Prediction of Protein Stability Changes for Single-Site Mutations Using Support Vector Machines

Jianlin Cheng,[1] Arlo Randall,[1] and Pierre Baldi[1,2]*

[1]*Institute for Genomics and Bioinformatics, School of Information and Computer Sciences, University of California, Irvine, California*

[2]*Department of Biological Chemistry, College of Medicine, University of California, Irvine, California*

ABSTRACT **Accurate prediction of protein stability changes resulting from single amino acid mutations is important for understanding protein structures and designing new proteins. We use support vector machines to predict protein stability changes for single amino acid mutations leveraging both sequence and structural information. We evaluate our approach using cross-validation methods on a large dataset of single amino acid mutations. When only the sign of the stability changes is considered, the predictive method achieves 84% accuracy—a significant improvement over previously published results. Moreover, the experimental results show that the prediction accuracy obtained using sequence alone is close to the accuracy obtained using tertiary structure information. Because our method can accurately predict protein stability changes using primary sequence information only, it is applicable to many situations where the tertiary structure is unknown, overcoming a major limitation of previous methods which require tertiary information. The web server for predictions of protein stability changes upon mutations (MUpro), software, and datasets are available at http://www.igb.uci.edu/servers/servers.html. Proteins 2006; 62:1125–1132.** © 2005 Wiley-Liss, Inc.

INTRODUCTION

Single amino acid mutations can significantly change the stability of a protein structure. Thus, biologists and protein designers need accurate predictions of how single amino acid mutations will affect the stability of a protein structure.[1–7] The energetics of mutants has been studied extensively both through theoretical and experimental approaches. The methods for predicting protein stability changes resulting from single amino acid mutations can be classified into four general categories: (1) physical potential approach; (2) statistical potential approach; (3) empirical potential approach; and (4) machine learning approach.[8] The first three methods are similar in that they all rely on energy functions.[9] Physical potential approaches[10–17] directly simulate the atomic force fields present in a given structure and, as such, remain too computationally intensive to be applied to large datasets.[9] Statistical potential approaches[16,18–25] derive potential functions using statistical analysis of the environmental propensities, substitution frequencies, and correlations of contacting residues in solved tertiary structures. Statistical potential approaches achieve predictive accuracy comparable to physical potential approaches.[26]

The empirical potential approach[9,27–34] derives an energy function by using a weighted combination of physical energy terms, statistical energy terms, and structural descriptors, and by fitting the weights to the experimental energy data. From a data fitting perspective, both machine learning methods[8,35,36] and empirical potential methods learn a function for predicting energy changes from an experimental energy dataset. However, instead of fitting a linear combination of energy terms, machine learning approaches can learn more complex nonlinear functions of input mutation, protein sequence, and structure information. This is desirable for capturing complex local and nonlocal interactions that affect protein stability. Machine learning approaches such as support vector machines (SVMs) and neural networks are more robust in their handling of outliers than linear methods, thus, explicit outlier detection used by empirical energy function approaches[9] is unnecessary. Furthermore, machine learning approaches are not limited to using energy terms; they can readily leverage all kinds of information relevant to protein stability. With suitable architectures and careful parameter optimization, neural networks can achieve performance similar to SVMs. We choose to use SVMs in this study because they are not susceptible to local minima and a general high-quality implementation of SVMs (SVMlight[37,38]) is publicly available.

Most previous methods use structure-dependent information to predict stability changes, and therefore cannot be applied when tertiary structure information is not available. Although nonlocal interactions are the principal determinant of protein stability,[19] previous research[19,34,35] shows that local interactions and sequence information can play important roles in stability prediction. Casadio et

Grant sponsor: the National Institute of Health Biomedical Informatics Training grant; Grant number: LM-07443-01; Grant sponsor: National Science Foundation MRI grant; Grant number: EIA-0321390; Grant sponsor: the University of California Systemwide Biotechnology Research and Education Program (UC BREP); Grant sponsor: the Institute for Genomics and Bioinformatics at UCI.

*Correspondence to: Pierre Baldi, Institute for Genomics and Bioinformatics, School of Information and Computer Sciences, University of California, Irvine, Irvine, CA 92697-3425. E-mail: pfbaldi@ics.uci.edu

Received 25 April 2005; Revised 4 August 2005; Accepted 19 September 2005.

Published online 21 December 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.20810

al.[35] uses sequence composition and radial basis neural networks to predict the energy changes caused by mutations. Gillis and Rooman[19,39] show that statistical torsion potentials of local interactions along the chain based on propensities of amino acids associated with backbone torsion angles is important for energy prediction, especially for the partially buried or solvent-accessible residues. The AGADIR algorithm,[28,29] which uses only local interactions, has been used to design the mutations that increase the thermostability of protein structures. Bordner and Abagyan[34] show that the empirical energy terms based on sequence information can be used to predict the energy change effectively, even though accuracy is still significantly lower than when using structural information. Frenz[36] uses neural networks with sequence-based similarity scores for mutated positions to predict protein stability changes in Staphylococcal nuclease at 20 residue positions.

