Revisiting Semi-Supervised Prediction of SH2-Peptide Interactions from Imbalanced High-Throughput Data

Diego Chialva[1], Marouan Belhaj[2], Jonathan de Wergifosse[2]
[1]VUB, Brussels, [2]ULB, Brussels

Abstract

Protein-protein interactions are at the basis of cellular signal transduction, and thus of a large number of cellular processes, from proliferation to apoptosis. A key aspect of transduction is the binding between certain protein domains and short linear peptides. Src homology 2 (SH2) domains are the largest family of domains binding to peptides that contains phosphotyrosine, instrumental in the transduction regulating several cellular processes. In this work we analyze and improve on a method proposed in [1] to predict the SH2 domains/peptides binding. The method was designed to cope with three main issues: serious unbalance in available datasets, lack of accounting for the amino acids correlations in the peptides, computational complexity. We carefully implemented the method on a different platform (Python 3.4 instead than C) with some personal implementations where heuristics had been involved. Beside re-implementing a *domain-specific* analysis as in [1] (where we studied the different SH2 domains independently) we also formulated an *all-domains*-encompassing approach. We compare our results with those in [1] for the domain-specific analysis, and discuss how our all-domains results complement and improve on those in certain respects (robustness, capability to detect common deeper patterns, improvement on the re-balancing of the datasets). Specifically we obtained 0.98 AUC ROC and 0.99 AUC PR, compared to their averages 0.83 AUC ROC and 0.93 AUC PR, which were however obtained, as we discuss, in a less rigorous way by simple-averaging over inhomogeneous partial results. We discuss possible issues of overfitting deeply connected with the proposed method. Finally, we also performed a genome-wide analysis on a carefully selected dataset of peptides and SH2 domains and present here a view over our results.

1 Introduction

Protein-protein interactions are relevant in several important aspects of cellular biology. A large part of research in the field therefore focuses on this subject, and, as it will appear evident from this work, the relevant issues one faces in this research domain are particularly suited for computational techniques, machine learning and high-throughput data techniques. The transduction of cellular signals is largely mediated by protein-protein interactions, hence a variety of processes (for instance, proliferation, differentiation, growth, apoptosis) depends on that dynamics [2–5]. A typical key aspect of transduction is the binding between particular domains of a protein and short linear peptides [2, 4]. A relevant example is represented by linear peptides with a phophotyrosine residue. Two type of proteins domain bind with such linear peptides: src homology 2 (SH2) and protein tyrosine binding (PTB). In this work we focus on the former case: our goal will be to analyse a computational tool to predict the positive (binding) or negative (lack of binding) interactions between SH2 domains and specific linear peptides.

SH2 domains are a large set (in number of over 120) of structurally preserved protein domains found in several signal transducing proteins, see figure 1 and its caption for a brief description. Each of them is sensitive to very specific peptides, distinguished by the correlations between the amino acid around the phosphorilated tyrosine. The coexistence of these two aspects (that is, having to deal with *i)* large sets and *ii)* with specific interactions) makes computational techniques an important tool of research. However a series of issues complicate this approach. Those issues are related both to features of the experiments providing the relevant data (which ultimately lead to data unbalance in the number of observed positive versus negative interactions), as well as to aspects of the biology (for instance the above-mentioned correlations of the amino acid in the peptide), and also to specific elements of the computational tools (for example the issue of complexity).

The authors of the article [1] proposed a methodology and tool to cope with these issues. We are now going to discuss its features, and how they relate to the specific aspects of our own work on the subject. We will also explain in more details the issues we have briefly mentioned here above, and how these bear upon specific characteristics of the computational approach. Moreover, we will point out potential concerns related to the new method and we will show and discuss how they can possibly be solved.

The proposed approach improves over various methods proposed in the literature to investigate SH2-domains and their interactions, like Scansite and SMALI [20]. Indeed, the latter ones are both *linear* PSSM-based methods, as such unable to account for the correlations between amino acid in the linear peptides relevant for the binding with the SH2 domains. Other methods, like energy models such as [21], have been applied to overcome these problems, but those methods can easily suffer for example from overfitting is-

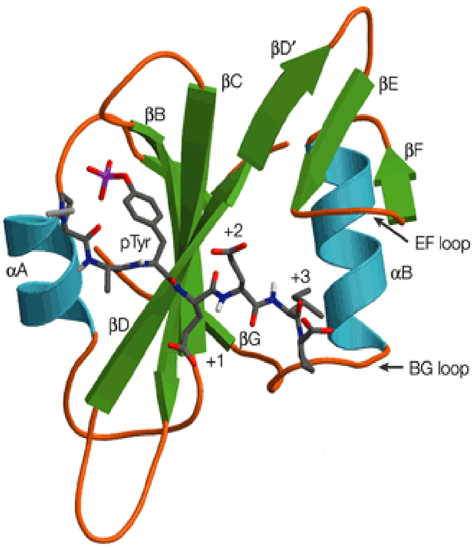

Figure 1: Representation of an SH2 domain binding with a phosphotyrosine, courtes of Nature Reviews, see [2]. Beta-strands are coloured green, alpha-helices blue, and connecting loops orange. Strand, helix and loop notation follows [6]. The bound phosphotyrosine(pTyr) is shown in a stick representation, with carbon atoms coloured grey, nitrogens blue, oxygens red and phosphates magenta.

sues, and also be computationally demanding.

2 Methods

The authors of [1] propose to address the three main issues we mentioned in the previous section in three specific ways:

- Use a non-linear classifier: in this way the correlations among amino acids in the linear peptide can be taken into account.

- Employ a regularized classifier to control the explosion in complexity of the learning approach, due to the above-mentioned correlations and the specific features of the learning problem.

- Apply a policy of re-balancing for the selected dataset based on self-training of the learning device (undersampling is discarded not to loose precious samples in order to avoid having an ineffective statistics).

2.1 Model

The chosen machine learning approach has been a Support Vector Machine classifier. This approach is indeed optimally suited to address the three issues presented here above. Indeed, kernel techniques allow to treat non-linear classification in a very effective way (low VC dimension) and the representation of the hypothesis space and of the sample space can be dealt with efficiently [8–12]. Furthermore, regularisation is straightforwardly implemented in SVMs [8, 9, 11].

