Protein Engineering, Design & Selection vol. 22 no. 9 pp. 553–560, 2009
Published online June 26, 2009 doi:10.1093/protein/gzp030

Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details

Vladimir Potapov[1], Mati Cohen[1] and Gideon Schreiber[2]

Department of Biological Chemistry, Weizmann Institute of Science, Rehovot, Israel

[2]To whom correspondence should be addressed.
E-mail: gideon.schreiber@weizmann.ac.il

Methods for protein modeling and design advanced rapidly in recent years. At the heart of these computational methods is an energy function that calculates the free energy of the system. Many of these functions were also developed to estimate the consequence of mutation on protein stability or binding affinity. In the current study, we chose six different methods that were previously reported as being able to predict the change in protein stability ($\Delta\Delta G$) upon mutation: CC/PBSA, EGAD, FoldX, I-Mutant2.0, Rosetta and Hunter. We evaluated their performance on a large set of 2156 single mutations, avoiding for each program the mutations used for training. The correlation coefficients between experimental and predicted $\Delta\Delta G$ values were in the range of 0.59 for the best and 0.26 for the worst performing method. All the tested computational methods showed a correct trend in their predictions, but failed in providing the precise values. This is not due to lack in precision of the experimental data, which showed a correlation coefficient of 0.86 between different measurements. Combining the methods did not significantly improve prediction accuracy compared to a single method. These results suggest that there is still room for improvement, which is crucial if we want forcefields to perform better in their various tasks.

Keywords: computational protein design/energy functions/estimating protein stability/protein engineering

Introduction

Proteins are the working horses of the cellular machinery. Protein structure and function are mutually interdependent (Fersht, 1999). The same chemical and physical forces govern structure formation, their molecular interactions and enzymatic activities (Cohen *et al*., 2008). The major driving forces for protein folding are the burial of hydrophobic groups, formation of optimal non-covalent interactions and maximization of the entropy (Pickett and Sternberg, 1993; Fersht, 1999). The final structure of the monomer or complex is a result of a subtle balance between the entropic effects and different stabilizing interactions (enthalpy) (Fersht, 1999).

Solving the three-dimensional structure of proteins (using NMR or X-ray crystallography) provides atomic details on their architecture, but not on the forces stabilizing them. These are studied by introducing mutations and measuring their energetic consequence (Matthews, 1993). Most of the mutations are either neutral or destabilizing proteins; however, many stabilizing mutations have been found as well (Serrano *et al*., 1993; Selzer *et al*., 2000). A good computational method to predict stability changes upon mutation will help in designing new or altered proteins with specific levels of stability, enzymatic activity and binding to other molecules (proteins, DNA, drugs, etc.). Moreover this will reflect our basic understanding of the rules that govern protein folding and binding processes, and will be very useful in drug design.

A computational method has to balance between two tasks to predict protein stability upon mutation. One is the search problem, i.e. to search through the three-dimensional conformational space (Voigt *et al*., 2000). As one further divides the space into smaller (higher resolution) bits, the search is growing exponentially (Pierce and Winfree, 2002). The second is the scoring problem. Theoretically, using QM calculations, these two entwined problems can have an exact solution (Morozov *et al*., 2004). However, for proteins with thousands of atoms embedded in water, this is not practical. Thus, the degrees of freedom for both the search and scoring functions must be reduced. For example, the search space can be reduced by considering only preferred side chain conformations (rotamers) (Dunbrack, 2002). Scoring functions can be simplified by considering only pairwise interactions. Forcefields take their energy functions from: physical-based potentials (PBP), which are based on the fundamental analyses of the forces between atoms (Brooks *et al*., 1983; Pearlman *et al*., 1995; Lazaridis and Karplus, 2000), and knowledge-based potentials (KBP), which rely on statistical analysis of different properties extracted from protein databases (Tanaka and Scheraga, 1976; Miyazawa and Jernigan, 1985). Both methods incorporate terms that describe different aspect of the protein structure. For example, PBP may use Coulomb's law to describe electrostatic interactions while the KBP may extract the probability to observe two charged atoms as function of the distance between them from known PDB structures. An intricate balance of the different energy terms is needed to correctly estimate the free energy of binding or folding (Makhatadze and Privalov, 1995; Frisch *et al*., 1997; Reichmann *et al*., 2007).

Many methods have been developed to predict change in protein stability utilizing PBP, KBP or a hybrid of them. In this study, we evaluate six different methods: CC/PBSA (Benedix *et al*., 2009), EGAD (Pokala and Handel, 2005), FoldX (Guerois *et al*., 2002), I-Mutant2.0 (Capriotti *et al*., 2005), Rosetta (Rohl *et al*., 2004) and Hunter (V.P., M.C. and G.S. unpublished). The first two rely on existing physical-based forcefields. EGAD utilizes OPLS-AA forcefield and was trained to reproduce experimental binding affinities. CC/PBSA generates an ensemble of structures and

[1]The two authors contributed equally to this work.

uses Gromacs to calculate averaged scores. FoldX and Hunter are KBPs that use different terms, though both were trained to reproduce experimental results. I-Mutant2.0 relies on SVM regression to reproduce experimental stability changes, whereas Rosetta is a hybrid of KBP and PBP. Rosetta was trained to reproduce the native sequence of native proteins. In this study, these methods were tested on a large set of single mutations, avoiding mutations that had been used for each software in its training.

Our results indicate that current computational methods predict the general trend of free energy change upon mutation, though they fail in details.

