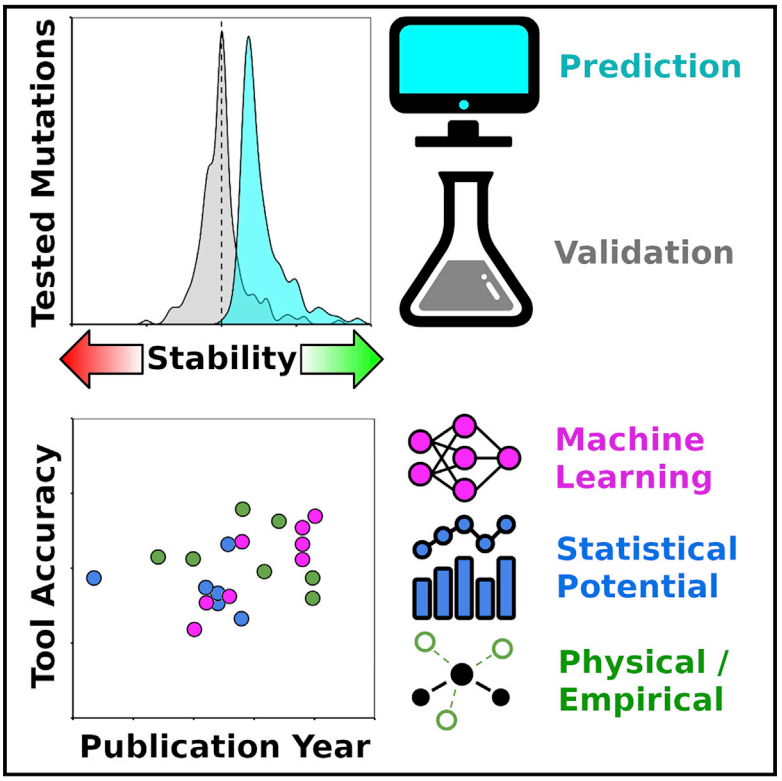
hotpaper.net

Resource

Structure

Computational Modeling of Protein Stability: Quantitative Analysis Reveals Solutions to Pervasive Problems

Graphical Abstract



Authors

Aron Broom, Kyle Trainor, Zachary Jacobi, Elizabeth M. Meiering

Correspondence

meiering@uwaterloo.ca

In Brief

Stabilizing proteins by mutation is critical for therapeutic and industrial applications. Many computational tools accurately predict highly destabilizing mutations; in contrast, mutations predicted to stabilize are neutral on average and often decrease solubility. Concurrent multi-mutations and suitable statistical analysis can markedly increase the probability of stabilization.

Highlights

- Experimentally, mutations predicted to stabilize are near neutral on average
- Stability predictors favor mutations that increase stability but decrease solubility
- Predictor performance is quantified well by the Matthews correlation coefficient
- Multi-mutants reach stability targets with higher probability than single mutants

Broom et al., 2020, Structure *28*, 717–726
June 2, 2020 © 2020 Elsevier Ltd.
https://doi.org/10.1016/j.str.2020.04.003

CellPress

hotpaper.net

Structure

CellPress

Resource

Computational Modeling of Protein Stability: Quantitative Analysis Reveals Solutions to Pervasive Problems

Aron Broom,[1] Kyle Trainor,[1,2] Zachary Jacobi,[1,2] and Elizabeth M. Meiering[1,3,*]
[1]University of Waterloo, Department of Chemistry, Waterloo, N2L 3G1, Canada
[2]These authors contributed equally
[3]Lead Contact
*Correspondence: meiering@uwaterloo.ca
https://doi.org/10.1016/j.str.2020.04.003

SUMMARY

Accurate modeling of the effects of mutations on protein stability is central to understanding and controlling proteins in myriad natural and applied contexts. Here, we reveal through rigorous quantitative analysis that stability prediction tools often favor mutations that increase stability at the expense of solubility. Moreover, while these tools may accurately identify strongly destabilizing mutations, the experimental effect of mutations predicted to stabilize is actually near neutral on average. The commonly used "classification accuracy" metric obscures this reality; accordingly, we recommend performance measures, such as the Matthews correlation coefficient (MCC). We demonstrate that an absurdly simple machine-learning algorithm—a neural network of just two neurons—unexpectedly achieves high classification accuracy, but its inadequacies are revealed by a low MCC. Despite the above limitations, making multiple mutations markedly improves the prospects for achieving a stabilization target, and modest improvements in the precision of future tools may yield disproportionate gains.

INTRODUCTION

Proteins are being used in an increasingly wide range of industrial, medical, and research applications (Bornscheuer et al., 2012; Choi et al., 2015; Truppo, 2017; Sheldon and Woodley, 2018). Although the societal and economic impacts of proteins have been growing, their full potential has yet to be realized because natural proteins have a limited repertoire of functions and insufficient stability to survive challenging application conditions (Bornscheuer et al., 2012; Bommarius and Paye, 2013; Huang et al., 2016; Sheldon and Woodley, 2018). Computational modeling of proteins promises to overcome these limitations via the rational design of requisite stability and novel functions. However, while notable successes have been reported, most designs still fail to be stably folded and soluble (Rocklin et al., 2017; Koga et al., 2012; Parmeggiani et al., 2015), and local or global instability commonly thwarts achieving desired activity (Procko et al., 2014; Bommarius and Paye, 2013; Gershenson et al., 2014; Khersonsky et al., 2012). As a consequence, the optimization of existing proteins or design of new ones typically requires laborious and costly high-throughput screening, a barrier that impedes broad commercial development (Bornscheuer et al., 2012; Truppo, 2017; Sheldon and Woodley, 2018). Reliable computational tools for modeling protein stability would enable the rapid and economical development of proteins for innumerable applications while also advancing understanding of the impacts of mutations during directed (Bloom et al., 2006; Tokuriki et al., 2008) and natural (Frey et al., 2010) evolution as well as in disease (Stefl et al., 2013; Stein et al., 2019).

Here, we test 21 methodologically diverse and commonly used protein stability prediction tools against experimentally characterized mutations. Th'e scoring or energy function is an essential element of computational engineering tools, used to discriminate between stable and unstable, active and inactive protein variants. The scoring functions for the examined tools cover current approaches, from molecular mechanics (CC/PBSA, Benedix et al., 2009; LIE, Wickstrom et al., 2012; and EGAD, Pokala and Handel, 2005) to statistical functions (DFire, Yang and Zhou, 2008; SDM, Worth et al., 2011; CUPSAT, Parthiban et al., 2006; Eris, Yin et al., 2007; MultiMutate, Deutsch and Krishnamoorthy, 2007; and Hunter, Cohen et al., 2009), and empirical combinations thereof (Rosetta, Kellogg et al., 2011; FoldX, Schymkowitz et al., 2005; Bioluminate, https://www.schrodinger.com/products/bioluminate; and ENCoM, Frappier and Najmanovich, 2014). Also, tools with scoring functions based on machine learning with physico-chemical input features (IMutant2, Capriotti et al., 2005; IMutant3, Capriotti et al., 2008; and MuPro, Cheng et al., 2006) and those using machine learning with statistical scoring functions as inputs (PoPMuSiC, Dehouck et al., 2009; MAESTRO, Laimer et al., 2015; NeEMO, Giollo et al., 2014; DUET, Pires et al., 2014a; and mCSM, Pires et al., 2014b) are included. Through systematic assessments using discerning

