SHORT COMMUNICATION

AUTO-MUTE: web-based tools for predicting stability changes in proteins due to single amino acid replacements

Majid Masso1 and Iosif I. Vaisman

Laboratory for Structural Bioinformatics, Department of Bioinformatics and Computational Biology, George Mason University, 10900 University Blvd. MS 5B3, Manassas, VA 20110, USA

1To whom correspondence should be addressed.
E-mail: mmasso@gmu.edu

Received December 17, 2009; revised April 28, 2010; accepted May 30, 2010
Edited by Joost Schymkowitz

Utilizing cutting-edge supervised classification and regression algorithms, three web-based tools have been developed for predicting stability changes upon single residue substitutions in proteins with known native structures. Trained models classify independent mutant test sets with accuracies ranging from 87 to 94%. Attributes representing each mutant protein are based on a computational mutagenesis methodology relying on a four-body statistical potential, illustrating a novel integration of both energy-based and machine learning approaches. The servers are written in PHP and hosted on a Linux platform, and they can be freely accessed online along with detailed data sets, documentation and performance results at http://proteins.gmu.edu/automute.

Keywords: knowledge-based statistical potential/prediction/protein design/stability changes/structure-based computational mutagenesis

Introduction

One of the most effective methods that researchers utilize for comprehensive protein structure and function analysis entails experimental studies on single amino acid residue replacements in proteins. The relative impact of such mutations on stability offers insights into the mechanism of protein folding, identifies possible structural or functional roles of residues, and provides useful information for the design of new proteins with desired thermodynamic and physicochemical characteristics (Lehmann et al., 2000; Vieille and Zeikus, 2001; Yang et al., 2004). However, the number of such studies that can be performed is limited due to time and cost constraints. Thus, reliable in silico models for predicting stability change upon mutation are in great demand, at the very least for the purpose of prioritizing these experiments. Energy-based methods include those that employ physical, statistical and empirical potentials. Physical potentials rely on first principles to model atomic force fields, making them computationally too demanding for practical large-scale applications, while statistical potentials are significantly faster yet achieve comparable predictive accuracy (Lazaridis and Karplus, 2000; Guerois et al., 2002). More recent machine learning approaches, which learn complex nonlinear functions with independent variables representing mutant protein sequence and/or structure information, are both robust and similarly capable of rapidly generating large numbers of reliable stability change predictions (Capriotti et al., 2004; Cheng et al., 2006).

Several energy-based methods relying on statistical and empirical potentials are presently available for the computational prediction of stability changes due to single residue substitutions in proteins. PoPMuSiC (Gulis and Rooman, 1996; Gilis and Rooman, 1997; Gilis and Rooman, 2000; Kwasiagrob et al., 2002) predicts values of $\Delta G$ free energy stability changes upon mutation by utilizing different combinations of database-derived torsion and amino acid pair distance potentials based on the solvent accessibility of the mutated position. DFIRE (Zhou and Zhou, 2002; Zhou et al., 2005) uses a distance-dependent, residue-specific, all-atom, knowledge-based pair potential for $\Delta G$ prediction. CUPSAT (Parthiban et al., 2006; Parthiban et al., 2007) relies on knowledge-based potentials, specifically protein structural environment specific atom pair distance potentials as well as torsion angle potentials, for predicting $\Delta G$ values. FoldX (Guerois et al., 2002; Schymkowitz et al., 2005) predicts values of $\Delta G$ by using full atomic descriptions of protein structures, where the linear combination of terms forming the defined energy functions are weighted using empirical data obtained from protein engineering experiments.