Methods

Data set of mutations

For the purpose of this study, a data set of 2156 mutations was compiled from two sources. The first source was a list of 964 single mutations that was published previously by Guerois *et al*. (2002). Five mutations were excluded from this list as either there was a mismatch with the wild-type residue in the protein crystal structure or the site of mutation had an incomplete side chain. Additionally, several PDB structures were replaced by others: 1LMB was used instead of 1LRP (C_α-only structure), 1BKS instead of 1WSY (obsolete), 1ANF instead of 2MBP (obsolete) and 3CHY instead of 1CEY (NMR structure).

The second set of 2972 single mutations was obtained from the ProTherm database (Kumar *et al*., 2006) and was filtered to exclude any mutation that was already listed in the first set, or where the structure of the protein was determined by NMR. The remaining 2048 mutations were joined with the first set, resulting in 3007 mutations. Some of the mutations in the data set were measured several times. Therefore, in such situations, the average value was calculated and mutations were presented only once. The final set lists 2156 mutations. In several cases, the protein structure of wild-type had incomplete side chains. Those were automatically reconstructed using Swiss-Pdb Viewer (Guex and Peitsch, 1997).

Mutations with multiple measurements

As was mentioned above, for some of the mutations in the ProTherm database, there were two or more experimental measurements. These 406 mutations were extracted and used to estimate the magnitude of the experimental error between the measurements. To evaluate a correlation between experimental data, we randomly picked two measurements for each mutation and plotted one against the other for all 406 mutations.

Backbone movement upon mutation

The set of 146 single mutations, for which wild-type and mutant structures exist, were obtained from the ProTherm database. It was assured that the structures differed exactly by one residue in the site of mutation. RMSDs between wild-type and mutant structures were calculated with the ProFit software (http://www.bioinf.org.uk/software/profit/).

Calculating change in protein stability

CC/PBSA This method uses a combined strategy to predict stability changes. At first, the Concoord program is used to generate an ensemble of structures. Then, the Gromacs forcefield is applied to evaluate structures and calculate averaged score. The designated web server (http://ccpbsa.bioinformatik.uni-saarland.de/ccpbsa/) was used to run the calculations. We used the 'Protein Stability' option in the job submission page. As the method is computationally demanding, only a limited set of 478 mutations was submitted for analysis.

EGAD EGAD is a method that utilizes the OPLS-AA forcefield, with the main focus of performing protein design on fixed backbone scaffolds. To run scanning mutagenesis, the 'scan_mut' option was used as outlined in the EGAD manual (http://egad.ucsd.edu/EGAD_manual/). Owing to the fixed backbone approach, EGAD does not allow calculating $\Delta\Delta G$ for mutations to or from: Cys, Gly or Pro, reducing the number of mutations by 515. Only structures with zero clashes were included in the final EGAD mutation subset. This further reduced the number of mutations by 576; thus, only half of the mutations could be evaluated using EGAD.

FoldX The recent version of FoldX forcefield (version 3.0) was obtained from http://foldx.crg.es/. As a first step, the structures of the wild-type proteins were minimized using the 'RepairPDB' command. Then, individual mutations were built using 'BuildModel' command and $\Delta\Delta G$ values were extracted from the FoldX output files.

Hunter Hunter is a KBP for accurate structure modeling that relies on a new method for high-resolution description of residue–residue interactions (V.P., M.C. and G.S., unpublished). This method defines the interaction of two residues in terms of four distances (4-*D*), which are calculated between two pairs of atoms. The pair of atoms can be chosen either on a side chain or on the backbone of a residue to define side chain–side chain (ScSc) and/or side chain–main chain (ScMc) interactions. Such description puts strict constraints on the mutual arrangement of chosen atoms in two interacting residues and allows analyzing in detail preferable geometry of residue interactions. The statistical preferences on 4-*D* geometry of 190 ScSc and 18 ScMc interactions were derived from a large set of high-resolution protein structures and were used as a basis for Hunter. Originally, Hunter was trained for optimal side chain modeling on a test set of proteins. In addition to a side chain–side chain term (E_{ScSc}) and a side chain–main chain term (E_{ScMc}), it includes a rotamer probability term (E_{rot}) and a modified Lennard-Jones term (E_{lj}):

$$\Delta E = \Delta E_{\mathrm{ScSc}} + \Delta E_{\mathrm{ScMc}} + \Delta E_{\mathrm{rot}} + \Delta E_{\mathrm{lj}} \qquad (1)$$

For the current study, Hunter was trained to predict the change in protein stability upon mutation. The mutant and the corresponding wild-type proteins were modeled for each mutation in the data set (fixed backbone). Any residue within 5 Å from the site of mutation was allowed to repack. Then, for each mutation, the difference in scores between mutant and wild-type protein was calculated ($\Delta\Delta E_{\mathrm{ScSc}}$,

$\Delta\Delta E_{ScMc}$, $\Delta\Delta E_{rot}$ and $\Delta\Delta E_{lj}$) and the optimal set of weights was derived to reproduce experimental $\Delta\Delta G$ values using least square fitting. The data set of 2156 mutations was divided into training and test sets. All mutations with $\Delta\Delta G$ values below -0.5 kcal/mol or above 1.5 kcal/mol were identified and half of them (456 mutations) were randomly chosen for the training set. The final regression model had a correlation factor of 0.47 on the training set and 0.45 on the test set (1700 mutations).

I-Mutant2.0 The Python program for calculating stability change was obtained from http://gpcr.biocomp.unibo.it/cgi/predictors/I-Mutant2.0/I-Mutant2.0.cgi and was run for every mutation in the data set using structural information.