hotpaper.net

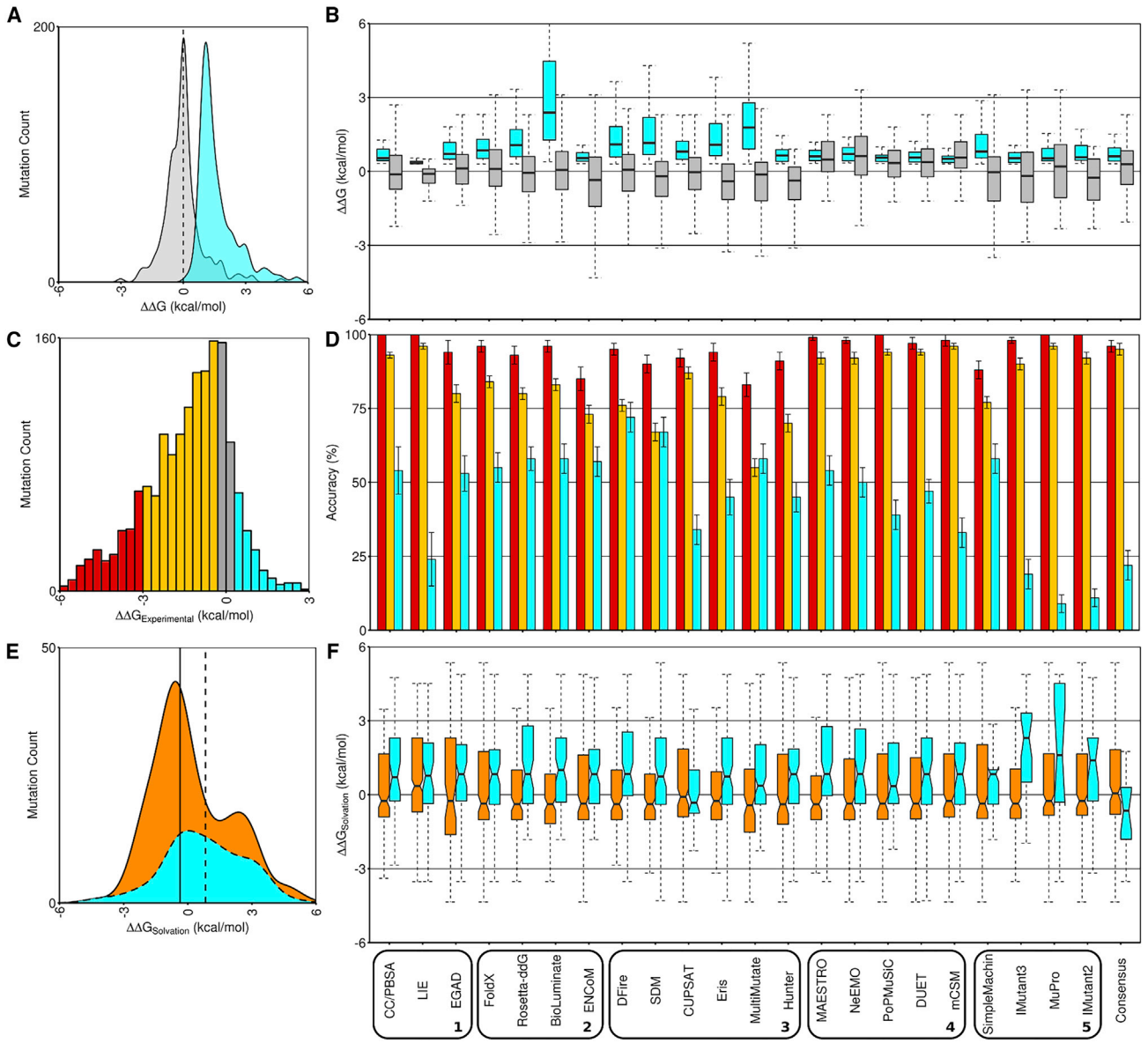


Figure 1. Performance of Computational Protein Stability Prediction Tools

(A) Using computational tools to predict stabilizing mutations (positive values of change in Gibbs free energy of unfolding, $\Delta\Delta G$, cyan distribution) in a range of proteins (Table S2) results in experimental changes in stability that are typically neutral.

(B) Testing prediction tools against a large dataset of point mutations (Table S3) reveals the general phenomenon that mutations predicted to stabilize (cyan boxplots) result in essentially no stability change experimentally (gray boxplots, see also Figure 2 for full density plots). Boxes encompass the middle 50% of the data, solid horizontal lines indicate the mean, notches correspond to the 95% confidence interval of the mean, and dashed whiskers extend to 1.5 times the interquartile range.

(C) The Protherm database (Bava et al., 2004), generally used for training and/or testing of computational protein design tools, consists predominantly of highly (red) and moderately (yellow) destabilizing mutations, with fewer neutral (gray) and moderately stabilizing (cyan) mutations.

(D) Tools often correctly classify highly (red) and moderately (yellow) destabilizing mutations as destabilizing, but have markedly lower accuracy when classifying moderately stabilizing mutations (cyan). Mutations with no effect on stability within experimental uncertainty (magnitude $\geq$ 0.3 kcal/mol) cannot be reliably analyzed and so are excluded. Error bars represent the standard-deviation from the mean after taking 1000 bootstrap samples.

(E) Analysis of the Protherm database reveals that experimentally validated stabilizing mutations on protein surfaces (cyan distribution) typically increase side chain hydrophobicity (positive $\Delta\Delta G_{solvation}$), with a median change of $\sim$0.8 kcal/mol (dashed vertical line) similar to mutating alanine to valine. Conversely, experimentally validated destabilizing mutations on the protein surface (orange distribution) are typically to more hydrophilic residues.

(F) The trend in (E) is recapitulated by computational protein engineering tools, with mutations predicted to stabilize on the protein surface (cyan boxplots) being to more hydrophobic residues, and those predicted to destabilize (orange boxplots) being to more hydrophilic residues.

Data for EGAD, FoldX, Rosetta-ddG, CUPSAT, DFire, Hunter, MultiMutate, SDM, PoPMuSiC, IMutant3, and MuPro are from Broom et al. (2017). Tool classes are identified by numbered boxes: 1, physical forcefields; 2, empirical potentials; 3, statistical potentials; 4, machine learning using statistical potentials; and 5, machine learning using physico-chemical features.

hotpaper.net

CellPress

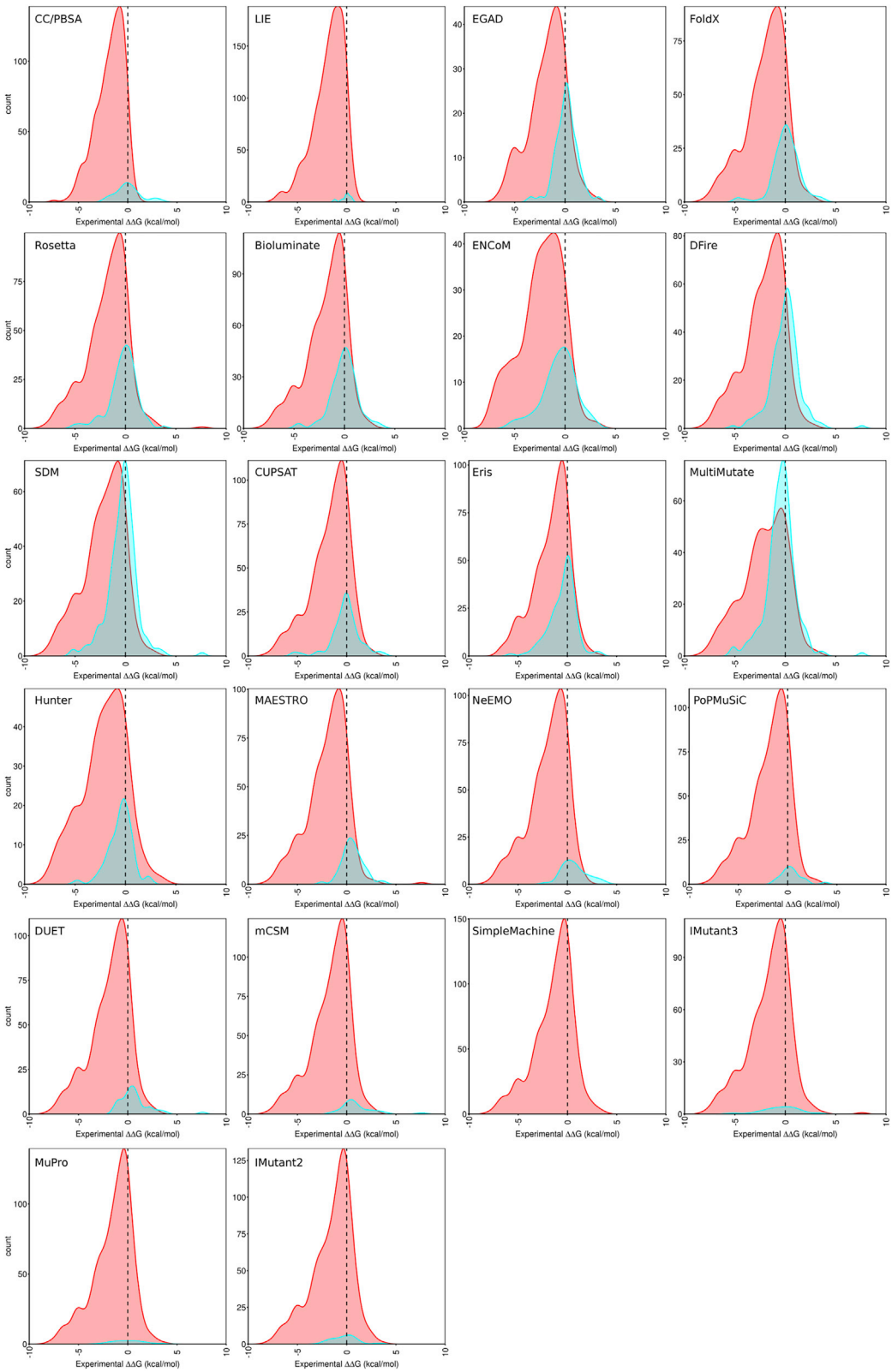


Figure 2. Prediction Tools Poorly Discriminate Stabilizing from Destabilizing Mutations

For each tool, a density distribution of mutations predicted to stabilize (cyan) or destabilize (red) is shown as a function of the experimentally determined change in stability (+ve stabilizing). Mutations predicted to stabilize typically form normal-like distributions centered around no change in experimental stability (dashed

(legend continued on next page)

hotpaper.net

metrics for tool performance (described in detail below), the analysis defines current capabilities and limitations for determining stability, as well as unexpected consequences for solubility. In addition, the assessments highlight current best practices and strategies for future advances; notably, how impressive gains in engineering protein stability can be achieved via multiple concurrent mutations.