It is convenient to illustrate a bit more in details the point concerning the non-linearity. A SVM implements an homonimous algorithm that maps a linear model $f(x)$ sgn$(\langle w, x\langle)$ (where $\langle , \rangle$ is an inner product, w are the weights of the SVM decision surface and x the data instance vectors) into a *quadratic* optimization problem where the weights are traded for coefficients dubbed Lagrange multiplier. he problem is also quadratic in the instance vectors, and according to Kerel Theory it is then possible to substitute the x-space inner product with any kernel function satisfying certain conditions [7–9]. One such exaple is given by polynomial kernels, that are then well suited to treat non-linear problems (for details on the specific kernel we employ, see section).

Having chosen the learning method, we are left with the precise formulation of the learning problem. There are two possible options in this respect. One consists in formulating a separate learning problem for each SH2-domain, and thus building up and training a series of optimal SVMs (one for each SH2 domain considered). We dub this approach the *domain-specific* one. The other approach consists instead in training a single vector machine capable of learning and predicting the binding patterns for the whole set of considered SH2 domains altogether. We dub it the *all-domains* approach.

The two approaches have both drawbacks and advantages. We will leave a more detailed commenting for the discussion sections, but it is important to present a few points here, to facilitate the comprehension of the following analysis for the readers' sake.

The *domain-specific* approach allows to deal with a series of smaller and more rapidly solvable learning problems, possibly also being sensitive to important specific features concerning the protein interaction for the specific domains. On the other hand, it implies using much smaller statistics (smaller datasets) and thus generally would yield less robust results. Furthermore, the small datasets relative to each SH2 domain exacerbate the issues of unbalance (some datasets do contain sample of essentially just one class) with overbearing on questions such as overfitting and others, see section 2.5. Finally, the results for the different domains cannot be really combined in a clear-cut way because they are not homogeneous: the n learning problems leading to n different trained SVMs deal indeed with different questions (different SH2 domains) and with different statistics (different databases for each domain, representative of different peptides distributions due to the various experiments).

The *all-domains* approach has the advantage of using a larger statistics and thus being generally more robust and ameliorating the unbalance problem. Furthermore, it allows to draw statistics and probabilities concerning all domains in an homogeneous and statistically correct way, and thus could better show common patterns across domain bindings and general deep features. It is however more computationally intensive, in particular for its memory footprint.

The analysis in the original paper [1] dealt only with the domain-specific approach. We defer the discussion of the issues and specific points of this choice to section 3.2.

2.2 Algorithm

The algorithm must cope with at least two major issues:

- the learning problem itself;
- the unbalance of the datasets.

Therefore, it is rather elaborate, consisting of five different parts, with several cycles of training/testing and prediction. For ease of comprehension we have decided to illustrate the algorithm via a diagrammatic representation, see figure 2 and provided more detailed explanations in sections 2.4, 2.5, 2.6 when we deal with the specific points.

2.3 Datasets

The learning problem we address is of the semi-supervised type. Indeed, a purely supervised learning approach is unfeasible, as we will show, due to the severe unbalance of the datasets. Our data is composed of four datasets having two different origins: two from microarray experiments and the other two from High Density Peptide Arrays databases, as in [1]. Our datasets have been kindly provided by the authors of [1] themselves, although they provided us with databases which are slightly (but in the end not appreciably) different from those they employed in their original analysis.

The unbalance in the dataset between the number of positive an negative interactions is determined by the state and nature of the relevant experiments and their reports. In the case of the microarray experiments, they do provide affinities, so that both negatives and positives can in principle be inferred, but they generally suffer from a low signal-to-noise ratio which makes the results inconsistent, see for example [14, 15]. Pool-oriented peptide arrays instead give evidence only for positive interactions, and nothing can be inferred about negatives. see for instance [33, 34]. Similarly, high density peptide array experiments do not report affinities, and so are in the same situation. Moreover, w would like to point out that the different origins of the datasets imply also inhomogeneities in the data distributions. Thus different magnitude and relevance of noise, for instance, have to be expected. We will use regularised vector machine, which will help coping with this issue.

More relevant is the problem of data unbalance. The datasets present an overwhelming majority of positive (binding) classifications compared to the negative ones (6792 total positives versus 2523 negatives). This has a negative effect leading to overfitting and loss of generalisation power, as we will discuss in section 2.5 where we will present the chosen approach for re-balancing using a semi-supervised viewpoint.

The actual datasets employed for our analysis are as follows.

- *Dataset I*

  The first dataset has been built from High Density Peptide Arrays in the NetPhorest database [13]. It comprises 14678 interactions, based on 61 SH2 domains and 920 phosphorylated peptides, from which 6792 positive interaction where finally selected. The database also provided a series of instances with no positive evidence, which therefore have been selected as unclassified (unlabeled) instances (in number of 38287).

- *Dataset II*

  The first microarray experiment [14] has considered 115 SH2 domains and 20 singly phosphorylated peptides from ErbB2 and ErbB3 proteins. 10 cases where a single protein has a SH2 domain with both C-terminal and N-terminal have been discarded since the dataset does not specify to which terminal a peptide bounds. In the end 1020 interaction (0 positives and 851 considered negatives in [1]) have been collected.

- *Dataset III*

  The second microarray experiment [15] has yielded information about 85 SH2 domains and 41 singly phosphorilated peptides from EGFR, FGFR, IGIFR proteins. This experiment accounts for 1886 interactions (0 positives and 1672 negatives).

- *Dataset IV*

  The dataset is taken from PhosphoELM [16]. It has been created selecting 28 SH2 domains that were also present in the other datasets and 339 peptides to yield 878 positive interactions. From these interactions 6 have been discarded due to a problem in the encoding of the data which makes making those instances unusable.

  This dataset has been used only for a final testing of our model: a simple count of the correct predictions, given that the dataset contains only positive-classified instances and is thus unsuitable for a full statistical analysis.