Rosetta The Rosetta software (version 2.2.0) was obtained from http://www.rosettacommons.org/. The structures of wild-type proteins were minimized using Rosetta 'idealize' protocol. Then, mutant proteins were modeled using 'design' mode. Any residue within 5 Å from the site of mutation was allowed to repack. For each mutant protein in the data set, a corresponding wild-type protein was built and evaluated by allowing repacking the same set of residues as in the mutant. The stability scores of the mutant (ΔG_{mut}) and wild-type protein (ΔG_{wt}) were extracted from the output files and the stability change upon mutation ($\Delta\Delta G$) was calculated as $\Delta G_{mut} - \Delta G_{wt}$. During structure modeling, the backbone was kept fixed. It should be noted that Rosetta was not specifically trained for estimating protein stability changes upon mutations, but for protein design.

Per structure prediction accuracy

All 2156 mutations in the data set were divided into separate groups based on the wild-type protein structure. Only proteins for which data of more than 50 mutations were available were used for the analysis. The correlation coefficient was used to evaluate the method's ability to predict experimental $\Delta\Delta G$ values for a specific protein. Since some mutations were excluded from the different methods (i.e. mutations that were used as part of the training set for a specific method; see Results), the number of mutations per protein could be <50 in some groups.

Combining the methods

Combining a number of methods may improve prediction. The six methods used here can be combined into 57 possible combinations of two or more methods. For each combination, a common subset of mutations was determined, i.e. those mutations that all methods under consideration include after removing mutations that have been used for their training. As the number of common mutations can be very low, only 16 combinations with >400 mutations were used. The combined $\Delta\Delta G$ of each mutation was calculated as a simple average of $\Delta\Delta G$s predicted by individual methods. The common subset of mutation for each combination was divided randomly in two halves (the training set and the test set). Then, each combination was evaluated separately on the two halves. As the result of the evaluation depends on the original division of the data, the whole procedure was repeated 100 times and the average result and standard deviation was obtained.

Classification

All mutations in the data set were classified based on two criteria. In the first classification, all mutations were considered either as stabilizing ($\Delta\Delta G < 0$) or destabilizing ($\Delta\Delta G > 0$). In the second classification, mutations were classified as being hot spot ($|\Delta\Delta G| > 2$ kcal/mol) or not ($|\Delta\Delta G| < 2$ kcal/mol). The prediction performances were evaluated based on three measures: accuracy, sensitivity and specificity. Accuracy is defined as a percentage of correctly identified mutations out of total number of mutations (TP + TN)/Total (TP, true positive; TN, true negative). Sensitivity is defined as TP/(TP + FN). Specificity is defined as TN/(TN + FP) (FN, false negative; FP, false positive).

Results

Assessment of the different algorithms in predicting stability changes

We calculated the change in protein stability ($\Delta\Delta G$) on a data set of 2156 mutations using six different methods. To objectively compare the different algorithms, we eliminated from each method those mutations that were used for training, as reported in the original publications. For EGAD and Rosetta, we noted that many predictions were highly unrealistic. Examining individual mutants indicated that the problem stems from clashes in the modeled structures. Therefore, when a clash was reported in the EGAD output file, we removed the mutant from further consideration. In total, 576 mutant structures had clashes. Adding to this the 515 mutants that could not be modeled by EGAD (due to residue type) resulted in a reduction of the number of mutations that were evaluated to half (from 2156 to 1065). On the positive side, cleaning the data, substantially improved the performance (from $r = 0.16$ to 0.59). The Rosetta method does not explicitly indicate whether a structure has a clash; therefore, we used the following strategy: the maximal repulsive energy per residue in an idealized structure was identified, and if any residue in the modeled mutant protein had a repulsive energy above a cutoff of 7.1 (which is the highest value observed in minimized wild-type structures), then the mutation was excluded. In total, 243 mutations were discarded, improving the correlation from 0.05 to 0.26 (Table I and Fig. 1).

A plot of the performance of the different methods is shown in Fig. 1. We also report on the correlation coefficient and the slope of the linear fit. A slope of 1 suggests that the method is well calibrated. Interestingly, all the slopes were below 1, and thus they underestimate the experimental results. The best performing method was EGAD ($r = 0.59$), followed closely by CC/PBSA, I-Mutant2.0, FoldX and Hunter. However, EGAD was also the only method that could not model most mutations. Table I shows how the different methods perform on specific type of mutations. Mutations were divided into three classes: special mutations (involving Gly and Pro), mutations to alanine and all others. Interestingly, $\Delta\Delta G$ values for special and alanine mutations were predicted with the same level of accuracy, although mutations of Gly and Pro may affect the backbone and thus will be more difficult to predict. Non-alanine mutations were always predicted less accurately than mutations to Ala. Predicting $\Delta\Delta G$s for buried residues worked better than for exposed residues, except for CC/PBSA and Rosetta.

Table I. Accuracy of predicted change in $\Delta\Delta G$ upon mutation

Method	Special[a]		Alanine		Non-alanine		Exposed		Buried	
	r	*n*	*r*	*n*	*r*	*n*	*r*	*n*	*r*	*n*
CC/PBSA	0.73	83	0.62	152	0.41	243	0.52	336	0.50	142
EGAD	—	—	0.63	507	0.45	558	0.45	747	0.56	318
FoldX	0.48	191	0.49	124	0.46	885	0.43	742	0.50	458
Hunter	0.46	310	0.40	477	0.37	807	0.35	1031	0.47	563
I-Mutant2.0	0.56	281	0.52	351	0.38	301	0.44	531	0.53	402
Rosetta	0.25	390	0.32	603	0.23	920	0.35	1245	0.13	668
Average	**0.50**		**0.50**		**0.38**		**0.42**		**0.45**	
Method	Extended		Helix		Loop		All mutations[b]		Outliers[c]	
	r	*n*	*r*	*n*	*r*	*n*	*r*	*n*	*n*	
CC/PBSA	0.68	140	0.54	153	0.42	185	0.56	478	–	
EGAD	0.66	307	0.56	367	0.50	391	0.59	1065	1091	
FoldX	0.48	318	0.48	460	0.55	422	0.50	1200	–	
Hunter	0.53	461	0.46	546	0.33	587	0.45	1594	–	
I-Mutant2.0	0.57	375	0.56	241	0.39	317	0.54	933	–	
Rosetta	0.22	551	0.20	633	0.34	729	0.26	1913	243	
Average	**0.52**		**0.47**		**0.42**		**0.48**			

For each group, the correlation coefficient (*r*) and the number of mutations (*n*) are given.
[a]Special are the mutations involving Gly and Pro.
[b]The total number of mutations used to evaluate each method.
[c]Number of excluded mutations due to clashes. The number for EGAD includes also 515 mutations involving Gly, Pro, and Cys.

