A statistical model for improved membrane protein expression using sequence-derived features

Shyam M. Saladi, Nauman Javed, Axel Müller, and William M. Clemons, Jr.

From the Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, California 91125

Edited by Karen G. Fleming

The heterologous expression of integral membrane proteins (IMPs) remains a major bottleneck in the characterization of this important protein class. IMP expression levels are currently unpredictable, which renders the pursuit of IMPs for structural and biophysical characterization challenging and inefficient. Experimental evidence demonstrates that changes within the nucleotide or amino acid sequence for a given IMP can dramatically affect expression levels, yet these observations have not resulted in generalizable approaches to improve expression levels. Here, we develop a data-driven statistical predictor named IMProve that, using only sequence information, increases the likelihood of selecting an IMP that expresses in *Escherichia coli*. The IMProve model, trained on experimental data, combines a set of sequence-derived features resulting in an IMProve score, where higher values have a higher probability of success. The model is rigorously validated against a variety of independent data sets that contain a wide range of experimental outcomes from various IMP expression trials. The results demonstrate that use of the model can more than double the number of successfully expressed targets at any experimental scale. IMProve can immediately be used to identify favorable targets for characterization. Most notably, IMProve demonstrates for the first time that IMP expression levels can be predicted directly from sequence.

The biological importance of integral membrane proteins (IMPs) motivates structural and biophysical studies that require large amounts of purified protein at considerable cost. Only a small percentage can be produced at high levels, resulting in IMP structural characterization lagging far behind that of soluble proteins; IMPs currently constitute less than 2% of deposited atomic-level structures. To increase the pace of structure determination, the scientific community created large government-funded structural genomics consortia facilities, like the National Institutes of Health–funded New York Consortium on Membrane Protein Structure (NYCOMPS). For this representative example, more than 8000 genes, chosen based on characteristics hypothetically related to success, yielded only 600 (7.1%) highly expressing proteins, resulting in a success rate of 5.6% of expressed proteins (based on annotation in the Protein Data Bank). This example highlights the funnel problem of structural biology, where each stage of the structure pipeline eliminates a large percentage of targets, compounding into an overall low rate of success. With new and rapidly advancing technologies like cryo-electron microscopy, serial femtosecond crystallography, and micro-electron diffraction, we expect that the latter half of the funnel, structure determination, will increase in success rate. However, IMP expression levels will continue to limit targets accessible for study.

Tools are needed for improving the number of IMPS with successful heterologous expression (we use expression synonymously for the term “overexpression” in the field). Whereas significant work has shown promise on a case-by-case basis (e.g. growth at lower temperatures, codon optimization, 5′-UTR randomization, and regulating transcription), a generalizable solution remains elusive. Several RNA structure prediction methods have been used to explain or predict soluble protein expression levels. For individual IMPS, simple changes can have dramatic effects on the amount of expressed proteins, only partially explained by computational methods in the case of modifications at the 5′ region. Broadly, each new IMP target must be addressed individually, as the conditions that were successful for a previous target seldom carry over to other proteins, even among closely related homologs.

Considering the scientific value of IMP studies, it is surprising that there are no methods that can provide solutions for improved expression outcomes with broad applicability across protein families and genomes. There are no approaches that can decode sequence-level information for predicting IMP expression, yet it is common knowledge that sequence changes that alter overall biophysical features of the protein and mRNA transcript can measurably influence IMP biogenesis. Whereas physics-based approaches have proven successful in correlating
integration efficiency and expression (18, 19), that and other work revealed that simple application of specific “sequence features” is inadequate to predict IMP expression (20, 21). As an example, whereas the positive-inside rule is an important indicator of proper IMP biogenesis, the number of positive charges on cytoplasmic loops alone does not predict expression level (22, 23). The reasons for this failure to connect sequence to expression probably lie in the complex underpinnings of IMP biogenesis, where the interplay between many sequence features at both the protein and nucleotide levels must be considered. Optimizing for a single sequence feature probably diminishes the beneficial effect of other features (e.g.increasing TM hydrophobicity might diminish favorable mRNA properties). Without accounting for the broad set of sequence features related to IMP expression, it is impossible to predict differences in expression levels.

Development of a low-cost, computational resource that significantly and reliably predicts improved expression outcomes would transform the study of IMPs. Attempts to develop such algorithms have so far failed. Several examples (10, 20, 21) as well as NYCOMPS3 were unable to use experimental expression data sets to train models that returned any predictive performance. This is not surprising, given the difficulty of expressing IMPs and the limits in the knowledge of the sequence features that drive expression. In other contexts, statistical tools based on sequence have been shown to work (e.g. those developed to predict soluble protein expression and/or crystallization propensities) (13, 24, 25). Such predictors are primarily based on available experimental results from the Protein Structure Initiative (26, 27). Whereas collectively, these methods have supported significant advances in biochemistry, none of the models are able to predict IMP outcomes due to limitations inherent in the model development process. As IMPs have an extremely low success rate, they are either explicitly excluded from the training process or are implicitly down-weighted by the statistical model (for representative methodology, see Ref. 28). Consequently, none have successfully been able to map IMP expression to sequence.

Here, we demonstrate for the first time that it is possible to predict IMP expression directly from sequence. The resulting predictor allows one to enrich expression trials for proteins with a higher probability of success. To connect sequence to prediction, we develop a statistical model that maps a set of sequences to experimental expression levels via calculated features, thereby simultaneously accounting for the many potential determinants of expression. The resulting IMProve model allows ranking of any arbitrary set of IMP sequences in order of their relative likelihood of successful expression. The IMProve model is extensively validated against a variety of independent data sets, demonstrating that it can be used broadly to predict the likelihood of expression in *Escherichia coli* of any IMP. With IMProve, we have built a way for more than 2-fold enrichment of positive expression outcomes relative to the rate attained from the current method of randomly selecting targets. We highlight how the model informs on the biological underpinnings that drive likely expression. Finally, we provide direct examples where the model can be used for a typical researcher. Our novel approach and the resulting IMProve model provide an exciting paradigm for connecting sequence space to complex experimental outcomes.

**Results**

For this study, we focus on heterologous expression in *E. coli*, due to its ubiquitous use as a host across the spectrum of the membrane proteome. Low cost and low barriers for adoption highlight the utility of *E. coli* as a broad tool if the expression problem can be overcome. Furthermore, proof-of-principle in *E. coli* will illustrate methodology for constructing similar predictive methods for expressing IMPs in other hosts (e.g. yeast, insect cells).

