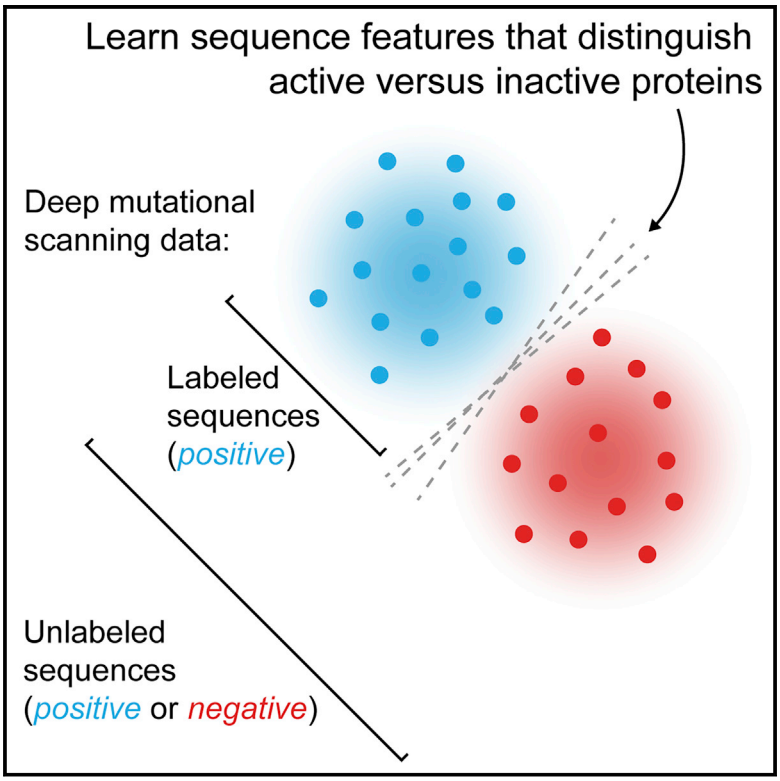
Cell Systems

Report

# Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning

## Graphical Abstract




## Authors

Hyebin Song, Bennett J. Bremer, Emily C. Hinds, Garvesh Raskutti, Philip A. Romero

## Correspondence

promero2@wisc.edu

## In Brief

The quantity of protein sequence-function data is growing rapidly with advances in high-throughput experimentation. Song et al. present a machine learning approach to infer sequence-function relationships from large-scale data generated by deep mutational scanning. The learned models capture important aspects of protein structure and function and can be applied to design new and enhanced proteins.

## Highlights

- Statistical learning framework to infer the mapping from protein sequence to function
- The models learn key aspects of protein structure and function from large-scale data
- The learned models are predictive and can be applied to design new proteins

Song et al., 2021, Cell Systems *12*, 1–10
January 20, 2021 
https://doi.org/10.1016/j.cels.2020.10.007

CellPress

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007

Cell Systems

CellPress

Report

# Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning

Hyebin Song,[1,2] Bennett J. Bremer,[3] Emily C. Hinds,[3] Garvesh Raskutti,[2] and Philip A. Romero[3,4,5,*]
[1]Department of Statistics, The Pennsylvania State University, State College, PA 16802, USA
[2]Department of Statistics, University of Wisconsin–Madison, Madison, WI 53706, USA
[3]Department of Biochemistry, University of Wisconsin–Madison, Madison, WI 53706, USA
[4]Department of Chemical and Biological Engineering, University of Wisconsin–Madison, Madison, WI 53706, USA
[5]Lead Contact
*Correspondence: promero2@wisc.edu
https://doi.org/10.1016/j.cels.2020.10.007

## SUMMARY

Machine learning can infer how protein sequence maps to function without requiring a detailed understanding of the underlying physical or biological mechanisms. It is challenging to apply existing supervised learning frameworks to large-scale experimental data generated by deep mutational scanning (DMS) and related methods. DMS data often contain high-dimensional and correlated sequence variables, experimental sampling error and bias, and the presence of missing data. Notably, most DMS data do not contain examples of negative sequences, making it challenging to directly estimate how sequence affects function. Here, we develop a positive-unlabeled (PU) learning framework to infer sequence-function relationships from large-scale DMS data. Our PU learning method displays excellent predictive performance across ten large-scale sequence-function datasets, representing proteins of different folds, functions, and library types. The estimated parameters pinpoint key residues that dictate protein structure and function. Finally, we apply our statistical sequence-function model to design highly stabilized enzymes.

