IMPROVING FUNCTIONAL ANNOTATION OF NON-SYNONOMOUS SNPs WITH INFORMATION THEORY

R. KARCHIN, L.KELLY, A. SALI

Departments of Biopharmaceutical Sciences and Pharmaceutical Chemistry, and
California Institute for Quantitative Biomedical Research
Mission Bay Genentech Hall, 600 16th Street, Suite N472D
University of California, San Francisco
San Francisco, CA 94143-2240

Automated functional annotation of nsSNPs requires that amino-acid residue changes are represented by a set of descriptive features, such as evolutionary conservation, side-chain volume change, effect on ligand-binding, and residue structural rigidity. Identifying the most informative combinations of features is critical to the success of a computational prediction method. We rank 32 features according to their *mutual information* with functional effects of amino-acid substitutions, as measured by *in vivo* assays. In addition, we use a greedy algorithm to identify a subset of highly informative features [1]. The method is simple to implement and provides a quantitative measure for selecting the best predictive features given a set of features that a human expert believes to be informative. We demonstrate the usefulness of the selected highly informative features by cross-validated tests of a computational classifier, a support vector machine (SVM). The SVM's classification accuracy is highly correlated with the ranking of the input features by their mutual information. Two features describing the solvent accessibility of "wild-type" and "mutant" amino-acid residues and one evolutionary feature based on superfamily-level multiple alignments produce comparable overall accuracy and 6% fewer false positives than a 32-feature set that considers physiochemical properties of amino acids, protein electrostatics, amino-acid residue flexibility, and binding interactions.

1. Introduction

Over 70,000 coding non-synonymous SNPs (single nucleotide polymorphisms that produce a changed amino acid residue in a gene's protein product) have been identified in the human genome to date [2]. These changes can alter protein function and thus contribute to variation in disease susceptibility, drug efficacy, and drug toxicity [3,4].

In the past few years, a number of groups have developed computational methods to predict the functional effects of coding nsSNPs [5-9]. The first studies in this area relied on sequence information only. Cargill *et al.* classified substitutions as conservative or non-conservative using BLOSUM62 scores [10]. Ng and Henikoff substantially increased prediction accuracy with their SIFT algorithm, by computing a combination of a position-specific substitution score and a conservation score from a multiple alignment [6]. Several studies have attempted to understand how nsSNPs affect protein stability and function by mapping them onto protein structures, and developing rules to predict functionally important mutations [8,9]. Most recently, a few groups have applied machine learning methods to this problem [11,12].

Progress in this field depends on selection of a feature set that best represents the effects of amino-acid substitution in a protein. Chasman *et al.* applied a standard statistical analysis to 16 structure- and phylogeny-based features describing nsSNPs [5]. They tested each feature for association with functional effects by comparing tolerated and deleterious substitutions (from studies of T4 lysozyme and lac repressor [13-16]) using ANOVA (analysis of variance) and *chi-square* statistics. They selected a measure of residue flexibility, an entropy-based evolutionary conservation measure, solvent accessibility, buried charge, and "unusual amino acid". Saunders *et al.* evaluated a set of 12 features, using a similar dataset [7]. Each feature was assessed according to its ability to discriminate between tolerated and deleterious mutations in a 20-fold cross-validation test, with thresholds set to minimize classification error using a held-out partition of the data. They identified two optimal features: SIFT score (a measure of evolutionary conservation) [6] and a solvent-accessibility measure, based on the density of $C_ß$ atoms in the neighborhood of each residue.

Although no published studies have applied mutual information analysis to this problem, feature selection algorithms using mutual information have been used in many machine-learning applications, such as computer-aided medical diagnosis and text categorization [17,18]. These algorithms are well suited to the problem of predicting the functional consequences of nsSNPs. The mutual information metric is supported by rigorous mathematics and does not make assumptions that data fit a known family of distributions or that there are linear relationships between the features and classes to be predicted [1,19].