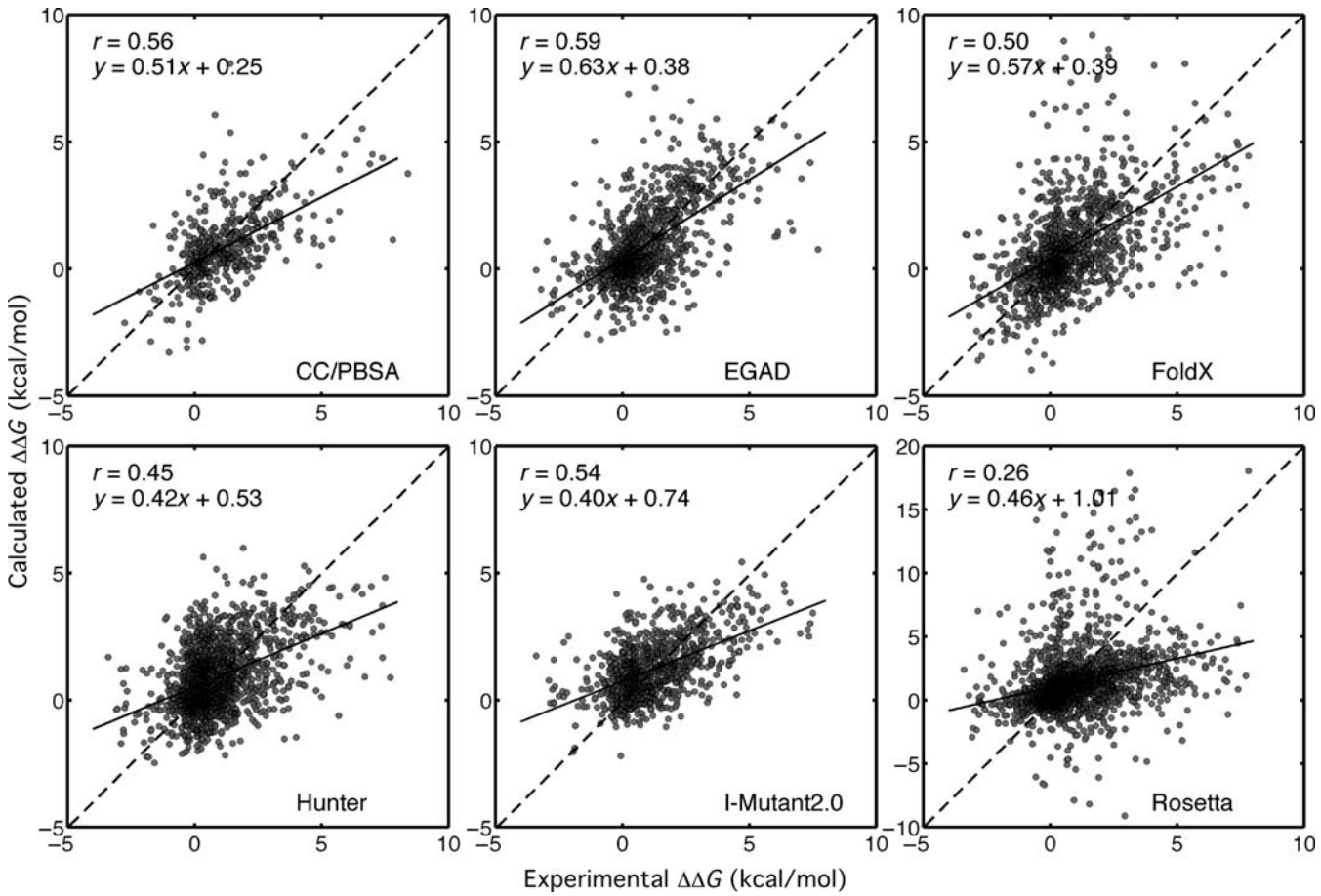

Fig. 1. Assessment of the different algorithms in predicting stability changes. Each method was tested on set of mutations that are not part of the training set. The correlation coefficient (*r*) and the equation of regression line (*y*) are given at the top left corner of each panel. A regression line is represented in solid, the dashed line is $y = x$. The number of mutations for each method is given in Table I.

Finally, classifying residues based on secondary structure type indicated that predicting $\Delta\Delta G$ of mutations located in β-sheets was best for four methods, whereas for FoldX and Rosetta, mutations in unstructured regions were best predicted.

As seen in Fig. 1, none of the methods was able to accurately predict $\Delta\Delta G$s for all mutations, as there is a significant deviation between experimental and calculated values. However, it is very important to note that in spite of errors in details, all the methods demonstrated a correct trend. This becomes particularly evident in Fig. 2. For this figure, all mutations were binned into 1 kcal/mol intervals, from -4 to 8 kcal/mol (based on the experimental $\Delta\Delta G$ values). Then for each bin, the average predicted $\Delta\Delta G$ was determined and plotted against the average experimental value. The average values show a much higher correlation coefficient compared with those shown in Fig. 1 (however, with a slope below 1). In this kind of analysis, Hunter and FoldX performed best ($r = 0.96$) followed closely by CC/PBSA, EGAD, Rosetta and I-Mutant2.0. This shows again that all the methods work well on averages, but less so on specific mutations.