## INTRODUCTION

A protein's sequence of amino acids encodes its function. This "function" could refer to a protein's natural biological function, or it could also be any other property including binding affinity toward a particular ligand, thermodynamic stability, or catalytic activity. A detailed understanding of how these functions are encoded would allow us to more accurately reconstruct the tree of life, diagnose genetic diseases before they manifest symptoms, and design new proteins with useful properties. The mapping from protein sequence to function is extraordinarily complex because it involves thousands of molecular interactions that are dynamically coupled across multiple length and timescales.

Machine learning can infer how protein sequence encodes function without needing to understand the underlying biophysical mechanisms (Yang et al, 2019; Mazurenko et al, 2020). These learning methods can be broadly categorized into unsupervised and supervised depending on whether the data points are labeled. In the protein context, unsupervised methods learn from examples of sequences that share some common function or property, while supervised methods learn from sequence-function examples. Unsupervised methods are often trained on natural sequence data derived from large genomic databases and effectively learn the rules of folding/function for a given protein family (Morcos et al, 2011; Hopf et al, 2017; Riesselman et al, 2018). In contrast, supervised methods are trained directly on sequence-function examples and, therefore, can learn the mapping to a particular protein property or set of properties. This capability to predict a target protein property is important in protein engineering, which seeks to design and optimize non-natural protein functions. Supervised models have been used to rationally engineer proteinases with improved activity at elevated temperatures, cytochrome P450s with enhanced stability, carbonic anhydrases for industrial carbon capture, and bacteriorhodopsins for optogenetics (Liao et al, 2007; Romero et al, 2013; Alvizo et al, 2014; Bedbrook et al, 2019).

The accuracy and resolution of statistical models improve with increasing data; however, existing supervised methods cannot directly learn from large-scale sequence-function data generated by deep mutational scanning (DMS) and related methods. DMS combines high-throughput screening and next-generation DNA sequencing to experimentally map sequence-function relationships for thousands to millions of protein variants (Fowler and Fields, 2014; Boucher et al, 2014; Weile and Roth, 2018). In principle, DMS data should provide rich sequence-function information for training supervised models. However, learning from DMS data is challenging due to its scale and dimensionality, correlations between sequence variables, sampling error caused by low numbers of observations, and missing/low quality sequence information. In addition, most DMS datasets do not contain negative sequence examples because these sequences are difficult or impossible to obtain using high-throughput

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007



screening/selection methods. These negative sequences are important to directly infer how sequence maps to function. Hence, DMS data are neither amenable to supervised learning due to the lack of negative sequences nor unsupervised learning since many sequences have positive labels.

In this paper, we present a supervised learning framework for inferring sequence-function relationships from large-scale data generated by DMS. We categorize DMS data as positive-unlabeled (PU) data because they contain examples of both positive sequences and sequences without labels. Learning from PU data has applications in domains such as text mining, gene identification, marketing, and ecological modeling (Liu et al, 2003; Mordelet and Vert, 2011; Yi et al, 2017; Ward et al, 2009). We develop a PU learning method that models a protein's function as an unobserved latent variable and then infers how sequence maps to this latent function by maximizing an observed likelihood function. Our learned PU models displayed excellent predictive ability and stability across ten diverse DMS datasets. The PU model's parameters describe how amino acid substitutions affect protein function, and the significance of these parameter estimates can be evaluated using statistical hypothesis testing. We demonstrate the extrapolative power of the learned sequence-function mapping by designing enzymes with increased thermostability.

## RESULTS

### A Statistical Framework for Learning Sequence-Function Relationships

Supervised learning methods can infer how sequence maps to function from a set of experimental sequence-function examples. However, it is challenging to apply existing learning methods to large-scale data generated by DMS due to the lack of negative sequence examples. We use the term DMS to broadly refer to any experiment that maps sequence-function relationships using a combination of gene mutagenesis, high-throughput screening/selections, and next-generation DNA sequencing. An overview of a standard DMS experiment is illustrated in Figure 1A. This section describes the DMS data generation process, introduces key statistical variables, and proposes a PU approach to learn from DMS data.