**Development of a computational model trained on E. coli expression data**

A key component of any data-driven statistical model is the choice of data set used for training. Having searched the literature, we identified two publications that contained quantitative data sets on overexpression of *E. coli* polytopic IMPs in *E. coli*. The first set, from Daley and Rapp et al. (20), contained activity measures, proxies for expression level, from C-terminal tags of either GFP or PhoA (alkaline phosphatase). The second set, from Fluman *et al.* (29), used a subset of constructs from the first and contained a more detailed analysis utilizing in-gel fluorescence to measure folded protein (see “Materials and methods” 4c). The expression levels strongly correlated (Spearman’s $\rho = 0.73$) between the two data sets, demonstrating that normalized GFP activity was a good measure of the amount of folded IMP (Fig. 1A) (29, 30). The experimental setup employed multiple 96-well plates over multiple days, resulting in pronounced variability in the absolute expression level of a given protein between trials. Daley and Rapp *et al.* (20) calculated average expression levels by dividing the raw expression level of each protein by that of a control protein on the corresponding plate.

To successfully map sequence to expression, we additionally needed to derive numerical features from a given gene sequence that have been empirically related to expression. 87 sequence features from protein and nucleotide sequence were calculated for each gene using custom code together with published software (codonW (31), tAI (32), NUPACK (33), Vienna RNA (34), Codon Pair Bias (35), Disembl (36), and RONN (37)). Relative metrics (e.g. codon adaptation index) are calculated with respect to the *E. coli* K–12 substrate MG1655 (38) quantity. The octanol-water partitioning (39), GES hydrophobicity (40), and $\Delta G$ of insertion (41) scales were employed as well. Transmembrane segment topology was predicted using Phobius constrained for the training data and Phobius for all other data sets (42). Two RNA secondary structure metrics were prompted in part by Goodman *et al.* (14). Table S1 includes a detailed description of each feature. All features are calculated solely from the coding region of each gene of interest excluding other portions of the open reading frame and plasmid (e.g. linkers and tags, 5’-untranslated region, copy number).

---

3 M. Punta, personal communication.
Fitting the data to a simple linear regression provides a facile method for deriving a weight for each feature. However, using the set of sequence features, we were unable to successfully fit a linear regression using the normalized GFP and PhoA measurements reported in the Daley and Rapp et al. (20) study. Similarly, using the same feature set and data, we were unable to train a standard linear support vector machine (SVM) to predict the expression data either averaged or across all plates (see Table S1 and “Materials and methods” 2 and 3). Due to the attempts by others to fit these data, this outcome may not be surprising and suggested that a more complex analysis was required.

We hypothesized that training on relative measurements across the entire data set introduced errors that were limiting. To address this, we instead only compare measurements within an individual plate, where differences between trials are less likely to introduce errors. To account for this, a preference-ranking linear SVM algorithm (SVMrank (43)) was chosen (see “Materials and methods” 4b). Simply put, the SVMrank algorithm determines the optimal weight for each sequence feature to best rank the order of expression outcomes within each plate over all plates, which results in a model where higher-expressing proteins have higher scores. The outcome is identical in structure to a multiple linear regression, but instead of minimizing the sum of squared residuals, the SVMrank cost function accounts for the platewise constraint specified above. In practice, the process optimizes the correlation coefficient Kendall’s $\tau$ (Equation 1) to converge upon a set of weights.

$$\tau_{\text{kendall}} = \frac{\text{No. of correctly ordered pairs} - \text{no. of swapped pairs}}{\text{No. of total pairs}}$$

(Eq. 1)

The SVMrank training metric shows varying agreement for all three groups (i.e. $\tau_{\text{kendall}} > 0$) (Fig. 1B). For individual genes, activity values normalized and averaged across trials were not directly used for the training procedure (see “Materials and methods” 4a), yet one would anticipate that scores for each gene should broadly correlate with the expression average. Indeed, the observed normalized activities positively correlate with the score (dubbed the IMProve score for integral membrane protein expression improvement) output by the model (Fig. 1C, $\rho > 0$). Because SVMrank transforms raw expression levels within each plate to ranks before training, Spearman’s $\rho$, a rank correlation coefficient describing the agreement between two ranked quantities, is better suited for quantifying correlation over more common metrics like the $R^2$ of a regression and Pearson’s $r$. Whereas these metrics demonstrate that the model can rank expression outcomes across all proteins in the training set, for PhoA-tagged proteins, the model appears less successful. The implications for this are discussed later (see Fig. 2G).

**Demonstration of prediction against an independent large expression data set**

Whereas the above analyses show that the model successfully fits the training data, we assess the broader applicability of the model outside the training set based on its success at predicting...
A membrane protein expression predictor

Figure 2. Success of the model against outcomes from NYCOMPS. A, an overview of the NYCOMPS outcomes; B, a histogram of the number of conditions tested per gene colored based on outcome. C, histograms of genes with Only Positive (red) and Only Negative outcomes (gray) across IMProve scores (binned as described in “Materials and methods” 5). The percentage of Only Positive outcomes in each bin is overlaid as a brown line (right axis). Yellow lines indicate the 75th (dashed) and 91st (solid) percentiles in IMProve score (> 0.2 and 0.5, respectively). D, receiver operating characteristics for positive groupings given by Only Positive outcome genes (red) and genes with at least one positive outcome (pink). The percent positive for each group (corresponding color), total counts (black), and AUC values with 95% CI are shown. The ROC considering genes with Mixed outcomes only as positive is shown as a blue dashed line with an AUC of 53.5 (51.8–55.2). The gray dashed line shows the performance of a completely random predictor (AUC = 50). E, the AUCs for outcomes across all trials and within the most-tested plasmids along with 95% CI. Performances are also split by predicted C-terminal localization (42). The numbers below indicate the total number of trials for each group and the percentage within that group that were positive. F, the NYCOMPS data set split by the presence or absence of a Pfam family in the training set with AUCs calculated by considering Only Positive genes as positive outcomes.

the outcomes of independent expression trials from distinct groups and across varying scales. The first test considers results from NYCOMPS, where 8444 IMP genes entered expression trials, in up to eight conditions, resulting in 17,114 expression outcomes (Fig. 2A) (2). The majority of genes were attempted in only one condition (Fig. 2B), and, importantly, outcomes were non-quantitative (binary: expressed or not expressed) as indicated by the presence of a band by Coomassie staining of an SDS-PAGE gel after small-scale expression, solubilization, and nickel-affinity purification (3). For this analysis, the experimental results are either summarized as outcomes per gene or broken down as raw outcomes across defined expression conditions. For outcomes per gene, we can consider various thresholds for considering a gene as positive based on NYCOMPS expression success (Fig. 2B). The most stringent threshold only regards a gene as positive if it has no negative outcomes (Fig. 2B, Only Positive, red). Because a well-expressing gene would generally advance in the NYCOMPS pipeline without further small-scale expression trials, this positive group probably contains the best-expressing proteins. A second category comprises genes with at least one positive and at least one negative trial (Fig. 2B, Mixed, blue). These genes probably include proteins that are more difficult to express.