- *DatasetV*

  The last dataset is a collection of unknown interaction used in [1] to make a new set of binding predictions once the SVM has been trained and tested. The database has been obtained from two sources ([17,18]) being careful in selecting SH2 domains and linear pepetides that were indeed containing verified phosphorylated tyrosine and that had the same cellular localisation (using the annotations in [19]). In this case we have, from the [1] additional material, the list of domains and peptides, but they are unlabeled. We have therefore run our prediction code over the

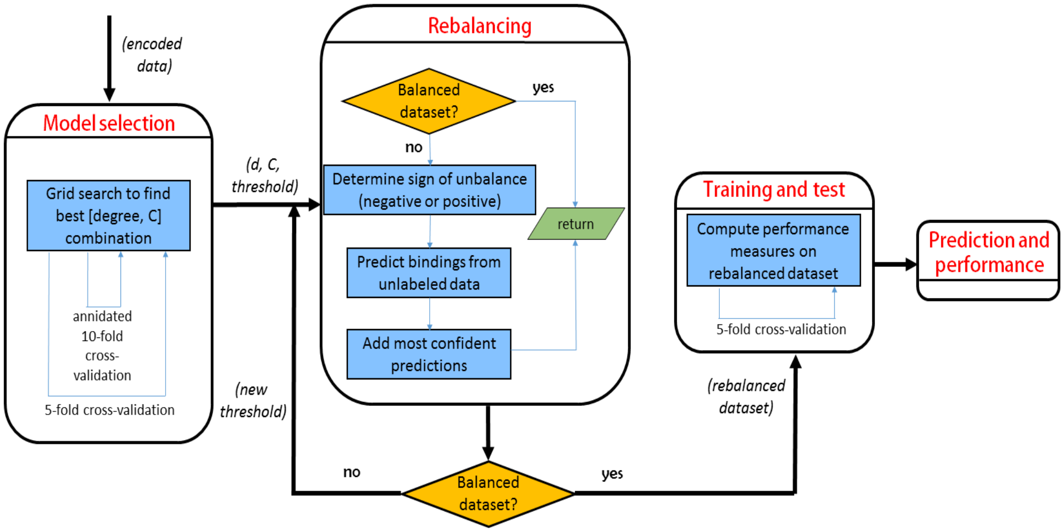
(encoded data)
Model selection
Grid search to find best [degree, C] combination
annidated 10-fold cross-validation
5-fold cross-validation
(d, C, threshold)
(new threshold)
Rebalancing
Balanced dataset?
yes
no
Determine sign of unbalance (negative or positive)
return
Predict bindings from unlabeled data
Add most confident predictions
no
Balanced dataset?
yes
Training and test
Compute performance measures on rebalanced dataset
5-fold cross-validation
(rebalanced dataset)
Prediction and performance

Figure 2: Schematisation of the algorithm. It consists of several connected parts that are explaied in details in the test in the respective sections. The most critical and delicate points are those of model selection and rebalancing, because of the serious imbalance in the available datasets and the possible consequences for the whole learning and analysis.

dataset and counted the number of positive and negative predictions. Lacking feedback from the author of [1], we can imagine that the additional material only contained newly classified positive cases.

When combining the datasets, some overlaps an inconsistencies have considered the removal of part of the interactions. We kindly received from the authors of [1] the already compiled datasets, after the removal of the inconsistencies. We present the final view over the datasets number of instances and classifications in table 1.

2.4 Model selection

The machine learning process starts with the model selection part, where the best model hyperparameters are selected. In our case those are the hyperparameters of polynomial-kernel support vector machines $k(x,x) = (x \cdot x + 1)^d$, the most relevant ones being the degree d of the polynomial kernel and the penalty factor C which accounts for regularisation and misclassifications[1]. Model selection is important in coping with overfitting, allowing to trade off between performance and complexity, and several techniques have been developed for this scope.

The model selection has been based on datasets I-II-III. We employed a *grid search* with grid values

$$d = 1,\ 2,\ 3 \tag{1}$$

$$C = 0.01,\ 0.1,\ 1,\ 10\,. \tag{2}$$

based on a double stratified cross- validation procedure.

More in details, we applied a 5-fold cross-validation where we take the input, split it in five parts and, along five cycles, use 4/5 of the data as training. The remaining 1/5 is used in turn as test. For each one of the five cycles, we then apply a 10-fold cross-validation for all the possible combinations of (d, C) values in the grid. We score each model by a ROC AUC measure for each of the five training sets independently, select the model maximizing this measure and finally pick up the most frequent (d, C) combination. We have applied this technique both to the *domain-specific* and *all-domains* approaches.

2.5 The issue of rebalancing

The features of the relevant biological experiments and reports create datasets with a severe unbalance between the number of positive-classified interactions (that is, positive bindings between SH2 domains and peptides) and negative-classified ones (no binding). As we mentioned in section 2.3, microarray experiments for instance, while in principle capable of dealing with the negative cases as well, as they indicate affinities, are too conditioned by a low signal-to-noise ratio and thus cannot always reliably be used in that sense [14, 15]. Peptide pool arrays are on their part effective at indicating positive interactions, but not negative ones [33, 34].

Dataset unbalance is however very detrimental to the performance of computational machine learning techniques. Indeed, the algorithms tend to overfit on the majority classes, which are over-represented compared to the minority ones in an unbalanced dataset. This leads to at least two problems:

[1] The authors of [1] fix the affine and linear term coefficient for the polynomial kernel to 1 by hand and do not model-select them, differently from other authors.

	Original Data						Selected data					
Datasource	# D	# P	# I	# Pos	# Neg	# Ukn	# D	# P	# I	# Pos	# Neg	# Ukn
NetPhorest	61	920	56120	7544	-	48576	51	880	448800	6792	-	38287
Microarray experiment I	105	20	2100	160	1940	-	51	20	1020	-	851	-
Microarray experiment II	85	41	3485	314	3171	-	46	41	1886	-	1672	-
PhosphoELM	63	359	-	878	-	-	28	197	-	339	-	-

Table 1: Data as present in the literature and as selected for this work. The meaning of the labels is as follows: # D =number of domains, # P = number of peptides, # I = number of interactions, # Pos = number of positive interactions, # Neg = number of negative interactions, # Ukn =number of unknowns. To see the (minor) differences with the selection presented in [1], see their corresponding table 3.

- it is difficult to select an effective measure of performance of the algorithm;
- it is difficult to correctly partition between train and test sets.

Possible counter measures to this situation are *undersampling* or *re-sampling*. In the first case, however, one looses precious information coming from the biological experiments, which in turns affect the learning and generalisation power of the model. In the second case, one can employ several different techniques for re-sampling. Instead of artificially create plausible negative interactions (the minority class in our case), the authors of [1] suggested to use the unlabeled samples already present in the experimental datasets to predict new possible negative cases. In this way the learning algorithm becomes a kind of semi-supervised one, aimed at obtaining information on the minority class. Also for this target a number of techniques are available; most of them, however, require certain assumptions on the underlying data, which, if not correct, would harm the final predictions, see [35]. For this reasons, following [1], we have relied on

i) the discriminative capabilities of the classifier (SVMs) itself after model selection,

ii) the abundance of unlabeled data from the high density peptide array experiments.