In this study, we rank a representative set of 32 candidate features according to their mutual information with categories of functional effects (*mutation classes*) observed in the T4 lysozyme and lac repressor assays (Section 2.2). We use the greedy MIFS algorithm to find an "optimized" set of features having large mutual information with the mutation classes and low redundancy with each other [1,20]. To validate that we are measuring the correct quantity, we evaluate the relationship between the performance of a state-of-the-art supervised learning method, a support vector machine (SVM), as well as the mutual information of its inputs (features) and desired outputs (mutation classes). We compare a variety of feature sets:

features selected by MIFS, features selected according to their mutual information rank, and a large set of all 32 features.

2. Methods

2.1. Dataset

Our evaluations were done on 6044 experimentally characterized point mutations in bacteriophage T4 lysozyme and *E. coli* lac repressor (data courtesy of Pauline Ng) [13-16]. 2015 mutations were from lysozyme and 4029 from lac repressor. Because the mutations were introduced in a systematic and unbiased fashion, this data has become a standard benchmark for methods that predict nsSNP functional effects. It has been used in several published studies, although direct comparison of results is difficult, as some groups have chosen to filter out mutations characterized as moderately deleterious or as temperature-sensitive [5-7,11].

2.2. Mutation Classes

The dataset is based on a plaque assay of bacteriophage T4 lysozyme mutants [13,14] and a colorimetric assay of *E. coli* lac repressor mutants [15,16]. Mutants were ranked according to reduced (or enhanced) function and assigned to a *mutation class*. In both experiments, mutants were assigned to four classes.

Because the mutation classes were based on visual inspection of plated cell cultures, these example labels are noisy. To deal with the problem, some groups reduce the four mutation classes to two by lumping all varieties of functional effect into a single class [5,6,11]. Others drop examples with moderate functional effect from the data set [7]. We chose the former definition (ANY-EFFECT, NO-EFFECT) because we are interested in predicting moderate as well as severe functional effects. Many disease susceptibility and drug response phenotypes are believed to result from interactions of SNPs in different genes that individually have moderate effects [21,22].

2.3. Candidate features

We evaluated 32 features potentially useful for computational prediction of the functional effects of nsSNPs, supplementing those found in the literature with a few of our own design. For detailed descriptions of each feature, see Appendix A. All features can be calculated inexpensively by a computer program or database lookup and thus are suitable for large-scale projects.

2.4. Feature Evaluation

In a computational classification method, each nsSNP is represented by a set of categorical- or numerically-valued features. From an information theoretic perspective, the classifier is a system that reduces our initial uncertainty about the experimentally-characterized mutation class of the nsSNP by "consuming" the information in the features. If there is sufficient information and it is used efficiently by the classifier, classification errors will be minimized [1]. The information that a feature X reveals about the mutation class Y can be quantified as *mutual information* (in units of bits):

$$I(X,Y) = \sum_{X,Y} p(X,Y) \log_2 (p(X,Y) / p(X)p(Y)) \qquad (1)$$

In this setting, the sum is over the cross-product of 6044 observations of feature X and mutation class Y in our data set.

Because we do not know the feature probability density functions p(X) and p(Y), we perform a histogram analysis to assign continuous data to discrete categories (bins). We build contingency tables for each (X,Y) pair to obtain the empirical estimates $\hat{p}(X,Y)$, $\hat{p}(X)$, $\hat{p}(Y)$ and $\hat{I}(X,Y)$.

Given the limited size of our data set, we opted to use a small number of bins in our histograms, rather than a large number of sparsely populated bins. All continuous-valued features were partitioned into five equal-frequency bins. Some of the tested features were categorical, such as buried vs. solvent-exposed residue position. These features had between two and five categories.

Mutual information is overestimated when sample size is small [23]; the effect becomes more pronounced as the number of bins increases and a greater number bins are undersampled. A feature with five bins will get a more inflated mutual information estimate than a feature with two bins. To deal with this problem, we applied a correction described by Cline *et al.* to compute *excess mutual information* I_E for each feature [24]:

$$\hat{I}_E(X,Y) = \hat{I}(X,Y) - E[\hat{I}_R(X,Y)] \qquad (2)$$

The correction term on the right-hand side of Eq. (2) is the expected value of *random mutual information* $E[I_R(X,Y)]$, where $E[I_R(X,Y)]$ is the mutual information between X and Y after the pairs are scrambled. (If our sample was sufficiently large, we should always have $I_R(X,Y)=0$, because the scrambling destroys all associations between X and Y.) We get stable estimates of $E[\hat{I}_R(X,Y)]$ by using 5,000 scramblings.