A histogram of the IMProve scores for genes separated by expression success shows the predictive power of the IMProve model (Only Positive, red; Mixed, blue; Only Negative, gray) (Fig. 2C). Visually, the distribution of the scores for the Only Positive group is shifted to a higher score relative to the Only Negative and Mixed groups. The dramatic increase in the percentage of Only Positive genes as a function of increasing IMProve score (overlaid as a brown line) further emphasizes this. In fact, simply selecting the top 25% of proteins by IMProve score (−0.2 or higher; Fig. 2C, yellow dotted line) would have increased the number of positive outcomes from 11 to 20%, a 1.8-fold improvement.

Although this is suggestive, we aim to more systematically assess predictive power. For NYCOMPS and subsequent data sets, the outcomes are either binary (e.g. in NYCOMPS expressed or not expressed) or categorical (e.g. arbitrary ranked scale), correlation coefficients cannot be used here. Instead, we turn to an alternative framework: the receiver operating characteristic (ROC). ROC curves quantify the tradeoff between true positive and false positive predictions across numerical scores from a predictor. The figure of merit is the area under the ROC curve (AUC), where a perfect predictor has an AUC = 100% and a random predictor has an AUC = 50% (Fig. 2D, gray dashed line). An AUC > 50% indicates that a model is predictive (percentage signs hereafter omitted) (see “Materials and methods” 5 and Ref. 44). We will use the ROC framework to quantitatively validate the ability of our model to predict protein expression data, all independent of the training process.

Returning to the NYCOMPS data, ROC analysis shows that the IMProve model markedly distinguishes genes in the most stringent positive group (Only Positive) from all other genes (AUC = 67.1) (Fig. 2C, red). A permissive threshold considering genes as positive with at least one positive trial (Only Positive plus Mixed genes) shows moderate predictive power (Fig. 2C, pink), AUC = 59.7). If instead the Mixed genes are considered alone (excluding the Only Positive), the model very weakly distinguishes the Mixed group from the Only Negative genes (Fig. 2D, dashed blue, AUC = 53.5). This probably supports the notion that the Mixed pool largely consists of more difficult-to-express genes.

We next confirm the ability of the IMProve model to predict within plasmids or sequence space distinct from those within the limited training set. For an overfit model, one might expect
that only the subset of targets that most closely mirror the training data would show strong prediction. On the contrary, the model shows consistent performance throughout each of the eight distinct experimental conditions tested (Fig. 2G and Table S2). One may also consider that the small size of the training set limited the number of protein folds sampled and, therefore, limited the number of folds that could be predicted by the model. To test this, we consider the performance of the model with regard to protein homology families, as defined by Pfam family classifications (45). The 8444 genes in the NYCOMPS data set fall into 555 Pfam families (~15% not classified). To understand whether the IMProve score is biased toward families present in the training set, we separate genes in the NYCOMPS data set into those either within the 153 Pfam families found in the training set or outside this pool (i.e. not in the training set). Satisfyingly, there is no significant difference in AUC at 95% confidence between these groups (68.2 versus 67.2) (Fig. 2H). Combined, this highlights that the model is not sensitive to the experimental design of the training set and predicts broadly across different vector backbones and protein folds.

The ability to predict the experimental data from NYCOMPS allows returning to the question of alkaline phosphatase as a metric for expression. For the training set, proteins with C termini in the periplasm show weaker fitting by the model (Fig. 1, orange). To assess the generality of this result, the NYCOMPS outcomes are split into pools for either cytoplasmic or periplasmic C-terminal localization, and AUCs are calculated for each. There are no significant differences in predictive capacity across all conditions (Fig. 2G, green versus orange) irrespective of whether the tag is at the N or C terminus. This demonstrates that the IMProve model is applicable for all topologies.

At this point, it is useful to consider the potential improvement in the number of positive outcomes by using the IMProve model. NYCOMPS tested about a tenth of the 74,000 possible IMPs from the 98 genomes of interest for expression (2). Had NYCOMPS tested the same number of genes from this pool, but selected to have an IMProve score >0.5 (at the 91st percentile (Fig. 2C, yellow solid line)), they would have increased their positive pool of 934 by an additional 1207 proteins. This represents a >2-fold improvement in the return on investment and is a clear benchmark of success for the IMProve model.

Further demonstration of prediction against small-scale independent data sets

The NYCOMPS example demonstrates the predictive power of the model across the broad range of sequence space encompassed by that data set. Next, the performance of the model is tested against relevant subsets of sequence space (e.g. a family of proteins or the proteome from a single organism), which are reminiscent of laboratory-scale experiments that precede structural or biochemical analyses. Whereas a number of data sets exist (5, 46–57), we identified seven for which complete sequence information could be obtained to calculate all of the necessary sequence features (46–52).

To understand the predictive performance within each of the small-scale data sets, we analyze the predictive performance of the model and highlight how the model could have been used to streamline those experiments. The clear predictive performance within the large-scale NYCOMPS data set (Fig. 2) serves as a benchmark of expected performance at the scale of the experimental efforts for an individual laboratory (Fig. 3A). As targets within the various data sets were tested only one or a few times, experimental variability within each set could play a large role on the outcomes noted. Therefore, we summarize positives within each data set as those genes with the highest level of outcome as reported by the original authors, as this outcome is probably most robust to such variability (e.g. seen via Coomasie Blue staining of an SDS-polyacrylamide gel). To be complete, we have plotted and considered predictive performance across all possible outcomes as well (Fig. 3 (B–D) and Fig. S1).

