Protein function prediction for poorly annotated species

Introduction

Computational methods for PFP and network-based methods

Protein function prediction (PFP) is one of the most important aims of modern biology. In crop and livestock species, PFP is conventionally based on annotation transfer from the few well-studied species, such as Arabidopisis and humans. While successful, these methods rely on the assumption that homologs proteins share function, which has been proved wrong in many cases [1]. It is thus desirable to complement the orthology-based methods with other approaches.

Network-based methods, for instance, infer the function of proteins exploiting the principle of guilt-by-association. Based on this, proteins that interact are likely to have similar function [2]. These methods have a lot of potential because they can utilize the information generated by high-throughput biological experiments, such as co-expression [3] or protein-protein-interactions [4] to construct networks from which to infer function.

Networks in poorly annotated species

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

In livestock species, network data is even more limited than for the aforementioned crop species. Nevertheless, it is expected that the data available will increase in the coming years. Efforts such as The Functional Annotation of Animal Genomes consortium (FAANG) [6], are currently generating functional annotations for some relevant species such as pig and chicken. This information could be utilized by the network-based methods. A previous study has used coexpression networks in chickens to infer function via defining GO-enrichment-modules [12]. In tehir approach, they stressed the importance of netwoprkk methods to identify gene modules, as well as regulatory genes that are relevant for a set of functions or functional cascades. This approach based on GO-enrichment, however, is indirect in that it first identify identify functional modules in the network. Sharan et. al. [16] judged the direct methods (directly predict the function of a protein) as slightly superior to the indirect ones. Furthermore, the method does not enjoy the advantage of the direct statistical-learning-based methods. Methods based on statistical learnig have more potential than other methods because they can identify combinations of features that correlate with certain functions [20]. An interesting question therefore is whether it is possible to use some of the direct statistical-learning-based methods that are efficient for PFP, in livestock species. Furthernmore, it is interetsing to9 investigate whether such a network defined for PFP can also throgh som elight on how genes interact with each other to redeem more or less sepcific functions.

BMRF

In order to develop a statistical-learning-based network method that is efficient for livestock species, a logical approach is to utilize one of the state-of-art methods used for crop species. Bayesian Markov Random Field (BMRF) [1] is a prediction method that was developed with the purpose of achieveing accurate predictions when the data was far from complete.

The prediction ability of BMRF was compared to other methods' in the Critical Assessment of protein Function Annotation (CAFA)[X] experiment and its prediction performance was high for some species (Illustration 1). BMRF is particularly efficient for poorly annotated species for two reasons mainly: First, it can synthesizes 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 directly exploits the information from the neighbors, we would expect that PFP via BMRF will be more accurate for the genes with a large number of neighbors (co-expressed genes) and whose neighbors information is accurate. Thus, we would expect better prediction performance for the most general GO-terms and, in particular, for the hub genes that are located in high-density regions in the network. However, on the other

hand, from a biological perspective we expect that the genes are more co-expressed in the most specific GO-terms and therefore the principle o guilt-by-association holds better in this cases. BMRF, therefore, may not only be a good starting point for inferring PFP in livestock, but it also may be useful to identify regulatory genes.

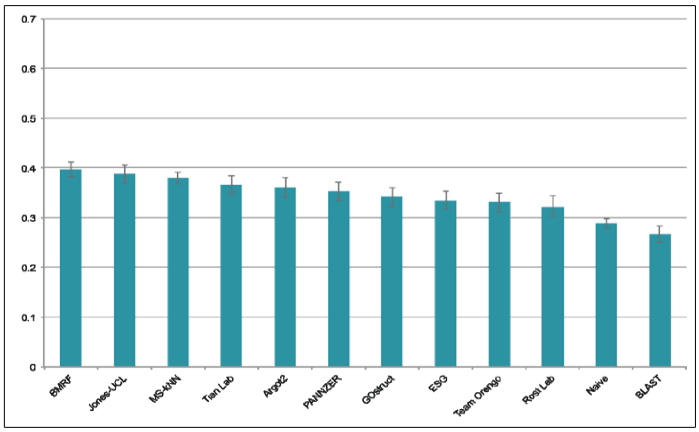


Illustration 1: Comparison of BMRF with other PFP methods. Evaluation for the Biological process category in H. Sapiens. Source:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3584181/f

Positive-unlabbeled learning

We discovered an important potential problem with BMRF. The learning process may be biased because BMRF attempts to solve a one class classification problem where the annotated data consists solely on positive cases. This is because it is very difficult to be certain that a protein does not perform a function. In fact, this problem applies to many situations in bioinformatics because from a biological perspective, the lack of evidence for a connection does not imply that such a connection does not exist. As an example, for rice 415 proteins have experimental evidence for a biological process, but not a single protein has a validated proof of no-connection with a function. Since only positive associations are reported, the negative set is composed of all unlabeled data. This leads to some bias in the prediction because the unlabeled data may contain some positive cases. This problem increases with increasing numbers of unannotated proteins, such as in the case of network data from livestock species. To overcome this problem, a new type of machine learning has emerged called Positive Unlabeled learning (PU). With PU it is possible to identify the proteins that are more unlikely to have a given function. Hence, the number of unlabeled cases can be minimized by extracting some “reliable negatives” proteins from the set of unlabeled. It has been proved theoretically that, by identifying sets of reliable negatives, PU improves the performance of machine learning algorithms in situations where only positive labels are known [7]. PU has been successfully applied to a variety of problems related to PFP [7-10,13]. In [7], for instance, the authors extracted a set of reliable-negatives proteins from the unlabeled dataset by defining a threshold of similarity based on the euclidean distance between a set of positive proteins and the set of unlabeled. Yang et al. [8] developed a multi-label version of PU learning to identify genes associated with diseases; [9] developed two novel approaches to identify reliable negatives that can be applied in different algorithms. Jiang et al. [10] applied PU on a support vector machine and outperformed all pre-existing methods for pupylation sites prediction, and then Nan et al. [13] improved their method by adding as a first step to the algorithmn in [10], the method described in [7]. Lastly, in another recent study, Nusrath et al. [14] used a Self organizing map to extract reliable negatives from unlabeled data-set of drug-drug interactions. None of these studies, however, have applied PU learning on a BMRF. BMRF is one of the few methods that can handle networks with a large portion of unknown cases, and PU can extract reliable information from these unknowns. Our hypothesis therefore is that a PU implementation of BMRF will be particularly effective for PFP in species with limited network resources, such as pig and chicken. Furthermore, since with PU it is possible to create an alternative database with negative examples, the task of identification of regulatory elements is, to some extend, simplified.

The aim of this study is to develop a PU implementation of an existing Bayesian Markov Random Field algorithm that can efficiently assign proteins to GO term categories and identify putative regulatory elements using network data from chickens.

