Simple ensemble methods are competitive with state-of-the-art data integration methods for gene function prediction

Matteo Re and Giorgio Valentini

DSI, Dipartimento di Scienze dell' Informazione,
Università degli Studi di Milano,
Via Comelico 39, 20135 Milano, Italia.
{re,valentini}@dsi.unimi.it

Abstract. Several works showed that biomolecular data integration is a key issue to improve the prediction of gene functions. Quite surprisingly only little attention has been devoted to data integration for gene function prediction through ensemble methods. In this work we show that relatively simple ensemble methods are competitive and in some cases are also able to outperform state-of-the-art data integration techniques for gene function prediction.

1 Introduction

The availability of an ever increasing amount of data sources due to recent advances in high throughput biotechnologies opens unprecedented opportunities for genome-wide gene function prediction. Indeed several works showed that biomolecular data integration play an essential role in the prediction of genes/gene products functions.

Gene function prediction in its general formulation is a complex classification problem characterized by the following items: a) each gene/gene product can be assigned to multiple terms/classes (a multiclass, multilabel classification problem); b) classes are structured according to a predefined hierarchy (a directed acyclic graph for the Gene Ontology [1] or a tree forest for FunCat [2]); c) classes are usually unbalanced (with positive examples usually less than negatives); d) known gene labels are in several cases be uncertain; e) multiple sources of data can be used to predict gene functions.

In this paper we focus on the last item, considering the problem of the prediction of a subset of FunCat classes in the model organism *S. cerevisiae*.

The main approaches proposed in the literature can be schematically subdivided in three categories: functional linkage networks, vector subspace integration and kernel fusion methods [3]. Modelling 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 [4]. In vector space integration (VSI) different vectorial data are concatenated [5], 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 [6].

All these methods suffer of limitations and drawbacks, due to their limited scalability to multiple data sources (e.g. Kernel integration methods based on semidefinite programming [6]), to their limited modularity when new data sources sources are added (e.g. vector-space integration methods), or when data are not available as relational data (e.g. functional linkage networks).

Quite surprisingly, as observed by William Noble and Asa Ben-Hur [3], only little attention has been devoted to ensemble methods 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 [7], 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 [8].

The main aim of this work consists in showing 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. 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. Moreover, as new types of biomolecular data, or updates of data contained in public databases, are made available to the research community, ensembles of learning machines are able to embed new data sources or to update existing ones by training only the base learners devoted to the newly added or updated data, without retraining the entire ensemble. 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.

2 Methods

2.1 Ensemble methods

Data fusion can be realized by means of an ensemble system composed by learners trained on different "views" of the data and then combining the outputs of the component learners. Each type of data may capture different and complementary characteristics of the objects to be classified and the resulting ensemble may obtain better prediction capabilities through the diversity and the anti-correlation of the base learner responses.

We programmatically considered simple methods:

Weighted majority voting [10], using linear or logarithmic weights, tuned on the F-measure estimated from the training data, since gene functional classes are usually unbalanced.

Naive Bayes : a combination of classifiers assuming independence between them, that estimates the class-conditional support given the observed vector of categorized component classifiers outputs [11].

Decision Templates : a combination method based on 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) [12].

The decision profile DP($\mathbf{x}$) for an instance $\mathbf{x}$ 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:

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

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 $\mathbf{x}$ to a class with the largest similarity [12]:

$$D(\mathbf{x}) = \arg\max_j \mathcal{S}_j(\mathbf{x}) \tag{2}$$

It is easy to see that with dichotomic problems the decision templates are reduced to two-columns matrices, and 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:

$$\mathcal{S}_1(\mathbf{x}) = 1 - \frac{1}{n} \sum_{t=1}^{n} [DT_1(t,1) - d_{t,1}(\mathbf{x})]^2 \tag{3}$$

$$\mathcal{S}_2(\mathbf{x}) = 1 - \frac{1}{n} \sum_{t=1}^{n} [DT_2(t,1) - d_{t,1}(\mathbf{x})]^2 \tag{4}$$

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

$$D(\mathbf{x}) = \arg\max_{\{1,2\}} (\mathcal{S}_1(\mathbf{x}), \mathcal{S}_2(\mathbf{x})) \tag{5}$$

2.2 Kernel fusion and vector space integration

Kernel fusion (KF) for data integration is based on the closure property of kernels with respect to the sum or other algebraic operators [6]. In our experiments we integrated the different data sets by simply summing their Gram matrices, and then we trained the SVMs directly with the resulting matrix. Vector space integration (VSI) consists in concatenating the vectors of the different data sets [5]. The resulting concatenated vectors are used to train a SVM. Note that training a linear SVM with concatenated vectors (VSI) is equivalent to kernel fusion with linear kernels. In our experiments we used gaussian kernels.