The performance of the IMProve model for each of the small-scale data sets is consistent with that seen for the NYCOMPS data set (Fig. 3A). This is most directly indicated by a mean AUC across all data sets of 65.6, highlighting the success across the varying scales. Whereas the overall positive rate is different for each data set, considering a cutoff in IMProve score (e.g. the top 50 or 10% of targets ranked by score) would have resulted in a greater percentage of positive outcomes. On average, ~70% of positives are captured within the top half of scores. Similarly, for the top 10% of scores, on average, >20% of the positives are captured. Simply put, for one-tenth of the work, one would capture a significant number of the positive outcomes within the pool of targets in each data set.

For broader consideration, one can consider the -fold change in positive rate by selecting targets informed by IMProve scores. Using the data available, only testing proteins within the top 10% of scores would result in an average -fold change of 2.0 in the positive rate (i.e. twice as many positively expressed proteins). As positive rate is a bounded quantity (maximum is 100%), the possible -fold change is bounded as well and becomes relative to the overall positive rate when considering various cutoffs (e.g. for Thermotoga maritima, the maximum -fold change is 15.4, whereas for archaeal transporters, it is 3.3). Taking the average maximum possible -fold change (7.5), the IMProve model achieves nearly a third of the possible improvement in positive rate compared with a perfect predictor.

Because the IMProve model was trained on quantitative expression levels, we also expect that it captures quantitative trends in expression (i.e. a higher score translates to a greater amount of expressed protein). Whereas the NYCOMPS results are consistent with this (Fig. 2b), of the various data sets, only the expression of archaeal transporters presents quantitative outcomes for consideration. For this data set, 14 archaeal transporters were chosen based on their homology to human proteins (46) and tested for expression in E. coli; total protein was quantified in the membrane fraction by Coomasie Blue staining of an SDS-polyacrylamide gel. Here, the majority of the expressing proteins fall into the higher half of the IMProve scores, seven of the nine with multiple positive outcomes (Fig. 3B). Strikingly, quantification of the Coomasie Blue staining highlights a clear correlation with the IMProve score, where the higher-expressing proteins have higher scores (Fig. 3C).

A final test considers the ability of the model to predict expression in hosts other than E. coli. The expression trials of
101 mammalian GPCRs in bacterial and eukaryotic systems (49) provide a data set for considering this question. For this experiment, trials in *E. coli* clearly follow the trend that IMProve can predict within this group of mammalian proteins (AUC = 77.7) (Fig. 3A and Fig. S1 (A and B)). However, the expression of the same set of proteins in *Pichia pastoris* fails to show any predictive performance (AUC = 54.8) (Fig. S1, A and B). This lack of predictive performance in *P. pastoris* suggests that the parameterization of the model, calibrated for *E. coli* expression, requires retraining to generate a different model that captures the distinct interplay of sequence parameters in other hosts.

**Biological importance of various sequence features**

Considering the success of IMProve, one might anticipate that biological properties driving prediction may provide insight into IMP biogenesis and expression. To derive principles about which features drive prediction, most statistical models (including SVM*rank* as used here) require that each data point and its features be drawn identically and distributed independently from an underlying distribution (referred to as “i.i.d.”). In the case of a linear model, if the features meet the i.i.d. criteria, then extracting the importance of each feature is straightforward; the weight assigned to each feature corresponds to its relative importance. However, in our case, the input features do not meet the i.i.d. criteria because significant correlation exists between individual features (Fig. S2).

Because overlapping sets of correlating features represent biological properties, the importance of a given biological property is distributed across many features. The result is that one cannot directly determine the biological underpinnings of the IMProve score. For example, the importance of transmembrane segment hydrophobicity for membrane partitioning is probably distributed between several features; among these, the average insertion (41) of TM segments has a positive weight, whereas average hydrophobicity, a correlating feature, has a negative weight (Table S1 and Fig. S2). Although this is counterintuitive, it is consistent with the expectation that the overall importance of transmembrane segment hydrophobicity is the sum of many correlated features. Although one cannot disentangle the correlation between features, we attempt to address this complication by coarsening our view of the features to two levels; first, we analyze features derived from protein versus those derived from nucleotide sequence, and then we look more closely at feature groups after categorizing by biological property.
A membrane protein expression predictor

The coarsest view of the features is a comparison of those derived from protein sequence versus those derived from nucleotide sequence. The summed weight for protein features is around zero, whereas for nucleotide features, the summed weight is slightly positive, suggesting that in comparison, these features may be more important to the predictive performance of the model (Fig. 4A). Within the training set, protein features more completely explain the score (Fig. 4B). However, comparison of the predictive performance of the two subsets of weights shows that the nucleotide features alone can give performance similar to that of the full model for the NYCOMPS data set (Fig. 4C). Within the small-scale data sets investigated, using only protein or nucleotide features shows no significant difference in predictive power at 95% confidence (Fig. 4D). In general, this suggests that neither protein nor nucleotide features are solely responsible for successful IMP expression. However, within the context of the trained model, nucleotide features are critical for predictive performance for a large and diverse data set, such as NYCOMPS. This finding corroborates the growing body of literature indicating that the nucleotide sequence holds significant determinants of biological processes (14, 29, 58–60).

We next collapse conceptually similar features into biological categories that allow us to infer the phenomena that drive prediction. Categories are chosen empirically (e.g., the hydrophobicity group incorporates sequence features, such as average hydrophobicity, maximum hydrophobicity, ΔGinsertion), which results in a reduction in overall correlation (Fig. S3A). The full category list is provided in Table S1. To visualize the importance of each category, the collapsed weights are summarized in Fig. S3B, where each bar contains individual feature weights within a category. Features with a negative weight are stacked to the left of zero, and those with a positive weight are stacked to the right. A red dot represents the sum of all weights, and the length of the bar gives the total absolute value of the combined weights within a category. Ranking the categories based on the sum of their weight suggests that some categories play a more prominent role than others. These include properties related to transmembrane segments (hydrophobicity and TM size/count), codon pair score, loop length, and overall length/pI.