A protein's biochemical activity $A_i$ is a function of its amino acid sequence, i.e., $A_i = f(x_i)$, where $x_i$ is a vector that specifies a protein's amino acid sequence (see STAR Methods). Now suppose a protein sequence can be categorized as active or inactive depending on whether its activity $A_i$ falls above/below a defined activity threshold $t$. There is also some error in experimentally determining whether a sequence is active or inactive. We define a protein's experimentally measured functional response as:

$$Y_i = \begin{cases} 1 & \text{if } A_i + \epsilon_i > t \\ 0 & \text{if } A_i + \epsilon_i \leq t \end{cases}, \qquad \text{(Equation 1)}$$

where $\epsilon_i$ is the random error associated with experimentally characterizing the $i$th sequence. We say a sequence is "positive" if its experimentally measured activity exceeds the threshold $t$ (i.e., $Y_i = 1$) and "negative" otherwise. Note that we make the subtle distinction between the terms active/inactive and positive/negative: we use active/inactive to describe the true functional state of a protein and positive/negative to indicate the result of an experimental measurement.

A DMS experiment starts with an initial library of sequences that each maps to a particular activity value and can be categorized as active or inactive. We refer to this initial library as "unlabeled" because it contains an unknown mixture of active and inactive sequences. A high-throughput screen/selection then samples this initial unlabeled library to obtain examples of positive sequences. Notably, it is often difficult or impossible to isolate negative sequences because most experimental methods are designed to identify positive sequences (e.g., growth selections). We refer to the sampled positive sequences as "labeled" because they are known to have positive labels. The sequences within the initial unlabeled set and positive labeled set are then determined using next-generation sequencing. The final data contain $n_u$ sequences sampled from the unlabeled set and $n_p$ sequences sampled from the positive labeled set (Figure 1B).

We aim to learn from DMS data to understand how amino acid sequence maps to function (i.e., infer $f$). If the data consisted of $(X_i, Y_i)$ pairs, we could simply train a binary classifier such as a logistic regression model or a multilayer perceptron. However, DMS experiments do not reveal the true functional response ($Y$) but instead only provide examples of positive sequences. The lack of negative sequence examples results in a mis-specified binary classification problem and makes it challenging to directly infer how sequence maps to function.

We propose a PU approach to learn from DMS data. PU learning estimates how input variables affect the positive-negative response from positive and unlabeled data (Liu et al, 2003; Elkan and Noto, 2008; Song and Raskutti, 2019). We introduce a binary variable $Z$ that specifies whether a sequence is labeled ($Z = 1$) or unlabeled ($Z = 0$). DMS experiments effectively generate $(X_i, Z_i)$ pairs with an unobserved functional response $Y_i$ (Figure 1C). From the setup of the problem, $Z$ and $Y$ are closely related. In particular, all labeled sequences are positive, and the proportion of positive sequences in the unlabeled set is the same as that of the initial library. In other words,

$$\mathbb{P}(Y = 1 | Z = z) = \begin{cases} 1 & \text{if } z = 1 \\ \pi & \text{if } z = 0 \end{cases}.$$

where $\pi$ is the proportion of positive sequences in the initial library. We aim to infer $f$ using the observed examples $(x_i, z_i)_{i=1}^n$, where $n = n_p + n_u$ denotes the total number of sequence examples.

### Algorithms for Large-Scale Positive-Unlabeled Learning

Sequence-function data obtained by DMS typically contain sequence examples from the initial (unlabeled) library and positive sequences (Figure 1). We develop algorithms for learning the sequence-function mapping from this PU sequence data. Our approach utilizes the distributional relationship between X, Y, and Z to infer the latent functional response $y_i$ from the observed labels $(z_i)_{i=1}^n$ and amino acid sequences $(x_i)_{i=1}^n$ and then infer the $f$ that best describes the latent responses $(y_i)_{i=1}^n$.

We model $f$ as a linear function of amino acid sequence $f(x) = \sum_{j,aa} x_{j,aa} a_{j,aa}$ and use a logistic function to describe the probability that a sequence is positive:

