

PROCEEDINGS

Open Access

# How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis

Mauno Vihinen<sup>1,2,3</sup>

From SNP-SIG 2011: Identification and annotation of SNPs in the context of structure, function and disease Vienna, Austria. 15 July 2011

## Abstract

**Background:** Prediction methods are increasingly used in biosciences to forecast diverse features and characteristics. Binary two-state classifiers are the most common applications. They are usually based on machine learning approaches. For the end user it is often problematic to evaluate the true performance and applicability of computational tools as some knowledge about computer science and statistics would be needed.

**Results:** Instructions are given on how to interpret and compare method evaluation results. For systematic method performance analysis is needed established benchmark datasets which contain cases with known outcome, and suitable evaluation measures. The criteria for benchmark datasets are discussed along with their implementation in VariBench, benchmark database for variations. There is no single measure that alone could describe all the aspects of method performance. Predictions of genetic variation effects on DNA, RNA and protein level are important as information about variants can be produced much faster than their disease relevance can be experimentally verified. Therefore numerous prediction tools have been developed, however, systematic analyses of their performance and comparison have just started to emerge.

**Conclusions:** The end users of prediction tools should be able to understand how evaluation is done and how to interpret the results. Six main performance evaluation measures are introduced. These include sensitivity, specificity, positive predictive value, negative predictive value, accuracy and Matthews correlation coefficient. Together with receiver operating characteristics (ROC) analysis they provide a good picture about the performance of methods and allow their objective and quantitative comparison. A checklist of items to look at is provided. Comparisons of methods for missense variant tolerance, protein stability changes due to amino acid substitutions, and effects of variations on mRNA splicing are presented.

## Background

Gene and genome sequencing speed is ever increasing and thus lots of genetic variation information is available. The technological development of sequencing methods has led to a situation where the interpretation of the generated data is a severe bottleneck for the use of genetic information. Numerous prediction methods have been developed during the last decade to address the relevance of gene and protein variants to pathogenicity. General

tolerance methods predict whether the variants are disease-related or not (or affect protein function or not), and specific methods are used to address variation effect mechanisms [1]. These methods can be useful. However, until recently their true applicability and performance have not been studied systematically [2-5]. When methods are originally published, authors provide some performance measures, which are usually not comparable with other methods due to the use of different training and test datasets, different reported measures etc. The scope of this article is to discuss how the assessment of method performance should be done and interpretation

Correspondence: [mauno.vihinen@med.tu.se](mailto:mauno.vihinen@med.tu.se)

<sup>1</sup>Institute of Biomedical Technology, FI-33014 University of Tampere, Finland  
Full list of author information is available at the end of the article

of the results and the choice of the best methods. The text is mainly intended for scientists who are users of predictors without training in statistics or computer science. Method developers are taken into account by providing a checklist of items to be reported with methods. The examples discussed are related to prediction of variant effects, but description of methods and evaluation measures is general and thereby not application domain specific.

### Method testing schemes

Three approaches can be used for testing method performance and can be classified according to increasing reliability (Fig. 1).

Challenges aim to test whether certain problems can be addressed with existing tools and to find out what kind of methods will be needed in the future. Critical Assessment of Structure Predictions (CASP) [6] was the first challenge of this kind in biosciences. The idea was, and still is, even when CASP has been running for 20 years, to test how prediction methods behave on different protein structure related tasks. The method developers apply their systems without knowing the correct result (blind test), which however is available for the challenge assessors. This setup allows independent testing of method performance. In a similar vein, other critical assessment challenges have been organized e.g. for Critical Assessment of protein Function Annotation (CAFA) [7] and Critical Assessment of PRediction of Interactions (CAPRI) [8].

CAGI, Critical Assessment of Genome Interpretation (<http://genomeinterpretation.org/>), is a challenge for method developers in the field of phenotypic impacts of genomic variation. The second CAGI prediction season was organized during fall 2011. These challenges do not aim for systematic analysis of predictions, instead they assess what is currently doable, providing proof of concept, charting where to direct future efforts, and identifying new areas where predictive approaches would be needed.

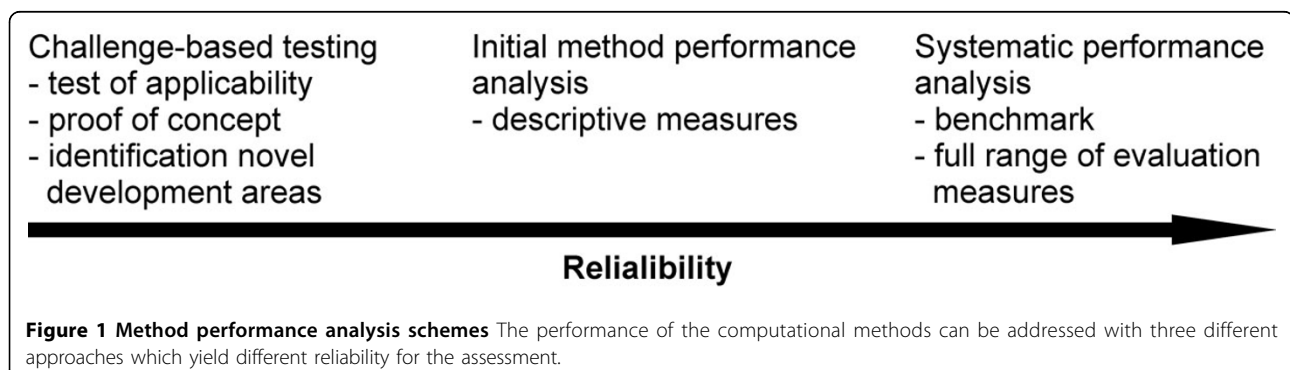
The second test strategy is typically used by method developers to test their approaches. These are usually done with developer collected test sets (especially when benchmark datasets are lacking) and report certain performance parameters. Most often the testing is not comprehensive, and the results are incomparable with those obtained from other methods e.g. due to using different test sets. Sometimes evaluation parameters are selectively presented which leads to problems in determining the true merits and pitfalls of methods.