Table 1. Datasets

Code	Dataset	examples	features	description
D_{ppi1}	PPI - STRING	2338	2559	protein-protein interaction data from [13]
D_{ppi2}	PPI - BioGRID	4531	5367	protein-protein interaction data from the *BioGRID* database [14]
D_{pfam1}	Protein domain log-E	3529	5724	Pfam protein domains with log E-values computed by the *HMMER* software toolkit
D_{pfam2}	Protein domain binary	3529	4950	protein domains obtained from *Pfam* database [15]
D_{expr}	Gene expression	4532	250	merged data of Spellman and Gasch experiments
D_{seq}	Pairwise similarity	3527	6349	Smith and Waterman log-E values between all pairs of yeast sequences

3 Experimental results

Even if the growing rate of the amount of biomolecular data available for many species was constantly increasing in the last years, the model organisms with a consistent amount of literature inherent to data fusion based gene function prediction are actually reduced to S.*cerevisiae* and M.*musculus*. Despite the availability of a well established public benchmark dataset, such as the one provided during the MouseFunc contest [18], a recent comparison between many model organisms showed that the fraction of genes annotated with experimental evidence is about 30% larger in S.*cerevisiae* than in M.*musculus* (85.4% and 57.8% respectively for the yeast and mouse model organisms) [19]. We thus decided to use yeast data for our experiments. In order to maximize the effective use of the larger experimental coverage of gene functional annotations available for the yeast, we also adopted as a reference functional ontology, the MIPS Functional Catalogue (FunCAT), which is composed by annotations mainly based on experimental evidences [2], allowing us to minimize the impact of non experimental functional annotations.

We predicted the top-level 15 functional classes of the FunCat taxonomy of the model organism *S. cerevisiae*, using 6 different sources of data (Tab. 1). Each dataset was split into a training set and a test set (composed, respectively, by the 70% and 30% of the available samples), considering yeast genes common to all data sets (about 1900) and with at least 1 FunCat annotation. A 3-fold stratified cross-validation has been performed on the training data for model selection, using gaussian SVMs with probabilistic output [9] as base learners for ensemble methods, and for VSI and KF data integration. We compared the performances of single gaussian SVMs trained on each data set with those obtained with vector-space-integration (VSI) techniques, kernel fusion through the sum of gaussian kernels, and with the ensembles described in Sect. 2.1.

Table 2 shows the average F-measure, recall, precision and AUC across the 15 selected FunCat classes, obtained through the evaluation of the test sets (each constituted by 570 genes). The four first columns refer respectively to the weighted linear, logarithmic linear, decision template and naive Bayes ensembles;

VSI and KF stands respectively for vector space integration and kernel fusion, D_{avg} represents the average results of the single SVMs across the six datasets, and D_{ppi2} represents the single SVM that achieved the best performance, i.e. the one trained using protein-protein interactions data collected from BioGrid. Tab. 3 shows the same results obtained by each single SVM trained on a specific biomolecular data set.

Looking at the values presented in Tab. 2, on the average, data integration through simple ensemble methods provide better results than single SVMs, VSI and Kernel fusion, independently of the applied combination rule. In particular, Decision Templates achieved the best average F-measure, and ensemble methods as a whole the best AUC. Among the ensemble of classifiers, with respect to the AUC, the worst performing method is the Naive Bayes combiner albeit its performances are still, on the average, higher than the ones reported for VSI, Kernel fusion and the single classifiers. Precision of the ensemble methods is relatively high: this is of paramount importance to drive the biological validation of "in silico" predicted functional classes: considering the high costs of biological experiments, we need to obtain a high precision (and possibly recall) to be sure that positive predictions are actually true with the largest confidence.

