to “type a”, especially in the situations where the number of RN extracted is low. For the process of extraction of RN, same as for BMRF, we used

the Area Under a Receiver operating characteristic (ROC) Curve.

**Reproducibility in the process of extraction of RN**

In order to assess the reproducibility in the process of extraction of RN, we investigated which portion of the RN was extracted in all the replicates of the analysis. Further, we assessed the reproducibility across folds within the same replicate. For this, we calculated how many different RN were extracted per replicate (combining the extraction of each fold).

**Novel predictions with PU-BMRF**

After the extraction of the RN (PU-learning step), the positive genes and the genes in the set of RN were used to the train the BMRF classifier. It is expected that the classification will be more accurate now that the set of unlabeled genes have been replaced by a smaller set of RN. In BMRF, the RN were treated in a similar way than the unlabeled genes are treated in BMRF.

We considered the 30 GO terms for which we extracted RN (using chicken data). For each GO term, we trained a BMRF with 3000 RN as the representative negative class and the genes associated with the GO terms as positives. We aimed to test whether more genes were predicted as positives in the set of unannotated genes that have a non-validated association with the GO term of interest than on the other unannotated genes. We expected that this could work as a short of validation for both, our method, and the other methods that have been applied to infer the (non-validated) associations.

### 10. Part 5-Co-expression cascades

In this part 5, we investigated whether the more specific GO terms have a higher degree of connection in the network. 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. Lastly, in part 5 we provided a hypothetical example of how this information, together with other information derived from the PU-BMRF approach, could potentially be used to identify genes that play relevant roles in co-expression cascades.

## Results

The study consists of four parts, as explained in Figure 5. Most of the analysis involved the computation of the AUC-roc (Area Under a Receiver operating characteristic Curve) as a measure of the accuracy or

prediction in the task of assigning genes to the Biological process (BP) GO category. We referred to AUC-roc as simply 'AUC.'

We considered as significant a p-values below 0.05. The analyses were carried using co-expression data from three species (chicken, human, and yeast) and protein-protein-interaction data from yeast (yeast_ppi). Fir simplicity, we often refer in the text to these 4 cases as four species. In the figures of the Results, these species appear with the following color code: chicken (blue), human (green), yeast(red), yeast_ppi (orange).

Unless specified, the standard deviation (sd) that is provided (in brackets) besides the mean AUC, for instance, in Tables 3 and 4, refers to the standard deviation across GO terms rather than across replicates of the same GO term.

Additional results are given in Appendix III-Additional results

## 1. Part 1- Tuning of model parameters and choice of the co-expression data

**Turning of model parameters**
We tuned the following model parameters from the BMRF code [4]:

- GO-size filter. This includes "minGOsize" and 'maxGOszie'. "minGOsize" and maxGOszie specifies which is the minimum and maximum number of genes that each GO term should be associated with. GO terms that do not satisfy the GO-szie criteria are removed from the analysis.

- DF-size filter. This includes "minDFsize" and 'maxDFszie'. These specify which are the minimum and maximum number of genes that a domain should be associated with. Domains that do not satisfy the DF-size criteria are removed from the analysis.

- k-fold CV. The number of folds in the CV

- # iterations in the Gibbs-sampling

Results were as follows:
- 20 replicates were required to achieve reproducible results (standard deviation across replicates below 0.02).

- A GO-size filter of "minGOsize":20, "maxGOsize":0.1 seemed to be reasonable option. We observed that the impact of the values of the filter in the prediction performance was very limited. Furthermore, "minGOsize":20 is adequate since it is guaranteed that there are at least 2 positive cases in each fold. 'maxGoszie:0.1 is also adequate since we are not interested in predicting the most general GO terms.

- The lowest possible value for minGOsize in BMRF is 8. Values lower than 8 cause problems in the computation of the sparse matrices that the BMRF code uses in the prediction computations.

- Increasing the value of k in the k-fold CV, increased the prediction performance. The highest possible value was 20, given that the "minGOsize"was set to 20 and since it is required that there is at least one positive example in the test set. Although the accuracy of prediction was slightly higher for 20-fold CV than for 10-fold CV, we decided to use 10-fold CV because it guarantees that there are at least two positive cases in the test set in each fold.

- Increasing the number of iterations in the Gibb-sampling did not have any impact on the prediction performance or the standard deviation even when the number of unknown genes (genes with label "-1") was very large.

- Adding domain information helped considerably (~5% in AUC) in the case of human and chicken data, and ~0.2 in yeast and yeast_ppi, still significant.

We estimated the prediction performance when “NONvalid” associations were included in the train set. The “NONvalid” associations were described in figure 7 and included non-validated associations for the BP GO category as well as associations from other GO term categories different than BP. We observed that, for chickens and humans, AUC slightly increased when the “NONvalid” associations were added.

In the case of chickens, the increase was not significant. The mean AUC(sd) was 0.724 (0.08) before the information was added and 0.726(0.08) after including the “NONvalid” associations in the train set (p-value 0.68). However, for humans the increase was significant, 0.724 (0.08) and 0.726(0.08), respectively (p-value 0.048). Interestingly, in the case of yeast and yeast_ppi, adding “NONvalid” associations slightly decreased the prediction performance (significant). AUC(sd) was 0.792 (0.09) and 0.74 (0.016) in the case of yeast, and 0.747 (0.0992) and 0.714 (0.02) in the case of yeast_ppi.

**Choice of the co-expression data**

Co-expression data was limited for chickens, and therefore, we investigated whether it was possible to achieve a higher prediction performance using a lower co-expression threshold lower than the conventional threshold (Pearson correlation:0.7)

The prediction performance was higher for chickens when a Pearson correlation of 0.35 was chosen as a co-expression threshold (Figure 9). Therefore, for the rest of the analysis with chicken data, we used the network extracted when a Pearson correlation of 0.35 was used as a threshold for co-expression.

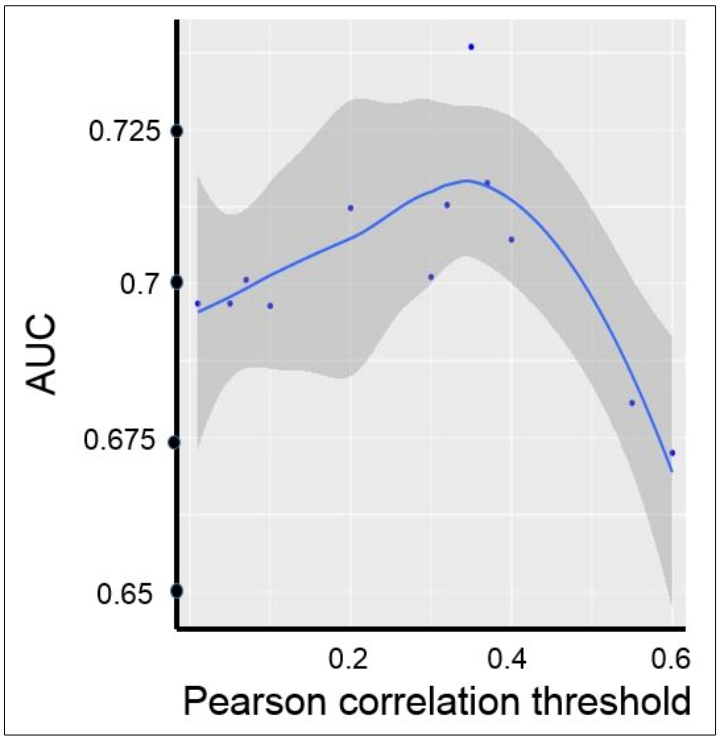



*Figure 9: AUC for different co-expression thresholds in chickens.*

*Twelve different Pearson correlation values were considered as co-expression thresholds.*

*The Y-axis corresponds to the average AUC for all GO terms considered in the analysis. The line corresponds to the lowness line.*

*Values for this plot are given in Table 20 in Appendix-III.*

## 2. Part 2- Impact of the data on the prediction performance

In this part, we aimed to investigate for which species BMRF is more effective for PFP. Moreover, we investigated to which extent do the quality of the data and the conditions of the co-expression experiment influence the prediction performance.

The main conclusions from part 2 were:

- BMRF achieved accurate PFP in chickens: AU (sdAUC): 0.726 (0.08).

- When we compared the prediction performance between the species considered at the level of individual GO terms, predictions were more accurate for humans than for chickens. Thus, as expected, BMRF performs worse for the less annotated species.

- Using a lower quality of the data (fewer annotations and less reliable edges) had a larger impact on the reproducibility of the results than on the overall prediction performance. Both, the reproducibility and the prediction performance decrease as the quality of the data worsens.

- The size of the network did not seem to have a strong impact on the prediction performance

- Reducing enn (edges negative-negative) and epn (edges-positive-positive) greatly improved the prediction performance.

- There was no difference between using the co-expression data derived from one or anther tissue.

Now we show the main results for the three sections of part 2.

2.a) Differences in prediction performance between species

2.b) Impact of the quality of the data on the prediction performance

2.c) Impact of the nature of the network on the prediction performance

## 2.a) Differences in prediction performance between species

In this section, we compared the prediction performance in the four cases considered (3 species and yeast_ppi).

We observed that predictions were more accurate for yeast, then for chickens, then yeast_ppi and lastly, for humans. The reason why the prediction performance was better for chickens than for yeast_ppi, was that a much lower number of GO terms were predicted when the chicken data was used and these GO terms were more easy to predict. Figure 10 and Tables 3-4 show the prediction performance for the species considered using BMRF.

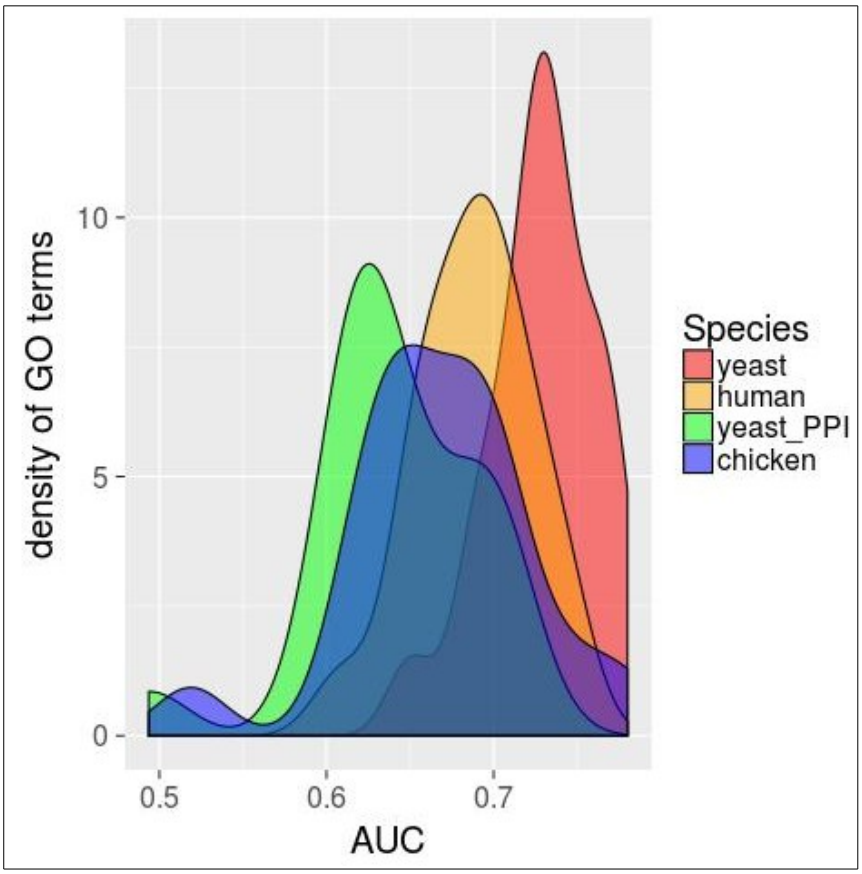



*Figure 10: AUC for the species considered.*

*AUC distribution considering only GO terms that are predicted in the 4 cases*

*Table 3: Overall prediction performance for the different species 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 4: Prediction performance for the common GO terms.*

| **species** | **AUC (sd)** |
|---|---|
| yeast | 0.729 ( 0.01) |
| yeast_ppi | 0.641 (0.014) |
| human | 0.688 (0.003) |
| chicken | 0.668 (0.039) |

In Figure 10, the density of the AUC for the GO terms is shown. In Table 3, the overall mean of AUC is shown for each species, and in Table 4, the AUC is shown after accounting only for the GO terms that were

predicted for the four species. Only 20 GO terms were predicted in the four species (common GO terms).

From Figure 10 and Tables 3-4, we concluded, that BMRF is able to achieve high accuracy for PFP (AUC>0.7) in chickens, which, to our knowledge, is the most poorly annotated species considered so far for PFP via networks.

Also from Tables 3-4 and Figure 10, we concluded that the overall value of prediction performance does not allow to learn for which species BMRF performs better, because the GO terms may be different in each case. Subsequently, we followed the analysis by comparing the prediction performance of the 20 GO terms that were common for the four species. Figure 11 shows a comparison of the prediction performance for these GO terms for the four cases considered. The names of the GO terms in Figure 11 are given in Table 5.

| | GO terms that are common in the 4 species |
|---|---|
| 1 | response to external stimulus |
| 2 | anatomical structure morphogenesis |
| 3 | response to endogenous stimulus |
| 4 | negative regulation of metabolic process |
| 5 | regulation of signal transduction |
| 6 | response to organic substance |
| 7 | negative regulation of macromolecule metabolic process |
| 8 | positive regulation of gene expression |
| 9 | negative regulation of gene expression |
| 10 | cell differentiation |
| 11 | negative regulation of cellular metabolic process |
| 12 | regulation of localization |
| 13 | cell development |
| 14 | positive regulation of response to stimulus |
| 15 | anatomical structure formation involved in morphogenesis |
| 16 | cellular response to chemical stimulus |
| 17 | regulation of developmental process |
| 18 | cellular developmental process |
| 19 | response to oxygen-containing compound |
| 20 | cell proliferation |

*Table 5: GO terms in Figure 11 (from left to right).*

In Figure 11, we observed that the prediction performance given the 20 GO terms that are common to all species, were overall larger for yeast, then for humans, then for chickens and lastly for 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. Interestingly, predictions were higher for chicken than for yeast_ppi. However, these results are not conclusive, since only 20 GO terms of the 1,019 GO terms that were predicted for yeast were considered.

The fact that predictions were high for chickens, even though a Pearson correlation of 0.35 was used (in the other three cases, 0.7 was used), suggests that, to some extent, BMRF has better prediction performance when the network has more edge (even if this comes at the cost of a lower reliability of the edges).

An interesting thing to notice in Figure 11 is that four of the GO terms (GO terms number 4, 7, 12 and 18) were better predicted in chickens and humans than in yeast. We did not find a biological explanation for why this was the case.

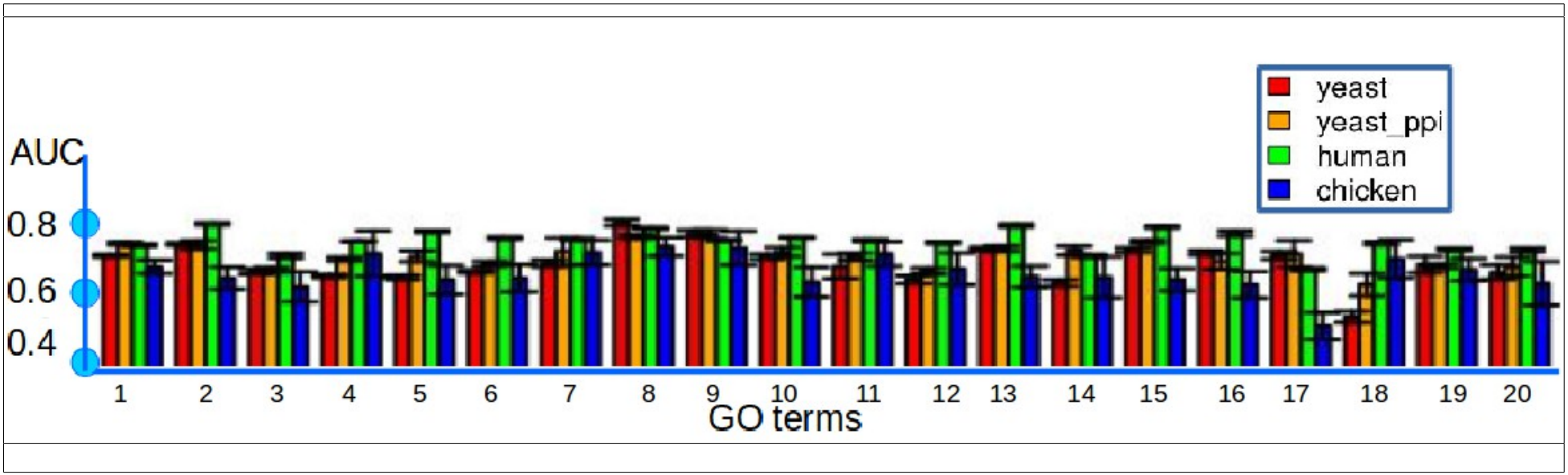



*Figure 11: AUC for individual GO terms, in the species considered.*

*The name of the GO terms from left to right appears in Table 3.*

## 2.b) Impact of the quality of the data on the prediction performance

In the section 2.b) we investigated the impact of the co-expression threshold on the reproducibility of the predictions, as well as the impact of removing associations or edges from the data.

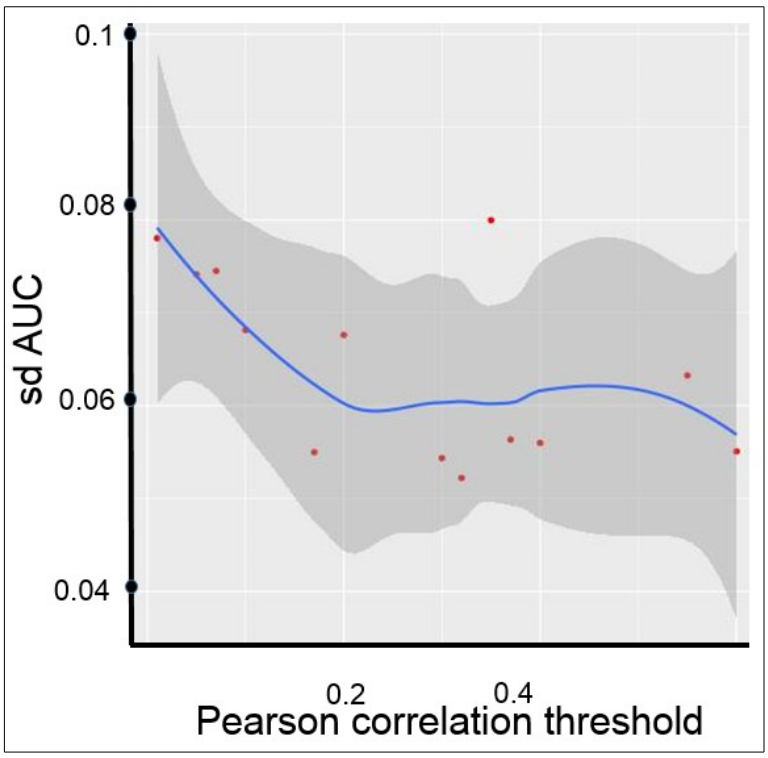



*Figure 12: Standard deviation across replicates for different co-expression thresholds.*

In Figure 12, we observed that the standard deviation across replicates decreased as more strict co-expression threshold was chosen. Thus, selecting a high co-expression threshold allowed for better reproducibility. Figure 13 shows a comparison of the standard deviation across replicates in the species considered. We observed that the standard deviation was considerably larger for the poorly annotated species than for the better-annotated species.