To explore the role of each biological category in prediction, the performance of the model is assessed using only features within a given category. First, the strength of the correlation coefficients for given categories within the training set suggests the relative utility of each category for prediction (Fig. S3C, as in Fig. 4B). Examples of categories with high correlation coefficients are 5’ Codon Usage, Length/pI, Loop Length, and Shine–Dalgarno (SD)-like Sites. To verify their importance for prediction, we consider the AUC for prediction using each feature category for the NYCOMPS data set (Fig. S3D). In this analysis, only Length/pI shows some predictive power. Overall, the analysis of the training and large-scale testing data set shows that no feature category independently drives the predictor. Excluding each individually does not significantly affect the overall predictive performance, except for Length/pI (data not shown). Sequence length composes the majority of the weight within this category and is one of the highest-weighted features in the model (Fig. S3A). This is consistent with the anecdotal observation that larger IMPs are typically harder to express. However, this parameter alone would not be useful for predicting within a smaller subset, like a single protein family, where there is little variance in length (e.g., Figs. 3 and 5). One might develop a predictor that was better for a given protein family under certain conditions with a subset of the entire features considered here, yet this would require a priori knowledge of the system (i.e., which sequence features were truly most impor-

---

**Figure 4. Feature contributions to the model.** A, classifying features by the type of sequence they are calculated from. B, considering the training set (as in Fig. 1), Spearman correlation coefficients with 95% CIs using individual feature categories for each grouping of data within the training set of E. coli IMPs. Colors indicate the subset being assessed (green, whole-cell GFP fluorescence; orange, alkaline phosphatase activity; purple, folded protein by in-gel fluorescence). C, AUC and 95% CIs using only protein or nucleotide features. D, protein/nucleotide feature dependence across small-scale data sets shown as AUCs of the ROC along with 95% CI for the condition with the best overall predictive power (black).
A membrane protein expression predictor

Figure 5. Usage of the model within IMP families and for optimization of expression level. A, outcomes for specific protein families with an optimal IMProve score threshold indicated. Genes are shown in the chart as dots colored based on outcomes from trials: Only Positive (red), Only Negative (gray), and Mixed (blue). Overall statistics, as in Table 3, are noted. Dashed lines represent the optimal threshold from the ROC curves. For the top two rows, each was only tested in a single condition (N: His-FLAG-tobacco etch virus gene). The bottom three rows are larger pools from NYCOMPS, where there are multiple trials for many of the genes. B, a table curated from Table S3, where Pfams were selected based on specific criteria (minimum 10 trials, 4 positive and 4 negative outcomes) and ordered by AUC. Proteins, as in A, that have known crystal structures within the family are highlighted in purple. DUFs are domains of unknown function. For context, the following Pfam families correspond to Transporter Classification Database classes: PF05425, 9.B.62; PF02667, 2.A.73; PF03595, 2.A.16; PF00999, 2.A.36; PF03601, 2.A.37; PF02537, 1.A.43; PF01757, 9.B.97; PF02378, 4.A.1, 4.A.2, 4.A.3; PF02690, 2.A.58; PF02632, 2.A.88 (95). C, a comparison of the predictive capacity of IMProve with the use of silent mutations engineered to increase anti-SD sequence binding propensity (29). The table presents experimental relative expression level (mutant over wildtype sequence) versus predictions from relative changes in either IMProve score or SD-like sites. The cells are colored as a heat map from red (lower than wildtype) to blue (higher than wildtype).
tant) and would preclude broad generalizability as shown for the IMProve model.

**Usage of the IMProve model for IMP expression**

We illustrate the IMProve model’s ability to identify promising homologs within a protein family by considering subsets of the broad range of targets tested by NYCOMPS. First, we consider two examples for protein families that do not have associated atomic resolution structures: copper resistance proteins (CopD, PF05425) and short-chain fatty-acid transporters (AtoE, PF02667). In the **first two rows of Fig. 5A**, genes from the two families are plotted by IMProve score and colored by experimental outcome. In both cases, as indicated by the AUCs of 88.2 and 80.7 (Fig. 5A), the model excels at predicting these families and provides a clear score cutoff to guide target selection for future expression experiments. For example, we expect that CopD homologs with IMProve scores above 1 will have a higher likelihood of expressing over other homologs.

We have calculated predictive performance for each Pfam found in the NYCOMPS data, which allows us to provide considerations for future experiments (Table S3). In particular, we highlight three families with many genes tested, multiple experimental trials and a spread of outcomes: voltage-dependent anion channels (PF03595), Na/H exchangers (PF00999), and glycosyltransferases (PF00535). For these, a very clear IMProve score cutoff emerges from the experimental outcomes (**dashed line** in Fig. 5A). Strikingly, for these families, the IMProve model clearly ranks the targets with Only Positive outcomes (red) at higher scores, again suggesting a preference for the best-expressing proteins (see Figs. 2 and 3). Similarly, many more families within NYCOMPS are predicted with high statistical confidence (Table 3); we provide a subset as Fig. 5B. For these, if only genes in the top 50% of IMProve score were tested, 81% of the total positives would be captured. As noted before, this is a dramatic increase in efficiency. Excitingly, many of these families remain to be resolved structurally. Considering these results with the broader experimental data sets (Fig. 3), no matter the number of proteins one is willing to test, the IMProve model enables selecting targets with a high probability of expression success in *E. coli*.

**Sequence optimization for expression**

The predictive performance of the model implies that the features defined here provide a coarse approximation of the fitness landscape for IMP expression. Attempting to optimize a single feature by modifying the sequence will probably affect the resulting score and expression due to changes in other features. Fluman et al. (29) provide an illustrative experiment. For that work, it was hypothesized that altering the number of SD-like sites in the coding sequence of a IMP would affect expression level. To test this, silent mutations were engineered within the first 200 bases of three proteins (genes *ygdD*, *brnQ*, and *ybjJ* from *E. coli*) to increase the number of SD-like sites with the goal of improving expression. Expression trials demonstrated that only one of the proteins (BrnQ) had improved expression of folded protein. Whereas the number of SD-like sites alone does not correlate, only 1 of 3, the resulting changes in the IMProve score correlate with the changes in measured expression level, 3 of 3 (Fig. 5C). The IMProve model’s ability to capture the outcomes in this small test case illustrates the utility of integrating the contribution of the numerous parameters involved in IMP biogenesis.

**Discussion**

Here, we have demonstrated a statistically driven predictor, IMProve, that decodes from sequence the sum of biological features that drive expression, a feat not previously possible (10, 21). The current best practice for characterization of an IMP target begins with the identification and testing of multiple homologs or variants for expression. The predictive power of IMProve enables this by providing a low-barrier-to-entry method to enrich for positive outcomes by >2-fold. IMProve allows for the prioritization of targets to test for expression, making more optimal use of limited human and material resources. For groups with small-scale projects, such as individual laboratories, this means that for the same cost, one would double the success rate. For large-scale groups, such as companies or consortia, IMProve can reduce by half the cost required to obtain the same number of positive results. We provide the current predictor as a web service where scores can be calculated, and the method, associated data, and suggested analyses are publically available to catalyze progress across the community.