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007



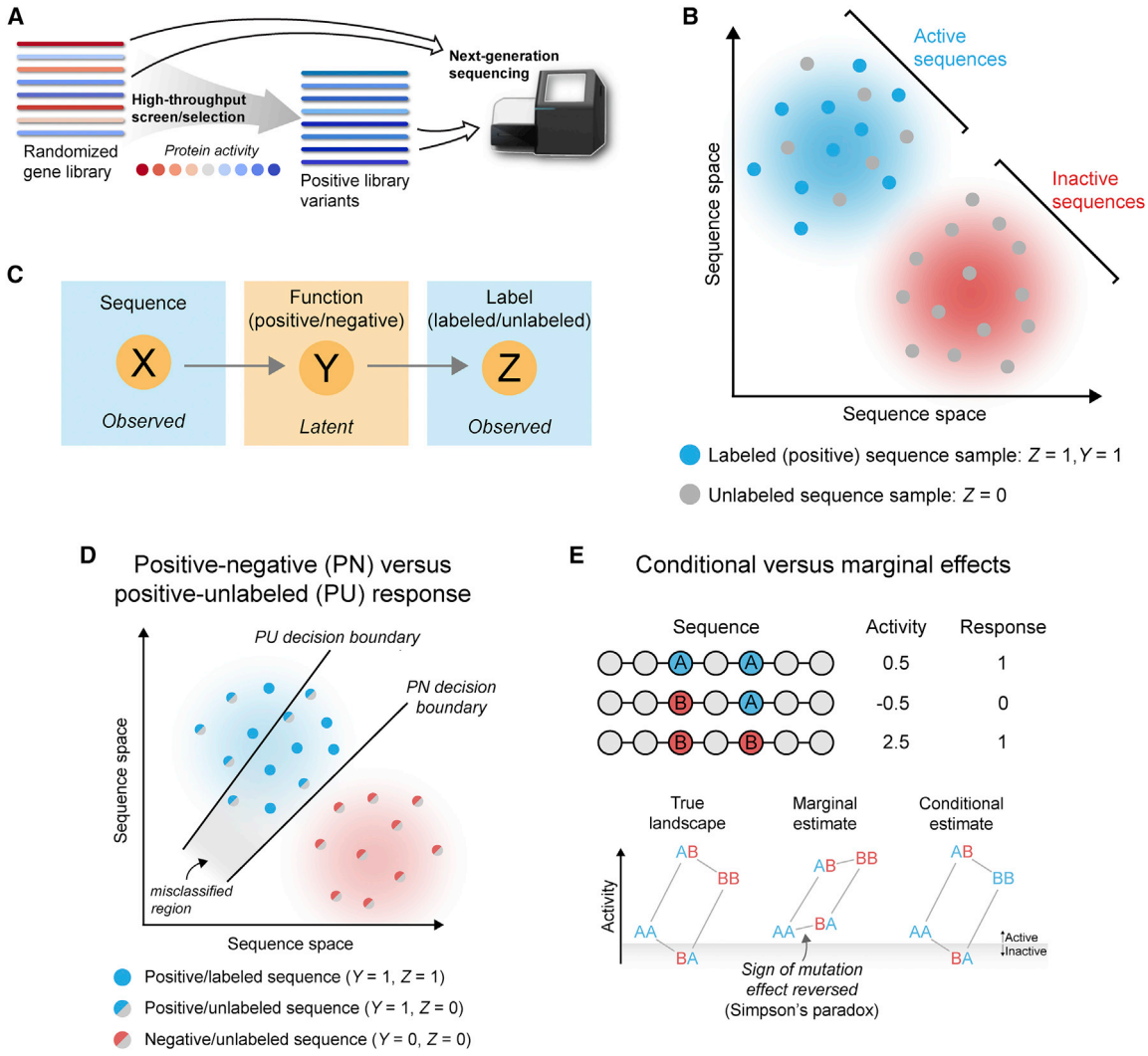



**Figure 1. Positive-Unlabeled Learning from DMS Data**

(A) Overview of a typical DMS experiment. DMS experiments start with a large library of gene variants that display a range of activities. The gene library is then expressed and passed through a high-throughput screen or selection that isolates the positive variants. The activity threshold to be categorized as positive will depend on the details of the particular high-throughput screen/selection. It is often difficult or impossible to experimentally isolate negative sequences. Genes from the initial library and the isolated positive variants are then extracted and analyzed using next-generation DNA sequencing. DMS experiments generate thousands to millions of sequence examples from both the initial and positive sets.

(B) DMS experiments sample sequences from protein sequence space. The resulting data contain positive labeled sequence examples ($Y = 1, Z = 1$) and unlabeled sequence examples ($Z = 0$) that contain a mixture of active and inactive sequences.

(C) The relationships between variables representing protein sequences ($X$), latent function ($Y$), and the observed labels ($Z$). The functional response $Y$ is not directly observed in DMS experiments and must be inferred from $X$ and $Z$.

