Evaluation of Machine Learning Algorithms on Protein-Protein Interactions

Conference Paper · January 2014
DOI: 10.1007/978-3-319-02309-0_22

7 authors, including:

Indrajit Saha
National Institute of Techical Teachers Training and Research, Kolkata
79 PUBLICATIONS **625** CITATIONS
SEE PROFILE

Tomas Klingström
Swedish University of Agricultural Sciences
16 PUBLICATIONS **83** CITATIONS
SEE PROFILE

Simon Forsberg
Princeton University
18 PUBLICATIONS **120** CITATIONS
SEE PROFILE

Julian Zubek
University of Warsaw
17 PUBLICATIONS **37** CITATIONS
SEE PROFILE

Molecular BioSystems

PAPER

View Article Online
View Journal | View Issue

Cite this: *Mol. BioSyst.*, 2014, **10**, 820

Ensemble learning prediction of protein–protein interactions using proteins functional annotations†

Indrajit Saha,‡*[ab] Julian Zubek,‡[c] Tomas Klingström,[d] Simon Forsberg,[e] Johan Wikander,[f] Marcin Kierczak,[e] Ujjwal Maulik[b] and Dariusz Plewczynski*[agh]

Received 1st November 2013,
Accepted 13th January 2014

DOI: 10.1039/c3mb70486f

www.rsc.org/molecularbiosystems

Protein–protein interactions are important for the majority of biological processes. A significant number of computational methods have been developed to predict protein–protein interactions using protein sequence, structural and genomic data. Vast experimental data is publicly available on the Internet, but it is scattered across numerous databases. This fact motivated us to create and evaluate new high-throughput datasets of interacting proteins. We extracted interaction data from DIP, MINT, BioGRID and IntAct databases. Then we constructed descriptive features for machine learning purposes based on data from Gene Ontology and DOMINE. Thereafter, four well-established machine learning methods: Support Vector Machine, Random Forest, Decision Tree and Naïve Bayes, were used on these datasets to build an Ensemble Learning method based on majority voting. In cross-validation experiment, sensitivity exceeded 80% and classification/prediction accuracy reached 90% for the Ensemble Learning method. We extended the experiment to a bigger and more realistic dataset maintaining sensitivity over 70%. These results confirmed that our datasets are suitable for performing PPI prediction and Ensemble Learning method is well suited for this task. Both the processed PPI datasets and the software are available at http://sysbio.icm.edu.pl/indra/EL-PPI/home.html.

1 Introduction

Protein–protein interactions (PPIs) occur when two or more proteins bind together, often enabling them to carry out their proper biological function. Many of the most important molecular processes in the cell such as DNA replication are carried out by large molecular machines that are built from a large number of protein components organized by protein–protein interactions.[1,2] However, while many interacting protein pairs have been identified through large-scale whole-genome experiments,[3–6] the residues involved in these interactions are generally not known and the vast majority of the interactions remain to be characterized structurally. The community of protein–protein interaction (PPI) researchers has been known for a wide and open distribution of proteomic data[7] *via* several PPI and pathway databases. While this ability to distribute and share data between various research groups has resulted in a large number of different source databases, the general overlap between these databases is very limited.[8,9] Hence, a common initial procedure for almost every researcher is to unify these diverse datasets to be able to pursue their own work.[10–13]

Usually the PPI source databases conduct in-house curation of data; this class includes both the pure-source databases and the databases combining in-house curation with meta-mining of other databases such as Drosophila Interactions Database (DroID),[14] Extracellular matrix interactions database (MatrixDB),[15] InnateDB (PPIs in the immune system),[16] Database of Interacting Proteins (DIP),[17] Molecular Interactions Database (MINT),[18] Biological General Repository for Interaction Datasets (BioGRID)[19] and IntAct.[20] Another class, pathway source databases, are databases that create networks, or pathways, and provide information not only regarding PPIs but also embed these interactions in a

[a] *Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw, Warsaw, Poland. E-mail: indra@icm.edu.pl, darman@icm.edu.pl*
[b] *Department of Computer Science and Engineering, Jadavpur University, Kolkata, West Bengal, India*
[c] *Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland*
[d] *Department of Animal Breeding and Genetics, SLU Global Bioinformatics Centre, Swedish University of Agricultural Sciences, Uppsala, Sweden*
[e] *Department of Clinical Sciences, Computational Genetics Section, Swedish University of Agricultural Sciences, Uppsala, Sweden*
[f] *Bioinformatics Program, Faculty of Technology and Natural Sciences, Uppsala University, Sweden*
[g] *The Jackson Laboratory for Genomic Medicine, c/o University of Connecticut Health Center, Administrative Services Building – Call Box 901, 263 Farmington Avenue, Farmington, CT 06030, USA*
[h] *Yale University, New Haven, CT, USA*

† Electronic supplementary information (ESI) available. See DOI: 10.1039/c3mb70486f

‡ These authors contributed equally to this work.

wider biological context. This category includes *e.g.* Kyoto Encyclopedia of Genes and Genomes (KEGG),[21] NetPath[22] and NCI-Nature Pathway Interaction Database (PDI).[23] Meta-mining databases are exclusively relying on source databases and high-throughput experiments published as finished datasets. This category includes Agile Protein Interaction DataAnalyzer (APID),[9] Michigan Molecular Interactions (MiMI),[24] Unified Human Interactome (UniHI),[8] Search Tool for the Retrieval of Interacting Genes (STRING)[25] and iRef Index.[26] The exchange of information among the databases is supported by three major data exchange formats: BioPAX,[27] PSI-MI[28] and SBML,[29] each with its own characteristics.[30]

In this article, we present high-throughout meta-mining datasets, for both yeast and human interactomes. Depending on the used selection criteria, for each organism we compiled two datasets: Gold and Silver. At the first stage of dataset compilation, we extracted information from the following databases: DIP,[17] MINT,[18] BioGrid,[19] and IntAct.[20] Our choice of these particular four databases has been motivated by the fact that they are the major providers of literature-curated protein–protein interactions.[31] We decided not to use available meta-mining databases such as iRef Index, because we have previously discovered that they are often outdated and contain only a subset of interactions from the original databases. After constructing interaction datasets, we added a number of additional attributes which are described in Section 3.1.

The datasets were constructed in a way that makes them immediately suitable for the application of machine learning methods. Our method of choice was an ensemble of different classifiers. Ensemble methods are gradually gaining popularity in the machine learning community and provide ways to increase stability and to overcome limitations of the individual classification algorithms.[32] Our goal was to combine individual classifiers by means majority voting, thus constructing a heterogeneous bagged ensemble.[33] Then we showed its usefulness through extensive experiments, both in smaller and larger settings. Detailed analysis of the results is presented in Section 4.

