Speed and Accuracy Benchmarks of Large-Scale Microbial Gene Function Prediction with Supervised Machine Learning

Vedrana Vidulin[1], Tomislav Šmuc[1], Fran Supek[1,2]

[1]Division of Electronics, Ruđer Bošković Institute, Bijenička cesta 54, 10000 Zagreb, Croatia

[2]EMBL/CRG Systems Biology Unit, Centre for Genomic Regulation, Dr. Aiguader 88, 08003 Barcelona, Spain

1 Introduction

Machine learning approaches for microbial gene function prediction (MGFP) from genome context data are mostly unsupervised [1] and rely on pairwise distances between individual examples arranged into "functional interaction networks" [2]. When supervised approaches were used, most of them typically predicted a limited set of functions and/or used a single-label approach to classification [3, 4], constructing a separate classifier for each function and ignoring the relationships between the functions. Multi-label approaches may perform better, especially those that can exploit the relations between functions readily available in gene function ontologies [5].

Our aim is to compare predictive accuracy and computational efficiency of single *vs.* multi-label approaches on supervised MGFP. High accuracy is a prerequisite for applying the classifier in real-life tasks, where confidence in predicted functions is of key importance for prioritizing downstream experimental work. Many such predictions have indeed been validated in biological experiments [6, 7]. A lower demand for computational time is of importance when the number of considered functions is high.

2 Data and Experimental Setup

We collected microbial genome data from three databases (NCBI Genome, eggNOG [8] and Gene Ontology – GO) and constructed separate data sets for three types of microbial data representations [1]: (1) the phyletic profiles (PP [9]) data set represents co-occurrences of clusters of orthologous groups (COG) across multiple genomes and contains 6018 examples representing COGs, 1690 attributes representing genomes, and binary values representing COGs' presence/absence in the genomes; (2) the translation efficiency profiles (TEP [10]) data set indicates COGs' predicted expression levels across the genomes and differs from PP in attribute values – for present COGs expression levels are measured with MELP measure [11] and absent COGs are represented with missing values (providing a representation orthogonal to PP); (3) in the conserved gene neighborhood (CGN) data set both attributes and examples represent COGs and attribute values represent relative pairwise distances between COGs (distances are averaged over all genomes where both genes of a pair are present). In all three data sets, the classes are functions taken from the controlled vocabulary of GO. When a COG is labelled with a function at the lower level of the GO, then all superordinate functions are attributed too. In our data sets, we accounted for a subset of 776 GO functions represented with 50 or more examples (COGs).

Classifiers were constructed with four algorithms: CLUS-HMC [12], Fast Random Forest (FRF [13]), k-nearest neighbors (kNN) and Naive Bayes (NB) (from Weka 3.7.11 [14]). Both CLUS and FRF construct a random forest ensemble of decision trees. However, CLUS is hierarchical multi-label classification algorithm, which is aware of the hierarchical relationships between GO functions, while FRF is a single-label algorithm. In both cases, ensembles of 500 trees were constructed. A single-label kNN (with k of 1 and inverse distance weighting) was selected as an algorithm that represents the closest approximation to the unsupervised MGFP approaches. We selected a single-label NB (with default Weka parameters) since it is typically faster on big data in comparison to other algorithms. A wrapper that creates 776 binary-class data sets, one for each function, and computes multiple classifiers was built around the single-label algorithms. Predictive accuracy was evaluated using 10-fold cross-validation and measured as an area under the precision-recall curve (AUPRC).

3 Results and Discussion

The results of comparisons on a subset of 776 GO functions belonging to the "biological process" GO namespace are presented in Figure 1. We have also made comparisons on subsets of functions from the other two GO subhierarchies, molecular function and cellular component, and they exhibit the same general behavior.

Accuracy of the two versions of random forest algorithms, that is, FRF and CLUS, significantly outperforms kNN and NB over all representations and GO functions (Wilcoxon signed-rank test $p<2.2 \times 10^{-16}$). Ensembles, here represented by the random forest classifiers, are often used in biological domains [15] since they capture complex relations in data while being robust to overfitting.

kNN exhibits better performance than NB for PP and CGN. For TEP, in most of the cases, NB performs better. TEP data set has a large proportion of missing values (77%) in comparison to the other two data sets (PP is without missing values, CGN has 18%). While NB ignores missing values while determining the priors, kNN sets the distance to maximum if one of the two attributes being compared has a missing value. With that strategy, kNN decreases classifier's accuracy with the increase of missing values.