The key-points of our re-balancing are therefore the 1) classification of the unlabeled instances, 2) scoring of the robustness of the classifications, 3) selection of the best new labeled negative instances (those that are the minority class in the original unbalanced dataset) and their adding to the original dataset.

As we will discuss in more details in sections 3.2 and 4, there could be/are some differences between our algorithm and the one adopted in [1]. In fact, the authors of [1] do not specify their stopping mechanism to the re-balancing process, and mention bootstrapping in case of positive interactions as the minority class, see figure 1 in [1]. In our case we have configured our algorithm in such a way that the process is then iteratively repeated until a certain minimal degree of inbalance is achieved. The score of robustness of the classification has been the distance from the discriminative hyperplanes and we have found best results (as measured by the number of iteration of re-training and re-balancing, which increase the stepwise predictive power of the machine) by choosing a few percent (10% maximum) of final allowed unbalance, both in the *domain-specific* and *all-domains* approaches. The specific details of our implementation will be presented together with the results, because they are important to fully appreciate the final results and comments.

2.6 Training

The training of the model have been performed in two different ways: first, we have adopted a stratified 5-fold cross-validation procedure (although the stratification is much less relevant once the dataset rebalancing has been achieved). We have scored our testing with the five performance measures detailed in section 2.7. Later, we have also performed an analysis in terms of a random split into train and test datasets, with a 75%-25% division pattern. The results have been completely concordant between the two approaches.

2.7 Evaluation of the final predictive performance

Once we can work on a re-balanced dataset we can more robustly progress with our learning problems (both the *domain-specific* and *all-domains* one). We have then used, following [1], five different performance evaluation measures:

- *sensitivity/recall*= $\frac{\text{TP}}{(\text{TP+FN})}$,
- *precision*= $\frac{\text{TP}}{(\text{TP+FP})}$,
- *specificity*= $\frac{\text{TN}}{(\text{TN+FP})}$,
- *receiver operating characteristics curve (ROC) and its area under the curve (AUC ROC)*: the curve is obtained plotting Sensitivity/True Positive Rate (TPR) versus to False Positive Rate/1-Specificity (FPR), larger area values representing better classifiers.

- *precision recall curve (PR) and its area under the curve (AUC)*: the curve is obtained plotting the precision in function of the recall.

In the above equations, we have defined TN (true negative) as SH2 domain-peptide interactions correctly predicted as not binding, TP (true positive) as positive binding interactions correctly predicted, while FP (false positives) and FN (false negatives) are the false prediction in the corresponding cases.

2.8 Implementation of the model

We have implemented the learning model in Python 3.4, taking advantage of the statistics library scikit-learn [25, 26], which provided a support vector machine library implementation. The code of the algorithm was written entirely by scratch.

Data encoding The encoding of the data proved to be a crucial point in the computational performance of our algorithm. We have coded each instance as follows, respectively for the domain-specific and all-domains approaches, see section 2.1. The different peptides are represented as vectors of 20x6 elements (where 20 accounts for all possible amino acid and 6 is the actual number of amino acid considered in the peptides, the central tyrosine is neglected because always present). These vectors are presence vectors: in each block of 20 elements there is a 1 in correspondence of the present amino acid in the peptide, and zero everywhere else.

This coding of the peptides has been equally used both in the domain-specific as well as in the all-domains approaches. In the latter case, each single instance had however a more complex structure, as it had also had to account for the relevant SH2 domain. As we have considered 51 such domains, as in [1], the final vector in the all-domains analysis comprises 51+20x6 = 171 elements. Also for the 51 SH2 domains component we have used presence vectors.

Finally, the classification labels are 1 for positive instances and -1 for negative instances.

3 Results

In this section we present the detailed results of both the domain-specific and the all-domains analysis. We will also compare our results with those obtained in [1]. We provide an in-depth discussion of possible remaining issues related to the adopted method, possible solutions and finally outlooks.

3.1 Model selection results

The hyperparameters selection gave the following results:

- for the all-domains analysis: $C = 0.1$ and $d = 3$
- for the domain-specific analysis: the series of best C and d values for each individual SH2 domain shows that overwhelmingly the best options vary, depending on the dataset, between the quadratic and the linear case ($d = 2, 3$) and with small penalty factors (ranging from $C = 0.01$ to $C = 0.1$). In a very few cases, as also in [1] the model selection algorithm has picked up a linear classifier.

It appears evident from these results that non-linearity is indeed essential for the best generalization power of the models for this learning problem.

3.2 Testing and performance evaluation

We have tested and evaluated our models with the metrics presented in section 2.7. We also compared the results of our analysis with the one in [1]. Indeed, one could expect somewhat different results, given the different implementation of the algorithm (in python 3.4 in our case, with a C library in their), and , more importantly, given that our heuristic strategy for the re-balancing might be different than the one applied in [1] and not detailed by the authors (see section 4.1).

Furthermore, our analysis went beyond the one described in [1] (which was only domain-specific), since we have provided also an all-domains analysis. We will discuss later on the relevance of this new analysis.

In table 2, we present our results for sensitivity and specificity for the domain-specific analysis, and compare with the values obtained in [1]. The average values and errors are obtained over the 5-fold cross-validation procedure. We see that the results are matching those of [1], but in a few cases they are actually improving on them. We also provide two graphs, see figure 3, illustrating the distribution in specificity values of our analysis, as compared with that in [1] for the domain-specific analysis.

We also calculated the ROC and PR curves for each single domain over the 5-fold procedure, which we show at the end of this article, in figures 5 and 6. Our plots are matching quite well those of [1] (see their additional material files figures S2 and S3).

In our study we have also improved on their result, through the *all-domains* analysis. We will discuss in the final section 4 the advantages that this kind of analysis brings about. We will also discuss the serious issues that could descend from the dataset unbalance and re-balacing procedure. For the moment we show the plots for the ROC and PR curves superimposed to the average ROC curve and PR curve shown in [1], see figure 4.

