# Decoding sequence-level information to predict membrane protein expression

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

**Affiliation**

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

*Correspondence to: clemons@caltech.edu

**Abstract**

The expression and purification of integral membrane proteins remains a major bottleneck in the characterization of these important proteins. Expression levels are currently unpredictable, which renders the pursuit of these targets challenging and highly inefficient. Evidence demonstrates that small changes in the nucleotide or amino-acid sequence can dramatically affect membrane protein biogenesis; yet these observations have not resulted in generalizable approaches to improve expression. In this study, we develop a data-driven statistical model that predicts membrane protein expression in *E. coli* directly from sequence. The model, trained on experimental data, combines a set of sequence-derived variables resulting in a score that predicts the likelihood of expression. We test the model against various independent datasets from the literature that contain a variety of scales and experimental outcomes demonstrating that the model significantly enriches expressed proteins. The model is then used to score expression for membrane proteomes and protein families highlighting areas where the model excels. Surprisingly, analysis of the underlying features reveals an importance in nucleotide sequence-derived parameters for expression. This computational model, as illustrated here, can immediately be used to identify favorable targets for characterization.

## Introduction

The central role of integral membrane proteins motivates structural and biophysical studies that require large amounts of purified protein, often at considerable cost of both material and labor. Only a small percentage can be produced at high-levels resulting in membrane protein structural characterization lagging roughly 20 years behind that of soluble proteins[1]. To increase the pace of structure determination, the scientific community created large government-funded structural genomics consortia facilities, like the NIH-funded New York Consortium on Membrane Protein Structure (NYCOMPS)[2]. For this representative example, more than 8000 genes, chosen based on characteristics hypothetically related to success, yielded only 600 (7.1%) highly expressing proteins[3] resulting to date in 34 (0.4%) unique structures (based on annotation in the RCSB PDB[4]). Despite considerable investment on many scales, the lack of expressed targets has hampered membrane protein structural studies[5].

Tools for improving the number of expressed membrane proteins are needed. While significant work has shown promise on a case-by-case basis, *e.g.* growth at lower temperatures, codon optimization[6], and regulating transcription[7], a generalizable solution remains elusive. Currently, each target must be addressed individually as the conditions that were successful for a previous target seldom carry over to other proteins, even amongst closely related homologs[8,9]. For individual cases, simple changes can have dramatic effects on the amount of expressed proteins[10,11]. Considering the scientific value of membrane protein studies and absence of systematic methods to improve expression, it is surprising that no method can provide solutions with broad applicability across protein families and genomes nor describe the variation in expression levels seen between closely related sequences.

While there are no approaches to broadly decode sequence-level information precluding its use for predicting membrane protein expression, the concept that sequence variation can measurably influence

membrane protein biogenesis is commonplace. For example, positive-charges on cytoplasmic loops are important determinants of membrane protein topology[12,13]; yet introduction of mutations presumed to enhance certain properties, such as the positive inside rule, has not proven generalizable for improving expression[9]. This reasons for this likely lie in the complex underpinnings of membrane protein biogenesis, where the interplay between sequence parameters at the protein and nucleotide levels must be considered. Optimizing for a single sequence-level feature likely diminishes the beneficial effect of other features (*e.g.* increasing positive residues on internal loops might diminish favorable mRNA properties). Without accounting for the broad set of features related to membrane protein expression, it is impossible to predict differences in expression.

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 determinants of expression. The resulting model allows ranking of any arbitrary set of membrane protein sequences in order of their relative likelihood of successful expression. In doing so, we leverage the corpus of work that shows that sequence-level characteristics are important determinants of protein biogenesis, *e.g.* RNA secondary structure[14,15], transmembrane segment hydrophobicity[16], the positive inside rule[17], and loop disorder[18]. 105 of these sequence-derived parameters were calculated for individual proteins within datasets of interest (Table S1). As the first report to predict expression, we train a linear equation that provides a score based on calculating the sum of weighted features where the weights are derived from fitting to experimental expression data, a "training set." This model can be used broadly to score any membrane protein based on its calculated features. We validate the model against a variety of independent datasets demonstrating its generalizability. To support further experimental efforts, we broadly score the membrane proteome from a variety of important genomes

and showcase the performance of the model across protein families. This approach and resulting model provides an exciting example 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 tool for membrane protein expression. While the benefits derived from low cost and low barriers for adoption are obvious, the applicability to the spectrum of the membrane proteome are becoming clearer. Of note, 43 of the 216 unique eukaryotic membrane protein structures were solved using protein expressed in *E. coli* (based on annotation in the RCSB PDB[4]). This demonstrates the utility of *E. coli* as a broad tool and its potential if the expression problem can be overcome.

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

A key component of any machine learning model is the choice of dataset used for training. Having searched the literature, we identified two publications that contained quantitative datasets on the heterologous expression of *E. coli* membrane proteins in *E. coli*. The first set, Daley, Rapp *et al.*, contained activity measures, proxies for expression level, from C-terminal tags of either GFP or PhoA (alkaline phosphatase)[19]. The second set, Fluman *et al.*, contained a more detailed analysis of a subset from the first utilizing in-gel fluorescence to measure folded protein[20] (see Methods 4c). The expression results strongly correlated between the two datasets notably in that normalized GFP activity was a good measure of the amount of folded membrane protein (Figure 1a, also [21]). The experimental set-up employed multiple 96-well plates over multiple days resulting in pronounced variability in the absolute expression level of a given protein between trials. Daley, Rapp, *et al.* calculated average expression levels by dividing the raw expression level of each protein by that of a control construct (Inverse LepB-GFP or LepB-PhoA) on the corresponding plate. While the resulting values were useful for the relevant

question of identifying topology, we were unable to successfully fit a linear regression or a standard linear-SVM on either the raw data compiled from all plates or averaged outcomes of each gene. This unexpected outcome suggested that the measurements required a more complex analysis.

We hypothesized that measurements could be more accurately compared within an individual plate then across the entire dataset. To account for this, a preference-ranking linear SVM algorithm ($SVM^{rank}$ [22]) was chosen (see Methods 4b). Simply put, the $SVM^{rank}$ algorithm determines the optimal weight for each 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 SVM cost function is used accounting for the plate-wise constraint specified above. In practice, the process optimizes as a training metric the correlation coefficient Kendall's $\tau$ to converge upon a set of weights. Kendall's $\tau$ measures the agreement between ordinal quantities by calculating correctly-ordered and swapped pairs.

Various metrics related to the training data can be derived to assess the accuracy with which the model fits the input data (see Methods 4C). The $SVM^{rank}$ training metric shows varying agreement for all groups (*i.e.*, $\tau_{kendall} > 0$) (Figure 1b). For individual genes, activity values normalized and averaged across trials were not directly used for the training procedure (see Methods 4a); yet one would anticipate that scores for each gene should broadly correlate with expression. Indeed, the observed normalized activities positively correlate with the $SVM^{rank}$ score output by the model (Figure 1c).

For a more quantitative approach to assessing the models success within the training data, we turn to the Receiver Operating Characteristic (ROC). ROC curves quantify the tradeoff between true positive and false positive predictions across the numerical scores output from a predictor. This is a more reliable assessment of prediction than simply calculating accuracy and precision from a single, arbitrary score

threshold [23]. The figure of merit that quantifies an ROC curve is the Area Under the Curve (AUC). Given that the AUC for a perfect predictor corresponds to 100% and that of a random predictor is 50% (Figure 1d, grey dashed line), an AUC greater than 50% indicates predictive performance of the model (percentage signs hereafter omitted) (see Methods 5 and [23]). Here, the ROC framework will be used to quantitatively assess the ability of our model to predict the outcomes within the various datasets.

The training datasets are quantitative measures of activity requiring that an activity threshold be chosen that defines positive or negative outcomes. For example, ROC curves using two distinct activity thresholds, at the 25$^{th}$ or 75$^{th}$ percentile of highest expression, are plotted with their calculated AUC values (Figure 1d). While both show that the model has predictive capacity, a more useful visualization would consider all possible activity thresholds. For this, the AUC value for every activity threshold is plotted showing that the model has predictive power regardless of an arbitrarily chosen expression threshold (Figure 1e). In total, the analysis demonstrates that the model can rank expression outcomes for all proteins. Interestingly, for PhoA-tagged proteins the model is progressively less successful with increasing activity. PhoA activity is an indirect measure of expression of proteins with their C-termini in the periplasm bringing into question either the utility of this quantification method relative to GFP activity or perhaps that this class of proteins are special in the model. An argument for the former is presented later (Figure 2e).

*Demonstration of prediction against an independent large expression dataset*

While the above analyses show that the model successfully fits the training data, we assess the broader applicability of the model based on its success at predicting the outcomes of independent large- and small-scale expression trials. The first test considers results from NYCOMPS, where 8444 membrane protein genes entered expression trials, in up to eight conditions, resulting in 17114 expression outcomes[2]. The majority of genes were attempted in only one condition (Figure 2a), and

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 purification[3]. Therefore, for this analysis, we consider the experimental results in various ways: either outcomes per gene (if at least one trial is positive, the gene is considered positive for expression), all conditions (each expression trial considered independently), or based on defined expression conditions. For the first, several metrics demonstrate prediction (Figure 2b-d).

A major aim of this work is to enrich the likelihood of choosing positively expressing proteins. The positive predictive value (PPV, true positives ÷ predicted positives) becomes a useful metric for positive enrichment as it conveys the degree of improvement over the experimental baseline of the dataset. The PPV of the model is plotted as a function of the percentile of the $SVM^{rank}$ score threshold for the definition of predicted positives (Figure 2b). In the figure, the overall positive percentage (~24%), an experimental baseline, is represented by a grey dashed line; therefore, a relative increase reflects the increased predictive power of the algorithm. For example, considering the top fourth of genes by $SVM^{rank}$ score ($75^{th}$ percentile) shows that the algorithm enriches for positive outcomes by 8.4% over baseline. Seen another way, a histogram of the $SVM^{rank}$ score for each protein is plotted separated by positive versus negative outcomes (Figure 2c). Visually, the distribution of the scores for the positive group is shifted to a higher score relative to the negative group, which is substantiated quantitatively by the ROC and its corresponding AUC (Figure 2d). Interestingly, considering the predictive power against all conditions as opposed to by gene shows slightly better statistics (AUC=62.6) reflective of the fact that many genes have mixed outcomes (Figure 2e). Importantly, the model shows consistent performance throughout each of the eight possible conditions tested (Figure 2e, black, numerically in Table S2).

The ability to predict the experimental data from NYCOMPS allows a return to the question of alkaline phosphatase as a metric for expression. To investigate the trend that the expression of proteins with periplasmic C-termini measured by alkaline phosphatase (Figure 1, orange) show less consistent fitting by the model, the NYCOMPS outcomes are split by putative C-terminal localization as predicted by Phobius[24]. No significant difference in AUC between C-terminal localizations across all conditions (Figure 2e, green vs. orange) indicate that the model is applicable for all topologies.

*Further demonstration of prediction against small-scale independent datasets*

The NYCOMPS example demonstrates the predictive power of the model across the broad range of sequence space encompassed by that dataset. 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. While a number of datasets exist[8,25–35], we could only identify six for which complete sequence information could be obtained to calculate all the necessary sequence parameters[25–30].

The first dataset is derived from the expression of 14 archaeal transporters in *E. coli* chosen based on their homology to human proteins[25]. For each putative transporter, expression was performed in three plasmids and two strains (six total conditions) with the membrane fraction quantified by both a Western blot against a histidine-affinity tag and Coomassie Blue staining of an SDS-PAGE gel. Here, the majority of the expressing proteins fall into the top half of the $SVM^{rank}$ scores, 7 out of 9 of those with multiple positive outcomes (Figure 3a, top). Strikingly, quantification of the Coomassie Blue staining highlights a clear correlation with the $SVM^{rank}$ score where the higher expressing proteins have the highest score (Figure 3a, bottom). ROC curves are plotted for the two thresholds: expression detected at least by Western blot or, for the smaller subset, by Coomassie Blue (Figure 3b). In both cases, the model shows predictive power.