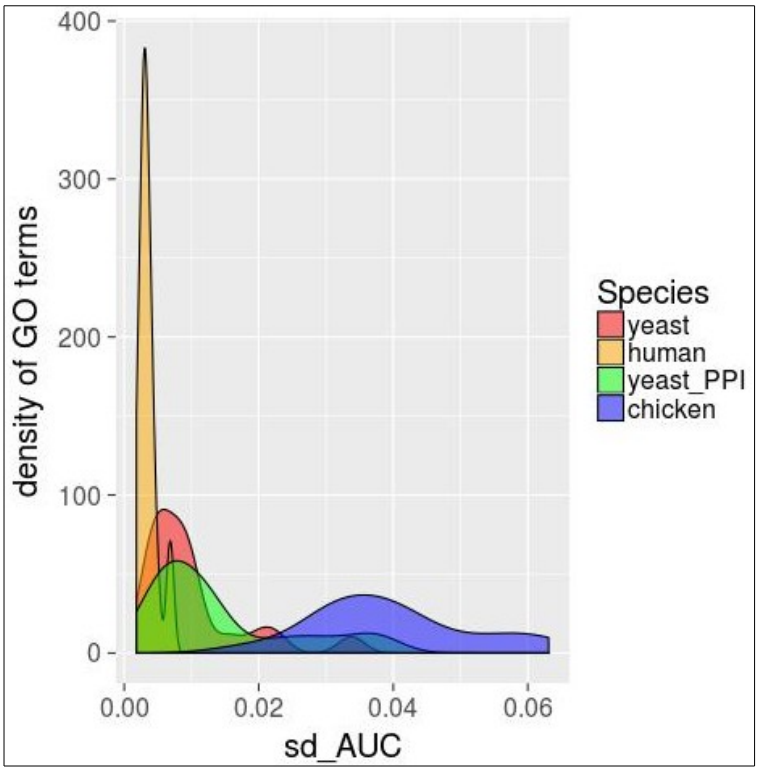



*Figure 13: Standard deviation of AUC across replicates. Values for the different GO terms in the different species.*

To investigate what is the minimum data required to achieve accurate PFP with BMRF, we removed randomly edges from the network and estimated the prediction performance in each case. Table 6 shows the AUC when 0, 10, 30, 50, 80 and 95% of the edges were removed.

*Table 6: Impact of the number of edges in the prediction performance.*

| 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 |

In Table 6, we observed that removing random edges from the data did not have a large impact on the prediction performance. The impact was slightly larger after removing 10 and 50% of the edges. It should be considered, however, that the decrease in AUC observed in Table 6 may be caused by different GO terms

being considered in the analysis. This may be the case since, in BMRF the GO term annotations are pruned based on the network data. Thus, we investigated the impact of the removal of edges on the prediction performance at the level of individual GO terms. From these analysis, we concluded that the removal of random edges from the network also had a small impact on the accuracy of prediction at the level of individual GO terms. (Table 20-21 in Appendix III).

Finally, some analysis were carried with human data to investigate further how the removal of data relates to an increment in AUC. The results showed that AUC increased almost linearly as 'epn' (edges-positive-negative) or enn (edges-negative-negative) were removed from the network. This could be regarded as an indicator that PU-learning may improve BMRF. Note that in PU-learning some negative examples are removed from the network, and therefore the number of epn and enn is expected to decrease.

### 2.c) Impact of the nature of the network on the prediction performance

We investigated whether the characteristics of the co-expression experiment had any impact on the performance. For instance, we would expect that if we use a network based on co-expression data from a particular tissue, predictions would be more accurate for the GO terms that are involved in relevant functions for that tissue. We defined 30 co-expression networks for chickens based on 30 different tissues, and for each GO term, we computed AUC. Then we calculated the difference between the AUC of the best performing tissue and the worst, for each GO term. The analysis showed that the difference was very low for most of the GO terms (Figure 14). Therefore, we concluded that there is barely any difference between using the co-expression data from one or another tissue.

Figure 14 illustrates the difference in prediction performance between the “best predictor tissue” and the “worst predicted tissue” for each GO term.

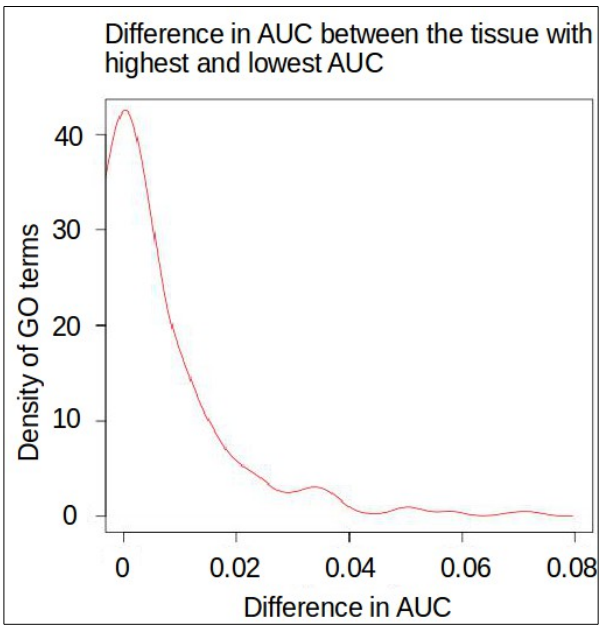



*Figure 14: Difference in AUC for the different GO terms.*

## 3. Part 3- Differences in prediction performance between GO terms

With the purpose of identifying for which GO term-properties we can expect that BMRF will achieve accurate PFP, we computed the correlation between AUC and the different GO term properties (Table 7).

*Table 7: Correlations with AUC.*

| | yeast | yeast_PPI | humans | chickens |
|---|---|---|---|---|
| **epp_V/tpEppV** | 0.62 | 0.373 | 0.462 | -0.178 |
| **epp/tpEpp** | 0.582 | 0.34 | 0.378 | 0.222 |
| **sdAUC** | -0.408 | . | -0.337 | -0.571 |
| **teV/tpeV** | 0.394 | 0.241 | -0.13 | -0.361 |
| **depth** | 0.265 | 0.104 | . | 0.439 |
| **spec** | -0.095 | . | . | . |
| **#genesV** | . | . | . | . |

*Comparison of the correlation across species. Only significant correlation s (p-value<0.05) were included in the Table.*

*epp_(V): edges-positive-positive, (considering only the validated associations)*

*tpEpp(V): total possible of edges-positive-positive, (considering only validated associations)*

*sdAUC: standard deviation of AUC across replicates of the same GO term*

*depth: depth of the GO term*

*spec: specificity*

In Table 7, we observed that epp/tpepp and sd were the parameters with the highest impact on the prediction performance. Epp/tpepp shows a favorable correlation with AUC, meaning that for GO terms whose associated genes are more interconnected in the network, the method distinguished better between genes that are associated with the GO term and genes that are not. The negative correlation between EppV/tpeppV and AUC found for chicken was counter-intuitive, but it was weak (-0.178) with a p-value of 0.0392 (Table 28 in Appendix III).

From table 7, we concluded that, to achieve high PFP accuracy, the ratio epp/tpepp should be as large as possible. Epp depends on the 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 dropped from the anlaysis. By improving the second ratio, PU-learning would be increasing the epp/tpepp ratio.

SdAUC showed 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 decreases as epp/tpepp increases. 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 genes are in the training or the test.

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 observed positive correlations between AUC and depth, even if lower values of depth imply that the GO term is at a more specific place in the GO hierarchy.

## 4. Part 4- PU-BMRF

### Performance evaluation

Table 8 contains the accuracy of prediction that was achieved in the two steps of PU-BMRF: first, the reliable negatives (RNs) were extracted from the set of unlabeled genes, and second, BMRF was trained on the set of positives and the set of RN. The results are given for different maximum numbers of RN extracted. (max # of RN extracted). Note that the total number of genes in the dataset was 12,424. Due to time constraints, we only made predictions with PU-BMRF for 30 GO terms. These 30 GO terms were unrelated, as explained in step 1 of part 4 in Material and Methods.

*Table 8: Accuracy in the two steps of PU-BMRF.*

| | 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) |

*Values are given for different choices of max #RN extracted. Results correspond to the average of 30 GO terms*

max # of RN extracted: Maximum number of RN that were allowed. Note that, as explained in Material and Methods we can specify how many RN we want to extract from the unlabeled set.

The prediction accuracy for these GO terms using BMRF was 0.706 (0.026). Thus, except for the situation in which 8000 RN were extracted, PU-BMRF outperformed BMRF. The maximum improvement (+0.052 AUC) was when a maximum of 3000 RNs per GO term were extracted (p-value: 0.0049). This is illustrated in Figure 15. Note that as we increased the number of RNs, PU-BMRF resembles more the BMRF scenario. This is because in BMRF, no RNs are extracted and the classifier is trained with all the genes (positives vs unlabeled). When, in PU-BMRF, a very large number of RNs are extracted from the set of unlabeled genes, the classifier (positives vs RN), becomes very similar to the BMRF scenario.

One disappointing results was that the standard deviation across replicates increased after PU-learning (0.03 with BMRF vs 0.076 with PU-BMRF). This increase was unexpected, since, as shown later, the reproducibility in the process of extraction of RN was very high. The most likely explanation for this is that the network that we used was smaller in the case of PU-BMRF than when BMRF was used. 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). Note that these values of AUC were computed as explained in Table 2.

To test the reproducibility in the process of extraction of RN, we estimated the portion of RN that were extracted in the four replicates considered. We carried the analysis in the situation where a maximum of 3000 was chosen. We observed that one average, 96.7% (0.0159) of the RNs were extracted in the four replicates. Results for 30 GO terms are given in Appendix III. Furthermore, we observed that, on average, 3033.583 (104.67) different RNs were extracted in the 1- folds of each replicate, which is very close to the 3000 minimum RN per fold. This also indicates that the large majority of the RN were common in the ten folds.

Another important result is that, as expected, the accuracy of prediction when the RNs were extracted randomly was lower than when BMRF or PU-BMRF were used (Figure 18).

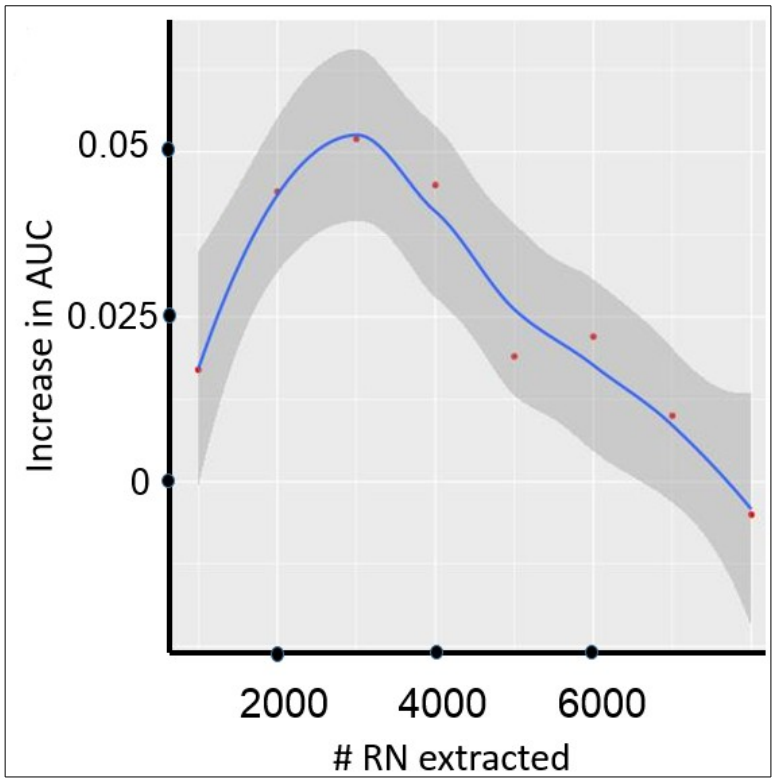



*Figure 15: Increase in AUC: PU-BMRF vs BMRF.*

*Values correspond to different choices of "maximum number of reliable negatives extracted".*

In Figure 16 we showed how the distribution of values of AUC changed when PU-BMRF was applied with 3000 RN. The AUC distributions only included values from the 30 GO terms that were predicted with PU-BMRF. We also carried the analysis extracting 3000 randomly. We observed that prediction was worst in the random analysis than when BMRF was used.

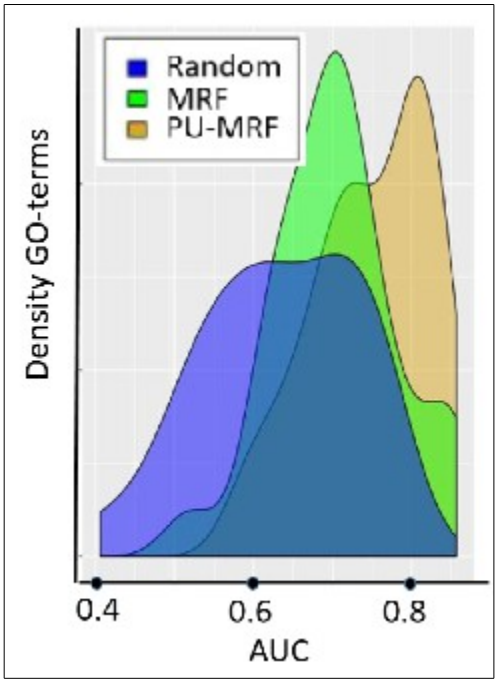



*Figure 16: AUC: PU-BMRF versus BMRF, density*

Figure 17 shows to which extend PU-BMRF meant an improvement in the accuracy of prediction for the 30 GO terms considered. It is shown that the portion of GO terms for which AUC was larger than 0.7 and 0.8 increased considerably after PU-BMRF was used, whereas there were fewer GO terms predicted with an AUC below 0.7, and no GO terms with AUC below 0.6.

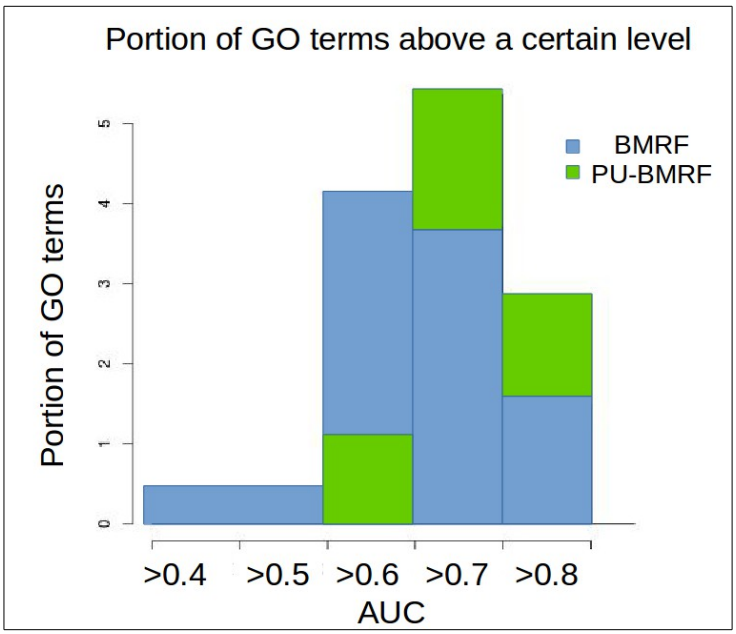



*Figure 17: Portion of GO terms above a certain AUC level (PU-BMRF vs BMRF)*

at the level of individual GO terms. We observed that the majority of the results for individual GO terms were not significant due to the fact that the standard deviation across folds increased considerably when PU-BMRF was applied. Results were significant only for the GO term “regulation of cellular component organization”. An interesting question, therefore, is whether this standard deviation would decrease if more than four replicates were considered.