Features that are individually most informative about the mutation classes may be redundant, so we looked at how to select features that are most informative as a group. Our approach is to identify a subset of the candidate features that maximizes the joint *excess* mutual information of features and mutation classes [25].

$$\mathrm{J}_\mathrm{E} = I_E(X_1,\dots,X_k;Y) \tag{3}$$

Given that we have 32 candidate features, the space of possible subsets is too large for exhaustive search, and we approximate maximization of Eq. (4) with the MIFS algorithm, a greedy algorithm that selects a subset of features *S* having high I_E with the mutation classes and low I_E with each other [1]. The algorithm iteratively selects the feature that maximizes Eq. (4). The parameter ß, which is chosen empirically, controls the relative importance of the two selection criteria. We experimented with values of 0.25-1.0 and obtained best results with ß=0.5, so that each feature's mutual information with the mutation classes is given twice the importance of the penalty for feature redundancy (data not shown). Eq. (4) is a modification of the original MIFS objective function and produced superior results in our tests. It uses the excess mutual information correction (Eq. 2) and an improved feature redundancy measure suggested by Kwak et al. [20].

$$\hat{\mathrm{J}}_\mathrm{E} = \hat{I}_E(X,Y) - \text{ß}\sum_{s\in S}\frac{\hat{I}_E(Y,s)}{H(s)}\hat{I}_E(X_i,s), \quad 0 \le \text{ß} \le 1 \tag{4}$$

To select a desired number of features *m*, the algorithm initially sorts the features in the candidate set *F* by $I_E(X,Y)$. The top ranked feature is moved from *F* to the subset of selected features *S*. It proceeds for *m* iterations; at each iteration, the feature X_i in *F* that maximizes Eq. (4) is selected and moved from *F* to *S*, until finally *m* features are selected.

2.5. Testing protocol

We expect that a computational classifier will perform better when the most informative features are used as inputs. To test the predictive value of features selected by mutual information, we compared performance of: a large set of 32 features (Table 1); the top-ranked two features, the top-ranked three features, a set of five features selected by maximizing Eq. (4); and 28 sets of five features selected according to mutual information ranking. In the "ranked sets", the features were ordered according to their excess mutual information with the mutation classes (Table 1). Features ranked 1-5 were assigned to set number one; features ranked 2-6 were assigned to set number two, and so forth.

The features were used as inputs to a computational classifier known as a *support vector machine* (*SVM*) [26]. The SVM uses a *kernel function* to map the feature vectors into a high-dimensional space and find an optimal separation of examples from the different classes. We chose the SVM to reduce the possibility that classification errors were produced by inefficient classifier operation. SVMs are state-of-the-art classifiers and have been shown to be robust to noise and overfitting. In the results reported here, we used a radial basis kernel function and a 1-norm soft margin (with C=1) [27]. Although these choices produced our best results, they have not been carefully optimized.

To identify informative features relevant to both lac repressor and lysozyme, we used all of our mutation data in the feature ranking and selection process. We applied a stringent *heterogeneous* cross-validation protocol in which the SVM was trained on mutation data from one of the proteins and then tested on data from the other (and *vice versa*). Lac repressor and lysozyme are not structurally or functionally similar, so a classifier that does well on such a test is potentially suitable for predicting nsSNP functional effects in a wide range of globular proteins. *Homogeneous* cross-validation (training and testing on data from a single protein) achieves 15-20% higher classification accuracy, but these SVMs generalize poorly when tested on the other protein (data not shown).

Our experiments were done with in-house tools coded in Perl and Java. All software and alignments used in this analysis are available from the authors upon request.