The next test considers the expression of 105 *Mycobacterium tuberculosis* proteins in *E. coli*[26]. Protein expression was measured both by Coomassie Blue staining of an SDS-PAGE gel and Western blot with only outcomes from the membrane fraction considered for this analysis. The highest expressing proteins (detected via Coomassie Blue) follow the trend given by the $SVM^{rank}$ score with 7 of the 9 falling within the top half of scoring proteins (Figure 3c) and is reflected in the ROC (Figure 3d). In contrast, using the positive Western blot outcomes as the minimum threshold (Figure 3c) shows an AUC no better than random (Figure 3d). Given that no internal standard was used and that each expression trial was performed only once, proteins that were positive by Western blot may represent a pool indistinguishable in expression from those not detected; alternatively, these results support that our statistical model accurately captures the most highly expressing proteins.

A broader test considers expression trials of 101 mammalian GPCRs in bacterial and eukaryotic systems[27]. Trials in *E. coli*, measured via Western blot of an insoluble fraction, again show highly expressing proteins at higher $SVM^{rank}$ scores while the expression of the same proteins in *P. pastoris*, measured via dot blot, fail to show broad agreement (Figure 3e,f). The lack of predictive performance in *P. pastoris* suggests that the parameterization of the model, calibrated for broadly characterizing *E. coli* expression, requires retraining to generate a different model that captures the distinct interplay of sequence parameters in yeast.

Further expression trials of membrane proteins from *H. pylori*, *T. maritima* as well as microbial secondary transporters continues to show the same broad agreement[28–30] (Figure S1). *H. pylori* membrane proteins showed that as the threshold for positive expressing proteins increases, the performance of the model improves (using the highest threshold n=46 and AUC=67.7) (Figure S1a,b). For *T. maritima* expression, the model weakly captures outcomes for two defined thresholds (n=5 and 19, AUC=61.7 and 58.7), but due to the small number of successful outcomes, the confidence intervals

are broad (Figure S1c,d). The expression of microbial secondary transporters shows varied agreement with the model. Taking proteins at the lower defined expression threshold shows predictive performance (n=59, AUC=60.5); however, considering the defined high-expressing proteins is less conclusive (n=26, AUC=52.0) (Figure S1e,f).

*Forward predictions on genomes of interest*

The model successfully enriches for heterologous expression of membrane proteins in *E. coli* strikingly across scales, laboratories, quantification methods, and protein families supporting its broad generalizability. While few genes express in every condition tested (Figure 2a and 3a), the model predicts the likelihood that a gene will express within a set of conditions and enriches for those that will work in any condition (Figure 2e, numerically in Table S2). Notably, had this model been implemented during the NYCOMPS target selection process, only testing targets with an $SVM^{rank}$ score greater than 0.5 ($90^{th}$ percentile or above), based on known outcomes, the percentage of successful genes would have increased from 25% to 37% (Figure 2b). For perspective, testing the same number of genes would have resulted in an additional 912 expressed proteins, representing a significant improvement in the return on investment.

To expand on the utility of this model, $SVM^{rank}$ scores were calculated for membrane proteins from a variety of metazoan and microbial genomes (Figure 4a and Figure 2a). Many genomes have a significant proportion of proteins with high scores particularly evidenced by portions of the distributions ahead of the median in *E. coli* given by the vertical dashed line (Figure 4a). The likelihood for successful expression may be inferred by equating $SVM^{rank}$ score with the PPV from the most prevalent NYCOMPS expression condition which rises dramatically at scores above zero (Figure 4b). The range of scores spans those representative of high-expressing membrane proteins in both *E. coli* (Figure 1c) as

well as in the NYCOMPS dataset (Figure 2c) and provides suggested targets for future biophysical studies (Table S4).

The predictions present several surprises at the biological level. One such is that the distribution of membrane proteins from representative thermophilic bacterial genomes have generally lower relative $SVM^{rank}$ scores than other genomes, which implies that these proteins, on average, are harder to express in *E. coli*. This contrasts the many empirical examples of proteins from thermophiles being used for biophysical characterization. In the case of the malarial parasite *P. falciparum*, the inverse trend is true with higher than expected relative $SVM^{rank}$ scores despite the expectation that these proteins would be hard to express in *E. coli*. A possible cause for the unexpected distribution of scores may lie in the differences in the parameters that define the proteins in these particular groups. As the training set consists only of native *E. coli* sequences, the range of values for each parameter in the training set may not represent the full range of possible values for the parameter. For the special cases highlighted, perhaps the underlying sequence parameters fall into a poorly characterized subset of sequence space bringing into question the applicability of the model for these cases.

To address the utility of the model relative to differences in the sampling of sequence parameters, we measure the overlap of the distributions of sequence parameters for a given subset (see Methods 7) (Figure S3b). Simply put, if two subsets contain the same distribution of sequence parameters the expectation is that a given parameter should approach 100%. In the simplest case, comparing the distribution of sequences parameters in all *E. coli* membrane proteins against the subset used in the training set shows that the majority of parameters have overlap values over 75% (Figure 4c), which provides a lower threshold for similarity of sequence parameter range. For NYCOMPS sequences, most of the overlap values relative to the training set are above the threshold. As this set shows predictive performance, comparison to the training set provides a baseline to assess the reliability of predictions

within other subsets (Figure 4d-f, x-axis). In the first case (Figure 4d), there is a strong correlation between all the forward predictions and NYCOMPS, *i.e.* values are near the diagonal (quantified by a Mean Absolute Deviation (MAD) = 11.6), suggesting that differences in parameter space do not significantly affect the predictive power of the model. For the thermophiles subset (Figure 4e), the values again are close to the diagonal (*i.e.* low MAD = 10.6) implying that the predictions are credible. *P. falciparum* (Figure 4f), on the other hand, clearly shows stark differences as most parameters fall below the 75% cut-off (MAD = 29.0) bringing into question the reliability of these predictions. A training set with broader coverage of the parameter space may generate a better predictor for all genomes.

*Performance of the model across protein families*

To provide a clear path forward for experiment, we consider the performance of the model with regards to protein homology families where protein family definitions are based on Pfam classifications[36]. For outcomes from NYCOMPS (Figure 5a), there are no significant difference in the predictive performance of the model between groups of genes whether or not they are part of a protein family found in the training set.

The scale of NYCOMPS allows us to investigate whether there are protein families for which the model does better or worse than the aggregate. For this, an AUC is calculated for each protein family that has minimally five total outcomes (including at least one positive and one negative). Figure 5B plots the AUC for each protein family in increasing order as a cumulative distribution function. The breadth of the AUC values highlights the variability in predictive power across families. Most families can be predicted by the model (115 of 159 have an AUC > 0.5, visually blue and purple) though some not at 95% confidence (57 of 115, blue), likely due to an insufficient number tested.

For the protein families that are well-predicted within the NYCOMPS set, the model gives accurate insight into the likelihood of expression of a given protein. We demonstrate the utility of this prediction by looking at protein families that have yet to be characterized structurally. While there are a number of choices, a first example is the protein family annotated as short-chain fatty-acid transporters (PF02667), characterized by AtoE in *E. coli*, that typically contains 10 transmembrane domains with an overall length of ~450 amino acids. A second example is the protein family annotated as copper resistance proteins (PF05425), characterized by CopD in *E. coli*, that typically contains eight transmembrane domains with an overall length of ~315 amino acids. In both cases, as indicated by the AUC values, the model provides a clear score cut-off for consideration for expression. For example, considering CopD homologs, one would expect that those with $SVM^{rank}$ scores above -1 will express.

*Biological importance of various sequence parameters*

Using a simple proof-of-concept linear model allowed for a straightforward and robust predictor; however, intrinsically this complicates determination of biological underpinnings due to the unequal distribution of weight across correlating features. For example, the average $\Delta G_{insertion}$ of transmembrane segments has a positive weight whereas average hydrophobicity, a correlating parameter, has a negative weight (Table S1, Figure S3). As many parameters, such as those related to hydrophobicity, are highly correlated; conclusive information cannot be obtained simply using weights of individual features to interpret the relative importance of their underlying biological phenomena. An alternative is to collapse related features into biologically meaningful categories reducing correlation (Figure 6a), thereby providing a mechanism to interpret information from the model. For example, the hydrophobicity group incorporates sequence features such as average hydrophobicity, maximum hydrophobicity, $\Delta G_{insertion}$, etc. The full list of groupings are provided in Table S1 and Figure S3.

Analysis of categories suggests the phenomena that drive prediction. To visualize this, the collapsed weights are summarized in Figure 6b 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 of 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.

To explore the role of each category in prediction, we calculate the performance of the model by either only using weights from features within a single category or excluding weights from within a single category. We assess predictive performances by calculating ROC curves across genes and expression trials from NYCOMPS dataset for each case (Figure 6c,d). Feature categories that are sufficient for prediction will have an AUC > 0.5 when used alone (Figure 6c) and those necessary for the model will show an AUC < 0.5 when excluded from prediction (Figure 6d). Notably, when only considering all genes independent of condition, most individual categories cannot predict expression (*i.e.* AUC with 95% CI straddling 0.5) (Figure 6c, red). A notable exception is tRNA Adaptation Index, where the per gene AUC is slightly higher than the performance of the model. However, since the model demonstrates predictive performance at 95% confidence across all experimental conditions (Figure 2e), feature categories that are sufficient for prediction must also perform across these conditions. In this case, tRNA Adaptation Index performs poorly against a number of the experimental subsets, so it is not sufficient for prediction. On the other hand, while Codon Pair Score alone shows predictive power across experimental conditions, when excluded from the model, the category alone cannot explain the model (only a single 95% confidence interval crossing AUC = 50, Figure 6d).

Importantly, no feature category independently drives the predictor as excluding each individually does not significantly affect the overall predictive performance, except for Length/pI (isoelectric point) (Figure 6d). Sequence length composes the majority of the weight within this category and is one of the highest weighted features in the model. This is consistent with the anecdotal observation that larger membrane proteins 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.* Figure 3,4). 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 important, and would preclude broad generalizability as shown for the predictor presented here.

A coarser view of the weights is a comparison of the features derived from either protein or 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 (Figure 6e). Comparison of the predictive performance of the two subsets of weights shows that the nucleotide features alone can give similar performance to the full model (Figure 6f). It is important to note that this does not suggest that protein features are not important for membrane protein expression. Instead, within the context of the trained model, nucleotide features are critical for predictive performance for a large dataset such as NYCOMPS. This finding corroborates growing literature that the nucleotide sequence holds significant determinants of biological processes[14,20,37–39].

*Sequence optimization for expression*

The predictive performance of the model implies that the parameters defined here provide a coarse approximation of the fitness landscape for membrane protein expression. Attempting to optimize a

single feature by modifying the sequence will likely affect the resulting score and expression due to changes in other features. Fluman, *et al.*[20] provides an illustrative experiment. They hypothesized that altering the number of Shine-Dalgarno (SD)-like sites in the coding sequence of a membrane protein would affect expression. 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 (Figure 6g). However, the resulting changes in the $SVM^{rank}$ score correspond with the changes in measured expression as the model considers changes to other nucleotide features. Capture of the outcomes in this small test case by the model illustrates the utility of integrating the contribution of the numerous parameters involved in membrane protein biogenesis.

## Discussion

The model developed here provides a robust predictor for membrane protein expression. The current best practice for characterization of a membrane protein target begins with the identification and testing of many homologs or variants for expression. The model presented here will allow for prioritization of targets to test for expression thereby making more optimal use of limited human and material resources. In addition, due to the scale of NYCOMPS, protein families that were extensively tested provide ranges of scores (*e.g.* Figure 5c) where the score of an individual target directly indicates its likelihood of expression relative to known experimental results. We provide the current predictor as web service where scores can be calculated, and the method, associated data, and suggested analyses are publically available to catalyze progress across the community (http://clemonslab.caltech.edu).

The generalizability of the model is remarkable despite several known limitations. Using data from a single study for training precludes including certain variables that empirically influence expression such

as the parameters corresponding to 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, Rapp, *et al.* dataset. 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. While we can extract some biological inference, a linear combination of sequence parameters does not explicitly reflect the reality of physical limits for host cells. To some extent, constraint information is likely 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 likely hone predictive power and more intricately characterize the interplay of variables that underlie membrane protein expression in *E. coli* and other systems.