Thesis setup

Illustration 1, shows a diagram of the tehsis setup.

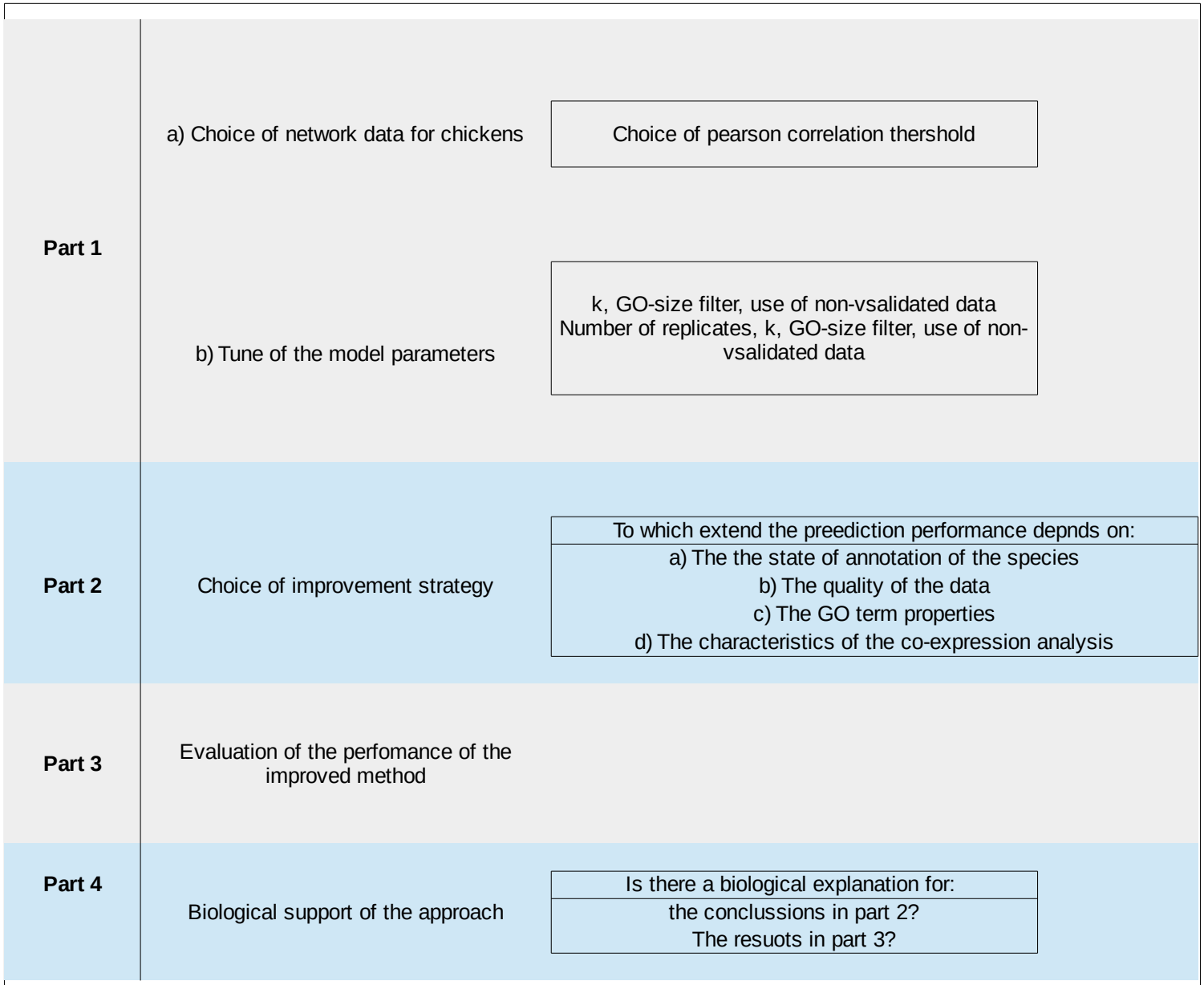


Illustration 2: Thesis setup

Material and methods

Data preparation

Data sources for the different species are shown in table XX in Appendix II-Overview_data. In the GO-terms file, the associations were coded as "valid" if, for at least on of the association available in data, they correspond to Experimental evidence scores (precisely: 'EXP', 'IDA', 'IEP', 'IMP', 'IPI', 'IGI') and to the category of Biological process (BP), and as "non-valid" otherwise. Note that “valid” and “NONvalid” are just a level in a binary class that is defined for each associations and should not be misleaded with the laebl of a gene with reagard to a GO term (0,1), as explaiend in the section “Validation in BMRF”.

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

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

Markov random fields

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

In a MRF the probability of a certain assignment of discrete states x=x1,...xN) is (as explained in [21]:

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

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

Computing this is hard and often the equation is kept only until the second order. Also, equation XX is often homogenize by defining the same potential function for all cliques of the same size. Thus, we have:

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

In the PFP context, the proteins are represent as nodes in a network and the edges represent a similarity relationship between the two nodes they are connecting (for instances, either protein-protein-interactions, domain information pr co-expression between genes). The annotation is 1/0, either the protein has the function or not.
Deng et al., 2003 adapted MRF to the PFP problem and stated that the probability over the enter network is proportional to $\exp(\alpha N_{01}+\beta N_{11}+N_{00})$, where N_{01}, N_{11},N_{00}, correspond to the number of pairs of proteins that, while interacting, only one has the function, both of them have the function, and non of them has the function, respectively. And α and β are weighting the contribution of each of these classes of pars of proteins. Then, combining the a priory probability of an assignment with N_1 '1's, which depends on the frequency of the function and is proportional to $(f/(1-f))^{N_{,1}}$ they obtain a homogeneous second order MRF and the following equation to estimate the probability that protein *v* is assigned with the function given the annotations of its neighbors *N(v)*

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

where N(v,i) is the number of neighbors of v that are assigned with $i\in\{0,1\}$ and logit is the logistic function logit(x)=1/(1+e−x). *Deng et al (2003)* estimate the two parameters of the model using a quasi-likelihood method and apply Gibbs sampling to infer the unknown functional annotations. The approach has two steps: first, they estimate the parameters and, second, they infer the label using gibb-smapling. The parameters are estimated by maximizing the PLF with logistic regression. In this, each protein is 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 eta la, overcame this by simply ignoring the unannotated proteins in the parameter estimation step. This can be problematic when the number of unknown proteins is large because by ignoring the unknowns, the neighbors are pruned and they may no longer express the complexity of the network.