Results

Table 1 shows the 32 evaluated features, ranked by mutual information with the mutation classes. If a particular mutation class Y and feature X always occurred together, such as all residues with functional effect having buried charges, the feature could be used to predict the correct mutation class to a certainty. In this case, $p(X,Y)=p(X)$ and the feature would have approximately 2 bits of information. This result can be derived by making the substitution in Eq (1), given the distributions of $\hat{p}(X)$ and $\hat{p}(Y)$. Here, the individual features are weakly informative – the best ones have only 0.1 bits. A select combination of several such features is required for accurate prediction of the mutation class.

Table 1. Thirty-two tested features, ordered by excess mutual information with the mutation classes (in bits). We performed a histogram analysis to assign continuous feature data to discrete categories (bins). WT=wild-type. MUT=mutant. HMM=hidden Markov model. For feature descriptions, see Appendix A. The top-ranked and bottom-ranked five-feature sets are shaded in gray.

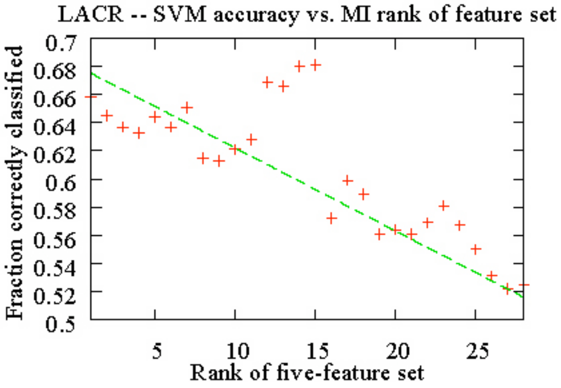
Features	Bins	Bits
Fractional solvent accessibility WT	5	0.104
HMM PHC score superfamily	5	0.103
Fractional solvent accessibility MUT	5	0.101
Solvent accessibility WT	5	0.096
Solvent accessibility MUT	5	0.089
HMM entropy subfamily	5	0.087
HMM relative entropy superfamily	5	0.083
Buried/exposed residue MUT	2	0.079
HMM PHC score subfamily	5	0.079
HMM entropy superfamily	5	0.073
HMM relative entropy subfamily	5	0.073
Buried/exposed residue WT	2	0.072
HMM relative entropy family	5	0.071
HMM PHC score family	5	0.071
HMM entropy family	5	0.071
DNA or small ligand contact	2	0.070
Change in solvent accessibility	5	0.052
Buried charge	2	0.045
Standardized residue B-factor	5	0.044
Change in residue hydrophobicity	5	0.026
Average residue B-factor	5	0.014
Change in fractional solvent acc.	5	0.014
EC/EU subfamily	2	0.007
SIFT score	2	0.005
Grantham values	5	0.004
Change from buried to exposed	3	0.002
Unusual residue	2	0.002
Change in residue formal charge	5	0.002
Domain interface contact	2	0.002
Change in residue volume	5	0.001
Turn breaker (P or G in turn)	2	-0.001
Helix breaker (P or G in helix)	2	-0.001

The top-three ranked features are fractional solvent accessibility (wild-type residue), HMM PHC score (superfamily alignments), and fractional solvent accessibility (mutant residue). Most of the information in the solvent-accessibility features comes from the fact that buried residue positions are most likely to be adversely effected by amino-acid substitutions, due to loss of structural stability [14,16,28,29]. Our structural modeling of amino-acid substitutions appears to capture some information about these energetic changes. This is reflected in the 2-3% accuracy increase between the top-two and top-three features shown in Table 2.

The superfamily-level PHC score, which considers several aspects of evolutionary conservation, is more informative than the simpler conservation measures we evaluated and more informative than family- or subfamily-based conservation measures (Appendix). The difference is not dramatic but suggests that superfamily alignments containing paralogous sequences can be useful in predicting untolerated amino-acid substitutions.

Table 2 shows the classification accuracy and false positive rate (fraction of mutations with effect that are incorrectly classified) of the SVM when tested on six feature sets. Accuracy is consistently better for lysozyme because a larger fraction of point mutations have no functional effect (0.68) than in lac repressor (0.56). If we "play the odds" (*e.g.,* randomly classifying effect and no-effect 32% and 68% of the time, respectively) for lysozyme, our classification accuracy will be 0.57 (vs. 0.51 for lac repressor). These accuracies are approximately what we get using the SVM with the five least informative features.

