Noise tolerance of Multiple Classifier Systems in data integration-based gene function prediction

Matteo Rè and Giorgio Valentini

Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano
v. Comelico 39 Milano, Italy, `http://www.dsi.unimi.it`

Summary

The availability of various high-throughput experimental and computational methods developed in the last decade allowed molecular biologists to investigate the functions of genes at system level opening unprecedented research opportunities. Despite the automated prediction of genes functions could be included in the most difficult problems in bioinformatics, several recently published works showed that consistent improvements in prediction performances can be obtained by integrating heterogeneous data sources. Nevertheless, very few works have been dedicated to the investigation of the impact of noisy data on the prediction performances achievable by using data integration approaches.
In this contribution we investigated the tolerance of multiple classifier systems (MCS) to noisy data in gene function prediction experiments based on data integration methods. The experimental results show that performances of MCS do not undergo a significant decay when noisy data sets are added. In addition, we show that in this task MCS are competitive with kernel fusion, one of the most widely applied technique for data integration in gene function prediction problems.

1 Introduction

Recently, several works highlighted the central role played by data integration methods for the gene function prediction problem [14, 1, 5, 22]. Indeed it is well-known that each source of biomolecular data can reveal specific features of gene/gene products, and this information can be useful to characterize their functional role in living systems.

Gene function prediction in its general formulation is a complex classification problem characterized by the following items:

- each gene/gene product can be assigned to multiple terms/classes (a multiclass, multilabel classification problem)
- classes are structured according to a predefined hierarchy (a directed acyclic graph for the Gene Ontology (GO) [23] or a tree forest for FunCat [19]);
- classes are usually unbalanced (with positive examples usually less than negatives)
- gene labels are in several cases uncertain or largely incomplete
- multiple sources of data can be used to predict gene functions.

doi:10.2390/biecoll-jib-2010-139

Even if the integration of multiple sources of data has enjoyed a certain attention in the computational biology community [9, 1, 14], at best of our knowledge, only few works tried to evaluate the impact of noisy data in data integration methods for gene function prediction problems [11].

According to [14], the main supervised gene function prediction methods based on heterogeneous data integration proposed in the literature can be schematically subdivided in three main categories: functional linkage networks, vector subspace integration and kernel fusion methods. Modeling interactions between gene products using functional linkage networks is realized through graphs, where gene products are modeled as nodes and relationships between genes through edges [6, 1]. In vector space integration (VSI) different vectorial data are concatenated [3], while kernel methods, by exploiting the closure property with respect to the sum or other meaningful algebraic operators represent another valuable research direction for the integration of biomolecular data [9].

Quite surprisingly, as observed in [14], only little attention has been devoted to multiple classifier systems (MCS) as a mean to integrate multiple biomolecular sources of data for gene function prediction. To our knowledge only few works very recently considered ensemble methods in this specific bioinformatics context: Naive-Bayes integration of the outputs of SVMs trained with multiple sources of data [4], and logistic regression for combining the output of several SVMs trained with different data and kernels in order to produce probabilistic outputs corresponding to specific GO terms [15]. We recently demonstrated that simple ensemble methods can obtain results comparable with state-of-the-art data integration methods, exploiting at the same time the modularity and scalability that characterize most of the ensemble algorithms [17]. Indeed biomolecular data differing for their structural characteristics (e.g. sequences, vectors, graphs) can be easily integrated, because with ensemble methods the integration is performed at the decision level, combining the outputs produced by classifiers trained on different datasets. In other words ensemble methods scale well with the number of the available data sources, and problems that characterize other data fusion approaches are thus avoided.

In this context, the resistance of prediction methods to noise is a fundamental issue for at least two reasons. At first, it should be noticed that any single data set involved in a data integration experiment, aimed at predicting gene functions at a whole ontology level, is likely to represent noise in a consistent fraction of the nodes composing the functional hierarchy. Indeed, a data source that can be predictive for a functional class can be non-predictive for other classes, considering the extreme complexity of the most widely used functional ontologies (GO and FunCat), both composed by hundreds or thousands of nested terms. Moreover, most of the commonly used biomolecular data are noisy due to the machinery and the complex procedures involved in the generation of the data through high-throughput biotechnologies.

In this paper we study the noise tolerance of data integration methods in the context of supervised gene function prediction problems, with a particular focus on multiple classifier systems.

The paper is structured as follows. In Section 2 two MCS-based and a kernel fusion-based method for biomolecular data integration are presented. Section 3 and 4 show the experimental set-up and results relative to the analysis of noise tolerance of multiple classifier systems in two different gene function prediction problems involving the integration of multiple data sources. The results are discussed in Section 5 and the conclusions end the paper.

2 Methods

2.1 Multiple Classifier Systems

An ensemble system can integrate multiple sources of biomolecular data by training different learning machines on different ”views” of the data, and then by combining the outputs of the component learners. In this work we programmatically considered simple methods: Weighted majority voting and Decision Templates.

2.1.1 Weighted majority voting

It is well-known, at least from the XVIII century, that the judgment of a committee is superior to those of individuals, provided the individuals have reasonable competence (*Condorcet Jury Theorem* [2]). *Weighted majority voting* ensembles are ensemble methods based on the cited theorem. They simply integrate multiple sources of data by aggregating classifiers trained on multiple ”views” of the data, and by weighting the decision of each base classifier on the basis of its ”competence” on the learned data.

More precisely, given a set of k classes whose labels $\omega_j \in \Omega$, $1 \leq j \leq k$, we denote by $d_{t,j}(x) \in [0,1]$ the support (e.g. the probability) estimated by the base t^{th} classifier of an ensemble composed by L base learners, that a given example x belongs to the class ω_j. For brevity we denote $d_{t,j}(x)$ as $d_{t,j}$. A simple way to integrate different data sources is represented by the weighed linear combination rule [7], by which the posterior probability $\hat{P}$ of the resulting ensemble is estimated as follows:

$$\hat{P}(\omega_j|x) = \sum_{t=1}^{L} w_t d_{t,j}(x) \tag{1}$$

Considering that gene classes are largely unbalanced (positive examples are largely less than negative ones), we chose the F-measure to compute the weights:

$$w_t = \frac{F_t}{\sum_{t=1}^{L} F_t} \tag{2}$$

where F_t is the F-measure assessed on the training data for the t^{th} base learner, and the w_t weights are obtained by a linear combination of the F-measures. The decision $D_j(x)$ of the ensemble about the class ω_j is taken using the estimated probability $\hat{P}$ (eq. 1):

$$D_j(x) = \begin{cases} 1, & \text{if } \hat{P}(\omega_j|x) > 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where output 1 correspond to positive predictions for ω_j and 0 to negatives.