The third approach, systematic analysis, uses approved and widely accepted benchmark dataset(s) and suitable evaluation measures to explain method performance. It is hoped that in the future the variation effect program developers would use benchmark test sets and comparable measures. This is already the general practice e.g. in the multiple sequence alignment (MSA) field.

### Prediction methods for classification

A plethora of pattern recognition methods have been applied to problems in bioinformatics including rule based, statistical methods and machine learning -based methodologies. The goal of machine learning is to train a computer system to distinguish i.e. classify cases based on known examples. Machine learning methods include several widely differing approaches such as support vector machines, neural networks, Bayesian classifiers, random forests and decision trees.

In the following discussion we concentrate on machine learning methods as they are nowadays widely used to tackle complex phenomena, which would be otherwise difficult to handle. Successful machine learning method development requires good quality training set. The dataset should represent the space of possible cases. This space is huge for genetic variations as they can have so many different effects and underlying mechanisms. Another aspect is the choice of the machine learning approach. There is not a superior architecture among them. Third, the quality of the predictor depends on how the training has been done, which features are used to



explain the phenomenon and optimization of the method.

Fig. 2 depicts the principle underlying machine learning in a two-class classification task. The predictor is trained with known positive and negative instances in an approach called supervised learning. This leads to reorganization of the system, details of which differ according to the architecture employed. Once the method has learned to distinguish between the cases it can be applied to predict the class of unknown cases. The predictors can be classified as discrete or probabilistic depending on whether they provide a score, not necessarily a p value, for predictions. In the case of methods with discrete output, more or less *ad hoc* thresholds have been used to detect the most reliable events. Many machine learning based predictors are binary classifiers, however, it is possible to have more than two outputs e.g. by using multi-tier two-class prediction system.

Features describe the characteristics of the investigated phenomenon. If several features are available it is important to choose those, which best capture the phenomenon. This is partly due to the curse of dimensionality, which means that much more data are needed when the number of features increases. The volume of the feature space grows exponentially with the dimensionality such that the data become sparse and insufficient to adequately describe the pattern in the feature space. Another problem is overfitting, which means that the learner, due to sparse data, complex model or excessive learning procedure, describes noise or random features in the training dataset, instead of the real phenomenon. It is crucial to avoid overfitting as it leads to decreased performance on real cases.

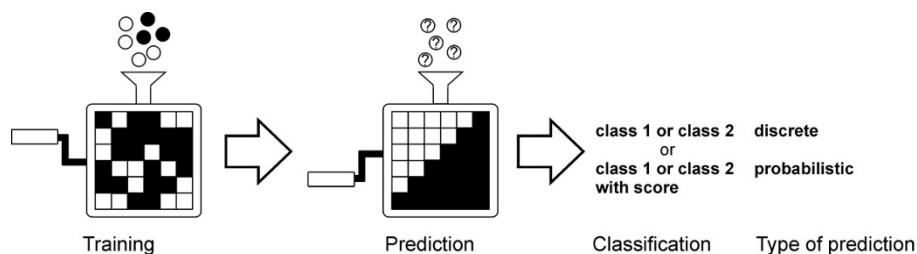
Many predictors provide a measure for the probability of prediction, in this domain a measure of how likely the variation is pathogenic. This information can be used for ranking the investigated cases. A more advanced version is to obtain e.g. by bootstrapping an estimate of the standard error of the prediction indicative of the prediction reliability.

Many types of biological data are limited in quantity. The same data cannot be used both for method training and testing. The trick is to partition the dataset. This can be done in different ways, with cross-validation probably being the most popular of these. The dataset is divided into  $k$  disjoint partitions, one of which is used for testing and the others for training. This is repeated  $k$  times until all the partitions have been used as test set. Ten partitions i.e. ten times cross validation is the most common partitioning scheme. The average performance measures computed from the splits are used to describe the overall prediction performance. Random sampling is another approach, however, a problem is that the same cases may appear more than once in the test set and others not at all. Another computationally intensive validation approach is leave one out validation, an extreme case of cross validation with partitioning to the total number of instances. As the name implies, one case at time is left for validation while the remaining cases are used for training. The computational requirements may be prohibitive with large datasets. A problem especially for the last scheme is if there are some very similar cases in the dataset.

Typically the training set should contain about equal amount of cases in each class. Imbalance in the numbers of cases in the classes can cause problems during performance evaluation as discussed below. There are some ways to handle class imbalance.

### Principles of method evaluation

To test and compare predictors two requirements have to be met. There has to be available test dataset with known outcome and there has to be in place suitable prediction performance evaluation measures. Benchmark is a gold standard dataset - cases with experimentally validated known effects which represent the real world. These can be used for training machine learning methods as well as for testing the developed methods. The same



**Figure 2 Principles of machine learning** Machine learning is a form of supervised learning in which a computer system learns from given positive and negative instances to distinguish between cases belonging to the two classes. During training, positive and negative cases (black and white balls) are provided for the system, which leads to organization of the predictor (indicated by the arrangement of the black and white squares inside the predictor) such that it learns to separate the cases and thus can classify unknown cases (balls with question marks). Depending on the classifier, whether it yields in addition to the classification also a score for the prediction, the results can be called as discrete or probabilistic.

data should however never be used for training and testing as that would only indicate the capacity of the method to memorize examples, not its generalization potential – how well it performs on instances outside the training set. High quality benchmark datasets require meticulous data collection often from diverse sources and careful checking of the correctness of the data.

