Parameters tuning boosts *hyperSMURF* predictions of rare deleterious non-coding genetic variants

Alessandro Petrini[1], Max Schubach[2], Matteo Re[1], Marco Frasca[1], Marco Mesiti[1], Giuliano Grossi[1], Tiziana Castrignanò[3], Peter N. Robinson[4] and Giorgio Valentini[1]

[1]AnacletoLab, Dipartimento di Informatica, Università degli Studi di Milano, Italy
[2]Berlin Institute of Health, Berlin, Germany
[3]CINECA, SCAI SuperComputing Applications and Innovation Department, Rome, Italy
[4]The Jackson Laboratory for Genomic Medicine, Farmington CT, USA

Introduction

The regulatory code that determines whether and how a given genetic variant affects the function of a regulatory element remains poorly understood for most classes of regulatory variation. Indeed the large majority of bioinformatics tools have been developed to predict the pathogenicity of genetic variants in coding sequences or conserved splice sites [5].

Computational algorithms for the prediction of non-coding deleterious variants associated with rare genetic diseases are faced with special challenges owing to the rarity of confirmed pathogenic mutations. Indeed in this context classical machine learning methods are biased toward neutral variants that constitute the large majority of genetic variation, and are not able to detect the potential deleterious variants that constitute only a tiny minority of all known genetic variation [8].

We recently proposed *hyperSMURF*, hyper-ensemble of SMOTE Undersampled Random Forests, an ensemble approach explicitly designed to deal with the huge imbalance between deleterious and neutral variants [7], and able to significantly outperform state-of-the-art methods for the prediction of non-coding variants associated with Mendelian diseases.

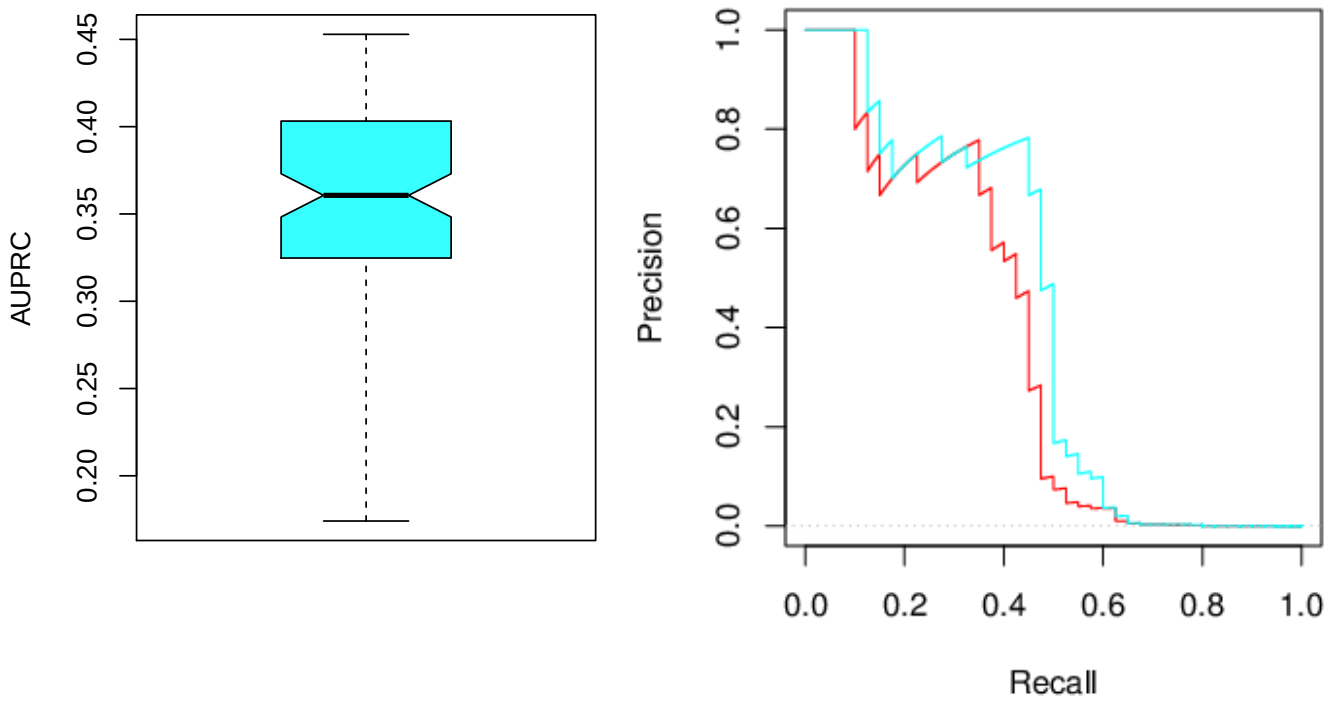
Despite its successful application to the detection of deleterious single nucleotide variants (SNV) as well as to small insertions or deletions (indels), *hyperSMURF* is a method that depends on several learning parameters, that strongly influence its overall performances. In this work we experimentally show that by tuning *hyperSMURF* parameters we can significantly boost the performance of the method, thus predicting with significantly better precision and recall rare SNVs associated with Mendelian diseases.