2.1.2 Decision Templates

The main idea behind Decision Templates consists in the the comparison of a ”prototypical answer” of the ensemble for the examples belonging to a given class (the template) with the current answer of the ensemble to a specific example whose class needs to be predicted (the decision profile) [8]. The decision profile DP(**x**) for an instance **x** is a matrix composed by $d_{t,j}$ ∈[0,1] elements representing the support (e.g. the probability) given by the t^{th} classifier to class ω_j. Decision templates DT_j are the averaged decision profiles obtained from $\mathbf{X}_j$, the set of training instances belonging to the class ω_j:

$$DT_j = \frac{1}{|\mathbf{X}_j|} \sum_{\mathbf{x} \in \mathbf{X}_j} DP(\mathbf{x}) \tag{4}$$

By computing the similarity $\mathcal{S}$ between $DP(\mathbf{x})$ and the decision template DT_j for each class ω_j, from a set of c classes, the final decision of the ensemble is taken by assigning a test instance **x** to a class with the largest similarity [8]:

$$D(\mathbf{x}) = \arg\max_j \mathcal{S}_j(\mathbf{x}) \tag{5}$$

In multi-class classification problems a decision template can be represented through a matrix with a number of columns equal to the number of classes, but it is easy to see that with dichotomic problems the decision templates are reduced to one-column matrices (since $DT_1(t, 2) = 1 - DT_1(t, 1)$), where $DT_1(t, y), y \in \{1, 2\}$ represents the element (t, y) of the decision template for class 1.

The similarity ($\mathcal{S}_1$) for the positive class and the similarity ($\mathcal{S}_2$) for the negative class can be computed as 1 minus the normalized squared euclidean distance:

$$\begin{aligned} \mathcal{S}_1(\mathbf{x}) &= 1 - \frac{1}{n} \sum_{t=1}^{n} [DT_1(t, 1) - d_{t,1}(\mathbf{x})]^2 && (6) \\ \mathcal{S}_2(\mathbf{x}) &= 1 - \frac{1}{n} \sum_{t=1}^{n} [DT_2(t, 1) - d_{t,1}(\mathbf{x})]^2 && (7) \end{aligned}$$

where DT_1 is the decision template for the positive and DT_2 for the negative class. The final decision of the ensemble is:

$$D(\mathbf{x}) = \arg\max_{\{1,2\}} (\mathcal{S}_1(\mathbf{x}), \mathcal{S}_2(\mathbf{x})) \tag{8}$$

2.2 Kernel fusion methods

Even if kernel functions are widely adopted in machine learning because they can be used with standard algorithms (i.e. SVM) to solve binary and multiclass classification problems, they can also be thought as a way to encode similarities among non-vectorial and heterogeneous data. Indeed as long as our datasets can be represented as a square kernel matrix, then any kernel method can be applied to the data. A key feature of kernels making them very attractive in problems requiring the integration of heterogeneous data is that their mathematics allow us to derive new kernels by combining two or more kernels each representing diverse types of data. Many algebraic operators are closed under positive semidefiniteness. Among them, the most important operation, w.r.t. data integration problems, is the sum: if K_1 and K_2 are valid kernel functions, then we can demonstrate that $K(x, y) = K_1(x, y) + K_2(x, y)$ is still a valid kernel.

Also, for positive coefficients, a weighted combination of kernels $(\mu_1 K_1(x,y) + \mu_2 K_2(x,y)$ preserves the Mercer's conditions [13].

Despite kernel combination methods have been successfully applied in a broad range of bioinformatics problems, the most effective way to compute the weights involved in the kernel fusion is still an open question. Quite interestingly, a recent work pointed out that a kernel fusion obtained simply by averaging the kernel matrices associated to each data source is able to perform equally well (if not better) than more sophisticated methods (such as multiple kernel methods based on semidefinite programming (SDP) [9]) that introduces weights on each kernel matrix [11]. In this work we compared the performances of less computationally intensive data integration methods, the ensemble systems, with performances achievable by a kernel fusion based on the average of kernel matrices.

3 Experimental setup

3.1 Data sets and functional labeling

Considering that a recent study showed that the fraction of genes annotated with experimental evidence in S.*cerevisiae* is one of the largest among many model organisms [18], we chose the yeast for our experiments. For our experiments we considered two groups of data sets: a) a small set composed by protein sequence and protein structural data, previously analyzed in [11]; b) a larger set that includes six data sets previously collected in [17] (see Tab. 1 in the supplementary materials for details about these data).

Sequence and structural data have been labeled according to the GO term prediction benchmark, originally proposed in [11]. It involves 5323 yeast genes labeled according to 56 GO classes: 27, 22 and 7 of them belong to the MF, BP and CC ontologies respectively. The number of positives genes in the considered functional class ranges from 101 to 516. The negatives sets in the training sets were constructed in order to equal the number of positives examples, using the same strategy proposed in [11] For the second group of data we labeled the genes with the 15 top-level FunCat functional classes (see Tab. 2 in the supplementary materials for more details). For this second task, considering that the classes are less unbalanced, we use all the available negative examples.

3.2 Experiments with noisy data

We performed two sets of experiments, respectively using the first and the second group of biomolecular data sets (Section 3.1). In both cases we analyzed the tolerance of ensemble systems and kernel fusion methods to noisy data, by adding to the original data noisy kernel matrices.

3.2.1 Experiments with sequence and structural data

The performances of the evaluated methods for the prediction of any of the 56 considered GO terms have been evaluated using a 5-fold cross validation scheme repeated three times as in [11] in order to allow a direct comparison. We then produced two noisy kernels by a random permutation of the rows and the corresponding columns of the structure kernel matrix. These two noisy kernels have been used to train two additional component classifiers and to evaluate the performances of data integration methods. At first, we trained two SVMs using the original no-noisy data: protein sequence similarities encoded through a mismatch kernel [10] and protein structure similarities encoded through an empirical kernel map [20]. In particular the structural alignments used in the construction of the structure kernel have been obtained using the MAMMOTH structural aligner [16]. The E-values produced by MAMMOTH have been converted in a valid kernel matrix using an empirical kernel map. Prior to integration, each kernel is centered around the origin in the feature space and each example is projected onto the unit sphere using a cosine normalization. We integrated the two kernel matrices by using the simple average kernel fusion method, an ensemble system based on the weighted average (using linear weights based on the F-scores obtained by the SVM associated to the structure and mismatch kernels), and a second ensemble that does not require the definition of explicit weights (the Decision Templates combiner, Sect. 2).