*Table 9: Name of the GO terms in figure 18.*

| | GO term description |
|---|---|
| 1 | response to stress |
| 2 | sensory organ development |
| 3 | cell proliferation |
| 4 | response to external stimulus |
| 5 | anatomical structure morphogenesis |
| 6 | response to endogenous stimulus |
| 7 | animal organ morphogenesis |
| 8 | regulation of localization |
| 9 | cellular response to stress |
| 10 | locomotion |
| 11 | single-organism process |
| 12 | single organism signaling |
| 13 | single-multicellular organism process |
| 14 | single-organism metabolic process |
| 15 | single-organism cellular process |
| 16 | single-organism developmental process |
| 17 | anatomical structure formation involved in morphogenesis |
| 18 | regulation of cellular component organization |
| 19 | positive regulation of multicellular organismal process |
| 20 | mesenchyme development |
| 21 | regulation of biological quality |
| 22 | nucleic acid-templated transcription |
| 23 | organic cyclic compound metabolic process |
| 24 | organic cyclic compound biosynthetic process |
| 25 | response to oxygen-containing compound |
| 26 | regulation of nucleic acid-templated transcription |

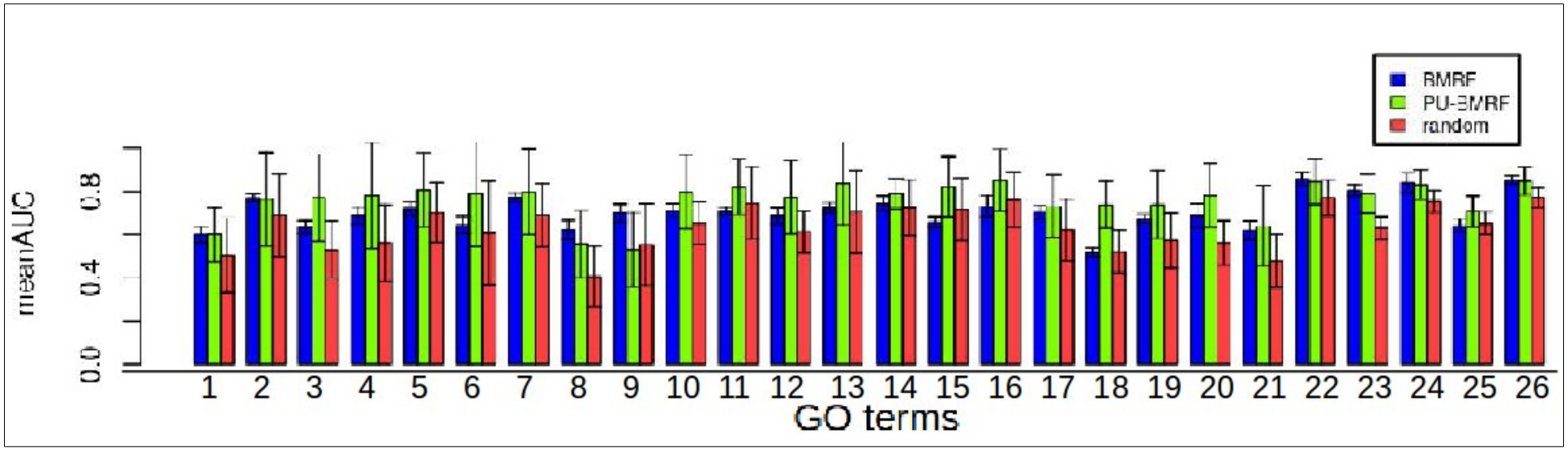



*Figure 18: PU-BMRF vs BMRF for individual GO terms*

*Values for this plot can be found in Table 31 of Appendix III-Additional results.*

Then, we investigated whether the increase in AUC with PU-BMRRF vs BMRF was correlated with any of the GO term properties described in Part 2. We observed that none of the GO term properties was significantly correlated with the increase in AUC. However, the strongest correlation was AUC-epp/tpepp (0.25), with a p-value of 0.21. We would expect that this p-value may become smaller if more than 30 GO terms were used to compute the correlation. This correlation may, potentially indicate that PU is particularly effective for chickens since the ratio epp/tpepp was larger for this species, as shown in Figure 19 and Tables 10-11. It is not clear, however, whether the reason why epp/tpepp was larger for chickens than for humans and yeast was that a lower co-expression threshold was used, or maybe that the ratio tends to increase as fewer positives are known, such as for chickens or other poorly annotated species.

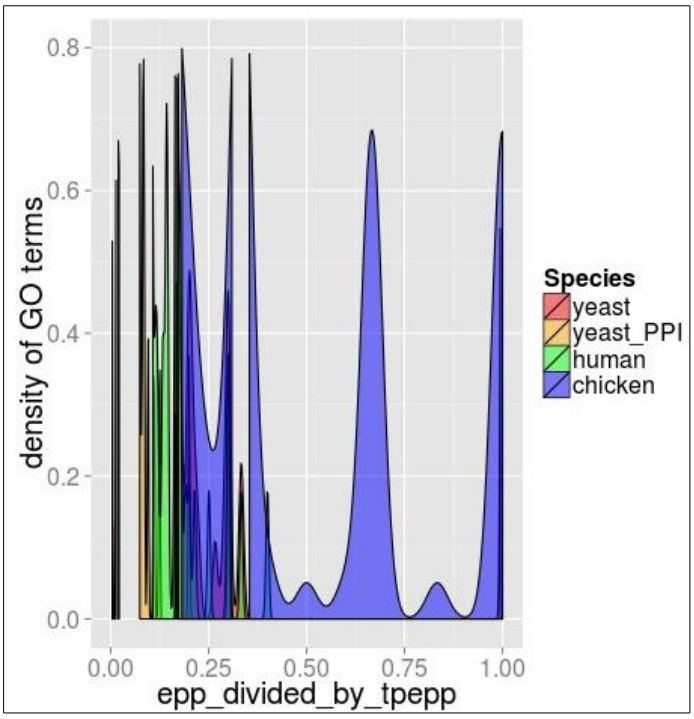



*Figure 19: Ratio epp/tpepp*

*Note: These analysis was for all the GO terms (not only those GO terms that are common in the four species)*

*Table 10: Average epp/tpepp per GO for the common GO terms*

| | Average epp/tpepp (sd) |
|---|---|
| yeast | 0.0582 (0.05) |
| yeast_ppi | 0.06 (0.14) |
| humans | 0.039 (0.018) |
| chickens | 0.144 (0.25) |

*Table 11: Average epp/tpepp per GO term for the different species*

| | Average epp/tpepp (sd) |
|---|---|
| yeast | 0.122 (0.204) |
| yeast_ppi | 0.106 (0.204) |
| humans | 0.066 (0.138) |
| chickens | 0.14 (0.27) |

**Novel predictions**

As a last step for Part 4, we applied PU-BMRF on a set of unlabeled genes in order to make novel predictions. We considered the 30 GO terms for which we extracted RN (so, chicken data). On average, for these GO terms, 2.1% (0.026) of the genes were predicted as positives. The GO term for which the highest number of genes were predicted as positives was GO:1901576 “organic substance biosynthetic process.” Of the 7,232 unlabeled genes for this GO term, 525 (7.3%) were predicted as associated with the GO term.

We compared the number of genes predicted as positives in two sets of genes: (1) the unlabeled genes that have a non validated association (“NONvalid”) with the GO term of interest, and (2) the rest of unlabeled genes. We expected to find find more novel predicted associations in the set (1). Although we did observed a certain difference, it was not significant (p-value 0.18). This suggests that the non-validated associations are not very reliable, as we also saw in Part 1, when we observed that including “NONvalid” associations did not contribute to a better prediction performance.

## 5. Part 5- Co-expression cascades

From a biological perspective, we expect that the genes that are involved in specific functions are more interconnected in the network. This is because as the biological processes become more complex, it becomes more unlikely that all the genes redeem “their part” of the function at the same time. In order to test this hypothesis in our data, we estimated the correlation between the specificity of the genes and the epp/tpepp ratio. We considered Epp/tpepp as an indicator of the 'connectivity' between genes.

| 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) |

*Table 12: Correlation between specificity and epp/tpepp*

In Table 12, we observed that the correlation between 'specificity' and epp/tpepp ranged between -0.1 and -0.24 depending on the species, and it was lower for poorly annotated species. Also, further analysis revealed that this correlation changed depending on the minGOsize. “minGOsize” was defined in Part 1 of Material and Methods and refers to the minimum number of genes in a selection of GO terms. Thus only GO terms with more than “minGOsize” associated genes are considered in the selection. Figure 20 shows the correlation between the (correlation epp/tpepp – specificity) and “minGOsize”.

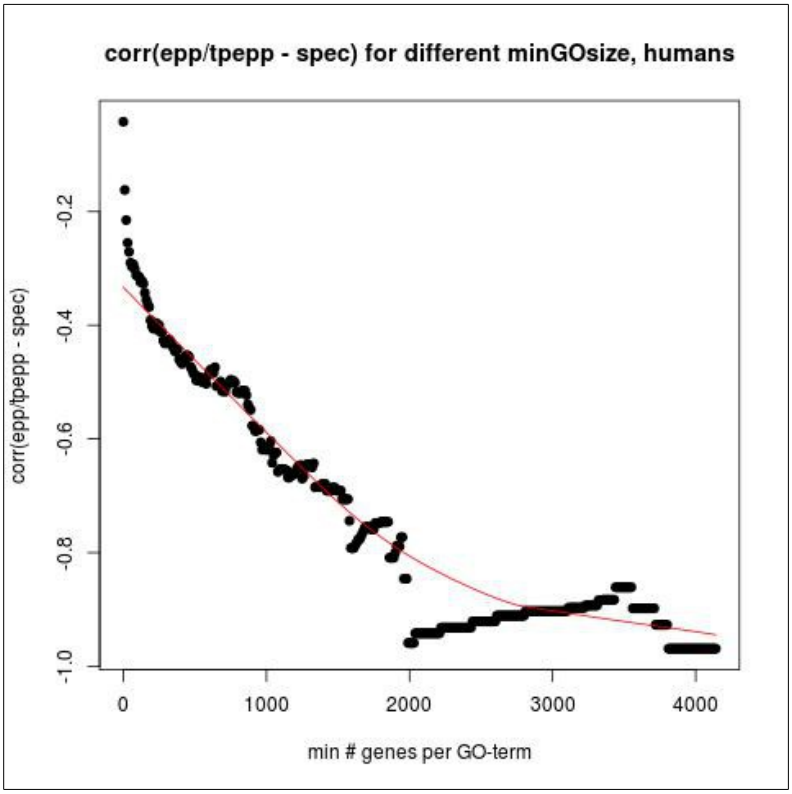



*Figure 20: Correlation between (correlation epp/tpepp - specificity) and minGOsize.*

*Human data*

In Figure 20, we observed that the correlation between (correlation epp/tpepp – specificity) and "minGOsize" is strong and reached its highest value when only GO terms with at minimum of 2,000 associated genes we considered. In other words, in Figure 20, we observed that within a group of very general GO terms (genes associated with a large number of genes), the hypothesis that "the genes that are associated with more specific GO terms are more interconnected in the network" holds to a larger extent. Interestingly, we observed that for yeast and yeast_ppi, the maximum value of the correlation between (correlation epp/tpepp – specificity) and "minGOsize" was also 2,000 (Part 5- Appendix III).

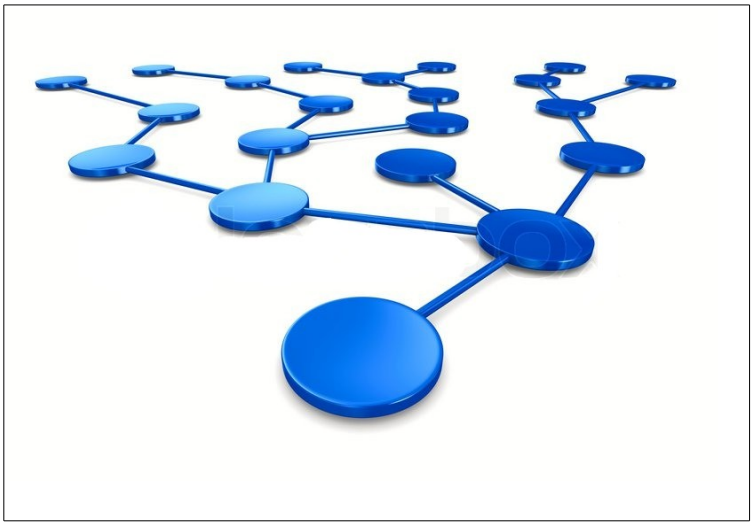
To investigate this hypothesis further, we compared the degree of connection between genes that are only involved in specific GO terms, and genes that are only involved in general GO terms.

We created 10,000 sets of genes of each type, and for each pair of sets, we compared the degree of connection in the network between members of the set. Out of the 10,000 comparisons, in 8,394 the genes the set of specific GO terms appeared to be more interconnected in the network than the genes in the set of general GO terms. On average, the set of genes that are involved in specific functions appeared to have 6.36 % more edges than the genes in the most general set.

**Conclusions for Part 5- Co-expression cascades**

In part 3, we showed that GO terms whose associated genes are more interconnected, are more easy to predict with BMRF (correlation of 0.62 between AUC and epp/tpepp for yeast, p-value:0.0001). In part 4, we

showed that GO terms whose associated genes are more interconnected benefit more from applying PU-learning. In part 5, we showed that the genes that are highly connected in the network are involved in more specific functions. We believe that this information could potentially be used to identify groups of genes that play relevant roles in the co-expression cascades. Drawing 3, shows a hypothetical example of how this could be the case.



*Drawing 3: Hypothetical example of how a method like PU-BMRF could potentially be used to identify genes that play relevant roles in co-expression cascades.*

*In drawing 3, one gene activates a co-expression cascade that is, in principle, a representation of a function cascade. The circle in the bigger size (closer), represents the gene that activates the cascade. First, this gene activates some genes that carry specific functions. Then, these genes start to activate other genes (as we farther from the original gene in the drawing). Eventually, due to the effect of all the genes in the cascade, some general functions are carried.*

*Hypothetically, the genes that activate the cascades (activator genes) can be identified based on certain aspects. For instance, we would expect that the activator genes are:*

*1- Associated with a large number of general functions. These are the functions that the whole cascade is involved in.*

*2- Associated with, at least, some specific functions. The functions required to activate the cascade.*

*3- Are co-express with fewer genes than what we would expect from -1-. This is because the activator genes only carry the general functions indirectly, through the activation of other genes that carry specific functions.*

*4- Their general functions are poorly predicted with BMRF. This is due to -3-.*

*5- The improvement in the accuracy of prediction when PU-BMRF is applied is more than for other genes.*

*Note that PU removes from the analysis certain genes within the set of unlabeled that, based on some features are very similar to the genes in the set of positives (Figure 4c). This may greatly improve the accuracy of prediction for the activator genes. This is because we would expect that the activator genes have some features in common with the set of positives of certain functions that it is not involved in. Features like, for instance, the number of neighbors that carry a function related to another function could be misleading in the case of activator genes, since these genes are indirectly involved in many different functions. In these cases, we could expect that PU-BMRF would exclude the activator genes from the analysis of that particular function by not considering it as a RN.*

## Discussion

Development of computational methods for PFP based on network data is a challenging problem in poorly annotated species. Here, we expanded upon a Bayesian Markov Random Field (BMRF) and develop a PU-learning implementation (PU-BMRF) that is more accurate than its predecessor for PFP in poorly annotated species, such as chickens. The efficiency of BMRF to infer the function of proteins in poorly annotated had been previously noted [4,8]. Nevertheless, this ability 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 this step, a set of reliable negatives (RN) are extracted. Subsequently, the BMRF classifier can be trained with a representative set of genes of each class (negatives and positives) and prediction become more accurate. Overall, 76% of the GO terms considered were more accurately predicted when PU-BMRF was used instead of BMRF.

The basis of PU-learning is in extracting genes within the set of unlabeled, that show strong differences with respect to the set of positives. For this to be effective, the features that are used to investigate these differences should 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. In our PU implementation, we accounted for 77 features to identify a set of reliable negatives (RN) for each GO term.

### Limitations

PU-BMRF has two main limitations. One of these limitations is that the standard deviation across replicates of the AUC for individual GO terms increased considerably when PU-BMRF was used instead of BMRF. This standard deviation was 0.03 with BMRF vs 0.076 with PU-BMRF. We did not find an explanation for this increase. In fact, the extraction of RN turned out to be a highly reproducible process. We observed that, on average, 96.7% (0.0159) of the RN were extracted in the four replicates considered. This is a remarkable result since based on the literature, the extraction of RN often has a low reproducibility. Hameed et al. (2017) [13], for instance, extracted 20,099 and 4,066 RN with two different approaches, and only 589 of the RN were common between the two sets.

To reduce the standard deviation across folds, a logical approach is to increase the number of replicates of the analysis. This is expected to be effective since the number of replicates that we used to test PU-BMRF was very low (4 replicates). However, this would increase the computational time and this is undesirable since the other main limitation of PU-BMRF is that it is computationally expensive.

PU-BMRF is computationally expensive because to extract the RN, it computes 77 features for each possible gene-GO term combination. The number of gene-GO combinations can be of the order of 2e+08. For instance, in humans, more than 12,000 genes are known, and the number of GO terms is more than 16,000. Furthermore, for the majority of these features, the value changes depending on which genes are on the train and the test set. The number of neighbors that are associated with a related GO term, for instance, is one of the features considered in PU-BMRF and needs to be estimated once per fold in the cross-validation.

One alternative to speed up the computation procedure is to replace some of the features by features that are not specific to each GO-gene combination. For instance, the number of neighbors that are associated with a related GO term is specific for each gene GO combination, but the number of neighbors of the gene at a given Pearson correlation threshold, needs to be computed only once and it is equally is valid for all the GO terms. In fact, Bhardwaj et al. 2010 [11] pointed that, to extract RN, it may be beneficial to take into account information that is independent of the GO terminology because some proteins defy the conventional annotation patterns.

Other approaches to reduce the computational time of PU-BMRF include: (a) Identifing and discarding features that are barely contributing to the extraction of RNs. (b) Discarding features whose values change depending on which genes are in the training set. This could lead to a significant decrease of the computational time required by PU-BMRF given that the computation of features accounts for nearly 60% of the total running time of PU-BMRF.

**Extraction of RN**

The reproducibility and the number of genes for which the predictions are made can, to some extent, be regulated in PU-BMRF by specifying the number of RN that we want to extract. The user can choose to extract a very large number of RN, although it would come at the cost of a lower accuracy in the process of extraction. For instance, Yang et al. 2012 [12] set a value of 1 as a cutoff in line 6 of the algorithm described in table 8. Schwikowski et al. 2000 [3] used the same algorithm but chose a value of 1.5. In PU-BMRF, we changed this value according to the GO term, as we aimed to extract a fixed number of RN for all the GO terms. We decided that this was a reasonable approach 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 line 6 of the algorithm in figure 8. or in any other equation that aims to separate RN form the unlabeled data, could be defined in such a way that none of the known positives would fall in the set of RN.

**Scope of improvement of PU approaches applied on network methods**

One important issue regarding the scope of improvement of PU is that when the predictions are made after using PU, fewer genes can be considered in the evaluation of the prediction performance. This is because in PU, the unlabeled genes that are not extracted as RN are excluded from the analysis. This is not a problem when the aim is to make novel predictions because most methods (including PU-BMRF) allow predicting any gene-GO association, regardless of whether the gene was removed during the PU step. However, the fact

that fewer genes are considered in the evaluation may be a problem when it comes to comparing different methods. To gain a fairer understanding the scope of improvement with PU, we consider that it is crucial to investigate why the AUC increases after PU. It may be the case that the AUC only increases because the unlabeled genes that were not extracted as RN are no longer considered for the estimation of AUC. This would lead to a higher AUC since these genes are expected to be more difficult to predict. On the contrary, it may be the case that the AUC increases because, as the nonRN are excluded from the analysis, so are their edges, and this is translated into better predictions of individual genes. Note that, if it was proven that the predictions do not improve at the level of individual genes, the increase in AUC that is observed after applying PU would not be a fair estimate of the improvement with PU.