Testing the methods on individual structures

Next, we evaluated the performance of the methods on individual proteins where sufficient data were available (Table II). A large variance in performance between methods for individual structures was observed. For example, effects of mutations in T4 lysozyme (PDB ID 2LZM) were predicted very well using I-Mutant2.0, but quite poorly using Rosetta. We found also extensive variance in prediction of the different structures using the same method. For example, the correlation coefficient of EGAD varied from 0.05 to 0.69 for mutants placed on different proteins. Yet, in general, all the methods seem to perform somewhat better on individual proteins than on all the mutations combined.

Combining different methods

We tested the possibility to improve $\Delta\Delta G$ predictions by combining results obtained for the same mutant using different methods. We tried all possible combinations of six different methods with sufficient number of mutations (see Methods for details). The largest improvement was observed for a set of 407 mutants for the combination of EGAD, I-Mutant2.0 and Rosetta (Fig. 3, $r = 0.64$). The different training sets persistently yield this combination as the best. However, even though the combined result of EGAD, I-Mutant2.0 and Rosetta was better than any of the six methods individually, correlation factor for EGAD alone ($r = 0.62$) on this set of 407 mutations was very close, and within the standard deviation of the combined method (SD = 0.03). In addition, only 7 out of 16 tested combinations showed improved correlation coefficient compared with the best result obtained by individual method on the same set of mutations.

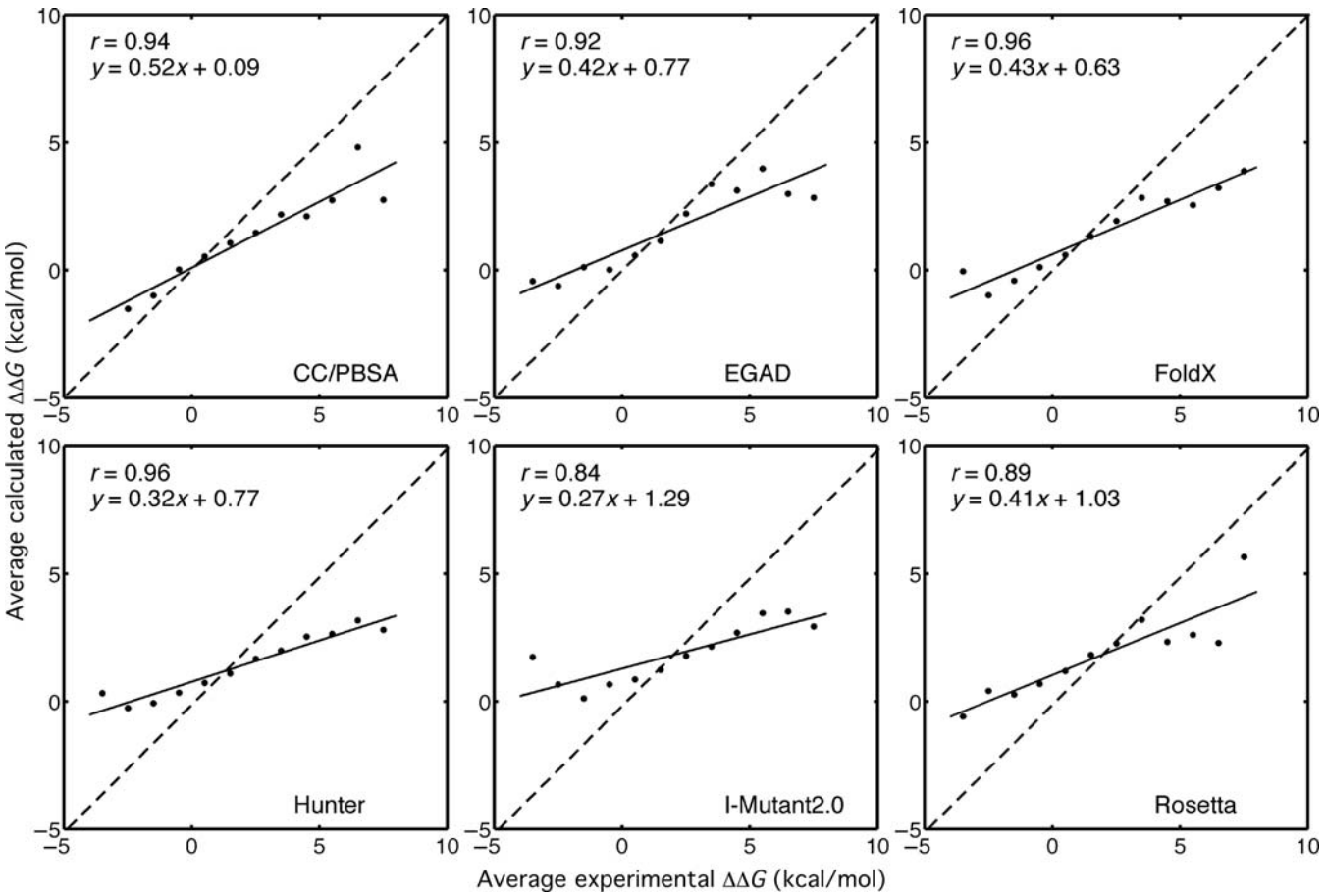

Fig. 2. Correlation between experimental and average predicted $\Delta\Delta G$ values. The data set of mutations was binned according to their experimental $\Delta\Delta G$s, from which the average $\Delta\Delta G$ was calculated for each interval (given as black solid dots). The solid line is a regression line, the dashed line is $y = x$. The correlation coefficient (r) between experimental and average predicted $\Delta\Delta G$, and the equation of the regression line are given at the top of each panel.