Figure 1**.** Support vector machine classification accuracy (fraction of correctly classified point mutations) is strongly correlated (*r*=-0.8) with the mutual-information rank of the five-feature sets described in Section 2.5. The line y = -0.006 x + 0.681 shows where the points would lie if correlation was at a maximum (*r*=-1.0**).** The outliers are four sets (ranks 12-15) that contain the feature "Ligand-binding-with-DNA". The feature is very informative for lac repressor, a DNA-binding protein, but not for lysozyme, which has no ligand or DNA contacts. The SVM cannot predict EFFECT or NO-EFFECT at the DNA-binding and ligand sites, given insufficient information. In this case, the default prediction value is assigned. In our implementation, the default prediction is EFFECT, producing the outliers.



We get best results with a redundant set of features (the five "top-ranked") that includes four descriptions of solvent accessibility. The top-three ranked features achieve classification accuracy comparable to the top-five ranked features, the optimized subset of five features, and the large set of 32 features. The remaining 29 features, which include buried charge, residue flexibility, subfamily-level evolutionary conservation, and physiochemical properties, make an insignificant contribution to classifier accuracy. In addition, classification specificity improves when the top-three features are used instead of all 32. The false positive rate (the fraction of mutations with no effect that are incorrectly classified) is reduced by more than 6%. Since experimental validation of predicted functional effects is often expensive and time-consuming, a low false-positive rate is desirable.

Table 2. Classification accuracy of SVM (fraction of correctly classified mutations) and false positive rate (fraction of mutations with no functional effect that are incorrectly classified) for six feature sets in cross-validation experiments described in Section 2.5. The dataset contains 4029 functionally characterized point mutations for lac repressor (LACR) and 2015 for lysozyme (LYS). The "optimized subset of five features" selected by the MIFS algorithm is shown in Table 3.

	Classification accuracy			**False positive rate**		
	LACR	LYS	Both	LACR	LYS	**Both**
Top-five ranked features	0.658	0.714	0.677	0.158	0.149	0.155
Top-three ranked features	0.668	0.702	0.679	0.172	0.153	0.165
Top-two ranked features	0.650	0.673	0.658	0.182	0.156	0.172
"Optimized" subset of five features	0.655	0.709	0.675	0.204	0.178	0.194
Large set of 32 features	0.654	0.748	0.685	0.272	0.157	0.229
Bottom-five ranked features	0.525	0.570	0.540	0.362	0.236	0.314

Table 3. Subset of five features selected by the MIFS algorithm, with $\hat{J}_E$ values (Eq. 4). Computations were done using all 6044 mutations in the lac repressor-lysozyme set.

Feature subset selected by MIFS algorithm	$\hat{J}_E$ (bits)
Fractional solvent accessibility (wild-type)	0.104
HMM PHC score, superfamily alignments	0.102
HMM entropy, subfamily alignments	0.081
Fractional solvent accessibility (mutant)	0.074
Residue in contact with DNA or within 5Å of small ligand	0.069

The "optimized" subset selected by MIFS does poorly compared to the five top-ranked features (the top-five set is equivalent to running MIFS with β=0). In this setting, reducing feature redundancy does not give an advantage, possibly because the information in individual features is weak and solvent accessibility is the most important predictor of deleterious amino-acid substitutions.

Figure 1 shows SVM classification accuracy (fraction of point mutations with correctly classified functional effects) vs. rank of the 28 five-feature sets for lac repressor (Section 2.5). Results are very similar for lysozyme. SVM classification accuracy is highly correlated with the excess mutual information ranking of the selected input features. Pearson's correlation coefficient *r* is -0.8 for lac repressor and -0.87 for lysozyme.