Our work differs from the previous research in the field of protein–protein interaction prediction in three aspects. First, we used a small set of high-level features describing directly the probability of interaction instead of protein domain composition or sequence. Second, we tried to improve the quality of the datasets used for training by merging and filtering the data from different databases. Third, rather than selecting any individual machine learning method we applied a heterogeneous ensemble technique.

2 Brief description of PPI databases and machine learning methods

2.1 PPI databases

The DIP[17] database catalogs experimentally determined interactions between proteins. It combines information from a variety of sources to create a single, consistent set of protein–protein interactions. The data stored within the DIP database were curated, both manually by expert curators and also automatically using computational approaches that use the knowledge about the protein–protein interaction networks extracted from the most reliable, core subset of the DIP data.§ It contains 76 823 protein interactions.

Currently, more then 95 000 physical interactions involving 27 461 proteins from 325 organisms are stored in MINT¶ database.[18] Although 90% of the stored interactions come from large-scale, genome-wide experiments, the value of MINT resides in the high number of curated articles. Each of the article reports a small number of interactions. This is reflected in the steady increase of curated articles that is mainly due to the curation of publications describing low-throughput experiments.

The BioGRID database‖[19] has been developed to house and distribute collections of protein and genetic interactions from all major model organism species. BioGRID currently contains over 340 000 interactions from the majority of model organism species. These interactions have been derived from both high-throughput studies and conventional focused studies. Through comprehensive curation efforts, BioGRID now includes a virtually complete set of interactions reported to-date in the primary literature for the budding yeast *S. cerevisiae* and the fission yeast *S. pombe*. Over time, several new features have been added to BioGRID including an improved user interface to display interactions based on different attributes, a mirror site and a dedicated interaction management system to coordinate curation across different locations. The BioGRID provides interaction data with monthly updates to Saccharomyces Genome Database, Flybase and Entrez Gene.

According to the data model of the PSI-MI standard, the basic curation unit in IntAct** is an experiment, defined by the set of interactions reported by Kerrien *et al.*[20] and derived using the same experimental technology. IntAct contains currently 281 793 distinct interactions. More detailed description of all four databases is given by Klingström and Plewczynski.[31]

2.2 Machine learning methods

Support Vector Machines (SVM) is a learning algorithm originally developed by Vapnik.[34] It has been extensively used for the purpose of classification in a wide variety of fields.[35] Support Vector Machines-based classifiers are inspired by statistical learning theory and they perform structural risk minimization on a nested set structure of separating hyperplanes.[36] Viewing the input data as two sets of vectors in a d-dimensional space, a SVM constructs a space-separating hyperplane which maximizes the margin between the two classes of points. To compute the margin, two parallel hyperplanes are constructed on each side of the separating one, which are ''pushed up'' against the two classes of points. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighboring data points of both classes. The larger margins (or distances) between these parallel hyperplanes indicate better generalization error of the classifier. In principle, the SVM classifier is designed for two-class problems. For linearly non-separable

§ http://dip.doe-mbi.ucla.edu/dip/Stat.cgi
¶ http://mint.bio.uniroma2.it/mint/Welcome.do
‖ http://www.thebiogrid.org
** http://www.ebi.ac.uk/intact/main.xhtml

problems, SVM transforms the input data into a high dimensional feature space and then employs a linear hyperplane for classification. Introduction of a feature space creates a computationally intractable problem. SVM handles this by defining appropriate kernels so that the problem can be solved in the input space itself. The problem of maximizing the margin can be reduced to the solution of a convex quadratic optimization problem, which has a unique global minimum. In this experiment, we have used the RBF (Radial Basis Function) kernel SVM. Here, the parameters of kernel function, γ, and the trade-off between training error and margin C are set to be 0.5 and 2.0, respectively.

Random Forest (RF)[37] is a statistical method that uses a combination of tree predictors. It ''grows'' many classification trees (in our case 100), each tree based on a subset of $m = \sqrt{\text{number of attributes}}$ attributes in this application. Maximal depth of individual tree was set to 3. Each tree is constructed on a bootstrap sample of the dataset. To classify a new object from an input vector, this algorithm applies the input vector to each tree of the forest. Each tree gives a classification, and the tree ''votes'' for that class. The forest chooses the classification having the most votes (over all of the trees in the forest). The forest error rate (OOB) depends on the correlation between any two trees in the forest (increasing the correlation increases the forest error rate) and the strength of each individual tree in the forest (a tree with a low error rate is a strong classifier, and increasing the strength of the individual trees decreases the forest error rate). Random forest can handle thousands of input variables without variable selection and gives estimates of variable importance the classification. Random forest generates an internal unbiased estimate of the generalization error as the forest-building progresses and has an efficient method for estimating missing data and maintains accuracy when a large proportion of data is missing. Random forest is, in general, resistant to overfitting and its execution time is reasonably short even for large datasets.

Decision trees (DT)[38] or recursive partitioning is a well-established machine learning method with the advantage of being easily understandable and transparent (easy to interpret). DTs extract interpretable classification rules as a path along the tree from the root to the leaf. DTs are employed to divide a large dataset into smaller and more homogeneous sets. The method identifies recursively the feature that created the most diverse subsets according to the criterion chosen. Simple DTs usually perform better than more complex ones and are easier to interpret, which is why several strategies have been developed for pruning DTs. A common strategy is post-pruning which overgrows the initial tree, eventually overfitting the training data and then prunes it back using cost complexity criteria. The pruning phase seeks a subtree that balances tree size and number of misclassifications or the mean square error on the training set. Other approaches for pruning include defining a minimum node size, a maximum tree-depth or stopping when partition significance falls below certain threshold. The strength of DTs lies in their capacity to handle very large and structurally diverse compound collections, to use large and heterogeneous descriptor sets, to ignore irrelevant descriptors and to generate a decision path for understanding the prediction of a test compound. The weakness of DTs is their relatively low prediction accuracy compared to other machine learning methods, and a number of extensions have been introduced to improve their prediction quality.

The Naïve Bayesian (NB)[39] algorithm is a simple classification method based on the Bayes rule for conditional probability. NB may be used in data analysis as a simple classifier between ''positive'' and ''negative'' training examples. It is guided by the frequency of occurrence of various features in a training set. NB classification is based on two core assumptions. First, the feature vectors in the training samples are equally important. Second, it assumes independence of the feature vectors from each other. Therefore, their combined probability is obtained by multiplying the individual probabilities. These two assumptions are often violated, and full independence of the feature vectors is rarely observed. However, NB is very robust to violations of these assumptions and tolerant toward noise. In this manuscript, conditional probabilities for discrete values were calculated using m-estimate with $m = 2$. Conditional probabilities for continuous values were estimated using LOESS[40] with window size 0.5 and 100 points sampled.

We used implementations of the listed algorithms available in Orange data-mining framework[41] and accessed through Python scripting interface.

