BIOINFORMATICS

Vol. 19 no. 17 2003, pages 2199–2209
DOI: 10.1093/bioinformatics/btg297

A comparative study of machine-learning methods to predict the effects of single nucleotide polymorphisms on protein function

V.G. Krishnan and D.R. Westhead*

School of Biochemistry and Molecular Biology, University of Leeds, Leeds LS2 9JT, UK

Received on February 26, 2003; revised on May 9, 2003; accepted on May 21, 2003

ABSTRACT

Motivation: The large volume of single nucleotide polymorphism data now available motivates the development of methods for distinguishing neutral changes from those which have real biological effects. Here, two different machine-learning methods, decision trees and support vector machines (SVMs), are applied for the first time to this problem. In common with most other methods, only non-synonymous changes in protein coding regions of the genome are considered.

Results: In detailed cross-validation analysis, both learning methods are shown to compete well with existing methods, and to out-perform them in some key tests. SVMs show better generalization performance, but decision trees have the advantage of generating interpretable rules with robust estimates of prediction confidence. It is shown that the inclusion of protein structure information produces more accurate methods, in agreement with other recent studies, and the effect of using predicted rather than actual structure is evaluated.

Availability: Software is available on request from the authors.

Contact: westhead@bmb.leeds.ac.uk

INTRODUCTION

An important aspect of the post-genome biology of model organisms and human is to understand the biological effects of inherited variations between individuals. For instance, a key problem for the pharmaceutical industry is to understand variations in drug treatment responses among individuals at the molecular level. Among these variations, single nucleotide polymorphisms (SNPs) have received much attention recently. SNPs are subtle variations, such as insertions, deletions and substitutions observed in the genomic DNA sequences of individuals of the same species. An enormous volume of SNP data are available in the public databases (http://snp.cshl.org and http://www.ncbi.nlm.nih.gov/SNP, Sherry *et al*., 2001).

SNPs in protein coding exons are classified as synonymous or non-synonymous according to whether or not they alter the protein sequence. Non-synonymous SNPs (nsSNPs) can affect gene function through their effect on the structure and function of the encoded protein. There are many examples of SNPs in coding regions that have a relationship with disease phenotypes (Chakravarti, 2001; Licinio and Wong, 2002). Equally, synonymous SNPs, and those outside protein coding regions can affect gene function through altered regulation, splicing and levels of protein expression. The volumes of SNP data now available pose a key question: can we predict which SNPs are likely to be neutral and which are likely to affect gene function?

Several recent studies have considered how deleterious and neutral nsSNPs might be distinguished using sequence and structural aspects of the proteins in which they occur. A study by Wang and Moult (2001) showed that most of the detrimental nsSNPs affect protein function indirectly through effects on protein structural stability, for instance by disruption to the protein hydrophobic core, and these authors provided a set of empirical rules to predict deleterious SNPs. Following this, other workers (Chasman and Adams, 2001; Sunyaev *et al*., 2001; Ramensky *et al*., 2002; Saunders and Baker, 2002) have asserted the importance of protein structural considerations, and developed prediction methods that depend on mapping SNPs to positions in (homologous) three-dimensional (3D) protein structures, as well as using information from multiple sequence alignments.

In contrast to the above studies, which use mapping to 3D structures, Ng and Henikoff (2001) have developed the SIFT (Sorting Tolerant from Intolerant) method, based on sequence conservation and scores from position-specific scoring matrices. These authors assert that their method performs ‘similarly’ to the structure-based methods. However, fair comparison of these tools is fraught with difficulties, and the intimate relationship of 3D structure with protein function and stability would suggest that the use of explicit structural information alongside sequence conservation will be found to improve performance, as supported by the recent study of Saunders and Baker (2002).

Of the tools described above, the Chasman and Adams method is the only one that involves automated learning from training data. While the other methods depend on rules derived empirically, this method uses a training data set to estimate the

*To whom correspondence should be addressed.

probability that a particular nsSNP will affect protein function. It is based on the description of a mutation or SNP in terms of a set of attributes, including sequence conservation and structural features. The probability of an effect is estimated from the proportion of training set mutations with matching attributes, in which an effect on function is known to occur. Effects are predicted if the probability is >0.5. The probability also serves as an estimate of the confidence level of the prediction. Attributes were chosen by a detailed statistical analysis of their effect on function, and the method was trained and cross-validated using the extensive systematic mutation data sets available for lysozyme (Alber *et al*., 1987; Rennell *et al*., 1991) and the lac repressor (Markiewicz *et al*., 1994; Suckow *et al*., 1996).

Automated learning from training data is an attractive alternative to manual tuning of empirical rules. After the set of descriptive attributes have been defined for each mutation, automated methods are able to explore much more fully how these attributes can be used to produce a prediction method that is, in some sense, approximately optimal. It is also much easier to perform rigorous cross-validation of such methods. Here, we report the application of two machine-learning methods, decision trees, as implemented in C4.5 (Quinlan, 1993) and support vector machines (SVMs) (Cristianini and Shawe-Taylor, 2000). The principal difference between these methods lies in the type of classifying function they attempt to learn. The decision tree represents the classifier as a tree structure in which each node represents a decision based on an attribute value, and it leads to a set of predictive rules that can be interpreted easily. On the other hand, the SVM relies on a mapping of the input attributes to a feature space that can be of very high dimension, where the classifier takes the form of a linear function (hyperplane). These methods have been found to be effective in many diverse fields (Mitchell, 1997). Here, we provide a comparison of the two methods and show that they have a contribution to make in SNP analysis.

The attributes used by our methods include both sequence and structure-based information, but in contrast to other methods we investigate the possibility of using only structural attributes that can be predicted with sufficient accuracy from sequence (secondary structure and solvent accessibility), rather than relying on mapping mutations to (homologous) 3D structures. By removing the need for a homologous structure the applicability of our method is extended significantly. It is not clear from the literature which of the currently available methods performs the best, but the availability of detailed cross-validation data and prediction confidence estimates for the Chasman and Adams method (above) is very convenient for comparison with our learning methods. Accordingly, we adopt their training sets (unbiased mutation data for lysozyme and lac repressor proteins), and replicate and extend their cross-validation techniques. Throughout this paper, the Chasman and Adams method is referred to as 'the probabilistic method'. As an example application of our method, we report the application of our method to the SNPs observed to occur between two strains of the nematode worm *Caenorhabditis elegans*.