To understand whether the differences between AUC scores in the 15 dichotomic tasks are significant, we applied a non parametric test based on the Mann-Whitney statistic [16], using a recently proposed software implementation [17]. Tab. 4 shows that at 0.01 significance level in most cases there is no significant difference between AUC scores of the weighted linear and logarithmic ensembles (E_{lin} and E_{log}) and the Decision Template (E_{dt}) combiner. A different behavior is observed for the Naive Bayes combiner: its performances are comparable to the ones obtained by the other ensemble methods only in 2 over 15 classification tasks and worse in the remaining 13.

Most interestingly, ensemble methods significantly outperform the other data integration methods. For instance, wins-ties-losses of E_{lin} vs VSI are $13-2-0$, and $9-6-0$ vs KF; Naive-Bayes, the worst performing ensemble method, achieves $9-6-0$ wins-ties-losses with VSI and $5-10-0$ with KF. It is worth noting that, among the tested ensemble methods, E_{lin}, E_{log} and E_{dt} undergo no losses when compared with single SVMs (Tab. 4, bottom): we can safely choose any ensemble method (but not the Naive Bayes combiner) to obtain equal or

Table 2. Ensemble methods, kernel fusion and vector space integration: average F-score, recall, precision and AUC (Area Under the Curve) across the data sets.

Metric	E_{lin}	E_{log}	E_{dt}	E_{NB}	VSI	KF	D_{avg}	D_{ppi2}
F	0.4347	0.4111	0.5302	0.5174	0.3213	0.3782	0.3544	0.4818
rec	0.3304	0.2974	0.4446	0.6467	0.2260	0.3039	0.2859	0.3970
prec	0.8179	0.8443	0.7034	0.5328	0.6530	0.6293	0.5823	0.6157
AUC	0.8642	0.8653	0.8613	0.7933	0.7238	0.7775	0.7265	0.8170