Bayesian markov random fields

Janis et al., 2010 developed a Bayesian Markov Random field (BMRF) to overcame the aforementioned problem. In BMRF, a 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 Gibb sampling. Thus, a candidate point θ'=(α', β') is obtained using the equation:

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

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

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

The labeling vector x is initialized using the output of the MRF-Deng, as explained in [plos1]

Validation in BMRF

BMRF [1] was used to do PFP and learn about the impact of different method and network parameters on the prediction performance. In this framework, predictions are made individually for each GO-term that passes the GO-size filter (see Appendix I - Concepts). The predictions, however are not completely independent for each GO-term in the dataset because the genes that are not associated with any of the GO-terms in the dataset are treated differently. These genes are coded as "unknowns", and the number of unknown genes depends on the number of GO-terms in the database (which can be regulated through the GO-size filter). Each gene in the network file will enter the BMRF code with one of three possible labels: '1', '0' or '-1', where '1' stands for positive, '0' stands for unlabeled (it is not known whether the gene has the function) and '-1' stands for unknown (it is not known whether the gene has the function and its label in the training set will be '0' or '1' based on Gibb-sampling taking into account the label of its neighbors). Thus, the training set consists on those genes that enter the BMRF prediction function with a '1', or a '0', and the test set consists on those genes which are labeled as a '-1'. Note each gene have one label for each GO term (1 or 0) depedning on wther the gene is associated or not with the GO term. However, that the unkown are labeled as '-1', for all the GO terms.

Then, the function provides a vector of posteriors corresponding to the probability that each gene has the function, and these posteriors are transformed into a '1' if the is above a certain threshold or '0' otherwise. The threshold is chose in such a way that AUC for the test set is maximized. Note that both, the train set and the test set will be predicted but the accuracy of prediction was solely based on the test set.

The advantage of treating the unknown genes (genes known to be associated with zero GO-terms) differently is that genes that have never been predicted as positives are less likely to be non-associated for a given function than those genes that have been found as positives for some function but not for the function of interest. This has to do with the fact that some function are more difficult to predict than others. Thus, if based on data, a gene has never been identified as positive for any function, it is more fair to assume that the function is particularly difficult to predict using experimental approaches, that assuming that a very low number of genes have the function. In other words, a large portion of unlabeled genes ("0") implies that the function is rare, (almost never observed), whereas a large portion of -1s implies that the function is difficult to predict. This distinction between unlabeled and unknowns genes, additionally, allows to do predictions in a fairer fashion. By treating the gene of interest as unknown (-1) instead of as unlabeled (0), the prediction of the gene-GO association we are interested in, will be more free from the prediction of the GO-term. If the portion of "-1" however becomes very large with respect to the portion of "0", however, Gibb-sampling will fail in the relabeling, because it will expect that the portion of genes that have the function is very large.

The labeling at a given fold in the k-fold CV is as follows:

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

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

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

Fold 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. Genes in one of these k sets will be in Positive-test, the rest will be in Positive-train. Within the same analysis, this is repeated k times. Each time corresponds to a fold with a new labeling configuration such as represented in Table 1. Note that only the label of those gene classes that have 'a' or 'b' in the fold column change with each fold. For each fold, a new k set takes place as Positive-test and the rest are Positive-train, until each of the the k sets that were defined at the start of the analysis have been assigned once to the Positive-test.

Fold b: k-fold CV for the unlabeled. Same as Fold abut for unlabeled genes instead of positives.

Part 1- Choosing the data and tunning the model parameters for prediction performance using BMRF.

A first step involved the choice of a co-expression threshold for chickens given a conditionaly independte co-expression netwrok. 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 we a perason correlation of 0.7 would lead to a scarcity of network data in BMRF. Furthermore, the assoictaions file would need to be heavily prunned because BMRF does not allow for any gene in the association file sthat is not in the network. Thus, we computed AUC for several pearsion correlations thersholds. Note that for each person correlation, the GO file needs to be uppropagted. Details about which pearson correl;ation cutoffs were considred are given in section 1A in Appendix III-Additional results.

Since the assignations of genes to folds in the k-fold CV is a random process, we investigated how many replicartes were required to achieve reprodiucbile results. Then, we tunned the follwoig model paramateres by chaniging values in each of them at a time: GO-size filter, Number of folds in k-validation, Number of iteration in Gibb-sampling. Lastly we computed AUC with and without domain information and non-validated associations. Details about which values were considered for each model parameter are given in section 1AB in Appendix III-Additional results.

Part 2- Choosing improvement strategy

In order to investigate which avenue of improvement for BMRF would be more effective, we estimated the impact of different parameters on the prediction perfomance. We performed foru types of analyses:

-Invetsigate the differences in prediction performance between species
-Investigate the impact of differnet GO-term-properties on the prediction performance
-Investigate the impact of the quality of the data on the prediction performance
-Invetsigate the Impact of the nature of the network on the prediction performance

- Invetsigate the differences in prediction performance between species

We analyzed the differences in data between the species considred and we investigated how this difference3s are translated in a differnet prediction performnace. Differences in network data that we considred were: #te, #epp, #epn, #enn, #epp*100/#te, #epp/tpepp; and differences in annotation data considred were: #genes per GO-trem, #GO-terms per gene, #edges per gene, and #epp per GO term. These differences can be found in Appendix II-data overview, as well as in section 3a in Appendix III-Additional results. Appendix II provides a more graphical representatuion of these differences hwreas Appendix III focuses on how these differences may be affecting the prediction perfomance. Formal definitions of these parameters can be found in Appendix I-concepts.

- Investigate the impact of differnet GO-term-properties on the prediction performance

A total of 9 GO-term-properties were defined: epp/tpEpp, eppV/tpEppV, #genesV, sepc, teV/tpeV, depth, AUC and sdAUC. We computed the correlation between these in order to investigate how they may be affecting the prediction perfomance. Definitions of these GO-term-properties care also

gioven in Appendix I-concepts.

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

In order to investigate the impact that the quality of the data has on the predictions, we computed AUC after randomly removal of 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: edges positive -positive, epn, enn, te. Portions subtracted were 0, 10, 30, 50, 90 and 95% when yeast data was used; and 0, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 0.99 when human data was used. Then the correlation between AUC and the percentage of removal was computed in order to asses the impact of the amount of data considered and the prediction performance.

- Investigate the impact of the nature of the network on the prediction performance

We investigated how thye consitions of the co-expression anlayses influence the preedictionperfomance. For this we took differnet substes of co-expression data and assesed the prediction perfomance in each case. The source of expression data for yeast is organized based on expreimnts and a briedf description of these is provided. Thus, we searched for jey words, like “oxidation” or “stress” and creaseed substes of networks with data from those experimntes that have those wors in the description.