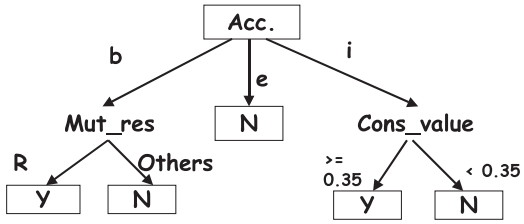

Fig. 1. A simplified example decision tree. Acc is the solvent accessibility (b = buried, e = exposed, i = intermediate), Mut_res is the mutated residue identity (R = Arginine), Cons_value is the conservation score of the original residue. In this case the final decision is binary, Y (effect) or N (no effect).

SYSTEM AND METHODS

Here, we provide only brief descriptions of the decision tree and SVM methods. More detail can be found in the references cited.

Decision trees

Decision tree learning (Mitchell, 1997; Witten and Frank, 2000) is a means for approximating discrete-valued target functions, in which the learned function is represented by a decision tree. Each instance (in this case a mutation or a SNP) is sorted down the tree from the root according to the values of its attributes (e.g. types of residues involved, sequence conservation, structural features) until it reaches a classifying leaf node ('effect' or 'no effect') where the prediction is made. This process is illustrated in the (fictitious) example shown in Figure 1.

Here we used the C4.5 decision tree software, which is derived from Hunt's method (Hunt *et al*., 1966) for constructing a decision tree. The software can be downloaded freely (http://www.cse.unsw.edu.au/~quinlan/). First, the decision tree was obtained for the training data using the program c4.5 and rules were generated by the program called c4.5rules, which uses the decision tree constructed by c4.5. Experiments were conducted to optimize the input parameters of the software (for both prediction accuracy and generalization), but the default values were found to be approximately optimal in most cases, and were used throughout this paper.

The decision tree software gives an estimated accuracy for each rule, which is derived from the training data. These estimated accuracies were used to assign confidence levels to the predictions. Rules with estimated accuracies of x% were taken to have a confidence level of x/100 (e.g. a rule with estimated accuracy of 90% was assigned an estimated confidence level of 0.9). The confidence level can be viewed as an estimate of the probability that a prediction from the rule is correct.

In order to facilitate comparison with other published methods, we report (see Results) error rates for predictions made from rules with confidence levels above a defined threshold (e.g. error rates at the confidence level threshold of 0.6 contain predictions from all rules whose confidence level is $\geqslant 0.6$).

SVMs

Currently, SVMs (Vapnik, 1998) are gaining great attention in the field of bioinformatics. Here, a classic two-class problem is addressed: SNPs have to be divided into two classes, 'effect' or 'no effect'. Like decision trees, SVMs use an input vector of attributes for each instance. Using a kernel function the input vectors are mapped to a feature space of high dimension in which the SVM method constructs a hyperplane that optimally separates instances from the two classes. Here, we constructed SVMs using mySVM (Vapnik, 1998, http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/). After various trials of different parameters for better performance (data not shown), we chose the polynomial kernel function of degree $d = 2$ given by

$$K(x, y) = (x^* y + 1)^d$$

The values of the other parameters for the mySVM software were $\varepsilon = 1.0 \times 10^{-12}$ and $C = 1$.

The mySVM software does not provide any estimate of confidence in classifications, and SVM theory in this area is currently not well developed. In contrast to our analysis of decision trees, therefore, we do not provide confidence levels for predictions made by SVMs.

Data sets

The systematic unbiased mutagenesis data set of lac repressor (Markiewicz *et al*., 1994; Suckow *et al*., 1996) and T4 lysozyme (Alber *et al*., 1987; Rennell *et al*., 1991) were used to train and validate the prediction methods. Mutations in the first 62 residues of lac repressor were omitted, because they are missing from protein databank structure of this protein (Berman *et al*., 2000). The number of mutations taken for the analysis was 3303 mutations for lac repressor and 1990 for lysozyme. The experimental results for both the proteins are given as four-valued expressions of the effect of each mutation on the protein function. In the case of lysozyme, plaque-forming ability was rated as $++$ (no effect), $+$ (slight effect), $+/-$ (larger effect) and $-$ (complete absence). Taking $++$ as a neutral mutation and all the rest as effects gives a data set in which 38% of mutations have an effect on function. In the case of the lac repressor the four values given were $+$ (no effect), $+-$ (slight effect), $-+$ (larger effect) and $-$ (complete absence). Here, '$+$' was considered as neutral and rest as effects, resulting in 45% of the mutations having an effect on the protein function. These definitions were adopted by Chasman and Adams.

It is not clear that the method above is the best way to convert the four experimental effect classes into a binary classification. Here, it was employed in order to give comparability to previous studies, which have adopted the same definition, and because it leads to a similar rate of mutations causing effects (38–45%) in each data set, suggesting that it defines a similar degree of effect in each protein. It is not clear how this definition relates to observable phenotypic effects on an organism, a point that we will discuss later.

In order to investigate these issues further we used a third, smaller data set (336 mutations) for the HIV protease (Loeb *et al*., 1989). This data set contains at least one mutation at every sequence position. In this case the experimentalists define only three degrees of effect, $+$ (no effect), $+/-$ (small effect), and $-$ (complete absence). Taking a definition analogous to the one adopted for the other data, where everything other than $+$ is considered as an effect, resulted in 67% of mutations being considered as effects. This percentage is significantly different to the 38–45% observed in lac repressor and lysozyme data. Therefore, in addition, we investigated how an alternative definition in which both $+$ and $+/-$ were treated as neutral would change the performance of the learning methods. With this latter definition 47% of protease mutations have an effect on function.

The SNPs data of *C.elegans* genome were obtained from St Louis Washington University web site (Wicks *et al*., 2001), (http://www.genome.wustl.edu/projects/celegans/index.php?snp=1) and the six chromosome data from Sanger centre website (http://www.sanger.ac.uk/Projects/C_elegans/WORMBASE/GFF_files.shtml). The SNPs data are between CB4856, an isolate from Hawaiian Islands, with reference to the completely sequenced N2 strain (from Bristol, UK). The SNP data are from part (5.4 Mb) of the *C.elegans* genome. The SNPs had to be associated to the position in the chromosome data by Fasta (Pearson and Lipman, 1988; Pearson, 1990) alignment. Using the exon, intron and intergenic information provided by Sanger centre web site, the protein coding regions were translated to get the corresponding protein sequences.