We also note that [1] does not specify in details how the average ROC and PR curves have been obtained by averaging over the individual-domain ROC and PR curves. The approach is very different from ours: in our case we formulate an all-domains-encompassing learning problem which is well-defined, whereas in their case they make a simple average average (from what is possible to infer from their

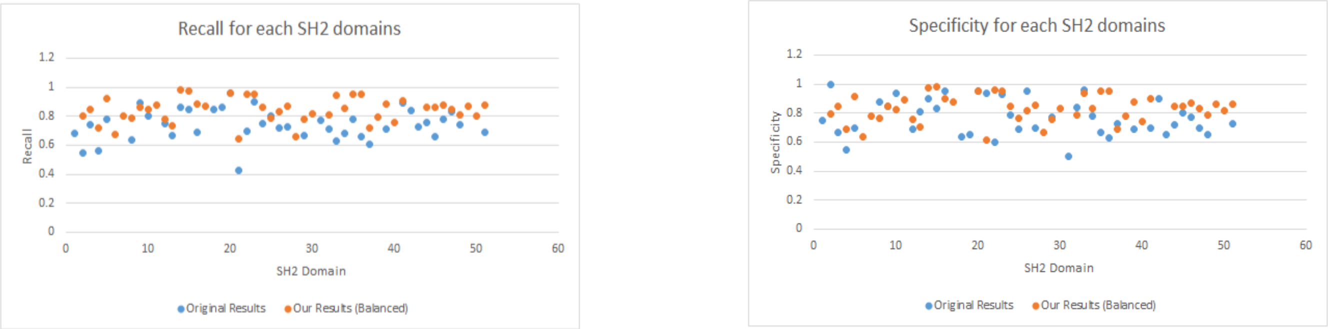

Figure 3: Comparison among our results for specificity and recall and those in [1] after re-balancing in the domain-specific analysis.

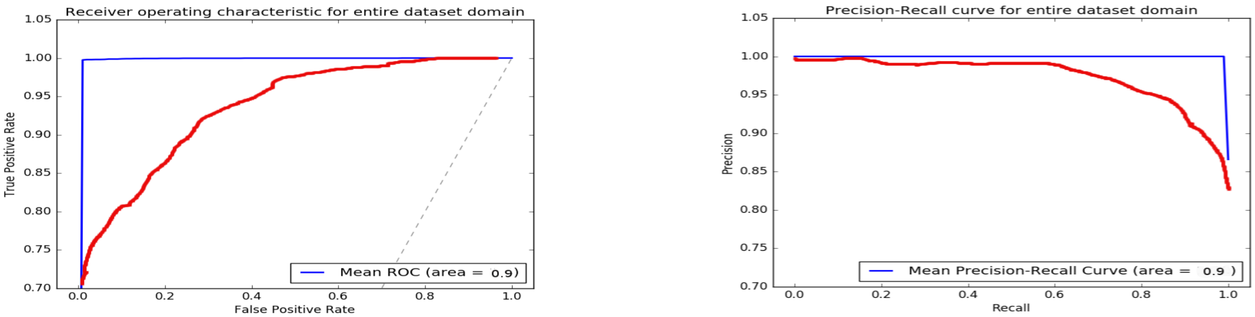

Figure 4: Comparison among our results for ROC and PR curves (in blue) and those of [1] (in red) after re-balancing in the all-domains analysis.

article) over the results for the 51 one *different* learning problems concerning each single SH2 domain considered in the work. However, such an averaging procedure is dubious under the point of view of consistency and robustness of the approach. Indeed, the learning problems they deal with are truly different problems, with different hypothesis domains (one for each SH2 domain) and different dataset statistics (see also their additional material, and how the unbalance, just to name one characteristic, is different among the various domain-specific datasets). We believe therefore that our approach is more sound in this respect.

Finally, we have perfomed a genome-wide analysis, domain per domain. We have found an overwwhelming number of positive (binding) instances (a few reuslts are as follows: ABL2 0 500 500 BCAR3: 8 neg, 378 pos; BMX: 81 neg, 419 pos; BTK: 3 neg, 497 pos; CRKL: 6 neg, 494 pos; E105251: 1 neg, 499 pos; E109111: 30 neg, 470 pos; FES: 6 neg, 494 pos; GRB2 3 neg, 497 pos; NCK1: 9 neg, 491 pos; VAV1: 84 neg, 416 pos; 500 VAV2: 70 neg, 430 pos). Unfortunately we cannot compare our results with those of [1] because they do not provide a file where all the classifications they have obtained are clearly indicated.

4 Discussion

In this section we will present our comments regarding the learning tool and algorithm analysed in our work and the final results, in comparison with those in [1], pointing out the improvements we have made to the open problems of that article. We will also discuss the serious potential issues connected with the proposed method. We will conclude with some final outlooks.

4.1 Re-balancing and related issues

The rebalancing phase of the algorithm is clearly its most delicate part. The new negative predictions to be added to the original dataset for re-balancing are determined by the same classifier that will be later on re-trained on the *same* rebalanced dataset itself.

This opens the way to some significant questioning. Indeed, in our opinion it is reasonable to suspect an increase in the bias of the model, because it is difficult to say that the balanced train and test datasets used for the final training (be it with 5-fold cross-validation or random splitting) are actually independent from each other: they all originate from the same unbalanced dataset via the action (self-training) of the same SVM model. This seems a potentially severe issue, because, as it is well-known, in order to correctly evaluate a models and assess its generalizing power the test set should be prepared without any connection with the train analysis. Here it is difficult to argue the case. We may argue that the issue arises only at the level of the hyperparameters, which remain indeed the same throughout all the re-balancing and later training phases, and thus a Bayesian analysis (if one wants to cast the problem in that framework) might help quantifying the actual bias introduced by the algorithm. Such analysis is however beyond the scope of this work and we defer it to another analysis.

Another significant question related to the specifics of the algorithm implementation arises as a consequence of the lack of details of the re-balancing procedure in [1]. Indeed, the newly added negative cases are selected on the basis of a best confidence score. It is however by no means obvious how to choose the threshold over a confidence score, nor how robust is the choice, made in in [1] of the distance from the discriminating hyperplane of the new negative instances. In particular, since no clue about their decision of a possible threshold has been provided in [1], we opted for a *heuristic* approach: we iteratively increased the threshold from some tightening value, finally choosing a best value based on the cross-fold procedure used in the iterative re-training at every step of the re-balancing. We also opted for a stop mechanism based on prefixed final minimal unbalance (fixed to be less or equal than 10%).

However, some of the issues caused by data unbalance did show up in our analysis. Generally, the issues one would expect are a consequence of the fact that a learning machine trained on an unbalance dataset would overfit on the majority class. Indeed, in our domain-specific analysis, for a handful of domains (six) the machine was so overfitted on the majority class that the specificity score was much lower than that of the other domains, signalling a problem in the estimation of the true negatives in the final training and testing phase. This problem is not resolvable in the framework of the proposed model, which utilizes the presented re-balancing procedure. It would be interesting to try other re-balancing approaches, although methods like expectation maximisation CITE rely on important assumptions that need to be confirmed in the scenario of SH2 bindings studied in this work.