Having shown that IMP expression can be predicted, the generalizability of the model is remarkable despite several known limitations that will be investigated in next-generation predictors. Using data from a single study for training precluded including certain features that empirically influence expression level, such as fusion tags and the context of the protein in an expression plasmid (e.g. the 5’-untranslated region), for which there was no variation in the Daley and Rapp et al. (20) data set. Moreover, using a simple proof-of-concept linear model allowed for a straightforward and robust predictor; however, intrinsically it cannot be directly related to the biological underpinnings. Whereas we can extract some biological inference, a linear combination of sequence features does not explicitly reflect the reality of physical limits for host cells. To some extent, constraint information is probably encoded in the complex architecture of the underlying sequence space (e.g. through the genetic code, TM prediction, RNA secondary structure analyses). Future statistical models that improve on these limitations will hone predictive power and more intricately characterize the interplay of variables that underlie IMP expression. By building upon the methodology here and by careful analysis of data sets in the literature, we expect to train IMP expression predictors for other host systems (e.g. eukaryotic cells).

A surprising outcome of our results is the observation of a quantitatively important contribution of the nucleotide sequence as a component of the IMProve score. This echoes the growing body of literature indicating that aspects of the nucleotide sequence are important determinants of protein biogenesis in general (14, 29, 58–60). Whereas one expects that there may be different weights for various nucleotide-derived features between soluble proteins and IMPs, it is likely that these features are important for soluble proteins as well. An example of this is the importance of codon optimization for soluble pro-
tein expression, which has failed to show any general benefit for IMPs (10). Current expression predictors that have predictive power for soluble proteins have only used protein sequence for deriving the underlying feature set (61, 62). Future prediction methods will probably benefit from including nucleotide sequence features as done here.

The ability to predict phenotypic results using sequence-based statistical models opens a variety of opportunities. As done here, this requires a careful understanding of the system and its underlying biological processes enumerated in a multitude of individual variables that impact the stated goal of the predictor, in this case enriching protein expression. As new features related to expression are discovered, future work will incorporate these, leading to improved models. This can include features derived from other approaches, such as the integration efficiency from coarse-grained molecular dynamics (18, 19) and 5′ mRNA secondary structure/translation initiation efficiency (11, 14, 15). Based on these results, expanding to new expression hosts, such as eukaryotes, seems entirely feasible, although a number of new features may need to be considered (e.g. glycosylation sites and trafficking signals). Moreover, the ability to score proteins for expressibility creates new avenues to computationally engineer IMPs for expression. The proof-of-concept described here required significant work to compile data from genomics consortia and the literature in a readily useable form. As data become more easily accessible, broadly leveraging diverse experimental outcomes to decode sequence-level information, an extension of this work, is anticipated.

Materials and methods

Sequence mapping and retrieval and feature calculation were performed in Python version 2.7 (63) using BioPython (64) and NumPy (65); executed and consolidated using Bash (shell) scripts; and parallelized wherever possible using GNU Parallel (66). Data analysis and presentation was done in R (67) within RStudio (68) using magrittr (69), plyr (70), dplyr (71), asbio (72), and datamart (73) for data handling; ggplot2 (74), ggbeeswarm (75), GGally (76), gridExtra (77), cowplot (78), scales (79), viridis (80), and RColorBrewer (81, 82) for plotting; multidplyr (83) with parallel (67) and foreach (84) with iterators (85) and doMC (86)/doParallel (87) for parallel processing; and roxygen2 (88) for code organization and documentation as well as other packages as referenced.

1) Collection of data necessary for learning and evaluation

E. coli sequence data—The nucleotide sequences from Ref. 20 were deduced by reconstructing forward and reverse primers (i.e. ~20-nucleotide stretches) from each gene in Colibri (based on EcoGene 11); the original source cited and later verified these primers against an archival spreadsheet provided directly by Daniel Daley.4 To account for sequence and annotation corrections made to the genome after the work of Daley and Rapp et al. (20), these primers were directly used to reconstruct the amplified product from the most recent release of the E. coli K-12 substrate MG1655 genome (38) (EcoGene 3.0; U00096.3). Although Daniel Daley mentioned that raw reads from the Sanger sequencing runs may be available within his own archives, it was decided that the additional labor to retrieve these data and parse these reads would not significantly impact the model. The deduced nucleotide sequences were verified against the protein lengths given in Table 1 of Ref. 20. The plasmid library tested in Ref. 29 was provided by Daniel Daley, and those sequences are taken to be the same.

E. coli training data—The preliminary results using the mean-normalized activities echoed the findings of Ref. 20 that these do not correlate with sequence features either in the univariate sense (many simple linear regressions; Table S1 (20)) or a multivariate sense (multiple linear regression; data not shown). This is presumably due to the loss of information regarding variability in expression level for given genes or due to the increase in variance of the normalized quantity (See “Materials and methods” 4a) due to the normalization and averaging procedure. Daniel Daley and Mikaela Rapp provided spreadsheets of the outcomes from the 96-well plates used for their expression trials and sent scanned copies of the readouts from archival laboratory notebooks where the digital data were no longer accessible.4 Those proteins without a reliable C-terminal localization (as given in the original work) or without raw expression outcomes were not included in further analyses. Similarly, Nir Fluman also provided spreadsheets of the raw data from the set of three expression trials performed in Ref. 29.

NYCOMPS data—Brian Kloss, Marco Punta, and Edda Kloppman provided a data set of actions performed by the NYCOMPS center, including expression outcomes in various conditions (2, 3). The protein sequences were mapped to NCBI GenInfo Identifier (GI) numbers via either the Entrez system (89) or the Uniprot mapping service (90). Each GI number was mapped to its nucleotide sequence via a combination of the NCBI Elink mapping service and the “coded_by” or “locus” tags of coding sequence (CDS) features within GenBank™ entries. Although custom-purpose code was written, a script from Peter Cock is available to do the same task via a similar mapping mechanism (91). To confirm all sequences, the TargetTrack (26) XML file was parsed for the internal NYCOMPS identifiers and compared for sequence identity with those that had been mapped using the custom script; 20 (less than 1%) of the sequences had minor inconsistencies and were manually replaced.

Archaeal transporter data—The locus tags (“Gene Name” in Table 1 of Ref. 46) were mapped directly to the sequences and retrieved from NCBI. Pikyee Ma and Margarida Archer clarified questions regarding their work to inform the analysis.