Numerous measures have been developed to describe predictor performance, but no single measure captures all aspects of predictor performance. The measures mainly used, and how to interpret them will be discussed. Typically prediction methods are used as classifiers to define whether a case has the investigated feature or not. Results of this kind of binary predictor can be presented in a 2x2 confusion table also called contingency table or matrix. This, at first glance may appear simple to interpret, but the contrary is the case, as various composite aspects have to be jointly taken into account.

**Benchmark criteria**

Benchmark can be defined as a standard or reference for evaluation, in this case prediction method performance. Benchmarks are widely used in computer science and technology. For example computer processor performance is tested with standardized benchmark methods. In bioinformatics there are benchmarks e.g. for multiple sequence alignment methods already 1990's [9]. Novel MSA construction methods are routinely tested with alignment

benchmarks such as BALiBASE [10] HOMSTRAD [11], OxBench suite [12], PREFAB [13], and SABmark [14]. Other bioinformatic benchmarks include protein 3D structure prediction [15-17], protein structure and function prediction [18], protein-protein docking [19] and gene expression analysis [20,21] benchmarks etc.

Benchmark usage varies between different communities. For variation effect predictions, benchmarks have not been available and thus authors have used different datasets. The situation has changed only recently with the release of VariBench (<http://bioinf.uta.fi/VariBench/>) (Nair and Vihinen, submitted).

To be useful a benchmark should fulfill certain criteria. These criteria vary somewhat between the domains, but there are also some common features (Fig. 3). The criteria laid by Gray originally for database systems and transaction processing systems are still valid [22]. Criteria for MSA [23] and variation data (Nair and Vihinen, submitted) benchmarks have been defined. These include relevance, which means that the data have to capture the characteristics of the problem domain. Portability allows testing of different systems. Scaleability of the benchmark allows testing systems of different sizes, and simplicity means that the benchmark has to be understandable and thereby credible. Accessibility means that the benchmark has to be publicly available, solvability to set the level of the task on suitable level (not too difficult, not too hard), independence to guarantee that the benchmark has not

<b>Benchmark criteria</b>		
<u>Gray</u>	<u>Aniba et al.</u>	<u>VariBench</u>
relevant	relevance	relevance
portable	solvability	representativeness
scaleable	scalability	non-redundancy
simple	accessibility	experimentally determined cases
	independence	positive and negative cases
	evolution	scalability
		reusability

**Figure 3 Benchmark criteria** Criteria for benchmarks in three different studies. VariBench is the database specifically designed for variation benchmark datasets.

been developed with tools to be tested, and evolution to keep the benchmark up-to-date during time.

When considering the variation benchmarks, datasets should be large enough to cover variations related to a certain feature or mechanism. For example in the case of missense variations this means very large numbers of instances as there are altogether 150 single nucleotide changes which cause amino acid substitution. To have statistical power several cases are needed. The required numbers of cases increase exponentially as features are combined. Datasets have to be non-redundant and devoid of similar or greatly overlapping entries. This criterion relates to independence requirement of [23]. Datasets have to contain both positive (showing the investigated feature) and negative (not having effect) cases so that the capability of methods to distinguish effects can be tested. This may cause problems in data collection as some phenomena are very rare and only a few known cases may exist.

VariBench is a database for variation-related benchmark datasets that can be used for developing, optimizing, comparing and evaluating the performance of computational tools that predict the effects of variations (Nair and Vihinen, submitted). VariBench datasets provide multilevel mapping of the variation position to DNA, RNA and protein as well as to protein structure entries in PDB [24] (when possible). Method developers are requested to submit their datasets to VariBench to be distributed to the community.

VariBench datasets have been collected from literature as well as with data mining approaches from diverse sources. Locus specific databases (LSDBs) are the most reliable source for disease-related data. Although lots of variation data are listed in LSDBs, it would be necessary to capture to databases all the cases from clinical and research laboratories [25,26].

An integral part of databases is the annotation of the entries. For variation information collection it would be extremely important to describe the cases in a systematic and unambiguous way.

Variation Ontology (VariO, <http://variationontology.org/>) has been developed for systematic description and annotation of variation effects and consequences on DNA, RNA and/or protein including variation type, structure, function, interactions, properties and other features (Vihinen, in preparation). VariO annotated data would allow easy collection of novel dedicated benchmarks.

### Evaluation measures

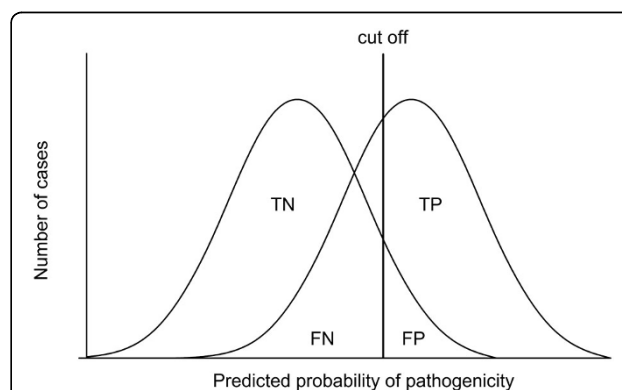
The outcome of binary (pathogenic/benign) style predictors are often presented in a 2x2 contingency table (Fig. 4). The number of correctly predicted pathogenic (non-functional) and neutral (functional) cases are indicated by *TP* (true positives) and *TN* (true negatives), and the number of incorrectly predicted pathogenic and neutral cases