4.2 Results, potential issues and outlooks

The result obtained by us and in [1] are comparable, for what concerns the domain-specific analysis. It is however interesting to look at how the two works have dealt with the whole of the considered domains. In the case of [1] the authors have adopted a simple procedure of averaging over the results of the single domains, for what it is possible to infer from their article (see for example their table 1, but also their results for the ROC and PR curves). As we have already mentioned, such averaging is dubious, give that the learning problems for each domain are indeed truly different problems and produce different models.

In our case, we have opted for a true all-domains-wide analysis. The results we have shown are remarkably positive. In fact, the doubt arises that the re-balancing procedure is not sufficient to cure the serious issues that the original imbalance might have caused, and that could influence the final results being based to the models selection phase (which, as we recall, has unfortunately to be performed on the unbalanced dataset). We believe such issues need to be carefully addressed in future works. We also believe that the true solution to these questions will come from the progress in the experimental techniques, that hopefully will ameliorate the unbalance issue in the future.

5 Acknowledgments

We acknowledge correspondence with Professor Kousik Kundu, through Professors Tom Lenaerts and Professor Wim Vranken.

References

[1] Kundu K, Costa F, Huber M, Reth M, Backofen R. *Semi-Supervised Prediction of SH2-Peptide Interactions from Imbalanced High-Throughput Data*. Kurgan L, ed. PLoS ONE. 2013;8(5):e62732. doi:10.1371/journal.pone.0062732.

[2] http://www.nature.com/nrm/journal/v3/n3/box/nrm759_BX1.html

[3] Seet BT, Dikic I, Zhou MM, Pawson T (2006) Reading protein modifications with interaction domains. Nat Rev Mol Cell Biol 7: 47383.

[4] Schlessinger J, Lemmon MA (2003) SH2 and PTB domains in tyrosine kinase signaling. Sci STKE 2003: RE12.

[5] Porter AC, Vaillancourt RR (1998) Tyrosine kinase receptor-activated signal transduction pathways which lead to oncogenesis. Oncogene 17: 134352.

[6] Nature Reviews Molecular Cell Biology 3, 177-186 (March 2002).

[7] Tom Mitchell (1997), “Machine learning”-McGraw-Hill.

[8] Simon Haykin (1999), “Neural networks”-Prentice Hall.

[9] Russel S.J., Norvig P. (2010), “Artificial Intelligence”-Prentice Hall.

[10] Statnikov, A. and Aliferis, C.F. and Hardin, D.P. *A Gentle Introduction to Support Vector Machines in Biomedicine: Theory and methods*. 2011 World Scientific

[11] Schlkopf B, Smola AJ. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Cambridge, Mass: MIT Press; 2002.

[12] Zhang, J. and Marszalek, M. and Lazebnik, S. and Schmid, C. *Local features and kernels for classification of texture and object categories: A comprehensive study International Journal of Computer Vision* 2007

[13] Miller ML, Jensen LJ, Diella F, Jorgensen C, Tinti M et al. *Linear motif atlas for phosphorilation-dependent signaling*. 2008 Sci Signal 1: ra2. doi: 10.1126/scisignal.1159433

[14] Jones RB, Gordus A, Krall JA, MacBeath G *A quantitative protein interaction network for the ErbB receptors using protein microarrays*. 2006 Nature 439: 16874. doi: 10.1038/nature04177

[15] Kaushansky A, Gordus A, Chang B, Rush J, MacBeath G *A quantitative study of the recruitement potential of all intracellular tyrosine residues on EGFR, EGFR1 and IGF1R*. 2008 Mol Biosyst 4: 64353. doi: 10.1039/b801018h

[16] Diella F, Gould CM, Chica C, Via A, Gibson TJ *Phospho-ELM: a database of phosphorylation sites-update* 2008 36: D2404. doi: 10.1093/nar/gkm772

[17] Magrane M, Consortium U *UniProt Knowledgebase: a hub of integrated protein data*. Database (Oxford) 2011: bar009. doi: 10.1093/database/bar009

[18] Hornbeck PV, Kornhauser JM, Tkachev S, Zhang B, Skrzypek E, et al. *PhosphoSitePlus: a compre-hensive resource for investigating the structure and function of experimentally determined post-translational modifications in man and mouse*. 2012 40: D26170. doi: 10.1093/nar/gkr1122

[19] Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, et al. *Gene ontology: tool for the unification of biology. The Gene Ontology Consortium*. 2000 Nat Genet 25: 259.

[20] Li L, Wu C, Huang H, Zhang K, Gan J, et al. *Prediction of phosphotyrosine signaling networks using a scoring matrix-assisted ligand identification approach*. 2008 36: 326373. doi: 10.1093/nar/gkn161

[21] Wunderlich Z, Mirny LA Using genome-wide measurements for computational prediction of SH2-peptide interactions. 2009 37: 462941. doi: 10.1093/nar/gkp394

[22] Liu BA, Jablonowski K, Shah EE, Engelmann BW, Jones RB, et al. SH2 domains recognize con-textual peptide sequence information to determine selectivity. Mol Cell Proteomics 2010 9: 2391404. doi: 10.1074/mcp.m110.001586

[23] Alex J. Smola, Bernhard Schlkopf *A Tutorial on Support Vector Regression*. Statistics and Computing archive Volume 14 Issue 3, August 2004, p. 199-222

[24] Chih-Chung Chang and Chih-Jen Lin *A Library for Support Vector Machines*. 2013 Department of Computer Science, National Taiwan University, Taipei, Taiwan

[25] http://scikit-learn.org/stable/

[26] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. *Scikit-learn: Machine Learning in Python*. 2011 Journal of Machine Learning Research 12: 2825-2830

[27] Machida K, Thompson CM, Dierck K, Jablonowski K, Krkkinen S, Liu B, et al. *High-throughput phosphotyrosine profiling using SH2 domains*. Molecular cell. 2007;26(6):899.

[28] Tinti M, Kiemer L, Costa S, Miller ML, Sacco F, Olsen JV, et al. *The SH2 domain interaction landscape*. Cell reports. 2013;3(4):1293-305.