Methods

HyperSMURF is a method specifically designed to provide genome-wide predictions of deleterious (e.g. disease-associated) variants explicitly taking into account the imbalance that characterize the number of deleterious variants (positive examples) vs neutral variants (negative examples) in the non-coding human genome.

To this end two main learning strategies are adopted: 1) Sampling techniques and 2) Ensembling and hyper-ensembling approaches. By oversampling the small set of available positive examples using SMOTE [1] and at the same time subsampling the set of negative examples, we can balance the data, thus avoiding the bias toward the majority class. By training a set of random forests, each one on a different sampled and balanced set of the data, we can obtain both accurate and diverse base learners and a large coverage of the available training data: these features represent key factors for the success of ensemble methods [4]. Note that with *hyperSMURF* we have an ensemble of ensembles, since each base learner is in turn a random forest (i.e. an ensemble of random decision trees), thus obtaining an hyper-ensemble (ensemble of ensembles) approach. For a detailed description of the *hyperSMURF* algorithm, please see [7].

HyperSMURF is characterized by a set of learning parameters (i.e. the number of random forests n, the oversampling f and the subsampling factor m) that can significantly affect the overall performance of the method. Indeed these parameters have a high impact on the runtime and the training success,



(a) (b)

Figure 1: (a): Distribution across 100 different choices of *hyperSMURF*parameters (n, f, m) of the AUPRC values obtained by cross-validation on the training data; (b) Comparison of precision-recall curves obtained by *hyperSMURF* with default parameters (red curve) and with the selected "best" parameters (cyan curve).

ranging from the coverage of the training data, to the accuracy and diversity of the base learner, to the balancing between the classes, and to the number of new synthetic instances added via SMOTE [1]. In other words it is not always straightforward the proper choice of the learning parameters of *hyperSMURF*, since it also strongly depends on the underlying distribution of the data. In principle, by properly tuning the learning parameters, we can better fit the data and significantly boost the overall performances. Considering the complexity of a "a priori" evaluating of the "optimal" *hyperSMURF* parameters, we adopted an unbiased empirical approach by selecting through internal cross-validation on the training data the parameters that lead to the beast estimated performance using the Area Under the Precision Recall Curve (AUPRC) as the the metric to be maximized. In principle other metrics could be maximized, but we used the AUPRC, since the data are highly imbalanced.

Results

We subdivided the Mendelian data that include 406 manually annotated "positive" deleterious SNV and more than 14 millions of neutral "negative" SNVs in a training set including about $9/10$ of the available data and a separated test set including the remaining $1/10$ of data, using the same set of genomic features described in [8]. We then compared the *hyperSMURF* results obtained by using the default parameters (i.e. $n = 100$, $f = 2$, $m = 3$) with the *hyperSMURF* results obtained by selecting the "best" learning parameters through cross-validation on the training data. More precisely we considered all combinations of the parameters $n \in \{10, 50, 100, 300\}$, $f \in \{1, 2, 3, 5, 10\}$ and $m \in \{1, 2, 3, 5, 10\}$. The resulting 100 *hyperSMURF* models have been cross-validated on the training set using the Marconi cluster available at CINECA Supercomputing Applications and Innovation Department. It is worth noting that we did not tune the random forest learning parameters, using always for each forest 10 trees and

randomly selecting for each node of the trees 5 features, in order to reduce the complexity of the overall parameter search space.

Fig 1 a) shows the distribution of the cross-validated results across 100 combinations of (n, f, m) triplets of the learning parameters on the training set. The results widely vary, depending on the choice of the parameters, from a minimum of the Area Under the Precision Recall Curve (AUPRC) equal to 0.1741 to a maximum AUPRC equal to 0.4529, achieved with parameters $n = 300, f = 1, m = 10$, i.e. with 300 random forests trained on samples having a doubled number of positive examples ($f = 1$), and a number of negatives 10 times larger than that of positives ($m = 10$), thus reducing in the training set the original imbalance between positive and negatives from about $1 : 36000$ to $1 : 10$.