Summary

We have shown that mutual information is a useful tool in identifying biologically important features, given a set of functionally characterized point mutations. The strongest signals in the lac repressor/lysozyme set are solvent accessibility and superfamily-level evolutionary conservation. In cross-validated tests with a SVM classifier, using the top five ranked features gave us the lowest false positive rate.

We are currently working on applying this method to membrane proteins, using point mutation datasets generated by the Pharmacogenomics of Membrane Transporters project [30].

Acknowledgments

This work was supported by NIH grant U01-GM-61390-04. Thanks to Drs. K. Giacomini, D. Kroetz and the PMT project, Dr. K. Karplus for contributing to the script used in mutual information evaluations, and Dr. P. Ng for mutation data.

Appendix

Sequence-based features used in our study are listed in Table 2.

Table 2. Features based on amino-acid sequence only.

Grantham values	physiochemical difference between sidechains [31]
Net residue charge change	formal charge change between wild-type and mutant
Residue volume change	change in van der Waals volume [32]
Residue hydrophobicity change	change in hydrophobicity values [33]
Unusual residues	proline/glycine
Helix/Turn-breaker	proline or glycine in a helix or turn as defined by DSSP [5]

Evolutionary-conservation features were extracted from multiple alignments of sequences related at the superfamily level (common structure and function but low sequence similarity), the family level (paralogs and orthologs), and the subfamily level (orthologs only). Superfamily alignments and hidden Markov models (HMMs) were built with the SAM-T02 webserver [31]. Family and subfamily alignments and HMMs were constructed manually from superfamily alignments.

EC/EU: Defines an alignment column as either 100% conserved or unconserved.

Shannon entropy. A measure of conservation in a column of interest, where $\hat{p}(\vec{x})$ are the observed frequencies. Computed as $H(\vec{x}) = \sum_i \hat{p}(x_i)(1 / \lg \hat{p}(x_i))$ [34].

Relative entropy. Difference in entropy between $\hat{p}(\vec{x})$ and the background distribution of amino acids $\hat{p}(\vec{v})$, estimated from a large sample of alignments. Computed as $R(\vec{x},\vec{v}) = \sum_i \hat{p}(x_i, v_i) \lg \hat{p}(x_i) / \hat{p}(x_i, v_i)$.

PHC score. A score that considers the difference in conservation between wild-type (*W)* and mutant (*M*) amino acids and the conservation of the most-probable (*consensus)* amino acid *C*.

$$\text{PHC} = \log(|\, p(W) - p(M)\,|) + \log(p(W)) + \log(P(C)) - \log(p(M)) \qquad (5)$$

SIFT score. SIFT is an automated method that builds a multiple sequence alignment and computes the probability that a mutation is deleterious [6]. We obtained SIFT mutation scores from the SIFT site at http://blocks.fhcrc.org/sift.

Structural features are shown in Table 3. The features are based on PDB structure 1efa for *E. coli* lac repressor (2.6 Å resolution) [35] and 2lzm for bacteriophage T4 lysozyme (1.7 Å) [36]. For each point mutation, we used MODELLER (version 7.0) to perform sidechain replacement on the crystal structure [37].

Table 3. Features based on amino-acid residue solvent accessibility.

Solvent accessibility of wild-type/mutant	calculated by DSSP [38]
Change in solvent accessibility	between wild-type and mutant
Fractional solvent accessibility (FSA) of wild-type/mutant	normalized by maximum solvent accessibility for each residue type, using values from Rost [39]
Change in FSA	between wild-type and mutant
Buried or exposed wild-type/mutant	buried defined with FSA < 16%
Change in buried/exposed state	between wild-type and mutant

Residue B-factor. Average crystallographic temperature factor of residue backbone and sidechain atoms (proxy for residue rigidity) [5,7].

Standardized residue B-factor. Obtained by subtracting the mean and dividing by the standard deviation of residue B-factors in a protein of interest.

Buried charge. Charged residue in position with FSA<16%.

Turn/Helix breaker. Proline/glycine in a turn (or helix) as identified by DSSP [5].

Interaction features

Ligand-binding. We used the *LigBase* database to identify ligand-binding residues with atoms within 5Å of any HETATM listed in the PDB structure [40]. Lac repressor residues in contact with DNA were annotated using PDBSUM [41].