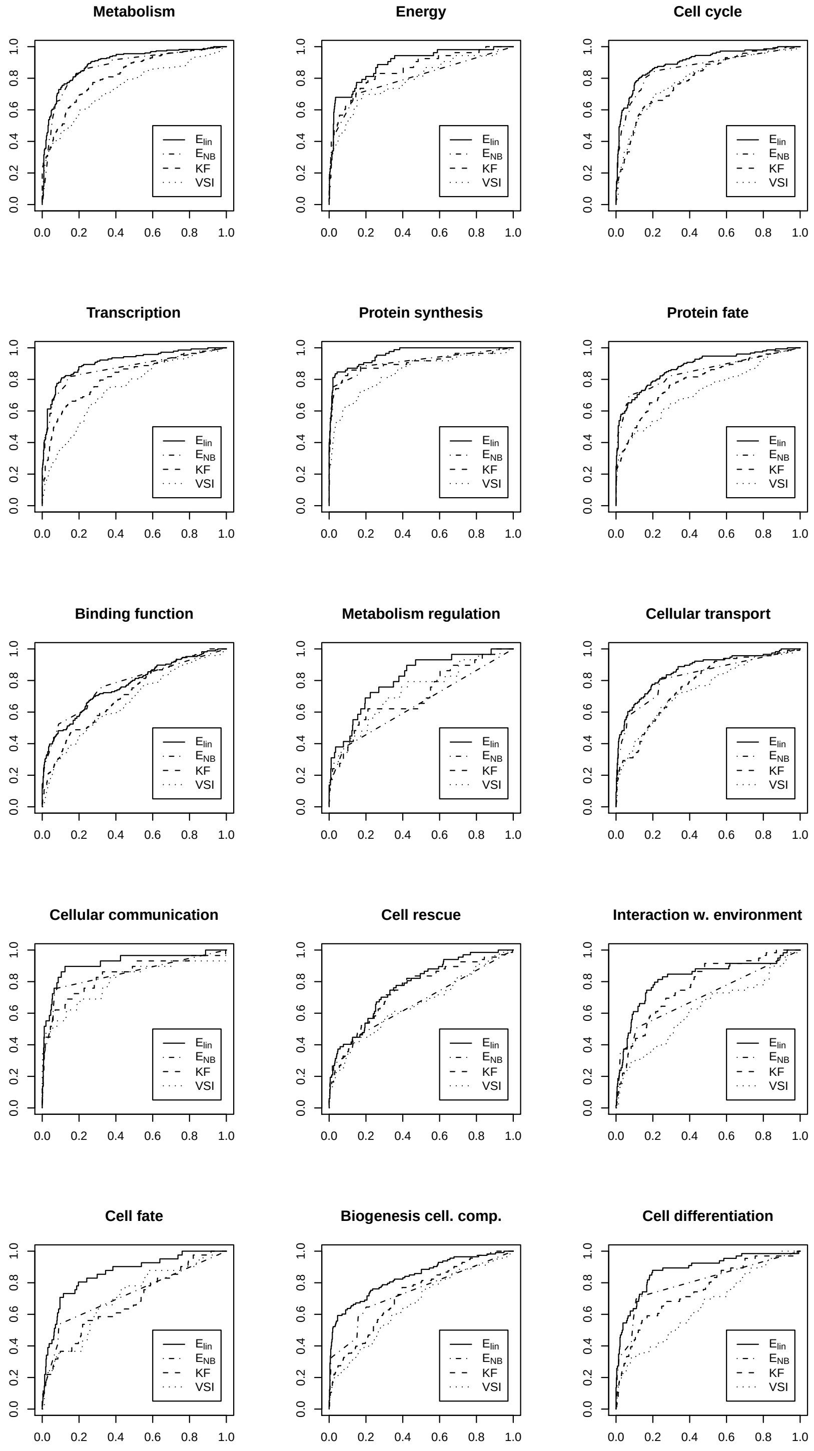

Fig. 1. Comparison of ROC curves between different data integration methods. E_{lin}: ensemble weighted majority voting; E_{NB}: Naive-Bayes ensemble integration; KF: kernel fusion; VSI: vector space integration.

Table 3. Single SVMs: average F-score, recall, precision and AUC. Each SVM is identified by the same name of the data set used for its training (Tab. 1).

Metric	D_{ppi1}	D_{ppi2}	D_{pfam1}	D_{pfam2}	D_{expr}	D_{seq}
F	0.3655	0.4818	0.2363	0.3391	0.2098	0.4493
rec	0.2716	0.3970	0.1457	0.2417	0.1571	0.5019
prec	0.6157	0.6785	0.7154	0.6752	0.3922	0.4162
AUC	0.7501	0.8170	0.6952	0.6995	0.6507	0.7469

Table 4. Results of the non-parametric test based on Mann-Whitney statistics to compare AUCs between ensembles, VSI, Kernel fusion and single SVMs. Each entry represents wins-ties-losses between the corresponding row and column at 0.01 significance level. Top: Comparison between ensemble methods, VSI and kernel fusion; Bottom: Comparison between data integration methods and single SVMs.

	VSI	E_{log}	E_{lin}	E_{dt}	E_{NB}
E_{log}	13-2-0	-	-	-	-
E_{lin}	13-2-0	0-14-1	-	-	-
E_{dt}	13-2-0	1-13-1	1-11-3	-	-
E_{NB}	9-6-0	0-2-13	0-2-13	0-2-13	-
KF	3-12-0	0-6-9	0-6-9	0-6-9	0-10-5

	D_{ppi1}	D_{ppi2}	D_{pfam1}	D_{pfam2}	D_{expr}	D_{seq}
E_{lin}	11-4-0	4-11-0	15-0-0	14-1-0	15-0-0	13-2-0
E_{log}	11-4-0	4-11-0	15-0-0	14-1-0	15-0-0	13-2-0
E_{dt}	11-4-0	4-11-0	15-0-0	14-1-0	15-0-0	13-2-0
E_{NB}	5-10-0	2-11-2	9-6-0	8-7-0	12-3-0	7-8-0
VSI	1-11-3	0-8-7	2-11-2	1-14-0	4-11-0	0-12-3
KF	1-14-0	0-9-6	5-10-0	5-10-0	11-4-0	3-12-0

better results than any of the single SVMs. On the contrary in many cases VSI, E_{NB} and the kernel fusion methods obtained worse results than single SVMs, although performances achieved by the Naive Bayes combiner and the kernel fusion methods are, in general, better than those obtained by VSI. Nevertheless, we can observe that a single SVM trained with Ppi-2 data achieves good results (11 ties with ensembles and an average AUC $\simeq 0.81$ w.r.t. 0.86 of the ensembles, Tab. 2 and 4), showing that large protein-protein interactions data sets alone provide information sufficient to correctly predict several FunCat classes.