[29] Obenauer JC, Cantley LC, *Yaffe MB Scansite 2.0: Proteome-wide prediction of cell signaling inter-actions using short sequence motifs*. 2003 31: 363541. doi: 10.1093/nar/gkg584

[30] He H, Garcia EA *Learning from imbalanced data*. 2009 IEEE Transactions on Knowledge and Data Engineering 21: 12631284. doi: 10.1109/tkde.2008.239

[31] Provost F Machine learning from imbalanced data sets 101. 2000 In: Proceedings of the AAAI-2000 Workshop on Imbalanced Data Sets.

[32] Jo Japkowicz Class imbalances versus small disjuncts. 2004 In: ACM SIGKDD Explorations Newsletter.

[33] Rodriguez M, Li SSC, Harper JW, Songyang Z (2004) An oriented peptide array library (OPAL) strategy to study protein-protein interactions 279: 88027. 23.

[34] Huang H, Li L, Wu C, Schibli D, Colwill K, et al. (2008) Defining the specificity space of the human SRC homology 2 domain. Mol Cell Proteomics 7: 76884.

[35] Zhu X (2005) Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.

	Original Results		Our Results	
SH2 Domain	Recall	Specificity	Recall	Specificity
ABL1	0.68±0.14	0.75±0.14	-	-
ABL2	0.55±0.17	1±0	0.80±0.17	0.79±0.18
APS	0.74±0.13	0.67±0.14	0.85±0.12	0.85±0.12
BCAR3	0.56±0.09	0.55±0.18	0.72±0.18	0.69±0.20
BLK	0.78±0.11	0.7±0.19	0.92±0.11	0.92±0.12
BMX	-	-	0.67±0.09	0.63±0.10
BRDG1	-	-	0.80±0.08	0.78±0.09
BTK	0.64±0.2	0.88±0.1	0.79±0.12	0.77±0.14
CRK	0.89±0.05	0.85±0.13	0.86±0.21	0.84±0.23
CRKL	0.8±0.12	0.94±0.09	0.84±0.13	0.83±0.15
CTEN	-	-	0.88±0.05	0.89±0.05
E105251	0.75±0.06	0.69±0.09	0.78±0.18	0.75±0.20
E109111	0.67±0.15	0.81±0.13	0.73±0.11	0.70±0.12
E185634	0.86±0.05	0.9±0.14	0.98±0.03	0.98±0.03
EAT2	0.85±0.11	0.83±0.1	0.97±0.03	0.98±0.03
FER	0.69±0.12	0.95±0.05	0.89±0.08	0.90±0.07
FES	-	-	0.87±0.10	0.88±0.09
FGR	0.85±0.09	0.64±0.15	-	-
FRK	0.86±0.07	0.65±0.25	-	-
GRAP2	0.96±0.04	0.95±0.08	0.96±0.03	0.95±0.04
GRB2	0.9±0.06	0.93±0.04	0.95±0.04	0.95±0.05
GRB10	0.43±0.16	0.94±0.09	0.65±0.11	0.61±0.12
GRB14	0.7±0.13	0.6±0.18	0.95±0.03	0.96±0.03
HCK	0.75±0.08	0.79±0.21	0.86±0.20	0.85±0.22
INPPL1	0.8±0.07	0.69±0.16	0.78±0.17	0.76±0.19
ITK	0.72±0.11	0.95±0.06	0.83±0.09	0.81±0.10
LCK	0.73±0.08	0.7±0.09	0.87±0.20	0.85±0.22
LCP2	-	-	0.66±0.04	0.67±0.04
LYN	0.67±0.18	0.77±0.12	0.78±0.12	0.76±0.13
MATK	-	-	0.82±0.08	0.84±0.07
MIST	0.77±0.07	0.5±0.5	-	-
NCK1	0.71±0.13	0.84±0.14	0.81±0.13	0.79±0.15
NCK2	0.63±0.09	0.96±0.06	0.94±0.02	0.94±0.02
PTK6	0.68±0.1	0.78±0.19	0.85±0.21	0.84±0.24
SH2B	0.78±0.06	0.67±0.19	0.95±0.04	0.95±0.04
SH2D1A	0.66±0.08	0.63±0.21	0.95±0.08	0.95±0.09
SH2D2A	0.61±0.1	0.73±0.18	0.72±0.23	0.69±0.25
SH2D3C	-	-	0.80±0.08	0.78±0.09
SHC1	0.71±0.12	0.69±0.17	0.89±0.11	0.88±0.12
SHC3	-	-	0.75±0.07	0.75±0.07
SOCS2	0.89±0.1	0.7±0.21	0.90±0.18	0.90±0.20
SOCS5	0.84±0.12	0.9±0.22	-	-
SRC	0.73±0.08	0.65±0.21	-	-
TEC	0.76±0.11	0.72±0.08	0.86±0.17	0.84±0.19
TENC1	0.66±0.07	0.8±0.12	0.86±0.20	0.85±0.22
TENS1	0.78±0.11	0.77±0.15	0.88±0.08	0.87±0.09
TNS	0.83±0.04	0.7±0.09	0.85±0.17	0.83±0.18
TXK	0.74±0.11	0.65±0.12	0.81±0.16	0.79±0.17
VAV1	-	-	0.87±0.08	0.86±0.08
VAV2	-	-	0.80±0.11	0.81±0.10
YES1	0.69±0.12	0.73±0.21	0.88±0.14	0.87±0.15
Average	0.74	0.77	0.84	0.77

Table 2: Results for specificity and recall/sensitivity compared to the results in [1]..