RESULTS AND DISCUSSION

Predicted Stabilizing Mutations Are Typically Neutral

We find that, although protein stability prediction tools report ~75%–80% accuracy for discriminating stabilizing from destabilizing mutations (Table S1), only ~20% of point mutations predicted to stabilize have the intended result. Compiling the results of recent studies that used various computational tools to engineer stability enhancing point mutations, the individual mutations are predicted to yield on average ~1–2 kcal/mol of stability (Figure 1A, cyan distribution; Table S2); however, the experimentally measured effect is typically no change in stability (Figure 1A, gray distribution; Table S2). Testing the 21 computational tools against single-point mutations derived from the Protherm database (Bava et al., 2004) (Table S3), we find a similar problem: the average experimental effect of mutations predicted to stabilize is neutral (Figures 1B and 2). We note that tools using a statistical scoring function as input to a machine-learning algorithm appear to provide the best performance (Figure 1, group 4, PoPMuSiC, etc.), yet even in these cases the typical experimental stabilization is only ~0.5 kcal/mol (Figure 1B, gray boxplots, group 4). Although not applied to the full dataset of mutations here, molecular dynamics and thermodynamic integration were the most successful approaches used by researchers seeking to stabilize proteins (~50% success rate, see Table S2, Thermodynamic Integration; Song et al., 2013). The high computational cost of these methods presents a barrier to their extensive application, which future computational advances may help overcome (Gapsys et al., 2012; Tian et al., 2015; Perez et al., 2016). Altogether the discrepancies noted here between the predicted and experimental effects of mutations arise because the tools are good at identifying substantially destabilizing mutations, but the impacts of mutations predicted to stabilize are distributed in a narrow range, near neutral on average (−2 to +2 kcal/mol, see Figures 1A, and 2). This finding has ramifications for practical protein engineering, which are analyzed further below.

Prediction Accuracy Is Skewed toward Highly Destabilizing Mutations

Most tools, while correctly identifying >95% of highly destabilizing (−6 to −3 kcal/mol) mutations, and >80% of moderately destabilizing mutations (−3 to 0 kcal/mol), only correctly identify stabilizing mutations in <50% of cases (Figures 1C and 1D, red, yellow, and cyan bars, respectively). Although this problem has been previously identified in select cases (Foit et al., 2009), its prevalence has yet to be appreciated. A few exceptions present themselves, such as DFire (Yang and Zhou, 2008) and SDM (Worth et al., 2011)—tools using statistical potential scoring functions with limited training against existing experimental mutation datasets. These tools correctly identify most stabilizing mutations, but often mistake moderately destabilizing mutations for stabilizing ones, resulting in poor discrimination. Most mutations are destabilizing (Broom et al., 2017); contaminating the very small pool of stabilizing mutations through classification errors limits reliability. Since existing datasets of experimental mutations contain predominantly destabilizing mutations (Figure 1C), training with more balanced data may be a useful strategy to eliminate bias toward recognizing destabilizing mutations more effectively than stabilizing ones. Deep sequencing is an example of an approach that may generate such datasets (Araya et al., 2012), as is measurement of all point mutations via robotic automation (Nisthal et al., 2019). Improved sampling of protein conformational space may also reduce the rate of classification errors made by these tools (Davey et al., 2015; Barlow et al., 2018). Together, these results illustrate the challenge of engineering stabilizing mutations using computational tools: of all possible mutations most will be destabilizing such that, while tools will filter out many of the worst, those predicted to be stabilizing are just as likely to be neutral or destabilizing.

Stabilizing Surface Mutations Often Sacrifice Solubility

Our analyses uncover another challenge that complicates stability engineering: unintended reduction of protein solubility. Analyzing mutations on the protein surface that are experimentally confirmed to stabilize reveals that such mutations often increase hydrophobicity (Figure 1E); exploiting this phenomenon to increase stability (Nisthal et al., 2019) (Figure 1F) may have the undesirable effect of reducing protein solubility. In general, natural proteins balance trade-offs between stability, solubility, folding, and function (Bloom et al., 2006; Tokuriki et al., 2008; Klesmith et al., 2017; Gosavi, 2013). The requirement for solubility may be observed in evolutionary preferences inferred from consensus mutations, which show a preference for hydrophilic mutations on the protein surface (Figure 1F, Consensus). Also, consensus mutations are often stabilizing (Magliery, 2015). Thus, the incorporation of consensus information into stability predictions (Berliner et al., 2014; Goldenzweig et al., 2016) might favor mutations that do not sacrifice solubility for stability. Still, evolutionary information exists only for natural proteins, limiting its utility for tailor-made novel proteins. Alternatively, treating solvent exposed and buried positions differently, as does the statistical potential used by CUPSAT (Parthiban et al., 2006) which does not show a preference for surface hydrophobics (Figure 1F, CUPSAT), may offer similar benefits. Mutations likely to decrease solubility could also be filtered out using computational solubility prediction tools (Trainor et al., 2017). Similar logic inspired a protocol for the avoidance of hydrophobic patches on the protein surface during *de novo* design with Rosetta (Jacak

lines). Tools using machine learning with a statistical potential as input (MAESTRO, NeEMO, PoPMuSiC, DUET, mCSM) have a distribution of predicted stabilizing mutations slightly shifted in favor of experimentally stabilizing, and thus tend to be more reliable when attempting to engineer more stable proteins. However, this reliability comes at the cost of incorrectly predicting most experimentally stabilizing mutations as destabilizing (relatively small cyan area compared with red area to the right of the dashed line). Thus, stability prediction tools face a no-win situation of either finding a small set of stabilizing mutations with moderate reliability or a large pool of stabilizing mutations intermixed with an equally large number of neutral or destabilizing mutations.

hotpaper.net

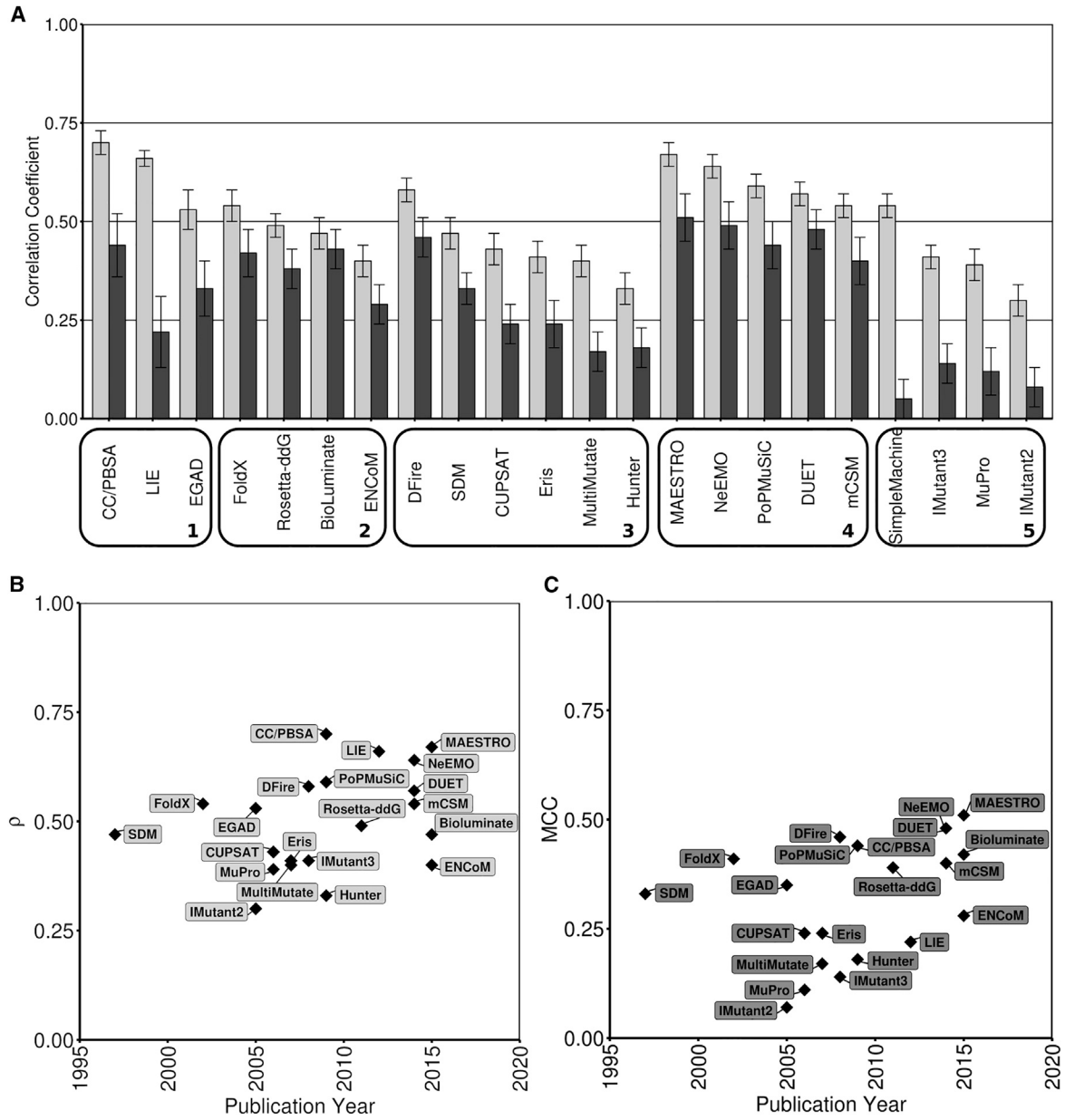


Figure 3. Performance Metrics Highlight Considerable Differences between Tools

(A) Using our recommended performance metrics (Spearman rank and Matthews correlation coefficients, light and dark gray, respectively) distinguishes the best performers within and between classes (classes identified by numbered boxes: 1, physical forcefields; 2, empirical potentials; 3, statistical potentials; 4, machine learning using statistical potentials; and 5, machine learning using physico-chemical features). Error bars represent the standard-deviation from the mean after taking 1000 bootstrap samples. Although computational tools have in general been improving over the years, much room for further improvement exists as measured by the (B) Spearman rank and (C) Matthews correlation coefficients. For example, the best tools approach a Spearman rank correlation coefficient of 0.75, where the square of this coefficient (0.56) indicates the best tools still only capture ~50% of what causes a mutation to be better or worse than another. Similarly, the tools most extensively used by other researchers to make mutations, Rosetta-ddG, FoldX, and PoPMuSiC, have Matthews correlation coefficients between 0.3 and 0.5, and produce success rates between ~15% and 40%.

et al., 2012). In fact, stabilization can be achieved without increasing surface hydrophobicity, as illustrated by work that optimized charge-charge networks on the protein surface leading to increased stability and foldability without sacrificing solubility (Tzul et al., 2015). These findings suggest various avenues for improving protein stability engineering while also addressing solubility.