Fig 1 b) shows the precision/recall curves obtained on the test set by using the default parameters and the best parameters selected by cross-validation on the training data. Here we obtain a significant increment of the AUPRC from 0.3568 to 0.4156 (the difference is statistically significant according to the Wilcoxon rank sum test, $p - value = 10^{-16}$). These results are also confirmed by the $AUROC_{50}, AUROC_{100},$ and $AUROC_{1000}$ results, where *hyperSMURF* with tuned parameters largely and significantly outperforms *hyperSMURF* with default parameters (Table 1). Note that we do not report $AUROC$ results, since in this highly imbalanced context pure $AUROC$ results are not as significant as $AUPRC$ or $AUROC$ limited to the top ranked SNVs [6].

Table 1: Comparison of *hyperSMURF* results obtained respectively with default parameters ($n = 100, f = 2, m = 3$) and with the best parameters obtained by internal cross-validation on the training data ($n = 300, f = 1, m = 10$).

	$AUPRC$	$AUROC_{50}$	$AUROC_{100}$	$AUROC_{500}$	$AUROC_{1000}$
hyperSMURF default par.	0.3568	0.8600	0.9300	0.9091	0.8868
hyperSMURF best par.	0.4156	0.9220	0.9610	0.9407	0.9460

We outline that we previously showed that *hyperSMURF* with default parameters just significantly outperforms existing state-of-the-art methods [3, 9, 2] on the prediction of deleterious variants in Mendelian diseases [7]. Our results summarized in Table 1 show that the proper selection of *hyperSMURF* learning parameters can further significantly improve the overall performance of the method.

Conclusions

This work shows the potentialities of *hyperSMURF* parameters tuning in the context of the detection and prioritization of deleterious genetic variants associated with Mendelian diseases, but we guess that also in other genomic contexts characterized by high imbalance between deleterious and neutral variants fine tuning of *hyperSMURF* parameters may lead to improved results.

Nevertheless the training and testing of hyper-ensembles trained and tested on millions of genetic variants is highly time-consuming: in the context of Mendelian disease, where we used low-dimensional features (more precisely 26 genomic features) training and testing a *hyperSMURF* model required from about 2 to 20 hours of computation with an Intel Xeon Processor $E5 - 2697v4$, with a clock of 2.30 GHz and 128 GB of memory. By enlarging the number of features to thousands (e.g. by using features extracted from DNA with deep convolutional networks [9]), the computational time can further dramatically increase. To overcome these drawbacks, a full parallel implementation of *hyperSMURF* by exploiting High Performance Computing architectures is a research line to be pursued to deeply explore the learning parameters of *hyperSMURF*, as well as the learning parameters of the random forests that constitute the base learners of the hyper-ensemble. On the other hand a parallel implementation could make feasible the automatic adaptation of *hyperSMURF* to different learning tasks and different analyses of genomic big data, ranging from the detection of deleterious variants in genetic diseases to the detection of somatic driver mutations in cancer.

References

[1] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Oversampling Technique. *Journal of Artificial Intelligence Research*, pages 321–357, 2002.

[2] Iuliana Ionita-Laza, Kenneth McCallum, Bin Xu, and Joseph D Buxbaum. A spectral approach integrating functional genomic annotations for coding and noncoding variants. *Nat Genet*, 48(2):214–20, Feb 2016.

[3] Martin Kircher, Daniela M. Witten, Preti Jain, Brian J. O'Roak, Gregory M. Cooper, and Jay Shendure. A general framework for estimating the relative pathogenicity of human genetic variants. *Nat Genet*, 46(3):310–315, Mar 2014.

[4] Ludmila I. Kuncheva. *Diversity in Classifier Ensembles*, pages 247–289. John Wiley & Sons, Inc., 2014.

[5] G.R. Ritchie and P. Flicek. Computational approaches to interpreting genomic sequence variation. *Genome Med.*, 6(87), 2014.

[6] T. Saito and M. Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, 10:e0118432, 2015.

[7] M. Schubach, M. Re, P.N. Robinson, and G. Valentini. Imbalance-aware machine learning for predicting rare and common disease-associated non-coding variants. *Scientific Reports*, doi: 10.1038/s41598-017-03011-5, 2017.

[8] Damian Smedley, Max Schubach, Julius O.B. Jacobsen, Sebastian Köhler, Tomasz Zemojtel, Malte Spielmann, Marten Jäger, Harry Hochheiser, Nicole L. Washington, Julie A. McMurry, Melissa A. Haendel, Christopher J. Mungall, Suzanna E. Lewis, Tudor Groza, Giorgio Valentini, and Peter N. Robinson. A Whole-Genome Analysis Framework for Effective Identification of Pathogenic Regulatory Variants in Mendelian Disease. *The American Journal of Human Genetics*, 99(3):595–606, sep 2016.

[9] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature methods*, 12(10):931–934, August 2015.