Attributes

The attributes of SNPs used for predictions were chosen from the following set: the residue identities of the original and mutated residue, the physicochemical classes of these residues (hydrophobic, polar, charged, glycine), sequence conservation score at the mutated position, molecular mass shift on mutation, hydrophobicity difference, secondary structure, solvent accessibility and buried charge. This set is based on other attribute sets from the literature; changes in these quantities on mutation are likely to affect protein function (e.g. by changing a key conserved functional residue) or protein structural stability (e.g. by disruption of the hydrophobic core through a residue size change reflected in a large molecular mass shift). Only the latter three attributes require information from protein structure rather than sequence, but these have been included because they can be predicted (see below) and so our method does not require mapping to a homologous 3D structure. In contrast to other methods, we do not use the

crystallographic B factor, which would limit the applicability of our method to sequences that can be mapped to homologous X-ray structures. In the Results section, we investigate the subset of these attributes that produces optimal cross-validated predictions.

The secondary structure and solvent accessibility information for lysozyme and lac repressor proteins was extracted from homology-derived secondary structure of proteins (Sander and Schneider, 1991) files, as 3D structures are available for these proteins. A three-state description of solvent accessibility (buried, intermediate and exposed) was used. In order to study the effect of using predicted rather than actual structure, and in cases of proteins of unknown structure, PHD (Rost and Sander, 1993, 1994) was used to predict secondary structure and solvent accessibility. The hydrophobicity values were taken from the literature. The sequence conservation score was calculated using the ScoreCons program (Valdar and Thornton, 2001a,b) using multiple sequence alignments of proteins extracted by BLAST (Altschul *et al*., 1990) searches of the SWALL database (Boeckmann *et al*., 2003) using an *E*-value cut-off of 0.01 and aligned with Clustal W (Thompson *et al*., 1994). The mutation mass shift was calculated as the difference between the relative molecular mass of the mutated residue and the original residue. The wild type residue was deemed to be a buried charge if it was one of K, R, D, E, H and its solvent accessibility was in the buried class.

Cross-validation methods

Machine learning methods are generally evaluated by a statistical technique called cross-validation. The data are divided into two sets randomly. The first ('training set') is used in training the learning method; the second ('test set') is used for subsequent evaluation of the accuracy of the trained method. This tests the ability of the method to generalize and make predictions on unknown data.

We report three types of cross-validation: homogeneous, heterogeneous (after Chasman and Adams) and mixed. For homogeneous cross-validation, each protein data set was taken separately and cross-validation performed on that set in isolation. For heterogeneous cross-validation, the data set of one protein (e.g. lysozyme) was used as training set and that of the other protein (e.g. lac repressor) was used as test set. For mixed cross-validation, the data from each protein was pooled as a single data set and cross-validation performed on this pooled set.

In case of homogeneous and mixed cross-validations, the data were randomized and split into 10 equal parts. One part was used as test set and the remainder as training set. This procedure was repeated 10 times so that each case or example (here it is each mutation) was used exactly once for testing. This is called 10-fold cross-validation, and has been shown to give good estimated error rates (Witten and Frank, 2000). In 10-fold cross-validation, the central tendency and spread of results were assessed as median and interquartile range (these were found to be very similar to the alternative mean and SD).

RESULTS

The results presented in this section concern error rates or misclassification rates, observed in predictions of functional effects of mutations (nsSNPs). Predictions are binary valued, 'effect' or 'no effect'; indicating whether or not a given SNP is predicted to have a deleterious effect on protein function. The error rate is the proportion of the total number of predictions that were wrong. In all cases, three different error rates are reported, the overall error rate, and separate error rates for positive (effect) predictions and negative (no effect) predictions. With some methods it is possible to attach an estimated confidence level to each prediction. This can be viewed as an estimate of the probability that the prediction is correct. If a threshold is set so that only predictions above a certain confidence level are accepted, then the number of predictions made usually decreases as this threshold is increased (i.e. if higher confidence is required then methods generally make fewer predictions). Therefore, we report number of predictions made as well as error rates: the better of two methods compared at the same confidence level or error rate is the one able to make the largest number of predictions.

Optimization of the set of attributes

Here, optimization means finding an attribute set that maximizes the total number of predictions while minimizing the overall error rate. Initially the sequence-based attributes, including conservation score and the identities of wild type and mutated residues and their physicochemical classes were chosen. Following this, attributes were added sequentially to this basic set to test their effect on the quality of the predictions. The performance of the decision tree method at a confidence level of 0.5 using mixed (lysozyme and lac repressor) cross-validation for an expanding attribute set is shown in Figure 2. It is clear from Figure 2 that each addition to the attribute set prompts a fall in the overall error rate and a slight increase in the number of predictions made. Also, the addition of structural attributes, such as buried charge, solvent accessibility and secondary structure information reduces error rates, in agreement with the conclusions drawn in previous study (Saunders and Baker, 2002). Given these observations, the full set of attributes (set 5 in Fig. 2) was used for learning in all the subsequent studies reported here.

The error rates in Figure 2 are all in the range 0.29–0.21. These are significantly lower than the best error rates that could be achieved with naïve prediction methods. A naïve method predicting either class randomly with equal probability would have an error rate of 0.5 on any test set, while one with knowledge of the composition of the test set could use this optimally by predicting the dominant class ('no effect' in this case) exclusively. In this case, the latter method would

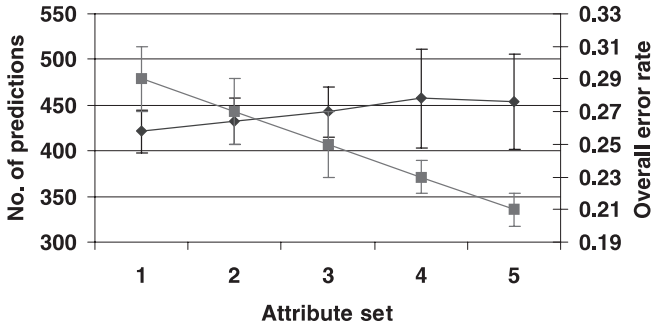

Fig. 2. The effect of including extra attributes on error rates (gray line and square markers) and prediction numbers (black line and diamond markers) in mixed cross-validation using decision tree learning (predictions with a confidence level of 0.5 or greater). Attribute set 1: the identities and physicochemical classes of wild type and mutated residues and also the sequence conservation score. Set 2: set 1 plus mass and hydrophobicity differences. Set 3: set 2 plus buried charge. Set 4: set 3 plus solvent accessibility. Set 5: set 4 plus secondary structure. Error rates are significantly lower (95% level) with set 5 compared to any other set (Wilcoxon rank sum test).

have an error rate of 0.42 (the proportion of 'effect' mutations in the data set).