Discerning Performance Metrics for Computational Tools

The performance of stability prediction tools is typically evaluated using well-established metrics: the linear correlation coefficient (r), accuracy, and error. These metrics have the advantage of being widely recognized but can also be misleading. In particular, linear correlation coefficients may be uninformative when

hotpaper.net

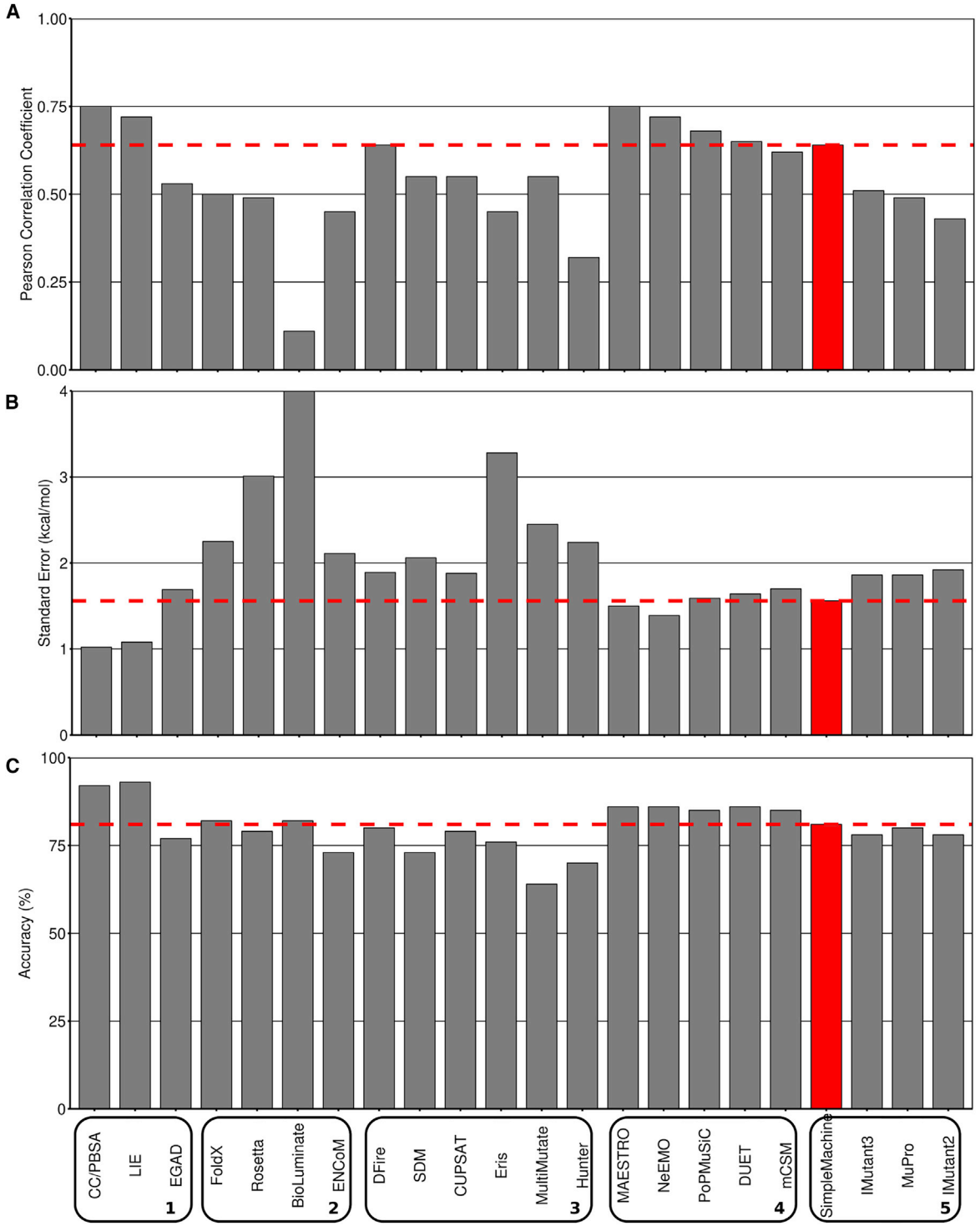


Figure 4. Evaluation of Tool Performance Using Typical Metrics

SimpleMachine (highlighted in red) performs better than many of the published tools on the basis of (A) the Pearson correlation coefficient and (B) standard error, and (C) accuracy.

the data contain outliers or anchoring points. The practice of removing said outliers on the basis of their poor fit inflate apparent performance metrics (Myers et al., 2010). Furthermore, binary (stabilizing/destabilizing) classification accuracy may be uninformative when the validation dataset is skewed toward one class (such as a preponderance of destabilizing mutations, Figure 1C). Error is also a poor discriminator in the case of protein stability as most point mutations for which changes in stability

hotpaper.net

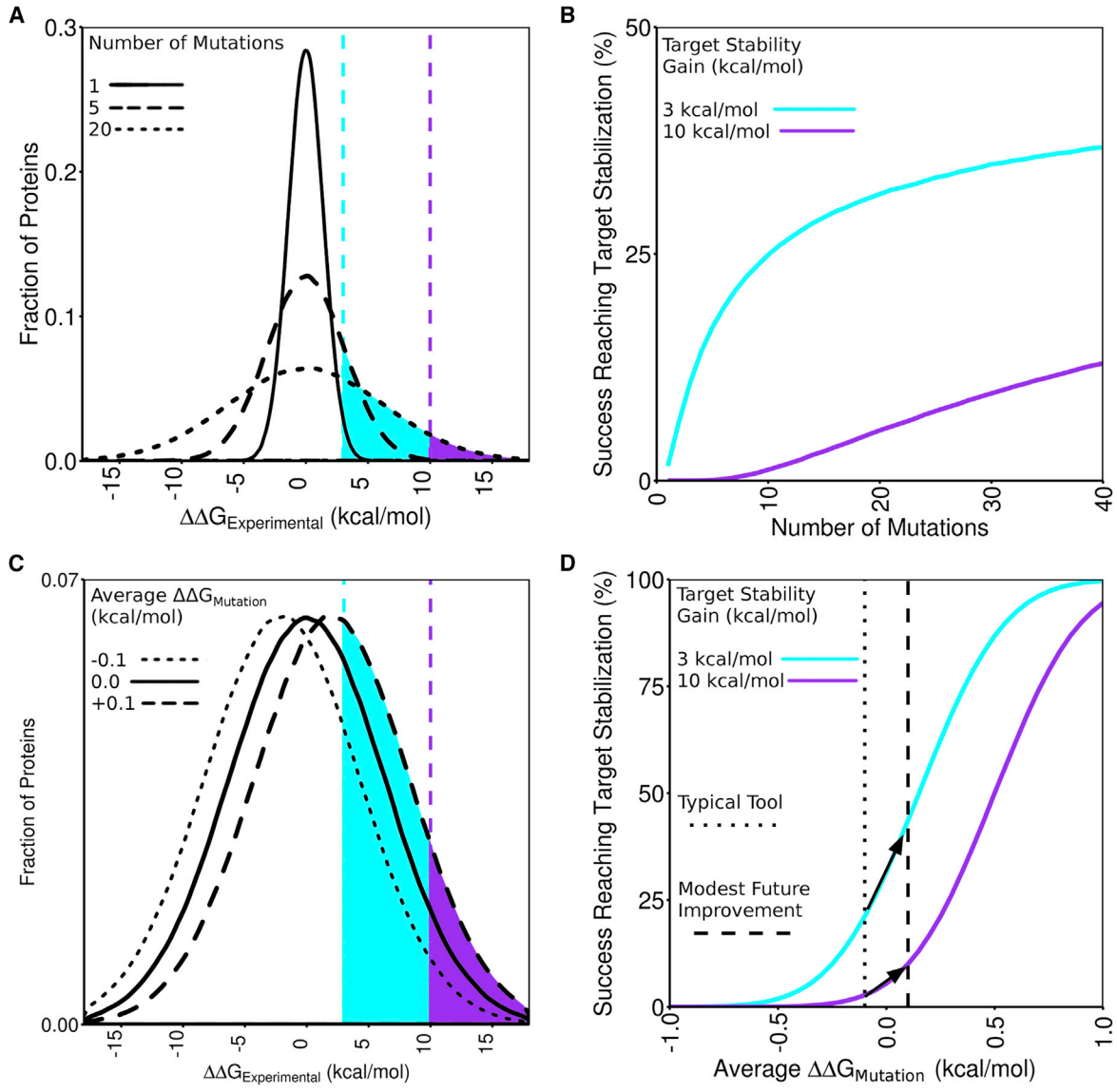


Figure 5. Multi-mutants Increase Protein Engineering Success

The effect on protein stability of making different numbers of predicted stabilizing mutations are based on experimental distributions from our dataset (see STAR Methods).