		True class		Measures
		Positive	Negative	
Predicted class	Positive	True positive <i>TP</i>	False positive <i>FP</i>	Positive predictive value (PPV) $\frac{TP}{TP+FP}$
	Negative	False negative <i>FN</i>	True negative <i>TN</i>	Negative predictive value (NPV) $\frac{TN}{FN+TN}$
Measures		Sensitivity $\frac{TP}{TP+FN}$	Specificity $\frac{TN}{FP+TN}$	Accuracy $\frac{TP+TN}{TP+FP+FN+TN}$

**Figure 4 Contingency matrix and measures calculated based on it** 2x2 contingency table for displaying the outcome of predictions. Based on the table it is possible to calculate row and column wise parameters, PPV and NPV, and sensitivity and specificity, respectively. These parameters are useful, but are not based on all the information in the table. Accuracy is a measure that is calculated based on all the four figures in the table.

are *FN* (false negatives) and *FP* (false positives), respectively.

The goal of two-class prediction methods is to separate positive cases from negative ones. Because the predictions for the two classes usually overlap a cut off distinguishing the categories has to be optimized (Fig. 5). By moving the cut off different amounts of misclassified cases *FN* and *FP* appear. By using well behaved representative data and well trained classifier the misclassifications can be minimized.



**Figure 5 Separation of classes** In most classification problems the two classes are overlapping. By moving the cut off position the amount of the overlap of the classes can be adjusted. *FN* and *FP* are misclassified cases. The prediction methods aim at optimizing the cut off and thereby adjusting the numbers in the contingency table.

Based on the four descriptors several further measures can be calculated (Fig. 4). Sensitivity, also called true positive rate (*TPR*) or recall, and specificity (true negative rate, *TNR*) show the ratio of the pathogenic and neutral cases correctly identified by the programs. Positive predictive value (*PPV*) (also called precision) and negative predictive value (*NPV*) is the conditional probability that a pathogenic or neutral variant is predicted as pathogenic or neutral, respectively. The mathematical basis of these and other parameters have been discussed in detail [27].

A single parameter cannot capture all the information of the contingency matrix. Unless representative numbers of positive and negative cases are used, the values of *NPV* and *PPV* may be biased, even meaningless. The usual requirement is that the numbers be equal. Sometimes in literature the datasets are very skewed. Table 1 indicates the effect of the class imbalance. Results are shown in addition to equally distributed dataset also for analyses when there is ± 25 % or ±50 % difference in the total number of negative and positive cases. In the column wise parameters, which are for the ratios of either positive or negative cases (sensitivity and specificity), are not affected whereas there is a significant difference in *NPV* and *PPV*, which are row wise ratios based on numbers of both positive and negative cases. In all the examples, 75 % of both positive and negative cases are correctly predicted and therefore sensitivity and specificity remain the same. It is thus apparent that imbalance in class sizes grossly affects the *NPV* and *PPV* evaluation criteria.

To overcome the class imbalance problem different approaches can be taken. One is to prune the size of the bigger class to be that of the smaller one. It is also possible to normalize in the contingency table the values of either positive or negative cases to have the total of the other class. Quite often in bioinformatics limited amount of data are available and therefore one would be reluctant to delete part of the datasets. When normalizing the data be

**Table 1 Evaluation measures for test data**

	-50 %	-25 %	Equal	+25 %	+50 %
<b>tp</b>	750	750	750	750	750
<b>fn</b>	250	250	250	250	250
<b>tn</b>	375	563	750	938	1125
<b>fp</b>	125	187	250	312	375
<b>sensitivity</b>	0.75	0.75	0.75	0.75	0.75
<b>specificity</b>	0.75	0.75	0.75	0.75	0.75
<b>PPV</b>	0.86	0.80	0.75	0.71	0.67
<b>NPV</b>	0.60	0.69	0.75	0.79	0.82
<b>accuracy</b>	0.75	0.75	0.75	0.75	0.75
<b>MCC</b>	0.48	0.50	0.50	0.50	0.49

Example of a situation when both positive and negative cases have the same rate of correct predictions (75 %) and when the amount of negative cases is either equal or 25 or 50 % lower or higher than that for positive cases.

sure that the existing dataset is representative otherwise bias in the set may further be increased.

**Accuracy and MCC**

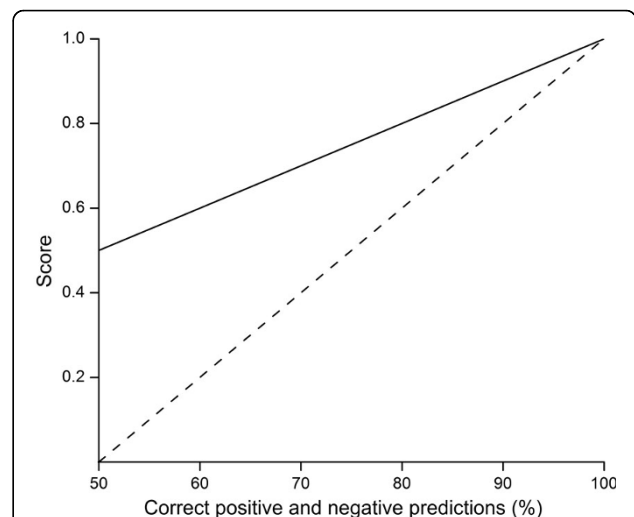
Specificity, sensitivity, *PPV* and *NPV* are calculated by using only half of the information in the contingency table and thus cannot represent all aspects of the performance. Accuracy (Fig. 4) and Matthews correlation coefficient (MCC) take benefit of all the four numbers and as such are more balanced, representative and comprehensive than the line or column wise measures.