In recent years, machine learning methods have also led to the development of several reliable predictive models of mutant stability change. I-Mutant (Capriotti et al., 2004) implements a neural network algorithm for predicting the sign of $\Delta G$ based on the protein sequence and I-Mutant2.0 (Capriotti et al., 2005a,b) implements support vector machine (SVM) and support vector regression (SVR) algorithms for predicting the sign and actual value of $\Delta G$, respectively, based on either the protein sequence or structure. MUPRO (Cheng et al., 2005; Cheng et al., 2006) also implements SVM for predicting only the sign of $\Delta G$ based on the protein sequence. iPTREE-STAB (Huang et al., 2007) predicts the sign of $\Delta G$ by coupling the C4.5 decision tree algorithm (Quinlan, 1993) with adaptive boosting (Freund and Schapire, 1996), and the value of $\Delta G$ by using the classification and regression tree (CART) algorithm (Breiman et al., 1984), both based on the protein sequence. The models in all cases are trained using data sets of mutants with experimentally known $\Delta G$ values collected from the ProTherm database (Bava et al., 2004). For sequence-based
prediction, each mutant is represented by a feature vector with input attributes (independent variables) that encode the identities of the native and new amino acids at the mutated position, a count of the 20 amino acid types at all other positions within a fixed-width residue window (19 for I-Mutant and I-Mutant2.0, 7 for MUpro and iPTREE-STAB) centered at the mutated position, and the pH and temperature of the experimental conditions. For I-Mutant2.0 structure-based prediction, among the input attributes also included are counts of all 20 amino acid types found within the three-dimensional (3D) environment defined by a sphere of radius 9 Å centered at the mutated position, and a final attribute encodes the relative solvent accessible area (RSA). The output attribute (dependent variable) in all cases is either categorical (sign of $\Delta \Delta G$) or continuous (actual value of $\Delta \Delta G$). Although the methods all perform reasonably well, more so for predicting the direction of stability change than the actual value, there still exists significant room for improvement (Potapov et al., 2009).

More generally, a variety of experimental techniques are used for measuring stability changes in proteins upon mutation, and values reported in the literature include free energy change of unfolding due to thermal ($\Delta \Delta G = \Delta G_{\text{mutant}} - \Delta G_{\text{native}}$) or denaturant ($\Delta \Delta G_{\text{H}_2\text{O}} = \Delta G_{\text{mutant}} - \Delta G_{\text{native}}$) denaturation in units of kcal/mol, as well as mutant thermal stability change ($\Delta T_m = \Delta T_{m,\text{mutant}} - \Delta T_{m,\text{native}}$) in units of °C (Bava et al., 2004). In a previous manuscript focusing on methodology, we described a novel approach combining both machine learning and energy-based methods to develop more accurate $\Delta \Delta G$, $\Delta \Delta G_{\text{H}_2\text{O}}$, and $\Delta T_m$ stability change models (Masso and Vaisman, 2008). This integration of two techniques has since been emulated successfully for the development of an improved PoPMuSiC-2.0 (Dehouck et al., 2009) as well as for the prediction of hot-spot residues at protein-protein interfaces (Lise et al., 2009). For our models, each feature vector representing a mutant includes input attributes obtained by applying a computational mutagenesis that utilizes a four-body, knowledge-based, statistical contact potential. This amino acid distance potential applies the inverse Boltzmann principle (Sippl, 1995) and is derived via Delaunay tessellation of protein structures, a classical computational geometry (CG) tiling technique that objectively identifies quadruplets of 3D nearest neighbor residues (Singh et al., 1996). With each of the three online tools we developed ($\Delta \Delta G$, $\Delta \Delta G_{\text{H}_2\text{O}}$ and $\Delta T_m$ stability changes), four distinct models are available for obtaining predictions. Trained random forest (RF) and SVM classifiers have been implemented for predicting the sign of $\Delta \Delta G$ and $\Delta \Delta G_{\text{H}_2\text{O}}$ (adaptively boosted C4.5 decision tree and SVM classifiers for $\Delta T_m$), while tree regression (REPTree) and SVR models are used to predict actual stability change values on all three servers. Details regarding algorithm descriptions, including parameter values selected for model training, were previously reported in our related methodology paper (Masso and Vaisman, 2008). We implemented all algorithms through use of the Weka software package (Frank et al., 2004).

**Methods**

Over 1400 diverse protein structures were selected from the Protein Data Bank (PDB) (Berman et al., 2000) and individually tessellated, by first abstracting each residue to a point in 3D using the Cα coordinate, in order to develop the four-body statistical potential. Delaunay tessellation utilizes as vertices the collection of Cα coordinates associated with a given structure to generate a convex hull of space-filling, non-overlapping, tetrahedral simplices (Fig. 1). The vertices of simplices in a protein structure tessellation objectively identify quadruplets of nearest neighbor residues in 3D Euclidean space (Singh et al., 1996). We further impose an upper limit of 12 Å on allowable tetrahedral edge lengths to ensure biochemically interacting quadruplets identified by each of the simplices. For each of the 8855 distinct types of amino acid quadruplets that can be generated from the standard 20-letter alphabet, a relative frequency $f$ of occurrence of the quadruplet as nearest neighbors is calculated based upon the proportion of all tetrahedral simplices generated by the tessellations for which the Cα vertices represent the four particular residues. A multinomial reference distribution ($n = 4; k = 20$), based upon the relative frequencies of occurrence of the 20 residue types among all the amino acids collectively