In the case of humnas, data is organized by tissues. Thus, each tissue was a subset. We homogenized the substes based on epp/tpepp instead of #te, because in part 1 we learnt that the former parameter has barelly any impacdt on the prediction perfoamnce. In order to homogeneizem, we removed randomly edges from the network. Then, we computed AUC with the expression data from te different tissue experoments.

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

Based on the results from these 4 types of analyses, we chose an approach to improve the BMRF PFP method. All analysis were carried out with the network data and the values for model parameters chosen in part 1. In order to estimate the reproducibility opf the approach, the standard deviation across folds within the same replicate, as well as across replicates, were computed.

Part 3- PU-BMRF development

Similarly to Part 1, in Part 2, the folds for the set of positives were created solely based on the validated associations. However, for simplicity, in part 2 the non-validated positives associations were always excluded from the analysis. This can be done by simply setting the “only_EES” parameter to “True”.

Steps 1 to 6 aim the computation 86 features including 70 non-GO-specific features and 16 GO-specific features These features wre computed for each of the 1,656,000 total gene-associations (138 GO-terms x 12,000 genes). Note that, for chickens, only 138 GO-terms passed the GO-size filter.

Based on these features, we computed the euclidean disatnce between each of the genes in the set of known genes that were not masked (positive-train) and the set of unkwon gene sthat were not masked (unlabelled train), and we estimated the average euclidean distance between the known unkown gens and the known positive genes. Then,

we classified the genes in the test set as positive or negatives based on whether their distance to the positive genes was larger or smaller than the distance form the known unlabeled genes to the positive set, as explained in the algorithm introduced by [] (Illustration XX):

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

Illustration 3: Algorithm for extracion of RN. Source: [1]

Note that step 6 in the algorithm allows to introduce a constant to specify how strict we want to be. A vluw of 1 measn that gene sin th etest set that are closer to the posotive set than the average unlabeleled genes will be clasified as positive, but a constant of 1.5 would mean that we are more strict and only genes that are 50% colser to the positive set than the unlableled genes will be classified as positive.

- **Step 1 – Similarity Matrix:**

A similarity matrix between the GO-terms is computed using the R package "GOSim". The use of computing this matrix is two folded: First, it allow to extract a set of unrelated GO-terms, in case we cannot carry the analysis for all the GO-terms that pass the filter (for instance, due to time constraints); and second, this matrix will be used in the computation of the features (step 5).

- **Step 2 – Creating the folds:**

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 is created, in which the associations in the test-set have been exuded. Also, the set of genes that, after "hidding" the test set, are associated with the GO-terms are stored, and also their neighbors. Thus, for each GO-term, two objects are stored: the set of positive and the neighbours of the genes that aree in the positives. Finally, another object is extracted with the neighbors of each gene. These objects are

different for each fold and will be used in steps 4 and 5.

- **Step 3 – Network features:**

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

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

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

The following features are computed for each of the 12,000 genes:

- features f1-f4 refer to the number of GO-terms of the gene. It is expected that the genes that are associated with a large number of GO-terms are more likely to be associated with a novel GO-term. We expect this probability to be higher for genes that are associated with a large number of specific GO-terms (before up-propagating) because they may be involve in a larg enumber of functions. Genes that are associated with a large number of GO-terms only after up-propagating, however, may be associated with these simply because they are associated with more general GO-terms.

f1) The number of GO-terms the gene is associated with.
f2) The number of GO-terms the gene is validated-associated with
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

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

f5) The sum of the GO-terms that are associated with the genes that are co-expressed with the target gene.
f6) Thes um of the number of GO-terms that are associated with the genes that are co-expressed with the target gene in a database where only validated GO tersm are considered.

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

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

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

- features f10 to f70 are same as f1-f7 but for different features Pearson correlation thresholds. Mainly: 0.1, 0.2, 0.35, 0.5, 0.6, 0.7 and 0.8. Note that as the network database changes, so does the GO-file because BMRF does not allow for any gene that is not in the network. Due to the constrains in the data that come after the different correlation thresholds, it is expected that if a gene is associated with a very large number of GO-terms when the Pearson correlation was high (i.e. 0.6), it must be a gene involved in many different functions, whereas it may be that other genes are associated with more GO-terms when the Pearson correlation is lower, and this should also be considered.
The possibilities of restricting the data-set based on the Pearson correlation cutoff greatly increases the information available. This is because based on different correlation cutoffs we may be be able to observe different patterns in data. Note that this is particularly useful since we are using a condition-independent network where the data of different experiments is combined.

- **Step 5 – GO-specific features:**

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

Intersections of genes:

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

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

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

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

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

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

- **Step 6 – extraction of RN:**

The databases with features information obtained in steps 4 and 5 were combined and the values of the features were scaled by divding each value by the square root of the saured sum of all the values. Thus, for each GO-term

we have a database with 86 features per gene. Checking the label of the genes that are in the training set, we apply the algorithm in illsutarion XX:

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

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

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

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

Part3- Predictions with PU-BMRF

- Evaluation of the process of extraction of RN

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

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

- Reproducibility of the process of extraction of RN

In order to asses the eproducibility of the extraction of RN across replicates, we investigated which portionof the RN exctraed were cmmon in all the replciateds of the analysis. Furrther, we assesed the reproducibility across

folds within the same replicate. For tis, we calculated how many differnet RN were extracted per repllicate (combining the extraction of each fold).

- Prediction perfoamnce using the RN

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

Part4-Biolgical support of the approach

In principle, PU-learning should contrinute in improving the epp/tpepp bty removing epn edges from the data. From a biological perspective we wopuold expect that the genes that are involved in more specific functions have a higher epp/tpepp. The im of thios part is to invetsigate whether we obsere that princuiple in the data, whether the ratio improves more in genes involvedin specific funtin sor in general functios, and finaly hoiw this related to the increase in accuracy with PU-BMRF with respect to BMRF in both types of genes.

To invetsigate whether the principle of more speficic-better connecte, we followed two approaches. First we computed the correlation between specificty and epp/tpepp, and second, we compareed epp./tpepp in two groups of genes depednding on wther these genes are asscoiated with specific or general GO terms.

To cpmapre epp/tpepp in these two grpus, we looked into genes that are exlusive form general or form specific Go terms. For this, we subseted GO tersm with a sopecificty larger than 5 and less than 3000 assoiciate dgenes. The first requirement is to make sure that the GO terms considered have at least a mimunm of edges. Otherwise we would be taking as specific GO tersm GO terms that are simly associated with almost non of the genes. The seiond requirement is guarantee that there are some gene sin the specific GO tersm that are not in the specific GO terms. For instance, if we choose the GO-term XXX-Biological process, which is associated with XXX genes, it wold be very difficult to find genes that are associated with a more specific GO term bvut not with XXX.