Related to the potential that PU approaches have, it should be noted that most of the methods defined so far consist of adding a previous step to the classification algorithms. Therefore, different approaches can be combined to extract a more reliable set of RN. Schwikowski et al. 2000 [3], for instance, extracted a set of likely negatives (LN) based on the approached introduced by Kourmpetis et al. 2010 [4] and then they extracted a set of RN using another approach [3]. Deng et al. (2003) [5], for instance, extracted RN based on two sets of features, one that considers each feature individually and one that considered only the mean of the features of each group of features, and then extracted those RN that was extracted with both approaches. It should be noted, however, that the quality of the methods will not depend so much on the number of features that are defined or how many PU approaches are integrated, but rather on two other aspects: (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 the first factor, Bhardwaj et al. 2010 [11] referred to the so-called “moonlighting” proteins to those proteins that have a unique combination of features and cannot be predicted from the conventional annotations. To be able to classify these “moonlighting” proteins, it is recommended, to define features of many different types.

**Scope of improvement of network methods**

The prediction performance of the network methods depends on the extent to which the guilt-by-association principle holds. It has been shown that this “guilt- by-association” heuristic is universal and preserved beyond organism boundaries [7, 16]. However, in the case of the co-expression networks, it should be noted that the phenotypic variation is controlled at many levels, some of which are independent of transcript abundance. Future research could allow learning more about the scope of improvement of the network methods for protein function prediction. Another aspect is that the extent to which the guilt-by-association principle holds depends to a large extent on the quality of the data.

**AUC in one-class classification problems**

One aspect to be considered is that, the AUC may not be the best way to estimate the prediction performance in situations like PFP, where only examples from one class are known. This is because a gene in the test set whose “hidden” label is a "0", may have been predicted as positive and be actually positive [16]. Thus, we would falsely claim a false positive when, in fact, it is a true positive. Hameed et al., 2017 [16] proposed that recall and specificity are better accuracy measures in these situations. Notwithstanding, in

this thesis, we computed AUC because we do not expect 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 a direct comparison with other methods, since AUC is the most common choice to express the accuracy of prediction.

**Avenues of improvement of BMRF and PU-BMRF**

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 an improvement in accuracy of prediction as explained in [4,8]. 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).

The number of GO terms for which predictions are allowed could be increased, for instance, by allowing for more sparsity in the matrices that BMRF uses to estimate the predictions. In fact, probably the best avenue of improvement for PU-BMRF is to extend the problem to more than two classes. This would allow doing predictions within the set of unlabeled genes, by taking into account the neighbor information from the set of positives and from the set of negatives. In fact, if this could be made, the matrices would become less sparse, and predictions could be made for a lower number of positives.

Further aspects to be improved are, for instance, a more accurate extraction of RN, for instance, using Self-organizing maps [16], or any of the two techniques proposed by Youngs et al. (2014) [13]. Moreover, different magnitudes of co-expression could be taken into account, or neighbor information could be extended to indirect neighbors (so, neighbors of the neighbors).

**Further analysis**

Further analysis that could provide valuable information are: (a) Investigating whether PU contributes more in cases where the portion of unannotated genes is larger. For instance, by applying PU-BMRF on a better-annotated species. (b) Investigating whether, in the case of chickens, the ratio epp/tpepp increased because the portion of unannotated genes is larger for these species, or whether it increased because a lower co-expression threshold was used.

## Conclusions

The aim of this study was to investigate whether it was possible to achieve accurate protein function prediction (PFP) in poorly annotated species using network methods. We have shown that the accuracy of prediction was high (AUC: 0.72 with sd:0.08) when a Bayesian Markov Random Field (BMRF) was used for PFP in chickens. Moreover, we expanded upon the BMRF method and developed "PU-BMRF", a positive unlabeled learning (PU-learning) version of BMRF. The accuracy of prediction for chickens was significantly larger (p-value: 0.0049) when PU-BMRF was used instead of BMRF (AUC:0.758, sdAUC:0.084 versus AUC:0.706, sdAUC:0.026). The main advantage of PU-BMRF with respect to BMRF is that PU-BMRF can efficiently handle one class classification problems, such as PFP by extracting a set of reliable negatives (RN)

before the application of BMRF. We defined a total of 77 network features that can be used to extract RN with high accuracy. Moreover, in this thesis, we learned for which type of species, GO terms, co-expression thresholds and experiments, BMRF is more effective for PFP. Lastly, we gained some understanding about how the co-expression networks can be used to try to identify co-expression cascades.

## References


1. C. Weichenberger, A. Palermo, P. Pramstaller and F. Domingues, "Exploring Approaches for Detecting Protein Functional Similarity within an Orthology-based Framework", Scientific Reports, vol. 7, no. 1, 2017
2. K. Yan, D. Wang, J. Rozowsky, H. Zheng, C. Cheng and M. Gerstein, "OrthoClust: an orthology-based network framework for clustering data across multiple species", *Genome Biology*, vol. 15, no. 8, p. R100, 2014.
3. B. Schwikowski, P. Uetz and S. Fields, "A network of protein–protein interactions in yeast", *Nature Biotechnology*, vol. 18, no. 12, pp. 1257-1261, 2000.
4. Y. Kourmpetis, A. van Dijk, M. Bink, R. van Ham and C. ter Braak, "Bayesian Markov Random Field Analysis for Protein Function Prediction Based on Network Data", *PLoS ONE*, vol. 5, no. 2, p. e9293, 2010.
5. M. Deng, K. Zhang, S. Mehta, T. Chen and F. Sun, "Prediction of Protein Function Using Protein–Protein Interaction Data", *Journal of Computational Biology*, vol. 10, no. 6, pp. 947-960, 2003.
6. R. Sharan, I. Ulitsky and R. Shamir, "Network-based prediction of protein function", *Molecular Systems Biology*, vol. 3, 2007.
7. Stanley, N. Watson-Haigh, C. Cowled and R. Moore, "Genetic architecture of gene expression in the chicken", *BMC Genomics*, vol. 14, no. 1, p. 13, 2013.
8. Bargsten, E. Severing, J. Nap, G. Sanchez-Perez and A. van Dijk, "Biological process annotation of proteins across the plant kingdom", *Current Plant Biology*, vol. 1, pp. 73-82, 2014.
9. 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.
10. Radivojac P, Clark WT, Ronnen Oron T, et al. A large-scale evaluation of computational protein function prediction. Nature methods. 2013;10(3):221-227. doi:10.1038/nmeth.2340.
11. Bhardwaj, N., Gerstein, M. & Lu, H. “Genome-wide sequence-based prediction of peripheral proteins using a novel semi-supervised learning technique”, *Bmc Bioinformatics*, 11, 2010.
12. P. Yang, X. Li, J. Mei, C. Kwoh and S. Ng, "Positive-unlabeled learning for disease gene identification", *Bioinformatics*, vol. 28, no. 20, pp. 2640-2647, 2012.
13. N. Youngs, D. Penfold-Brown, R. Bonneau and D. Shasha, "Negative Example Selection for Protein Function Prediction: The NoGO Database", *PLoS Computational Biology*, vol. 10, no. 6, p. e1003644, 2014.
14. M. Jiang and J. Cao, "Positive-Unlabeled Learning for Pupylation Sites Prediction", *BioMed Research International*, vol. 2016, pp. 1-5, 2016.
15. X. Nan, L. Bao, X. Zhao, X. Zhao, A. Sangaiah, G. Wang and Z. Ma, "EPuL: An Enhanced Positive-Unlabeled Learning Algorithm for the Prediction of Pupylation Sites", 2017.
16. P. Hameed, K. Verspoor, S. Kusljic and S. Halgamuge, "Positive-Unlabeled Learning for inferring drug interactions based on heterogeneous attributes", *BMC Bioinformatics*, vol. 18, no. 1, 2017.
17. E. Serin, H. Nijveen, H. Hilhorst and W. Ligterink, "Learning from Co-expression Networks: Possibilities and Challenges", *Frontiers in Plant Science*, vol. 7, 2016.
18. R. Sharan, I. Ulitsky and R. Shamir, "Network-based prediction of protein function", *Molecular Systems Biology*, vol. 3, 2007.

# Appendices

# Appendix I – Concepts

## 1. Network elements

One network per GO term is defined. Thus, predictions are independent for each GO term
By predicting the function, we aim to report a possible association between the gene and a GO-term. More specifically, the Biological process (BP) category of GO.

- Target GO-term: the GO term for which we want to identify genes that are associated. Given the target GO term, we will investigate its possible association with each of the genes in the data set.

- Gene of interest: The gene whose association with the target GO term we are interested in at a specific moment.

- Nodes: genes or proteins. In this thesis, we work with genes, but sometimes the literature refers to proteins.

- label: a category that specifies whether a node is or not associated with the target GO term. In the network, the labels can be interpreted as the colors of the nodes. And in the script the labels are coded as follows:
  1: The gene has the function
  0: The gene is not known to have the function
  -1: The gene may or may not be known to have the function. If the function is known, we hide that information by placing a -1 instead of a 0 or a 1. Predictions are evaluated based on the fraction of genes that are -1 and whose label is known (test set).

- Edges: Two nodes are linked by an edge if they are co-expressed. The magnitude of the connection is neglected.

- Neighbors. Genes that are co-expressed or linked by an edge

## 2. Sets of genes

**(A) P**ositive, Non-validated-positives, Negatives, Reliable negatives, Unlabeled, Unknown

- Positive: a known-positive gene with validated proof, where positive means that its association with the target GO term has been validated with experimental evidence scores (EES). We only considered as positives those associations that regard the BP category.

- Non-validated-positives: the gene is associated with the target GO term but: (1) the

association has not been validated with EES, or (2) the association does not regard the BP GO-category

- Negatives: genes that are not associated with the GO terms. Since in biology the lack of evidence for an association does not imply that such an association does not exist, this set of genes cannot be known.

- Reliable negatives: set of genes that stand for representative set of genes that are not associated with the target GO-term (negative genes). Note that although they stand for representative set of negative genes, we cannot be certain of this. Therefore, more strictly speaking these are genes that are, to some extend, unlikely to be associated with the target GO-term

- Unlabeled: gene that is not known to be associated with the target GO term. This excludes the genes Non-validated-positives genes. The set of unlabeled includes: (1) genes that are associated with the target GO term but whose association is not known yet. These are the ones that we want to identify; (2) The reliable negatives that we aim to extract in the first step of PU-BMRF

- Unknown: a special case of unlabeled gene that is unlabeled for all the GO-terms in the database. Unknown genes are treated differently because we expect that these genes are not associated with any GO term in the database not because they are less functional than the other genes but because, for some reason, their function is more difficult to predict.

**(B)** training and test

- Train set: a set of genes whose label is either known or unknown, and that will enter the model with its label.

- Test: a set of genes whose function is known but hidden (see concept 'label')

## 3. Folds and replicates

- Folds, each fold consists of a set of test and a set of train. In total the test and the train set account for all the genes in the dataset. However the assignation of the genes to the train or test set changes with with k-fold in the crossvalidation. When all the folds are created, each gene has been assigned to the test set once and k-1 times to the train set. However, there are some genes that are never included in any test set, and remains always in the train set. These genes are the unknown genes and the Non-validated-positives genes (explained in the section 'Sets of genes' in this Appendix)

- Replicates, different runs of one analysis, each of which involves running k folds.

Since the folds require from random sampling, the replicates will also have a certain randomization.

## 4. Differences in network data

- #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 (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 term, 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 known 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 calculated as: n*(n-1)/2, where n is the number of genes that are associated with the GO Term.
- #epp/tpepp standardized. The ratio between #epp and tpepp standarized.
- *epp_(V): edges-positive-positive, (considering only the validated associations)*
- tpEpp(V): total possible of edges-positive-positive, (considering only validated associations)

## 5. Differences in annotation data

- #genes per GO term: average of the # genes associated with each GO term
- #GO terms per gene: average of the # GO terms that are associated with each GO term.
- #assoc*1000/total possible assoc: # of associations between genes and GO terms for a given 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.

## 6. GO properties

- epp/tpEpp: (edges positive-positive divided by total possible edges positive positive), described also in part 2, but here it is GO-term-specific rather than species-specific
- eppA/tpEppA: (edges positive-positive divided by total possible edges positive positive, including also NONvalid associations). As epp/tpEpp but includes also associations coded as “NONvalid” (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. More specifically, it is the Area Under a Receiver operating characteristic (ROC) Curve.
- sdAUC (standard deviation of AUC): mean of the standard deviation across replicates, for the GO term of interest.

## 7. Model parameters

- GO-size filter. This includes "minGOsize" and 'maxGOszie'. "minGOsize" and maxGOszie specifies which is the minimum and maximum number of genes that each GO term should be associated with. GO terms that do not satisfy the criteria are removed from the analysis.

- DF-size filter. This includes "minDFsize" and 'maxDFszie'. These specify which are the minimum and maximum numbers of genes that a domain should be associated with. Domains that do not satisfy the criteria are removed from the analysis.

- k-fold CV. The number of folds in the CV

- # iterations in the Gibbs-sampling

## 8. Other terms commonly used

- *"NONvalid": "other experiment scores, and/or GO-category different than BP.*
- *"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.*
- *minDFsize and maxGOszie specify the minimum and maximum number of genes that each domain should be associated with, respectively.*
- *GO-file: input file for BMRF. Contains the asscosiation between genes and GO terms*
- *EES (Experimental evidence scores): 'EXP', 'IDA', 'IEP', 'IMP', 'IPI', 'IGI'*

# Appendix II – Data

**Sources of the data used in this study**

**Yeast**

| | |
|---|---|
| Network file: | http://www.inetbio.org/yeastnet/downloadnetwork.php |
| GO file: | http://www.yeastgenome.org/download-data/curation |
| Domains file: | http://www.uniprot.org/docs/yeast |

**yeast_ppi**

| | |
|---|---|
| Network file: | /mnt/scratch/dijk097/Fernando/BMRF-R/ |
| GO file: | http://www.yeastgenome.org/download-data |
| Domains file: | http://www.uniprot.org/docs/yeast |

**Humans**

| | |
|---|---|
| Network file: | http://mostafavilab.stat.ubc.ca/gnat/ |
| GO file: | http://www.geneontology.org/page/download-annotations |
| Domains file: | http://www.uniprot.org/help/homo_sapiens |

**Chickens**

| | |
|---|---|
| Network file: | http://coxpresdb.jp/download.shtml |
| GO file: | http://www.geneontology.org/page/download-annotation |
| Domains file: | http://www.uniprot.org/help/homo_sapiens |

*Table 13: Main differences between the data available for the different species.*

| | | total data | validated | validated after filter | Portion of data that is validated and passes the filter |
|---|---|---|---|---|---|
| #GO | yeast ppi | 8,680 | 4,723 | 1,073 | 12.36 |
| | yeast | 8,680 | 4,723 | 1,104 | 12.72 |
| | humans | 19,549 | 10,271 | 1,982 | 10.14 |
| | Chickens_07 | 9,247 | 877 | 9 | 0.10 |
| | Chickens_035 | 16,205 | 2,350 | 142 | 0.88 |
| #genes | yeast ppi | 5,437 | 4,488 | 4,168 | 76.66 |
| | yeast | 5,760 | 4,488 | 4,453 | 77.31 |
| | humans | 9,998 | 5,582 | 5,535 | 55.36 |
| | Chickens_07 | 2,152 | 53 | 53 | 2.46 |
| | Chickens_035 | 12,424 | 300 | 296 | 2.38 |
| #edges | yeast ppi | 474,389 | 227,420 | 98,192 | 20.70 |
| | yeast | 474,389 | 227,420 | 104,303 | 21.99 |
| | humans | 1,213,376 | 410,215 | 219,796 | 18.11 |
| | Chickens_07 | 181,735 | 2,253 | 263 | 0.14 |
| | Chickens_035 | 734,840 | 14,733 | 7,892 | 1.07 |

*ppi: protein-protein-interaction #assoc: # of associations between GO terms and labels; Chickens_07 and Chickens_05: Network data for Chicken when the Pearson correlation was 0.7 and 0.5, respectively.*

In Table 5, we observed:

- The network is considerably smaller for chickens.
- Validated data for chickens_0.7 and chickens_0.35 is very poor in comparison to yeast and humans. In the case of chickens_0.35, less than 1% of the #associations are validated and pass the GO-size filter, whereas in humans it is 10.14%.
- For yeast, co-expression data is slightly more complete than ppi data.
- Total data for humans is larger than for yeast but the proportion of validated data that passes the filter is lower (18% in humans vs 22% in yeast-co-expression in the case of #associations).
- The number of edges is large for chickens when a Pearson Correlation of 0.35 was used (734,840). However, the number of edges is still around half that in humans (1,213,376)
- For Pearson correlation equal to 0.07, the number of association was around 1/5

smaller than for Pearson correlation 0.05.

# Appendix III – Additional results

## 1. Part 1- Tuning of model parameters and choice of the co-expression data

### Choice of the co-expression data

Conventionally, a Pearson Correlation of 0.7 is used as a threshold for co-expression analysis. However, in the case of Chickens, using a Pearson correlation of 0.7, led to an excessively low number of validated associations. Subsequently, only 9 GO term passed the GO-size filter used by BMRF (see Appendix I-Concepts). Furthermore, we observed that the number of GO terms that passed the filter did not improve significantly when the GO-size filter was adjusted to include also GO terms with a low number of associated genes (for minGOsize=0.8, 52 GO passed the GO-size filters). We, therefore, investigated whether by lowering the Pearson correlation threshold we obtained better prediction performance. We chose different co-expression threshold values and we computed AUC, the standard deviation across replicates and the number of genes in the network.

A Pearson correlation of 0.35 seemed to lead to the highest prediction performance (Figure 9). Details about how the co-expression threshold affects the data and the reproducibility of the results are given under the title "Impact of the quality of the data" in this Appendix and in Results.

### Tuning of model parameters

In this section, we investigated the impact on the prediction performance of the number of replicates of analysis, the GO-size filters, the number of *k-folds* in the k-fold validation and the number of iterations in the Gibbs-sampling used by BMRF. Also, we investigated the effect of adding non-validated data and domain information.

- Number of replicates

One extra analysis showed that the standard deviation across five runs of 20 replicates was slightly lower for a GO-size filter of (20,0.1) than for (5,0.9): 0.008 vs 0.01, respectively. This makes sense since the standard deviation is slightly larger for those GO terms with fewer genes and more of these GO terms were considered in the analysis when the GO-size filter was (5,0.9) (less strict).

- GO-size filter

The default value for maxGOsize was 0.9, however, for in this thesis, we are interested in the most specific GO terms, and we chose value 0.1. A preliminary analysis carried on yeast co-expression data showed that there was not significant increase in the accuracy of prediction when 0.1 was used instead of 0.9. In this analysis, AUC was 0.779 with a minGOsize of 0.1

and 0.775 for 0.9. The standard deviations were 0.08 in both cases. The same analysis showed how the data changed with the GO-size filter (Table 14).