(D) PU learning models the true PN response, while enrichment-based estimates capture the PU response. Modeling the PU response gives rise to a decision boundary that is shifted toward the positive class, resulting in positive sequences that are misclassified as negative.

(E) PU learning estimates the conditional effect of a mutation, while site-wise enrichment estimates the marginal effect. Marginal estimates are biased and in extreme cases can result in a sign reversal phenomenon known as Simpson's paradox. In the example, we consider amino acid substitutions A→B at two independent sites in a protein. If we observe sequences AA, BA, and BB, the marginal estimate will reverse the sign of substitution A→B at the first position. The marginal model will also misclassify sequence BA as positive, even though it was observed to be negative. In contrast, the conditional estimate correctly models the true protein function landscape.

$$\mathbb{P}_\theta(Y = 1|X = x) = \frac{e^{x^\top \theta}}{1 + e^{x^\top \theta}}, \qquad \text{(Equation 2)}$$

where $x^\top \theta = \sum_{j,aa} x_{j,aa} a_{j,aa} - t$ represents a relative activity level of amino acid sequence $x$ with respect to the activity threshold $t$ in (1), and $\theta$ parameterizes the effect of making an amino acid substitution from a defined reference sequence (see STAR Methods). The model's parameters ($\theta$) are closely related to the site-wise enrichment scores that are commonly used to analyze DMS data (Bloom, 2015; Klesmith and Hackel, 2019; Wrenbeck et al., 2017; Abriata et al, 2016). However, site-wise enrichment

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007



is a biased estimator for $\theta$ because it is derived from PU data and makes strong assumptions about the independence between sequence positions. We derive the mathematical relationship between site-wise enrichment and a mutation's true effect ($\theta$) in the STAR Methods.

We take a likelihood-based approach for estimating $\theta$ from the observed examples $(x_i, z_i)_{i=1}^n$. To account for the fact that the true responses $(y_i)_{i=1}^n$ are latent, we use an observed likelihood that is a product of the marginalized probabilities (Ward et al, 2009)

$$\ell(\theta) = \prod_{i=1}^{n}\left(\sum_{y_i\in(0,1)} \mathbb{P}(Y_i = y_i, Z_i = z_i | X_i = x_i, S_i = 1)\right)$$

$$= \prod_{i=1}^{n}\left(\frac{n_p}{n_p+\pi n_u}\frac{e^{x_i^\top\theta+b}}{1+e^{x_i^\top\theta+b}}\right)^{z_i}\left(1-\frac{n_p}{n_p+\pi n_u}\frac{e^{x_i^\top\theta+b}}{1+e^{x_i^\top\theta+b}}\right)^{1-z_i}$$

(Equation 3)

for $b := \log\frac{n_p+\pi n_u}{\pi n_u}$, where $S_i\in(0,1)$ is an indicator variable representing whether the $i$th example is present in the data. We use the maximum likelihood approach to estimate $\theta$. In particular, we minimize the *negative* observed log-likelihood and define the estimated coefficients $\widehat{\theta}$ as

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmin}}\{-\log\ell(\theta)\} \quad \text{(Equation 4)}$$

The negative observed log-likelihood, $-\log\ell(\theta)$, is a non-convex function of $\theta$. Obtaining a global minimizer of a non-convex function is in general a challenging problem, so the feasibility of obtaining $\widehat{\theta}$ is not immediate. We previously found similar classes of problems can be solved when the likelihood function is calculated with sufficiently large sample size $n$ (Song et al., 2020). In these cases, any stationary point is the global minimizer with high probability. Since our sequence-function datasets typically contain millions of observations, we can find the maximum likelihood estimate by identifying a stationary point of the negative-observed log-likelihood.

We solved this optimization problem using the Majorization-Minimization (MM) algorithm to obtain a stationary point of the negative observed log-likelihood function (Equation 3). Note that the likelihood (Equation 3) involves the hyperparameter $\pi$, the proportion of positive sequences in the unlabeled set. $\pi$ was experimentally determined for some datasets, and we used this value for the hyperparameter if it was available. Otherwise, we carried out a grid search over $\pi$ values and chose the $\pi$ value that maximized the area under the receiver operating characteristic curve. We used the learned model parameters to calculate p values to test whether amino acid substitutions have a significant impact on protein function. These p values were adjusted using the Benjamini-Hochberg (BH) procedure to account for multiple hypothesis testing (STAR Methods). An overview of our data processing, parameter estimation, and model analysis workflow is provided in Figure S2.