Here we develop a new machine-learning approach based on support-vector machines to predict the stability changes for single site mutations in two contexts taking into account structure-dependent and sequence-dependent information, respectively. In the first classification context, we predict whether a mutation will increase or decrease the stability of protein structure as in Capriotti et al.[8] In this framework, we focus on predicting the sign of the relative stability change ($\Delta\Delta G$). In many cases, the correct prediction of the direction of the stability change is more relevant than its magnitude.[8] In the second regression context, we use an SVM-based method to predict directly the $\Delta\Delta G$ resulting from single site mutations, as most previous methods do. A direct prediction of the value of relative stability changes can be used to infer the directions of mutations by taking the sign of $\Delta\Delta G$.

There are a variety of ways in which sequence information can be used for protein stability prediction. Previous methods use residue composition[35] or local interactions derived from a sequence. Our method directly leverages sequence information by using it as an input to the SVM. We use a local window centered around the mutated residue as input. This approach has been applied successfully to the prediction of other protein structural features, such as secondary structure and solvent accessibility.[40–43] The direct use of sequence information as inputs can help machine learning methods extract the sequence motifs which are shown to be important for protein stability.[29] We take advantage of the large amount of experimental mutation data deposited in the ProTherm[44] database to train and test our method. On the same dataset compiled in Capriotti et al.,[8] our method yields a significant improvement over previous energy-based and neural network-based methods using 20-fold cross-validation.

An important methodological caveat results from the dataset containing a significant number of identical mutations applied to the same sites of the same proteins. We find that it is important to remove the site-specific redundancy to accurately evaluate the prediction performance for mutations at different sites. On the redundancy-reduced dataset, the prediction accuracy obtained using sequence information alone is close to the accuracy obtained using additional structure-dependent information. Thus, our method can make accurate predictions in the absence of tertiary structure information. Furthermore, to estimate the performance on unseen and nonhomologous proteins, we remove the mutations associated with the homologous proteins and split the remaining mutations by individual proteins. We use the mutations of all proteins except for one to train the system and use the remaining one for testing (leave-one-out cross validation). Thus we empirically estimate how well the method can be generalized to unseen and nonhomologous proteins.

MATERIALS AND METHODS

Data

We use the dataset S1615 compiled by Capriotti et al.[8] S1615 is extracted from the ProTherm[44] database for proteins and mutants. The dataset includes 1615 single site mutations obtained from 42 different proteins. Each mutation in the dataset has six attributes: PDB code, mutation, solvent accessibility, pH value, temperature, and energy change ($\Delta\Delta G$). To make the values of solvent accessibility, pH, and temperature in the same range as the other attributes, they are divided by 100, 10, and 100, respectively. If the energy change $\Delta\Delta G$ is positive, the mutation increases stability and is classified as a positive example. If $\Delta\Delta G$ is negative, the mutation is destabilizing and is classified as a negative example. For the classification task, there are 119 redundant examples that have exactly the same values as some other example for all six attributes, provided only the sign of the energy changes is considered. These examples correspond to identical mutations at the same sites of the same proteins with the same temperature and pH value, only the magnitudes of the energy changes are slightly different. To avoid any redundancy bias, we remove these examples from the classification task. We refer to this redundancy-reduced dataset as SR1496. To leverage both sequence and structure information, we extract full protein sequences and tertiary structures from the Protein Data Bank[45] for all mutants according to their PDB codes.

We test three different encoding schemes (SO: sequence only, TO: structure only, ST: sequence and structure) (see Inputs and Encoding Schemes, below). Since solvent accessibility contains structure information, to compare SO with TO and ST fairly, we test the SO scheme without using solvent accessibility on the SR1496 dataset. All schemes are evaluated using 20-fold cross validation. Under this procedure, the dataset is split randomly and evenly into 20 folds. Nineteen folds are used as the training dataset and the remaining fold is used as the test dataset. This process is repeated 20 times where each fold is used as the test dataset once. Performance results are averaged across the 20 experiments. The cross-validated results are compared with similar results in the literature obtained using a neural network approach.[8] Using the same experimental settings as in Capriotti et al.,[8] the subset S388 of the S1615 dataset is also used to compare our predictor with other predictors based on potential

functions and available over the web. The S388 dataset includes 388 unique mutations derived under physiological conditions. We gather the cross validation predictions restricted to the data points in the S388 dataset, and then compute the accuracy and compare it with the three energy function based methods[9,23,24,39] available over the web.

There is an additional subset of 361 redundant mutations that are identical to other mutations in the S1615 dataset, except for differences in temperature or pH. The energy changes of these mutations are highly correlated; and the signs of the energy changes are always the same with a few exceptions. This is in contrast to the S388 subset, which contains no repeats of the same mutations at the same site. We find that it is important to remove this redundancy for comparing the performance of structure-dependent and sequence-dependent encoding schemes. Thus we derive a dataset without using solvent accessibility, pH, and temperature information and remove all the mutations—with the same or different temperature and pH value—at the same site of the same proteins. This stringent dataset includes 1135 mutations in total. We refer to this dataset as SR1135.