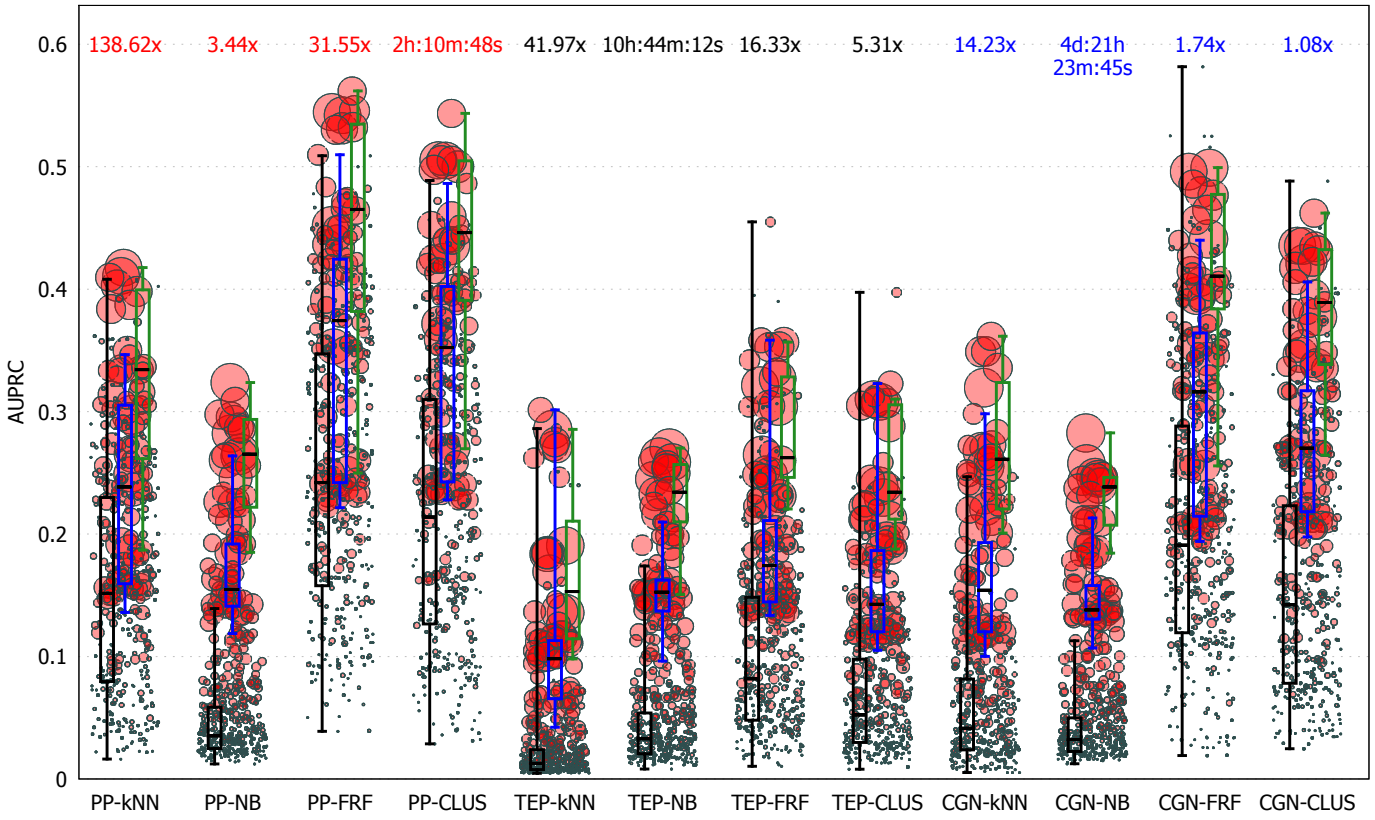

Figure 1: Results of comparisons: The names of classifiers on x-axis are composed of representation and algorithm name abbreviations. Circles represent GO functions and their size denote GO categories' frequencies. Very general functions (frequency $> 30\%$) are omitted. The three box-plots for each classifier represent classifier's performance on small (frequency $<= 10\%$ – black), medium ($10\% <$ frequency $<= 20\%$ – blue) and large ($20\% <$ frequency $<= 30\%$ – green) categories. Box-plots represent minimum, first quartile, median, third quartile and maximum accuracy. Execution times are given above the box-plots, where comparisons for each representation are marked in different color. The fastest execution times are expressed in absolute numbers, while the other executions times are expressed relative to the fastest times.

In single to multi-label comparisons of the two ensemble classifiers, FRF outperformed CLUS (Wilcoxon signed-rank test $p<2.2 \times 10^{-16}$). This result suggests that CLUS does not fully benefit from the hierarchical relationships between the functions in this particular setup, although it has been shown to do so in a non-ensemble (single tree) applications to other datasets [5]. This agrees with research showing that hierarchical and/or multi-label setups are consistently beneficial for predictive accuracy of decision trees across various datasets, but this advantage is less evident with tree ensembles [16]. Consistently, a previous application of CLUS-HMC ensembles to phylogenetic profiling found no effect of varying the weights of GO hierarchy levels, implying that the ensemble did not draw on the information contained in the hierarchy [7].