PR Curve achieved by the SVM for each SH2 domain

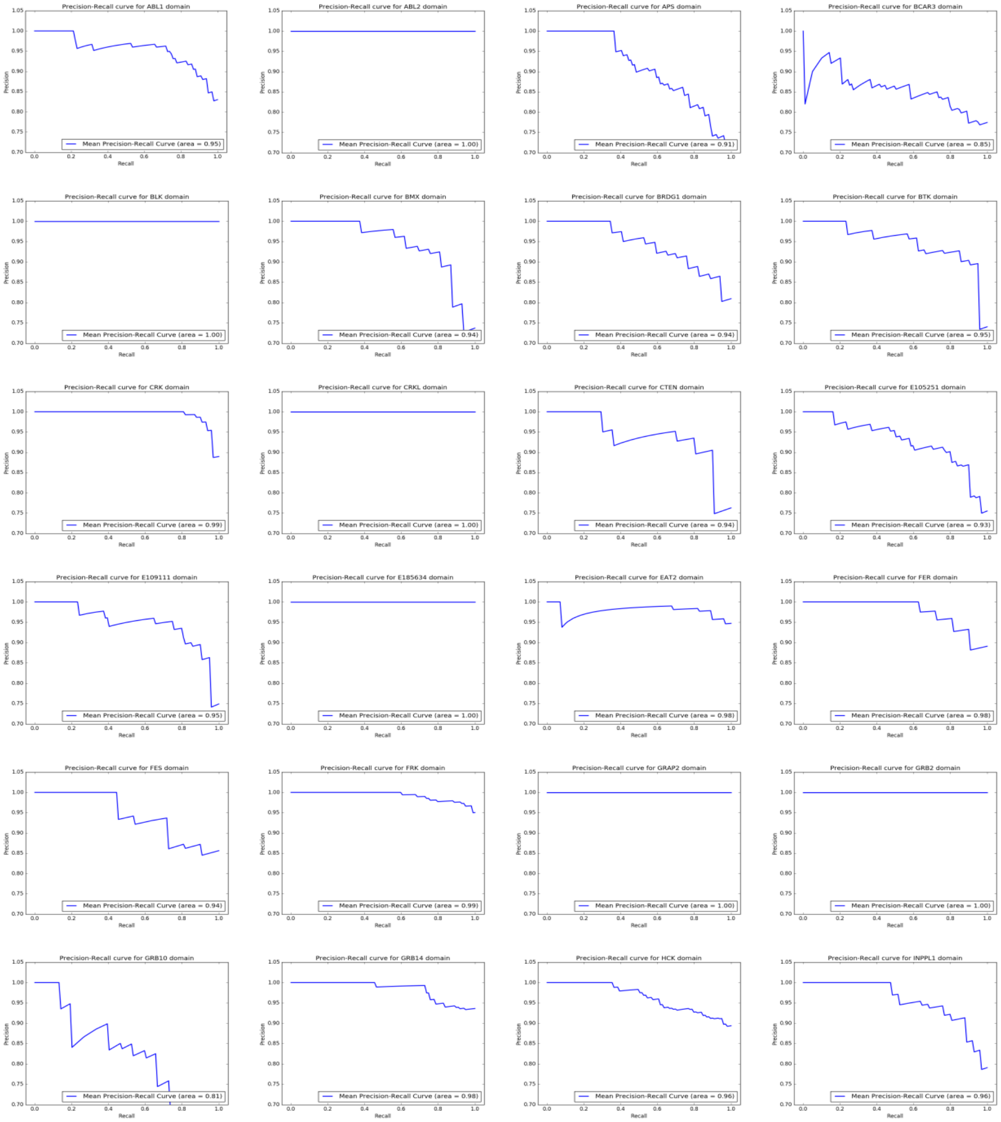

Figure 5: PR curves after re-balancing in the domain-specific analysis.

ROC Curve achieved by the SVM for each SH2 domain

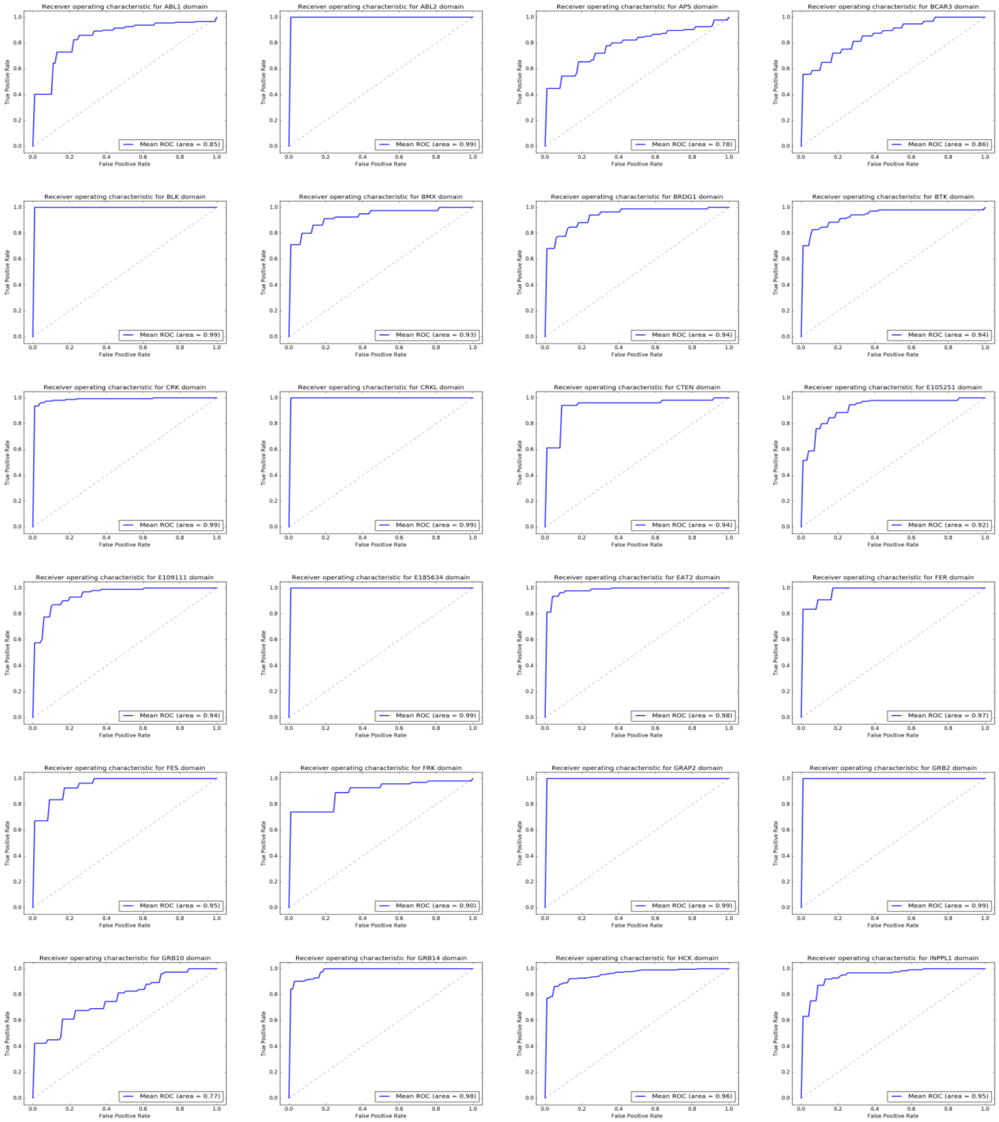

Figure 6: ROC curves after re-balancing in the domain-specific analysis.