In order to estimate the performance of mutation stability prediction on unseen and nonhomologous proteins, we use also UniqueProt[46] to remove homologous proteins by setting the HSSP threshold to 0, so that the pairwise similarity between any two proteins is below 25%. Because the proteins in the S1615 dataset are very diverse, only six proteins (1RN1, 1HFY, 1ONC, 4LYZ, 1C9O, and 1ANK) are removed. We remove 154 mutations associated with these proteins. Then we split the mutation data into 36 folds according to the remaining 36 proteins. For each fold, we further remove all the identical mutations at the same sites. There are 1023 mutations left in total. We refer to this dataset as SR1023. We apply an encoding scheme using only sequence information to this dataset without using solvent accessibility, pH, and temperature. We use 36-fold cross validation to evaluate the scheme by training SVMs on 35 proteins and testing them on the remaining one. Thus, we empirically estimate how well the method can be generalized to unseen and nonhomologous proteins.

For the regression task, we use sequence or structure information without considering solvent accessibility, temperature, and pH. We remove identical mutations at identical sites with identical energy changes. The final dataset has 1539 data points. We refer to this dataset as SR1539.

Inputs and Encoding Schemes

Most previous methods, including the neural network approach in Capriotti et al.,[8] use tertiary structure information for the prediction of stability changes and in general do not use the local sequence context directly. To investigate the effectiveness of sequence-dependent and structure-dependent information, we use three encoding schemes: Sequence-Only (SO), Structure-Only (TO), and the combinations of sequence and structure (ST). All the schemes include the mutation information consisting of 20 inputs, which code for the 20 different amino acids. We set to -1 the input corresponding to the deleted residue and to 1 the newly introduced residue; all other inputs are set to 0.[8]

The SO scheme encodes the residues in a window centered on the target residue. We investigate how window size affects prediction performance. A range of window sizes work well for this task, however, we chose to use windows of size 7 because this is the smallest size which produces accurate results. As more data becomes available, a larger window may become helpful. Since the target residue is already encoded in the mutation information, the SO scheme only needs to encode three neighboring residues on each side of the target residue. 20 inputs are used to encode the residue type at each position. So the total input size of the SO scheme is 140 ($6 \times 20 + 20$). The TO scheme uses 20 inputs to encode the three-dimensional environment of the target residue. Each input corresponds to the frequency of each type of residue within a sphere of 9 Å radius around the target mutated residue. The cut-off distance threshold of 9 Å between C_α atoms worked best in the previous study.[8] So the TO encoding scheme has 40 ($20 + 20$) inputs. The ST scheme containing both sequence and structure information in SO and TO scheme has 160 inputs ($6 \times 20 + 20 + 20$).

On the SR1496 dataset, two extra inputs (temperature and pH) are used with the SO scheme; three extra inputs (solvent accessibility, temperature, and pH) are used with the TO and ST schemes. These additional inputs are not used for all other experiments on the SR1135, SR1023, and SR1539 datasets.

Prediction of Stability Changes Using Support Vector Machines

From a classification standpoint, the objective is to predict whether a mutation increases or decreases the stability of a protein, without concern for the magnitude of the energy change, as in Capriotti et al.[8] From a regression perspective, the objective is to predict the actual value of $\Delta\Delta G$. Here we apply SVMs[47] (see Burges[48] and Smola and Scholkopf[49] for tutorials on SVMs) to the stability classification and regression problems.

SVMs provide nonlinear function approximations by nonlinearly mapping input vectors into feature spaces and using linear methods for regression or classification in feature space[47,50–52] (Figs. 1, 2). Thus SVMs, and more generally kernel methods, combine the advantages of linear and nonlinear methods by first embedding the data into a feature space equipped with a dot product and then using linear methods in feature space to perform classification or regression tasks based on the dot product between data points. One important feature of SVMs is that computational complexity is reduced because data points do not have to be explicitly mapped into the feature space. Instead SVMs use a kernel function, $K(x,y) = \Phi(x) \times \Phi(y)$ to calculate the dot product of $\Phi(x)$ and $\Phi(y)$ implicitly, where x and y are input data points, $\Phi(x)$ and $\Phi(y)$ are the corresponding data vectors in feature space, and Φ is the map from input space to feature space. The linear classifi-

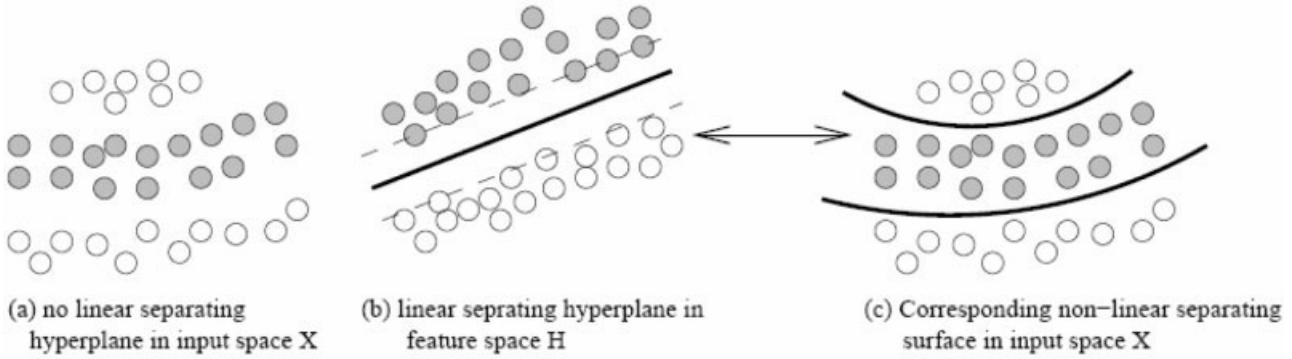

Fig. 1. Classification with SVMs. **a:** The negative and positive examples (white and grey circles) cannot be separated with a line in the input space X. **b:** Instead of looking for a separating hyperplane (thick line) directly in the input space, SVMs map training data points implicitly into a feature space H through a function Ø, so that the mapped points become separable by a hyperplane in the feature space. **c:** This hyperplane corresponds to a nonlinear complex surface in the original input space. The two dashed lines in the feature space correspond to the boundaries of the positive and negative examples respectively. The distance between these lines is the margin of the SVM.