In computational time requirement comparisons, CLUS has an advantage over FRF: FRF needs 31.55× more time than CLUS to compute the classifier for PP, 3.08× for TEP and 1.61× for CGN). Please note that in our setup FRF computed 10 trees in parallel, while CLUS was computing the trees one after another.

In the case of TEP and CGN, NB has the shortest execution time. The advantage of NB over the random forest approaches is higher for TEP than for CGN. The main cause is in handling missing values: decision trees, when testing on an attribute with a missing value, fragment an instance, while NB ignores the missing values. While TEP has 77% of missing values, in the case of CGN with 18% of missing values CLUS has almost comparable execution time (1.08× slower than NB). In the case of PP, which are without missing values, CLUS performs better that NB (3.44× faster than NB).

CLUS and FRF have the shortest execution time on PP, since, in contrast to TEP and CGN, PP has categorical attribute values. When splitting on a categorical attribute, a decision tree tests one possible split, while when splitting on a numeric attribute several possible splitting points are tested.

In the presented setup, kNN has the longest execution time for all representations because the construction of each classifier was treated as a separate process. The kNN execution time can be reduced by computing the distance matrix once and reusing it in all processes.

In conclusion, the best choice for MGFP is CLUS when a trade-off between predictive accuracy and computational time demands are considered. When the aim is to strictly achieve the highest possible predictive accuracy, FRF may be preferrable. NB is generally the fastest, but constructs classifiers of considerably lower quality that are not suitable candidate for real-life MGFP applications.

Acknowledgement

We would like to acknowledge the support of the European Commission through the project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944).

References

[1] V.Y. Muley and V. Acharya. *Genome-wide prediction and analysis of protein-protein functional linkages in bacteria*. Springer, 2013.

[2] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork, et al. The string database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic acids research*, 39(suppl 1):D561–D568, 2011.

[3] F.P.Y. Lin, E. Coiera, R. Lan, and V. Sintchenko. In silico prioritisation of candidate genes for prokaryotic gene function discovery: an application of phylogenetic profiles. *BMC bioinformatics*, 10(1):86, 2009.

[4] P. Hu, S.C. Janga, M. Babu, J.J. Díaz-Mejía, G. Butland, W. Yang, O. Pogoutse, X. Guo, S. Phanse, P. Wong, et al. Global functional atlas of escherichia coli encompassing previously uncharacterized proteins. *PLoS biology*, 7(4):e1000096, 2009.

[5] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214, 2008.

[6] P.R. Kensche, V. van Noort, B.E. Dutilh, and M.A. Huynen. Practical and theoretical advances in predicting the function of a protein by its phylogenetic distribution. *Journal of The Royal Society Interface*, 5(19):151–170, 2008.

[7] N. Škunca, M. Bošnjak, A. Križko, P. Panov, S. Džeroski, T. Šmuc, and F. Supek. Phyletic profiling with cliques of orthologs is enhanced by signatures of paralogy relationships. *PLoS computational biology*, 9(1):e1002852, 2013.

[8] S. Powell, K. Forslund, D. Szklarczyk, K. Trachana, A. Roth, J. Huerta-Cepas, T. Gabaldón, T. Rattei, C. Creevey, M. Kuhn, et al. eggnog v4. 0: nested orthology inference across 3686 organisms. *Nucleic acids research*, page gkt1253, 2013.

[9] M. Pellegrini, E.M. Marcotte, M.J. Thompson, D. Eisenberg, and T.O. Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proceedings of the National Academy of Sciences*, 96(8):4285–4288, 1999.

[10] A. Križko, T. Čopić, T. Gabaldón, B. Lehner, and F. Supek. Inferring gene function from evolutionary change in signatures of translation efficiency. *Genome biology*, 15(3):R44, 2014.

[11] F. Supek and K. Vlahoviček. Comparison of codon usage measures and their applicability in prediction of microbial gene expressivity. *BMC bioinformatics*, 6(1):182, 2005.

[12] Clus. `http://dtai.cs.kuleuven.be/clus`, 2014.

[13] Fast random forest. `https://code.google.com/p/fast-random-forest`, 2014.

[14] Weka 3: Data mining software in Java. `http://www.cs.waikato.ac.nz/ml/weka`, 2014.

[15] L. Schietgat, C. Vens, J. Struyf, H. Blockeel, D. Kocev, and S. Džeroski. Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC bioinformatics*, 11(1):2, 2010.

[16] J. Levatić, D. Kocev, and S. Džeroski. The importance of the label hierarchy in hierarchical multi-label classification. *Journal of Intelligent Information Systems*, 2014.