*Table 14: Impact of the GO-file filter on the data*

| scenarios | | | data | | | |
|---|---|---|---|---|---|---|
| **scenario name** | **Min GO-size** | **Max GO-size** | **Network size (#conn)** | **#unkown genes*** | **#assoc.** | **#GO-terms** |
| normal | 20 | 0.1 | 598,174 | 655 | 132,249 | 1,104 |
| default** | 20 | 0.9 | 598,174 | 4 | 264,279 | 1,187 |
| more GO-terms | 10 | 0.9 | 598,174 | 4 | 273,977 | 1,738 |
| Only large GO-terms | 30 | 0.07 | 598,174 | 688 | 104,582 | 832 |

Then, we investigated the effect of the GO-size filter in the 3 species considered and yeast_ppi. For this, we carried the analysis in three scenarios with different GO-size filters. We called “normal scenario” to the analysis in which the minGOsize was 0.1 and minGOsize was 20. We considered that by changing the minGOsize to 9, we were adding to the analysis more specific GO-terms and by removing the filter on maxGOsize (maxGOsize=1), we were allowing for more general GO terms.

*Table 15: Impact of GO-size filter in the prediction performance*

| | # GO-terms | | |
|---|---|---|---|
| **scenario** | **MinGOsize:9 maxGOsize:0.1** | **MinGOsize:20 maxGOsize=0.1** | **MinGOsize:20 maxGOsize=1** |
| **Description scenario** | **Adding more specific GO-terms** | **Normal** | **Adding more general GO-terms** |
| **humans** | 3328 | 1982 | 2069 |
| **Chickens0.35** | 307 | 138 | 138 |
| **yeast** | 1772 | 1104 | 1187 |
| **yeast PPI** | 1734 | 1057 | 1153 |

In Table 15, we observed that the number of GO-terms after passing the filter was still low for chickens (307). Due to time constrains we carried the analysis for the 138 GO terms in chickens when the GO-size filter was (20,0.1). The analysis, however, could be extended to 307 GO terms if the filter was changed to (8,0.1).

*Table 16: Impact of GO-size filter in the prediction performance*

| | Average AUC (sd) | | |
|---|---|---|---|
| **scenario** | **MinGOsize:9 maxGOsize:0.1** | **MinGOsize:20 maxGOsize=0.1** | **MinGOsize:20 maxGOsize=1** |
| **Description scenario** | **Adding more specific GO-terms** | **Normal** | **Adding more general GO-terms** |
| **humans** | 0.701 (0.017) | 0.701 (0.017) | 0.703 (0.017) |
| **Chickens0.35** | 0.726 (0.08) | 0.726(0.08) | 0.726 (0.08) |
| **yeast** | 0.757 (0.023) | 0.764 (0.016) | 0.761 (0.015) |
| **yeast PPI** | 0.707 (0.028) | 0.714 (0.02) | 0.712 (0.018) |

In Table 16, we observed that in the case of humans and chickens, the effect of the GO-size filter on the prediction performance was almost null (for the GO-size filters considered). For yeast and yeast PPI, the impact was very small, leading to slightly better performance when the filter was “normal”. Intuitively, we would expect a better performance in the scenario “Including more General GO-terms”. However, we did not observe this. This could be regarded as an indiator that the relationship between the specificity of the GO-term and the prediction performance is not linear.

The increase in AUC observed in Figure 16 may be due to subseting the of GO terms. For instance, more strict filters allow for less GO terms passing the filter. An additional analysis, however, showed that the prediction of individual GO terms was not affected by the GO-size filter. We applied BMRF on human data with minGOsize:20, 150 and 400, and we computed AUC only on those GO terms that were predicted with the three filters. The mean AUC (and sd) were: 0.693 (0.05), 0.693 (0.05) and 0.692(0.05), respectively, indicating that the prediction performance of a GO term was independent of the number of GO terms considered.

- Number of folds in k-validation

The number of folds in the k-validation is an important model parameter because the association data is highly unbalanced. For each GO term, the number of unlabeled genes is much larger than the number of labeled genes, and therefore low values of k-fold CV may result in an inadequate use of the train set (using less labeled genes than are actually available), and a high values of k-fold CV may result in an inaccurate prediction performance because AUC may be estimated based on an excessively low number of labeled genes (in the test set). Table 16 showed the results when we carried analysis using different values of k-fold CV.

| | Average AUC (sd) | | | |
|---|---|---|---|---|
| **K-fold** | **2** | **5** | **10** | **20** |
| **humans** | 0.667 (0.034) | 0.694 (0.021) | 0.701 (0.017) | 0.705 (0.01) |
| **Chickens0.35** | 0.7 (0.083) | 0.718 (0.082) | 0.726(0.08) | 0.75 (0.066) |

*Table 17: Impact of the number of folds in the prediction performance.*

Table 17 showed that in human data k:10 is sufficient to achieve the highest possible prediction performance, whereas in chickens, the prediction performance using k:20 instead of k:10 is slightly larger. This not surprising since in chickens the number of labeled genes per GO term is much lower than in humans, and a higher value of k is translated in train set with more labeled genes and better prediction performance. In humans, however, the train set seems to have already a large number of positive cases when k is equal to 10. We will choose the value 10 for the model parameter 'k', because larger values imply that the number of positive cases in the training set may not be sufficient (at least for some of the folds), and the estimates of the accuracy of prediction become less reproducible.

- Number of iteration in Gibb-sampling

In BMRF, Gibb sampling is used to estimate the label of the unknown genes (a definition of unknown genes is given in Appendix I-Concepts. We investigated whether more iterations are required when the number of unknown genes was large (default value of GS is 30 iterations). We carried analysis when the number of unknown genes was above 3000 (this was achieved with a minGOsize of 400), and we estimated AUC when GS was 30 and 500. We observed that the mean AUC and the mean standard deviation across replicates were near identical in both analysis. We, therefore, concluded that increasing the number of GS iterations is not helpful when the number of unknown genes becomes very large. We thus chose a value for the number of GS iterations 30 in all analysis.

- Non-validated data and domain information

We investigated the effect of adding domain information and non-validated GOterm-gene associations in the analysis. Table 18 showed the results in three scenarios defined based on the information used.

*Table 18: Impact of domain info. and non-valid data on the prediction performance.*

| | **Average AUC (sd)** | | |
|---|---|---|---|
| **scenario** | **Not including information** | **Including domain information** | **Including both, domain info. and non-valid info. (normal approach)** |
| humans | 0.654 (0.027) | 0.701 (0.017) | 0.705 (0.017) |
| Chickens0.35 | 0.57 (0.068) | 0.724 (0.08) | 0.726(0.08) |
| yeast | 0.773 (0.093) | 0.792 (0.089) | 0.764 (0.016) |
| yeast PPI | 0.73 (0.103) | 0.747 (0.0992) | 0.714 (0.02) |

We expected an increasing in AUC from the left hand side of Table 18 to the right (as more information was included in the analysis), however, in the case of yeast and yeast PPI, we observed that the best prediction performance when the domain information was included but not the non-valid info. Furthermore, predictions were better when none of the sources of information was included (left) than when both sources were considered. This is a clear indicator that in the case of yeast, and yeast_ppi, the non-valid information worsens the prediction performance. A possible explanation for this is that in the case of yeast, a very large portion of the gene-function associations are validated and therefore including the non validated information increases the noise without a corresponding increase in the accuracy of prediction.

In the case of chickens and humans, the domain information was more relevant than for yeast and yeast_ppi, accounting for roughly 5% higher AUC. Whereas the non-valid information slightly helped in chickens and humans. An explanation could be that in the absence of enough validated data for humans and chickens, adding non-validated information may add noise but it also improved the resources in the network method.

## 2. Part 2- Impact of the data on the prediction performance

### 2.a) Differences in prediction performance between species

By comparing the characteristics of the network in the different species and the prediction performance, we gained some understanding on which network characteristics are more relevant for protein function prediction via BMRF. In order to have more cases and species to compare, in this section, we carried the analysis with the chicken data when the co-expression threshold was 0.7 (common threshold), in addition to 0.35 (chosen threshold).

It is expected that the total number of edges of the network may be of limited importance for PFP. It may be that most of these edges are connecting genes that are not known to have the function or genes that are known to have a given function with genes that are not known to have the same function. A more important network parameter, therefore, may be, for instance, '#epp' (edges of positive-positive). We compared the #te, #epp, #epn and #enn (Appendix I-Concepts) for the different species and we studied the relationship between these parameters and the prediction performance (AUC - Area under the curve). Tables 18 and 19 summarize the main differences in the characteristics of the network of the different species and the prediction performance. Note that here, #epp, for instance, refers to the sum of all the epp of all the GO terms in that particular species. And the same applies to #epn, #enn and #te.

*Table 19: Relationship between #edges and AUC*

| | #te | #epp (epp*100/te) | #epn (epn*100/te) | #enn (enn*100/te) | AUC |
|---|---|---|---|---|---|
| yeast ppi | 401,820 | 264,347 (65.79) | 123,152 (30.65) | 14,321 (3.56) | 0.734 |
| yeast | 598,174 | 382,450 (63.94) | 186,722 (31.22) | 29,002 (4.85) | 0.775 |
| humans | 1,548,622 | 481,792 (31.11) | 754,276 (48.71) | 312,554 (20.18) | 0.712 |
| Chicken_07 | 100,764 | 24 (0.02) | 2,232 (2.22) | 98,508 (97.76) | 0.728 |
| Chicken_035 | 2,094,870 | 576 (0.03) | 51,610 (2.46) | 2,042,684 (97.51) | 0.762 |

*#te: total number of edges, epp: edges positive-positive. epn: edges positive-negative. enn: edges negative-negative*

Based on Table 19, we concluded that #te, #epp, #epn and #enn may be of limited importance for the prediction performance. We would expect that epp*100/te would be more directly related to the prediction performance than #epp because #te is the sum of #epp, #enn and #epn and, in principle, the prediction will be more difficult when #epn and #enn are larger. This is because #enn and #epn may be connecting positives and negatives genes, and BMRF will fail to classify the positives and negatives genes. In Table 19 we observed that epp*100/te might be playing a role in the prediction accuracy, because in yeast, epp*100/te it was higher than for humans and so it is the prediction performance. However, in the case of chickens, the epp*100/te was very low and predictions performance was still high. Therefore, we concluded that the epp100/te is also not informative of the prediction performance.

We then studied the degree of connections between the genes of a given GO terms, in the different species. One way to do this is by comparing the portion of epp concerning the total possible number of epp (tpepp). Tpepp is a constant different for each GO term and refers to

the total number of edges if all the genes associated with the GO term were interconnected. Tpepp is calculated as: n*(n-1)/2, where n is the number of genes that are associated with the GO Term.

*Table 20: Ratio epp/tpepp standarized*

*epp: edges positive-positive; tpepp: total possible epp*

| | epp*1000/tpepp | epp/tpepp*1000 corrected by epp and standarized | AUC |
|---|---|---|---|
| yeast ppi | 47.88 | -0.449 | 0.734 |
| yeast | 63.37 | -0.449 | 0.775 |
| humans | 38.63 | -0.449 | 0.712 |
| Chickens_07 | 210.56 | 1.789 | 0.728 |
| Chickens_035 | 28.15 | -0.442 | 0.762 |

*AUC: area under the curve. Mean AUC of all GO terms that pass the filter considering only validated associations between the GO term and genes.*

From Table 20, we learn that with the exception of chickens data, there seemed to be a favorable relation between epp/tpepp and AUC. For chickens_07, epp/tpepp is considerably larger than for the other cases. A possible explanation is that for chickens_07, the quality of the network data is very high and epn are less common. AUC, nevertheless, is not larger for chickens_07. Thus, we concluded that epp/tpepp is not a direct indicator of the prediction performance.

Then, we investigated the relationship between the level of annotation for one species (this includes the average of genes per GO-term and average GO-terms per gene), the #edges per gene and the #epp per GO-term (Appendix I-Concepts), with the prediction performance.

*Table 21: Differences between the network data of the different species*

| | mean (sd) | | | | AUC |
|---|---|---|---|---|---|
| | labels/go | go/labels | edges/label | Epp/GO | |
| yeast_ppi | 11.31 (46.29) | 17.05 (22.67) | 156.39 (179.3) | 960.12 (10844.65) | 0.734 |
| yeast | 12.01 (48.78) | 18.12 (22.77) | 213.7 (146.61) | 1311.15 (15209.25) | 0.775 |
| humans | 11.24(59.85) | 25.63 (42.51) | 310.36 (80.50) | 954.16 (11417.71) | 0.712 |
| Chickens_07 | 0.28 (0.97) | 0.12(0.85) | 43.02(49.12) | 0.14642(1.199437) | 0.728 |
| Chickens_035 | 24.55 (26.14) | 0.87 (6.23) | 811.29 (1097.3) | 6778.91 (110825.9) | 0.762 |

Co-expression data has higher in #GO/labels and #edges/label. Since we achieve a higher overall AUC for yeast, we can expect that #labels per GO and #epp/GO are more directly affecting the AUC. Further, we could expect that to achieve higher AUC (>0.75), data should have a large #labels/GO (~12), and ~1000 epp/GO. We observe that co-expression data for chickens_07 is currently far from these numbers. However, when we used chicken_035 data "#labels per GO", #edges/label and "Epp/GO" increased to levels even higher than for the other species (#go/labels reminds much lower than for the other species). It is therefore not surprising that the mean AUC is higher for chickens_035 than for yeast_ppi and humans.

In figures 21 and 22, we show the distribution of AUC for the 20 GO terms that were common in the four species. Figure 23 shows the portion of GO term for which AUC was above certain values.

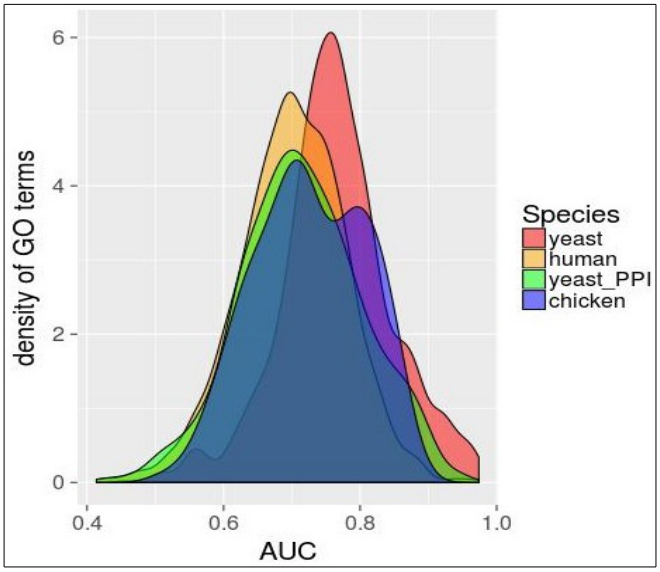



*Figure 21: Distribution of the AUC for the 20 GO terms that are common in the four species.*

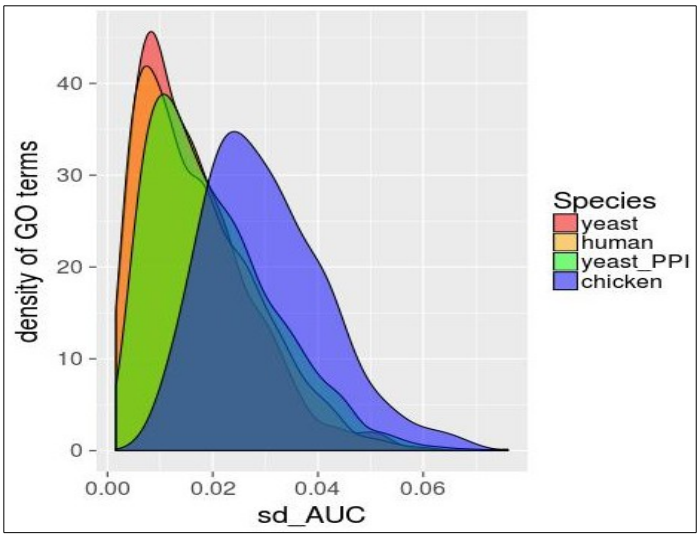



*Figure 22: Distribution of the standard deviation of AUC(sd) for the common GO terms*

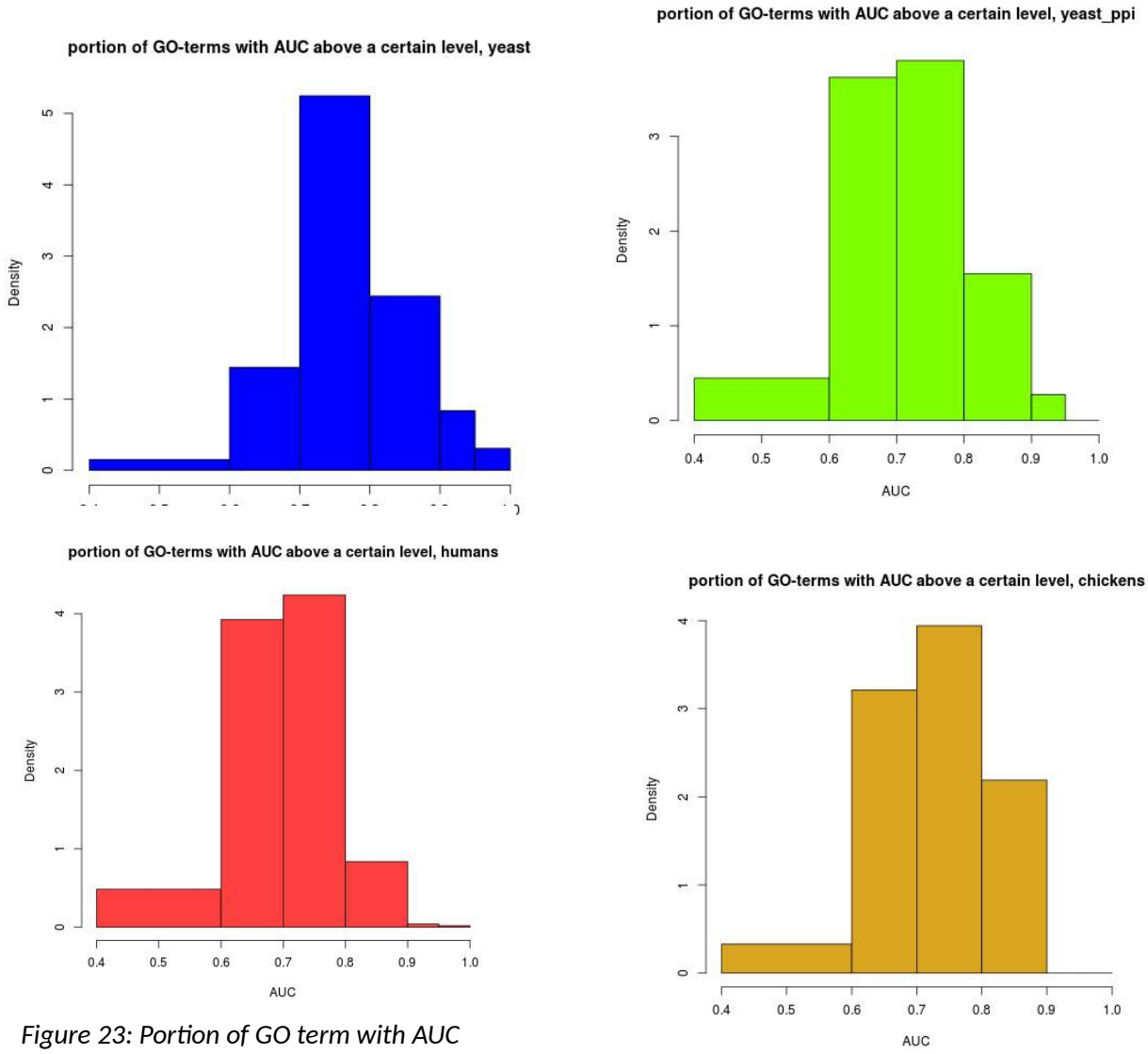



*Figure 23: Portion of GO term with AUC above a certain value*

**Section 2 B) Impact of the quality of data on the prediction performance**

In this section we investigated how the predictions vary when the data becomes more incomplete. This information can be used to get some insights on to which extend BMRF can be used for poorly annotated species. For this, we investigate how performance is altered in different situations. Using yeast data we investigated the effect of the network size on the prediction performance and then with humans data we investigated the effect of decreasing the quality of the data by removing edges of a specific class (i.e. epp, epn and enn) and associations between GO-terms and genes.