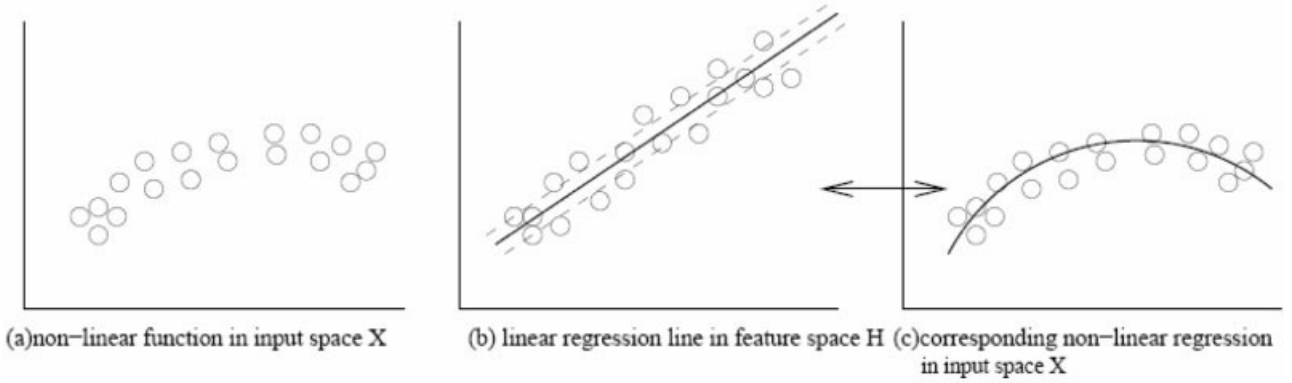

Fig. 2. Regression with SVMs. **a:** The data points cannot be fit with a line in the input space X. **b:** SVMs map data points implicitly into a feature space H through a function Ø, so that the mapped points can be fit by a line in the feature space. **c:** This line corresponds to a nonlinear regression curve in the original input space. The two virtual lines centered on the regression line in feature space form a regression tube with width 2e.

cation or regression function can be computed from the Gram matrix of kernel values between all training points.

Given a set of data points S (S^+ denotes the subset of positive training data points ($\Delta\Delta G > 0$) and S^- denotes the subset of negative training data points ($\Delta\Delta G < 0$), based on structure risk minimization theory,[47,50–52] SVMs learn a classification function $f(x)$ in the form of

$$f(x) = \sum_{x_i \varepsilon S^+} \alpha_i K(x,x_i) - \sum_{x_i \varepsilon S^-} \alpha_i K(x,x_i) + b \qquad (1)$$

$$f(x) = \sum_{x_i \varepsilon S} (\alpha_i - \alpha_i{}^*) K(x,x_i) + b \qquad (2)$$

where α_i or $\alpha_i{}^*$ are non-negative weights assigned to the training data point x_i during training by minimizing a quadratic objective function and b is the bias term. K is the kernel function, which can be viewed as a function for computing the similarity between two data points. Thus the function $f(x)$ can be viewed as a weighted linear combination of similarities between the training data points x_i and the target data point x. Only data points with positive weight α in the training dataset affect the final solution—these are called the support vectors. For classification problems, a new data point x is predicted to be positive ($\Delta\Delta G > 0$) or negative ($\Delta\Delta G < 0$) by taking the sign of $f(x)$. For regression, $f(x)$ is the predicted value of $\Delta\Delta G$.

We use SVM-light (http://svmlight.joachims.org)[37,38] to train and test our methods. We experimented with several common kernels including linear kernel, Gaussian radial basis kernel (RBF), polynomial kernel, and sigmoid kernel. In our experience, the RBF kernel [$\exp(-\gamma \mid x - y \mid^2)$] works best for mutation stability prediction. Using the RBF kernel, $f(x)$ is a weighted sum of Gaussians centered on the support vectors. Almost any separating boundary or regression function can be obtained with this kernel,[53] thus it is important to tune the parameters of SVMs to achieve good generalization performance and avoid overfitting. We adjust three critical parameters in both classification and regression. For both tasks, we adjust the width parameter γ of the RBF kernel and the regularization parameter C. γ is the inverse of the variance of the RBF and controls how peaked are the Gaussians centered on the support vectors. The bigger is γ, the more peaked are the Gaussians, and the more complex are the resulting decision boundaries.[53] C is the maximum value that the weights α can have. C controls the trade-off between

TABLE I. Results (Correlation Coefficient, Accuracy, Specificity, Sensitivity of Both Positive and Negative Examples) on the SR1496 Dataset†

Method	Correlation Coefficient	Accuracy	Sensitivity (+)	Specificity (+)	Sensitivity (−)	Specificity (−)
SO	0.59	0.841	0.711	0.693	0.897	0.888
TO	0.60	0.845	0.711	0.712	0.895	0.895
ST	0.60	0.847	0.671	0.733	0.910	0.883
NeuralNet*	0.49	0.810	0.520	0.710	0.910	0.830

†The last row (NeuralNet*) is the current best results reported in Capriotti et al.[8]

training error and the smoothness of $f(x)$ (particularly, the margin for classification).[47,50–52] A larger C corresponds to less training errors and a more complex (less smooth) function $f(x)$ which can overfit training data.