Then we analyzed the resistance of ensemble systems to noisy data. We tested the average kernel fusion (KF), the weighted average ensemble and the Decision Templates ensemble for the integration of the structure and mismatch kernels adding one or two noisy kernels. Following the experimental setup of [11] we used, as component classifiers for the ensemble systems, linear SVMs with a fixed C regularization parameter set to 10. To compare our results with those published in [11], we computed the AUC scores for each prediction task, averaged across the 15 splits resulting from the 5-fold cross-validation repeated 3 times.

3.2.2 Experiments with six different types of biomolecular data

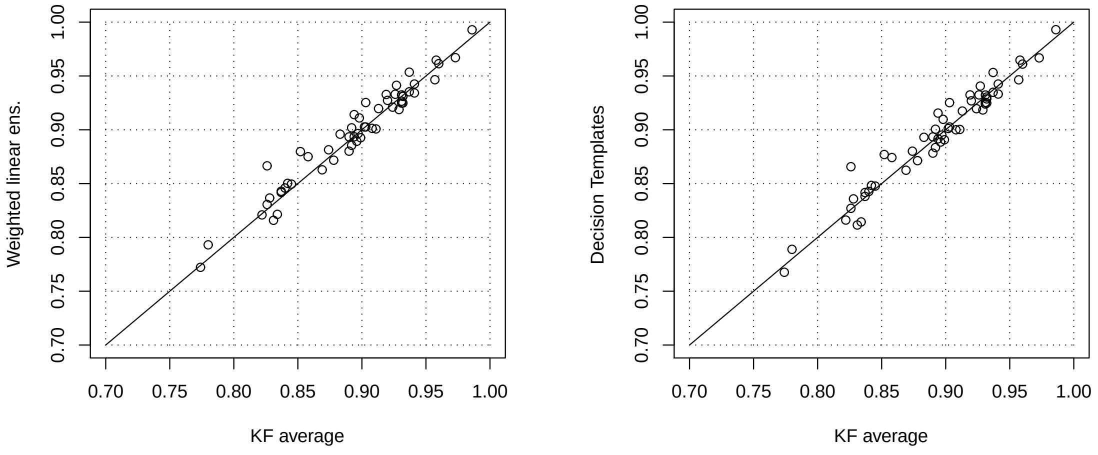
In these experiments we tested the tolerance to noise of MCS using a larger set of data. For each available data set, we generated a noisy copy by applying the same procedures described in Section 3.2.1, and we compared the performances of ensemble methods and KF using: a) the six original no-noisy data sets described in Section 3.1; b) the same six no-noisy data plus six noisy copies of all the available data. We evaluated the performances on 15 top-level FunCat classes using 5-fold cross-validation techniques. We used as base learners linear probabilistic SVMs [12] with a fixed C regularization parameter set to 10.

4 Results

4.1 Results with sequence and structural data

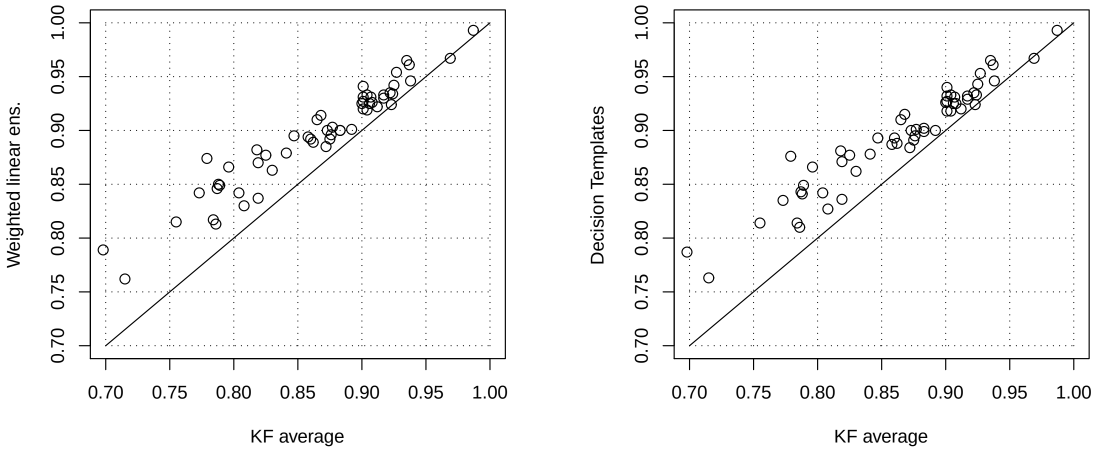
Fig. 1 shows the comparative results, in absence of noisy data, obtained by the average kernel fusion, the weighted average ensemble and the decision template combiner for the integration of the structure and mismatch kernels. Each point in the graph represents the AUC results of one of the 56 classification tasks: Fig. 1 (a) represents KF vs. weighted average ensembles, and Fig. 1 (b) KF vs. decision templates.

Fig. 2 and 3 compare the AUC results achieved by adding respectively one and two noisy data sets. In both cases KF average methods are compared with weighted linear and decision tem-



(a) (b)

Figure 1: Comparison of AUC results between average kernel fusion and ensemble data integration methods without noisy data. Points represent the AUC score of kernel fusion (abscissa) and ensemble (ordinate) methods for 56 GO terms. (a) Average KF vs Weighted ensembles (b) Average KF vs Decision Templates.



(a) (b)

Figure 2: Comparison of AUC results between average kernel fusion and ensemble data integration methods with 1 noisy data set. Points represent the AUC score of kernel fusion (abscissa) and ensemble (ordinate) methods for 56 GO terms. (a) Average KF vs Weighted ensembles (b) Average KF vs Decision Templates.

plate ensembles. Detailed results for each GO term are shown in Table 3 in the Supplementary Information.