Using chicken data, we investigated how the number of genes in the network changes with the Pearson correlation threshold (Figure 24).

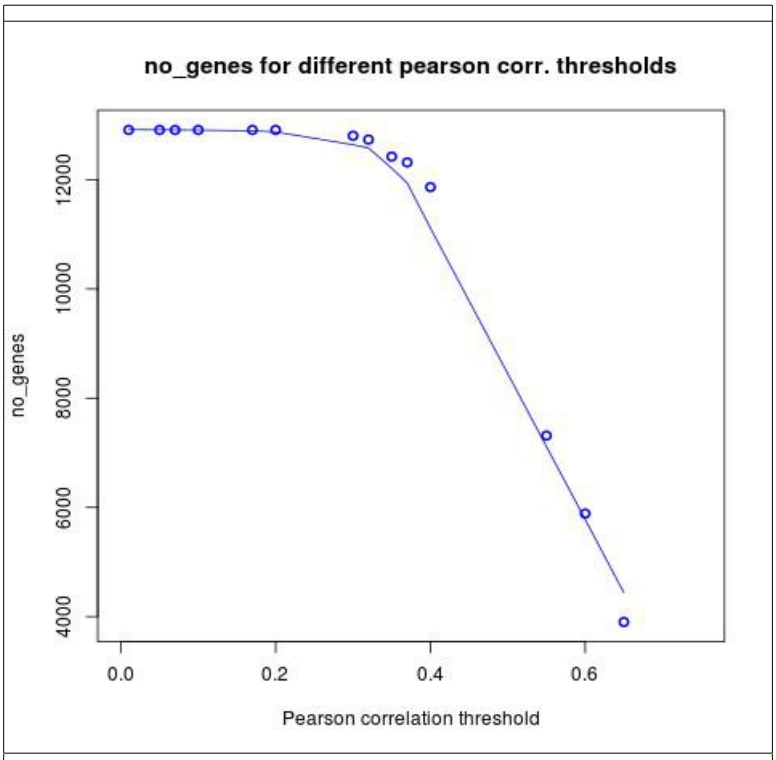



*Figure 24: Number of genes in the network for different co-expression thresholds.*

*Values corresponding to the plot are in Table 1.*

In Figure 24, we observed a decrease in the number of genes as the threshold for co-expression became more strict. The decrease became more sharp when the threshold was above 0.4. Nevertheless, it should be considered that this may depend on the characteristics of the co-expression analysis.

*Table 22: Prediction performance choosing different Pearson correlation thresholds.*

| Pearson corr. Threshold | #GO terms pass normal filter | #genes in network | AUC | sd AUC | Mean of SD across replicates |
|---|---|---|---|---|---|
| 0.7 | 9 | 2,784 | 0.714 | 0.065 | 0.022 |
| 0.65 | 21 | 3,901 | 0.603 | 0.064 | 0.022 |
| 0.6 | 33 | 5,887 | 0.660 | 0.055 | 0.019 |
| 0.55 | 54 | 7,314 | 0.668 | 0.063 | 0.022 |
| 0.4 | 134 | 11,866 | 0.695 | 0.056 | 0.025 |
| 0.37 | 140 | 12,317 | 0.704 | 0.056 | 0.027 |
| **0.35** | **138** | **12,424** | **0.726** | **0.080** | **0.030** |
| 0.32 | 148 | 12,735 | 0.700 | 0.052 | 0.028 |
| 0.3 | 148 | 12,805 | 0.689 | 0.054 | 0.030 |
| 0.2 | 150 | 12,912 | 0.700 | 0.068 | 0.046 |
| 0.1 | 150 | 12,912 | 0.684 | 0.068 | 0.042 |
| 0.07 | 150 | 12,912 | 0.688 | 0.075 | 0.035 |
| 0.05 | 150 | 12,912 | 0.684 | 0.074 | 0.036 |
| 0.01 | 150 | 12,912 | 0.684 | 0.078 | 0.039 |

*Data: chickens*

In Table 22, we observed that the predictions were higher when a Pearson correlation threshold of 0.35 was chosen. However, It should be considered that this value depends on the characteristics of the co-expression analysis and the species considered. For instance, in species with a large number of validated data, it is expected that the best possible threshold is higher, given that overall quality of the data is higher. However, in a species in which the portion of validated data is low (as is the case for chickens), including data of relative quality may be helpful. In other words, the quality of the data may become more important once the criteria for minimum of data has been satisfied.

**Using yeast data**

We randomly extracted from the network a known percentage of the edges and calculated the prediction performance. Table 20 shows the results of this analysis.

In Table 23 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 may be caused by the fact that less GO terms were considered in the analysis. This is because in BMRF the GO-term annotation is pruned based on the network.

*Table 23:* Impact of the extraction of edges in prediction performance for individual GO terms

| | | | portion of edges substracted | | | | | |
|---|---|---|---|---|---|---|---|---|
| GO-term | total_labels | valid_labels | **0% (all data used)** | **10%** | **30%** | **50%** | **90%** | **95%** |
| GO:0042981 | 30 | 30 | 0.741 | 0.726 | 0.734 | 0.778 | 0.685 | 0.699 |
| GO:0014068 | 30 | 30 | 0.48 | 0.479 | 0.514 | 0.495 | 0.504 | 0.493 |
| GO:0045931 | 31 | 25 | 0.649 | 0.628 | 0.632 | 0.63 | 0.665 | 0.682 |
| GO:0000209 | 36 | 23 | 0.862 | 0.872 | 0.773 | 0.837 | 0.857 | 0.837 |
| GO:0006664 | 39 | 32 | 0.844 | 0.853 | 0.837 | 0.821 | 0.77 | 0.775 |
| GO:0031670 | 61* | 50* | 0.811 | 0.789 | 0.796 | 0.789 | 0.796 | 0.79 |
| GO:0036503 | 62* | 49* | 0.855 | 0.844 | 0.85 | 0.827 | 0.803 | 0.819 |
| GO:0006414 | 65* | 40 | 0.756 | 0.752 | 0.755 | 0.757 | 0.733 | 0.714 |
| GO:0006417 | 144 | 100* | 0.728 | 0.732 | 0.741 | 0.745 | 0.73 | 0.731 |
| GO:0044270 | 195* | 166* | 0.714 | 0.703 | 0.701 | 0.705 | 0.644 | 0.649 |

*Data: yeast*

**Humans data**

We also carried analysis on human data to investigate further how the removal of data relates to an increment in AUC. We considered, two types of removal of data:

Removal of edges: 3 types of edges were considered: epp (edges-positive-positive), epn(edges positive negative), enn (edges negative negative). We removed from the data 5, 10, 30, 50, 70, 90, 99% of each of these groups of associations and we computed the correlation between the portion that was removed and AUC. Therefore, we got three correlation estimates ('reduceEpp', 'reduceEpn', and 'reuduceEnn').

Removal of associations: 2 types of associations were considered: associations between the target GO term and the validated genes, and associations between other GO terms and their validated-labeled genes. We removed from the data 5, 10, 30, 50, 70, 90, 99% of each of these groups of associations and we computed the correlation between the portion that was removed and AUC. Therefore, we got two correlation estimates ('reduceAmg' and 'reduceOa').

The correlation estimates calculated in these analysis are given in Table 24.

Removing associations from other GO terms is expected to affect the prediction performance because more genes will be considered as 'unknown' (genes associated with 0 GO terms), and the label of these will be estimated through Gibbs-sampling.

*Table 24: Correlation between AUC and data quality.*

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

*Data: humans*

From Table 24 we observed:

- AUC increased linearly as we removed epn from the data.
- AUC also increased almost linearly as we removed enn. This also makes sense since we would expect that some of the enn are actually connecting negative genes with positive genes.
- AUC also increased by removing associations from the target GO term. A possible explanation for this is that α will be lower in Equation 3, and therefore less genes will be classified as positive. Since a very low portion of the genes are true positives, AUC increased.
- AUC however decreased when we removed annotations between genes and other GO terms. A possible explanation of this is that some of these genes will enter the category of "unknown" and their labeling will be initialized through Gibb-sampling. This may increase the margin error.
- Lastly, as expected, AUC decreased when some epp were removed from the data.

**3C. Impact of the nature of the networks on the prediction performance**

By nature of the network here we refer to the characteristics of the co-expression analysis. It is important to investigate whether, for instance, a co-expression analysis addressed to one specific tissue allows to make more accurate predictions for those GO terms whose function is more relevant in that tissue. Note that, from a biological perspective, and considering that network analysis exploits the principle of guilt-by-association, we would expect that the nature of the network has an impact on the prediction performance.

**Using yeast data**

Using yeast data, we investigated whether the nature of the co-expression network (characteristics of the experiment) have any impact on the prediction performance. For this, we choose five different subsets of the network. In the first subset, we included the co-expression data from all experiments refereed to stress, in the second subset we considered

all co-expression analysis involving oxidation. Then, since the size of the network differs between these subsets, we choose other subsets of experiments of a controlled size. Subsets 3 and 4 are of similar size and refer the names of the experiments are “Sodium arsenate response of wild-type and slt2 deficient cells” and “Integration of the general amino acid control and nitrogen regulatory pathways in yeast nitrogen assimilation”, respectively. The last subset of network corresponds to an experiment with a very low number of co-expressed genes. The name of the experiment is “The metabolic response to iron deficiency in Saccharomyces cerevisiae”, under the data source indicated in Appendix II.

*Table 25: Impact of the nature of the network on the prediction performance*

| **scenarios** | **data** | | | | | |
|---|---|---|---|---|---|---|
| **scenario name** | **Network size (#edges)** | **#unknown genes*** | **#proteins.** | **#assoc.** | **# GO-terms** | **AUC mean(sd) [median]** |
| “stress” co-expression | 98,479 | 471 | 4,879 | 110,682 | 1,021 | 0.727 (0.089) [0.723] |
| “oxidiation” co-expression | 64,167 | 499 | 4,923 | 111,480 | 1,022 | 0.72 (0.086) [0.714] |
| similar_size_ network Experiment1 | 28,800 | 298 | 1,865 | 32,336 | 426 | 0.684 (0.101) [0.677] |
| similar_size_ network Experiment1 | 27,488 | 255 | 2,899 | 58,358 | 681 | 0.682 (0.089) [0.687] |
| very small network | 7,073 | 112 | 661 | 8,862 | 203 | 0.635 (0.113) [0.614] |

*Data: yeast*

In Table 25, we observed that the nature of the network does not seem to have a strong impact on the prediction performance, even if the network size and other parameters ()#unknown genes, #proteins, #edges and #GOs) differed between the different subsets of data.

**Using human data**

In order to investigate whether there was biological support in the data, we identified the GO terms for which a highest AUC was achieved using network data from different tissues. For a fairer analysis, we normalized the networks of the different tissues based on epp/tpepp. Table26 shows a list of the GO terms that were more accurately predicted using networks based on co-expression analysis in different tissues, and Table 27 shows a list of the Tissues for which the co-expression analysis led to better and worse accuracy of prediction, for 10 randomly chose GO terms.

*Table 26: List of GO terms that were more accurately predicted for each tissue*

| tissue | top1_GOterm | top2_GOterm |
|---|---|---|
| Stomach | post-Golgi vesicle-mediated transport | positive regulation of lipid transport |
| Esophagus-Muscularis | anoikis | intrinsic apoptotic signaling pathway in response to oxidative stress |
| Thyroid | erythrocyte differentiation | cell aging |
| Whole_Blood | negative regulation of epithelial cell migration | keratinocyte proliferation |
| Brain-Amygdala | histone H4 acetylation | protein destabilization |
| Adrenal_Gland | regulation of protein oligomerization | negative regulation of response to biotic stimulus |
| Brain-Putamen(basal_ganglia) | regulation of protein complex disassembly | negative regulation of protein binding |
| Brain-Cortex | receptor internalization | regulation of heart rate |
| Skin-Not_Sun_Exposed(Suprapubic) | regulation of toll-like receptor signaling pathway | positive regulation of proteasomal ubiquitin-dependent protein catabolic process |
| Testis | positive regulation of viral genome replication | negative regulation of telomere maintenance |
| Brain-Anterior_cingulate_cortex(BA24) | positive regulation of viral genome replication | peroxisome organization |
| Pancreas | regulation of receptor internalization | TOR signaling |
| Brain-Spinal_cord(cervical_c-1) | regulation of receptor internalization | regulation of microtubule polymerization |
| Brain-Hypothalamus | negative regulation of DNA binding | positive regulation of telomere maintenance |
| Brain-Caudate(basal_ganglia) | negative regulation of dephosphorylation | cellular extravasation |
| Artery-Tibial | regulation of cell adhesion mediated by integrin | negative regulation of telomere maintenance |
| Pituitary | negative regulation of blood vessel endothelial cell migration | protein localization to cytoskeleton |
| Esophagus-Mucosa | negative regulation of cell projection organization | response to temperature stimulus |
| Lung | intrinsic apoptotic signaling pathway in response to oxidative stress | histone deacetylation |
| Skin-Sun_Exposed(Lower_leg) | regulation of interferon-beta production | myeloid cell homeostasis |
| Nerve-Tibial | negative regulation of cell-substrate adhesion | anoikis |
| Muscle-Skeletal | homotypic cell-cell adhesion | regulation of cell adhesion mediated by integrin |
| Breast-Mammary_Tissue | receptor internalization | regulation of protein complex disassembly |
| Brain-Nucleus_accumbens(basal_ganglia) | negative regulation of epithelial cell migration | positive regulation of DNA binding |
| Adipose-Subcutaneous | regulation of protein oligomerization | negative regulation of blood vessel endothelial cell migration |
| Heart-Atrial_Appendage | positive regulation of macroautophagy | negative regulation of blood vessel endothelial cell migration |
| Adipose-Visceral(Omentum) | regulation of cell adhesion mediated by integrin | regulation of smooth muscle cell migration |
| Artery-Aorta | positive regulation of actin filament bundle assembly | cellular response to amino acid starvation |
| Brain-Substantia_nigra | homotypic cell-cell adhesion | regulation of epithelial to mesenchymal transition |
| Heart-Left_Ventricle | regulation of DNA recombination | regulation of sodium ion transport |
| Brain-Hippocampus | interleukin-10 production | histone ubiquitination |
| Brain-Cerebellar_Hemisphere | lipid storage | smooth muscle cell migration |
| Colon-Transverse | regulation of cell adhesion mediated by integrin | positive regulation of proteasomal ubiquitin-dependent protein catabolic process |
| Brain-Cerebellum | peroxisome organization | ATP-dependent chromatin remodeling |
| Brain-Frontal_Cortex(BA9) | regulation of phosphatase activity | cell aging |

*Table 27: Tissues for which the co-expression analysis led to better and worse AUC.*

| GOterm | tissue_highest_AUC | highest_AUC | tissue_loest_AUC | lowest_AUC |
|---|---|---|---|---|
| regulation of receptor internalization | Pituitary | 0.586 | Pancreas | 0.459 |
| peroxisome organization | Brain-Caudate(basal_ganglia) | 0.734 | Brain-Anterior_cingulate_cortex(BA24) | 0.612 |
| mitotic cytokinesis | Adipose-Subcutaneous | 0.783 | Testis | 0.665 |
| post-Golgi vesicle-mediated transport | Testis | 0.758 | Stomach | 0.644 |
| regulation of DNA recombination | Brain-Hippocampus | 0.806 | Heart-Left_Ventricle | 0.693 |
| histone ubiquitination | Nerve-Tibial | 0.74 | Brain-Hippocampus | 0.628 |
| negative regulation of response to biotic stimulus | Brain-Anterior_cingulate_cortex(BA24) | 0.695 | Brain-Hippocampus | 0.585 |
| negative regulation of epithelial cell migration | Brain-Frontal_Cortex(BA9) | 0.626 | Brain-Nucleus_accumbens(basal_ganglia) | 0.517 |
| erythrocyte differentiation | Muscle-Skeletal | 0.683 | Thyroid | 0.577 |

*Results for 10 randomly chose GO terms*

We hypothesized that, as long as there is enough data, the accuracy of prediction will depend more on the properties of the GO term, than on the quality of the data. Therefore, in order to improve PFP, the development of method that can improve the GO-term-properties, like positive unlabeled learning (PU-learning), that can improve the epp/tpepp ratio, may be more helpful than increasing the number of co-expression experiments.

## 3. Part 3. Differences in prediction performance between GO-terms

We defined nine GO-term-properties including epp/tpEpp, eppV/tpEppV, #genesV, sepc, teV/tpeV, depth, AUC and sdAUC. Definitions of these are in Appendix I-Concepts. Tables 24-27 show the correlation between these properties for yeast, humans , yeast_PPI and chickens, respectively. Only Significant correlations (p-value<0.05) were shown. Note that Tables 28-31 are a detailed version of the correlation shown in Table 6 of results. Although in results only the correlations with AUC were shown.