Table II. Evaluating predicted values on individual structures

PDB ID	CC/PBSA			EGAD			FoldX			Hunter			I-Mutant2.0			Rosetta		
	r	*n*	*s*	*r*	*n*	*s*	*r*	*N*	*s*	*r*	*n*	*s*	*r*	*n*	*s*	*r*	*n*	*s*
1a2p	0.58	68	0.5	0.60	56	0.8	-	-	-	0.56	37	0.5	-	-	-	0.29	64	0.4
1bni	0.61	94	0.3	0.17	68	0.5	0.73	95	0.8	0.55	44	0.6	0.83	12	0.4	0.42	91	1.0
1bvc	0.57	57	1.1	0.49	46	1.2	0.45	54	0.9	0.30	30	0.4	-	-	-	0.56	37	2.0
1hz6	-	-	-	0.65	43	0.7	-	-	-	0.52	32	0.6	0.75	57	0.6	0.46	56	0.8
1lz1	-	-	-	0.05	48	0.3	0.61	82	1.3	0.44	36	0.5	−0.65	9	−0.3	0.14	74	0.4
1stn	-	-	-	0.62	156	0.8	0.73	16	0.9	0.70	135	0.5	0.62	249	0.5	0.28	279	0.4
1vqb	-	-	-	0.17	62	0.4	0.53	92	0.3	0.45	46	0.2	-	-	-	0.17	83	0.3
1ypc	-	-	-	0.69	47	0.7	-	-	-	0.59	31	0.3	-	-	-	0.76	64	0.7
2lzm	-	-	-	0.48	194	0.8	0.63	163	0.8	0.57	116	0.7	0.87	14	0.6	0.23	216	0.6
2rn2	-	-	-	0.31	52	0.4	0.51	65	0.6	0.13	35	0.1	-	-	-	0.33	50	0.7
Average	**0.58**		**0.6**	**0.40**		**0.7**	**0.59**		**0.8**	**0.48**		**0.4**	**0.48**		**0.4**	**0.36**		**0.7**

For each group, the correlation coefficient (*r*), the number of mutations (*n*) and the slope of the regression line (*s*) are given. The dash sign (-) denotes cases where the number of available mutations was zero.

Predicting hot spots

Often we are more interested to know whether a mutation is stabilizing or destabilizing, or to identify hot spots, than to obtain the exact $\Delta\Delta G$ value. Results presented in Table III show that depending on the method, 69–79% of the mutations were correctly predicted as stabilizing or destabilizing. Next, we evaluated the ability of the methods to detect hot spot residues ($|\Delta\Delta G| > 2$ kcal/mol). Table III reports hot spot predictions based on accuracy, sensitivity and specificity (see Methods for details). Among the evaluated methods, EGAD was the most accurate in identifying hot spots. It had the highest percentage of correctly identified hot spots (accuracy) and the highest sensitivity, even though EGAD is not the most specific method.

Computational and experimental errors

We calculated the average unsigned error for all methods under consideration. As can be seen in Table IV, the average difference between experimental and predicted $\Delta\Delta G$ values was ∼1.2 kcal/mol. To reassure us that this is not the result of inaccurate experimental data, we also estimated the magnitude of experimental error. This was determined based on a set of mutations for which data were reported at least twice (see Methods for details). Plotting the individual measurements one against the other resulted in a straight line with a correlation coefficient of $r = 0.86$ (Fig. 4). The average unsigned error of the experimental data was 0.44 kcal/mol,

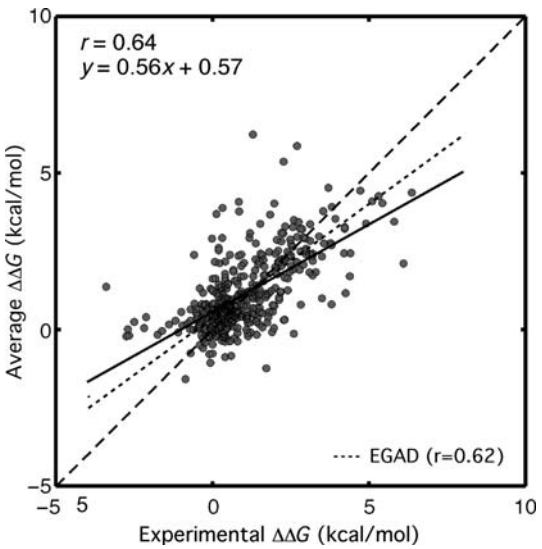

Fig. 3. Combining methods to improve $\Delta\Delta G$ prediction. EGAD, I-Mutant2.0 and Rosetta were used to predict $\Delta\Delta G$ over a set of 407 mutations. The average of these three methods was calculated for each mutation and plotted versus experimental $\Delta\Delta G$s. The results were compared with those achieved using EGAD alone on the same set (dotted line).

Table III. Predicting stabilizing/destabilizing mutations and hot spots

Method	Stabilizing/ destabilizing Accuracy (%)	Hot spot		
		Accuracy (%)	Sensitivity (%)	Specificity (%)
CC/PBSA	78.6	73.8	38.4	89.5
EGAD	71.0	80.0	63.1	87.0
FoldX	69.5	74.2	48.4	83.5
Hunter	69.4	73.7	49.6	85.4
I-Mutant2.0	77.5	75.4	50.0	87.0
Rosetta	73.4	67.5	52.0	75.4

For definitions of accuracy, sensitivity and specificity, see the Methods section.