The MCC is calculated as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TN + FP)(TP + FP)(TN + FN)}}$$

For all the measures discussed in here applies that higher the value the better. Except for MCC, the values range from 0 to 1. MCC ranges from -1 to 1. -1 indicates perfect negative correlation, 0 random distribution and 1 perfect correlation. Accuracy and MCC are affected by class imbalance only in extreme cases.

The effect of the correctly predicted cases on the parameters in equally distributed dataset is shown in Fig. 6. The value for MCC grows slower than the others reaching 0.5 when 75 % of cases are correctly predicted. Random results (50 % of both negative and positive correctly predicted) gives a value of 0, while the other parameters - sensitivity, specificity, *PPV*, *NPV*, and accuracy are 0.5. Fig. 6. can be used to check the performance of equally



**Figure 6 The growth of the performance measures along increasing reliability** Graphs for quality measures for equally distributed data (same amount positive and negative cases) when the performance increases equally in both classes. The solid curve indicates the growth of sensitivity, specificity, *PPV*, *NPV*, and accuracy. The dotted line is for MCC.

distributed dataset if e.g. some parameters in an article are not provided. Biases can easily be seen as deviations from the relationships in the figure. To obtain full picture of the predictor performance it is important to evaluate all the six measures together.

### Other parameters

Several other parameters can be derived from the contingency matrix. These are not discussed further as they are not widely used in literature and can be easily calculated from the six previously presented parameters. These include false positive rate (FPR) which equals 1-specificity and false negative rate (FNR) which is 1-sensitivity. False discovery rate (FDR) is 1-PPV.

Positive and negative likelihood ratios are calculated as follows:

$$LR+ = \frac{\text{sensitivity}}{1 - \text{specificity}} = \frac{\text{sensitivity}}{\text{FPR}} \text{ and}$$

$$LR- = \frac{1 - \text{sensitivity}}{\text{specificity}} = \frac{\text{FNR}}{\text{specificity}}.$$

F measure is another one that uses all the data. It is calculated as:

$$F = 2 \times \frac{\text{PPV} \times \text{sensitivity}}{\text{PPV} + \text{sensitivity}}.$$

Other measures include e.g. Hamming distance and quadratic distance (also called for Euclidean distance), which are the same for binary data, and relative entropy and mutual information [27].

### ROC analysis

Receiver operating characteristics (ROC) analysis is a visualization of prediction performance, that can be used

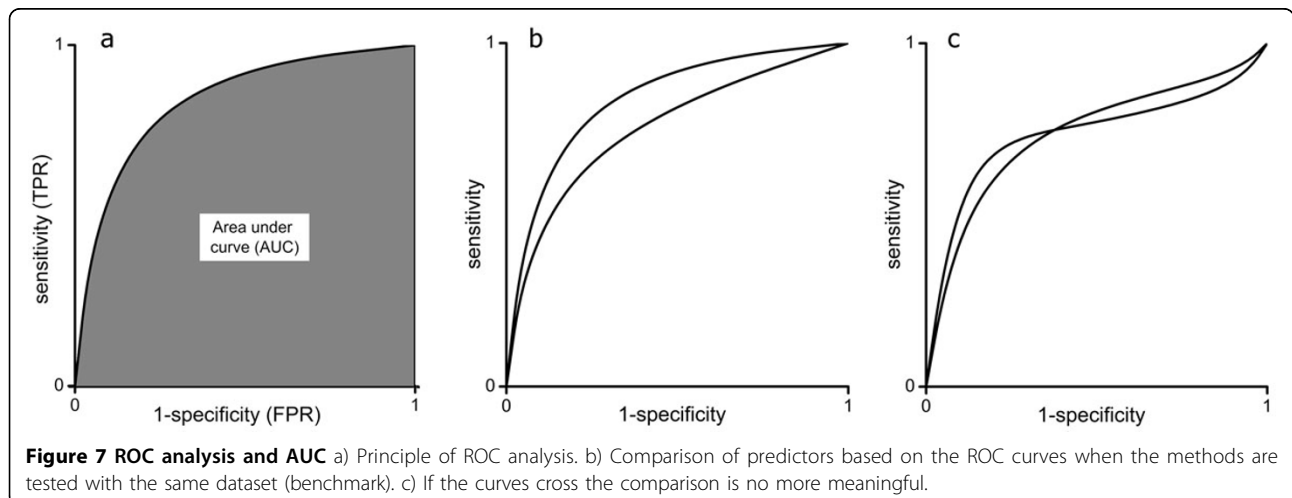
to select suitable classifier (for review see [28,29]). It indicates the tradeoffs between sensitivity and specificity. ROC curves can be drawn with specific programs when the predictor is of probabilistic type and provides a score for the classification. The score is usually not a real p value, but a value usable for ranking the predictions.

ROC curve (Fig. 7a) is drawn by first ranking the data based on the prediction score. Then the data are divided to intervals of equal size. The upper limit for the partitions is the number of cases in the dataset. ROC curve has on x-axis 1-specificity also called FPR and on the y-axis sensitivity (TPR).