*Table 28: Correlations between GO-properties, yeast*

| Var1 | Var2 | corr | p_value |
|---|---|---|---|
| epp/tpEpp | epp_V/tpEppV | 0.968 | 0 |
| #genesV | spec | -0.875 | 0 |
| epp_V/tpEppV | teV/tpeV | 0.834 | 0 |
| epp/tpEpp | teV/tpeV | 0.83 | 0 |
| AUC | epp_V/tpEppV | 0.62 | 0 |
| #genesV | sdAUC | -0.591 | 0 |
| AUC | epp/tpEpp | 0.582 | 0 |
| sdAUC | spec | 0.497 | 0 |
| AUC | sdAUC | -0.408 | 0 |
| AUC | teV/tpeV | 0.394 | 0 |
| depth | spec | 0.341 | 0 |
| depth | #genesV | -0.303 | 0 |
| AUC | depth | 0.265 | 0 |
| depth | epp_V/tpEppV | 0.264 | 0 |
| depth | epp/tpEpp | 0.262 | 0 |
| epp_V/tpEppV | spec | 0.192 | 0 |
| epp/tpEpp | spec | 0.186 | 0 |
| depth | sdAUC | 0.176 | 0 |
| epp_V/tpEppV | #genesV | -0.145 | 0 |
| epp/tpEpp | #genesV | -0.135 | 0 |
| epp_V/tpEppV | sdAUC | -0.104 | 0.0006 |
| depth | teV/tpeV | 0.099 | 0.001 |
| AUC | spec | 0.095 | 0.0015 |
| sdAUC | teV/tpeV | -0.093 | 0.002 |
| epp/tpEpp | sdAUC | -0.092 | 0.0021 |
| spec | teV/tpeV | 0.085 | 0.0046 |

*Table 29: Correlations between GO-properties, humans*

| Var1 | Var2 | corr | p_value |
|---|---|---|---|
| #genesV | spec | -0.999 | 0 |
| epp/tpEpp | epp_V/tpEppV | 0.728 | 0 |
| #genesV | sdAUC | -0.539 | 0 |
| sdAUC | spec | 0.538 | 0 |
| AUC | epp_V/tpEppV | 0.462 | 0 |
| AUC | epp/tpEpp | 0.378 | 0 |
| AUC | sdAUC | -0.337 | 0 |
| epp/tpEpp | #genesV | -0.248 | 0 |
| epp/tpEpp | spec | 0.247 | 0 |
| epp_V/tpEppV | #genesV | -0.236 | 0 |
| epp_V/tpEppV | spec | 0.235 | 0 |
| depth | spec | 0.134 | 0 |
| epp/tpEpp | sdAUC | 0.13 | 0 |
| AUC | teV/tpeV | -0.13 | 0 |
| sdAUC | teV/tpeV | 0.129 | 0 |
| spec | teV/tpeV | 0.129 | 0 |
| #genesV | teV/tpeV | -0.128 | 0 |
| epp_V/tpEppV | sdAUC | 0.122 | 0 |
| epp/tpEpp | teV/tpeV | -0.104 | 0 |
| depth | #genesV | -0.09 | 0.0001 |
| depth | sdAUC | 0.073 | 0.0011 |

*Table 30: Correlations between GO term properties, yeast_PPI*

| Var1 | Var2 | corr | p_value |
|---|---|---|---|
| epp/tpEpp | epp_V/tpEppV | 0.959 | 0 |
| #genesV | spec | -0.754 | 0 |
| epp_V/tpEppV | teV/tpeV | 0.575 | 0 |
| #genesV | sdAUC | -0.533 | 0 |
| epp/tpEpp | teV/tpeV | 0.479 | 0 |
| AUC | epp_V/tpEppV | 0.373 | 0 |
| sdAUC | spec | 0.359 | 0 |
| AUC | epp/tpEpp | 0.34 | 0 |
| depth | spec | 0.268 | 0 |
| AUC | teV/tpeV | 0.241 | 0 |
| depth | epp_V/tpEppV | 0.234 | 0 |
| depth | epp/tpEpp | 0.23 | 0 |
| depth | #genesV | -0.226 | 0 |
| epp_V/tpEppV | spec | 0.199 | 0 |
| epp/tpEpp | spec | 0.196 | 0 |
| epp/tpEpp | #genesV | -0.163 | 0 |
| epp_V/tpEppV | #genesV | -0.158 | 0 |
| AUC | depth | 0.104 | 0.0007 |
| depth | sdAUC | 0.102 | 0.0009 |

*Table 31: Correlations between GO-properties, chickens*

| Var1 | Var2 | corr | p_value |
|---|---|---|---|
| #genesV | spec | -1 | 0 |
| epp_V/tpEppV | teV/tpeV | 0.741 | 0 |
| AUC | sdAUC | -0.571 | 0 |
| #genesV | sdAUC | -0.45 | 0 |
| sdAUC | spec | 0.45 | 0 |
| epp/tpEpp | teV/tpeV | 0.446 | 0 |
| depth | #genesV | -0.441 | 0 |
| depth | spec | 0.441 | 0 |
| AUC | depth | 0.439 | 0 |
| epp/tpEpp | sdAUC | -0.381 | 0 |
| AUC | teV/tpeV | -0.361 | 0 |
| depth | teV/tpeV | -0.255 | 0.0029 |
| AUC | epp/tpEpp | 0.222 | 0.0101 |
| epp/tpEpp | epp_V/tpEppV | 0.18 | 0.0369 |
| AUC | epp_V/tpEppV | -0.178 | 0.0392 |

In Tables 28-31, we observed:

- The strong correlation between epp_V/tpeppV (or epp/tpepp) and teV/tpeV could be regarded as an indicator of the positive genes are more interconnected than the genes that do not have the functions. Thus, to some extend, a higher correlation between epp_V/tpeppV and teV/tpeV is related to a most completed state of annotation. We observe that the correlation is higher for yeast, yeast_ppi (0.83, 0.57, respectively), than for humans and chickens (not significant in both cases).

- The strong negative correlation between #genesV and sdAUC, was observed in the four cases, and suggests that reproducibility is higher for GO terms with a large number of validated labeled genes. Which makes sense, because when the number of labeled genes is low. The prediction will depend on which of the few labeled genes enter the train or the test set.

- The correlation between specificity and depth was strong and negative (0.34 for yeast), indicating that GO terms that are associated with less GO terms have a higher depth, which is in line with the definition of depth.

- Corr between epp/tpEpp and epp_V/tpeppV is stronger for yeast than for yeast_ppi and stronger in yeast_ppi than in humans, and is considerably lower for chickens. This is because a large portion of the gene-GO-term associations are validated in the first three cases whereas in chickens the portion is much

lower.

- The correlation between specificity and #genesV is slightly weaker for yeast and yeastPPI than for humans and chickens. An explanation for this is that spec by definition is the inverse of the sum of all associated genes, whereas in the case of yeast or yeast_ppi, most of the annotations found are also found in the other three cases, whereas in chickens and humans some associations are found only in these species.

Figures 25-28 show in a graphical manner how these correlations differ between the species considered.

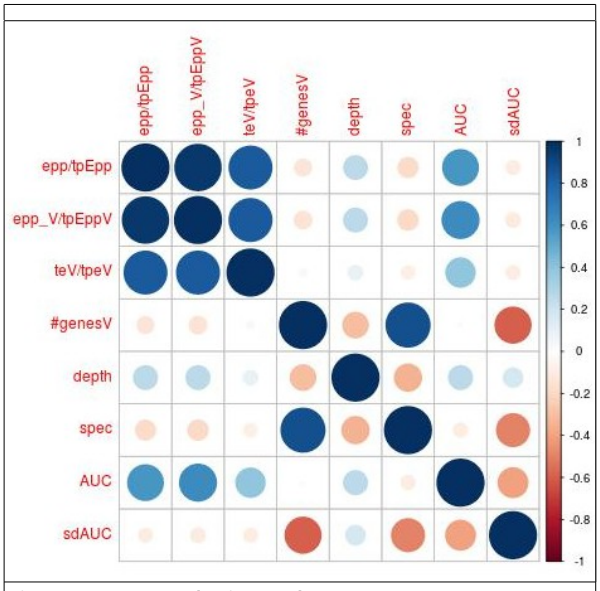



*Figure 25: Correlations plot yeast*

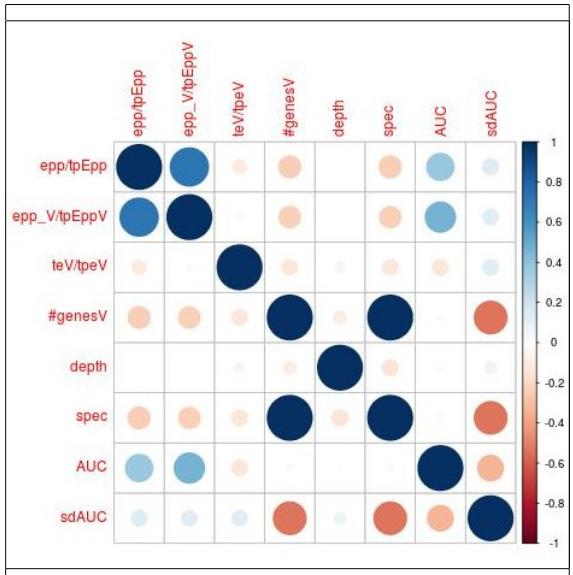



*Figure 26: Correlations plot, humans*

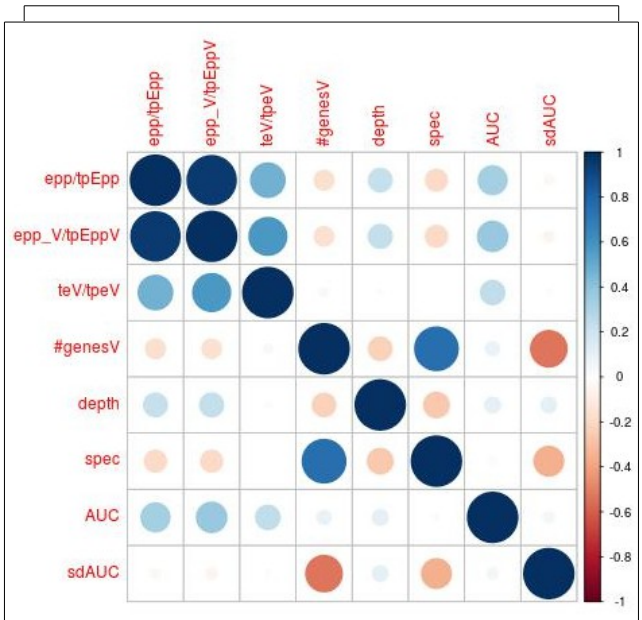



*Figure 27: Correlations yeast PPI*

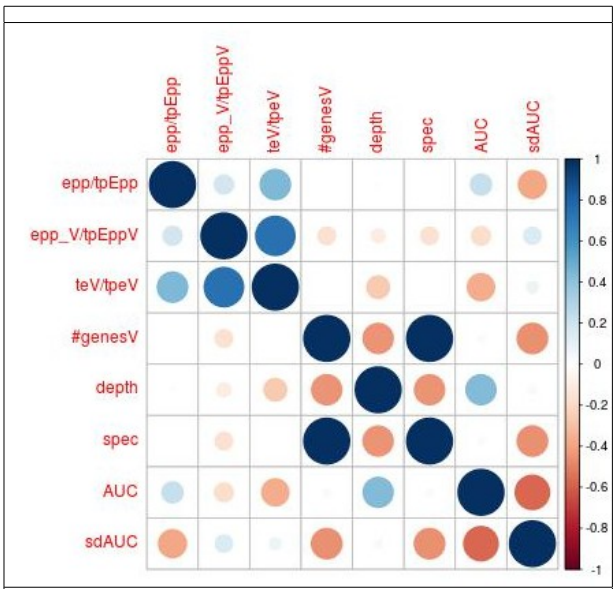



*Figure 28: Correlations plot, chickens*

Additional analysis on yeast data showed that the correlation between AUC and the number of edges became stronger as the number of annotations in the data became smaller. Table 33 shows the value of the correlation AUC-#edges for different sizes of association files.

*Table 32: Impact of the number of annotations on the correlations AUC-#edges.*

| #assoc. | CorrAUC_ #edges |
|---|---|
| 264,279 | 0.025 |
| 132,249 | 0.124 |
| 111,480 | 0.252 |
| 110,682 | 0.255 |
| 104,582 | 0.174 |
| 104,303 | 0.133 |
| 58,358 | 0.376 |
| 32,336 | 0.356 |
| 8,862 | 0.219 |

From Table 32 we learnt that the highest correlation between AUC and t#edges (0.376) was reached when the number of association in data was around 1/4 of the total of associations available for yeast. Then, with this number of associations (58,358), we compared AUC in different subsets of GO-terms defined based on the number of edges. Table 33 shows the prediction performance for 5 bins of GO terms. The first bin corresponds to the 1th/5th of the GO terms with lower number of edges, thee second bin corresponds to the 2th/5th, and so on.

*Table 33. Differences on AUC between groups of GO terms with different # of edges*

| Bin of GO terms | AUC |
|---|---|
| 1th/5 | 0.648 |
| 2th/5 | 0.654 |
| 3th/5 | 0.674 |
| $4^{th}$/5 | 0.706 |
| $5^{th}$/5 | 0.730 |

*Data was from yeast and had a total of 58.358 associations. The first bin (1th/5) refers to the 1th/5th of the GO terms with a lowest number of edges, and so on.*

Additional analysis on humans data showed a strong correlations between epn and epp (0.897). This could be regarded as an indicator that either, the annotation is incomplete or

the quality of the co-expression data is low. We would expect that as the quality of the data increases, this correlation would weaken. The same analysis also showed a significant negative correlation between epn and Epp/tpepp (-0.25), which was expected, since the tpepp is the sum of epn and epp.

Since epp/tpepp happened to be the most important GO property for predicting the accuracy of prediction, we investigated how the ratio epp/tpepp changes across the species considered. We compared epp/tpepp at the level of individual GO terms by creating a histogram for each species and considering the same GO order of GO terms (thus, only the GO terms that were present in the 4 data-sets were considered

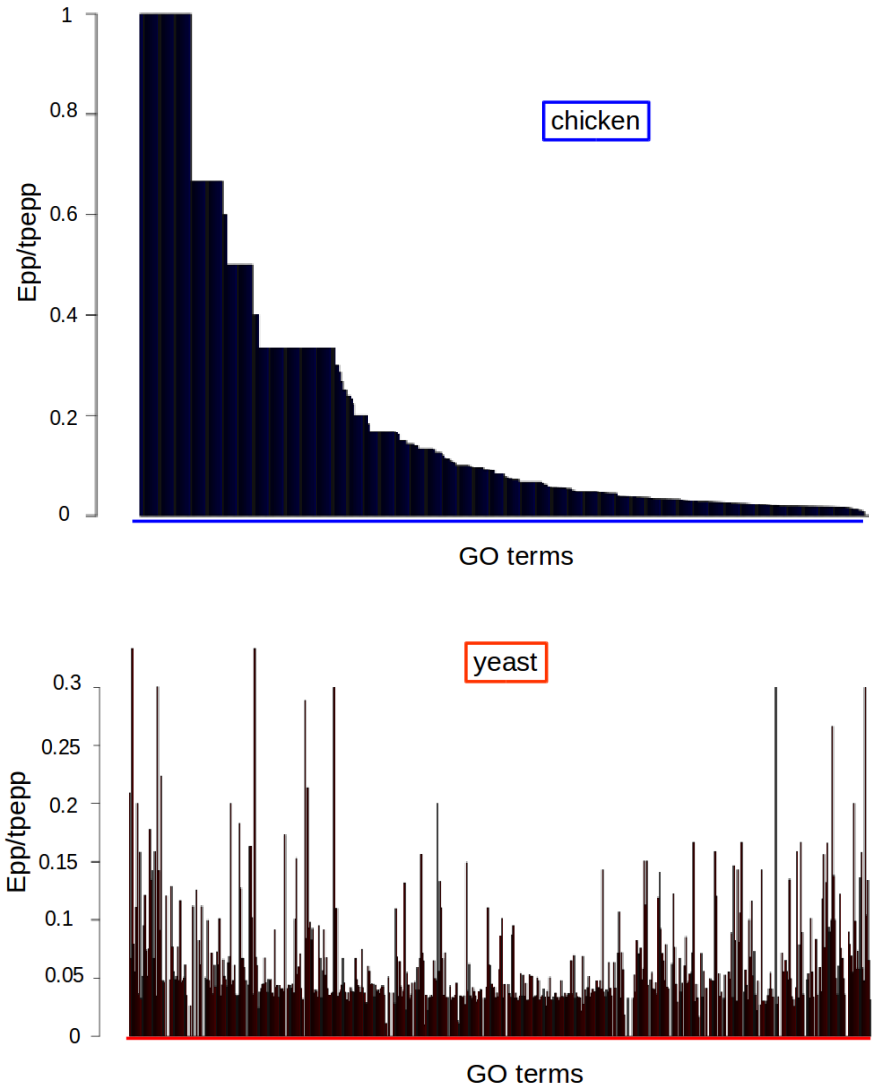



*Figure 29: Ratio epp/tpepp at the level of individual GO terms (chicken and yeast)*

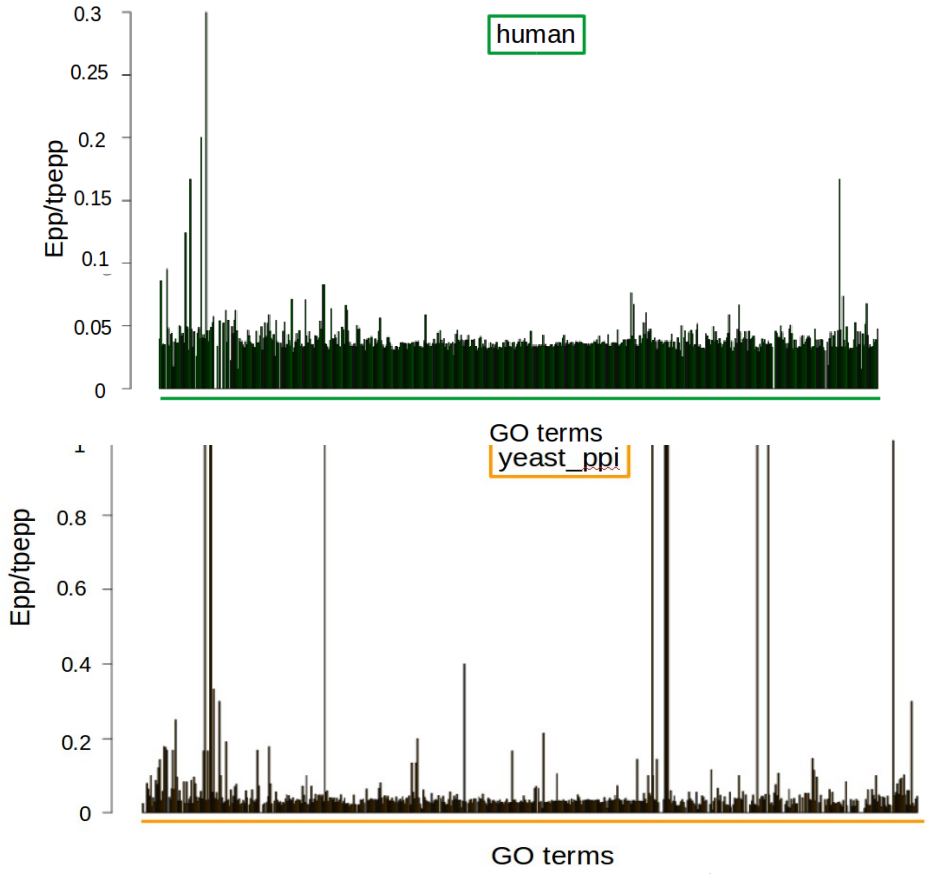



*Figure 30: epp/tpepp at the level of individual GO terms (human and yeast_ppi)*

In Figures 39-30 we observed that the epp/tpepp is a GO property that greatly depends on the data, and the value is not extendible to other species.