3 Materials and methods

3.1 Preparation of Gold and Silver datasets

In order to create our high-throughput meta-mining PPI datasets, we used information extracted from DIP, MINT, BioGrid and IntAct. These four databases are the major providers of literature-curated protein–protein interactions.[31] Creating the final datasets involved several steps and procedures outlined in Fig. 1. We operated on the MySQL dumps of the original databases. All protein identifiers were mapped to Uniprot IDs. Each interaction was uniquely represented with a pair of protein IDs and Pubmed link. Following merging, all redundant interactions, having identical protein IDs and Pubmed link, were removed before we proceeded to construct the Gold and Silver datasets. The Gold datasets contain PPIs confirmed at least two times with two different experimental methods: yeast two hybrid (Y2H) and one of the affinity-based methods. Unlike the Gold ones, the Silver datasets contain PPIs that are confirmed more than once, but not necessarily with different experimental methods. Following from this definition, the Gold datasets are subsets of the Silver datasets. The Yeast Gold and Human Gold datasets contain 2117 and 1582 interacting protein pairs, respectively. In the same way, 14 677 and 27 419 interacting protein pairs are present in Silver dataset for respective species.

In addition to Gold and Silver, in further experiments, we used All Interactions dataset that contains human PPIs that have been confirmed using at least one experimental method. This dataset was created by our collaborators for the purpose of previous PPI research. It contains 57 576 PPIs for human and

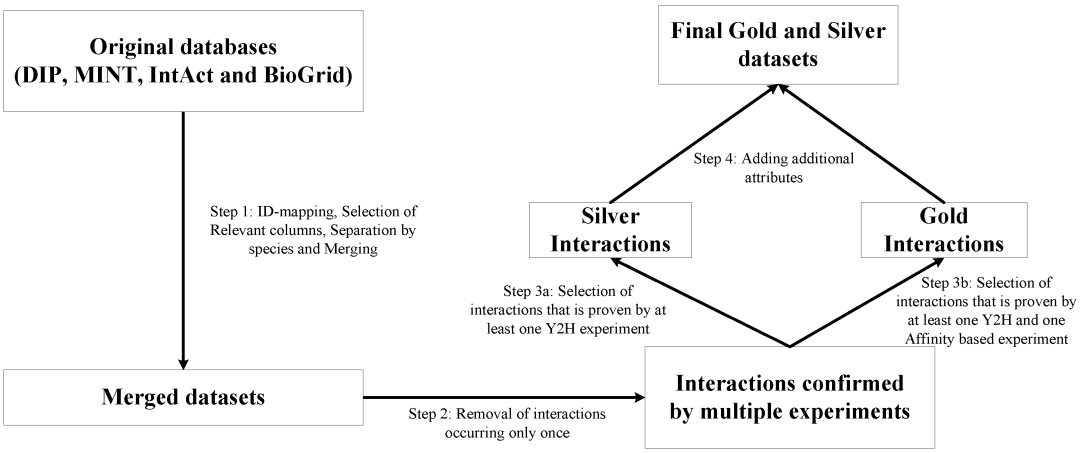

Fig. 1 Block diagram of different steps used in curation of Gold and Silver datasets.

190 377 for yeast. We expect it to be of much lower quality than Gold and Silver and contain a large portion of false positives. This is especially true in case of All Yeast: the number of interactions between yeast proteins is estimated to be around 26 000 by Grigoriev[42] and the size of All Yeast dataset exceeds it drastically.

Gold and Silver datasets reflect state of the source databases as of 27/08/2012. All dataset contains older data collected on 04/07/2011.

After compilation of Gold and Silver interactions for yeast and human, we collected additional attributes deemed plausible to affect the probability of an interaction between two proteins. The gene ontology (GO) annotation describes the cellular localization, function and process of a protein,[43] storing the information as an acyclic graph. These three GO graphs were collected for every protein. The amount of overlap between the graphs of the two interacting proteins was then calculated using the method implemented in the R-package GOSemSim.[44] Co-expression of two genes is an indication that the corresponding proteins tend to be synthesized at the same time and hence, simultaneously available in the cell. This might increase the probability of an interaction between the two proteins. Based on this assumption, we collected co-expression data for the interactions in our datasets, using the web resource MEM.[45]

We used DOMINE[46] to calculate the number of common interacting domains in a protein pair. Finally, we collected information on every PPI given by the Paralogous Verification Method (PVM), described by Deane *et al.*[47] This method deems an interaction probable if the putatively interacting pair have paralogs that are known to interact. The information was collected from DIP, which only stores PVM information for PPIs in yeast. Because of this, the attribute is only present in the Gold and Silver datasets for yeast. All the attributes are summarized in Table 1. We also created a dataset of negative examples by pairing protein entries, randomly selected from UniProtKB. These random pairs were then cross-checked against our PPI datasets to remove any true positives. All the additional attributes were added and the resulting dataset was used in the learning phase. Due to a large number of missing values, we did not use the co-expression attribute in machine learning phase. We also excluded all entries with missing GO terms. Table 2 presents the number of examples in each dataset. Note that due to filtering out examples with missing values, the number of instances in the original dataset and the number of instances used for machine learning differ. However, both the original datasets and the filtered data used for learning are available for download from the project website.

3.2 Ensemble learning

In the Ensemble Learning (EL), four well-established machine learning methods: SVM, RF, DT and NB, were used to produce different classification results. Each result is given equal importance

Table 1 Description of attributes included in the Gold and Silver datasets of Yeast and Human. The attributes used to train the classifiers are in bold

Attribute	Description
protA, protB	UniProt IDs of the two interacting proteins.
Y2H	Number of Y2H-studies supporting the interaction.
TAPMS mm	Number of affinity based experiments supporting the interaction.
Co-expression	Score indicating that the two proteins are co-expressed. The values are significance scores and do not correspond to the level of co-expression.
GO_cellComponent	Overlap between the GO cellular component graphs of the interacting proteins.
GO_function	Same as above but for GO cellular function.
GO_process	Same as above but for GO cellular process.
PVM	Binary attribute indicating whether the interaction is supported by DIPs Paralogous Verification (PVM) method.[a] Only present in the yeast datasets.
DomainDomain	Number of possibly interacting domains in the protein pair.
protA_seq, protB_seq	Amino acid sequences of the protein pair.

[a] http://dip.doe-mbi.ucla.edu/dip/Services.cgi?SM=2

Table 2 Number of positive instances in the original dataset (Unfiltered), positive instances actually used in machine learning (Positive), and generated negative instances (Negative) for Yeast and Human datasets

	Yeast			Human		
Dataset	Unfiltered	Positive	Negative	Unfiltered	Positive	Negative
Gold	2117	1503	1503	1582	1067	1067
Silver	14 677	7271	7271	27 419	12 704	12 704
All interactions	190 377	43 901	131 704	57 576	30 080	99 891

to yield better classification result. Hence, the majority voting is applied on the ensemble results. The steps of EL method are described below and the block diagram is shown in Fig. 2.

Step 1: train SVM, RF, DT and NB separately on the selected training dataset.