(A) Single mutations (solid line) are unlikely to yield substantial increases in stability (3 kcal/mol, cyan, area corresponds to ~2% probability), whereas multi-mutants of 5 (dashed line) or 20 (dotted line) residues are more likely to reach these targets (~15% and ~30% probability, respectively), and in the latter case may provide extreme stabilization (10 kcal/mol, purple area corresponds to an ~5% probability).

(B) More generally the success chance for reaching a target stability (3 kcal/mol, cyan; 10 kcal/mol purple) increases with increasing number of mutations.

(C) Increasing the accuracy of protein design tools (average experimental $\Delta\Delta G$ of mutations predicted to stabilize of −0.1, 0.0, and 0.1 kcal/mol shown as dotted, solid, and dashed smoothed histograms, respectively) increases the probability of reaching target levels of stabilization (shown here for making a 20-residue multi-mutant). The average experimental $\Delta\Delta G$ values of −0.1 and 0.0 kcal/mol are representative of most tools, whereas 0.1 kcal/mol illustrates the effect of a modest improvement in tool accuracy.

(D) When using a computational protein engineering tool with typical accuracy (average experimental $\Delta\Delta G$ of mutations predicted to stabilize of −0.1 kcal/mol, black dotted line), the probability of gaining 3 kcal/mol of stability (cyan curve) is ~25%, and for gaining 10 kcal/mol of stability (purple curve) the probability is only ~5%, even when making 20 mutations (the number of mutations illustrated in this panel). The success in reaching these targets is considerably improved to ~40% and ~10%, respectively (black arrows) with even small gains in tool accuracy (average experimental $\Delta\Delta G$ of mutations predicted stabilize of 0.1 kcal/mol, black dashed line).

have been experimentally determined are conservative (e.g., hydrophobic truncations), resulting in a narrow range of stability changes (typically, −2 to 0 kcal/mol); this unbalanced distribution results in low prediction error achieved simply by predicting small changes in stability and underestimation of the magnitude of stability changes resulting from types of mutations that are not

hotpaper.net

well-represented in the training or validation datasets. To address these problems, we propose the use of metrics better suited to the analysis of stability prediction data, the rank-order (ρ) coefficient (Myers et al., 2010) and Matthews correlation coefficient (MCC) (Matthews, 1975). Using these metrics, we demonstrate machine-learning approaches whose inputs include statistical potential terms as particularly effective, whereas physico-chemical features as machine-learning inputs yield the poorest performance (Figure 3A). Overall, while computational protein engineering tools have progressed over the years, there remains much room for improvement (Figures 3B and 3C).

To illustrate the advantages of the proposed metrics for stability prediction, we present an intentionally simplistic prediction tool, SimpleMachine. SimpleMachine is a neural network comprised of a single, two-neuron hidden layer and accepts as input four commonly used structural descriptors of amino acid substitution: change in volume, change in hydrophobicity, change in secondary structure propensity, and location (i.e., solvent exposure) (Figure S1). SimpleMachine would not be expected to achieve predictive power comparable with the other machine-learning tools used herein which include between 30 and 180 inputs and use complex architectures (Table S1). Notably, however, SimpleMachine appears to perform better than or equal to 15, 12, or 17 of the 21 tools tested when ranked based on *r*, accuracy, or error, respectively (Figure 4). When evaluated instead using rank-order coefficient and MCC, SimpleMachine's deficiencies are revealed, as are the deficiencies of other tools (Figure 3A).

Making Multiple Mutations Increases Stabilization Success

In light of the finding that computational tools identify stabilizing point mutations with a relatively low success rate of ~20% (Figure 1; Table S2), it is counter-intuitive that similar success rates have been reported for *de novo* design and that the resulting proteins are often extremely stable (Koga et al., 2012; Parmeggiani et al., 2015). To understand this, we use the simplifying assumption that each point mutation is independent and model the expected experimental stabilization resulting from multiple mutations (from the pool of those predicted to stabilize, Figure 1A). The model reveals that the greater the number of mutations, the better the odds are of reaching a given stabilization goal (Figures 5A and 5B). Although the multi-mutants—like the single mutants that constitute them—are still most likely to have no change in stability, the distribution of changes is broader (Figure 5A), increasing the odds of achieving impressively high stability. Although mutations made during whole-scale redesign or *de novo* design are unlikely to be independent, the success rates and typical total stability predicted by our model are in reasonable agreement with the experimental data (Koga et al., 2012; Parmeggiani et al., 2015; Dantas et al., 2003). These results offer some explanation for the remarkable successes of *de novo* design given the difficulty of accurately predicting even single-point mutations and suggest making multiple mutations is a useful strategy for achieving stabilization targets.

Another key result of the above model is that even relatively small improvements in stability prediction accuracy will produce markedly higher success rates when making multiple mutations. Consider, for example, a 20-position multi-mutant; an improvement in the mean experimental effect of mutations predicted to stabilize from 0.0 kcal/mol (the current value for most tools, see Figure 1A) to +0.2 kcal/mol—a modest improvement—greatly increases the chances of gaining 3 kcal/mol of stability (from ~33% to ~60%) (Figures 5C and 5D). To put this in context, we note that, for a protein of 100 amino acids, with a moderate melting temperature of 40°C, an increase in thermodynamic stability of 3 kcal/mol corresponds roughly to an increase in melting temperature of 20°C (Rees and Robertson, 2001). Such a gain in stability would be extremely valuable for industrial applications, as it translates to much longer enzyme lifetimes at higher reaction temperatures used to increase activity (Bommarius and Paye, 2013).

Conclusions

Reliable computational tools promise to be invaluable in the development of proteins with a wide range of impactful applications (Bornscheuer et al., 2012; Huang et al., 2016; Truppo, 2017; Sheldon and Woodley, 2018), shed light on the roles of mutations resulting from both directed and natural evolution, and help us better understand the molecular mechanisms of disease (Frey et al., 2010; Stefl et al., 2013; Stein et al., 2019). Our analyses identify important shortcomings in the computational protein stability prediction tools currently available: (1) the tools effectively identify strongly destabilizing mutations, but their performance is substantially lower for stabilizing mutations; (2) mutations predicted to stabilize actually produce a distribution of stability changes that is near neutral on average; and (3) stabilizing mutations may increase the surface hydrophobicity of proteins, consequently decreasing solubility. Our results suggest both strategies for the use of currently available stability prediction tools and ways in which they may be improved.

We find that making multiple concurrent mutations can greatly increase the odds, relative to single mutations, of achieving a given stabilization target. The positive impact of this strategy will be amplified as stability prediction tools improve, i.e., when mutations predicted to stabilize are better than neutral, on average. Using balanced experimental datasets (with suitable representation of stabilizing and destabilizing mutations) for training and parameterization, using molecular dynamics-based methods that do not rely so heavily on experimental data, and incorporating evolutionary preferences, are auspicious avenues for future stability prediction tool development, guided by the robust metrics we have suggested (ρ, MCC).

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead Contact
  - Materials Availability
  - Data and Code Availability
- METHOD DETAILS
  - Dataset Construction and Curation
  - Experimental Application of Stability Prediction Tools

hotpaper.net

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.str.2020.04.003.

ACKNOWLEDGMENTS

This paper is dedicated to the memory of Zach Jacobi, a fine researcher, energetic colleague, and kind person. This work was supported by a grant awarded to E.M.M. and scholarship to K.T. by the Natural Sciences and Engineering Research Council of Canada (NSERC).

AUTHOR CONTRIBUTIONS

A.B., Z.J., and K.T. collected the data and performed the analysis. A.B. and E.M.M. designed the project. A.B., K.T., and E.M.M. wrote the paper.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: December 22, 2019
Revised: March 26, 2020
Accepted: April 6, 2020
Published: May 5, 2020

REFERENCES

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

Araya, C.L., Fowler, D.M., Chen, W., Muniez, I., Kelly, J.W., and Fields, S. (2012). A fundamental protein property, thermodynamic stability, revealed solely from large-scale measurements of protein function. Proc. Natl. Acad. Sci. U S A *109*, 16858–16863.

Barlow, K.A., ÓConchúir, S., Thompson, S., Suresh, P., Lucas, J.E., Heinonen, M., and Kortemme, T. (2018). Flex ddG: Rosetta ensemble-based estimation of changes in protein-protein binding affinity upon mutation. J. Phys. Chem. B *122*, 5389–5399.