## 4. Part 4. PU-BMRF

We investigated which type of GO terms respond better to PU-learning. For this, we computed the correlation between the increase in AUC and the GO properties defined in Part 2 (Table 34).

*Table 34: Correlation between the increase in AUC with PU and GO-properties*

| Var1 | Var2 | corr | p_value |
|---|---|---|---|
| AUC_increase | epp_V/tpEppV | 0.251 | 0.2058 |
| AUC_increase | te/tp_e | 0.222 | 0.2662 |
| AUC_increase | epp/tpEpp | 0.18 | 0.3689 |
| AUC_increase | epp | 0.153 | 0.4468 |
| AUC_increase | #genes | 0.148 | 0.4608 |
| AUC_increase | eppV | 0.134 | 0.5038 |
| AUC_increase | #genesV | 0.126 | 0.5319 |
| AUC_increase | spec | 0.126 | 0.5319 |
| AUC_increase | teV | 0.124 | 0.5361 |
| AUC_increase | te | 0.101 | 0.617 |
| AUC_increase | depth | 0.085 | 0.673 |
| AUC_increase | teV/tpeV | 0.04 | 0.8425 |

Additional results in Part 3 include:

- A comparison of AUC between BMRF, PU-BMRF and random PU-BMRF (when the RN were extracted randomly) (Table 35)

- Estimates of the standard deviation across replicates in the process of extraction of RN (Table 36). The standard deviation was low.

-A density plot of the portion of RN that was extracted in the four replicates considered, in the different GO terms (Figure31). The percentage of RN extracted in the four replicates was high for almost all GO terms. Thus, the process of extraction of RN has high reproducibility.

- A comparison between the prediction performance for chickens when PU-BMRF was used and humans when BMRF was used. We observed for chickens (AUC>80), whereas for humans using BMRF, only one GO term was predicted with an average of 80% (Figure 32)

- A density plot for the number of RN that were extracted for the different GO terms when the AUC in the process of extraction was 0.9, 0.95 or 1 (Figure 33).

-A density plot o the standard deviation of the number of RN that was extracted, for different values of AUC. From the plot, we learn that the number of RN extracted varies considerably even AUC was 1 (Figure 34).

-An overview of the computational time required by PU-BMRF(Table 37)

| | | AUC(sd) | |
|---|---|---|---|
| **#RN** | **BMRF** | **BMRF random extract** | **BMRF-PU** |
| 1000 | | 0.643 (0.094) | 0.723 (0.08) |
| 2000 | | 0.65 (0.09) | 0.75 (0.072) |
| **3000** | | **0.64 (0.1)** | **0.758(0.084)** |
| 4000 | | 0.636 (0.095) | 0.751(0.086) |
| 5000 | | 0.652 (0.087) | 0.725 (0.094) |
| 6000 | | 0.644 (0.087) | 0.728 (0.089) |
| 7000 | | 0.638 (0.096) | 0.716 (0.092) |
| 8000 | | 0.636 (0.095) | 0.701 (0.095) |
| all | 0.706 (0.0793) | | |

*Table 35: Comparison accuracy of prediction BMRF vs PU-BMRF*

*Table 36: sd across replicates in the process of RN extraction*

| **max #RN** | **sd_across_replicates_in the_process_of_Rnextraction** |
|---|---|
| 1000 | 0.033 |
| 2000 | 0.04 |
| 3000 | 0.043 |
| 4000 | 0.044 |
| 5000 | 0.045 |
| 6000 | 0.045 |
| 7000 | 0.045 |
| 8000 | 0.046 |

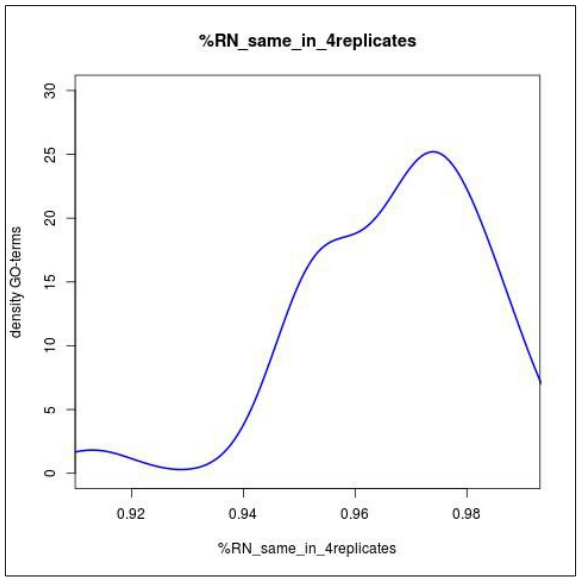



*Figure 31: Reproducibility in the process of extraction of RN*

*Considering a maximum of 3000 RN were extracted*

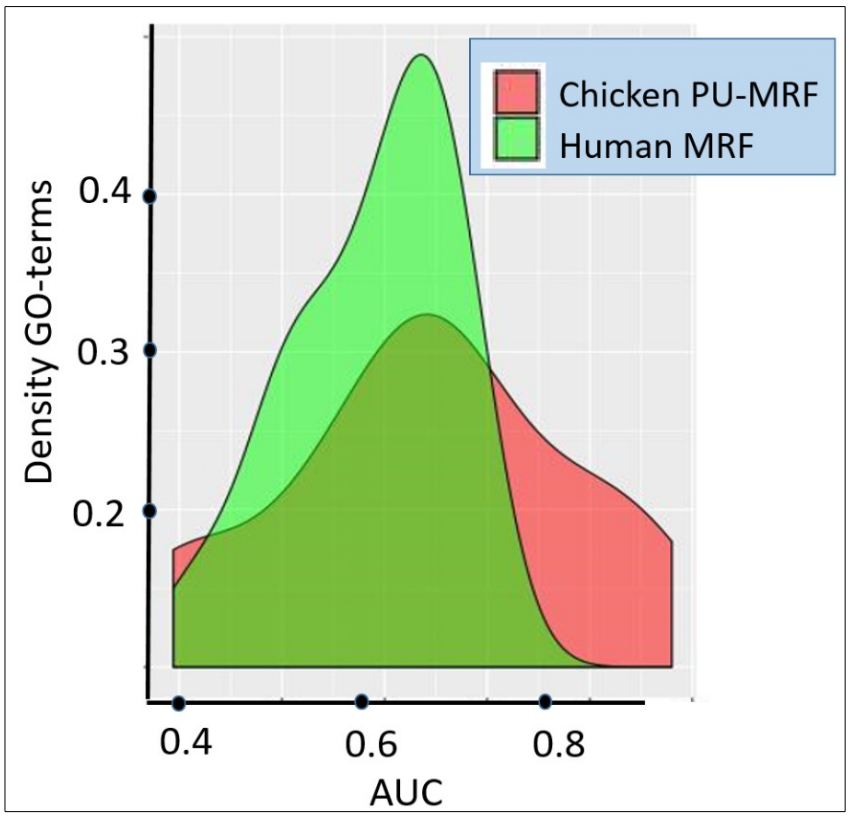



*Figure 32: AUC distribution, chickens PU-BMRF vs humans BMRF*

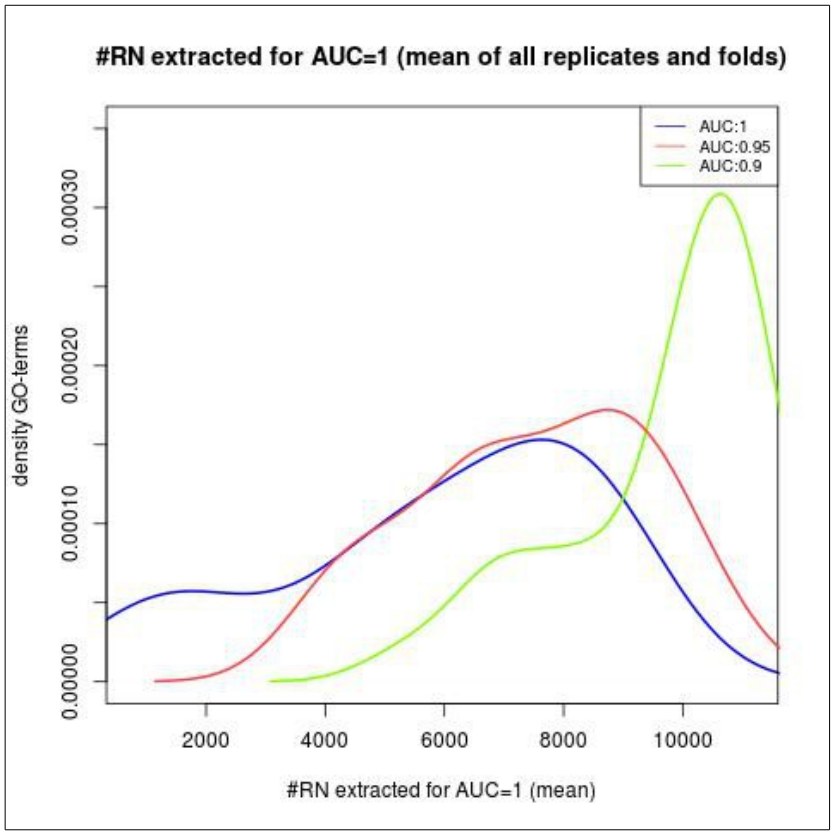



*Figure 33: Numbers of RN extracted given minimum values of AUC*

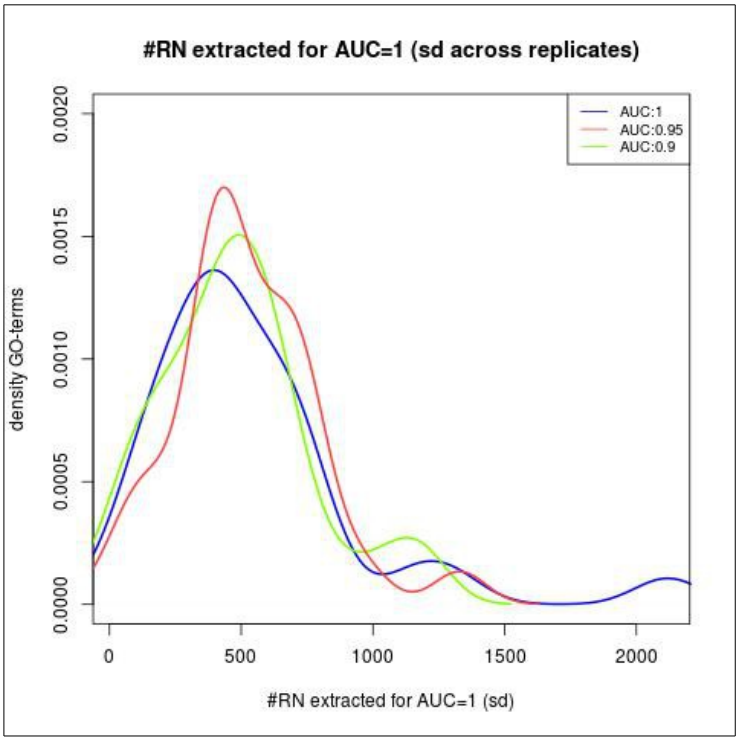



*Figure 34: Sd in the process of extraction of RN, given minimum values of AUC*

| Step | Description | computational time in minutes for 1 GO-terms (% of total time) |
|---|---|---|
| 1 | Similarity Matrix | 40 (18.2%) |
| 2 | Creating the folds | 10 (4.5%) |
| 3 | Network features | 40 (18.2%) |
| 4 | non-GO-specific features | 20 (9.1%) |
| 5 | GO-specific features | 60 (27.27%) |
| 6 | extraction of RN | 50 (22.73%) |

*The time corresponds to using k:10 and 4 replicates*

*Table 37: Approximate computational time for the different steps of PU-BMRF*

Computational time of BMRF for k:10 and 4 replicates was ~5 minutes.

## 5. Part 5- Co-expression cascades

Three additional plots are provided about the correlation of (correlation epp/tpepp – Specificty) with minGOsize. These correspond to yeast, yeast_ppi data and chickens, Figures 35-37, respectively. Interestingly, in yeast, yeast_PPIs, as well as for humans (shown in Results), the correlation was maximum when the minGOsize was 2000. In the case of chickens, the pattern is very different.

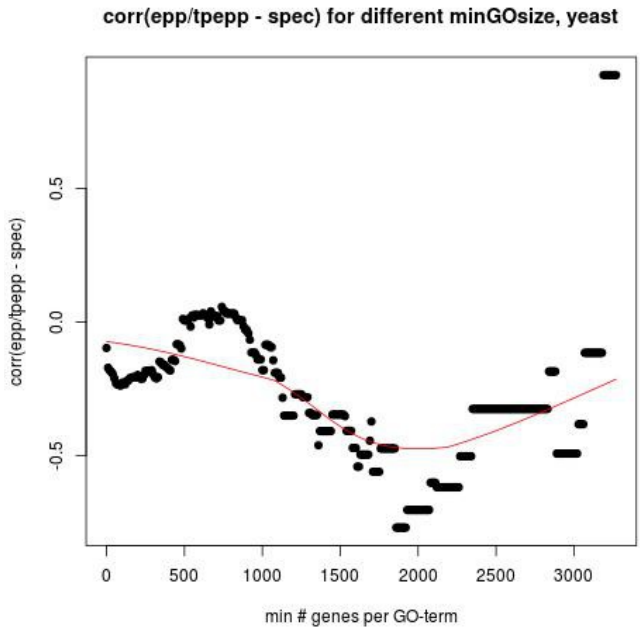



*Figure 35: Correlation between (correlation epp/tpepp - specificity) and minGOsize, yeast*

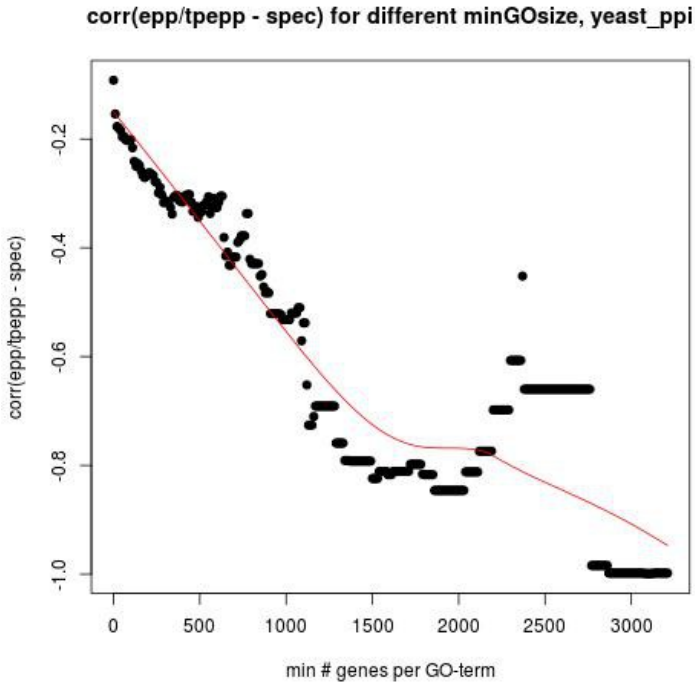



*Figure 36: Correlation between (correlation epp/tpepp - specificity) and minGOsize, yeast_ppi*

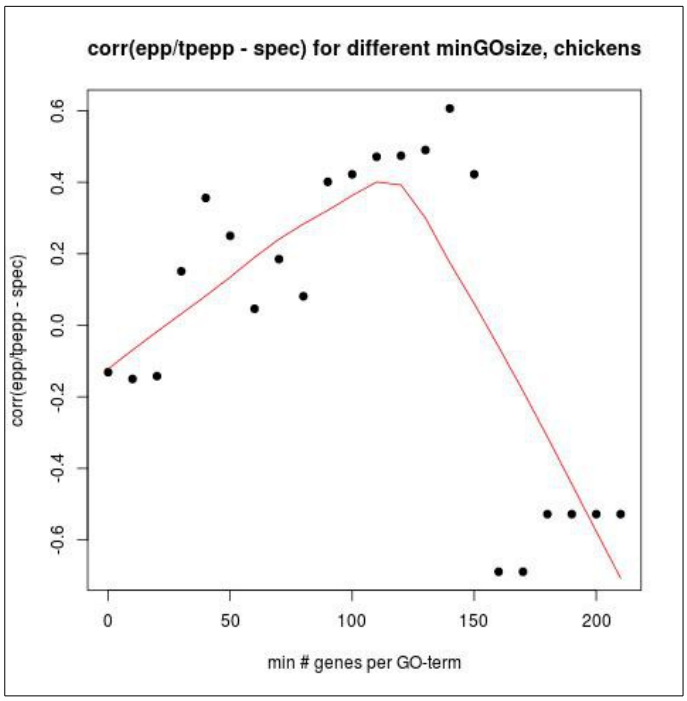



*Figure 37: Correlation between (correlation epp/tpepp - specificity) and minGOsize, chickens*