Fig. 1 compares the ROC curves of the different data integration methods used in our experiments. ROC curves of weighted majority voting (E_{lin}) are consistently above the corresponding ROC curves of kernel fusion and vector space integration for all the considered FunCat classes. ROC curves of Naive Bayes combiner are below those of kernel fusion only for four classes: “Energy”, “Metabolism”, “Regulation”, “Cell rescue” and “Interaction with the environment”.

4 Conclusions

The main objective of this contribution is to demonstrate that simple ensemble methods are competitive with state-of-the-art methods for gene function prediction based on heterogeneous biomolecular data integration.

It is well-known that gene function prediction methods need to take into account the hierarchical relationships between classes to improve their predictions [7, 8, 20]. Nevertheless, in this investigation we focused on data integration, in order to study the improvement due to the usage of multiple sources of data, without exploiting any knowledge about the hierarchical relationships between classes. In this way we can separate the contribution due to data fusion techniques from the improvement due to hierarchical methods.

Considering the increasing growing rate of available biomolecular data, the modularity and scalability that characterize ensemble methods can favour an easy update of existing sources of data and an easy integration of new ones. Our preliminary experiments show that relatively simple ensemble methods are competitive with kernel fusion and vector space integration, two of the most largely applied machine learning data integration techniques for gene function prediction. This could seem quite surprisingly, but considering the uncertainty that characterize both annotations and measurements of data values, we can expect that relatively simple methods are able to nicely work in a similar context. Moreover it is worth noting that each type of data can only capture a particular characteristic of a protein, and for different functional classes the same type of data can be highly informative or completely unuseful to discriminate positive and negative examples. For these reasons the inherent modularity and adaptivity of ensemble systems can explain their effectiveness for the integration of multiple biomolecular data sources. In particular we think that ensemble methods devoted to biomolecular data integration can be a valuable research line to improve the accuracy of gene function prediction problems.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments and suggestions. The authors gratefully acknowledge partial support by the PASCAL2 Network of Excellence under EC grant no. 216886. This publication only reflects the authors' views.

References

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

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

[3] Noble, W., Ben-Hur, A.: Integating information for protein function prediction. In Lengauer, T., ed.: Bioinformatics - From Genomes to Therapies. Volume 3. Wiley-VCH (2007) 1297–1314

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

[5] desJardins, M., Karp, P., Krummenacker, M., Lee, T., Ouzounis, C.: Prediction of enzyme classification from protein sequence without the use of sequence similarity. In: Proc. of the 5th ISMB, AAAI Press (1997) 92–99

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

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

[8] Obozinski, G., Lanckriet, G., Grant, C., M., J., Noble, W.: Consistent probabilistic output for protein function prediction. Genome Biology **9** (2008)

[9] Lin, H., Lin, C., Weng, R.: A note on Platt's probabilistic outputs for support vector machines. Machine Learning **68** (2007) 267–276

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

[11] Titterington, D., Murray, G., Spiegelhalter, D., Skene, A., Habbema, J., Gelpke, G.: Comparison of discriminant techniques applied to a complex data set of head injured patients. Journal of the Royal Statistical Society **144** (1981)

[12] Kuncheva, L., Bezdek, J., Duin, R.: Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition **34** (2001) 299–314

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

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

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

[16] Delong, E., Delong, D., Clarke-Pearson, D.: Comparing the areas under two or more or more correlated Receiver Operating Characteristics Curves: a non parametric approach. Biometrics **44** (1988) 837–845

[17] Vergara, I., Norambuena, T., Ferrada, E., Slater, A., Melo, F.: StAR: a simple tool for the statistical comparison of ROC curves. BMC Bioinformatics **9** (2008)

[18] Pena Castillo, L., et al.: A critical assessment of Mus musculus gene function prediction using integrated genomic evidence. Genome Biology **9:S2** (2008)

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

[20] Valentini, G. and Re, M.: Weighted True Path Rule: a multilabel hierarchical algorithm for gene function prediction. In: MLD-ECML 2009, 1st International Workshop on learning from Multi-Label Data, Bled, Slovenia (2009) 133–146