GPCR expression data—Nucleotide sequences were collected by mapping the protein identifiers given in Table 1 of Ref. 49 to protein GIs via the Uniprot mapping service (90) and subsequently to their nucleotide sequences via the custom mapping script described above (see “NYCOMPS data”). The sequence length and pl were validated against those provided. Renaud Wagner assisted in providing the nucleotide sequences for genes whose listed identifiers were unable to be mapped and/or did not pass the validation criteria, as the MeProtDB (the sponsor of the GPCR project) does not provide a public archive.
Helicobacter pylori data—Nucleotide sequences were retrieved by mapping the locus tags given in Table S1 from Ref. 50 to locus tags in the January 31, 2014 release of the H. pylori 26695 genome (AE000511.1). To verify sequence accuracy, sequences whose molecular weight matched that given by the authors were accepted. Those that did not match, in addition to the one locus tag that could not be mapped to the January 31, 2014 genome version, were retrieved from the April 9, 2015 release of the genome (NC_000915.1). Both releases are derived from the original sequencing project (92). After this curation, all mapped sequences matched the reported molecular weight.

In this data set, expression tests were performed in three expression vectors and scored as 1, 2, or 3. Two vectors were scored via two methods. For these two vectors, the two scores were averaged to give a single number for the condition, making them comparable with the third vector while yielding two additional thresholds (1.5 and 2.5) resulting in the five total curves shown (Fig. S2B).

Mycobacterium tuberculosis data—The authors note using TubercuList through GenoList (93); therefore, nucleotide sequences were retrieved from the archival website based on the original sequencing project (94). The sequences corresponding to the identifiers and outcomes in Table 1 from Ref. 48 were validated against the provided molecular weight.

Secondary transporter data—GI numbers given in Table 1 from Ref. 52 were matched to their CDS entries using the custom mapping script described above (see NYCOMPS data). Only expression in E. coli with isopropyl 1-thio-β-D-galactopyranoside–inducible vectors was considered.

T. maritima data—Gene names given in Table 1 of Ref. 51 were matched to CDS entries in the January 31, 2014 release of the T. maritima MSB8 genome (AE000512.1), a revised annotation of the original release (96). The sequence length and molecular weight were validated against those provided.

Pseudomonas aeruginosa data—Outcomes in Additional File 1 of Ref. 47 were matched to coding sequences provided by Constance Jeffrey.

Shine–Dalgarno–like mutagenesis data—Folded protein is quantified by densitometry measurement (97, 98) of the relevant band in Fig. 6 of Ref. 29. Relative difference is calculated as is standard.

\[
\text{Relative difference} = \frac{\text{metric}_{\text{mutant}} - \text{metric}_{\text{wildtype}}}{\frac{1}{2}[\text{metric}_{\text{mutant}} - \text{metric}_{\text{wildtype}}]}
\]

(Eq. 2)

2) Details related to the calculation of sequence features

Transmembrane segment topology was predicted using Phobius Constrained for the training data and Phobius for all other data sets (42). We were able to obtain the Phobius code and integrate it directly into our feature calculation pipeline, resulting in significantly faster speeds than any other option. Several features were obtained by averaging per-site metrics (e.g. per-residue RONN3.2 disorder predictions) in windows of a specified length. Windowed tAI metrics are calculated over all 30-base windows (not solely over 10-codon windows). Table S1 includes an in-depth description of each feature. Future work will explore contributions of elements outside the gene of interest (e.g. ribosomal binding site, linkers, tags).

3) Preparation for model learning

Calculated sequence features for the IMPs in the E. coli data set as well as raw activity measurements (i.e. each 96-well plate) were loaded into R. As is best practice in using support vector machines, each feature was “centered” and “scaled” where the mean value of a given feature was subtracted from each data point and then divided by the S.D. value of that feature using preprocess (99). As is standard practice, the resulting set was then culled for those features of near-zero variance, over 95% correlation (Pearson’s r), and linear dependence (nearZeroVar, findCorrelation, findLinearCombos) (99). In particular, this procedure removed extraneous degrees of freedom during the training process, which carry little to no additional information with respect to the feature space and which may overrepresent certain redundant features. Features and outcomes for each list (“query”) were written into the SVMlight format using a modified svmlight.write (100).

The final features were calculated for each sequence in the test data sets, prepared for scoring by “centering” and “scaling” by the training set parameters via preprocess (99), and then written into SVMlight format again using a modified svmlight.write.

4) Model selection, training, and evaluation using SVMrank

(a) At the most basic level, our predictive model is a learned function that maps the parameter space (consisting of nucleotide and protein sequence features) to a response variable (expression level) through a set of governing weights (\(w_1, w_2, \ldots, w_n\)). Depending on how the response variable is defined, these weights can be approximated using several different methods. As such, defining a response variable that is reflective of the available training data is key to selecting an appropriate learning algorithm.

The quantitative 96-well plate results (20) that comprise our training data do not offer an absolute expression metric valid over all plates; the top-expressing proteins in one plate would not necessarily be the best-expressing within another. As such, this problem is suited for preference-ranking methods. As a ranking problem, the response variable is the ordinal rank for each protein derived from its overexpression relative to the other members of the same plate of expression trials. In other words, the aim is to rank highly expressed proteins (based on numerous trials) at higher scores than lower expressed proteins by fitting against the order of expression outcomes from each constituent 96-well plate.

(b) As the first work of this kind, the aim was to employ the simplest framework necessary, taking in account the considerations above. The method chosen computes all valid pairwise classifications (i.e. within a single plate), transforming the original ranking problem into a binary classification problem. The algorithm outputs a score for each input by minimizing the number of swapped pairs, thereby maximizing Kendall’s \(\tau\) (101). For example, consider the following data generated via
context A \((X_{A,1}, Y_{A,1}), (X_{A,2}, Y_{A,2})\) and B \((X_{B,1}, Y_{B,1}), (X_{B,2}, Y_{B,2})\), where observed response follows as index \(i\) \(i.e. Y_i < Y_{i+1}\). Binary classifier \(f(X, Y)\) gives a score of 1 if an input pair matches its ordering criteria and -1 if not; \(i.e. Y_i < Y_j\):

\[
\begin{align*}
  f(X_{A,1}, Y_{A,2}) &= 1; f(X_{A,2}, Y_{A,1}) = -1 \\
  f(X_{B,1}, Y_{B,2}) &= 1; f(X_{B,2}, Y_{B,1}) = -1 \\
  f(X_{A,1}, Y_{B,2}), f(X_{B,2}, Y_{A,1}) &\text{ are invalid}
\end{align*}
\]

Free parameters describing \(f\) are calculated such that those calculated orderings \(f(X_{A,1})f(X_{A,2}) \ldots ; f(X_{B,1})f(X_{B,2}) \ldots \) most closely agree (overall Kendall’s \(\tau\) with the observed ordering \(Y_nY_{n+1} \ldots \)). In this sense, \(f\) is a pairwise learning to rank method.