![Fig. 1.](https://academic.oup.com/peds/article-abstract/23/8/683/1512499/1)
comprising the tessellated proteins, is used to calculate a rate of occurrence expected by chance \( p \) for each quadruplet. Modeled after the inverse Boltzmann principle, a knowledge-based empirical potential of quadruplet interaction is computed as \( \log(f_{jlp}) \) for each of the 8855 types of residue quadruplets, and collectively these log-likelihood scores define the four-body statistical potential (Singh et al., 1996). Downloadable text files, one listing the PDB accession codes of the protein data set and another providing the derived four-body statistical potential, are accessible as links within the documentation supplied on the main portal to the stability change server pages at http://proteins.gmu.edu/automute.

Based on this potential, the tessellation of any solved protein structure can be used to calculate a residue environment score for every position, by identifying all tetrahedral simplices sharing the \( \text{Co} \) of a given position as a vertex and summing the log-likelihood scores of the residue quadruplets that they respectively represent (Masso and Vaisman, 2007; Masso and Vaisman, 2008). Collectively, the vector whose components consist of the residue environment scores for all positions in a protein is referred to as a 3D-1D potential profile (Bowie et al., 1991; Masso and Vaisman, 2007). A subsequently formulated computational mutagenesis methodology entails replacing the residue identity at a particular \( \text{Co} \) vertex in the native protein structure tessellation and recalculating the environment scores for all positions, yielding a mutant 3D-1D potential profile. Taking the difference between the mutant and original 3D-1D potential profiles (i.e. subtracting the original environment score from the respective recalculated score at every position) quantifies the relative perturbations at all positions, referred to as environmental change (EC) scores, due to a single residue replacement (Masso and Vaisman, 2007; Masso and Vaisman, 2008). The methodology is limited to local effects, as only the mutated position and those participating in tetrahedral simplices with it exhibit altered environment scores and therefore have nonzero EC scores, while for all other positions, \( \text{EC}_i = 0 \) (Fig. 1).

Among the feature vector components that we use to represent each mutant protein are the EC score at the mutated position, as well as the EC scores of the six nearest positions (determined by lengths of tessellation edges to the mutated position) and ordered by 3D Euclidean distance. Additional input attributes consist of native and new residues at the mutated position, the ordered amino acid identities at the six nearest positions and the ordered differences in primary sequence numbers between those of the six neighbors and that of the mutated position. Next, we include CG-based input features of mean tetrahedrality (Singh et al., 1996) and mean volume of tetrahedral simplices that share the relabeled \( \text{Co} \) vertex (i.e. the mutated position), mutated position depth (surface—\( \text{Co} \) vertex participates in forming at least one triangular face of a tetrahedron not shared as a facet by any other tetrahedron; undersurface—not a surface position, but the \( \text{Co} \) vertex contacts a surface position through a tessellation edge; buried—neither a surface nor an undersurface position), and the number of edge contacts that the mutated position has with surface positions, all obtained from the protein structure tessellation and collectively taking place of an RSA (solvent accessibility) attribute that was used for developing the models explored in our earlier methodology article (Masso and Vaisman, 2008). The final mutant feature vector components are secondary structure (Helix, Strand, Coil, or Turn) at the mutated position, and in the case of the \( \Delta \Delta G \) or \( \Delta \Delta G^{\text{H2O}} \) predictors, temperature and pH of experimental conditions. Each mutant input feature vector is associated with an output attribute representing the experimental stability change, a real-valued variable employed with regression algorithms that is replaced with a categorical variable (‘decreased’/‘increased’ for negative/non-negative stability change values) in classification algorithms.