Computer program establishes cut offs at intervals, calculates contingency table for data in the interval, and determines the values for sensitivity and 1-specificity, which is plotted to the graph. The procedure is repeated for each partition. If cross validation has been used, then the ROC curve can be used to show the average and variance of the results.

In an ideal case all the true positive cases are on the first half of the ranked list and the plot rises to (0,1) and then continues straight to the right with all the true negative cases. A random classification would be on the diagonal i.e. mixed correct and wrong cases. The faster the curve rises and the higher it reaches in the beginning the better the method is. Methods can be compared with ROC analysis when the same test dataset (benchmark) is used (Fig 7b). The curve that runs higher is for a better method. If the curves cross (Fig 7c) the comparison is no more meaningful.

Area under the ROC curve (AUC) has been used as a measure of goodness for predictions (Fig. 7a). It approximates the probability of ranking a randomly chosen positive instance higher than a randomly chosen negative one. A value of 0.5 indicates random and useless classification while 1 would indicate perfect classifier. Note that AUC



can be even smaller than 0.5. One should bear in mind that the ROC curve does not directly indicate the performance of a method. It shows the method's ranking potential, which is related to overall performance, further strengthening the fact that a single measure cannot fully describe the predictive performance even if it produces a graph.

### What if the data is classified to more than two classes?

If there are more than two classes the measures described above cannot be applied. The data can still be presented in an  $N \times N$  contingency table. One approach is to divide the data into several partitions of two categories.

If parameters are needed for all the classes there are some options available, however, single measures are more problematic. It is possible to calculate row and column wise ratios in the same way as in Fig. 4. MCC is in fact a special case for binary data of linear correlation coefficient, which can be used for several classes in its general format. Mutual information analysis can be used in these cases, as well. Applicable measures have been discussed e.g. in [27].

### Examples of performance comparisons

This section discusses examples of variation effect prediction method evaluations. These include methods for amino acid substitution (missense variation) tolerance, point variation effects on protein stability and variations related to mRNA splicing. The discussion concentrates on the comparison principles, especially in the light of the discussion on requirements mentioned above. The actual comparisons are not presented as it would have required publication of substantial parts of the reports. As a single parameter is insufficient for ranking methods, the readers are directed to the original articles to find all the details. Here a summary to the methodology and use of the evaluation parameters is provided.

#### Protein tolerance predictors

Single nucleotide alterations are the most common genetic variation type. Human genomes contain these variations on average at every kilobase. Several computational methods have been developed to classify these variations [1]. The evaluated methods were MutPred, nsSNPAnalyzer, Panther, PhD-SNP, PolyPhen, PolyPhen2, SIFT, SNAP, and SNPs&GO [5]. The methods differ in the properties of the variant they take into account, as well as in the nature and the classification method. Panther, PhD-SNP and SIFT are based on evolutionary information. MutPred, nsSNPAnalyzer, PolyPhen2, SNAP and SNP&GO combine protein structural and/or functional parameters and sequence analysis derived information. Most of these are based on machine-learning methods.

The positive test dataset included 19,335 missense variations from the PhenCode database [30], IDbases [31] and from 18 additional LSDBs. The negative dataset consisted of 21,170 nonsynonymous coding SNPs with an allele frequency  $>0.01$  and chromosome sample count higher than 49 from the dbSNP database. As large numbers of individual predictions were the Pathogenic-or-not Pipeline (PON-P) [32] was used for the submission of sequences and variants into the analysed programs.

The performance was evaluated with the six measures described above. The performances of the programs ranged from poor (MCC 0.19) to reasonably good (MCC 0.65) [5].

It has been widely accepted that information about protein three dimensional structure would increase prediction performance. The very best methods use also structural and functional information, whereas others that are solely based on sequence level information perform rather well.

Further analyses were made to compare the methods pairwise, and to study whether the type of original or substituting amino acid residue, the structural class of the protein, or the structural environment of the amino acid substitution, had an effect on the prediction performance.

Existing programs thus have widely varying performance and there is still need for better methods. Considering all the evaluation measures, no single method could be rated as best by all of them.

#### Protein stability predictors

Stability as a fundamental property affects protein function, activity, and regulation. Changes to stability are often found to be involved in diseases. Systematic performance evaluation analysis has been made for eleven stability predictors performances including CUPSAT, Dmutant, FoldX, I-Mutant2.0, two versions of I-Mutant3.0 (sequence and structure versions), MultiMutate, MUpro, SCide, Scpred, and SRide [2]. SCide and Scpred, which predict stability centers, as well as SRide, which predicts stabilizing residues, predict only destabilizing effects, while all the others evaluate both stabilizing and destabilizing changes.

The major database for protein stability information is ProTherm [33]. The pruned dataset for testing contained 1784 variations from 80 proteins, with 1154 positive cases of which 931 were destabilizing ( $\Delta\Delta G \geq 0.5$  kcal/mol), 222 were stabilizing ( $\Delta\Delta G \leq -0.5$  kcal/mol), and 631 were neutral ( $0.5 \text{ kcal/mol} \geq \Delta\Delta G \geq -0.5$  kcal/mol). The majority of the methods had been trained using data from ProTherm, and thus only those cases that had been added to the database after training had occurred were used for testing.

Of the measures recommended in here the authors used four, namely accuracy, specificity, sensitivity, and



MCC and the remaining row wise parameters could be calculated from the confusion tables.

There were three groups of data, stability increasing, neutral and stability decreasing. The authors solved the problem of multiple classes by presenting three tables of results. The first one was grouped so that both stability increasing and decreasing were considered as pathogenic i.e. positive. In these analyses only two classes were considered, stabilizing or destabilizing and neutral cases.