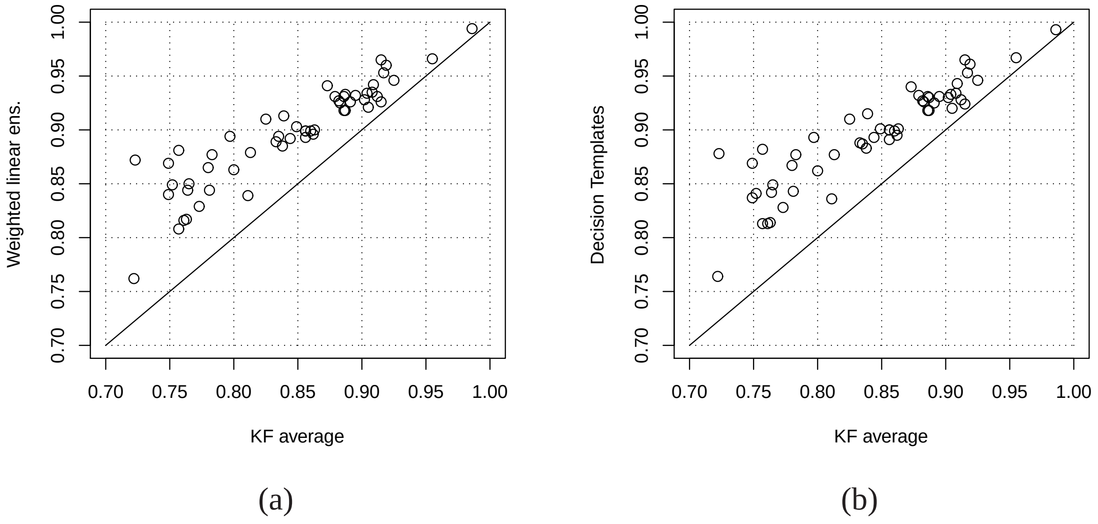


Figure 3: Comparison of AUC results between average kernel fusion and ensemble data integration methods with 2 noisy data sets. Points represent the AUC score of kernel fusion (abscissa) and ensemble (ordinate) methods for 56 GO terms. (a) Average KF vs Weighted ensembles (b) Average KF vs Decision Templates.

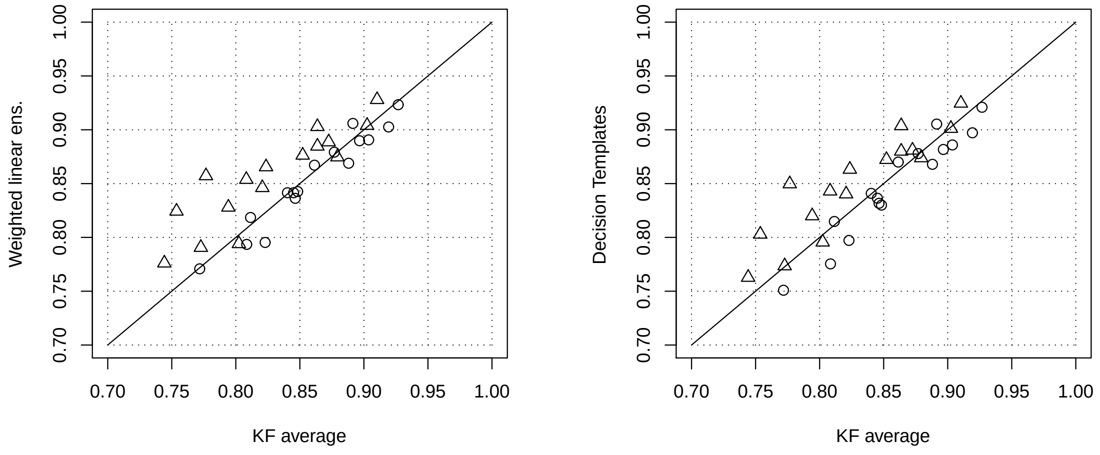
4.2 Results with six different types of biomolecular data

Figure 4 presents in a synthetic way the results obtained with no-noisy and noisy data by average KF and the ensemble methods described in Section 2. Each point represents the AUC score for a specific FunCat class, achieved with no-noisy (circular points) and noisy data (triangular points) using KF and respectively weighted linear (Figure 4 (a)) and decision templates (Figure 4 (b)) ensembles. Full results for each ensemble, kernel fusion and single SVMs trained on a single source of data for each FunCat class are available in Table 4 in the Supplementary Information.

5 Discussion

Considering no noisy data integration, the performances obtained by the weighted average ensembles appear comparable with those achieved by the average kernel fusion (Fig. 1(a) and Fig. 4 (a)), while decision templates achieve comparable (Fig. 1 (b)) or slightly worse (Fig. 4 (b)) performances w.r.t. KF. These visual clues are confirmed by the Wilcoxon signed-ranks test [24]: at 0.01 significance level no difference can be registered between weighted linear and KF methods for both the considered tasks, while a significant difference can be reported in favour of KF with respect to decision templates only for the second task (integration of the six original no-noisy data sets).

The AUC scores obtained by the ensemble and kernel fusion methods when one noisy kernel is added to the structure and mismatch kernels are compared in Fig. 2. Looking at the data plotted in Fig. 2, it is clear that, in presence of one noisy kernel, the performances achieved by the ensemble systems are higher than the ones obtained by using the average KF approach.



(a) (b)

Figure 4: Experiments with 6 no-noisy and 6 noisy data sets. Points represent the AUC score of kernel fusion (abscissa) and ensemble (ordinate) methods for 15 FunCat functional classes. Circular points: results with no-noisy data. Triangular points: results with noisy data. (a) Average KF vs Weighted ensembles (b) Average KF vs Decision Templates.

Most of the points lie above the bisector in Fig. 2 (a) and (b), showing that both Weighted linear and Decision Templates ensembles tend to outperform the average based kernel fusion. Using the Wilcoxon signed-ranks test we register a significant difference in favour of Weighted linear and Decision Template w.r.t. average KF (*p-value* $\simeq 4.30 \cdot 10^{-11}$). The resistance to noise of the tested ensemble methods is also confirmed by the results collected after the inclusion in the data sources collection of the second noisy kernel (Fig. 3). In this extreme test in which half of the considered datasets are merely representing noise, the AUC averaged across both the cross validation splits and the GO functional classes is 0.895, 0.894 and 0.841 for the weighted linear ensemble, the decision templates ensemble and the average kernel fusion respectively. Again the Wilcoxon signed-ranks test shows a significant difference in favour of Weighted linear and Decision Template w.r.t. average KF (*p-value* $\simeq 3.87 \cdot 10^{-11}$). These results indicate that ensemble systems are more robust to noisy data than the KF method.