The ability to predict phenotypic results from 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 variables related to expression are discovered, future work will incorporate these leading to improved models. Based on these results, expanding to new expression hosts such as eukaryotes seems entirely feasible, although a number of new parameters 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 membrane proteins 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 becomes more easily accessible, broadly leveraging diverse experimental outcomes to decode sequence-level information, an extension of this work, is anticipated.

## Acknowledgements

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[40], and Welison Floriano and Naveed Near-Ansari for maintaining local computing resources. We thank James Bowie, Michiel Niesen, Stephen Marshall, Thomas Miller, Reid van Lehn, and Tom Rapoport for critical reading of the manuscript. Models and analyses are possible thanks to raw experimental data provided by Daniel Daley and Mikaela Rapp[19]; Nir Fluman[20]; Edda Kloppmann, Brian Kloss, and Marco Punta from NYCOMPS[2,3]; Pikyee Ma[25]; Renaud Wagner[27]; and Florent Bernaudat[31]. We acknowledge funding from an NIH Pioneer Award to WMC (5DP1GM105385); a Benjamin M. Rosen graduate fellowship, a NIH/NRSA training grant (5T32GM07616), and a NSF Graduate Research fellowship to SMS; and an Arthur A. Noyes Summer Undergraduate Research Fellowship to NJ. Computational time was provided by Stephen Mayo and Douglas Rees. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575[41].

## Author Contributions

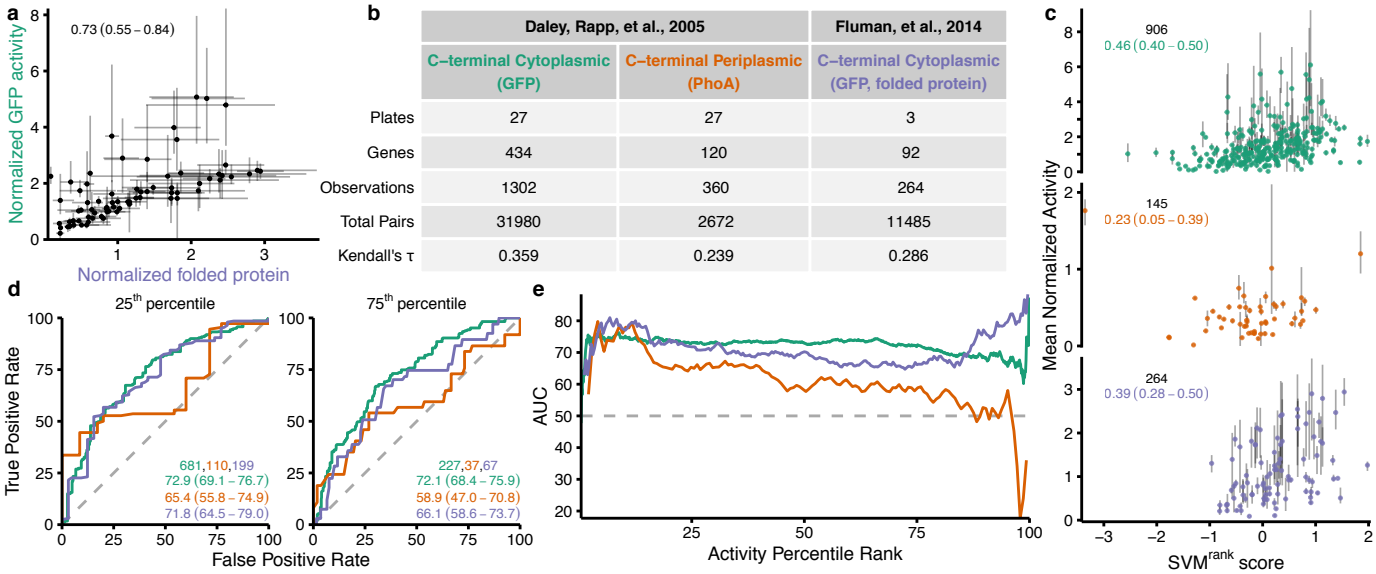
S.M.S., A.M., and W.M.C. conceived the project. S.M.S. developed the approach. S.M.S., A.M., and N.J. compiled sequence and experimental data. N.J. created code to demonstrate feasibility. S.M.S. performed all published calculations. S.M.S. and WMC wrote the manuscript.

## Author Information

The authors declare no competing financial interests.

Correspondence and requests for materials should be addressed to clemons@caltech.edu.

Figure 1




| | Daley, Rapp, et al., 2005 | | Fluman, et al., 2014 |
|---|---|---|---|
| | C–terminal Cytoplasmic (GFP) | C–terminal Periplasmic (PhoA) | C–terminal Cytoplasmic (GFP, folded protein) |
| Plates | 27 | 27 | 3 |
| Genes | 434 | 120 | 92 |
| Observations | 1302 | 360 | 264 |
| Total Pairs | 31980 | 2672 | 11485 |
| Kendall's τ | 0.359 | 0.239 | 0.286 |

**Figure 1.** Training performance. **a**, A comparison of GFP activity[19] with folded protein[20] where each point represents the mean for a given gene tested in both works, and error bars plot the extrema. Spearman's rank correlation coefficient ρ and 95% confidence interval (CI)[42] are shown. **b**, Plates are the number of independent sets of measurements within which expression levels can be reliably compared. Genes are the number of proteins for which the C-terminus was reliably ascertained[19]. Observations are the total number of expression data points accessible. Total pairs are the number of comparable expression measurements (*i.e.* those within a single plate). Kendall's τ is the metric maximized by the training process (See Methods 4b). The color of the column heading identifying each experimental set is retained throughout the figure. **c**, Agreement against the normalized outcomes plotted as the mean activity (see Methods 5 for definition) versus the score with error bars providing the extent of observed activities (Spearman's ρ and 95% CI noted). **d**, Illustrative ROCs for thresholds at $25^{th}$ and $75^{th}$ percentile in activity with the number of positive outcomes at that threshold, the AUC, and 95% CI[43] indicated. **e**, The AUC of the ROC at every possible activity threshold.

Figure 2

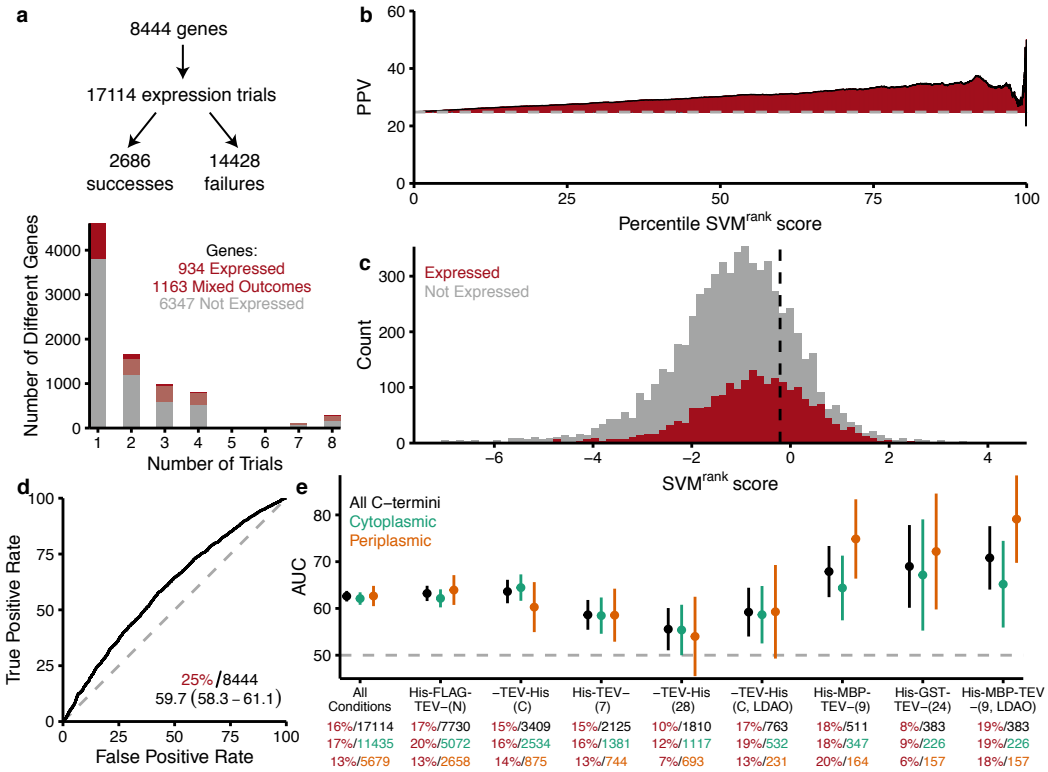

a
8444 genes
17114 expression trials
2686 successes
14428 failures
Number of Different Genes
Genes:
934 Expressed
1163 Mixed Outcomes
6347 Not Expressed
Number of Trials
b
PPV
Percentile SVM^rank score
c
Expressed
Not Expressed
Count
SVM^rank score
d
True Positive Rate
False Positive Rate
25%/8444
59.7 (58.3 – 61.1)
e
All C–termini
Cytoplasmic
Periplasmic
AUC
All Conditions
His-FLAG-TEV–(N)
–TEV-His (C)
His-TEV–(7)
–TEV-His (28)
–TEV-His (C, LDAO)
His-MBP-TEV–(9)
His-GST-TEV–(24)
His-MBP-TEV –(9, LDAO)
16%/17114
17%/11435
13%/5679
17%/7730
20%/5072
13%/2658
15%/3409
16%/2534
14%/875
15%/2125
16%/1381
13%/744
10%/1810
12%/1117
7%/693
17%/763
19%/532
13%/231
18%/511
18%/347
20%/164
8%/383
9%/226
6%/157
19%/383
19%/226
18%/157

**Figure 2.** Success of the model against outcomes from NYCOMPS. **a**, An overview of the NYCOMPS outcomes and a plot of the number of conditions tested per gene with outcomes highlighted. **b**, The PPV plotted for each percentile $SVM^{rank}$ score, *e.g.* 75 on the x-axis indicates the PPV for the top 25% of genes based on score. The grey dashed line shows the ~25% overall success rate of the NYCOMPS experimental outcomes. **c**, Histograms of the total count of proteins at a given $SVM^{rank}$ score colored by NYCOMPS-determined outcomes. **d**, ROC curve, positive (red) and total (black) counts, and AUC values with 95% CI. The grey dashed line shows the performance of a completely random predictor (AUC=50). **e**, The AUCs for all trials together (left) followed by outcomes in individual plasmid and solubilization conditions (DDM except LDAO where noted) along with 95% CI (numerically in Table S2). Performances are also split by predicted C-terminal localization[24]. Overall positive percentage (red) and total number of outcomes within each group is noted below the axis.