Step 2: predict class labels for the remaining dataset (test dataset) using the trained SVM, RF, DT and NB classifier separately.

Step 3: obtain label vector γ_j, where $j = 1, 2,\ldots,N$, where N is the number of machine learning methods, for test dataset.

Step 4: apply majority voting on all ensemble label vectors $\{\gamma_1, \gamma_2,\ldots,\gamma_N\}$, to obtain the final classification label vector.

3.3 Performance evaluation

Performance of each supervised ML classifier was measured by Accuracy ($\mathscr{A}$), Precision ($\mathscr{P}$), Recall ($\mathscr{R}$) and $\mathscr{F}$-measure ($\mathscr{F}_1$ score) values, together with confusion tables. Error estimates were calculated in a 10-fold cross-validation procedure using the following equations:

$$\mathscr{A} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \quad (1)$$

$$\mathscr{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (2)$$

$$\mathscr{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad (3)$$

$$\mathscr{F}_1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \quad (4)$$

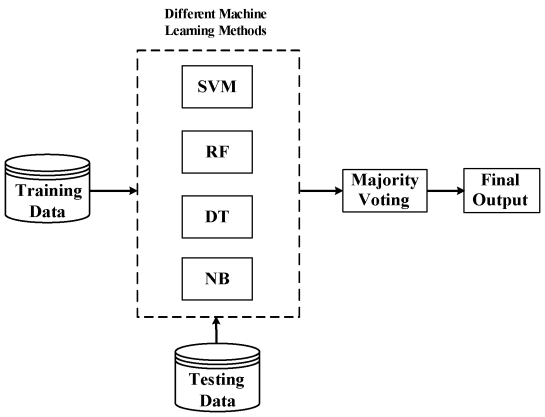

Fig. 2 Block diagram of Ensemble Learning.

where TP is the number of true positives, FP is the number of false positives, TN is the number of true negatives and FN is the number of false negatives. Classification accuracy $\mathscr{A}$ provides an overall accuracy measure, whereas precision $\mathscr{P}$ gives the percentage of observed positives that are correctly predicted (the measure of the reliability of positive instances prediction) and recall $\mathscr{R}$ measures the percentage of correct predictions (the probability of correct prediction). $\mathscr{F}_1$ score is a measure of test accuracy. $\mathscr{F}_1$ score takes values from 0 to 1 (from the worst to the best). In addition, we used Area Under the Receiver Operating Characteristic Curve (AUC)[48] to assess the quality of the constructed classifiers.

3.4 Feature importance evaluation

Having assessed predictive value of our datasets, we were also interested in predictive power of particular features and subsets of features. We performed the assessment using two widely-used feature selection algorithms as implemented in Weka:[49]

- CFSubsetEval – an algorithm proposed by Hall *et al.*[50] evaluates the predictive power of a subset of features by considering the individual predictive ability of each feature along with the degree of redundancy between them. We used Exhaustive search algorithm to evaluate the full space of feature subsets.
- InfoGainAttributeEval – an algorithm that evaluates the worth of an attribute by measuring the information gain with respect to the class.[49] We used the InfoGain evaluator in combination within the Ranker algorithm to assess the individual predictive power of each feature.

We performed both types of feature selection within a 5-fold cross-validation.

4 Discussion

Apart from creating and sharing reliable, carefully-curated datasets of interacting proteins, we were particularly interested in the performance of heterogeneous ensemble classifier in the context of the four constructed datasets. We evaluated both single classifiers built using four different, widely-used machine learning algorithms and our ensemble method. Accuracy ($\mathscr{A}$), precision ($\mathscr{P}$), recall ($\mathscr{R}$) and $\mathscr{F}_1$ score were computed to describe quality of the constructed classifiers.

Results are reported in Tables 3 and 4. We observed that the decision tree classifier performed significantly worse in comparison to other methods. Fig. 3–5 demonstrate box plot visualizations of recall, precision and AUC values for different ML methods, while Fig. 6 presents ROC curves. It transpired that on Silver Yeast dataset Random Forest have smaller recall

Table 3 Average values of performance measures of different methods for Yeast dataset

Dataset	Method	$\mathscr{A}$	AUC	$\mathscr{F}_1$	$\mathscr{P}$	$\mathscr{R}$
Gold	EL	0.91	0.97	0.91	0.94	0.89
	SVM	0.91	0.97	0.91	0.93	0.89
	RF	0.89	0.96	0.89	0.90	0.88
	DT	0.89	0.84	0.89	0.89	0.88
	NB	0.91	0.97	0.90	0.94	0.88
Silver	EL	0.80	0.87	0.78	0.84	0.73
	SVM	0.79	0.87	0.78	0.84	0.73
	RF	0.74	0.85	0.66	0.94	0.51
	DT	0.77	0.77	0.76	0.80	0.73
	NB	0.78	0.86	0.77	0.83	0.71

Table 4 Average values of performance measures of different methods for Human dataset

Dataset	Method	$\mathscr{A}$	AUC	$\mathscr{F}_1$	$\mathscr{P}$	$\mathscr{R}$
Gold	EL	0.90	0.95	0.90	0.89	0.91
	SVM	0.89	0.95	0.89	0.88	0.91
	RF	0.89	0.95	0.88	0.92	0.85
	DT	0.87	0.88	0.86	0.87	0.86
	NB	0.89	0.95	0.89	0.89	0.90
Silver	EL	0.81	0.88	0.81	0.81	0.81
	SVM	0.81	0.86	0.81	0.80	0.82
	RF	0.80	0.86	0.81	0.80	0.82
	DT	0.80	0.77	0.79	0.81	0.78
	NB	0.80	0.85	0.79	0.81	0.78

in comparison to other methods, but higher precision value. This effect can probably be compensated by adjusting classifier threshold. Ensemble learning classifier was not inferior to any of the individual classifiers. In terms of AUC it performed better than other classifiers on Silver Human dataset. Moreover, statistical significance of the results is shown using Friedman test[51,52] and mentioned in the ESI (http://sysbio.icm.edu.pl/indra/EL-PPI/supplementary.pdf).†

To better assess generalisation capability of the trained classifiers we evaluated them on All Human dataset, containing all interactions of human proteins listed in the source databases and random protein pairs as negatives. All Human dataset is more realistic than the smaller sets. It does not only contain more examples but also introduces class imbalance (there are three times more negatives than positives). Classifiers trained on Gold Human and Silver Human were tested separately.