The experiments with the larger group of biomolecular data sets enforce these findings. Indeed, while the results between the different integration methods are quite comparable with the no-noisy data (circular points, Fig. 4), when the six noisy data sets are added to the original data, ensemble methods clearly outperform KF (triangular points, Fig. 4). Interestingly enough, it seems that adding noisy data sets does not significantly worsen the performances of weighted linear ensembles (*p-value* $\simeq 0.803$) and decision templates (*p-value* $\simeq 0.561$), while a significant decrement can can be detected in KF (*p-value* $\simeq 1.52 \cdot 10^{-5}$).

These results, at least for weighted linear ensembles, can be explained by considering that these methods are able to learn which are the noisy data and penalize them by assigning low weights to base classifiers trained with noisy data. From this standpoint, kernel fusion methods based on semi-definite or semi-infinite programming techniques [21] could be considered for future experiments, because they also are able to learn weights from the data.

6 Conclusions and future work

The impact of noisy data in the performances of data integration methods for gene function prediction is of paramount importance for at least two reasons. At first, most of the available data from high throughput biotechnologies are inherently noisy. At second, the same source of data can be predictive for certain functional classes, but can be non-predictive for others: from this standpoint a certain type of data can be considered ”noisy” for the prediction of certain subsets of terms in the GO or FunCat hierarchy.

In this work we evaluated the performances of simple ensemble systems in data integration based gene function prediction problems with an without noisy data. We compared their results with those obtained using state-of-the-art methods and previously published datasets. We found that, in these empirical benchmarks based on both real data and artificially generated noisy datasets, the ensemble systems are able to perform better than average kernel fusion systems, one of the most widely applied data integration methods. It is worth noting that the tolerance of ensemble methods to noisy data sets is confirmed for the integration of both a small and a relatively large number of different sources of biomolecular data. Future research could try to evaluate the impact of an even large number of noisy data taking into account also the hierarchical structure of gene functional classes.

Considering that ensemble systems are often less computationally intensive than other data integration approaches and that they perform data integration at decision level, avoiding all the potential problems affecting experiments aimed at integrating structurally different types of data, we believe that this class of data integration approaches constitutes a valuable research line in data integration based gene function prediction.

Motivated by these promising results, we plan to experiment with other more refined ensemble systems based on meta-learning or boosting techniques to integrate multiple sources of noisy data and to compare ensemble methods with other Multi-Kernel learning algorithms that showed tolerance to noisy data [11].

Another interesting research item could be the evaluation of the resistance to noisy datasets by gradually varying the amount of artificial noise. In this way we could characterize at a more refined level the response of data integration methods to noise, and we could better understand the behaviour of these methods in real-world whole-genome and whole-ontology gene function prediction experiments.

Acknowledgements

We would like to thank the reviewers for their comments. The authors gratefully acknowledge partial support by Università degli Studi di Milano (PUR project ATE-2009-0356, ”Computational methods for biomedical pattern analysis”) and by PASCAL2 Network of Excellence under EC grant no. 216886. This publication only reflects the authors’ views.

References

[1] H.N. Chua, W. Sung, and L. Wong. An efficient strategy for extensive integration of diverse biological data for protein function prediction. *Bioinformatics*, 23(24):3364–3373, 2007.

[2] N.C. de Condorcet. *Essai sur l' application de l' analyse à la probabilité des decisions rendues à la pluralité des voix*. Imprimerie Royale, Paris, 1785.

[3] M. desJardins, P. Karp, M. Krummenacker, T.J. Lee, and C.A. Ouzounis. Prediction of enzyme classification from protein sequence without the use of sequence similarity. In *Proc. of the 5th ISMB*, pages 92–99. AAAI Press, 1997.

[4] Y Guan, C.L. Myers, D.C. Hess, Z. Barutcuoglu, A. Caudy, and O.G. Troyanskaya. Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biology*, 9(S2), 2008.

[5] X. Jiang, N. Nariai, M. Steffen, S. Kasif, and E. Kolaczyk. Integration of relational and hierarchical network information for protein function prediction. *BMC Bioinformatics*, 9(350), 2008.

[6] U. Karaoz et al. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc. Natl Acad. Sci. USA*, 101:2888–2893, 2004.

[7] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[8] L.I. Kuncheva, J.C. Bezdek, and R.P.W. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2):299–314, 2001.

[9] G.R. Lanckriet, T. De Bie, N. Cristianini, M. Jordan, and W.S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20:2626–2635, 2004.

[10] C. Leslie et al. Mismatch string kernels for svm protein classification. In *Advances in Neural Information Processing Systems*, pages 1441–1448, Cambridge, MA, 2003. MIT Press.

[11] D.P. Lewis, T. Jebara, and W.S. Noble. Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. *Bioinformatics*, 22(22):2753–2760, 2006.

[12] H.T. Lin, C.J. Lin, and R.C. Weng. A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, 68:267–276, 2007.

[13] J. Mercer. Function of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, 209:415–446, 1909.

[14] W.S. Noble and A. Ben-Hur. Integrating information for protein function prediction. In T. Lengauer, editor, *Bioinformatics - From Genomes to Therapies*, volume 3, pages 1297–1314. Wiley-VCH, 2007.

[15] G. Obozinski, G. Lanckriet, C. Grant, Jordan. M., and W.S. Noble. Consistent probabilistic output for protein function prediction. *Genome Biology*, 9(S6), 2008.

[16] A.R. Ortiz et al. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci.*, 11:2606–2621, 2002.

[17] M. Re and G. Valentini. Simple ensemble methods are competitive with state-of-the-art data integration methods for gene function prediction. *Journal of Machine Learning Research, W&C Proceedings*, 2010. (in press).

[18] S.Y Rhee et al. Use and misuse of the gene ontology annotations. *Nature Rev. Genetics*, 9(17-18):509–515, 2008.

[19] A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Guldener, G. Mannhaupt, M. Munsterkotter, and H.W. Mewes. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32(18):5539–5545, 2004.

[20] B. Scholkopf, K. Tsuda, and J.P. Vert. *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA, 2004.