Figure 3

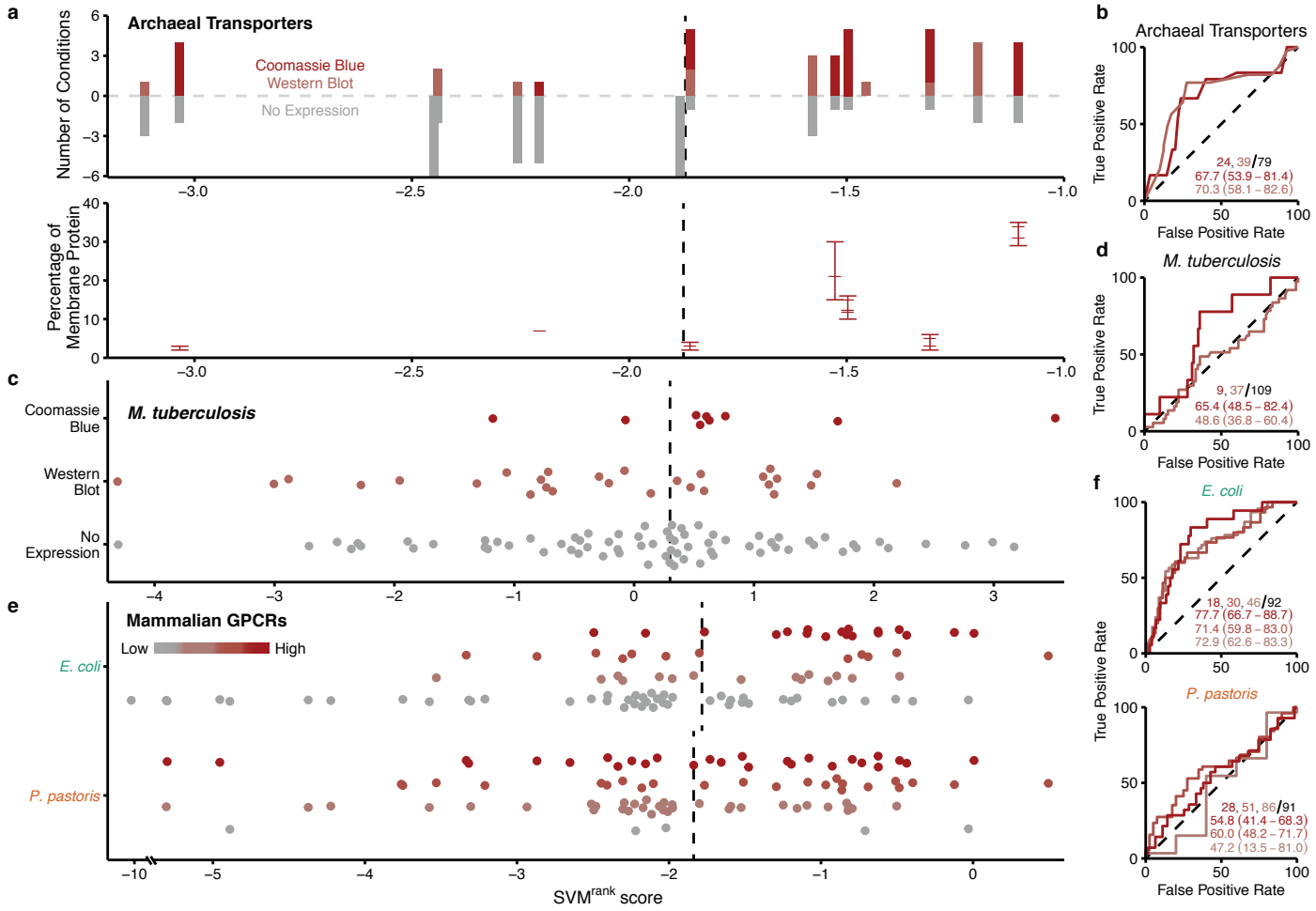

a
Archaeal Transporters
Number of Conditions
Coomassie Blue
Western Blot
No Expression
Percentage of Membrane Protein
b
Archaeal Transporters
True Positive Rate
False Positive Rate
24, 39/79
67.7 (53.9 – 81.4)
70.3 (58.1 – 82.6)
d
M. tuberculosis
9, 37/109
65.4 (48.5 – 82.4)
48.6 (36.8 – 60.4)
c
M. tuberculosis
Coomassie Blue
Western Blot
No Expression
f
E. coli
18, 30, 46/92
77.7 (66.7 – 88.7)
71.4 (59.8 – 83.0)
72.9 (62.6 – 83.3)
P. pastoris
28, 51, 86/91
54.8 (41.4 – 68.3)
60.0 (48.2 – 71.7)
47.2 (13.5 – 81.0)
e
Mammalian GPCRs
Low
High
E. coli
P. pastoris
SVM^rank score

**Figure 3.** Success of the model against a variety of small scale outcomes. For each set, vertical lines indicate the median $SVM^{rank}$ score. ROCs along with AUCs and 95% CI as well as the total number of positives for the given threshold (red hues) along with the total outcomes (black) are presented. In each curve, increasing expression thresholds as defined by the original publication are displayed as deeper red. **a**, The expression of archaeal transporters in up to 6 trials. Top, positive expression count is plotted above the dashed line and negative outcomes below the line. Bottom, from the same work, the expression of proteins detected by Coomassie Blue[25]. **b**, ROC curves for each positive threshold (*i.e.* Coomassie Blue or Western Blot) from trials in **a**. **c**, Experimental expression of *M. tuberculosis* membrane proteins plotted based on outcomes. **d,** ROC curves for each possible threshold from trials in **c**. **e**, Mammalian GPCR expression in either *E. coli* (top) or *P. pastoris* (bottom). **f**, ROC curves for each possible threshold from trials in **e**.

Figure 4

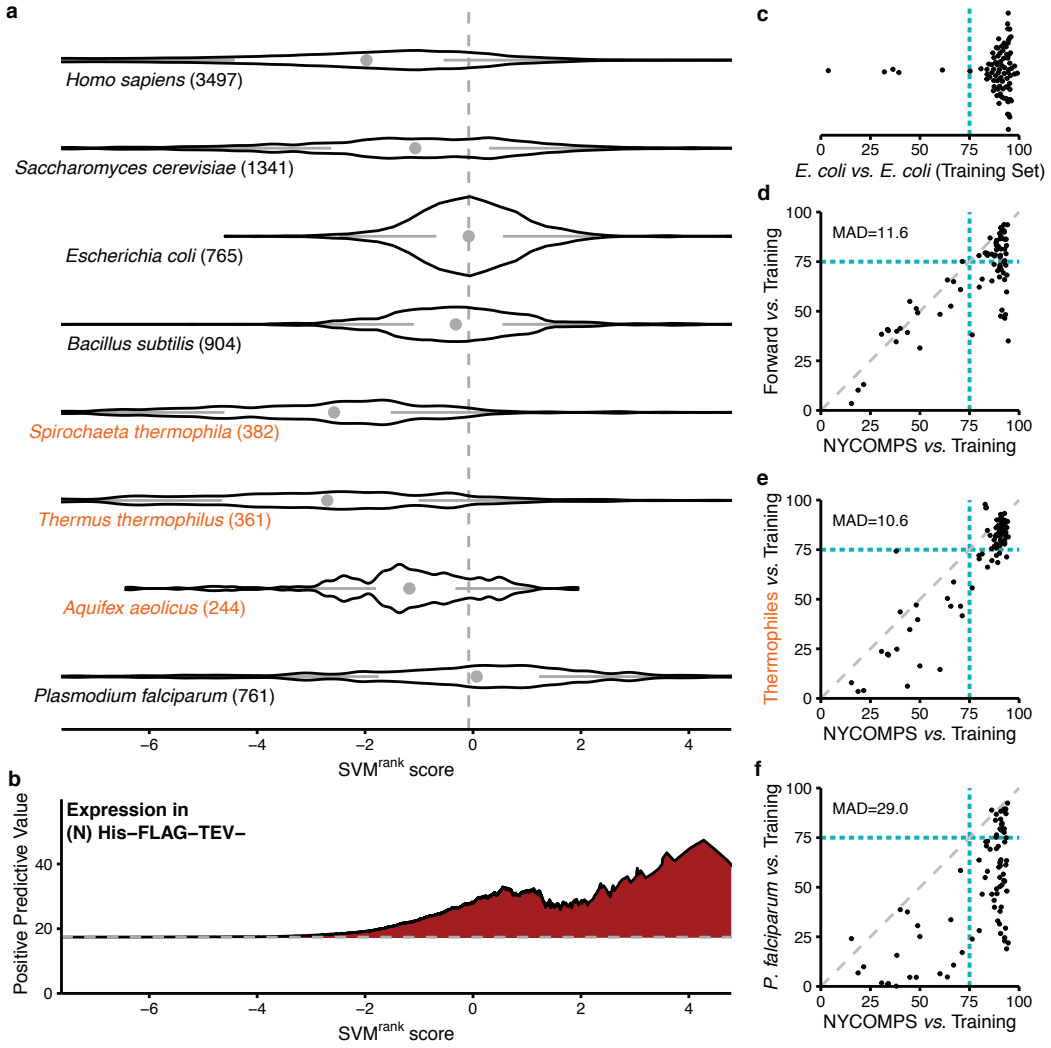

a
Homo sapiens (3497)
Saccharomyces cerevisiae (1341)
Escherichia coli (765)
Bacillus subtilis (904)
Spirochaeta thermophila (382)
Thermus thermophilus (361)
Aquifex aeolicus (244)
Plasmodium falciparum (761)
−6
−4
−2
0
2
4
SVMrank score
b
Expression in
(N) His−FLAG−TEV−
Positive Predictive Value
0
20
40
−6
−4
−2
0
2
4
SVMrank score
c
0
25
50
75
100
E. coli vs. E. coli (Training Set)
d
MAD=11.6
Forward vs. Training
NYCOMPS vs. Training
e
MAD=10.6
Thermophiles vs. Training
NYCOMPS vs. Training
f
MAD=29.0
P. falciparum vs. Training
NYCOMPS vs. Training

**Figure 4.** Forward predictions of membrane protein expression for various genomes. **a**, Calculated scores for proteins from a variety of genomes (count in parentheses; complete set provided in Figure S2a) plotted as contours of kernel density estimates of the number of proteins at a given score. Amplitude is only relative within a genome. The dot indicate the median, and the lines depict quantities of an analogous Tukey boxplot[44,45]. The vertical line shows the median score in *E. coli* to provide context for other distributions. **b**, PPV of the model within the most tested NYCOMPS condition. **c**, Distribution of overlap coefficients (see Methods 7) for each sequence parameter comparing the entire *E. coli* membrane proteome vs. the training set from *E. coli*. The dashed line provides a threshold separating the cluster of highly-related features from those with lower overlap. **d-f**, A comparison of overlap coefficients with the training set between NYCOMPS and **d,** all forward predictions (Figure S2a), **e**, thermophilic genomes (orange), or **f**, *P. falciparum*. Mean Absolute Deviation is indicated for each plot.

Figure 5

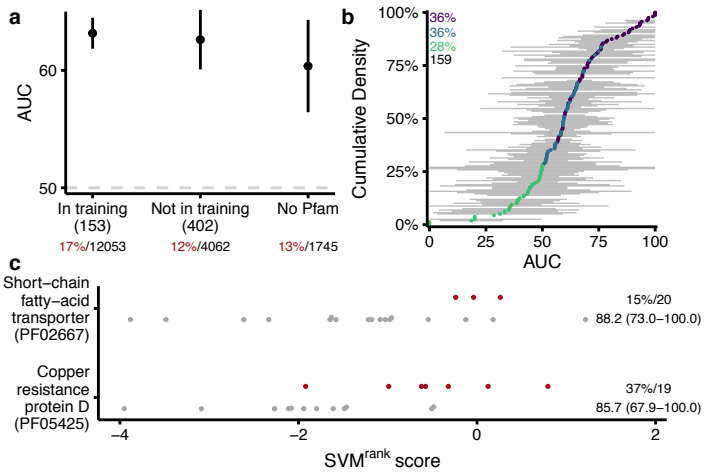

a
AUC
60
50
In training (153)
Not in training (402)
No Pfam
17%/12053
12%/4062
13%/1745
b
Cumulative Density
100%
75%
50%
25%
0%
36%
36%
28%
159
0
25
50
75
100
AUC
c
Short−chain fatty−acid transporter (PF02667)
Copper resistance protein D (PF05425)
15%/20
88.2 (73.0–100.0)
37%/19
85.7 (67.9–100.0)
−4
−2
0
2
SVM^rank score

**Figure 5.** Model performance across protein families **a**, The NYCOMPS dataset is split by those proteins with a Pfam found either in the training set, not in the training set or those without a Pfam (Pfam counts in parentheses). The AUC and 95% confidence interval for each set is plotted with the positive rate (red) and number of trials (black) indicated below the x-axis. **b**, For each family, the AUC across all outcomes is plotted arranged in order of the value, an empirical cumulative distribution function, with horizontal bars indicating the 95% confidence interval. The color indicates the significance of the prediction within the family: purple, predictive at 95% confidence, blue, predictive but not at 95% confidence, green, not predictive. The size of each significance group and total number of families (black) are indicated on the plot. **c**, Outcomes for specific protein families. Each was only tested in a single condition (N). The overall positive percentage within the group, total number of outcomes, and AUC with 95% CI is labelled to the right.