Domain-interface. We used the *PIBase* database to identify interface residues with atoms within 6Å of atoms in the residue of an oligomeric partner [40].

References

[1] Battiti, R. (1994) IEEE Trans. Neural Networks 5, 537-550.

[2] UCSC Genome Browser (build hg16) http://genome.ucsc.edu.

[3] McKusick, V. (2000) OMIM http://ncbi.nlm.nih.gov/omim.

[4] Klein, T.E. *et al.* (2001) Pharmacogenomics J 1, 167-70.

[5] Chasman, D. and Adams, R.M. (2001) J.Mol.Biol. 307, 683-706.

[6] Ng, P.C. and Henikoff, S. (2001) Genome Res. 11, 863-874.

[7] Saunders, C.T. and Baker, D. (2002) J.Mol.Biol. 322, 891-901.

[8] Sunyaev, S. *et al.* (2001) Hum.Mol.Genet. 10, 591-597.

[9] Wang, Z. and Moult, J. (2001) Hum.Mutat. 17, 263-270.

[10] Cargill, M. *et al.* (1999) Nat.Genet. 22, 231-238.

[11] Krishnan, V.G. and Westhead, D.R. (2003) Bioinformatics 19, 2199-209.

[12] Yue, P., Li, Z. and Moult, J. (2004).

[13] Alber, T. *et al.* (1987) Biochemistry 26, 3754-8.

[14] Rennell, D. *et al.* (1991) J Mol Biol 222, 67-88.

[15] Suckow, J. *et al.* (1996) J Mol Biol 261, 509-23.

[16] Markiewicz, P. *et al.* (1994) J Mol Biol 240, 421-33.

[17] Tourassi, G.D.F. *et al.* (2001) Med. Phys. 28, 2394-2402.

[18] Dumais, S.P. *et al.* (1998) in: 7th ACM International Conference on Information and Knowledge Management, pp. 148-155 ACM Press.

[19] Li, W. (1990) J Stat Phys 60, 823-837.

[20] Kwak, N. and Choi, C.H. (1999) in: IJCNN, Vol. 2, pp. 1313-1318.

[21] Evans, W.E. and Relling, M.V. (1999) Science 286, 487-91.

[22] Dean, M. (2003) Hum Mutat 22, 261-74.

[23] Wolpert, D.H.W., D.R; (1995) Phys. Rev. E. 52, 6841-6854.

[24] Cline, M.S. *et al.* (2002) Proteins 49, 7-14.

[25] Cover, T.M. and Joy, T.A. (1991) John Wiley and Sons, New York.

[26] Vapnik, V. (1995) Springer-Verlag.

[27] Cristianini, N. *et al.* (2000) Cambridge University Press.

[28] Sunyaev, S..*et al* (2000) Trends Genet 16, 198-200.

[29] Bowie, J.U. *et al.* (1990) Science 247, 1306-10.

[30] Leabman, M.K. et al. (2003) Proc Natl Acad Sci U S A 100, 5896-901.

[31] Grantham, R. (1974) Science 185, 862-864.

[32] Zamyatin, A.A. (1972) Prog. Biophys. Mol. Biol. 24, 107-123.

[33] Engelman, D.M. *et al.*(1986) Annu Rev Biophys Chem 15, 321-53.

[34] Shenkin, P.S. *et al.* (1991) Proteins 11, 297-313.

[35] Bell, C.E. *et al.*(2000) Cell 101, 801-11.

[36] Weaver, L.H. and Matthews, B.W. (1987) J Mol Biol 193, 189-99.

[37] Fiser, A. and Sali, A. (2003) Methods Enzymol 374, 461-91.

[38] Kabsch, W. and Sander, C. (1983) Biopolymers 22, 2577-637.

[39] Rost, B. and Sander, C. (1994) Proteins 20, 216-26.

[40] Pieper, U. *et al.* (2004) Nucleic Acids Res. 32, D217-D222.

[41] Laskowski, R.A. *et al.* (1997) Trends Biochem Sci 22, 488-90.