[21] S. Sonnenburg, G. Ratsch, C. Schafer, and B. Scholkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[22] I. Tetko, I. Rodchenkov, M. Walter, T. Rattei, and H. Mewes. Beyond the 'best' match: machine learning annotation of protein sequences by integration of different sources of information. *Bioinformatics*, 24(5):621–628, 2008.

[23] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genet.*, 25:25–29, 2000.

[24] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.

Noise tolerance of Multiple Classifier Systems in data integration-based gene function prediction: Supplementary Information

Matteo Rè and Giorgio Valentini

Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano
v. Comelico 39 Milano, Italy, `http://www.dsi.unimi.it`

The supplementary material provides additional information about the data sets used in the experiments and detailed gene function prediction results relative to the 56 GO terms and 15 FunCat classes considered in two sets of experiments.

1 Data sets

In the second set of experiments, we predicted the top-level 15 functional classes of the FunCat taxonomy [4] of the model organism *S. cerevisiae*, using 6 different sources of data (Tab. 1). We considered yeast genes common to all data sets (about 1900) and with at least 1 FunCat annotation. We also removed from the list of the target functional classes all those represented by less than 20 genes. This corresponds to restrict our classifications to only 15 FunCat classes (Tab. 2) In other words, we selected the roots of the trees of the FunCat forest (that is the most general and wide functional classes of the overall FunCat taxonomy). Table 2 provides a brief description of the FunCat classes predicted in the experiments. The first column corresponds to the FunCat Identifier of the functional class.

2 Supplementary results

Performance relative to the first set of experiments in presence of artificial noise are reported in Tab. 3. The first three columns are dedicated to the Gene Ontology term ID, the ontology type of the GO term and the number of examples associated to the functional term respectively. Columns 4, 6 and 8 report the averaged AUCs obtained in the 15 folds by integrating the structured, mismatch kernels and a single noisy kernel using the average kernel fusion, the weighted average and decision templates integration methods respectively. Columns 5, 7 and 9 contains the results obtained in the same test repeated by including the second noisy kernel during data integration. Standard deviation results for each classification task are reported in parenthesis.

Table 4 shows the detailed results relative to the 15 FunCat classes considered in the second set of experiments. Results for both single SVMs trained on single sources of data, and kernel fusion (KF), weighted linear and decision template ensembles, with and without noisy data are provided. Note that SVMs trained with a single source of data achieve AUC scores only slightly larger than random guessing. Data integration methods always outperform the best single source classifiers (best results for each FunCat class are highlighted in boldface). Weighted linear and decision templates do not significantly worsen their performances with noisy data, while kernel fusion data integration undergoes a certain decrement of AUC scores.

doi:10.2390/biecoll-jib-2010-139

Table 1: Datasets

Code	Dataset	n.genes	n.features	description
K_{string}	PPI - STRING	2338	2559	protein-protein interaction data from [6]
K_{BG}	PPI - BioGRID	4531	5367	protein-protein interaction data from the *BioGRID* database [5]
K_{pfam1}	Protein domain log-E	3529	5724	Pfam protein domains with log E-values computed by the *HMMER* software toolkit [1]
K_{pfam2}	Protein domain binary	3529	4950	protein domains obtained from *Pfam* database [2]
K_{expr}	Gene expression	4532	250	merged data of Spellman and Gasch experiments [3]
K_{seq}	Pairwise similarity	3527	6349	Smith and Waterman log-E values between all pairs of yeast sequences

Table 2: FunCat classes

Code	**Description**
01	Metabolism
02	Energy
10	Cell cycle and DNA processing
11	Transcription
12	Protein synthesis
14	Protein fate
16	Protein with binding function or cofactor requirement
18	Regulation of metabolism and protein function
20	Cellular transport and transport routes
30	Cellular communication/Signal transduction mechanism
32	Cell rescue, defense and virulence
34	Interaction with the environment
40	Cell fate
42	Biogenesis of cellular components
43	Cell type differentiation

Table 3: Performances of data integration methods in presence of one and two synthetic noisy kernels (AUC averaged across the 15 folds). KF-noisy1 and KF-noisy2 stands for kernel fusion methods with 1 and 2 noisy kernels; Wens-noisy1 and Wens-noisy2 stands for weighted average ensembles with 1 and 2 noisy kernels; DT-noisy1 and DT-noisy2 for decision templates with 1 and 2 noisy kernels.