## Relationship between Learned PU Model Parameters and Site-Wise Enrichment Scores

There is a close connection between our PU model's parameters and the site-wise enrichment scores that are commonly used to analyze DMS data (Bloom, 2015; Klesmith and Hackel, 2019; Wrenbeck et al, 2017; Abriata et al., 2016). Both quantities evaluate how amino acid substitutions affect a protein's functional response (i.e., estimate $a'_{j,aa}$). However, our PU model provides a consistent estimate of $a'_{j,aa}$ because it directly models the true positive-negative (PN) response and considers the conditional effects of amino acid substitutions. In this section, we define site-wise enrichment scores, contrast them with the PU model parameters, and identify two different sources of bias in their estimate of $a'_{j,aa}$.

Site-wise enrichment scores are calculated using marginal amino acid frequencies:

$$E_{j,aa} = \log\left(\frac{\sum_{i\in P} x_{i,j,aa}/n_p}{\sum_{i\in U} x_{i,j,aa}/n_u}\right) - \log\left(\frac{\sum_{i\in P} x_{i,j,\mathrm{ref}_j}/n_p}{\sum_{i\in U} x_{i,j,\mathrm{ref}_j}/n_u}\right) \quad \text{(Equation 5)}$$

where $P$ and $U$ are the positive and unlabeled sets of sequences, respectively. $E_{j,aa}$ compares the prevalence of amino acid *aa* with the reference sequence (typically wild type) in the positive and unlabeled sets. A residue with a negative enrichment score is underrepresented in the positive set and therefore associated with decreased protein activity. Conversely, a residue with a positive enrichment score is associated with the increased protein activity. These enrichment scores provide a simple and convenient method to compare frequencies before/after selection and estimate the effects of amino acid substitutions.

Enrichment and our PU learning method capture different response variables related to protein function (Figure 1D). Enrichment models the PU response, and since all labeled sequences are positive, this is equivalent to modeling a sequence's label $Z$. In contrast, our PU learning method directly models the PN response by inferring the latent function $Y$ from observed labels $Z$. The presence of positive sequences in the unlabeled set causes enrichment-based methods to provide attenuated estimates of an amino acid substitution's effect (see STAR Methods). This leads to a decision boundary that is shifted toward the positive class and results in misclassified sequences. Our PU learning method models the true PN response and thus provides an unbiased estimate of a substitution's effect $a'_{j,aa}$.

The second key difference between PU model parameters and enrichment is related to marginal versus conditional effects. Our PU learning method estimates the effect of an amino acid substitution from a defined sequence background, typically wild type. This is a *conditional* estimate because the effect of the mutation is conditioned on all other sites in the protein. This conditional effect provides an unbiased estimate of an amino acid substitution's effect $a'_{j,aa}$. In contrast, site-wise enrichment estimates the effect of an amino acid substitution in combination with averaged effects from all other sites in the dataset. These *marginal* estimates include the true effect $a'_{j,aa}$, in addition to indirect effects from other sequence positions. These indirect effects lead to bias in the estimate of $a'_{j,aa}$ (see STAR Methods). In more extreme instances, marginal estimates can reverse the sign of an effect and lead to incorrect conclusions about whether a substitution is beneficial or deleterious (Figure 1E). This effect reversal is referred to as Simpson's paradox.