Bava, K.A., Gromiha, M.M., Uedaira, H., Kitajima, K., and Sarai, A. (2004). ProTherm, version 4.0: thermodynamic database for proteins and mutants. Nucleic Acids Res. *32*, D120–D121.

Benedix, A., Becker, C.M., de Groot, B.L., Caflisch, A., and Böckmann, R.A. (2009). Predicting free energy changes using structural ensembles. Nat. Methods *6*, 3–4.

Berliner, N., Teyra, J., Colak, R., Garcia Lopez, S., and Kim, P.M. (2014). Combining structural modeling with ensemble machine learning to accurately predict protein fold stability and binding affinity effects upon mutation. PLoS One *9*, e107353.

Bloom, J.D., Labthavikul, S.T., Otey, C.R., and Arnold, F.H. (2006). Protein stability promotes evolvability. Proc. Natl. Acad. Sci. U S A *103*, 5869–5874.

Bommarius, A.S., and Paye, M.F. (2013). Stabilizing biocatalysts. Chem. Soc. Rev. *42*, 6534–6565.

Bornscheuer, U.T., Huisman, G.W., Kazlauskas, R.J., Lutz, S., Moore, J.C., and Robins, K. (2012). Engineering the third wave of biocatalysis. Nature *485*, 185–194.

Broom, A., Jacobi, Z., Trainor, K., and Meiering, E.M. (2017). Computational tools help improve protein stability but with a solubility tradeoff. J. Biol. Chem. *292*, 14349–14361.

Capriotti, E., Fariselli, P., and Casadio, R. (2005). I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. Nucleic Acids Res. *33*, W306–W310.

Capriotti, E., Fariselli, P., Rossi, I., and Casadio, R. (2008). A three-state prediction of single point mutations on protein stability changes. BMC Bioinformatics *9* (*Suppl 2*), S6.

Cheng, J., Randall, A., and Baldi, P. (2006). Prediction of protein stability changes for single-site mutations using support vector machines. Proteins *62*, 1125–1132.

Choi, J.-M., Han, S.-S., and Kim, H.-S. (2015). Industrial applications of enzyme biocatalysis: current status and future aspects. Biotechnol. Adv. *33*, 1443–1454.

Cohen, M., Potapov, V., and Schreiber, G. (2009). Four distances between pairs of amino acids provide a precise description of their interaction. PLoS Comput. Biol. *5*, e1000470.

Dantas, G., Kuhlman, B., Callender, D., Wong, M., and Baker, D. (2003). A large scale test of computational protein design: folding and stability of nine completely redesigned globular proteins. J. Mol. Biol. *332*, 449–460.

Darby, N.J., and Creighton, T.E. (1993). Protein Structure (IRL Press at Oxford University Press).

Davey, J.A., Damry, A.M., Euler, C.K., Goto, N.K., and Chica, R.A. (2015). Prediction of stable globular proteins using negative design with non-native backbone ensembles. Structure *23*, 2011–2021.

Dehouck, Y., Grosfils, A., Folch, B., Gilis, D., Bogaerts, P., and Rooman, M. (2009). Fast and accurate predictions of protein stability changes upon mutations using statistical potentials and neural networks: PoPMuSiC-2.0. Bioinformatics *25*, 2537–2543.

Deutsch, C., and Krishnamoorthy, B. (2007). Four-body scoring function for mutagenesis. Bioinformatics *23*, 3009–3015.

Efron, B., and Tibshirani, R.J. (1993). An Introduction to the Bootstrap (CRC Press).

Floor, R.J., Wijma, H.J., Colpa, D.I., Ramos-Silva, A., Jekel, P.A., Szymański, W., Feringa, B.L., Marrink, S.J., and Janssen, D.B. (2014). Computational library design for increasing haloalkane dehalogenase stability. Chembiochem *15*, 1660–1672.

Foit, L., Morgan, G.J., Kern, M.J., Steimer, L.R., von Hacht, A.A., Titchmarsh, J., Warriner, S.L., Radford, S.E., and Bardwell, J.C.A. (2009). Optimizing protein stability in vivo. Mol. Cell *36*, 861–871.

Frappier, V., and Najmanovich, R.J. (2014). A coarse-grained elastic network atom contact model and its use in the simulation of protein dynamics and the prediction of the effect of mutations. PLoS Comput. Biol. *10*, e1003569.

Frey, K.M., Georgiev, I., Donald, B.R., and Anderson, A.C. (2010). Predicting resistance mutations using protein design algorithms. Proc. Natl. Acad. Sci. U S A *107*, 13707–13712.

Gapsys, V., Seeliger, D., and de Groot, B.L. (2012). New soft-core potential function for molecular dynamics based alchemical free energy calculations. J. Chem. Theor. Comput. *8*, 2373–2382.

Gershenson, A., Gierasch, L.M., Pastore, A., and Radford, S.E. (2014). Energy landscapes of functional proteins are inherently risky. Nat. Chem. Biol. *10*, 884–891.

Giollo, M., Martin, A.J.M., Walsh, I., Ferrari, C., and Tosatto, S.C.E. (2014). NeEMO: a method using residue interaction networks to improve prediction of protein stability upon mutation. BMC Genomics *15* (*Suppl 4*), S7.

Goldenzweig, A., Goldsmith, M., Hill, S.E., Gertman, O., Laurino, P., Ashani, Y., Dym, O., Unger, T., Albeck, S., Prilusky, J., et al. (2016). Automated structure- and sequence-based design of proteins for high bacterial expression and stability. Mol. Cell *63*, 337–346.

Gosavi, S. (2013). Understanding the folding-function tradeoff in proteins. PLoS One *8*, e61222.

hotpaper.net

Heselpoth, R.D., Yin, Y., Moult, J., and Nelson, D.C. (2015). Increasing the stability of the bacteriophage endolysin PlyC using rationale-based FoldX computational modeling. Protein Eng Des Sel *28*, 85–92.

Huang, P.-S., Boyken, S.E., and Baker, D. (2016). The coming of age of de novo protein design. Nature *537*, 320–327.

Jacak, R., Leaver-Fay, A., and Kuhlman, B. (2012). Computational protein design with explicit consideration of surface hydrophobic patches. Proteins *80*, 825–838.

Jones, D.T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. *292*, 195–202.

Kabsch, W., and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers *22*, 2577–2637.

Kellogg, E.H., Leaver-Fay, A., and Baker, D. (2011). Role of conformational sampling in computing mutation-induced changes in protein structure and stability. Proteins *79*, 830–838.

Khersonsky, O., Kiss, G., Röthlisberger, D., Dym, O., Albeck, S., Houk, K.N., Baker, D., and Tawfik, D.S. (2012). Bridging the gaps in design methodologies by evolutionary optimization of the stability and proficiency of designed Kemp eliminase KE59. Proc. Natl. Acad. Sci. U S A *109*, 10358–10363.

Klesmith, J.R., Bacik, J.-P., Wrenbeck, E.E., Michalczyk, R., and Whitehead, T.A. (2017). Trade-offs between enzyme fitness and solubility illuminated by deep mutational scanning. Proc. Natl. Acad. Sci. U S A *114*, 2265–2270.

Koga, N., Tatsumi-Koga, R., Liu, G., Xiao, R., Acton, T.B., Montelione, G.T., and Baker, D. (2012). Principles for designing ideal protein structures. Nature *491*, 222–227.

Komor, R.S., Romero, P.A., Xie, C.B., and Arnold, F.H. (2012). Highly thermostable fungal cellobiohydrolase I (Cel7A) engineered using predictive methods. Protein Eng Des Sel *25*, 827–833.

Krivov, G.G., Shapovalov, M.V., and Dunbrack, R.L. (2009). Improved prediction of protein side-chain conformations with SCWRL4. Proteins *77*, 778–795.

Laimer, J., Hofer, H., Fritz, M., Wegenkittl, S., and Lackner, P. (2015). 'MAESTRO—multi agent stability prediction upon point mutations. BMC Bioinformatics *16*, 116.

Magliery, T.J. (2015). Protein stability: computation, sequence statistics, and new experimental methods. Curr. Opin. Struct. Biol. *33*, 161–168.

Matthews, B.W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim. Biophys. Acta *405*, 442–451.

Myers, J.L., Well, A.D., and Lorch, R.F. (2010). Research Design and Statistical Analysis (Routledge).

Nisthal, A., Wang, C.Y., Ary, M.L., and Mayo, S.L. (2019). Protein stability engineering insights revealed by domain-wide comprehensive mutagenesis. Proc. Natl. Acad. Sci. U S A *116*, 16367–16377.

Parmeggiani, F., Huang, P.-S., Vorobiev, S., Xiao, R., Park, K., Caprari, S., Su, M., Seetharaman, J., Mao, L., Janjua, H., et al. (2015). A general computational approach for repeat protein design. J. Mol. Biol. *427*, 563–575.

Parthiban, V., Gromiha, M.M., and Schomburg, D. (2006). CUPSAT: prediction of protein stability upon point mutations. Nucleic Acids Res. *34*, W239–W242.

Perez, A., Morrone, J.A., Simmerling, C., and Dill, K.A. (2016). Advances in free-energy-based simulations of protein folding and ligand binding. Curr. Opin. Struct. Biol. *36*, 25–31.

Pires, D.E.V., Ascher, D.B., and Blundell, T.L. (2014a). DUET: a server for predicting effects of mutations on protein stability using an integrated computational approach. Nucleic Acids Res. *42*, W314–W319.