GO$_{term}$	Ont	#	KF-noisy1	KF-noisy2	Wens-noisy1	Wens-noisy2	DT-noisy1	DT-noisy2
GO:0008168	MF	108	0.922(0.035)	0.908(0.040)	0.935(0.025)	0.935(0.025)	0.935(0.026)	0.934(0.026)
GO:0005506	MF	129	0.901(0.040)	0.873(0.045)	0.941(0.031)	0.941(0.031)	0.940(0.030)	0.940(0.029)
GO:0006260	BP	109	0.819(0.060)	0.749(0.050)	0.870(0.066)	0.869(0.065)	0.871(0.067)	0.869(0.069)
GO:0048037	MF	118	0.892(0.037)	0.860(0.038)	0.901(0.042)	0.899(0.042)	0.900(0.041)	0.899(0.042)
GO:0046483	BP	128	0.927(0.025)	0.917(0.027)	0.954(0.016)	0.953(0.016)	0.953(0.016)	0953(0.016)
GO:0044255	BP	101	0.818(0.051)	0.757(0.049)	0.882(0.059)	0.881(0.059)	0.881(0.058)	0.882(0.058)
GO:0016853	MF	124	0.773(0.073)	0.749(0.069)	0.842(0.072)	0.840(0.072)	0.835(0.073)	0.837(0.073)
GO:0044262	BP	209	0.883(0.032)	0.856(0.029)	0.900(0.031)	0.899(0.031)	0.899(0.030)	0.900(0.030)
GO:0009117	BP	124	0.825(0.058)	0.783(0.070)	0.877(0.045)	0.877(0.045)	0.877(0.044)	0.877(0.046)
GO:0016829	MF	201	0.901(0.022)	0.879(0.022)	0.931(0.021)	0.931(0.021)	0.932(0.021)	0.932(0.021)
GO:0016779	MF	142	0.830(0.077)	0.800(0.075)	0.863(0.075)	0.863(0.075)	0.862(0.074)	0.862(0.073)
GO:0016043	BP	106	0.779(0.082)	0.723(0.083)	0.874(0.052)	0.872(0.055)	0.876(0.050)	0.878(0.049)
GO:0008270	MF	234	0.862(0.028)	0.833(0.035)	0.889(0.026)	0.889(0.026)	0.888(0.026)	0.888(0.026)
GO:0006066	BP	111	0.917(0.034)	0.887(0.027)	0.933(0.033)	0.933(0.033)	0.932(0.034)	0.930(0.033)
GO:0003723	MF	212	0.847(0.036)	0.797(0.047)	0.895(0.037)	0.894(0.038)	0.893(0.038)	0.893(0.039)
GO:0004518	MF	125	0.786(0.073)	0.761(0.076)	0.813(0.058)	0.816(0.058)	0.810(0.056)	0.813(0.057)
GO:0006811	BP	117	0.715(0.060)	0.722(0.078)	0.762(0.087)	0.762(0.081)	0.763(0.085)	0.764(0.081)
GO:0006725	BP	164	0.858(0.040)	0.835(0.037)	0.894(0.050)	0.894(0.050)	0.887(0.051)	0.887(0.050)
GO:0016491	MF	516	0.938(0.013)	0.925(0.013)	0.946(0.009)	0.946(0.009)	0.946(0.008)	0.946(0.008)
GO:0009405	BP	118	0.925(0.044)	0.909(0.046)	0.942(0.040)	0.942(0.040)	0.943(0.040)	0.943(0.040)
GO:0005524	MF	485	0.875(0.032)	0.856(0.032)	0.892(0.026)	0.893(0.025)	0.891(0.027)	0.891(0.027)
GO:0030246	MF	102	0.904(0.032)	0.887(0.034)	0.919(0.032)	0.918(0.031)	0.918(0.033)	0.918(0.032)
GO:0006508	BP	330	0.924(0.019)	0.904(0.023)	0.934(0.015)	0.934(0.015)	0.933(0.015)	0.933(0.015)
GO:0008652	BP	121	0.912(0.034)	0.905(0.034)	0.922(0.035)	0.921(0.034)	0.920(0.034)	0.920(0.034)
GO:0045184	BP	108	0.789(0.063)	0.765(0.059)	0.849(0.041)	0.850(0.038)	0.849(0.041)	0.849(0.042)
GO:0020037	MF	104	0.987(0.016)	0.986(0.013)	0.993(0.013)	0.994(0.012)	0.993(0.014)	0.993(0.013)
GO:0003700	MF	214	0.907(0.020)	0.886(0.021)	0.931(0.022)	0.931(0.022)	0.931(0.022)	0.931(0.021)
GO:0016070	BP	140	0.868(0.058)	0.839(0.059)	0.914(0.039)	0.913(0.039)	0.915(0.038)	0.915(0.039)
GO:0005102	MF	120	0.917(0.046)	0.902(0.047)	0.930(0.043)	0.928(0.042)	0.929(0.042)	0.930(0.041)
GO:0006355	BP	340	0.908(0.017)	0.891(0.023)	0.926(0.021)	0.926(0.020)	0.925(0.022)	0.925(0.021)
GO:0016874	MF	161	0.872(0.039)	0.838(0.040)	0.885(0.035)	0.885(0.036)	0.884(0.034)	0.883(0.034)
GO:0006468	BP	160	0.860(0.056)	0.844(0.063)	0.892(0.056)	0.892(0.057)	0.893(0.055)	0.893(0.055)
GO:0016798	MF	227	0.969(0.021)	0.955(0.028)	0.967(0.017)	0.966(0.017)	0.967(0.017)	0.967(0.017)
GO:0006118	BP	392	0.937(0.013)	0.919(0.017)	0.961(0.012)	0.960(0.012)	0.961(0.012)	0.961(0.012)
GO:0004672	MF	164	0.873(0.063)	0.856(0.050)	0.900(0.048)	0.899(0.048)	0.900(0.046)	0.900(0.046)
GO:0004872	MF	138	0.904(0.047)	0.895(0.051)	0.933(0.030)	0.932(0.033)	0.933(0.029)	0.931(0.030)
GO:0015075	MF	110	0.755(0.072)	0.757(0.055)	0.815(0.059)	0.808(0.062)	0.814(0.059)	0.813(0.061)
GO:0005489	MF	196	0.935(0.024)	0.915(0.024)	0.965(0.021)	0.965(0.021)	0.965(0.021)	0.965(0.021)
GO:0005576	CC	352	0.876(0.031)	0.862(0.032)	0.896(0.022)	0.896(0.022)	0.895(0.020)	0.895(0.020)
GO:0019012	CC	101	0.841(0.064)	0.813(0.078)	0.879(0.056)	0.879(0.056)	0.878(0.054)	0.877(0.055)
GO:0030234	MF	132	0.808(0.059)	0.773(0.062)	0.830(0.052)	0.829(0.053)	0.827(0.057)	0.828(0.056)
GO:0016021	CC	136	0.698(0.054)	0.675(0.049)	0.789(0.054)	0.786(0.051)	0.787(0.051)	0.787(0.051)
GO:0006412	BP	170	0.865(0.038)	0.825(0.043)	0.910(0.037)	0.910(0.037)	0.910(0.036)	0.910(0.037)
GO:0005634	CC	347	0.901(0.026)	0.882(0.027)	0.927(0.026)	0.927(0.026)	0.927(0.026)	0.927(0.026)
GO:0017111	MF	154	0.796(0.056)	0.780(0.066)	0.866(0.049)	0.865(0.050)	0.866(0.048)	0.867(0.047)
GO:0005737	CC	490	0.877(0.027)	0.849(0.030)	0.903(0.024)	0.903(0.024)	0.901(0.023)	0.901(0.023)
GO:0051188	BP	118	0.784(0.054)	0.763(0.066)	0.817(0.051)	0.817(0.051)	0.814(0.052)	0.814(0.051)
GO:0043232	CC	153	0.804(0.060)	0.781(0.060)	0.842(0.050)	0.844(0.048)	0.842(0.050)	0.843(0.050)
GO:0043234	CC	414	0.787(0.028)	0.764(0.027)	0.846(0.025)	0.844(0.025)	0.843(0.025)	0.842(0.025)
GO:0005509	MF	173	0.901(0.033)	0.886(0.037)	0.920(0.021)	0.918(0.021)	0.918(0.020)	0.918(0.020)
GO:0050874	BP	144	0.883(0.051)	0.863(0.056)	0.900(0.044)	0.900(0.044)	0.902(0.041)	0.901(0.042)
GO:0006732	BP	119	0.788(0.058)	0.752(0.070)	0.850(0.050)	0.849(0.050)	0.841(0.052)	0.841(0.051)
GO:0007242	BP	140	0.900(0.029)	0.883(0.041)	0.925(0.031)	0.925(0.031)	0.926(0.032)	0.926(0.032)
GO:0005525	MF	104	0.906(0.043)	0.912(0.043)	0.925(0.042)	0.931(0.045)	0.925(0.042)	0.928(0.044)
GO:0004252	MF	140	0.923(0.046)	0.915(0.055)	0.924(0.042)	0.926(0.042)	0.924(0.042)	0.924(0.041)
GO:0005198	MF	179	0.819(0.031)	0.811(0.035)	0.837(0.035)	0.839(0.036)	0.836(0.038)	0.836(0.038)