Values of performance measures from this experiment are given in Tables 6 and 7. Evaluating classifiers on imbalanced data is usually more difficult. In particular, the default classification threshold of 0.5 is often suboptimal. Taking this into account, we adjusted thresholds of all classifiers to maximize their accuracy on the test set. As expected, performance metrics are worse than on smaller datasets. For All Human the results are still relatively good and suggest that the predictors are of practical value. Classifiers trained on Silver dataset performed slightly better than the ones trained on Gold dataset. Ensemble of Silver predictors, which achieved precision of 0.74 and recall of 0.73, turned out to be the best. For All Yeast, results are far

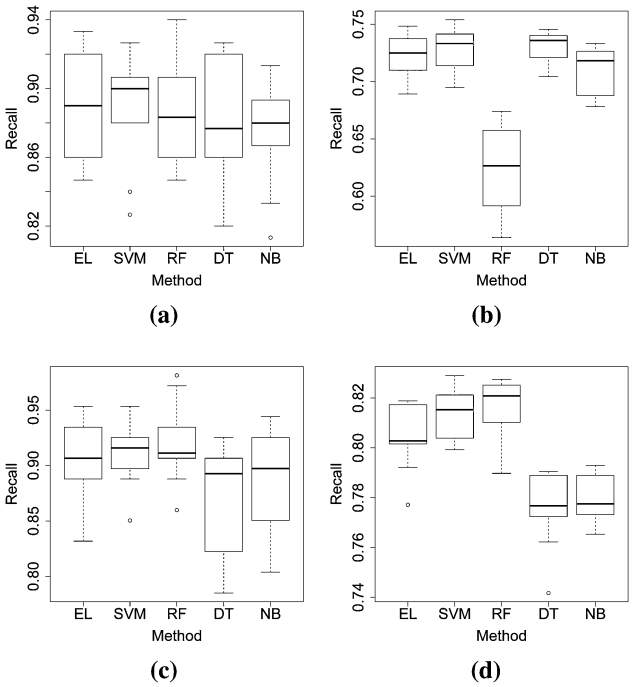

Fig. 3 Box plot of recall values ($\mathscr{R}$) of different methods for (a) Gold Yeast, (b) Silver Yeast, (c) Gold Human and (d) Silver Human.

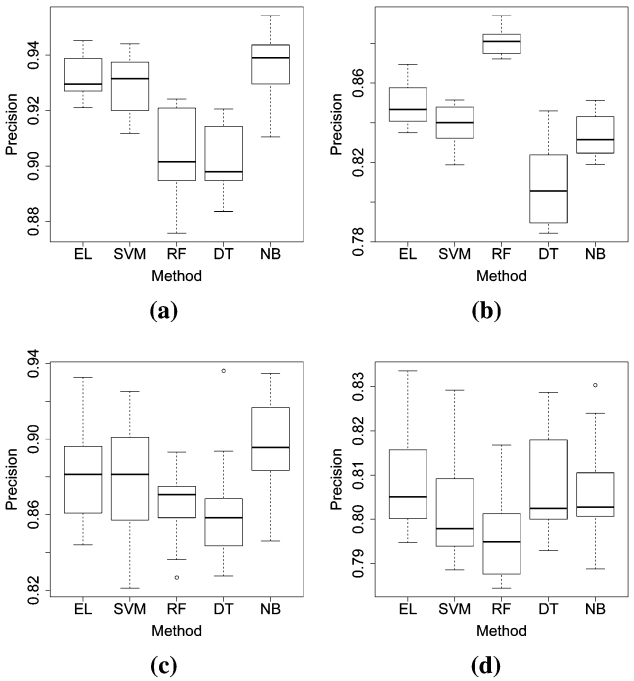

Fig. 4 Box plot of precision values ($\mathscr{R}$) of different methods for (a) Gold Yeast, (b) Silver Yeast, (c) Gold Human and (d) Silver Human.

worse, especially in the terms of recall. This means that predictions based on Gold or Silver subset does not generalize well to All dataset. When interpreting these results one should remember that some noise is expected in the dataset itself. All dataset is less reliable than Silver or Gold and may contain

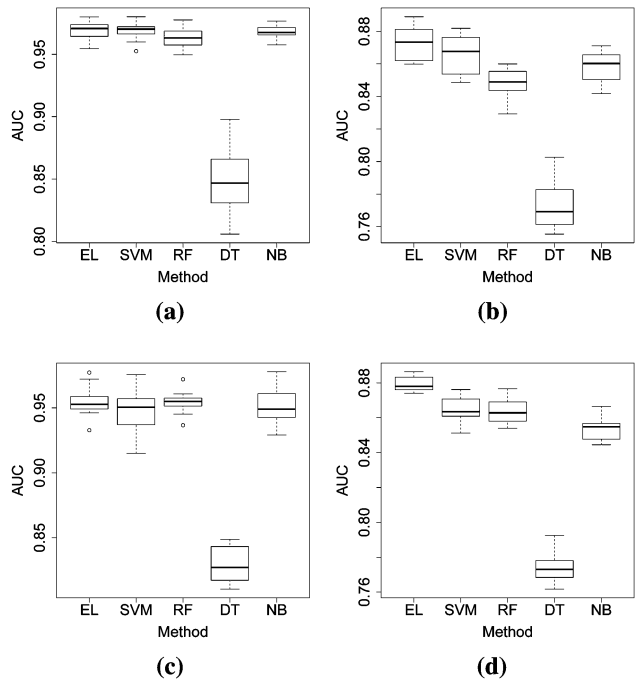

Fig. 5 Box plot of AUC values ($\mathscr{R}$) of different methods for (a) Gold Yeast, (b) Silver Yeast, (c) Gold Human and (d) Silver Human.

much more false positives. Also, among the negative examples there may be some number of interacting protein pairs that have not been discovered yet.

Some general conclusions can be drawn by looking at the distributions of scores returned by the predictors, as presented by Fig. 7. The EL classifier trained on Human Gold produced a very narrow distribution for the negative class: most of the examples received scores below 0.1. EL trained on Human Silver produced wider score distribution for the negative class but at the same time a narrower one for the positive class. When score values increase, shape of the negatives' distribution starts to resemble shape of the positives' distribution with a characteristic "bump" at the end. This means that the negatives contain more examples strongly resembling positive ones than vaguely resembling ones, which get scores on the decision boundary. This may be caused either by a significant number of false negatives in the dataset, *i.e.* undiscovered protein interactions or by a subset of true negatives with are indistinguishable from the positives using the features constructed. The first seems more plausible, although it cannot be confirmed at the moment. Since no analogous tendency can be seen in the positives' distribution, we can conclude that the number of false positives in All Human dataset is much smaller. There are practical consequences to it: when we want to make a more realistic dataset by introducing class imbalance and increasing the number of negatives, we are also introducing more noise. This problem should be addressed more thoroughly in further research.