Results

An extensive documentation page accompanies each of the three stability change tools, accessible via a link labeled ‘details’ located on the input page of each server. Each documentation page includes a link to a downloadable text file of the respective training set, specifying all attribute values for each experimental mutant. For each of the four trained models available with each server, detailed cross-validation performance results are also tabulated on the pages. In particular, the data show that there is negligible difference with respect to overall model accuracy measures, but not necessarily individual mutant predictions, when the solvent accessibility (RSA) attribute is substituted with the four CG features; the latter attributes are used instead for training the models underpinning the online tools and serve to strengthen the novelty of our methodology. We initially incorporated RSA (as well as temperature and pH) to be consistent with previous approaches and to evaluate the theoretical benefits to model performance of additionally employing our energy-based features rather than the supplemental attributes used by other methods (Masso and Vaisman, 2008). By integrating energy-based attributes and machine learning tools, our models for predicting stability change frequently outperform by significant margins many of the machine learning tools described in the Introduction (Masso and Vaisman, 2008). On the other hand, our method is similar to others in that predictions for surface mutations are generally not as accurate as those for substitutions at buried positions (Gillis and Rooman, 2000; Capriotti et al., 2004; Capriotti et al., 2005a; Masso and Vaisman, 2008).

For illustration purposes, Tables I and II, respectively, summarize classification and regression 20-fold cross-validation performance results for the \( \Delta \Delta G \) mutant data set, obtained using our current CG attributes-based models, our prior RSA attribute-based models, and models developed by another research group. In the case of classification, mutants for which \( \Delta \Delta G \geq 0 \) (\( \Delta \Delta G < 0 \)) are labeled as ‘increased’ or ‘+’ (‘decreased’ or ‘−’) as described previously in the Methods. Given that TP (TN) = total number of correctly predicted mutants in the ‘increased’ or ‘+’ (‘decreased’ or ‘−’) class and FN (FP) = total number of, respectively, misclassified mutants, overall accuracy is defined as

\[
Q = \frac{TP + TN}{TP + TN + FP + FN}.
\]

Also, for the ‘increased’ class,

\[
S(+) = \text{sensitivity} = \frac{TP}{TP + FN} \quad \text{and} \quad P(+) = \text{precision} = \frac{TP}{TP + FP}.
\]
Finally, the balanced error rate is defined as

\[
\text{BER} = 0.5 \times \left( \frac{\text{FN}}{\text{FN} + \text{TP}} + \frac{\text{FP}}{\text{FP} + \text{TN}} \right),
\]

and Matthew’s correlation coefficient (MCC) is given by

\[
\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{[(\text{TP} + \text{FN})(\text{TP} + \text{FP})(\text{TN} + \text{FN})(\text{TN} + \text{FP})]^{1/2}}.
\]

In the case of regression, performance is evaluated by calculating the Pearson correlation coefficient (r) of the predicted and experimental \( \Delta \Delta G \) values, and the standard error is also reported. Lastly, Table III summarizes the results of \( \Delta \Delta G \) predictions made by trained classification models on an independent test set of 142 mutants, available as a downloadable text file from the \( \Delta \Delta G \) documentation page. Similar predictions on independent test sets of mutants were obtained with respect to models corresponding to the other two types of stability changes (tables of results and downloadable text files containing mutant test sets are available on their respective documentation pages), and the test set classification accuracy in all cases ranges between 87 and 94%.

### Required inputs and server outputs

Prediction of stability change upon mutation requires providing a PDB accession code for the corresponding native protein structure as well as a letter identifying the specific protein chain. Delaunay tessellation can only be performed on single chains that utilize the 20 standard amino acids and have consecutive, ungapped residue numbering in the ATOM lines of the respective PDB files, such that all amino acid \( \alpha \) coordinates are uniquely provided without any possessing parallel occupancy. Gaps in residue numbering are principally due to unresolved portions of the structure, although in rare cases researchers intentionally introduce gaps based on sequence alignment to a homologous reference protein. While it is possible to perform Delaunay tessellation on a protein chain with gaps (e.g. via a consecutive residue renumbering), the tetrahedral simplices generated would incorrectly identify nearest neighbor quadruplets due to the missing \( \alpha \) coordinates for structurally unresolved residues. Our computational mutagenesis and mutant attribute vectors rely on this data and subsequently both would be faulty, leading to unreliable predictions. In addition to protein chains from X-ray PDB structures that satisfy the above constraints, single chains from NMR structures can also be tessellated if the corresponding PDB files consist of single minimized average structures and exclude multiple models (i.e. unique \( \alpha \) coordinates for each residue). Finally, the residue to be mutated must possess at least six tessellation-based nearest neighbor residues (i.e. \( \alpha \) of the mutated position must be shared by a minimum of two adjacent tetrahedral simplices, in which case it must be the only shared vertex between them).