Performance of decision tree learning

The results for homogeneous cross-validation are given in Table 1A. Results from the decision tree method are compared with the published results of the probabilistic method of Chasman and Adams (2001). Note that data in this table are cumulative: each confidence level threshold includes predictions made at a confidence level equal to *or higher than* the threshold. Both methods show the expected increase in both the overall number of predictions and observed error rates with decreasing confidence level threshold. The overall error rates of the decision tree method are generally significantly lower than those of the probabilistic method for both proteins. The exceptions to this are the error rates at high confidence levels (0.8 and 0.9) for the lac repressor. Viewing the 'effect' predictions separately, the error rates follow the same trend as the overall error rates, with the decision tree performing best at lower confidence thresholds. At higher thresholds (e.g. 0.9), error rates from the probabilistic method are lower, but these rates are achieved at the expense of making fewer predictions. For instance in the case of the lac repressor at a confidence threshold of 0.9, the probabilistic method makes 10 effect predictions with no errors (repeat cross-validations were not reported for this method), while (in multiple cross-validations) the decision tree makes on average 21.5 predictions and 1.5 errors. In the case of 'no effect' predictions the decision tree performs best on the lysozyme data in terms of both error rate and prediction numbers at each confidence level, but performance is more comparable on the lac repressor data.

It is interesting to compare observed error rates with the estimated confidence levels of predictions. For instance, in the case of the decision tree method the error rate at confidence threshold 0.9 (Table 1A) is close to the approximate expected value of 0.1 ($= 1-0.9$). In the case of both proteins, reflecting the use of rules with confidence value 0.9 or above. Comparing error rates to confidence levels for thresholds lower than 0.9 in Table 1A is more difficult because data are given cumulatively. For instance, the predictions at the threshold 0.5 include all predictions of confidence 0.5 *and above*, including many of much higher confidence and the corresponding error rate is therefore significantly <0.5.

Homogeneous cross-validation tests the ability of a method to learn rules applicable to a single protein. Heterogeneous cross-validation is a much more stringent and realistic test. It examines the ability of a method to learn rules that generalize from one protein to another. The results of heterogeneous cross-validation are shown in Table 1B. In this case, the overall error rates at all confidence levels are higher than in homogeneous cross validation, as expected for a more difficult test. However, in contrast with homogeneous cross validation the decision tree error rates are higher than those of the probabilistic method at all but the highest confidence level thresholds. This is an indication that while the decision tree performs best in homogeneous cross validation it is more prone to learning protein-specific rules that do not generalize well to other protein examples. The effect seems to be particularly marked for 'effect' predictions, with performance of the methods being more comparable for 'no effect' predictions.

Although extensive in the cases of lysozyme and lac repressor, the mutation data for the two proteins are still a very small sample of naturally occurring proteins, and it is almost certainly unreasonable to expect rules learned from a single protein to be universally applicable. To form our most accurate prediction method, we therefore used all the data in training. Error rates for such a method can be estimated by mixed cross-validation. The results of this for various confidence levels are presented in Table 2 (numbers *not* in parentheses). In this case, the observed error rates at each confidence level are much more similar to those observed in homogeneous cross validation, indicating that when both data sets are used for training the decision tree method is able to learn rules applicable to both proteins. Mixed cross-validation was not performed by Chasman and Adams (2001) so no comparison can be made with the probabilistic method in this case.

It is noticeable in several of the above cases that the cumulative number of 'effect' predictions tends to increase for each successive decrease in the confidence level threshold, while the number of 'no effect' predictions often reaches a plateau. For instance, in homogeneous cross-validation (Table 1A) the number of no effect predictions made with the decision tree method does not increase between confidence level thresholds of 0.7–0.5 in the case of either protein. This indicates that the decision tree rules predicting 'no effect' are all of a

Table 1. Prediction results and error rates from homogeneous (A) and heterogeneous (B) cross-validation at several confidence level thresholds

(A) Homogeneous

Prediction	Actual	Lysozyme: confidence threshold					Lac repressor: confidence threshold				
		0.9	0.8	0.7	0.6	0.5	0.9	0.8	0.7	0.6	0.5
Effect	Effect	11 ± 4 [3]	32.5 ± 3.3 [8]	42 ± 2.5 [24]	50.5 ± 3.7 [37]	52.5 ± 2.5 [43]	21.5 ± 10 [10]	52.5 ± 7 [34]	78 ± 6 [52]	95 ± 5.5 [66]	95 ± 2.9 [78]
	No effect	2 ± 9 [0]	3.5 ± 0.5 [2]	10 ± 2.5 [9]	16.5 ± 2.5 [14]	18.5 ± 2.3 [24]	1.5 ± 1.3 [0]	6.5 ± 1.9 [8]	15 ± 2.2 [17]	23.5 ± 2.7 [24]	24 ± 2.4 [27]
No Effect	No effect	5.5 ± 3.4 [2]	76 ± 9.2 [15]	86 ± 4 [37]	86 ± 4.4 [46]	86 ± 4.4 [63]	72 ± 35.8 [90]	127 ± 20.6 [122]	146 ± 17.9 [142]	146 ± 18.7 [156]	146 ± 18.6 [169]
	Effect	0 ± 0.38 [1]	9.5 ± 1.3 [7]	11 ± 1 [10]	11 ± 1 [18]	11 ± 1 [28]	4.5 ± 3 [3]	12 ± 2.2 [5]	17.5 ± 4.9 [22]	17.5 ± 5.3 [32]	17.5 ± 5.3 [43]
Overall error rate		0.11 ± 0.02 [0.17]	0.11 ± 0.02 [0.28]	0.15 ± 0.02 [0.24]	0.20 ± 0.03 [0.28]	0.2 ± 0.02 [0.33]	0.08 ± 0.02 [0.03]	0.12 ± 0.03 [0.08]	0.14 ± 0.03 [0.17]	0.16 ± 0.01 [0.20]	0.16 ± 0.01 [0.22]
Effect error rate		0.14 [0.00]	0.10 ± 0.04 [0.20]	0.18 ± 0.04 [0.27]	0.25 ± 0.04 [0.27]	0.25 ± 0.04 [0.36]	0.07 ± 0.06 [0.00]	0.12 ± 0.03 [0.19]	0.16 ± 0.02 [0.25]	0.20 ± 0.01 [0.27]	0.19 ± 0.01 [0.26]
No effect error rate		0 ± 0.03 [0.33]	0.12 ± 0.01 [0.32]	0.12 ± 0.02 [0.21]	0.12 ± 0.02 [0.28]	0.12 ± 0.02 [0.31]	0.07 ± 0.02 [0.03]	0.10 ± 0.03 [0.04]	0.13 ± 0.03 [0.13]	0.13 ± 0.03 [0.17]	0.13 ± 0.03 [0.20]