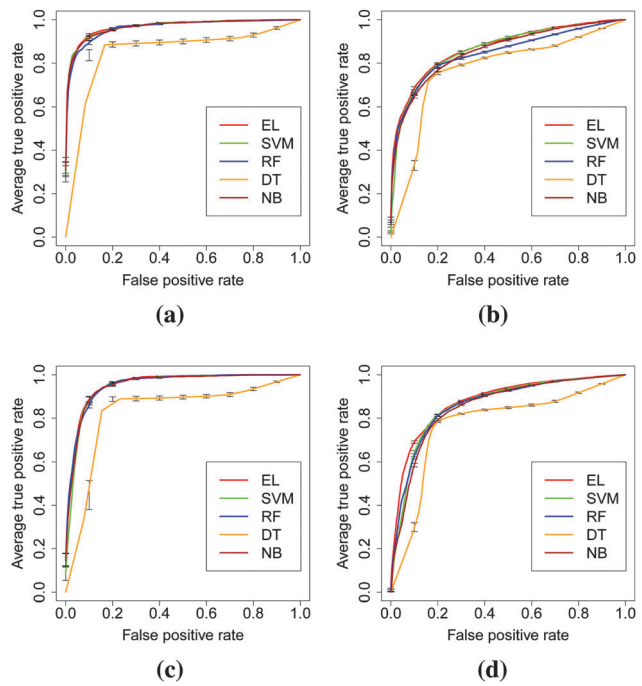

Fig. 6 ROC curves of different methods for (a) Gold Yeast, (b) Silver Yeast, (c) Gold Human and (d) Silver Human.

In case of All Yeast, distributions of scores for the positives and the negatives overlap and are difficult to separate. Both EL classifiers trained on Yeast Gold and on Yeast Silver yielded distributions where the maximum density for the positive class corresponds to the maximum density for the negative class. This could be caused by a very large number of false positives in All Yeast dataset. Such possibility is supported by the estimates by Deane *et al.*,[47] who claimed that the reliability of yeast interactions contained in DIP was only around 50% at the time of their study. Estimated size of yeast interactome (26 000)[42] compared with the size of the dataset (190 377) also supports this conclusion. This would mean that All Yeast is unreliable and should not be used in any further studies.

Class imbalance present in all datasets is still much lower than the expected imbalance in a real world application, *i.e.* reconstructing protein interaction network for a given organism. For example, for yeast proteome comprising around 6300 proteins there are $\binom{6300}{2} = 19\,841\,850$ possible protein pairs. The estimated total number of interactions is 26 000, so ratio of non-interacting to interacting pairs should be

$$\frac{19\,841\,850 - 26\,000}{26\,000} = 762.1481$$

Similarly, the human proteome consists of at least 22 500 proteins. Tran *et al.*[53] estimated that human interactome contains 210 000 interactions. Repeating the calculations above, the expected imbalance ratio would be

$$\frac{253\,113\,750 - 210\,000}{210\,000} = 1204.304$$

If we assumed that the distributions of scores for positive and negative class obtained on All Human datasets are valid, it would turn out that after introducing imbalance of that order of

Table 5 The results of feature importance evaluation

		InfoGain			CFSubsetEval	
Dataset	Attribute	Avg. rank	Avg. merit	SE	Times used	% used
HumanAll	GO function	1	0.358	0.002	5	100
	GO process	2	0.141	0.001	0	0
	GO cell component	3	0.13	0.002	0	0
	Domain–domain	4	0.067	0	5	100
HumanGold	GO function	1	0.498	0.007	5	100
	GO cell component	2	0.239	0.005	5	100
	Domain–domain	3	0.225	0.008	5	100
	GO process	4	0.198	0.008	5	100
HumanSilver	GO function	1	0.358	0.004	5	100
	GO process	2.4	0.134	0.012	0	0
	GO cell component	2.6	0.128	0.002	0	0
	Domain–domain	4	0.06	0.001	5	100
YeastAll	GO function	1	0.038	0	5	100
	Domain–domain	2	0.036	0	5	100
	GO process	3.4	0.025	0.001	0	0
	GO cell component	3.6	0.025	0	0	0
	PVM	5	0.013	0	5	100
YeastGold	GO function	1	0.53	0.007	5	100
	GO process	2	0.354	0.012	5	100
	GO cell component	3	0.299	0.004	5	100
	PVM	4	0.231	0.004	5	100
	Domain–domain	5	0.121	0.004	5	100
YeastSilver	GO function	1	0.301	0.003	5	100
	PVM	2	0.108	0.002	5	100
	GO process	3	0.064	0.004	0	0
	Domain–domain	4.2	0.05	0.001	5	100
	GO cell component	4.8	0.049	0.003	0	0

Table 6 Values of performance measures of different methods for Yeast All dataset

Train dataset	Method	T	$\mathscr{A}$	AUC	$\mathscr{F}_1$	$\mathscr{P}$	$\mathscr{R}$
Gold	EL	0.82	0.64	0.77	0.25	0.67	0.15
	SVM	0.71	0.65	0.77	0.28	0.66	0.18
	RF	0.70	0.68	0.76	0.21	0.65	0.12
	DT	1.00	0.69	0.81	0.37	1.00	0.23
	NB	0.98	0.63	0.77	0.20	0.70	0.12
Silver	EL	0.84	0.60	0.76	0.21	0.65	0.13
	SVM	0.84	0.60	0.76	0.21	0.64	0.13
	RF	0.52	0.67	0.77	0.29	0.62	0.19
	DT	1.00	0.65	0.81	0.40	1.00	0.25
	NB	1.00	0.61	0.76	0.18	0.69	0.10

Table 7 Values of performance measures of different methods for Human All dataset

Train dataset	Method	T	$\mathscr{A}$	AUC	$\mathscr{F}_1$	$\mathscr{P}$	$\mathscr{R}$
Gold	EL	0.68	0.90	0.85	0.66	0.71	0.61
	SVM	0.42	0.89	0.84	0.68	0.63	0.75
	RF	0.67	0.90	0.85	0.70	0.67	0.73
	DT	1.00	0.80	0.86	0.56	1.00	0.39
	NB	0.61	0.88	0.84	0.64	0.66	0.62
Silver	EL	0.75	0.92	0.88	0.73	0.74	0.73
	SVM	0.71	0.90	0.86	0.71	0.69	0.74
	RF	0.72	0.91	0.87	0.70	0.73	0.67
	DT	1.00	0.84	0.86	0.56	1.00	0.39
	NB	0.70	0.90	0.85	0.70	0.67	0.75

magnitude our method trained on Silver Human would have precision below 10% even for threshold values very close to 1. In reality, we expect those distributions to be biased because of the occurrence of false positives/negatives in the datasets and real-world precision should be higher. However, we are aware that our method is probably not ready yet for large-scale network reconstruction. It can be still applied in more limited scenarios, for example to validate and clean up an unreliable PPI database. We are not aware of any other method which overcomes the described problems.

An interesting extension of this work would be to apply the same methodology for the reconstruction of other biological networks. One possibility could be to, instead of defining every protein pair as interacting or non-interacting, to define it in terms of network connectedness. Given a training data set, all pairs above a certain connectedness threshold would be defined as positive and the others as negative. A classifier trained in this way would predict, not physical protein interactions, but proteins with a certain degree of connectedness within the whole network. Thus, it might be able to give biological insights on a higher structural level.