Figure 6

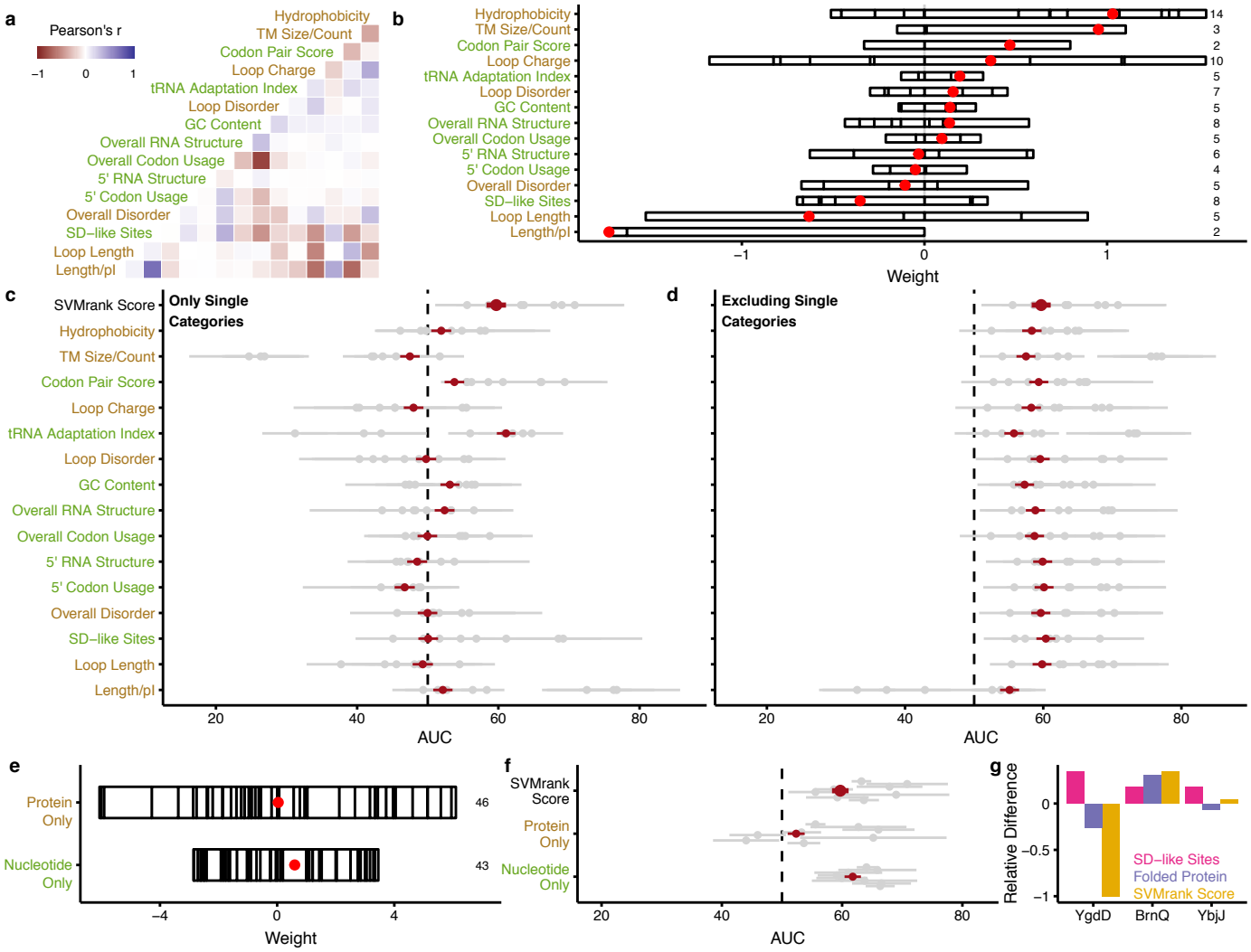
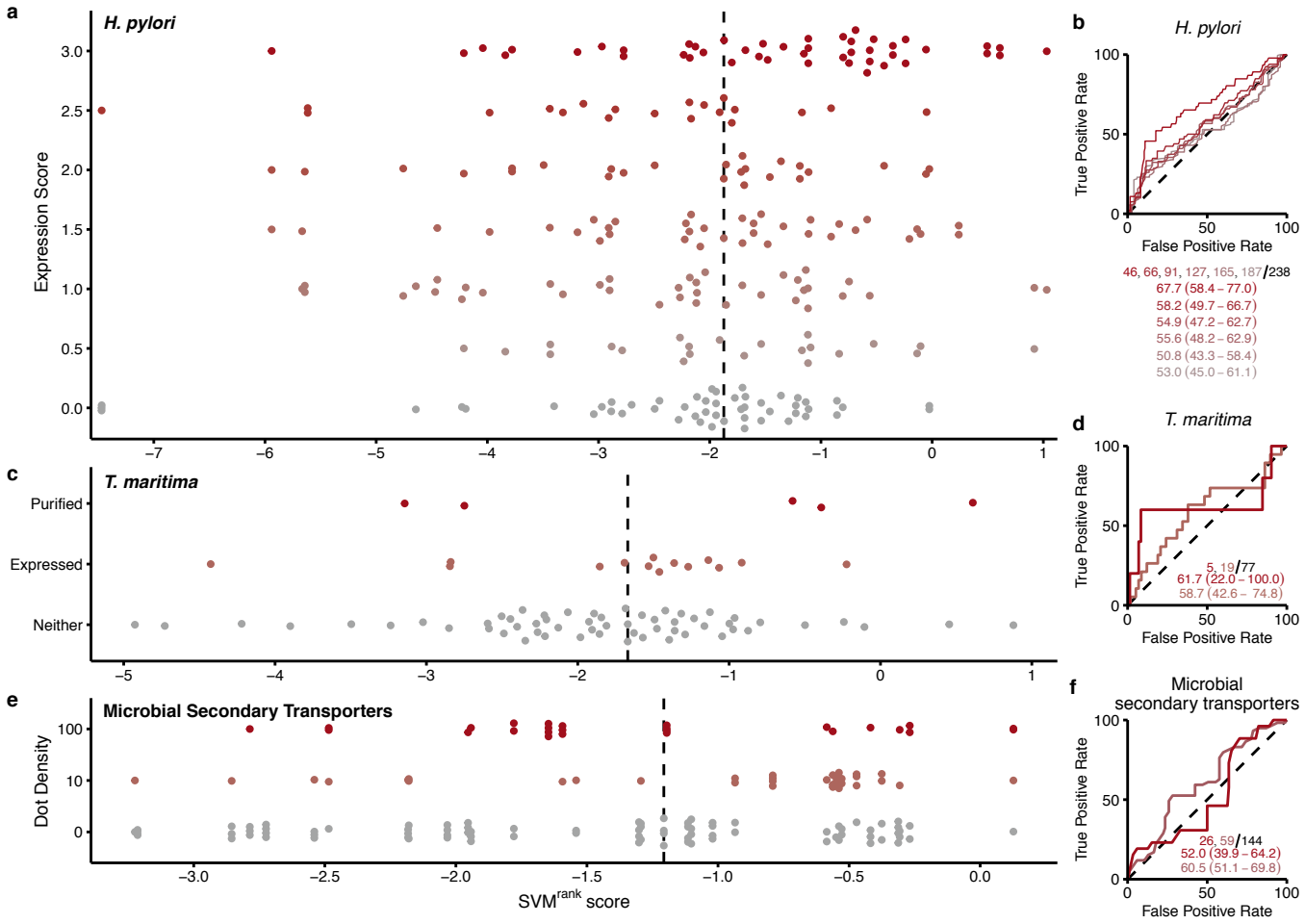

a
Pearson's r
−1
0
1
Hydrophobicity
TM Size/Count
Codon Pair Score
Loop Charge
tRNA Adaptation Index
Loop Disorder
GC Content
Overall RNA Structure
Overall Codon Usage
5' RNA Structure
5' Codon Usage
Overall Disorder
SD−like Sites
Loop Length
Length/pI
b
Hydrophobicity 14
TM Size/Count 3
Codon Pair Score 2
Loop Charge 10
tRNA Adaptation Index 5
Loop Disorder 7
GC Content 5
Overall RNA Structure 8
Overall Codon Usage 5
5' RNA Structure 6
5' Codon Usage 4
Overall Disorder 5
SD−like Sites 8
Loop Length 5
Length/pI 2
−1
0
1
Weight
c
SVMrank Score
Only Single Categories
Hydrophobicity
TM Size/Count
Codon Pair Score
Loop Charge
tRNA Adaptation Index
Loop Disorder
GC Content
Overall RNA Structure
Overall Codon Usage
5' RNA Structure
5' Codon Usage
Overall Disorder
SD−like Sites
Loop Length
Length/pI
20
40
60
80
AUC
d
Excluding Single Categories
20
40
60
80
AUC
e
Protein Only
46
Nucleotide Only
43
−4
0
4
Weight
f
SVMrank Score
Protein Only
Nucleotide Only
20
40
60
80
AUC
g
Relative Difference
0
−0.5
−1
SD−like Sites
Folded Protein
SVMrank Score
YgdD
BrnQ
YbjJ

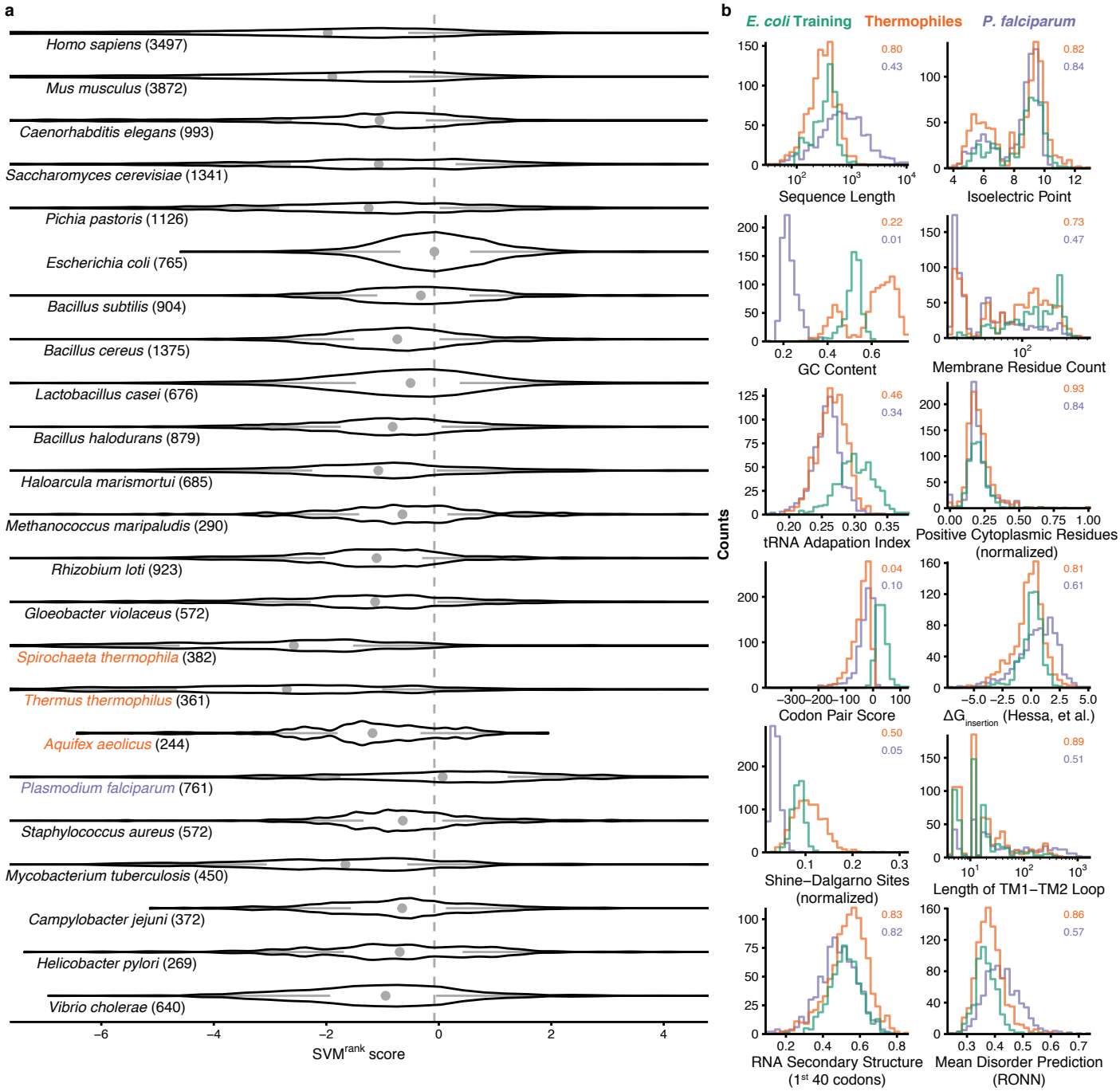
**Figure 6.** Feature contributions to the model **a**, Pearson correlation coefficients between feature categories are shown. Feature labels are green for protein-sequence derived and brown for nucleotide-sequence derived. **b**, Total weight for each category is represented as a bar. The contribution of each feature to the category is shown by partitioning the bar. The red dot indicates the total sum of weights within the category. **c**, The AUC and 95% confidence interval when predicting with the entire model ($SVM^{rank}$ score) or single category specified on the NYCOMPS dataset. Red shows the outcome of predicting at the level of individual genes (Figure 2b-d) and grey show the outcome within each vector individually (as in Figure 2e). **d**, The AUC and 95% confidence interval when predicting with the entire model ($SVM^{rank}$ score) or by excluding single category specified on the NYCOMPS dataset. **e**, As **b**, but classifying features by the type of sequence they are calculated from. **f**, The AUC and 95% confidence intervals using only protein or nucleotide features. **g**, Relative difference in SD-like sites (green), expression (purple), and $SVM^{rank}$ score (yellow) between wild-type and mutants with silent mutations. See Methods 7 for further detail.