(B) Heterogeneous

Prediction	Actual	Training: Lysozyme Test: Lac repressor Confidence threshold					Training: Lac repressor Test: Lysozyme Confidence threshold				
		0.9	0.8	0.7	0.6	0.5	0.9	0.8	0.7	0.6	0.5
Effect	Effect	74 [68]	459 [227]	531 [358]	792 [483]	851 [551]	104 [30]	220 [70]	291 [156]	295 [271]	299 [341]
	No effect	6 [10]	301 [33]	405 [101]	632 [233]	708 [345]	23 [8]	81 [13]	172 [49]	174 [108]	176 [166]
No Effect	No effect	0 [101]	998 [259]	1107 [515]	1107 [666]	1107 [786]	225 [232]	388 [368]	483 [436]	483 [534]	483 [644]
	Effect	0 [9]	243 [27]	365 [109]	365 [132]	365 [182]	79 [90]	177 [135]	232 [183]	232 [233]	232 [328]
Overall error rate		0.07 [0.10]	0.27 [0.11]	0.32 [0.19]	0.34 [0.24]	0.35 [0.28]	0.24 [0.27]	0.30 [0.25]	0.34 [0.28]	0.34 [0.30]	0.34 [0.33]
Effect error rate		0.07 [0.13]	0.40 [0.13]	0.43 [0.22]	0.44 [0.33]	0.45 [0.39]	0.18 [0.21]	0.27 [0.16]	0.37 [0.24]	0.37 [0.28]	0.37 [0.33]
No effect error rate		NA [0.08]	0.2 [0.09]	0.25 [0.17]	0.25 [0.17]	0.25 [0.19]	0.26 [0.28]	0.31 [0.27]	0.32 [0.30]	0.32 [0.30]	0.32 [0.34]

Decision tree results (data outside parentheses) are compared with the probabilistic method (data in parentheses). The upper half of the table gives total prediction numbers and the lower half gives error rates.
Data for the probabilistic method taken from Chasman and Adams (2001) Tables 4 and 5. Decision tree results given as median ± interquartile range (repeat cross validations not available for the probabilistic method, or for heterogeneous cross validation with either method).

Table 2. Prediction results and error rates from mixed cross-validation at several confidence level thresholds

Prediction	Actual	Confidence threshold				
		0.9	0.8	0.7	0.6	0.5
Effect	Effect	28.5 ± 12.1	70 ± 7.4	105 ± 3.4	133 ± 4.1	135 ± 4.1
		(6.5 ± 2.6)	(32 ± 4.4)	(80 ± 4.8)	(110 ± 7.8)	(124 ± 7.9)
	No effect	2 ± 0.5	9 ± 2.7	26.5 ± 4.7	42.5 ± 2.6	44.5 ± 3.1
		(0 ± 3.8)	(4 ± 0.9)	(16.5 ± 4.8)	(31.5 ± 7.8)	(42.5 ± 6.8)
No Effect	No effect	32 ± 13.4	197 ± 37.4	228 ± 41.9	233 ± 37.4	233 ± 37.4
		(17.5 ± 2.4)	(115 ± 39.4)	(149 ± 45.8)	(157 ± 43.6)	(157 ± 43.6)
	Effect	1.5 ± 1.4	28 ± 3.8	40.5 ± 8.3	41.5 ± 7.3	41.5 ± 7.3
		(1.5 ± 1)	(17.5 ± 5.5)	(32 ± 4.9)	(37.5 ± 4)	(37.5 ± 4)
Overall error rate		0.08 ± 0.04	0.13 ± 0.02	0.18 ± 0.01	0.2 ± 0.01	0.21 ± 0.01
		(0.1 ± 0.04)	(0.14 ± 0.02)	(0.2 ± 0.03)	(0.21 ± 0.03)	(0.23 ± 0.02)
Effect error rate		0.09 ± 0.02	0.11 ± 0.02	0.20 ± 0.02	0.24 ± 0.01	0.25 ± 0.02
		(0 ± 0.03)	(0.12 ± 0.00)	(0.17 ± 0.04)	(0.23 ± 0.03)	(0.25 ± 0.03)
No effect error rate		0.08 ± 0.05	0.13 ± 0.01	0.16 ± 0.01	0.16 ± 0.02	0.16 ± 0.02
		(0.09 ± 0.05)	(0.15 ± 0.02)	(0.17 ± 0.03)	(0.18 ± 0.04)	(0.18 ± 0.04)

Decision tree results using actual structure (data outside parentheses) are compared with those using predicted structure (data in parentheses). The upper half of the table gives total prediction numbers and the lower half gives error rates.
Results given as median ± interquartile range.

confidence level 0.7 or above, while there are some rules of lower confidence for 'effect' predictions. It would seem to be easier to find high confidence rules for predicting 'no effect'.

Effect of predicted data

Here we compare decision tree performance using predictions for secondary structure and solvent accessibility instead of experimentally determined values. The results from mixed cross-validation using decision trees for various confidence levels are given in Table 2. It is encouraging that observed error rates are generally similar, or very slightly higher when predicted structure is used. However, the real effect of using predicted structure is seen when numbers of predictions are considered. When predicted structure is used it is clear that fewer predictions are possible at each confidence level threshold (e.g. at the confidence threshold of 0.9, on average 28.5 successful predictions of effects are made with actual structures and only 6.5 with predicted). The degradation of performance is therefore evident in prediction numbers rather than error rates, and this suggests that the predicted confidence levels of the decision tree rules are at least robust enough to recognize when lower quality data leads to less certain predictions.

The three-state secondary structure predictions used have accuracies of 74% (lysozyme) and 80% (lac repressor), and the three-state solvent accessibility predictions have accuracies of 51% (lysozyme) and 57% (lac repressor). It would be expected that our methods might perform less well on proteins for which these predictions were less accurate. However, with only two proteins available there is insufficient data to enable assessment of any trend relating the performance of our methods to the accuracy of these predictions.

Rules derived from decision trees

An advantage of the decision tree method is that it produces intelligible rules, and attaches a confidence level to each rule. For example, using the pooled data the final predictions are made on the basis of 50 rules predicting 'effect' and 39 rules predicting 'no effect'. It is not practical to analyse all the rules in this paper, but some illustrative examples are given below.

Rule 1: residue = L; mut_res = P; Obs_Acc = b $\geqslant$ class 'effect' [90.2%].