Then we distinguish between two gropups of GO terms, one specific and one of general GO terms. We decided that the specific GO terms shuiold have a maxium of 7 genes whereas the general should have 2700. The choice of these values is trivial because we want a large diffrence between the two group, at the sime time that enough number of represnetatives in both gropups. Further more, if the groups are two large, then it will be very difficult to find genes that are exlcusive only of one of the groups. Then we sampled 200 genes that are excklusive from each group and we compared the number of edges that connect the genes of each of these sets.

Data

Analyses were carried with co-expression data from three species: yeast, humasn and chickens. In the case of yeast, the anlayses were also performed with ppi data as well as co-expression data. Data sources in the different cases are given in Table X in Appendix II-Data overview.

Most of the anlayses were carried in the 3 species and yeast ppi. However, some analysis were carried on yeast data because it was easily accessible, whereas some other analysis were carried on human as we though that chicken data would not become available and it resembles more the situation in chickens.

Results

Part 1- Impact of the different model parameters and network characteristics in the predictin performance using BMRF.

Using human and yeast data, we choose values for the differnet model parameters (Appendix I, tables 1-3). Then, using chicken data, we defined a network for this species. The conclusions were as follows:

In part2, we decided to use the different values of the different model parameters. A definition of the model parameters is given in Appendix I-Concepts.

-k:10 -20 replicates -30 iterations for the Gibbs-sampling -GO-size filter of 20 and 0.1, for minGOsize and maxGOsize, respectively -Using domain information as well as non-validated associations. For chicken data, we concluded that a peason correlation of 0.35 is the best cutoff, as it leads to the highest prediction perfomance. Thus, the chicken network will be extracted with pearson corr 0.35

Table 2: Correlations with AUC_comparison species

(i) Using human data

- Using domain information increases the overall performance considerably (0.043 higher AUC). Thus, domain information will be used
- The filters of number of GO terms does not seem to have an effect. However, this may be the case only in those cases in which the number of GO terms remains high.
- Increasing the number of folds in the k-validation from 2 to 5 or 10 leads to an improvement of AUC of 0.026 and 0.037, respectively. K:10 will be used whenever possible.
- Due to time constrains, we will apply the filterminGOsize:20 and maxGOsize:0.1. However, the anlaysis could be extended to GO terms withminGOsize>9.

(ii) Using yeast data

- The GO-size filters affect the overall prediction perfoamnce. We observed that as the filter form ninGOsize becomes more strict and the filter for mxGOsiuze beceome less strict, the overall predeiction performance increases.
- the GO-size filter seem toi cause e some alterations at the level of individual GO terms. In line with this, the preditiocn of one GO term are not enterlly indeoendt of the other GO terms considerd in the analysis.
- We will perform the anlaysis with 20 replicates
- Subsetting...

(iii) Using chicken data

- A peasin correlation of 0.35 will be considered

(iv) Other conlcussions.

- Non-validated data seem to be more reliable for yeast than for humans.

With the purpose of identifying which GO-term-properties have a more direct effect on the predictions performance, we computed the correlation between AUC and the different GO-term properties (Results)

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

-: No significant

Table 3: Correlations with AUC_comparison species

spec and AUC as expected. Depth is not good measure

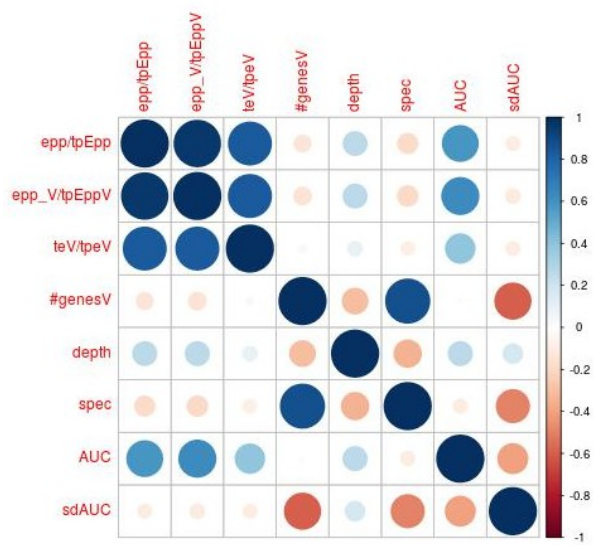
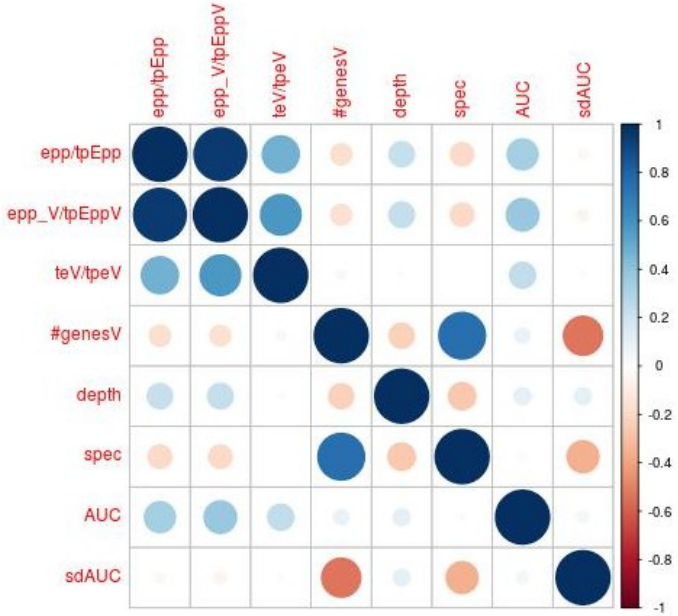