Supplementary Figure 1



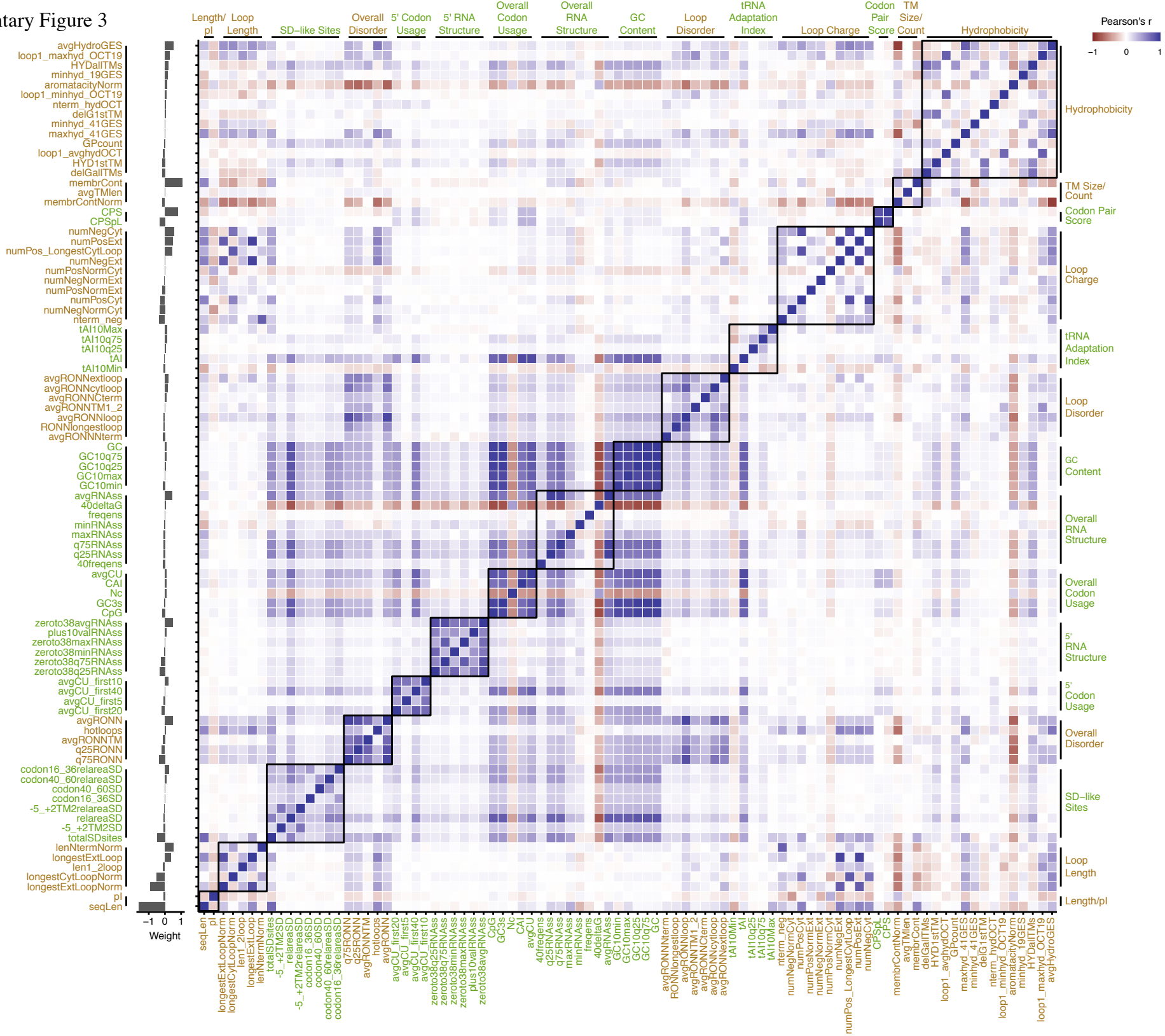
**Supplementary Figure 1.** Additional small-scale predictions and outcomes. **a**, Experimental expression of 116 *H. pylori* membrane proteins in *E. coli* in at most 3 vectors (238 trials) scored as either a 1, 2, or 3 from the outcome of a dot blot as well as Coomassie Staining of an SDS-PAGE gel for two of the vectors. To compare the three vectors with a single set of scores, the two scores were averaged to give a single number for a condition making them comparable to the third vector while yielding 2 additional thresholds (1.5 and 2.5) and the 6 total levels shown. **b**, The ROC with each cutoff is plotted, where a higher cutoff is represented by a deeper red, followed by the AUCs (directly below) in colors that correspond to the respective curve. **c**, Expression of 77 *T. maritima* membrane proteins in *E. coli* noted as purified (5), not purified but expressed (14), or neither. **d**, ROC curve for each threshold. **e**, Expression of 37 microbial secondary transporters in 4 IPTG-inducible vectors (144 trials) in *E. coli* quantified as 10 ng/mL (pink) or 100 ng/mL (red) via dot blot. **f**, ROC curve for each threshold.

Supplementary Figure 2



**Supplementary Figure 2.** Complete set of forward predictions. **a**, Extended from Figure 4c, the full complement of score distributions calculated by genome is plotted and arranged to accentuate similar features by physiology, *e.g.* growth condition, or scientific interest, *e.g.* pathogenic. Raw scores along with sequence identifiers are available in the Table S3. **b**, Histograms of representative sequence parameters between the training data set (green), thermophiles (orange), and *P. falciparum* (purple). Values for sequence parameter overlap coefficients derived from kernel density estimates (Methods 7) versus the *E. coli* training data are included. See Table S1 for parameter descriptions.

Supplementary Figure 3



**Supplementary Figure 3.** Complete set of feature correlations and their individual contributions to the model. Features are ordered first by category (as in Figure 5) and then by weight (grey bars). Labels are green for protein-sequence derived and brown for nucleotide-sequence derived features. Pearson correlation coefficient between each pair of features across the NYCOMPS dataset is plotted (right). See Table S1 for a detailed description of each feature. Feature categories are overlaid as square boxes and indicated by black bars on the top, left, and right of the correlation matrix.

**Table S1 (separate file)**

Sequence parameter weights and descriptions. Weights are presented after normalizing to the mean value for clarity. Parameters that were calculated but removed in pre-processing are noted (Methods 3).

**Table S2 (separate file)**

AUC values for the NYCOMPS dataset. AUC values and 95% confidence intervals are presented in summary, by expression condition, and by predicted C-terminal localization as well as for $SVM^{rank}$ scores calculated without the most computationally expensive RNA secondary structure calculation (as in Figure 4).

**Table S3 (separate file)**

Full list of predicted membrane proteins from the various genomes in Supplementary Figure 2 with corresponding identifiers, descriptions, Pfam families, coding sequences, and $SVM^{rank}$ scores. This data is available in an interactive format at clemonslab.caltech.edu.

**Table S4 (separate file)**

Predictive performances of the model across protein families tested by NYCOMPS summarized in Figure 5. This data is available in an interactive format at clemonslab.caltech.edu.

## Materials and Methods

Sequence mapping & retrieval and feature calculation was performed in Python 2.7[46] using BioPython[47] and NumPy[48]; executed and consolidated using Bash (shell) scripts; and parallelized where possible using GNU Parallel[49]. Data analysis and presentation was done in R[50] within RStudio[51] using magrittr[52], plyr[53], dplyr[54], asbio[55], and datamart[56] for data handling; ggplot2[57], ggbeeswarm[58], GGally[59], gridExtra[60], cowplot[61], scales[62], viridis[63], and RColorBrewer[64,65] for plotting; multidplyr[66] with parallel[50] and foreach[67] with iterators[68] and doMC[69]/doParallel[70] for parallel processing; and roxygen2[71] for code organization and documentation as well as other packages as referenced.

<u>1. Collection of data necessary for learning and evaluation</u>

*E. coli Sequence Data* – The nucleotide sequences from [19] 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 (personal communication). To account for sequence and annotation corrections made to the genome after Daley, Rapp, *et al.*'s work, these primers were directly used to reconstruct the amplified product from the most recent release of the *E. coli* K-12 substr. MG1655 genome[72] (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 this data and parse these reads would not significantly impact the model. The deduced nucleotide sequences were verified against the protein lengths given in Table S1 from [19]. The plasmid library tested in [20] 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 [19] that these do not correlate with sequence features either in the univariate sense (many simple linear regressions, Table S1[19]) 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 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 was no longer accessible (personal communication). 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 [20].

*New York Consortium on Membrane Protein Structure (NYCOMPS) Data* – Brian Kloss, Marco Punta, and Edda Kloppman provided a dataset 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 either via the Entrez system[73] or the Uniprot mapping service[74]. 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. Though a custom script was created, a script from Peter Cock on the BioPython listserv to do the same task via a similar mapping mechanism was found[75]. To confirm all the sequences, the TargetTrack[76] XML file was parsed for the internal NYCOMPS identifiers and compared for sequence identity to those that had been

mapped using the custom script; 20 (less than 1%) of the sequences had minor inconsistencies and were manually replaced.

*Archaeal transporters Data* – The locus tags ("Gene Name" in Table 1) were mapped directly to the sequences and retrieved from NCBI[25]. 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 from [27] to protein GIs via the Uniprot mapping service[74] and subsequently to their nucleotide sequences via the custom mapping script described above (see NYCOMPS). The sequence length and pI 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 Supplemental Table 1 from [28] to locus tags in the Jan 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 Jan 31, 2014 genome version, were retrieved from the Apr 9, 2015 release of the genome (NC_000915.1). Both releases are derived from the original sequencing project[77]. 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 to the third vector while yielding 2 additional thresholds (1.5 and 2.5) result in the 5 total curves shown (Figure S1b).

*Mycobacterium tuberculosis Data* – The authors note using TubercuList through GenoList[78], therefore, nucleotide sequences were retrieved from the archival website based on the original sequencing project[79]. The sequences corresponding to the identifiers and outcomes in Table 1 from [26] were validated against the provided molecular weight .

*Secondary Transporter Data* – GI Numbers given in Table 1 from [30] were matched to their CDS entries using the custom mapping script described above (see NYCOMPS). Only expression in *E. coli* with IPTG-inducible vectors was considered.

*Thermotoga maratima Data* – Gene names given in Table 1[80] were matched to CDS entries in the Jan 31, 2014 release of the *Thermotoga maritima* MSB8 genome (AE000512.1), a revised annotation of the original release[81]. The sequence length and molecular weight were validated against those provided.

2. Calculation of sequence features

Based on experimental analyses and anecdotal evidence, approximately 105 different protein and nucleotide sequence features thought to be relevant to expression were identified and calculated for each protein using custom code together with published software (codonW[82], tAI[83], NUPACK[40], Vienna RNA[84], Codon Pair Bias[85], Disembl[18], and RONN[86]). Relative metrics (*e.g.* codon adaptation index) are calculated with respect to the *E. coli* K-12 substr. MG1655[72] quantity. The octanol-water partitioning[87], GES hydrophobicity[88], ΔG of insertion[16] scales were employed as well. Transmembrane segment

topology was predicted using Phobius Constrained for the training data and Phobius for all other datasets[24]. Two RNA secondary structure metrics were prompted in part by [14]. 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 lists a description of each feature. Features are calculated solely from a gene of interest excluding portions of the ORFs such as linkers and tags derived from the plasmid backbone employed (future work will explore contributions of these elements).