For classification, the ratio of penalty of training error between positive examples and negative examples, is another parameter that we tune. A cost >1 penalizes training error of positive examples more than that of negative examples. For regression, the width of the regression tube (ε) which controls the sensitivity of the cost associated with training errors [$f(x) - \Delta\Delta G$], needs to be tuned as well. The training error within range [$-\varepsilon$, $+\varepsilon$] does not affect the regression function.

The three parameters for each task (penalty ratio, γ, and C for classification; tube width, γ, and C for regression) are optimized on the training data. For each cross-validation fold, we optimize these parameters using the LOOCV (leave-one-out cross validation) procedure. Under the LOOCV procedure, for a training dataset with N data points, in each round, one data point is held out and the model is trained on the remaining $N - 1$ data points. Then the model is tested on the held-out data point. This process is repeated N times until all data points are tested once and the overall accuracy is computed for the training dataset.

For all the parameter sets we tested, we choose a set of parameters with the best accuracy to build the model on the training dataset; and then it is blindly tested on the testing dataset. A set of good parameters for classification on the SR1496 dataset is (penalty ratio = 1, $\gamma = 0.1$, $C = 5$) for the SO scheme, (penalty ratio = 2, $\gamma = 0.1$, $C = 5$) for the TO schemes, and (penalty ratio = 2, $\gamma = 0.1$, $C = 5$) for the ST scheme. A set of good parameters on the SR1135 dataset is (penalty ratio = 1, $\gamma = 0.05$, $C = 2$) for the SO scheme, (penalty ratio = 1, $\gamma = 0.05$, $C = 4$) for the TO scheme, and (penalty ratio = 1, $\gamma = 0.06$, $C = 0.5$) for the ST scheme. For the regression task, a set of good parameters for all schemes is (tube width = 0.1, $\gamma = 0.1$, $C = 5$).

RESULTS AND DISCUSSION

For classification, we use a variety of standard measures to evaluate the prediction performance of our method and compare it with previous methods. In the following equations, TP, FP, TN, and FN refer to the number of true positives, false positives, true negatives, and false negatives respectively. The measures we use include correlation coefficient {[(TP × TN) − (FP × FN)]/sqrt[(TP + FN) × (TP + FP) × (TN + FN) × (TN + FP)]}, accuracy [(TN + TP)/(TN + TP + FN + FP)], specificity [TP/(TP + FP)] and sensitivity [TP/(TP + FN)] of positive examples, and specificity [TN/(TN + FN)] and sensitivity [TN/(TN + FP)] of negative examples.

Table I reports the classification performance of three schemes on the SR1496 dataset. The results show that the performance of all three schemes is improved over the neural network approach[8] using most measures, even though we use a redundancy-reduced dataset instead of the S1615 dataset. (On the original S1615 dataset, the accuracy is about 85–86% for all three schemes). For instance, on average, the accuracy is improved by 3% to about 84%, and the correlation coefficient is increased by 0.1. The sensitivity of positive examples is improved by more than 10% using these three schemes, while the specificity of positive examples is very similar. The sensitivity of negative examples using the SO and TO schemes is slightly worse than for the neural network approach, but the specificity of negative examples is improved by more than 5% over the neural network approach. The accuracy of the SO scheme is slightly lower than that of the TO and ST schemes.

Following the same comparison scheme, we compare our methods with energy-based methods[9,23,24,39] available on the web and with the neural network method[8] in the classification context on the S388 dataset. We compare the predictions of the following methods: FOLDX,[9] DFIRE,[24] and PoPMuSiC,[23,39] and NeuralNet.[8] In Table II, we show the results obtained with the three schemes (SO, TO, ST) and the four external predictors on the S388 dataset, where results for the energy function based methods are taken from Capriotti et al.[8] The results show that our method, using the three encoding schemes for this specific task, performs similarly to, or better than, all other methods using most evaluation measures. For instance, the correlation coefficient of our method is better than all other methods, while the accuracy is better than DFIRE, FOLDX, and PoPMuSiC, but slightly worse than NeuralNet. FOLDX and DFIRE have relatively higher sensitivity but lower specificity on positive examples than other methods.