Table 4: Performances of SVMs trained on single sources of data compared with performances of data integration methods (AUC averaged across the 5 folds) with 6 no-noisy and 6 noisy data. First column: Identifiers starting with K correspond to SVMs trained on single sources of data. The subscripts correspond to those of the data sets of Table 1. KF stands for Kernel Fusion, $Wens$ for weighted linear ensembles and DT for Decision Templates. The ending ”n” refers to the SVM or data integration method trained with the corresponding noisy data. The other columns show the AUC results for the 15 FunCat classes. FunCat classes are represented through their two-digits identifiers (see Table 2).

Methods	01	02	10	11	12	14	16	18	20	30	32	34	40	42	43
K_{BG}	0.7783	0.7432	0.8516	0.8717	0.8969	0.7738	0.7001	0.7739	0.8278	0.8393	0.7109	0.7981	0.8594	0.7822	0.8264
K_{BGn}	0.5399	0.5546	0.5212	0.5142	0.5412	0.5271	0.5110	0.5791	0.5332	0.5577	0.5278	0.5255	0.5299	0.5414	0.5199
K_{pfam1}	0.7731	0.7629	0.7430	0.7205	0.8104	0.7056	0.6730	0.7404	0.7431	0.7995	0.6151	0.6818	0.7206	0.6013	0.7669
K_{pfam1n}	0.5196	0.5505	0.5257	0.5162	0.5283	0.5168	0.5187	0.5517	0.5242	0.5584	0.5253	0.5383	0.5529	0.5266	0.5500
K_{pfam2}	0.7667	0.6920	0.6393	0.6625	0.7567	0.7295	0.6710	0.6096	0.7275	0.7940	0.6844	0.6414	0.6631	0.5899	0.6174
K_{pfam2n}	0.5152	0.5125	0.5248	0.5167	0.5206	0.5241	0.5090	0.5174	0.5156	0.5356	0.5186	0.5160	0.5324	0.5172	0.5310
K_{string}	0.7423	0.7558	0.7855	0.7957	0.8460	0.7403	0.6127	0.6319	0.7794	0.7118	0.6862	0.7592	0.7697	0.7099	0.7705
$K_{stringn}$	0.5098	0.5291	0.5271	0.5224	0.5323	0.5282	0.5131	0.5540	0.5082	0.5596	0.5381	0.5377	0.5566	0.5372	0.5523
K_{seq}	0.8105	0.7650	0.7548	0.7773	0.8207	0.7332	0.7192	0.8067	0.7770	0.8476	0.6669	0.7039	0.7574	0.6560	0.7930
K_{seqn}	0.5184	0.5245	0.5344	0.5350	0.5241	0.5413	0.5210	0.5698	0.5175	0.5557	0.5217	0.5282	0.5383	0.5135	0.5283
K_{expr}	0.8091	0.7787	0.7594	0.7928	0.8171	0.7327	0.7080	0.8206	0.7671	0.8492	0.6469	0.6982	0.7626	0.6521	0.7957
K_{exprn}	0.5158	0.5232	0.5414	0.5132	0.5316	0.5228	0.5202	0.5643	0.5317	0.5490	0.5328	0.5312	0.5595	0.5302	0.5398
KF	**0.8966**	0.8451	**0.8881**	**0.9193**	0.9268	0.8483	**0.8085**	0.8403	**0.9037**	0.8914	0.7718	**0.8464**	0.8614	**0.8229**	0.8769
KFn	0.8725	0.8207	0.8793	0.9025	0.9103	0.8082	0.7727	0.7767	0.8637	0.8637	0.7442	0.7942	0.8236	0.8024	0.8522
$Wens$	0.8898	0.8415	0.8689	0.9026	0.9233	0.8426	0.7935	0.8415	0.8906	**0.9059**	0.7708	0.8363	0.8671	0.7953	0.8792
$Wensn$	0.8887	**0.8462**	0.8747	0.9040	**0.9280**	**0.8538**	0.7906	**0.8572**	0.8847	0.9029	**0.7761**	0.8280	0.8655	0.7941	0.8763
DT	0.8817	0.8364	0.8679	0.8972	0.9209	0.8300	0.7754	0.8409	0.8859	0.9053	0.7509	0.8317	**0.8700**	0.7972	**0.8779**
DTn	0.8810	0.8403	0.8739	0.9012	0.9246	0.8430	0.7735	0.8497	0.8800	0.9037	0.7628	0.8200	0.8632	0.7956	0.8723

References

[1] SR Eddy. Profile hidden markov models. *Bioinformatics*, 14(9):755–763, 1998.

[2] R.D. Finn, J. Tate, J. Mistry, P.C. Coggill, J.S. Sammut, H.R. Hotz, G. Ceric, K. Forslund, S.R. Eddy, E.L. Sonnhammer, and A. Bateman. The Pfam protein families database. *Nucleic Acids Research*, 36:D281–D288, 2008.

[3] P. Gasch and M. Eisen. Exploring the conditional regulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology*, 3:11, 2002.

[4] A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Guldener, G. Mannhaupt, M. Munsterkotter, and H.W. Mewes. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32(18):5539–5545, 2004.

[5] C. Stark, B. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: a general repository for interaction datasets. *Nucleic Acids Res.*, 34:D535–D539, 2006.

[6] C. vonMering et al. STRING: a database of predicted functional associations between proteins. *Nucleic Acids Research*, 31:258–261, 2003.