3. Preparation for model learning

Calculated sequence features for the membrane proteins in the *E. coli* dataset 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 standard deviation of that feature using `preprocess`[89]. 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`)[89]. 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 over represent certain redundant features. Features and outcomes for each list ("query") were written into the SVM$^{light}$ format using a modified `svmlight.write`[90].

The final features were calculated for each sequence in the test datasets, prepared for scoring by "centering" and "scaling" by the training set parameters via `preprocess`[89], and then written into SVM$^{light}$ format again using a modified `svmlight.write`.

4. Model selection, training, and evaluation using $SVM^{rank}$

*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 ($w_1$, $w_2$, ..., $w_N$). 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[19] 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 $\tau$[91]. For example, consider the following data generated via context A $(X_{\mathrm{A},1}, Y_{\mathrm{A},1}), (X_{\mathrm{A},2}, Y_{\mathrm{A},2})$ and B $(X_{\mathrm{B},1}, Y_{\mathrm{B},1}), (X_{\mathrm{B},2}, Y_{\mathrm{B},2})$ where observed response follows as index $i$, *i.e.* $Y_n < Y_{n+1}$. Binary classifier $f\,(X_i, X_j)$ gives a score of 1 if an input pair matches its ordering criteria and $-1$ if not, *i.e.* $Y_i < Y_j$:

$$f(X_{\mathrm{A},1}, X_{\mathrm{A},2}) = 1; f(X_{\mathrm{A},2}, X_{A,1}) = -1$$

$$f(X_{\mathrm{B},1}, X_{\mathrm{B},2}) = 1; f(X_{\mathrm{B},2}, X_{\mathrm{B},1}) = -1$$

$$f(X_{\mathrm{A},1}, X_{\mathrm{B},2}), f(X_{\mathrm{A},2}, X_{\mathrm{B},1}) \text{ are invalid}$$

Free parameters describing $f$ are calculated such that those calculated orderings $f(X_{\mathrm{A},1}), f(X_{\mathrm{A},2}) \ldots$; $f(X_{\mathrm{B},1}), f(X_{\mathrm{B},2}) \ldots$ most closely agree (overall Kendall's τ) with the observed ordering $Y_n, Y_{n+1}, \ldots$. 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[92]. 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 hyper-plane in the feature space between two classes, where class membership is determined by which side of the hyper-plane points reside. For some $n$ linear separable training examples $D = \{ (x_i) | \, x_i \, \epsilon \, \mathbb{R}^d \}^n$ and two classes $y_i \, \epsilon \, \{-1, 1\}$, a linear SVM seeks a mapping from the *d*-dimensional feature space $\mathbb{R}^d \rightarrow \{-1, 1\}$ by finding two maximally separated hyperplanes $w \cdot x - b = 1$ and $w \cdot x - b = -1$ with constraints that $w \cdot x_i - b \geq 1$ for all $x_i$ with $y_i \, \epsilon \, \{1\}$ and $w \cdot x_i - b \leq -1$ for all $x_i$ with $y_i \, \epsilon \, \{-1\}$. The feature weights correspond to the vector *w*, which is the vector perpendicular to the separating hyperplanes, and are computable in $\mathrm{O}(n \log n)$ implemented as part of the SVM$^{\mathrm{rank}}$ software package, though in $\mathrm{O}(n^2)$[22]. See [92] 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, Rapp, *et al*. with a grid of candidate $C$ values ($2^n \; \forall \, n \, \epsilon \, [-5, \; 5]$) and then evaluated against the raw "folded protein" measurements from Fluman, *et al*. 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 to 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 they do not come at the cost of increased false positive diagnoses. This is formalized in Receiver Operating Characteristic (ROC) theory (for a primer see [23]), 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 area under an ROC curve (AUC) where $AUC_{perfect}$=100% and $AUC_{random}$=50% (percentage signs are omitted throughout the text and figures). All ROCs are calculated through pROC[93] using the analytic Delong method for AUC confidence intervals[43]. Bootstrapped AUC CIs ($N = 10^6$) were precise to 4 decimal places suggesting that analytic CIs are valid for the NYCOMPS dataset.

With several of our datasets, 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 dataset in order to evaluate predictive performance as follows:

*Training data* – The outcomes are quantitative (activity level), so each ROC is calculated by normalizing within each dataset to the standard well subject to the discussion in 4a above (LepB for PhoA, and InvLepB for GFP) (examples in Figure 1D) for each possible threshold, *i.e.* each normalized

expression value with each AUC plotted in Figure 1e. 95% confidence intervals of Spearman's ρ are given by $10^6$ iterations of a bias-corrected and accelerated (BCa) bootstrap of the data (Figure 1a,c)[42].

*Large-scale* – ROCs were calculated for each of the expression classes (Figure 2e). Regardless of the split, predictive performance is noted. The binwidth for the histogram was determined using the Freedman-Diaconis rule, 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 (Figure 3b,d,f, Figure S1b,d,f). See Methods 1 for any necessary details about outcome classifications for each dataset.

## 6. Feature Weights

Weights for the learned SVM are pulled directly from the model file produced by $SVM^{light}$ and are given in Table S1 after normalizing to the mean value.

## 7. Forward Predictions

*Data collection* – We selected several genomes for comparison as shown in Figure 4, Figure S2a, and Table S3. Coding sequences of membrane proteins from human and mouse genomes were gathered by mapping Uniprot identifiers of proteins noted to have at least one transmembrane segment by Uniprot[74] to Ensembl (release 82) coding sequences[94] via Biomart.[95] *C. elegans* coding sequences were similarly mapped via Uniprot but to WormBase coding sequences[96] also via Biomart. *S. cerevisiae* strain S288C coding sequences[97] were retrieved from the Saccharomyces Genome Database. *P. pastoris* strain GS115 coding sequences[98] were retrieved from the DOE Joint Genome Institute (JGI) Genome Portal[99]. Those sequences without predicted[24] TMs were excluded from subsequent analyses. Microbial sequences were

gathered via a custom, in-house database populated with data compiled primarily from Pfam[36], DOE JGI Integrated Microbial Genomes[100], and the Microbial Genome Database[101].

*Feature calculation* – Because of the incredible number of sequences, we did not calculate the features derived from the most computationally expensive calculation (whole sequence mRNA pairing probability). Since predictive performance on the NYCOMPS dataset is slightly smaller, but not significantly different at 95% confidence, in the absence of these features (Table S2), the forward predictions are still valid. For future experiments, these features can be calculated for the subset of targets of interest.

*Parameter space similarity* – As a first approximation of the similarity of the ~90 dimensional sequence parameter space between two groupings, features were compared pairwise via the following metric. Let $f_i$ and $g_i$ represent the true distributions for a given feature $i$ between two groups of interest. The distribution overlap, *i.e.* shared area, $\Delta_i$ is formalized as

$$\Delta_i(f_i, g_i) = \int \min\{f_i(x), g_i(x)\}\, dx$$

ranging from 0, for entirely distinct distributions, to 1 for entirely identical distributions.

As written $f_i$ and $g_i$ are probability densities, they need to be approximated before calculating $\Delta_i$ and are done so via kernel density estimates (KDE) of the observed samples $[x_1^f, \ldots, x_n^f]$ and $[x_1^g, \ldots, x_n^g]$ using a nonparametric, locally adaptive method allowing for variable bandwidth smoothing implemented in LocFit[102] (`adpen`=$2\sigma^2$) providing $\hat{f}_i$ and $\hat{g}_i$. The distribution overlap $\Delta_i$ is evaluated over a grid of $2^{13}$ equally spaced points over the range of $f_i$ and $g_i$.

*Shine-Dalgarno-like mutagenesis* – Folded protein is quantified by densitometry measurement[103,104] of the relevant band in Figure 6 of [20]. Relative difference is calculated as is standard:

$$\frac{\text{metric}_{\text{mutant}} - \text{metric}_{\text{wildtype}}}{\frac{1}{2}\left|\text{metric}_{\text{mutant}} - \text{metric}_{\text{wildtype}}\right|}$$

## 8. Availability

All analysis is documented in a series of R notebooks[105] available openly at github.com/clemlab/ml-ecoli-svmrank. These notebooks provide fully executable instructions for the reproduction of the analyses and the generation of figures and statistics in this study. The ranking engine is available as a web service at clemonslab.caltech.edu. Additional code is available upon request.

## References


1. White, S. H. Biophysical dissection of membrane proteins. *Nature* **459,** 344–346 (2009).
2. Punta, M. *et al.* Structural genomics target selection for the New York consortium on membrane protein structure. *J. Struct. Funct. Genomics* **10,** 255–268 (2009).
3. Love, J. *et al.* The New York Consortium on Membrane Protein Structure (NYCOMPS): a high-throughput platform for structural genomics of integral membrane proteins. *J. Struct. Funct. Genomics* **11,** 191–199 (2010).
4. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Res.* **28,** 235–242 (2000).
5. Bill, R. M. *et al.* Overcoming barriers to membrane protein structure determination. *Nat. Biotechnol.* **29,** 335–340 (2011).
6. Nørholm, M. H. H. *et al.* Manipulating the genetic code for membrane protein production: what have we learnt so far? *Biochim. Biophys. Acta* **1818,** 1091–1096 (2012).
7. Wagner, S. *et al.* Tuning Escherichia coli for membrane protein overexpression. *Proc. Natl. Acad. Sci. U. S. A.* **105,** 14371–14376 (2008).
8. Lewinson, O., Lee, A. T. & Rees, D. C. The funnel approach to the precrystallization production of membrane proteins. *J. Mol. Biol.* **377,** 62–73 (2008).
9. Marshall, S. S. *et al.* A Link between Integral Membrane Protein Expression and Simulated Integration Efficiency. *Cell Rep.* **16,** 2169–2177 (2016).
10. Sarkar, C. A. *et al.* Directed evolution of a G protein-coupled receptor for expression, stability, and binding selectivity. *Proc. Natl. Acad. Sci. U. S. A.* **105,** 14808–14813 (2008).
11. Schlinkmann, K. M. *et al.* Critical features for biosynthesis, stability, and functionality of a G protein-coupled receptor uncovered by all-versus-all mutations. *Proc. Natl. Acad. Sci. U. S. A.* **109,** 9810–9815 (2012).
12. Seppälä, S., Slusky, J. S., Lloris-Garcerá, P., Rapp, M. & von Heijne, G. Control of membrane protein topology by a single C-terminal residue. *Science* **328,** 1698–1700 (2010).
13. Van Lehn, R. C., Zhang, B. & Miller, T. F. Regulation of multispanning membrane protein topology via post-translational annealing. *eLife* **4,** (2015).
14. Goodman, D. B., Church, G. M. & Kosuri, S. Causes and effects of N-terminal codon bias in bacterial genes. *Science* **342,** 475–479 (2013).
15. Mirzadeh, K. *et al.* Enhanced Protein Production in Escherichia coli by Optimization of Cloning Scars at the Vector-Coding Sequence Junction. *ACS Synth. Biol.* (2015). doi:10.1021/acssynbio.5b00033
16. Hessa, T. *et al.* Molecular code for transmembrane-helix recognition by the Sec61 translocon. *Nature* **450,** 1026–1030 (2007).
17. Heijne, G. The distribution of positively charged residues in bacterial inner membrane proteins correlates with the trans-membrane topology. *EMBO J.* **5,** 3021–3027 (1986).
18. Linding, R. *et al.* Protein disorder prediction: implications for structural proteomics. *Structure* **11,** 1453–1459 (2003).
19. Daley, D. O. *et al.* Global topology analysis of the Escherichia coli inner membrane proteome. *Science* **308,** 1321–1323 (2005).
20. Fluman, N., Navon, S., Bibi, E. & Pilpel, Y. mRNA-programmed translation pauses in the targeting of E. coli membrane proteins. *eLife* **3,** (2014).
21. Geertsma, E. R., Groeneveld, M., Slotboom, D.-J. & Poolman, B. Quality control of overexpressed membrane proteins. *Proc. Natl. Acad. Sci. U. S. A.* **105,** 5722–5727 (2008).
22. Tsochantaridis, I., Joachims, T., Hofmann, T. & Altun, Y. Large Margin Methods for Structured and Interdependent Output Variables. *J. Mach. Learn. Res.* **6,** 1453–1484 (2005).