4.1 Comparison with existing methods

It is difficult to compare obtained results with other PPI prediction studies. The properties of the dataset used, such as its coverage and reliability, are different in every case,

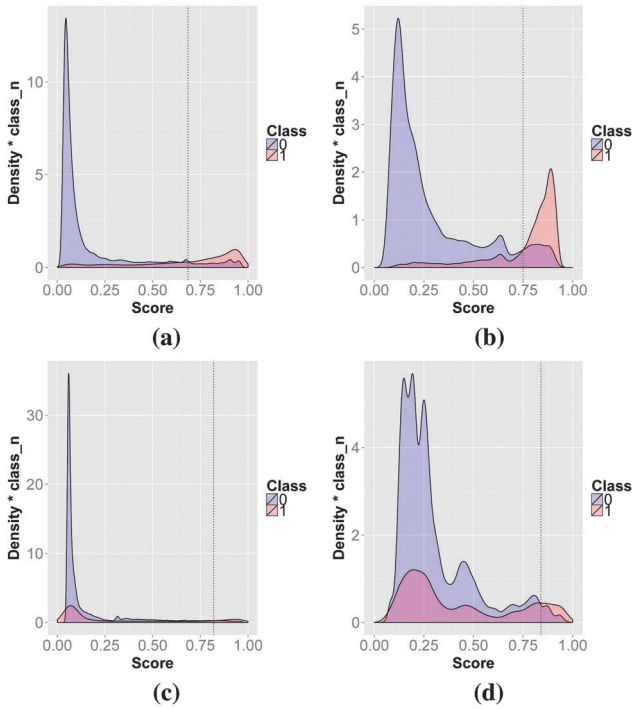

Fig. 7 Distributions of scores returned by the predictor tested on All Human and All Yeast. Predictors: (a) EL trained on Gold Human, tested on All Human, (b) EL trained on Silver Human, tested on All Human, (c) EL trained on Gold Yeast, tested on All Yeast, (d) EL trained on Silver Yeast, tested on All Yeast. Vertical lines mark adjusted threshold values. Value on Y-axis is the estimated density multiplied by $k = \frac{N_c}{10000}$, where N_c is the number of examples in the respective class.

therefore performance scores me may not be directly comparable. Using the same dataset to run a couple of methods would also pose some problems, since many methods, including ours, require special features available only for a subset of known proteins. Taking this into account, a cautious comparison with published results can still be made. PPI_SVM[54] method, which used domain information to predict PPI from different organisms, obtained 76% recall and 95% precision on a reliable subset of DIP database. PreSPI,[55] another domain-based method evaluated on DIP, achieved very similar values: 95% precision and 77% recall. Liu[56] used sequence-based method predicting interactions for yeast proteins from DIP and reported 87% precision and 90% recall. On Gold dataset, our EL method obtained 94% precision, 89% recall for human and 89% precision, 91% recall for yeast proteins. These values are comparable with the top results reported in the literature. However, the tests on bigger datasets yielded much lower performance scores. We believe that the setting of these tests was more realistic but the results are heavily biased by poor quality of the datasets. Analogous experiments were not performed in the majority of the reported studies.

4.2 Importance of features

Besides assessing the predictive value of the assembled datasets, we were interested in the value of particular features. Feature importance can be measured using several different criteria and one may be interested in either the importance of an individual feature or in the predictive value of a subset of features. The latter is more informative in the case of interacting features which may have low value when considered individually, but be good predictors when considered together. On the other end of the scale, there are pairs of highly correlated features that can be used interchangeably without affecting the quality of classification.[57] Here, we were interested in both types of feature importance. Hence, we evaluated features using two different algorithms: one measuring individual feature importance and one measuring the value of an ensemble of features. The results are presented in Table 5. Both GO_function and DomainDomain features transpired to be highly informative and they are also included in all best subsets of features. Interestingly enough, in some cases (*e.g.*, HumanGold) all the features were selected by feature subset evaluator which may indicate interactions between those features. Similarly, in the YeastAll dataset, the PVM feature has low importance when considered individually yet it is always included in the best subset of features. The above findings clearly show the complexity of classification tools required for the accurate prediction of protein–protein interactions given the set of features used in this work. Our future work will focus on better understanding the observed feature importance patterns and on unraveling the nature of potential feature interactions.

5 Conclusion

In this article, we have constructed four high-throughput meta-mining protein–protein interaction datasets for yeast and human. For this purpose, four major literature curated protein–protein interactions databases DIP, MINT, BioGrid and IntAct have been used. A number of attributes was then collected for every interaction. The four resulting datasets are called: Gold Yeast, Silver Yeast, Gold Human and Silver Human. Thereafter, different machine learning methods have been used to build a heterogeneous ensemble classifier for PPI prediction. The ensemble classifier performed very well in cross-validation experiments. Additional experiments with bigger All Human dataset demonstrated applicability of the trained classifiers in a more realistic setting. Here the advantage of ensemble learning over individual classifiers was even more visible. Classifiers evaluated on All Yeast performed poorly, which can be possibly attributed to the poor quality of the test set.

Our future research will aim at extending this experiment for the local sequence of interacting proteins by using our current datasets. The authors are currently working in this direction. The detailed analysis of the results suggested issue with false negatives in the training data. We are also closely looking at possible solutions to this issue.

Funding

This work was supported by the Polish Ministry of Education and Science (N301 159735, NCN 2013/09/B/NZ2/00121 and others). IS was supported by University with Potential for

Excellence (UPE) – Phase II project grant from University Grants Commission (UGC) in India. MK was supported by Swedish Foundation for Strategic Research. JZ was financed by research fellowship within Project ''Information technologies: Research and their interdisciplinary applications'', agreement number UDA-POKL.04.01.01-00-051/10-00.

Acknowledgements

Computations were performed at the Interdisciplinary Centre for Mathematical and Computational Modelling (ICM) at Warsaw University. We would like to thank Jesper Gådin, Andreas E. Lundberg, Niklas Malmqvist, Lidaw Pello-Esso and Emil Marklund, Anton Berglund, Susanna Trollvad, David Majlund for valuable help and input during data preparation. We would also like to thank Uppsala Science Association and the anonymous reviewers for their valuable comments to improve the quality of this manuscript.