The results for the all the cases show that accuracy ranges from 0.37 to 0.64 and MCC from -0.37 to only 0.12. All the programs succeeded better when predicting stability increasing or decreasing variations individually. The MCC reaches 0.35 and 0.38 for the methods best in predicting stability increasing and decreasing variants, respectively [2].

Further analyses were made about variations located in different protein secondary structural elements, on the surface or in the core of a protein, and according to protein structure type.

The conclusion was that even at best, the predictions were only moderately accurate (~60%) and significant improvements would be needed. The correlation of the methods was poor.

In another study six programs including CC/PBSA, EGAD, FoldX, I-Mutant2.0, Rosetta, and Hunter were compared [3]. The dataset contained 2156 single variations from ProTherm. The goal of the study was to compare the performance of the methods in  $\Delta\Delta G$  prediction. Thus, they did not directly predict the effect on protein function, just the extent of free energy change. The only measure used was correlation between the experimental and predicted  $\Delta\Delta G$  values.

The ability of Dmutant, two versions of I-Mutant 2.0, MUpro, and PoPMuSiC to detect folding nuclei affected by variations has been evaluated [34]. The dataset contained 1409 variations from the ProTherm and some methods were tested with the same data which they had been trained. They used only correlation coefficients as quality measures. The best being in the range of ~0.5.

The performance of structure-based stability predictors, Dmutant, FoldX, and I-Mutant 2.0, were investigated with data for two proteins. There were 279 rhodopsin and 54 bacteriorhodopsin variations [35]. The best prediction accuracy for the rhodopsin dataset was <0.60, while it was somewhat greater for the bacteriorhodopsin dataset.

### Splice site predictors

mRNA maturation is a complex process, which may be affected by variations in many steps. Prediction behaviour of nine systems, GenScan, GeneSplicer, Human Splicing Finder (HSF), MaxEntScan, NNSplice, SplicePort, SplicePredictor, SpliceView and Sroogle was tested [4].

The test dataset contained altogether 623 variations. The first dataset contained 72 variations that affect the four invariant positions of 5' and 3' splice sites. The second one included 178 variations either localized at splice sites in non-canonical positions, distant intronic variations, and short distance variations. The third set of 288 exonic variations included 10 exonic substitutions that activate a cryptic splice site. In the fourth dataset were negative controls, altogether 85 variations without effect on splicing.

The results contain just the numbers of predicted cases and the percentage of correct ones, thus detailed analysis of the merits of the methods cannot be made.

The authors recommended some programs but stated that the *in silico* predictions need to be validated *in vitro*.

### Checklist for method developers and users

This checklist is provided to help when comparing and measuring performance of predictors and when selecting a suitable one. These are items that method developers should include in articles, or as supplement to articles, as they enable effective comparison and evaluation of the performance of predictors.

Items to check when estimating method performance and comparing performance of different methods:

- Is the method described in detail?
- Have the developers used established databases and benchmarks for training and testing (if available)?
- If not, are the datasets available?
- Is the version of the method mentioned (if several versions exist)?
- Is the contingency table available?
- Have the developers reported all the six performance measures: sensitivity, specificity, positive predictive value, negative predictive value, accuracy and Matthews correlation coefficient. If not, can they be calculated from figures provided by developers?
- Has cross validation or some other partitioning method been used in method testing?
- Are the training and test sets disjoint?
- Are the results in balance e.g. between sensitivity and specificity?
- Has the ROC curve been drawn based on the entire test set?
- Inspect the ROC curve and AUC.
- How does the method compare to others in all the measures?
- Does the method provide probabilities for predictions?

### Acknowledgements

This work was supported by the Sigrid Jusélius Foundation, Biocenter Finland and the Competitive Research Funding of Tampere University Hospital.

This article has been published as part of *BMC Genomics* Volume 13 Supplement 4, 2012: SNP-SIG 2011: Identification and annotation of SNPs in the context of structure, function and disease. The full contents of the supplement are available online at <http://www.biomedcentral.com/bmcgenomics/supplements/13/S4>.

#### Author details

<sup>1</sup>Institute of Biomedical Technology, FI-33014 University of Tampere, Finland.  
<sup>2</sup>BioMediTech, Tampere, Finland. <sup>3</sup>Department of Experimental Medical Science, Lund University, SE-221 84 Lund, Sweden.

#### Competing interests

The author declares that they have no competing interests in relation to the SNP-SIG issue article.