23. Swets, J. A., Dawes, R. M. & Monahan, J. Better decisions through science. *Sci. Am.* **283,** 82–87 (2000).
24. Käll, L., Krogh, A. & Sonnhammer, E. L. L. A combined transmembrane topology and signal peptide prediction method. *J. Mol. Biol.* **338,** 1027–1036 (2004).
25. Ma, P. *et al.* An efficient strategy for small-scale screening and production of archaeal membrane transport proteins in Escherichia coli. *PloS One* **8,** e76913 (2013).
26. Korepanova, A. *et al.* Cloning and expression of multiple integral membrane proteins from Mycobacterium tuberculosis in Escherichia coli. *Protein Sci.* **14,** 148–158 (2005).
27. Lundstrom, K. *et al.* Structural genomics on membrane proteins: comparison of more than 100 GPCRs in 3 expression systems. *J. Struct. Funct. Genomics* **7,** 77–91 (2006).
28. Psakis, G. *et al.* Expression screening of integral membrane proteins from Helicobacter pylori 26695. *Protein Sci.* **16,** 2667–2676 (2007).
29. Dobrovetsky, E. *et al.* High-throughput production of prokaryotic membrane proteins. *J. Struct. Funct. Genomics* **6,** 33–50 (2005).
30. Surade, S. *et al.* Comparative analysis and 'expression space' coverage of the production of prokaryotic membrane proteins for structural genomics. *Protein Sci.* **15,** 2178–2189 (2006).
31. Bernaudat, F. *et al.* Heterologous expression of membrane proteins: choosing the appropriate host. *PloS One* **6,** e29191 (2011).
32. Eshaghi, S. *et al.* An efficient strategy for high-throughput expression screening of recombinant integral membrane proteins. *Protein Sci.* **14,** 676–683 (2005).
33. Gordon, E. *et al.* Effective high-throughput overproduction of membrane proteins in Escherichia coli. *Protein Expr. Purif.* **62,** 1–8 (2008).
34. Petrovskaya, L. E. *et al.* Expression of G-protein coupled receptors in Escherichia coli for structural studies. *Biochem. Mosc.* **75,** 881–891 (2010).
35. Szakonyi, G. *et al.* A genomic strategy for cloning, expressing and purifying efflux proteins of the major facilitator superfamily. *J. Antimicrob. Chemother.* **59,** 1265–1270 (2007).
36. Finn, R. D. *et al.* Pfam: the protein families database. *Nucleic Acids Res.* **42,** D222–230 (2014).
37. Li, G.-W., Oh, E. & Weissman, J. S. The anti-Shine-Dalgarno sequence drives translational pausing and codon choice in bacteria. *Nature* **484,** 538–541 (2012).
38. Gamble, C. E., Brule, C. E., Dean, K. M., Fields, S. & Grayhack, E. J. Adjacent Codons Act in Concert to Modulate Translation Efficiency in Yeast. *Cell* **166,** 679–690 (2016).
39. Chartron, J. W., Hunt, K. C. L. & Frydman, J. Cotranslational signal-independent SRP preloading during membrane targeting. *Nature* **536,** 224–228 (2016).
40. Zadeh, J. N. *et al.* NUPACK: Analysis and design of nucleic acid systems. *J. Comput. Chem.* **32,** 170–173 (2011).
41. Towns, J. *et al.* XSEDE: Accelerating Scientific Discovery. *Comput. Sci. Eng.* **16,** 62–74 (2014).
42. Canty, A. & Ripley, B. D. *boot: Bootstrap R (S-Plus) Functions*. (2015).
43. DeLong, E. R., DeLong, D. M. & Clarke-Pearson, D. L. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics* **44,** 837–845 (1988).
44. Tukey, J. W. *Exploratory data analysis*. (Addison-Wesley Pub. Co, 1977).
45. Tufte, E. R. *The visual display of quantitative information*. (Graphics Press, 2001).
46. Van Rossum, G. & Drake Jr, F. L. *Python reference manual*. (Centrum voor Wiskunde en Informatica Amsterdam, 1995).
47. Cock, P. J. A. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25,** 1422–1423 (2009).

48. van der Walt, S., Colbert, S. C. & Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Comput. Sci. Eng.* **13,** 22–30 (2011).
49. Tange, O. GNU Parallel - The Command-Line Power Tool. *Login USENIX Mag.* **36,** 42–47 (2011).
50. R Core Team. *R: A Language and Environment for Statistical Computing*. (R Foundation for Statistical Computing, 2015).
51. RStudio Team. *RStudio: Integrated Development Environment for R*. (RStudio, Inc., 2015).
52. Bache, S. M. & Wickham, H. *magrittr: A Forward-Pipe Operator for R*. (2014).
53. Wickham, H. The Split-Apply-Combine Strategy for Data Analysis. *J. Stat. Softw.* **40,** 1–29 (2011).
54. Wickham, H. & Francois, R. *dplyr: A Grammar of Data Manipulation*. (2015).
55. Aho, K. *asbio: A Collection of Statistical Tools for Biologists*. (2015).
56. Weinert, K. *datamart: Unified access to your data sources*. (2014).
57. Wickham, H. *ggplot2: elegant graphics for data analysis*. (Springer New York, 2009).
58. Clarke, E. & Sherrill-Mix, S. *ggbeeswarm: Categorical Scatter (Violin Point) Plots*. (2015).
59. Schloerke, B. *et al. GGally: Extension to 'ggplot2'*. (2016).
60. Auguie, B. *gridExtra: Miscellaneous Functions for 'Grid' Graphics*. (2015).
61. Wilke, C. O. *cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'*. (2015).
62. Wickham, H. *scales: Scale Functions for Visualization*. (2015).
63. Garnier, S. *viridis: Default Color Maps from 'matplotlib'*. (2016).
64. Neuwirth, E. *RColorBrewer: ColorBrewer Palettes*. (2014).
65. Harrower, M. & Brewer, C. A. ColorBrewer.org: an online tool for selecting colour schemes for maps. *Cartogr. J.* **40,** 27–37 (2003).
66. Wickham, H. *multidplyr: Partitioned data frames for 'dplyr'*.
67. Revolution Analytics & Weston, S. *foreach: Provides Foreach Looping Construct for R*. (2015).
68. Revolution Analytics & Weston, S. *iterators: Provides Iterator Construct for R*. (2015).
69. Revolution Analytics & Weston, S. *doMC: Foreach Parallel Adaptor for 'parallel'*. (2015).
70. Revolution Analytics & Weston, S. *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*. (2015).
71. Wickham, H., Danenberg, P. & Eugster, M. *roxygen2: In-Source Documentation for R*. (2015).
72. Zhou, J. & Rudd, K. E. EcoGene 3.0. *Nucleic Acids Res.* **41,** D613–624 (2013).
73. Schuler, G. D., Epstein, J. A., Ohkawa, H. & Kans, J. A. Entrez: molecular biology database and retrieval system. *Methods Enzymol.* **266,** 141–162 (1996).
74. UniProt Consortium. Reorganizing the protein space at the Universal Protein Resource (UniProt). *Nucleic Acids Res.* **40,** D71–75 (2012).
75. Cock, P. [BioPython] Downloading CDS sequences. (2009).
76. Chen, L., Oughtred, R., Berman, H. M. & Westbrook, J. TargetDB: a target registration database for structural genomics projects. *Bioinformatics* **20,** 2860–2862 (2004).
77. Tomb, J. F. *et al.* The complete genome sequence of the gastric pathogen Helicobacter pylori. *Nature* **388,** 539–547 (1997).
78. Lechat, P., Hummel, L., Rousseau, S. & Moszer, I. GenoList: an integrated environment for comparative analysis of microbial genomes. *Nucleic Acids Res.* **36,** D469–474 (2008).
79. Cole, S. T. *et al.* Deciphering the biology of Mycobacterium tuberculosis from the complete genome sequence. *Nature* **393,** 537–544 (1998).
80. Dobrovetsky, E. *et al.* High-throughput production of prokaryotic membrane proteins. *J. Struct. Funct. Genomics* **6,** 33–50 (2005).
81. Nelson, K. E. *et al.* Evidence for lateral gene transfer between Archaea and bacteria from genome sequence of Thermotoga maritima. *Nature* **399,** 323–329 (1999).

82. Peden, J. F. Analysis of codon usage. (University of Nottingham, 2000).
83. dos Reis, M., Wernisch, L. & Savva, R. Unexpected correlations between gene expression and codon usage bias from microarray data for the whole Escherichia coli K-12 genome. *Nucleic Acids Res.* **31,** 6976–6985 (2003).
84. Lorenz, R. *et al.* ViennaRNA Package 2.0. *Algorithms Mol. Biol. AMB* **6,** 26 (2011).
85. Coleman, J. R. *et al.* Virus attenuation by genome-scale changes in codon pair bias. *Science* **320,** 1784–1787 (2008).
86. Yang, Z. R., Thomson, R., McNeil, P. & Esnouf, R. M. RONN: the bio-basis function neural network technique applied to the detection of natively disordered regions in proteins. *Bioinformatics* **21,** 3369–3376 (2005).
87. Wimley, W. C., Creamer, T. P. & White, S. H. Solvation energies of amino acid side chains and backbone in a family of host-guest pentapeptides. *Biochemistry (Mosc.)* **35,** 5109–5124 (1996).
88. Engelman, D. M., Steitz, T. A. & Goldman, A. Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins. *Annu. Rev. Biophys. Biophys. Chem.* **15,** 321–353 (1986).
89. Kuhn, M. Building predictive models in R using the caret package. *J Stat Soft* (2008).
90. Weihs, C., Ligges, U., Luebke, K. & Raabe, N. in *Data Analysis and Decision Support* (eds. Baier, D., Decker, R. & Schmidt-Thieme, L.) 335–343 (Springer-Verlag, 2005).
91. Kendall, M. G. A New Measure of Rank Correlation. *Biometrika* **30,** 81 (1938).
92. Joachims, T. Optimizing search engines using clickthrough data. in 133 (ACM Press, 2002). doi:10.1145/775047.775067
93. Robin, X. *et al.* pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics* **12,** 77 (2011).
94. Cunningham, F. *et al.* Ensembl 2015. *Nucleic Acids Res.* **43,** D662–669 (2015).
95. Smedley, D. *et al.* The BioMart community portal: an innovative alternative to large, centralized data repositories. *Nucleic Acids Res.* **43,** W589–598 (2015).
96. Harris, T. W. *et al.* WormBase 2014: new views of curated biology. *Nucleic Acids Res.* **42,** D789–793 (2014).
97. Engel, S. R. *et al.* The reference genome sequence of Saccharomyces cerevisiae: then and now. *G3 Bethesda Md* **4,** 389–398 (2014).
98. De Schutter, K. *et al.* Genome sequence of the recombinant protein production host Pichia pastoris. *Nat. Biotechnol.* **27,** 561–566 (2009).
99. Nordberg, H. *et al.* The genome portal of the Department of Energy Joint Genome Institute: 2014 updates. *Nucleic Acids Res.* **42,** D26–31 (2014).
100. Markowitz, V. M. *et al.* IMG 4 version of the integrated microbial genomes comparative analysis system. *Nucleic Acids Res.* **42,** D560–567 (2014).
101. Uchiyama, I., Mihara, M., Nishide, H. & Chiba, H. MBGD update 2015: microbial genome database for flexible ortholog analysis utilizing a diverse set of genomic data. *Nucleic Acids Res.* **43,** D270–276 (2015).
102. Loader, C. *locfit: Local Regression, Likelihood and Density Estimation.* (2013).
103. Schindelin, J. *et al.* Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9,** 676–682 (2012).
104. Schneider, C. A., Rasband, W. S. & Eliceiri, K. W. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* **9,** 671–675 (2012).
105. Xie, Y. in *Implementing Reproducible Computational Research* (eds. Stodden, V., Leisch, F. & Peng, R. D.) (Chapman and Hall/CRC, 2014).