Table III reports the results on the SR1135 dataset without any site-specific redundancy. All the schemes do not use solvent accessibility, pH, and temperature. The results show that the accuracy of the structure-dependent schemes (TO and ST) are about 1% higher than the sequence-dependent scheme (SO). Specifically, the correlation coefficient of the TO scheme is significantly higher than the SO scheme. But the accuracy of the SO scheme is still very close to the accuracy derived using tertiary

TABLE II. Results on the S388 Dataset

Method	Correlation Coefficient	Accuracy	Sensitivity (+)	Specificity (+)	Sensitivity (−)	Specificity (−)
FOLDX	0.25	0.75	0.56	0.26	0.78	0.93
DFIRE	0.11	0.68	0.44	0.18	0.71	0.90
PoPMuSic	0.20	0.85	0.25	0.33	0.93	0.90
NeuralNet	0.25	0.87	0.21	0.44	0.96	0.90
SO	0.26	0.86	0.30	0.40	0.94	0.90
TO	0.28	0.86	0.31	0.42	0.94	0.91
ST	0.27	0.86	0.31	0.40	0.93	0.91

TABLE III. Results on the SR1135 Dataset

Method	Correlation Coefficient	Accuracy	Sensitivity (+)	Specificity (+)	Sensitivity (−)	Specificity (−)
SO	0.31	0.78	0.28	0.64	0.95	0.80
TO	0.39	0.79	0.46	0.60	0.90	0.83
ST	0.34	0.79	0.29	0.71	0.97	0.80

TABLE IV. Specificity and Sensitivity of the SO Scheme for Helix, Strand, and Coil on the SR1135 Dataset

Secondary structure	Sensitivity (+)	Specificity (+)	Sensitivity (−)	Specificity (−)
Helix	0.31	0.67	0.94	0.79
Strand	0.16	0.48	0.97	0.84
Coil	0.30	0.68	0.95	0.79

TABLE V. Results on the SR1023 Dataset Using the SO Scheme

Method	Correlation Coefficient	Accuracy	Sensitivity (+)	Specificity (+)	Sensitivity (−)	Specificity (−)
SO	0.13	0.74	0.15	0.42	0.93	0.77

structure information. This is probably due to two reasons. First, the sequence window contains a significant amount of information related to the prediction of mutation stability. Second, the method for encoding structural information in the TO and ST schemes is not optimal for the task and does not capture all the structural information that is relevant for protein stability. On this redundancy-reduced dataset, we also compare the accuracy according to the type of secondary structure encountered at the mutation sites. The secondary structure is assigned by the DSSP program.[54] Table IV reports the specificity and sensitivity for both positive and negative examples according to three types of secondary structure (helix, strand, and coil) using the SO scheme. The SO scheme achieves similar performance on helix and coil mutations. Sensitivity and specificity for positive examples on β-strands, however, is significantly lower. This is probably due to the long-range interactions between β-strands.

Table V reports the results of the SO scheme on the SR1023 dataset after removing both the homologous proteins and site-specific redundancy. The overall accuracy is 74%. Not surprisingly, the accuracy is lower than the accuracy obtained when mutations on homologous or identical proteins are included in the training and test dataset. The sensitivity and specificity of the positive examples drop significantly. This indicates that the accuracy of the method depends on having seen mutations on similar or identical proteins in the training dataset. The results show that the prediction of mutation stability on unseen and nonhomologous proteins remains very challenging.

TABLE VI. Results (Correlation Between Predicted Energy and Experimental Energy, and Standard Error) on the SR1539 Dataset Using SVM Regression

Scheme	SO	TO	ST
Correlation	0.75	0.76	0.75
STD	1.10	1.09	1.09

The performance of SVM regression is evaluated using the correlation between the predicted energy and experimental energy, and the standard error (std or root-mean-square error) of the predictions. Table VI shows the performance of the direct prediction of $\Delta\Delta G$ using SVM regression with the three encoding schemes. The three schemes have similar performance. The TO scheme performs slightly better with a correlation of 0.76, and std of 1.09. Figure 3 shows the scatter plots of predicted energy versus experimental energy using the SO and TO schemes. Overall, the results show that our method effectively uses sequence information to predict energy changes associated with single amino acid substitutions both in regression and classification tasks.

CONCLUSIONS

In this study, we have used support vector machines to predict protein stability changes for single-site mutations.

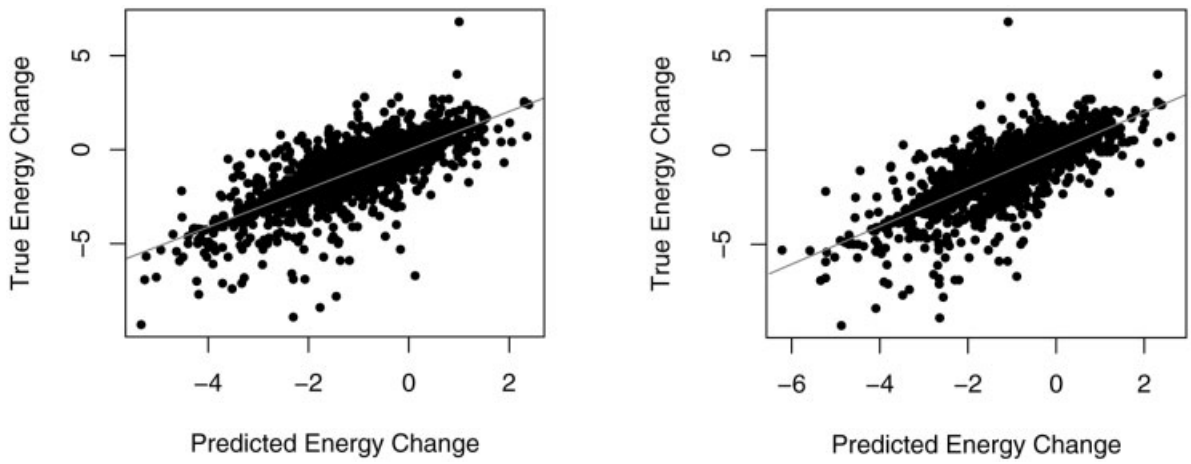

Fig. 3. The experimentally measured energy changes versus the predicted energy changes using SVM regression with the SO scheme on the SR1539 dataset (left). The correlation is 0.75. The std is 1.10. The slope of the red regression line is 1.03. The experimentally measured energy changes versus the predicted energy changes using SVM regression with the TO scheme on the SR1539 dataset (right). The correlation is 0.76. The std is 1.09. The slope of the red regression line is 1.01.