Rule 2: residue = G; Obs_Acc = b; Cons_value > 0.252; Cons_value $\leqslant$ 0.352 $\geqslant$ class 'effect' [77.1%]

Rule 3: residue = A; mut_res = G $\geqslant$ class 'no effect' [96.9%].

Here residue is the original residue, mut_res represents the mutated residue, Obs_Acc is the observed solvent accessibility of the original residue, and Cons_value is the conservation score of the original residue. The number in parentheses is the estimated percentage accuracy of the rule. The first rule indicates that changing a buried leucine residue to a proline tends to affect function, and can be understood in the light of our knowledge of the effect of proline on secondary structure. The second reflects the special nature of glycine (small side chain and high flexibility); replacing glycine in a buried and conserved position tends to affect the function. This second

Table 3. Comparison of the performance of decision trees (C4.5), SVMs and the probabilistic method for homogenous (A) and heterogeneous (B) cross-validation

(A) Homogeneous

Prediction	Actual	Lysozyme			Lac repressor		
		C4.5	SVM	Probabilistic method	C4.5	SVM	Probabilistic method
Effect	Effect	52.5 ± 2.5	44 ± 5.1	43	95 ± 2.9	92 ± 2.9	78
	No effect	18.5 ± 2.3	23 ± 1.6	24	24 ± 2.4	53 ± 4.5	27
No Effect	No effect	86 ± 4.4	100 ± 4.8	63	146 ± 18.6	144 ± 6.9	169
	Effect	11 ± 1	35 ± 2	28	17.5 ± 5.3	39 ± 3.4	43
Overall error rate		0.2 ± 0.02	0.29 ± 0.01	0.33	0.16 ± 0.01	0.27 ± 0.01	0.22
Effect error rate		0.25 ± 0.04	0.37 ± 0.04	0.36	0.19 ± 0.01	0.37 ± 0.02	0.26
No effect error rate		0.12 ± 0.02	0.26 ± 0.02	0.31	0.13 ± 0.03	0.20 ± 0.02	0.20

(B) Heterogeneous

Prediction	Actual	Training: Lysozyme Test: Lac repressor			Training: Lac repressor Test: Lysozyme		
		C4.5	SVM	Probabilistic method	C4.5	SVM	Probabilistic method
Effect	Effect	851	858	551	299	323	341
	No effect	708	475	345	176	186	166
No Effect	No effect	1107	1503	786	483	1042	644
	Effect	365	467	182	232	439	328
Overall error rate		0.35	0.28	0.28	0.34	0.31	0.33
Effect error rate		0.45	0.36	0.39	0.37	0.36	0.33
No effect error rate		0.25	0.24	0.19	0.32	0.30	0.34

The upper half of the table gives total prediction numbers and the lower half gives error rates.
Data for decision trees and probabilistic method taken from Tables 1 and 2 at confidence level threshold 0.5.

Table 4. Comparison of decision trees (C4.5 confidence level threshold 0.5) and SVM in mixed cross-validation

Prediction	Actual	C4.5	SVM
Effect	Effect	135 ± 4.1 (124 ± 7.9)	131 ± 4 (62 ± 4.5)
	No effect	44.5 ± 3.1 (42.5 ± 6.8)	70 ± 4.3(29 ± 6.6)
No Effect	No effect	233 ± 37.4 (157 ± 43.6)	251 ± 6.9 (291 ± 5.1)
	Effect	41.5 ± 7.3 (37.5 ± 4)	76 ± 5 (151 ± 6.4)
Overall error rate		0.21 ± 0.01 (0.23 ± 0.02)	0.28 ± 0.01 (0.33 ± 0.02)
Effect error rate		0.25 ± 0.02 (0.25 ± 0.03)	0.36 ± 0.01 (0.36 ± 0.05)
No effect error rate		0.16 ± 0.02 (0.18 ± 0.04)	0.23 ± 0.02 (0.34 ± 0.01)

Results using actual structure (data outside parentheses) are compared to those using predicted structure (data in parentheses). The upper half of the table gives total prediction numbers and the lower half gives error rates. Results given as median ± interquartile range.

rule gives both lower and upper limits on the conservation score, but we regard the upper limit as a feature reflecting learning on what is still a relatively small data set. On the other hand, rule 3 shows that substituting residues with similar properties tends not to affect the function. It is noteworthy that the first two 'effect' rules relate to changes in the stability of the structure, rather than specific effects on key functional residues.

Performance of SVMs

The results of our second learning method, SVMs, are given in detail in Tables 3 and 4. Here comparative results are taken from the 50% confidence threshold predictions of decision trees and the probabilistic method. It is not possible to provide confidence levels for SVM predictions (see System and Methods), but since SVMs provide a prediction for every data point, it is most meaningful to compare results with the

other methods where they make the largest number of predictions, i.e. including all predictions from the 0.5 confidence level upwards.

The results of homogeneous cross-validation in Table 3A indicate that decision trees tend to have the lowest error rates. The performances of SVM and probabilistic methods are similar, with the SVM performing better on lysozyme and the probabilistic method performing better on lac repressor. Prediction numbers vary from method to method but are broadly comparable. However, in heterogeneous cross-validation (Table 3B), it is clear that the decision tree method has higher error rates, particularly when the lysozyme data are used for training and lac repressor for testing (we earlier attributed this effect to the learning of protein-specific rules). In contrast with the decision tree, the SVM produces performance in heterogeneous cross-validation that is better than the probabilistic method. The error rates from the two methods are very similar, but at these rates the SVM is able to make significantly more successful predictions. For example, with lysozyme training and lac repressor test the SVM makes 858 successful 'effect' predictions and 1503 successful 'no effect' predictions, to be compared with 551 and 786 for the probabilistic method. This indicates that the SVM is less susceptible than the decision tree to protein-specific effects in the small learning set associated with a single protein.

In Table 4, we give a comparison of decision trees and SVMs in mixed cross-validation using both actual and predicted (numbers in parentheses) secondary structure and solvent accessibility. In this test, the decision tree out-performs the SVM in both cases. It is also interesting to note that while the effect of predicted data on the decision tree is easy to understand (error rates remain broadly similar but prediction numbers fall), it is more complicated for the SVM. The 'no effect' error rate increases significantly (from 0.23 to 0.34) when predicted data are used. This is probably related to the fact that our decision tree predictions are limited to those with a confidence level of 0.5 or greater. There is no estimate of confidence for the SVM and low confidence predictions (confidence level lower than 0.5) cannot be filtered out as they can with decision trees.