Table IV. Computational and experimental error in $\Delta\Delta G$ determination

Method	Average unsigned error (kcal/mol)	Standard deviation (kcal/mol)
CC/PBSA	1.03	0.96
EGAD	1.00	0.95
FoldX	1.28	1.37
Hunter	1.09	0.92
I-Mutant2.0	1.01	0.86
Rosetta	1.68	2.32
Average	**1.20**	**1.23**
Best combination[a]	0.80	0.78
Experimental data[a]	0.44	0.61

[a]See Methods and Results for details.

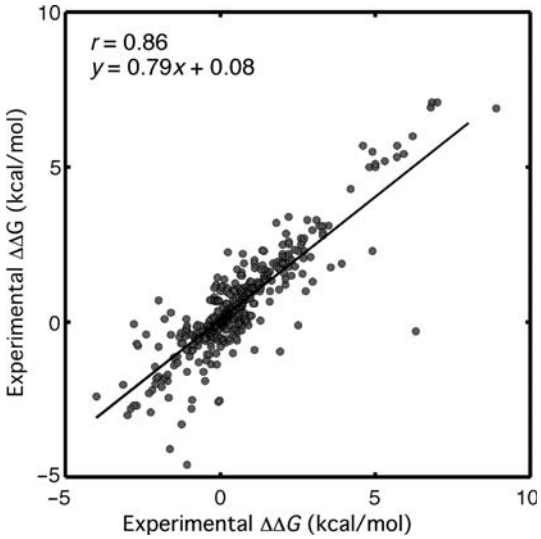

Fig. 4. Agreement in experimental measurements. For each of 406 mutations with two or more experimental measurements, two randomly chosen values were plotted against each other.

much below that found between experimental and calculated values.

Discussion

In the current study, we evaluated the accuracy of common methods used to predict stability changes in proteins upon mutation. We did this to compare the methods in a non-biased way, in order to assess their strength and weaknesses. On the positive side, all the methods were able to produce a correct trend in their predictions. However, they frequently failed in details. Comparing our results with those published in the original papers showed that none of them performed as good as reported (between $r = 0.7$ and 0.8). Mutations may introduce structural changes, which can be modeled with some degree of success (Kelley and Sternberg, 2009). To evaluate whether structural rearrangements upon mutations is indeed a major cause of concern, we calculated the RMSD of backbone movements upon single mutation and found it to be on average only 0.34 Å (as evaluated on a set of determined mutant structures, see Methods). Therefore, the additional complexity of potential backbone movements can be neglected, as indeed done in all of the methods. Therefore, problems in correctly estimating $\Delta\Delta G$s suggest inaccuracies in the scoring functions, which only estimate the forces. The six methods evaluated here utilize different classes of energy functions. EGAD and CC/PBSA are PBP, Hunter, FoldX and Rosetta are hybrid of PBP and KBP and I-Mutant2.0 is an SVM-based tool. The prediction accuracy achieved by the three forcefield classes was comparable. A more detailed analysis of the types of mutations that were easier to predict has shown that mutations to alanine were more predictable, as were buried mutations and those located in β-sheets (Table I). Non-alanine mutations are expected to be more difficult to predict as they introduce a new residue that may occupy a larger volume and make new interactions, whereas Ala mutations basically leave a cavity (Baldwin *et al*., 1993; Serrano *et al*., 1993). This introduces additional complexity for computational methods when predicting mutant structure. Indeed, predictions of mutations located in unstructured regions were among the least accurate (Table I).

From the six methods tested, EGAD performed best, even though the method was trained originally to reproduce experimental binding affinities. This method is also fast. However, the drawbacks of this method were its inability to predict mutations to Cys, Gly and Pro due to fix backbone paradigm and that about one-third of the mutations introduced clashes, making those results unusable. Those problematic mutations are not excluded automatically, and may introduce ambiguity in the results. In total, about 50% of mutations were not suitable for analysis by EGAD. Rosetta was the least accurate among the methods. It should, however, be noted that we used Rosetta design, which is not specifically trained for this task. A Robetta server does exist, but it uses as input only interfaces between proteins (Kim *et al*., 2004).

All of the methods underestimate change in protein stability for strongly stabilizing and destabilizing mutations. As can be seen in Fig. 1, the regression lines have slopes <1. However, all of the methods were able to correctly identify 68–80% of hot spots (Table III) or classify mutations as stabilizing or destabilizing with about the same quality. An important issue in predicting changes in stability of proteins upon mutation is accuracy of experimental data. Inherently, any experimental technique gives a spread between measurements. Results may vary even more when using different techniques, different conditions or performing the experiments by different groups. In such situation, none of the methods would be able to correctly predict $\Delta\Delta G$s. However, we found that experimental data agree very well; the correlation between different measurements is 0.86 with an average unsigned error 0.44 kcal/mol (Fig. 4). At the same time, an average unsigned error for the tested computational methods was about 3-fold larger (1.2 kcal/mol). Therefore, the limited accuracy of current methods is not due to poor experimental data but due to inherent inaccuracies in the scoring functions.

Combining results of EGAD, I-Mutant2.0 and Rosetta allowed improving $\Delta\Delta G$ predictions; however, the improvement over a single method on this data set of 407 mutants was not statistically significant (0.64 versus 0.62 with SD of 0.03). Although these three methods belong to different classes of predictors (physical, KBP and SVM) and are thus orthogonal, the different directions did apparently not add much new predictive power. In addition, only few combinations could yield any better correlation coefficient (7 out of 16). Giving a different weight for each method did not improve results significantly (data not shown).