Our method consistently shows better performance than previous methods evaluated on the same datasets. We demonstrate that sequence information can be used to effectively predict protein stability changes for single site mutations. Our experimental results show that the prediction accuracy based on sequence information alone is close to the accuracy of methods that depend on tertiary structure information. This overcomes one important shortcoming of previous approaches that require tertiary structures to make accurate predictions. Thus, our approach can be used on a genomic scale to predict the stability changes for large numbers of proteins with unknown tertiary structures.

ACKNOWLEDGMENTS

Work supported by an NIH Biomedical Informatics Training grant (LM-07443-01), an NSF MRI grant (EIA-0321390), a grant from the University of California Systemwide Biotechnology Research and Education Program (UC BREP) to P.B., and by the Institute for Genomics and Bioinformatics at UCI.

REFERENCES

1. Dahiyat BI. In silico design for protein stabilization. Curr Opin Biotech 1999;10:387–390.
2. DeGrado WF. De novo design and structural characterization of proteins and metalloproteins. Ann Rev Biochem 1999;68:779–819.
3. Street AG, Mayo SL. Computational protein design. Struct Fold Des 1999;7:R105–R109.
4. Saven J. Combinatorial protein design. Curr Opin Struct Biol 2002;12:453–458.
5. Mendes J, Guerois R, Serrano L. Energy estimation in protein design. Curr Opin Struct Biol 2002;12:441–446.
6. Bolon DN, Marcus JS, Ross SA, Mayo SL. Prudent modeling of core polar residues in computational protein design. J Mol Biol 2003;329:611–622.
7. Looger LL, Dwyer MA, Smith JJ, Hellinga HW. Computational design of receptor and sensor proteins with novel functions. Nature 2003;423:185–190.
8. Capriotti E, Fariselli P, Casadio R. A neural network-based method for predicting protein stability changes upon single point mutations. In: Proceedings of the 2004 conference on intelligent systems for molecular biology (ISMB04), Bioinformatics (Suppl. 1), volume 20. New York: Oxford University Press; 2004. p 190–201.
9. Guerois R, Nielsen JE, Serrano L. Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. J Mol Biol 2002;320:369–387.
10. Bash PA, Singh UC, Langridge R, Kollman PA. Free-energy calculations by computer simulations. Science 1987;256:564–568.
11. Dang LX, Merz KM, Kollman PA. Free-energy calculatons on protein stability: Thr-157 val-157 mutation of t4 lysozyme. J Am Chem Soc 1989;111:8505–8508.
12. Prevost M, Wodak SJ, Tidor B, Karplus M. Contribution of the hydrophobic effect to protein stability: analysis based on simulations of the ile-96-ala mutation in barnsase. Proc Natl Acad Sci USA 1991;88:10880–10884.
13. Tidor B, Karplus M. Simulation analysis of the stability mutant r96h of t4 lysozyme. Biochemistry 1991;30:3217–3228.
14. Lee C, Levitt M. Accurate prediction of the stability and activity effects of site-directed mutagenesis on a protein core. Nature 1991;352:448–451.
15. Miyazawa S, Jernigan RL. Protein stability for single substitution mutants and the extent of local compactness in the denatured state. Protein Eng 1994;7:1209–1220.
16. Lee C. Testing homology modeling on mutant proteins: predicting structural and thermodynamic effects in the ala98-val mutants of t4 lysozyme. Fold Des 1995;1:1–12.
17. Pitera JW, Kollman PA. Exhaustive mutagenesis in silico: multicoordinate free energy calculations on proteins and peptides. Proteins 2000;41:385–397.
18. Sippl MJ. Knowledge-based potentials for proteins. Curr Opin Struct Biol 1995;5:229–235.
19. Gillis D, Rooman M. Predicting protein stability changes upon mutation using database-derived potentials: solvent accessibility determines the importance of local versus non-local interactions along the sequence. J Mol Biol 1997;272:276–290.
20. Topham CM, Srinivasan N, Blundell TL. Prediction of the stability of protein mutants based on structural environment-dependent amino acid substitution and propensity tables. Protein Eng 1997;101:46–50.
21. Gillis D, Rooman M. Prediction of stability changes upon single-site mutations using database-derived potentials. Theor Chem Acc 1999;101:46–50.
22. Carter CW, LeFebvre BC, Cammer SA, Torpsha A, Edgell MH. Four-body potentials reveal protein-specific correlations to stability changes caused by hydrophobic core mutations. J Mol Biol 2001;311:625–638.
23. Kwasigroch JM, Gillis D, Dehouck Y, Rooman M. Popmusic,