Difference in results depending on definitions of effect and no effect

The effect of a mutation on protein function takes a continuum of possible values from complete abolition of function through degrees of loss of functional efficiency to no observable effect. Yet this type of study requires conversion of this data to a binary valued 'effect' or 'no effect'. For the process of training a machine learning method to work it is clearly important that this conversion process be approximately consistent, i.e. that it defines an equivalent level of functional effect between different proteins. In the heterogeneous cross-validation using lysozyme and lac repressor data we found no evidence of inconsistency, and this was reinforced by the similar proportion of 'effect' mutations in each data set (38–45%, see System and Methods for details). However, this was not the case with the much smaller data set from the HIV protease.

Table 5. Prediction results for the HIV test set from methods trained on lysozyme and lac repressor data

Prediction	Actual	C4.5	SVM
Effect	Effect	173 (135)	160 (140)
	No effect	42 (80)	33 (94)
No Effect	No effect	25 (37)	78 (83)
	Effect	21 (9)	65 (19)
Overall error rate		0.24 (0.34)	0.29 (0.34)
Effect error rate		0.20 (0.37)	0.17 (0.40)
No effect error rate		0.46 (0.20)	0.45 (0.19)

The upper half of the table gives total prediction numbers and the lower half gives error rates. Numbers outside parentheses treat the HIV mutations as 'effect' if any effect on function was detected, those in parentheses require complete abolition of function for an 'effect'.
Decision tree (C4.5) results use prediction from confidence level threshold 0.5.

In Table 5, we show results obtained for predictions on the HIV data by methods trained on the combined lysozyme and lac repressor data. The numbers not in parentheses use a conversion analogous to that used for the training data, i.e. where any experimentally detected loss in activity is regarded as an effect (see Methods). With the first conversion, it is clear that both machine-learning methods produce very high 'no effect' error rates, in excess of 0.45 and close to the expected 0.5 for random predictions. Many of the mutations predicted to have no effect actually do have an effect according to this definition. This led us to suspect inconsistency of the conversion of experimental observations to binary values, and the observed high proportion of 'effect' mutations in the data set with this conversion (67%, much greater than that in the training data) was further evidence for this possibility. With this in mind, we reassessed the HIV predictions using the alternative conversion where only complete abolition of function was considered an effect (reducing the proportion of effect mutations in the HIV data to 47%, which is more consistent with the training data from lysozyme and lac repressor). This alternative conversion was applied to the HIV test data only and not to the training data, so there is no issue of unbalanced training. The results are shown in parentheses in Table 5. Changing to this conversion method clearly improves the 'no effect' error rate, but also significantly increases the error rate observed in 'effect' predictions. We interpret this to indicate that neither of these conversions is really consistent with the level of functional effects defined in the training data, the first defining too many minor functional changes as effects, and the second requiring too great a functional change to define an effect.

Applications of methods to *C.elegans* SNPs

As an illustration, we have applied our methods to predictions of the functional effects of a set of 803 nsSNPs between two

C.elegans strains. Using SVMs or decision trees trained on the combined lysoszyme and lac repressor data resulted in the prediction that around 300 (37%) of these might affect protein function (see Discussion).

DISCUSSION

We have made a thorough study of the use of two machine learning methods to predict the functional effects of SNPs, and compared the results with those from an existing probabilistic method (Chasman and Adams, 2001). Our results suggest that the machine learning methods we use are competitive with the probabilistic method and perform significantly better in some circumstances. Decision trees are able to provide predictions with significantly lower error rates in homogeneous cross-validation, but seem to do less well in the more difficult and realistic test of heterogeneous cross-validation. However, in this more difficult test our results show that the SVM was able to perform at the same error rates as the probabilistic method, while out-performing it in providing a significantly greater number of predictions.

In comparison with the SVM, and also with the probabilistic method, we found that decision tree learning was more susceptible to learning protein specific rules, resulting in very low error rates in homogeneous cross-validation, but significantly higher error rates in heterogeneous cross-validation. This might suggest that decision trees are not the method of choice for this problem. Nevertheless, decision trees do have advantages. First, they produce interpretable rules, and we have shown that these often make sense from a protein structure and stability perspective. Second, confidence levels can be derived for decision tree rules. Apart from the obvious utility of a confidence estimate to go with each prediction, we showed that these confidence estimates are generally very robust. When we moved from actual structural data to lower quality predicted data, this was recognized in the derivation of decision tree rules with reduced confidence. This effect was manifested as falling prediction numbers at each confidence level, while the observed error rates were maintained at approximately constant values. It is hardly surprising that decision trees learn protein-specific rules when faced with training data from a single protein, but this effect is clearly reduced when training is on data from more than one protein (see the Results for mixed cross-validation). In time, it is likely that suitable training sets for other proteins will become available, which should lead to the production of even higher quality decision trees.

The lack of confidence level estimates for SVM learning is a disadvantage of that method. It would seem likely that more confident SVM predictions would be those from data points located further from the optimal separating hyperplane, and that it should be possible to fit suitable probability distributions to the data to provide confidence estimates based on this distance. However, this theory is not well developed, and such calculations are not available in the software we used, or other commonly available software to our knowledge.

The inclusion of protein structure data in the attribute set has been discussed much in the literature (see Introduction). We find that the use of structural attributes like secondary structure, solvent accessibility and buried charge produces machine-learning methods that have lower error rates than those based on sequence features alone. It is likely that most mutations affecting protein function actually affect it indirectly through changes in structural stability, and therefore structural information should be valuable. An interesting further observation from decision tree learning is that rules predicting ‘no effect’ seem to have higher confidence levels on average. It would seem to be easier to predict if a mutation does not affect stability than to predict if it does.

It is important to appreciate that the way functional effects are defined can seriously affect predictions. For instance, it might be required to predict all observable effects on function in some applications, but just complete abolition of function in others. Methods are trained to predict a certain level of effect, and if applied to data sets where different levels of effect need to be predicted they will perform badly, as we illustrated with the HIV protease test data. It is very difficult to define equivalent levels of functional effect between two completely different proteins and this highlights a general problem with methods of this type. However, the heterogeneous cross-validation results we report here, and also those from the probabilistic method, suggest that the definitions we adopted for lysozyme and the lac repressor (an enzyme and a regulatory protein) are approximately equivalent. Based on this observation, it would seem reasonable to accept that a definition of effect resulting in a similar proportion of ‘effect’ mutations in unbiased mutation data sets for two proteins would indicate approximately equivalent definitions. However, application of this rule in the case of the HIV data did not produce a conclusive answer. More systematic mutation data sets with functional effects defined for different proteins are now required to assess the generalizability of both definitions and prediction methods.