Pires, D.E.V., Ascher, D.B., and Blundell, T.L. (2014b). mCSM: predicting the effects of mutations in proteins using graph-based signatures. Bioinformatics *30*, 335–342.

Pokala, N., and Handel, T.M. (2005). Energy functions for protein design: adjustment with protein-protein complex affinities, models for the unfolded state, and negative design of solubility and specificity. J. Mol. Biol. *347*, 203–227.

Procko, E., Berguig, G.Y., Shen, B.W., Song, Y., Frayo, S., Convertine, A.J., Margineantu, D., Booth, G., Correia, B.E., Cheng, Y., et al. (2014). A computationally designed inhibitor of an Epstein-Barr viral Bcl-2 protein induces apoptosis in infected cells. Cell *157*, 1644–1656.

Rees, D.C., and Robertson, A.D. (2001). Some thermodynamic implications for the thermostability of proteins. Protein Sci. *10*, 1187–1194.

Rocklin, G.J., Chidyausiku, T.M., Goreshnik, I., Ford, A., Houliston, S., Lemak, A., Carter, L., Ravichandran, R., Mulligan, V.K., Chevalier, A., et al. (2017). Global analysis of protein folding using massively parallel design, synthesis, and testing. Science *357*, 168–175.

Schymkowitz, J., Borg, J., Stricher, F., Nys, R., Rousseau, F., and Serrano, L. (2005). The FoldX web server: an online force field. Nucleic Acids Res. *33*, W382–W388.

Sheldon, R.A., and Woodley, J.M. (2018). Role of biocatalysis in sustainable chemistry. Chem. Rev. *118*, 801–838.

Sievers, F., Wilm, A., Dineen, D., Gibson, T.J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., et al. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. Mol. Syst. Biol. *7*, 539.

Song, X., Wang, Y., Shu, Z., Hong, J., Li, T., and Yao, L. (2013). Engineering a more thermostable blue light photo receptor *Bacillus subtilis* YtvA LOV domain by a computer aided rational design method. PLoS Comput. Biol. *9*, e1003129.

Stefl, S., Nishi, H., Petukh, M., Panchenko, A.R., and Alexov, E. (2013). Molecular mechanisms of disease-causing missense mutations. J. Mol. Biol. *425*, 3919–3936.

Stein, A., Fowler, D.M., Hartmann-Petersen, R., and Lindorff-Larsen, K. (2019). Biophysical and mechanistic models for disease-causing protein variants. Trends Biochem. Sci. *44*, 575–588.

Tian, J., Woodard, J.C., Whitney, A., and Shakhnovich, E.I. (2015). Thermal stabilization of dihydrofolate reductase using Monte Carlo unfolding simulations and its functional consequences. PLoS Comput. Biol. *11*, e1004207.

Tokuriki, N., Stricher, F., Serrano, L., and Tawfik, D.S. (2008). How protein stability and new functions trade off. PLoS Comput. Biol. *4*, e1000002.

Trainor, K., Broom, A., and Meiering, E.M. (2017). Exploring the relationships between protein sequence, structure and solubility. Curr. Opin. Struct. Biol. *42*, 136–146.

Truppo, M.D. (2017). Biocatalysis in the pharmaceutical industry: the need for speed. ACS Med. Chem. Lett. *8*, 476–480.

Tzul, F.O., Schweiker, K.L., and Makhatadze, G.I. (2015). Modulation of folding energy landscape by charge-charge interactions: linking experiments with computational modeling. Proc. Natl. Acad. Sci. U S A *112*, E259–E266.

Wickstrom, L., Gallicchio, E., and Levy, R.M. (2012). The linear interaction energy method for the prediction of protein stability changes upon mutation. Proteins *80*, 111–125.

Wijma, H.J., Floor, R.J., Jekel, P.A., Baker, D., Marrink, S.J., and Janssen, D.B. (2014). Computationally designed libraries for rapid enzyme stabilization. Protein Eng Des Sel *27*, 49–58.

Wimley, W.C., Creamer, T.P., and White, S.H. (1996). Solvation energies of amino acid side chains and backbone in a family of host-guest pentapeptides. Biochemistry *35*, 5109–5124.

Worth, C.L., Preissner, R., and Blundell, T.L. (2011). SDM—a server for predicting effects of mutations on protein stability and malfunction. Nucleic Acids Res. *39*, W215–W222.

Yang, D.F., Wei, Y.T., and Huang, R.B. (2007). Computer-aided design of the stability of pyruvate formate-lyase from Escherichia coli by site-directed mutagenesis. Biosci Biotechnol Biochem *71*, 746–753.

Yang, Y., and Zhou, Y. (2008). Ab initio folding of terminal segments with secondary structures reveals the fine difference between two closely related all-atom statistical energy functions. Protein Sci. *17*, 1212–1219.

Yin, S., Ding, F., and Dokholyan, N.V. (2007). Eris: an automated estimator of protein stability. Nat. Methods *4*, 466–467.

hotpaper.net

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and Algorithms		
SDM	Worth et al. (2011)	http://biosig.unimelb.edu.au/duet/stability
FoldX	Schymkowitz et al. (2005)	http://foldxsuite.crg.eu/
IMutant2	Capriotti et al. (2005)	http://folding.biofold.org/i-mutant/i-mutant2.0.html
EGAD	Pokala and Handel (2005)	Source code from author
CUPSAT	Parthiban et al. (2006)	http://cupsat.tu-bs.de/
MuPro	Cheng et al. (2006)	http://mupro.proteomics.ics.uci.edu/
Eris	Yin et al. (2007)	https://dokhlab.med.psu.edu/eris/login.php
MultiMutate	Deutsch and Krishnamoorthy (2007)	http://www.math.wsu.edu/math/faculty/bkrishna/DT/Mutate/
IMutant3	Capriotti et al. (2008)	http://gpcr2.biocomp.unibo.it/cgi/predictors/I-Mutant3.0/I-Mutant3.0.cgi
DFire	Yang and Zhou (2008)	https://sparks-lab.org/downloads/(DFire2 was used)
PoPMuSiC	Dehouck et al. (2009)	http://dezyme.com/en/Software (may no longer be free)
Hunter	Cohen et al. (2009)	http://bioinfo41.weizmann.ac.il/hunter/
Rosetta-ddG	Kellogg et al. (2011)	https://www.rosettacommons.org/software/
mCSM	Pires et al. (2014a)	http://biosig.unimelb.edu.au/duet/stability
DUET	Pires et al. (2014b)	http://biosig.unimelb.edu.au/duet/stability
NeEMO	Giollo et al. (2014)	http://protein.bio.unipd.it/neemo/
ENCoM	Frappier and Najmanovich (2014)	https://github.com/NRGlab/ENCoM
MAESTRO	Laimer et al. (2015)	https://pbwww.che.sbg.ac.at/?page{_}id=477
BioLuminate	Commercial software	https://www.schrodinger.com/products/bioluminate
CC/PBSA	Benedix et al. (2009)	Webserver no longer available, data used was from publication
LIE	Wickstrom et al. (2012)	No Webserver or code available, data used was from publication
SCWRL4	Krivov et al. (2009)	http://dunbrack.fccc.edu/SCWRL3.php/
VMD		http://www.ks.uiuc.edu/Research/vmd/
DSSP	Kabsch and Sander (1983)	https://swift.cmbi.umcn.nl/gv/dssp/DSSP_3.html
PsiPred	Jones (1999)	https://github.com/psipred/psipred
pBLAST	Altschul et al. (1990)	https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastDocs&DOC_TYPE=Download
Clustal Omega	Sievers et al. (2011)	https://www.ebi.ac.uk/Tools/msa/clustalo/
scikit-learn		https://scikit-learn.org/stable/
Other		
Protherm database	Bava et al. (2004)	https://www.iitm.ac.in/bioinfo/ProTherm/
Web of Science		https://clarivate.com/webofsciencegroup/solutions/web-of-science/

RESOURCE AVAILABILITY

Lead Contact

Further information and requests for resources should be directed to and will be fulfilled by the Lead Contact, Elizabeth M. Meiering (meiering@uwaterloo.ca).

Materials Availability

This study did not generate new materials.

hotpaper.net

Data and Code Availability

The published article includes all datasets generated or analyzed during this study.

METHOD DETAILS

Dataset Construction and Curation

For testing tools against point mutations with experimentally determined stability we used a previously curated database of point mutations (Table S1) (Broom et al., 2017). Briefly, unique point mutations with measured $\Delta\Delta G$ values from the Protherm database (Bava et al., 2004) where a crystal structure of the wild-type was available, and the experimental pH was between 5 and 9, were selected. From these, only those where the $\Delta\Delta G$ value was determined at, or could be extrapolated to, a temperature between 20 and 30°C, were kept. Finally, proteins with cofactors or prosthetic groups were removed (unless the experimental conditions specifically were done in the apo state). Manual inspection of the primary citations for all remaining 605 point mutations was performed in order to correct cases where the $\Delta\Delta G$ value was entered into the Protherm database with the incorrect sign (a known problem (Bava et al., 2004)) or where the value from the primary citation had been entered with incorrect units (kJ/mol versus kcal/mol), which we found to happen in several cases.