rationally designing point mutations in protein structures. Bioinformatics 2002;18:1701–1702.
24. Zhou H, Zhou Y. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. Protein Sci 2002;11:2714–2726.
25. Zhou H, Zhou Y. Quantifying the effect of burial of amino acid residues on protein stability. Proteins 2004;54:315–322.
26. Lazaridis T, Karplus M. Effective energy functions for protein structure prediction. Curr Opin Struct Biol 2000;10:139–145.
27. Villegas V, Viguera AR, Aviles FX, Serrano L. Stabilization of proteins by rational design of alpha-helix stability using helix/coil transition theory. Fold Des 1996;1:29–34.
28. Munoz V, Serrano L. Development of the multiple sequence approximation within the AGADIR model of alpha-helix formation: comparison with zimm-bragg and lifson-roig formalisms. Biopolymers 1997;41:495–509.
29. Lacroix E, Viguera AR, Serrano L. Elucidating the folding problem of alpha-helices: local motifs, long-range electrostatics, ionic-strength dependence and prediction of NMR parameters. J Mol Biol 1998;284:173–191.
30. Takano K, Ota M, Ogasahara K, Yamagata Y, Nishikawa K, Yutani K. Experimental verification of the stability profile of mutant protein(spmp) data using mutant human lysozymes. Protein Eng 1999;12:663–672.
31. Domingues H, Peters J, Schneider KH, Apeler H, Sebald W, Oschkinat H, Serrano L. Improving the refolding yield of interleukin-4 through the optimization of local interactions. J Biotechnol 2000;84:217–230.
32. Taddei N, Chiti F, Fiaschi T, Bucciantini M, Capanni C, Stefani M. Stabilization of alpha-helices by site-directed mutagenesis reveals the importance of secondary structure in the transition state for acylphosphatase folding. J Mol Biol 2000;300:633–647.
33. Funahashi J, Takano K, Yutani K. Are the parameters of various stabilization factors estimated from mutant human lysozymes compatible with other proteins? Protein Eng 2001;14:127–134.
34. Bordner AJ, Abagyan RA. Large-scale prediction of protein geometry and stability changes for arbitrary single point mutations. Proteins 2004;57:400–413.
35. Casadio R, Compiani M, Fariselli P, Viarelli F. Predicting free energy contributions to the conformational stability of folded proteins from the residue sequence with radial basis function networks. In Proc Int Conf Intell Syst Mol Biol, volume 3, p 81–88. 1995.
36. Frenz C. Neural network-based prediction of mutation-induced protein stability changes in staphylococcal nuclease at 20 residue positions. Proteins 2005;59:147–151.
37. Joachims T. Making large-scale SVM learning practical. Advances in Kernel Methods—Support Vector Learning, In: Schölkopf B, Burges C, Smola A, editors. Cambridge, MA: MIT Press; 1999.
38. Joachims T. Learning to classify text using support vector machines. Dissertation. Springer; 2002.
39. Gillis D, Rooman M. Stability changes upon mutation of solvent-accessible residues in proteins evaluated by database-derived potentials. J Mol Biol 1996;257:1112–1126.
40. Rost B, Sander C. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. Proc Natl Acad Sci USA 1993;90:7558–7562.
41. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 1999;292:195–202.
42. Pollastri G, Przybylski D, Rost B, Baldi P. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. Proteins 2001;47:228–235.
43. Pollastri G, Baldi P, Fariselli P, Casadio R. Prediction of coordination number and relative solvent accessibility in proteins. Proteins 2001;47:142–153.
44. Gromiha M, An J, Kono H, Oobatake M, Uedaira H, Prabakaran P, Sarai A. Protherm, version 2.0: thermodynamic database for proteins and mutants. Nucleic Acids Res 2000;28:283–285.
45. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, et al. The protein data bank. Nucleic Acids Res 2000;28:235–242.
46. Mika S, Rost B. Uniqueprot: creating representative protein-sequence sets. Nucleic Acids Res 2003;31:3789–3791.
47. Vapnik V. Statistical learning theory. New York: Wiley; 1998.
48. Burges C. A tutorial on support vector machines for pattern recognition. Knowledge Discovery and Data Mining 1998;2.
49. Smola A, Scholkopf B. A tutorial on support vector regression. In: NeuroCOLT Technical Report NC-TR-98-030. Royal Holloway College, University of London, UK, 1998.
50. Vapnik V. The nature of statistical learning theory. Berlin, Germany: Springer-Verlag, 1995.
51. Drucker H, Burges CJC, Kaufman L, Smola A, Vapnick V. Support vector regression machines. In Petsche T, Mozer MC, Jordan MI, editor. Advances in neural information processing systems, volume 9. Cambridge, MA; MIT Press, 1997. p 155–161.
52. Scholkopf B, Smola AJ. Learning with kernels: support vector machines, regularization, optimization, and beyond. Cambridge, MA: MIT Press; 2002.
53. Vert J, Tsuda K, Scholkopf B. A primer on kernel methods. In Vert J, Scholkopf B, Tsuda K, editor. Kernel methods in computational biology. Cambridge, MA: MIT press; 2004. p 55–72.
54. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features. Biopolymers 1983;22:2577–2637.