This leads to the question of what level of effect should be predicted. The methods reported here were trained to predict any observable effect on protein function. In application to the *C.elegans* SNP data this leads to the prediction that 37% of nsSNPs might affect protein function. This number is quite large, given that the SNPs in this case are between two healthy strains, but it is not out of line with estimates made using other methods applied to human SNP data (e.g. Chasman and Adams, 2001; Sunyaev *et al.*, 2001). The degree of effect on protein function needed to produce observable phenotypic consequences or diseases will vary from gene to gene, but one possible explanation of these relatively large numbers is that the some of the protein functional consequences we predicted are too minor to cause major phenotypic effects. However, it is

not possible to rule out errors in the SNP data or the annotation of the *C.elegans* genome as alternative explanations.

In conclusion, we have shown that machine-learning methods can make a useful contribution to SNP prediction problems, and compete well with currently available methods. The generalization capability of the SVM is clearly a great advantage, but we have shown that decision trees too have significant advantages. A clear limitation of this study is the availability of only two really systematic and extensive mutation data sets for different proteins, but as more become available the power of all learning methods is sure to increase.

ACKNOWLEDGEMENTS

We thank Ian Hope, Matthew Woodwark and Cary O'Donnell for valuable discussions. V.G.K. would like to thank ORS (Overseas Research Students) awards scheme, Tetley Lupton and AstraZeneca Healthcare for support.

REFERENCES

Alber,T., Sun,D.P., Nye,J.A., Muchmore,D.C. and Matthews,B.W. (1987) Temperature-sensitive mutations of bacteriophage T4 lysozyme occur at sites with low mobility and low solvent accessibility in the folded protein. *Biochemistry*, **26**, 3754–3758.

Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Boeckmann,B., Bairoch,A., Apweiler,R., Blatter,M.C., Estreicher,A., Gasteiger,E., Martin,M.J., Michoud,K., O'Donovan,C., Phan,I. *et al*. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, **31**, 365–370.

Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.

Chakravarti,A. (2001) To a future of genetic medicine. *Nature*, **409**, 822–823.

Chasman,D. and Adams,R.M. (2001) Predicting the functional consequences of non-synonymous single nucleotide polymorphisms: structure-based assessment of amino acid variation. *J. Mol. Biol.*, **307**, 683–706.

Cristianini,N. and Shawe-Taylor,J. (2000) *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, UK.

Eisenberg,D., Weiss,R.M. and Terwilliger,T.C. (1984) The hydrophobic moment detects periodicity in protein hydrophobicity. *Proc. Natl Acad. Sci. USA*, **81**, 140–144.

Hunt,E.B., Martin,J. and Stone,P.J. (1966) *Experiments in Induction*. Academic Press, New York.

Loeb,D.D., Swanstrom,R., Everitt,L., Manchester,M., Stamper,S.E. and Hutchison,C.A., III (1989) Complete mutagenesis of the HIV-1 protease. *Nature*, **340**, 397–400.

Licinio,J. and Wong,M. (2002) *Pharmacogenomics*. WILEY-VCH Verlag GmbH, Weinheim, Germany.

Markiewicz,P., Kleina,L.G., Cruz,C., Ehret,S. and Miller,J.H. (1994) Genetic studies of the lac repressor. XIV. Analysis of 4000 altered *Escherichia coli* lac repressors reveals essential and non-essential residues, as well as 'spacers' which do not require a specific sequence. *J. Mol. Biol.*, **240**, 421–433.

Mitchell,M.T. (1997) *Machine Learning*. McGraw-Hill, US.

Ng,P.C. and Henikoff,S. (2001) Predicting deleterious amino acid substitutions. *Genome Res.*, **11**, 863–874.

Pearson,W.R. (1990) Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol.*, **183**, 63–98.

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Quinlan,J.R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.

Ramensky,V., Bork,P. and Sunyaev,S. (2002) Human non-synonymous SNPs: server and survey. *Nucleic Acids Res.*, **30**, 3894–3900.

Rennell,D., Bouvier,S.E., Hardy,L.W. and Poteete,A.R. (1991) Systematic mutation of bacteriophage T4 lysozyme. *J. Mol. Biol.*, **222**, 67–88.

Rost,B. and Sander,C. (1993) Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proc. Natl Acad. Sci. USA*, **90**, 7558–7562.

Rost,B. and Sander,C. (1994) Conservation and prediction of solvent accessibility in protein families. *Proteins*, **20**, 216–226.

Sander,C. and Schneider,R. (1991) Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, **9**, 56–68.

Saunders,C.T. and Baker,D. (2002) Evaluation of structural and evolutionary contributions to deleterious mutation prediction. *J. Mol. Biol.*, **322**, 891–901.

Sherry,S.T., Ward,M.H., Kholodov,M., Baker,J., Phan,L., Smigielski,E.M. and Sirotkin,K. (2001) dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res.*, **29**, 308–311.

Suckow,J., Markiewicz,P., Kleina,L.G., Miller,J., Kisters-Woike,B. and Muller-Hill,B. (1996) Genetic studies of the Lac repressor. XV: 4000 single amino acid substitutions and analysis of the resulting phenotypes on the basis of the protein structure. *J. Mol. Biol.*, **261**, 509–523.

Sunyaev,S., Ramensky,V., Koch,I., Lathe,W., III, Kondrashov,A.S. and Bork,P. (2001) Prediction of deleterious human alleles. *Hum. Mol. Genet.*, **10**, 591–597.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.

Valdar,W.S. and Thornton,J.M. (2001a) Conservation helps to identify biologically relevant crystal contacts. *J. Mol. Biol.*, **313**, 399–416.

Valdar,W.S. and Thornton,J.M. (2001b) Protein–protein interfaces: analysis of amino acid conservation in homodimers. *Proteins*, **42**, 108–124.

Vapnik,V. (1998) *Statistical Learning Theory*. John Wiley and Sons, Inc., New York.

Wang,Z. and Moult,J. (2001) SNPs, protein structure, and disease. *Hum Mutat.*, **17**, 263–270.

Wicks,S.R., Yeh,R.T., Gish,W.R., Waterston, R.H. and Plasterk,R.H. (2001) Rapid gene mapping in Caenorhabditis elegans using a high density polymorphism map. *Nat. Genet.*, **28**, 160–164.

Witten,I. and Frank,E. (2000) *Data Mining.* Morgan Kaufmann, Academic Press, USA.