Experimental Application of Stability Prediction Tools

Construction of a 270 point mutant dataset was made by exhaustively examining forward-citations (using Web of Science, www.webofknowledge.com) for each of the 20 published tools, and recording all cases where a stability prediction tool was used to recommend a point mutation that was later tested experimentally (see Table S2).

Stability Prediction by Individual Tools

The following individual tools were used by supplying the wild-type (WT) PDB structure and desired point mutation (experimental temperature and pH where applicable): BioLuminate (stand-alone application, version 2.1) (www.schrodinger.com/products/bioluminate), CUPSAT (Parthiban et al., 2006) (webserver: cupsat.tu-bs.de/), DUET (Pires et al., 2014a) (webserver: biosig.unimelb.edu.au/duet/stability), EGAD (Pokala and Handel, 2005) (stand-alone application), ENCoM (Frappier and Najmanovich, 2014) (stand-alone application), Eris (Yin et al., 2007) (webserver: redshift.med.unc.edu/eris/login.php), FoldX (Schymkowitz et al., 2005) (stand-alone application, version 3.0), Hunter (Cohen et al., 2009) (stand-alone application), IMutant3 (Capriotti et al., 2008) (stand-alone application, version 3.0.7), IMutant2 (Capriotti et al., 2005) (stand-alone application), MAESTRO (Laimer et al., 2015) (stand-alone application), mCSM (Pires et al., 2014b) (webserver: biosig.unimelb.edu.au/mcsm/), NeEMO (Giollo et al., 2014) (webserver: protein.bio.unipd.it/neemo/), MultiMutate (Deutsch and Krishnamoorthy, 2007) (stand-alone application), MuPro (Cheng et al., 2006) (stand-alone application, version 1.1), PoPMuSiC (Dehouck et al., 2009) (webserver, version 2.1: dezyme.com/), SDM (Worth et al., 2011) (webserver: mordred.bioc.cam.ac.uk/sdm/sdm.php).

In the case of DFire, WT and mutant structures were generated using SCWRL4 (Krivov et al., 2009) followed by energy evaluation using DFIRE2 (Yang and Zhou, 2008) (stand-alone application, version 1.1) with the $\Delta\Delta G$ computed by taking the difference between the mutant and WT energy evaluations (ΔG_{Mutant} - ΔG_{WT}). In the case of CC/PBSA the webserver is no longer available, and a 581 point mutation dataset was constructed from reported predictions (Benedix et al., 2009) after filtering based on the criteria used in our primary dataset. In the case of LIE, the computational procedure has not been fully automated and a 822 point mutation dataset was constructed from reported predictions (see method 2 in Wickstrom et al. (Wickstrom et al., 2012)) after filtering based on the criteria used in our primary dataset. Despite using somewhat different datasets, the dataset sizes and composition are similar in all cases suggesting the results are comparable to the other tools used herein. As both of these tools represent the use of physical forcefields (e.g. molecular mechanics forcefields) which are rare in current stability prediction, they represent important points of comparison.

Training SimpleMachine

Simple machine was trained on a dataset of 1058 point mutations that were not part of our 605 point mutation test set, yet had been used in the training of several other tools (Broom et al., 2017). The machine learning architecture was a feed-forward neural network with 4 input neurons, 2 hidden layer neurons, and a single output. The 4 inputs, which were computed for each point mutation were: **1)** change in amino acid polarity, measured by $\Delta\Delta$G of solvation between the mutant and wild-type (WT) sidechains as determined by data reported by Wimley et al. (Wimley et al., 1996), **2)** change in amino acid size between the mutant and WT, measured in $Å^3$ as determined by data reported by Darby and Creighton (Darby and Creighton, 1993), **3)** solvent accessible surface area of the WT residue, measured in $Å^2$ (determined by VMD, http://www.ks.uiuc.edu/Research/vmd/), and **4)** change in secondary structure propensity relative to the native structure. The change in secondary structure propensity relative to the native structure was determined by estimating the secondary structure propensity of the wild-type and mutant residue using PsiPred (Jones, 1999) (without the use of BLAST so as to not bias the estimate with known sequence information) and compared to the native secondary structure in the WT PDB, determined using DSSP (Kabsch and Sander, 1983). If both the mutated and WT residue were predicted to adopt the secondary structure present in the native PDB, the score was calculated as: $Propensity_{Mutant}$ - $Propensity_{WT}$, thus a positive score when the mutation increases the propensity of the existing secondary structure, and negative otherwise. If the mutated residue was predicted to adopt the native secondary structure but the WT was not, the score was calculated as: $Propensity_{Mutant}$ + $Propensity_{WT}$, thus a positive score. If WT was predicted to adopt the native secondary structure but the mutated residue was not, the score was

hotpaper.net

calculated as: -(Propensity$_{Mutant}$ + Propensity$_{WT}$), thus a negative score. Finally, if both the mutated and WT residue were predicted to adopt secondary structures other than the native secondary structure, the score was 0. All inputs were normalized to be between 0 and 1 based on the full range of data available in the training set. The network was trained to minimize the average unsigned error between predictions and the experimentally determined $\Delta\Delta G$ values for the 1058 point mutation training set. The **MLPRegressor** class of the **neural_network** module from scikit-learn (www.scikit-learn.org) was used with the default parameters except that the minimization solver used was stochastic gradient descent, and the regularization parameter α was set to 0.1.

Prediction of Sequence Preference from Consensus

In order to predict the $\Delta\Delta G$ of a mutation based on sequence information (e.g. a consensus $\Delta\Delta G$), the WT sequence was submitted to pBLAST (Altschul et al., 1990) and any sequences with an E-value < 0.001: were kept (up to a maximum of 10,000 sequences) and a multiple-sequence alignment (MSA) generated using Clustal Omega (Sievers et al., 2011). The expected $\Delta\Delta G$ was computed as:

$$-RT \times \ln(frequency_{aminoacid} / 0.05) \quad \text{(Equation 1)}$$

where R is 0.001987 kcal mol^{-1} K^{-1} and T was 298.15 K. Here, no removal of highly redundant sequences was performed, nor were background probabilities corrected for the codon usage or overall amino acid frequencies, nor were MSAs manually curated. Any of the previously mentioned modifications may improve the quality of the MSA and thus reliability of the $\Delta\Delta G$ predictions (Magliery, 2015), but here we were primarily interested in whether consensus prediction could be useful in counter-balancing the stability prediction tools' tendency to predict hydrophobic surface mutations as stabilizing.

Modelling Expected Stabilities and Success Rates for Single and Multiple Mutations

The data shown in Figure 5 are based on assuming a hypothetical ''average'' stability prediction tool is used for engineering stabilizing point or multiple mutants. To model the behaviour of such a tool we looked at the experimentally determined $\Delta\Delta G$ values for mutations predicted to stabilize by the existing tools (Figure 1B, grey distributions) and chose a typical representative. The effect of any mutations in our model were then chosen at random for a normal distribution with mean and standard deviation matching that of the typical tool. In particular, for the typical tool, where mutations predicted to stabilize have a mean $\Delta\Delta G$ of 0.0 kcal/mol and standard deviation of 1.4 kcal/mol, a point mutation has an ~2% likelihood of increasing stability by 3 kcal/mol, whereas a multi-mutant with 5 or 20 positions changed has an ~15% or ~33% chance of this stability increase, respectively (Figures 5A and 5B). When modeling the effect of improving stability prediction tools, the standard deviation was left unchanged but the mean was altered and mutations again drawn at random from a normal distribution. Here for instance, increasing the mean experimental $\Delta\Delta G$ for mutations predicted to stabilize from 0.0 kcal/mol to 0.1 kcal/mol changes the success rate of increasing stability by 10 kcal/mol when making 20 mutations from ~4% to ~10% (Figures 5C and 5D).

QUANTIFICATION AND STATISTICAL ANALYSIS

Performance Evaluation Metrics

Pearson's R, Spearman's ρ, and the Standard Error were computed as defined in standard texts (Myers et al., 2010). For computation of Accuracy, any mutation with an experimental $\Delta\Delta G$ value between -0.3 and 0.3 kcal/mol was ignored. Overall Accuracy was calculated as the number of predictions that successfully classified the mutation as stabilizing versus destabilizing divided by the total number of predictions. In the case of Accuracy for highly and moderately destabilizing mutations and moderately stabilizing mutations, only those mutations with experimental $\Delta\Delta G$ values with the ranges: -6 to -3 (exclusive), -3 (inclusive) to -0.3, and 0.3 to 3.0 kcal/mol were used, respectively. Calculation of MCC was also as standard in the literature (Matthews, 1975), but as with Accuracy any mutation with an experimental $\Delta\Delta G$ value between -0.3 and 0.3 kcal/mol was not included. Mutations with $\Delta\Delta G$ values in the range -0.3 to 0.3 kcal/mol were ignored in the above classification metrics because typical experimental error in determining $\Delta\Delta G$ is ~0.3 kcal/mol (Pokala and Handel, 2005). For calculation of hydrophobicity, the $\Delta\Delta G$ of solvation was calculated as the difference in ΔG of solvation between the WT and mutant sidechain (Wimley et al., 1996). In all measurement cases the reported values and plotted error-bars were determined by bootstrapping (Efron and Tibshirani, 1993) the data (re-sampling with replacement) 1000 times and computing the average and standard deviations.