Within this class of models, a linear preference-ranking support vector machine was employed (102). To be clear, as an algorithm, a preference-ranking SVM operates similarly to the canonical SVM binary classifier. In the traditional binary classification problem, a linear SVM seeks the maximally separating hyperplane in the feature space between two classes, where class membership is determined by which side of the hyperplane points reside. For some \(n\) linear separable training examples, \(D = \{(x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}\}\) and two classes \(y_i \in \{-1, 1\}\), a linear SVM seeks a mapping from the \(d\)-dimensional feature space \(\mathbb{R}^d \rightarrow \{-1, 1\}\) by finding two maximally separated hyperplanes \(wx - b = 1\) and \(wx - b = -1\) with constraints that \(wx_i - b \geq 1\) for all \(x_i\), \(y_i = 1\), and \(wx_i - b \leq -1\) for all \(x_i\), \(y_i = -1\). The feature weights correspond to the vector \(w\), which is the vector perpendicular to the separating hyperplanes and are computable in \(O(n \log n)\) implemented as part of the SVM^*^ package (43). See Ref. 102 for an in-depth, technical discussion.

(c) In a soft-margin SVM where training data is not linearly separable, a tradeoff between misclassified inputs and separation from the hyperplane must be specified. This parameter \(C\) was found by training models against raw data from Daley and Rapp et al. (20) with a grid of candidate \(C\) values (\(2^n \forall n \in [-5, 5]\)) and then evaluated against the raw “folded protein” measurements from Fluman et al. (29). The final model was chosen by selecting that with the lowest error from the process above (\(C = 2^5\)). To be clear, the final model is composed solely of a single weight for each feature; the tradeoff parameter \(C\) is only part of the training process.

Qualitatively, such a preference-ranking method constructs a model that ranks groups of proteins with higher expression level higher than other groups with lower expression value. In comparison with methods such as linear regression and binary classification, this approach is more robust and less affected by the inherent stochasticity of the training data.

5) Quantitative assessment of predictive performance

In generating a predictive model, one aims to enrich for positive outcomes while ensuring that they do not come at the cost of increased false positive diagnoses. This is formalized in ROC theory (for a primer, see Ref. 44), where the true positive rate is plotted against the false positive rate for all classification thresholds (score cutoffs in the ranked list). In this framework, the overall ability of the model to resolve positive from negative outcomes is evaluated by analyzing the AUC, where \(AUC_{\text{perfect}} = 100\%\) and \(AUC_{\text{random}} = 50\%\) (percentage signs are omitted throughout). All ROCs are calculated through pROC (103) using the analytic Delong method for AUC confidence intervals (104). Bootstrapped AUC confidence intervals (CIs) \((n = 10^n)\) were precise to 4 decimal places, suggesting that analytic CIs are valid for the NYCOMPS data set.

With several of our data sets, no definitive standard or clear-cut classification for positive expression exists. However, the aim is to show and test all reasonable classification thresholds of positive expression for each data set to evaluate predictive performance as follows.

Training data—The outcomes are quantitative (activity level), so each ROC is calculated by normalizing within each data set to the standard well subject to the discussion in 4a above (LepB for PhoA and InvLepB for GFP) (examples in Fig. 1D) for each possible threshold (\(i.e.\) each normalized expression value with each AUC plotted in Fig. 1E). 95% confidence intervals of Spearman’s \(p\) are given by \(10^6\) iterations of a bias-corrected and accelerated bootstrap of the data (Fig. 1, A and C) (105).

Large scale—ROCs were calculated for each of the expression classes (Fig. 2E). Regardless of the split, predictive performance is noted. The binwidth for the histogram was determined using the Freedman–Diaconis rule (106), and scores outside the plotted range comprising <0.6% of the density were implicitly hidden.

Small scale—Classes can be defined in many different ways. To be principled about the matter, ROCs for each possible cutoff are presented based on definitions from each publication (Fig. 3 (C, E, and G) and Fig. S2 (B, D, and F)). See “Materials and methods” 1 for any necessary details about outcome classifications for each data set.

6) Feature weights

Weights for the learned SVM are pulled directly from the model file produced by SVM^*^ and are given in Table S1.

7) Availability

All analysis is documented in a series of R notebooks (107) available openly at github.com/clemlab/IMProve. These notebooks provide fully executable instructions for the reproduction of the analyses and the generation of figures and statistics in this study. The IMProve model is available as a web service. Additional code is available upon request.

Author contributions—S.M.S., A.M., and W.M.C. conceptualization; S.M.S. resources; S.M.S., N.J., and A.M. data curation; S.M.S., N.J., and A.M. software; S.M.S. formal analysis; S.M.S. validation; S.M.S. and W.M.C. investigation; S.M.S. and W.M.C. visualization; S.M.S. methodology; S.M.S. and W.M.C. writing-original draft; S.M.S. and W.M.C. project administration; S.M.S. and W.M.C. writing-review and editing; W.M.C. supervision; W.M.C. funding acquisition.

---

5 Please note that the JBC is not responsible for the long-term archiving and maintenance of this site or any other third party hosted site.
Acknowledgments—We thank Daniel Daley and Thomas Miller’s group for discussion, Yaser Abu-Mostafa and Yisong Yue for guidance regarding machine learning, Niles Pierce for providing NUPACK source code (33), Welison Floriano and Naved Nair-Ansari for maintaining local computing resources, and Samuel Schultz for suggesting the model’s name. We thank Michiel Niesen, Stephen Marshall, Thomas Miller, Reid van Lehn, James Bowie, and Tom Rapoport for comments on the manuscript. Models and analyses were possible thanks to raw experimental data provided by Daniel Daley and Mikaela Rapp (20); Nir Fluman and Douglass Rees. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation Grant ACI-1053575 (108).

References


A membrane protein expression predictor

Sci. U.S.A. 105, 5722–5727 CrossRef Medline