The input page for each tool requires a PDB code, chain identifier and the mutation (in the form [native residue][position number from PDB file ATOM lines][new residue], for example D25E, or D25_ to inquire about all 19 substitutions at a given position) as input for a prediction. On the \( \Delta \Delta G \) and \( \Delta \Delta G_{\text{H}2\text{O}} \) web servers, the temperature and pH for each mutation are also needed as inputs. Lastly, the user must make a single selection from among two classification and two regression models. The servers are written in PHP and hosted on a Linux platform (Fig. 2). For each requested mutant, structural tessellation is performed and computational mutagenesis is applied, from which a mutant input feature vector is generated and utilized by the model selected for making a prediction. Classification models predict stability change as either increased or decreased along with a confidence measure, while regression models predict the actual stability change value. The output page returns the input data, reveals the prediction and provides CG features (mean tetrahedrality and volume of simplices sharing the mutated position \( \alpha \) vertex, depth of mutated position and the number of surface residue contacts) and secondary structure at the mutated position, along with a glossary.

Taking into consideration the earlier discussion concerning the unreliability of predictions based on consecutive renumbering of gapped protein chains, an option for users to upload their own structure files is not provided with the current releases of the stability change servers. A future update may allow for submitted files, in order to make the

### Table I. \( \Delta \Delta G \) training set 20-fold cross-validation classification performance

<table>
<thead>
<tr>
<th>Method</th>
<th>Q</th>
<th>S(+)</th>
<th>P(+)</th>
<th>S(-)</th>
<th>P(-)</th>
<th>BER</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF (CG)</td>
<td>0.86</td>
<td>0.69</td>
<td>0.81</td>
<td>0.93</td>
<td>0.88</td>
<td>0.19</td>
<td>0.65</td>
</tr>
<tr>
<td>SVM (CG)</td>
<td>0.83</td>
<td>0.69</td>
<td>0.74</td>
<td>0.90</td>
<td>0.87</td>
<td>0.21</td>
<td>0.60</td>
</tr>
<tr>
<td>RF (RSA)</td>
<td>0.86</td>
<td>0.70</td>
<td>0.81</td>
<td>0.93</td>
<td>0.88</td>
<td>0.18</td>
<td>0.66</td>
</tr>
<tr>
<td>SVM (RSA)</td>
<td>0.84</td>
<td>0.70</td>
<td>0.75</td>
<td>0.90</td>
<td>0.87</td>
<td>0.20</td>
<td>0.61</td>
</tr>
<tr>
<td>Capriotti et al. (2005b) (SVM/RSA)</td>
<td>0.80</td>
<td>0.56</td>
<td>0.73</td>
<td>0.91</td>
<td>0.83</td>
<td>0.28</td>
<td>0.51</td>
</tr>
</tbody>
</table>

### Table II. \( \Delta \Delta G \) training set 20-fold cross-validation regression performance

<table>
<thead>
<tr>
<th>Method</th>
<th>r</th>
<th>Standard error (kcal/mol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPTree (CG)</td>
<td>0.79</td>
<td>1.1</td>
</tr>
<tr>
<td>SVR (CG)</td>
<td>0.76</td>
<td>1.2</td>
</tr>
<tr>
<td>REPTree (RSA)</td>
<td>0.79</td>
<td>1.1</td>
</tr>
<tr>
<td>SVM (RSA)</td>
<td>0.76</td>
<td>1.2</td>
</tr>
<tr>
<td>Capriotti et al. (2005b) (SVR/RSA)</td>
<td>0.71</td>
<td>1.3</td>
</tr>
</tbody>
</table>
services available both to researchers with new, unreleased PDB structures as well as to users interested in modifying and using structure files with artificial gaps as described earlier, with a caveat to individuals considering a consecutive renumbering and uploading of PDB files containing bona fide structural gaps.

Acknowledgements

T4 lysozyme ribbon diagram was produced using the UCSF Chimera package and tessellation visualizations were produced using Matlab, Version 7.0.1.24704 (R14) Service Pack 1.

References