Illustration 5: Correlations yeast



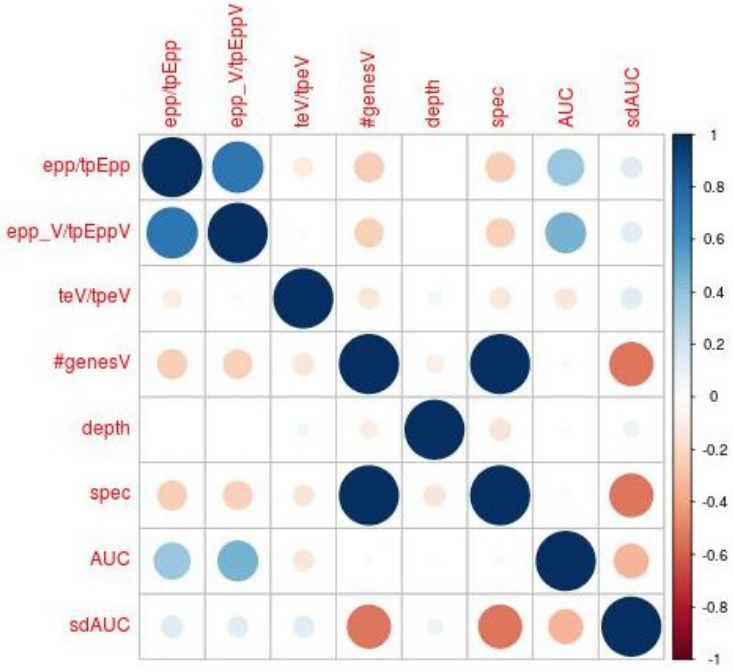


Illustration 7: Correlations humans

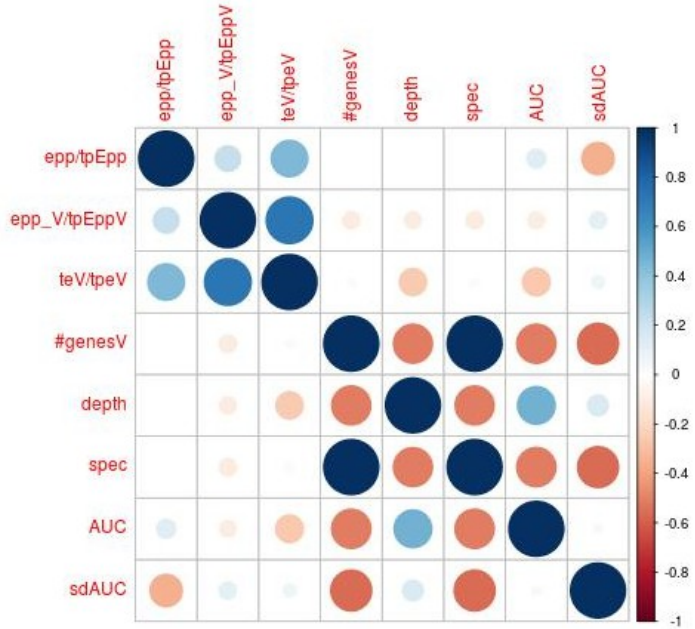


Illustration 6: Correlations chickens

From atble 4 we learn that only epp/tpepp and sd seem to affect the prediction performance. Epp/tpepp shows a favorable correlation with AUC, meaning that for GO terms whose associated genes are interconnected in the network (coexpressed), the method has more chances to distinguish genes that are associated associated with the GO term from genes that are not.

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

From Table 9 we conclude that some parameters that we may have considered important, such as #epp, depth or #lables are in fact not related to AUC. Epp/tpepp at the level of individual GO-terms seems to be a good indicator of the prediction performance.

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

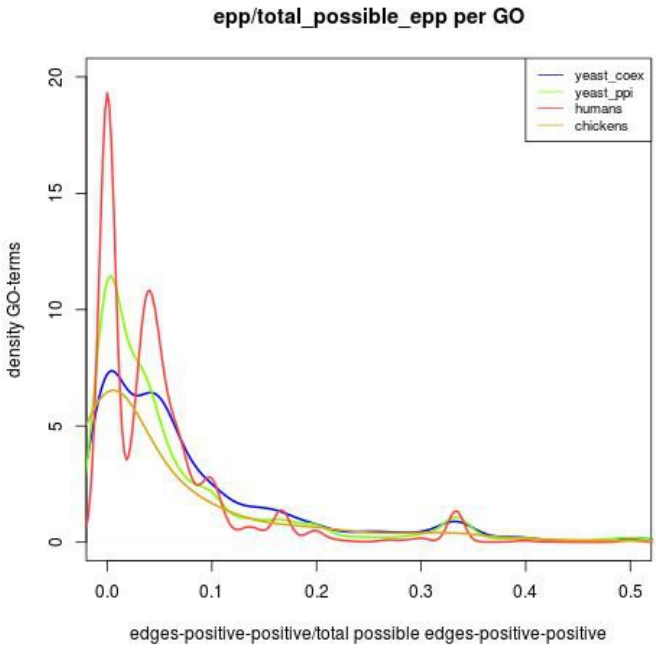


Illustration 8: epp/tpepp. Value sin table x appendix I

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

Table 4: average epp/tpepp for the different species

Then, we investigated which characteristics in the network data have a larger impact of the prediction performance. For this, we used data from different species, since the network data differs considerably between these (Tables 7-9 in Appendix I). For a more detailed description of the differences in data between the species, as well as the data sources, see Tables 4-7 and Illustraions 1-3 in Appendix I.

The main conclusions from part 1b are:

- epp/tpepp seem to be the network parameter with a larger impact in the prediction performance.
- The epp/tpepp standirez is very smilar for the species considered.
- Reducing Epn increases the perfomance
- Approaches like Positive-Unlabelled learnig can improve these parameter.

Part 2 – BMRF results

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

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

Table 5: Overall prediction performance for the different species using BMRF

Predictions are more accuracet for yeast, thne for chickens, yeast_ppi and finally for humans. The fact that we achieve better prediction performance for chickens than for yeast_ppi, is most probably linked to the fact that a much lower numbner of GO terms are predicted when the chciken data was used, than in the other cases. It may be the case that the GO terms that wre predicted with chcikens data are more easy to opredict than the overal set of GO terms that are predicted in yeast, yeast_ppi and humans.

Teherfore, we aimed to compare the prediction performance of the GO terms tahte wer epredicted in the 4 cases. We observed that of the 138 GO terms predicted with chicken data, 20, were also predicted in the other 3 cases. Illustraion 7 shows, a comparison of the predictio perfomance using data from the differnet species:

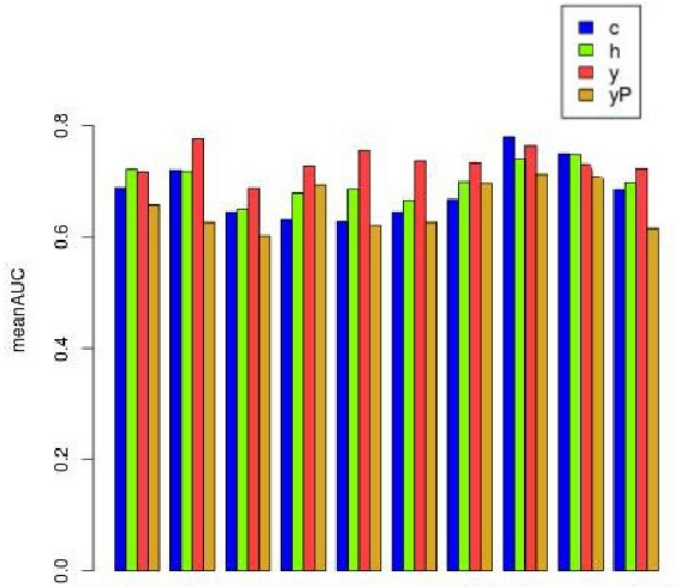


Illustration 9: GO_0009605,GO_0009653,GO_0009719,GO_0009892,GO_0009966,GO_0010033,GO_0010605,GO_0010628,GO_0010629,GO_0030154