References

1 T. E. Creighton, *Proteins: Structures and Molecular Properties*, W. H. Freeman, NY, 2nd edn, 1996.
2 J. Kyte, *Structure in Protein Chemistry*, Garland Publishing Inc., NY, 2nd edn, 1995.
3 A. Gavin, M. Bösche and R. Krause, *et al.*, *Nature*, 2002, **415**, 141–147.
4 L. Giot, J. S. Bader and C. Brouwer, *et al.*, *Science*, 2003, **302**, 1727–1736.
5 S. Li, C. M. Armstrong and N. Bertin, *et al.*, *Science*, 2004, **303**, 540–543.
6 P. Uetz, L. Giot and G. Cagney, *et al.*, *Science*, 2000, **403**, 623–627.
7 S. Orchard, H. Hermjakob and R. Apweiler, *Proteomics*, 2003, **3**, 1374–1376.
8 G. Chaurasia, Y. Iqbal and C. Hanig, *et al.*, *Nucleic Acids Res.*, 2007, **35**, D590–D594.
9 C. Prieto and J. D. L. Rivas, *Nucleic Acids Res.*, 2006, **34**, W298–W302.
10 M. A. Mahdavi and Y. H. Lin, *Genomics, Proteomics Bioinf.*, 2007, **5**, 177–186.
11 D. R. Rhodes, S. A. Tomlins and S. Varambally, *et al.*, *Nat. Biotechnol.*, 2005, **23**, 951–959.
12 R. Sharan and T. Ideker, *Nat. Biotechnol.*, 2006, **24**, 427–433.
13 N. Yosef, M. Kupiec and E. Ruppin, *et al.*, *Nucleic Acids Res.*, 2009, **37**, e88.
14 J. Yu, S. Pacifico and G. Liu, *et al.*, *BMC Genomics*, 2008, **9**, 461.
15 E. Chautard, L. Ballut and N. Thierry-Mieg, *et al.*, *Bioinformatics*, 2009, **25**, 690–691.
16 M. Korb, A. G. Rust and V. Thorsson, *et al.*, *BMC Immunol.*, 2008, **9**, 7.
17 L. Salwinski, C. S. Miller and A. J. Smith, *et al.*, *Nucleic Acids Res.*, 2004, **32**, D449–D451.
18 A. Chatr-aryamontri, A. Ceol and L. M. Palazzi, *et al.*, *Nucleic Acids Res.*, 2007, **35**, D572–D574.
19 B. J. Breitkreutz, C. Stark and T. Reguly, *et al.*, *Nucleic Acids Res.*, 2008, **36**, D637–D640.
20 S. Kerrien, Y. Alam-Faruque and B. Aranda, *et al.*, *Nucleic Acids Res.*, 2007, **35**, D561–D565.
21 M. Kanehisa, M. Araki and S. Goto, *et al.*, *Nucleic Acids Res.*, 2008, **36**, D480–D484.
22 T. S. Prasad, K. Kandasamy and A. Pandey, *Methods Mol. Biol.*, 2009, **577**, 67–79.
23 C. F. Schaefer, K. Anthony and S. Krupa, *et al.*, *Nucleic Acids Res.*, 2009, **37**, D674–D679.
24 M. Jayapandian, A. Chapman and V. G. Tarcea, *et al.*, *Nucleic Acids Res.*, 2007, **35**, D566–D571.
25 A. Franceschini, D. Szklarczyk, S. Frankild, M. Kuhn, M. Simonovic, A. Roth, J. Lin, P. Minguez, P. Bork, C. von Mering and L. J. Jensen, *Nucleic Acids Res.*, 2013, **41**, 808–815.
26 S. Razick, G. Magklaras and I. M. Donaldson, *BMC Bioinf.*, 2008, **9**, 405.
27 w. BioPAX, *BioPAX - Biological Pathways Exchange Language*, Biopax.org technical report, 2005 (19 January 2010, date last accessed).
28 S. Kerrien, S. Orchard and L. Montecchi-Palazzi, *et al.*, *BMC Biol.*, 2007, **5**, 44.
29 M. Hucka, A. Finney and H. M. Sauro, *et al.*, *Bioinformatics*, 2003, **19**, 524–531.
30 L. Strömbäck and P. Lambrix, *Bioinformatics*, 2005, **21**, 4401–4407.
31 T. Klingström and D. Plewczynski, *Briefings Bioinf.*, 2010, **12**, 702–713.
32 R. Polikar, *IEEE Circuits Syst. Mag.*, 2006, **6**, 21–45.
33 Q.-L. Zhao, Y.-H. Jiang and M. Xu, ADMA (2), 2010, pp. 1–12.
34 V. Vapnik, *The nature of statistical learning theory*, Springer-Verlag, Berlin, Germany, 1995.
35 R. Collobert and S. Bengio, *J. Mach. Learn. Res.*, 2001, **1**, 143–160.
36 V. Vapnik, *Statistical Learning Theory*, Wiley, New York, USA, 1998.
37 L. Breiman, *Mach. Learn.*, 2001, **45**, 5–32.
38 Y. Yuan and M. J. Shaw, *Fuzzy Sets and Systems*, 1995, **69**, 125–139.
39 H. George and J. P. Langley, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995, vol. 69, pp. 338–345.
40 W. S. Cleveland, *J. Am. Stat. Assoc.*, 1979, **74**, 829–836.
41 J. Demšar, T. Curk, A. Erjavec, Črt Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik and B. Zupan, *J. Mach. Learn. Res.*, 2013, **14**, 2349–2353.
42 A. Grigoriev, *Nucleic Acids Res.*, 2003, **31**, 4157–4161.
43 T. G. O. Consortium, *Nat. Genet.*, 2000, **25**, 25–29.
44 G. Yu, F. Li, Y. Qin, X. Bo, Y. Wu and S. Wang, *Bioinformatics*, 2010, **26**, 976–978.

45 P. Adler, R. Kolde, M. Kull, A. Tkachenko, H. Peterson, J. Reimand and J. Vilo, *Genome Biol.*, 2009, **10**, R139.
46 S. Yellaboina, A. Tasneem, D. V. Zaykin, B. Raghavachari and R. Jothi, *Nucleic Acids Res.*, 2011, **39**, D730–D735.
47 C. M. Deane, *Mol. Cell. Proteomics*, 2002, **1**, 349–356.
48 F. Provost and T. Fawcett, *Mach. Learn.*, 2001, **44**, 203–231.
49 M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, *SIGKDD Explorations*, 2009, **11**, 10–18.
50 M. A. Hall, PhD thesis, University of Waikato, Hamilton, New Zealand, 1998.
51 M. Friedman, *J. Am. Stat. Assoc.*, 1937, **32**, 675–701.
52 M. Friedman, *Ann. Math. Stat.*, 1940, **11**, 86–92.
53 N. H. Tran, K. P. Choi and L. Zhang, *Nat. Commun.*, 2013, **4**, 2241.
54 P. Chatterjee, S. Basu, M. Kundu, M. Nasipuri and D. Plewczynski, *Cell. Mol. Biol. Lett.*, 2011, **16**, 264–278.
55 D. S. Han, H. S. Kim, W. H. Jang, S. D. Lee and J. K. Suh, *Nucleic Acids Res.*, 2004, **32**, 6312–6320.
56 H.-W. Liu, *Optimization and Systems Biology*, 2009, pp. 198–206.
57 M. Kierczak, PhD thesis, Uppsala University, Uppsala, Sweden, 2009.