Some of the studied methods (e.g. EGAD, FoldX, and Rosetta) were successfully applied to design new folds, improving protein stability or binding specificity (Kuhlman *et al*., 2003; Kortemme *et al*., 2004; Pokala and Handel, 2005; Szczepek *et al*., 2007). The questions then arise how did these method accomplish such complex tasks, while demonstrating moderate results in predicting $\Delta\Delta G$ values? A major component of any method is its scoring function. The inability to exactly estimate energy changes, as shown in this study, should lead inevitably to failure in structure prediction. However, it seems that the ability of all the scoring functions to produce correct trends is sufficient to predict and design protein structure. Moreover, this is also the

reason why MD simulations, protein modeling, etc. work. In most of these tasks, we are actually not interested in the exact fate of an individual residue, but of the protein as a thermodynamic identity. As is clearly shown in Fig. 2, averaging the energies of groups of residues provided the accuracy needed for computational biology. In summary, the current computational methods are clearly good enough for most of the tasks they are used for. Further improvement in scoring functions is, however, needed to calculate exact details, which are often required. It is not clear from our work, whether the current approaches will lead to this, or much more realistic energy functions will be needed.

Funding

This work was partially supported by the Israel Ministry of Science and Technology [grant number 263]; and MINERVA [grant number 8525].

References

Baldwin,E.P., Hajiseyedjavadi,O., Baase,W.A. and Matthews,B.W. (1993) *Science*, **262**, 1715–1718.
Benedix,A., Becker,C.M., de Groot,B.L., Caflisch,A. and Bockmann,R.A. (2009) *Nat. Methods*, **6**, 3–4.
Brooks,B.R., Bruccoleri,R.E., Olafson,B.D., States,D.J., Swaminathan,S. and Karplus,M. (1983) *J. Comput. Chem.*, **4**, 187–217.
Capriotti,E., Fariselli,P. and Casadio,R. (2005) *Nucleic Acids Res.*, **33**, W306–W310.
Cohen,M., Reichmann,D., Neuvirth,H. and Schreiber,G. (2008) *Proteins*, **72**, 741–753.
Dunbrack,R.L. (2002) *Curr. Opin. Struct. Biol.*, **12**, 431–440.
Fersht,A. (1999) In Freeman,W.H. (ed.), *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. Basingstoke, New York.
Frisch,C., Schreiber,G., Johnson,C.M. and Fersht,A.R. (1997) *J. Mol. Biol.*, **267**, 696–706.
Guerois,R., Nielsen,J.E. and Serrano,L. (2002) *J. Mol. Biol.*, **320**, 369–387.
Guex,N. and Peitsch,M.C. (1997) *Electrophoresis*, **18**, 2714–2723.
Kelley,L.A. and Sternberg,M.J. (2009) *Nat. Protoc.*, **4**, 363–371.
Kim,D.E., Chivian,D. and Baker,D. (2004) *Nucleic Acids Res.*, **32**, W526–W531.
Kortemme,T., Joachimiak,L.A., Bullock,A.N., Schuler,A.D., Stoddard,B.L. and Baker,D. (2004) *Nat. Struct. Mol. Biol.*, **11**, 371–379.
Kuhlman,B., Dantas,G., Ireton,G.C., Varani,G., Stoddard,B.L. and Baker,D. (2003) *Science*, **302**, 1364–1368.
Kumar,M.D., Bava,K.A., Gromiha,M.M., Prabakaran,P., Kitajima,K., Uedaira,H. and Sarai,A. (2006) *Nucleic Acids Res.*, **34**, D204–D206.
Lazaridis,T. and Karplus,M. (2000) *Curr. Opin. Struct. Biol.*, **10**, 139–145.
Makhatadze,G.I. and Privalov,P.L. (1995) *Adv. Protein Chem.*, **47**, 307–425.
Matthews,B.W. (1993) *Annu. Rev. Biochem.*, **62**, 139–160.
Miyazawa,S. and Jernigan,R.L. (1985) *Macromolecules*, **18**, 534–552.
Morozov,A.V., Kortemme,T., Tsemekhman,K. and Baker,D. (2004) *Proc. Natl Acad. Sci. USA*, **101**, 6946–6951.
Pearlman,D., Case,D., Caldwell,J., Ross,W., Cheatham,T., Debolt,S., Ferguson,D., Seibel,G. and Kollman,P. (1995) *Comput. Phys. Commun.*, **91**, 1–41.
Pickett,S.D. and Sternberg,M.J. (1993) *J. Mol. Biol.*, **231**, 825–839.
Pierce,N.A. and Winfree,E. (2002) *Protein Eng.*, **15**, 779–782.
Pokala,N. and Handel,T.M. (2005) *J. Mol. Biol.*, **347**, 203–227.
Reichmann,D., Rahat,O., Cohen,M., Neuvirth,H. and Schreiber,G. (2007) *Curr. Opin. Struct. Biol.*, **17**, 67–76.
Rohl,C.A., Strauss,C.E., Misura,K.M. and Baker,D. (2004) *Methods Enzymol.*, **383**, 66–93.
Selzer,T., Albeck,S. and Schreiber,G. (2000) *Nat. Struct. Biol*, **7**, 537–541.
Serrano,L., Day,A.G. and Fersht,A.R. (1993) *J. Mol. Biol.*, **233**, 305–312.
Szczepek,M., Brondani,V., Buchel,J., Serrano,L., Segal,D.J. and Cathomen,T. (2007) *Nat. Biotechnol.*, **25**, 786–793.
Tanaka,S. and Scheraga,H.A. (1976) *Macromolecules*, **9**, 945–950.
Voigt,C.A., Gordon,D.B. and Mayo,S.L. (2000) *J. Mol. Biol.*, **299**, 789–803.

Received June 2, 2009; revised June 2, 2009; accepted June 3, 2009

Edited by Michael Sternberg