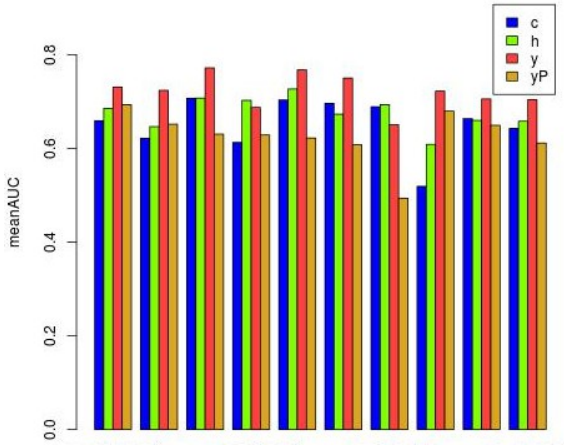


Illustration 10: GO_0031324,GO_0032879,GO_0048468,GO_0048584,GO_0048646,GO_0048869,GO_0050793,GO_0051128,GO_0070887,GO_1901700

Illustration 7 revealed that the prediction perfoamnce is generally larger for yeast, the n for humasn, the chiocksn and finally yeast_ppi. Then mean AUC (and sd) were, 0.729(0.03), 0.688(0.03), 0.668(0.05) and 0.642(0.05), respectivelly for the 4 species.

The fact that predictions are still high for chcikens wnev though a pearson correlation of 0.35 was used instead of 0.7 in the other 3 cases, suggets that BMRF hasa better prediction performance when the network is extense even at the expesnes of a lower reliability of the edges.

Part 3- Results using PU-BMRF.

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

	Max # of RN extracted							
	1000	**2000**	**3000**	**4000**	**5000**	**6000**	**7000**	**8000**
Accuracy of extraction of RN	1	0.99	0.97	0.96	0.93	0.93	0.92	0.92
PFP AUC using PU-BMRF (sd)	0.723 (0.08)	0.75 (0.072)	0.758 (0.084)	0.751 (0.086)	0.725 (0.094)	0.728 (0.089)	0.716 (0.092)	0.701 (0.095)

Table 6: Accuracy in the two steps of PU-BMRF, for different choices of max #RN extracted. Results from 30 GO-terms

The prediction accuracy for these GO terms using BMRF was 0.706 (0.03). Thus, except when a maximum of 8000 RN were extracted, PU-BMRF outperformed BMRF. The maximum improvement (+0.052 AUC) was when a maximum of RN was set to 3000. This is also illustarted in Illustration 7.

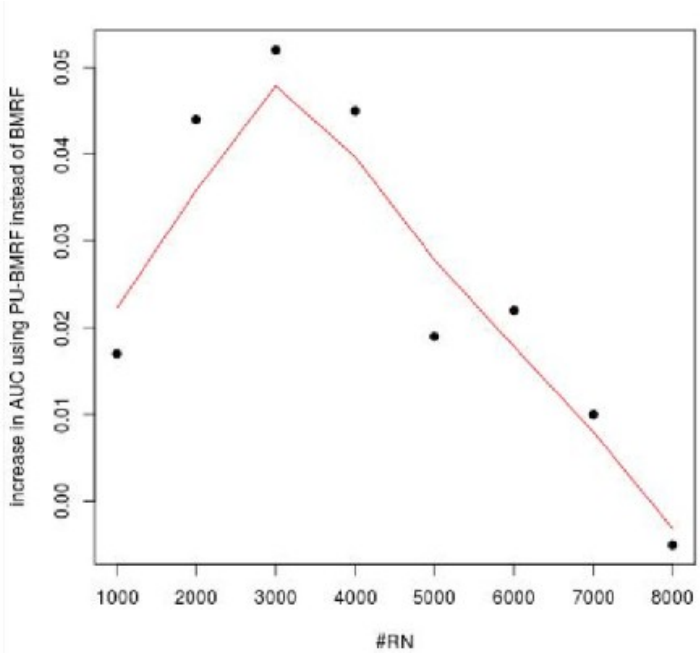


Illustration 11: Increase in AUC PU-BMRF vs BMRF, for different choices of max #RN extracted

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

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

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

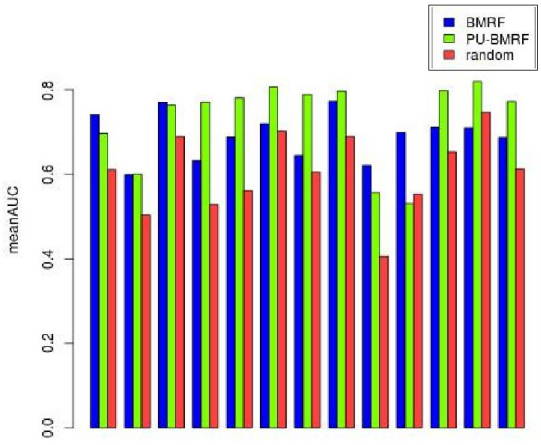


Illustration 12: GO_0006928,GO_0006950,GO_0007423, GO_0008283,GO_0009605,GO_0009653, GO_0009719,GO_0009887,GO_0032879, GO_0033554,GO_0040011,GO_0044699, GO_0044700