Published: 18 June 2012

#### References

1. Thusberg J, Vihinen M: Pathogenic or not? And if so, then how? Studying the effects of missense mutations using bioinformatics methods. *Hum Mutat* 2009, **30**:703-714.
2. Khan S, Vihinen M: Performance of protein stability predictors. *Hum Mutat* 2010, **31**:675-684.
3. Potapov V, Cohen M, Schreiber G: Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details. *Protein Eng Des Sel* 2009, **22**:553-560.
4. Desmet F, Hamroun G, Collod-Beroud G, Claustres M, Beroud C: Res. Adv. in Nucleic Acid Research. Global Research Network; 2010.
5. Thusberg J, Olatubosun A, Vihinen M: Performance of mutation pathogenicity prediction methods on missense variants. *Hum Mutat* 2011, **32**:358-368.
6. Moulton J, Fidelis K, Krysztofowicz A, Tramontano A: Critical assessment of methods of protein structure prediction (CASP)-round IX. *Proteins* 2011, **79**(Suppl 10):1-5.
7. Rodrigues AP, Grant BJ, Godzik A, Friedberg I: The 2006 automated function prediction meeting. *BMC Bioinformatics* 2007, **8**(Suppl 4):S1-4.
8. Wodak SJ: From the Mediterranean coast to the shores of Lake Ontario: CAPRI's premiere on the American continent. *Proteins* 2007, **69**:697-698.
9. McClure MA, Vasi TK, Fitch WM: Comparative analysis of multiple protein-sequence alignment methods. *Mol Biol Evol* 1994, **11**:571-592.
10. Thompson JD, Plewniak F, Poch O: BALiBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics* 1999, **15**:87-88.
11. Mizuguchi K, Deane CM, Blundell TL, Overington JP: HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci* 1998, **7**:2469-2471.
12. Raghava GP, Searle SM, Audley PC, Barber JD, Barton GJ: OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinformatics* 2003, **4**:47.
13. Edgar RC: MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res* 2004, **32**:1792-1797.
14. Van Walle I, Lasters I, Wyns L: SABmark—a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics* 2005, **21**:1267-1268.
15. Orengo CA, Michie AD, Jones S, Jones DT, Swindells MB, Thornton JM: CATH—a hierarchical classification of protein domain structures. *Structure* 1997, **5**:1093-1108.
16. Kolodny R, Koehl P, Levitt M: Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *J Mol Biol* 2005, **346**:1173-1188.
17. Lo Conte L, Ailey B, Hubbard TJ, Brenner SE, Murzin AG, Chothia C: SCOP: a structural classification of proteins database. *Nucleic Acids Res* 2000, **28**:257-259.
18. Sonogo P, Pacurar M, Dhir S, Kertesz-Farkas A, Kocsor A, Gaspari Z, Leunissen JA, Pongor S: A protein classification benchmark collection for machine learning. *Nucleic Acids Res* 2007, **35**:D232-236.
19. Hwang H, Vreven T, Janin J, Weng Z: Protein-protein docking benchmark version 4.0. *Proteins* 2010, **78**:3111-3114.
20. Cope LM, Irizarry RA, Jaffee HA, Wu Z, Speed TP: A benchmark for Affymetrix GeneChip expression measures. *Bioinformatics* 2004, **20**:323-331.
21. Zhu Q, Miecznikowski JC, Halfon MS: Preferred analysis methods for Affymetrix GeneChips. II. An expanded, balanced, wholly-defined spike-in dataset. *BMC Bioinformatics* 2010, **11**:285.
22. Gray J: *The Benchmark Handbook for Database and Transaction Systems*. Morgan Kaufmann; 1993.
23. Aniba MR, Poch O, Thompson JD: Issues in bioinformatics benchmarking: the case study of multiple sequence alignment. *Nucleic Acids Res* 2010, **38**:7353-7363.
24. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE: The Protein Data Bank. *Nucleic Acids Res* 2000, **28**:235-242.
25. Cotton RG, Al Aqeel AI, Al-Mulla F, Carrera P, Claustres M, Ekong R, Hyland VJ, Macrae FA, Marafie MJ, Paalman MH, et al: Capturing all disease-causing mutations for clinical and research use: toward an effortless system for the Human Variome Project. *Genet Med* 2009, **11**:843-849.
26. Kohonen-Corish MR, Al-Aama JY, Auerbach AD, Axton M, Barash CI, Bernstein I, Beroud C, Burn J, Cunningham F, Cutting GR, et al: How to catch all those mutations—the report of the third Human Variome Project Meeting, UNESCO Paris, May 2010. *Hum Mutat* 2010, **31**:1374-1381.
27. Baldi P, Brunak S, Chauvin Y, Andersen CA, Nielsen H: Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* 2000, **16**:412-424.
28. Fawcett T: An introduction to ROC analysis. *Pattern Recognition Letters* 2006, **27**:861-874.
29. Sonogo P, Kocsor A, Pongor S: ROC analysis: applications to the classification of biological sequences and 3D structures. *Brief Bioinform* 2008, **9**:198-209.
30. Giardine B, Riemer C, Hefferon T, Thomas D, Hsu F, Zielenski J, Sang Y, Elnitski L, Cutting G, Trumbower H, et al: PhenCode: connecting ENCODE data with mutations and phenotype. *Hum Mutat* 2007, **28**:554-562.
31. Piirilä H, Väliäho J, Vihinen M: Immunodeficiency mutation databases (IDbases). *Hum Mutat* 2006, **27**:1200-1208.
32. Olatubosun A, et al: PON-P: Integrated predictor for pathogenicity of missense variants. *Human Mutation* [<http://onlinelibrary.wiley.com/doi/10.1002/humu.22102/pdf>].
33. Kumar MD, Bava KA, Gromiha MM, Prabakaran P, Kitajima K, Uedaira H, Sarai A: ProTherm and ProNIT: thermodynamic databases for proteins and protein-nucleic acid interactions. *Nucleic Acids Res* 2006, **34**:D204-206.
34. Lonquety M, Lacroix Z, Chomilier J: *Pattern recognition in bioinformatics*. Heidelberg: Springer; 2008.
35. Tastan O, Yu E, Ganapathiraju M, Aref A, Rader AJ, Klein-Seetharaman J: Comparison of stability predictions and simulated unfolding of rhodopsin structures. *Photochem Photobiol* 2007, **83**:351-362.

doi:10.1186/1471-2164-13-S4-S2

Cite this article as: Vihinen: How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis. *BMC Genomics* 2012 **13**(Suppl 4):S2.

Submit your next manuscript to BioMed Central and take full advantage of:

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at  
[www.biomedcentral.com/submit](http://www.biomedcentral.com/submit)