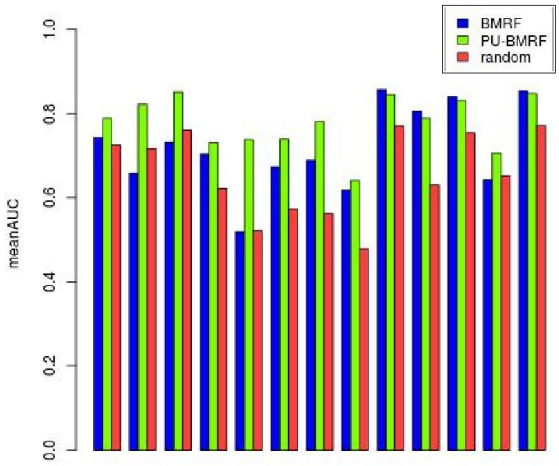
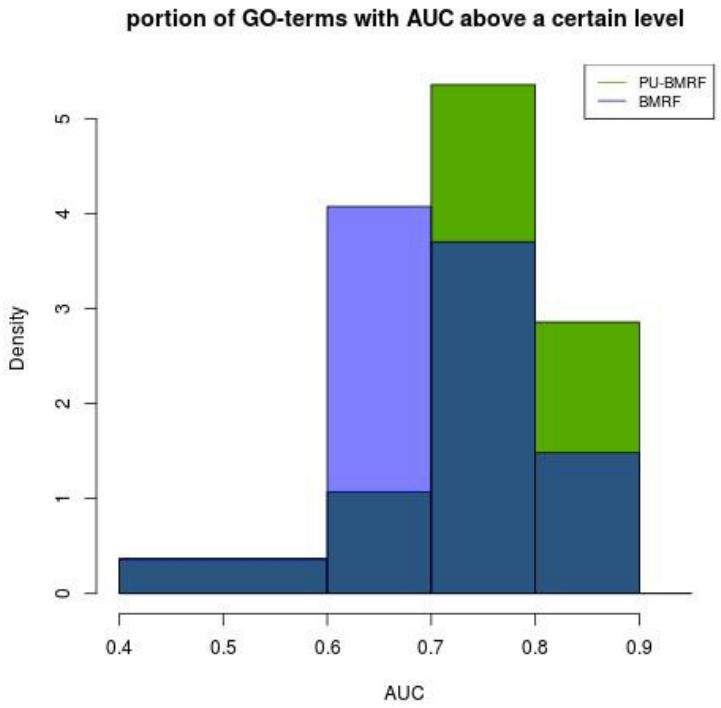


Illustration 13: GO_0044710,GO_0044763,GO_0044767, GO_0048646,GO_0051128,GO_0051240, GO_0060485,GO_0065008,GO_0097659, GO_1901360,GO_1901362,GO_1901700, GO_1903506

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



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

Part 4- <u>Biological support of the approach</u>

Plot also the specifricity. Put there a plot of delta

PU is more easy when high epp/tpepp, and epp/tpepp shows a negatoive corr with spec (-0.2)!, meaning that PU helps more in specif GO terms!! Also, PU is more easy when high epp/tpepp, so, iyt is more useful in pporly annotated species! (look at chcknes!).
Note that spec is actually “generaity”

What if you look only within GO terns with a minimum epp/tpepp. However (unfortunatelly) we obsered that the corr bt increase with PU and spec is poitive (not signifcant)

is there difference in how connected they are, as biology suggests? In part 2 of results

is that difference trsnalted to a better improvement in PU. Result snot ready
And/or does PU help more with more, less epp/tpepp and/or more/less speci

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)

However, we obsrved that th evalue of this correlation changes depnding on whether we include in th ecomputation of the cotrrelation Gotrems with a gven minGOsize. This can be due to the fact that epp/tpepp tends to be larger when more gens are considered. And it was when minGOsize was 2000, that the corre between spec and epp/tpepp took its highest value.In other worors, when we select GO terms that are general (above 2000 genes), is when we observed to a larger extend that the most specific functions have more interconnected genes than the most general ones. This makes, sense since when the number of positive genes is very lareg it ismore difficult that they all are coexpressed at the same time. And this can be observed to a higer degree when we extract from the corr those GO terms that having very low numbre of genes, they do not have the luck to be interconnected, and point as 0 in epp/tpepp

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

The increase in AUC was not significantly correlated with any of the variables consired. The highest correlation was with epp/tpepp (0.25), with a p-value of 0.21.

looking only ion gnes that are in specific vs only in general GO terms
1606 out of 10000 replicates were negative and on average the increase in the number of edges was 6.36%

Discussion

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

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

Lack of reprodubility

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

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

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

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

Factors that determine the prediction perfromance

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

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

Avenues of improvemnet

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

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

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

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

Conclusions

References

1.KOURMPETIS, Y. A. I., VAN DIJK, A. D. J., BINK, M., VAN HAM, R. & TER BRAAK, C. J. F. 2010. Bayesian Markov Random Field Analysis for Protein Function Prediction Based on Network Data. Plos One, 5.
2.SCHWIKOWSKI, B., UETZ, P. AND FIELDS, S.. 2000. A network of protein-protein interactions in yeast. Nature Biotechnology 18:1257 − 1261.
3.MINGHUA DENG, KUI ZHANG, SHIPRA MEHTA, TING CHEN & SUN., F. 2004. Prediction of Protein Function Using Protein–Protein Interaction Data. Journal of Computational Biology, 10, 6.
4.YAN, K. K., WANG, D. F., ROZOWSKY, J., ZHENG, H., CHENG, C. & GERSTEIN, M. 2014. OrthoClust: an orthology-based network framework for clustering data across multiple species. Genome Biology, 15.
5.JOACHIM W. BARGSTENA, EDOUARD I. SEVERINGB, J.-P. N., GABINO F. SANCHEZ-PEREZA & DIJK, A. D. J. V. 2014. Biological process annotation of proteins across the plant kingdom. Current Plant Biology, 1.
6.Consortium TF, Andersson L, Archibald AL, et al. Coordinated international action to accelerate genome-to-phenome with FAANG , the Functional Annotation of Animal Genomes project. Genome Biol. 2015:4-9. doi:10.1186/s13059-015-0622-4.
7.BHARDWAJ, N., GERSTEIN, M. & LU, H. 2010. Genome-wide sequence-based prediction of peripheral proteins using a novel semi-supervised learning technique. Bmc Bioinformatics, 11.
8.YANG, S. K. 2012. Positive-unlabeled learning for disease gene identification. Bioinformatics, 28, 2640-2647.
9.YOUNGS, N., PENFOLD-BROWN, D., BONNEAU, R. & SHASHA, D. 2014. Negative Example Selection for Protein Function Prediction: The NoGO Database. Plos Computational Biology, 10.10.YANG, P., LI, X. L., MEI, J. P., KWOH, C. K. &
10.JIANG, M. & CAO, J. Z. 2016. Positive-Unlabeled Learning for Pupylation Sites Prediction. Biomed Research International.
11.XIAOLI, LI & BING, LIU. 2003. IJCAI'03 Proceedings of the 18th international joint conference on Artificial intelligence. 587-592.
[12]**https://bmcgenomics.biomedcentral.com/articles/10.1186/1471-2164-14-13**
[13]https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1546-7
16. Sharan R, Ulitsky I, Shamir R (2007) Network-based prediction of protein function. Molecular Systems Biology 3: 1–13.R. SharanI. UlitskyR. Shamir2007Network-based prediction of protein function.Molecular Systems Biology3113
19. https://www.frontiersin.org/articles/10.3389/fpls.2016.00444/full
20. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3584181/f
21. http://msb.embopress.org/content/3/1/88.long