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007



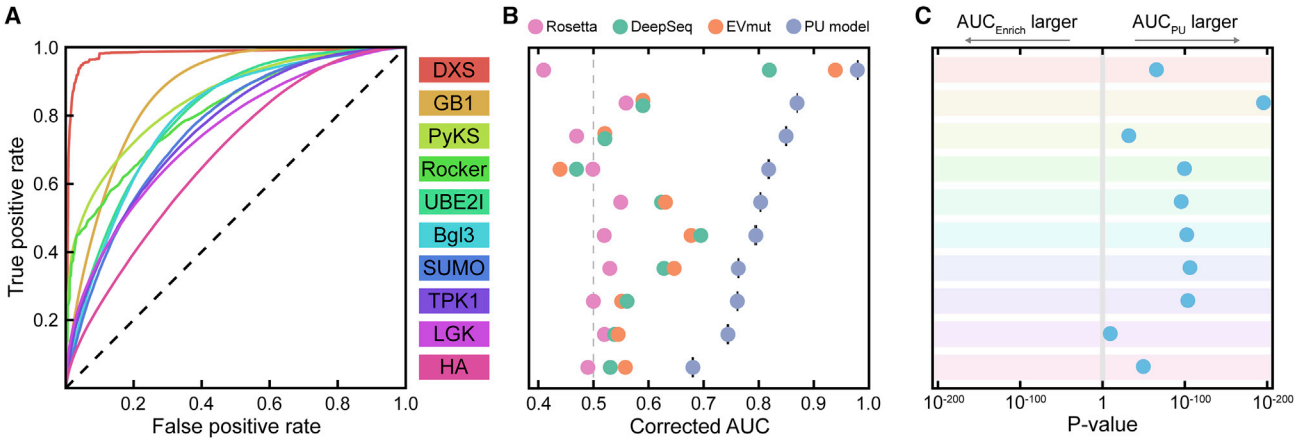



**Figure 2. Performance of the PU Learning Method across Protein Datasets**

(A) ROC curves for the ten tested protein datasets. ROC curves were generated using a 10-fold cross-validation and were corrected to account for PU data (see STAR Methods and Figure S1).

(B) The PU model's corrected ROC-AUC values range from 0.68 to 0.98 and outperform structure-based (Rosetta) and unsupervised learning methods (EVmutation and DeepSequence). Error bars for the PU model predictions were calculated by taking the standard deviation of AUCs across ten cross-validation folds.

(C) A statistical comparison between PU model predictions and site-wise enrichment. The PU model outperformed enrichment on all ten tested datasets, with $p < 10^{-9}$. p values were calculated using a corrected repeated cross-validation test.

## Learning from Large-Scale Sequence-Function Data

We applied our PU learning method to infer the sequence-activity mapping from ten large sequence-function datasets (see Table S1). These datasets represent proteins of diverse folds/functions, were generated using different library mutagenesis methods, span several orders of magnitude in size, and have varying levels of missing sequence information. The PU models displayed excellent predictive ability on all ten datasets, with cross-validated area under the receiver operating characteristic curve (ROC-AUC) ranging from 0.68 to 0.98 (Figures 2A and 2B). For comparison, we also evaluated predictions from structure-based (Alford et al, 2017) and unsupervised learning methods (Hopf et al, 2017; Riesselman et al, 2018). Rosetta, EVmutation, and DeepSequence all displayed substantially lower AUC values than the PU model (Figures 2B and S3A).

Our PU learning method estimates how mutations affect a protein's functional response. This PU estimate is closely related to the site-wise enrichment scores that are commonly used to analyze DMS data. We compared the predictive ability of the PU model versus enrichment using a corrected cross-validation test. We found the PU model predictions were better than enrichment for all ten datasets, with $p<10^{-9}$ (Figure 2C). However, the PU models' AUCs were only marginally higher than enrichment, with AUC differences ranging from 0.002 to 0.017 (Figure S3B).

We evaluated the robustness of the learned PU models to data sampling and the hyperparameter $\pi$. We analyzed the stability of each model's parameter estimates by calculating the coefficient of variation (CV) across different cross-validation folds for all significant parameters (i.e., BH-adjusted $p<0.05$). The parameters displayed average absolute CVs ranging from 0.01 to 0.08 (Figure 3C), indicating the estimates were highly insensitive to different training sets. We also evaluated the feature selection stability by computing the average fraction of commonly selected features across different cross-validation folds (STAR Methods). We found the selected features were nearly identical across cross-validation folds for each dataset (Figure S3D). Finally, we tested how the choice of the hyperparameter $\pi$ (experimentally determined or estimated) affects the learned PU models. We found the parameter values estimated using our chosen $\pi$ value were highly correlated with parameters estimated across the entire range of $\pi$ values tested (Figure S3E).

## Learned Parameters Relate to Protein Structure and Function

The B1 domain of protein G (GB1) is a small 8-kDa alpha-beta roll that binds to IgG. We performed further analyses relating the learned GB1 model with protein G structure and function. The PU model's coefficients describe how an amino acid substitution (mutation) affects the protein's functional response (Equation 7). A negative coefficient indicates that a substitution decreases protein activity, whereas a substitution with a positive coefficient increases activity. We found that most amino acid substitutions in GB1 are slightly deleterious, while a smaller subset is highly deleterious (Figure 3A). Each position in the amino acid sequence displayed a range of mutational effects (Figure 3B). Substitutions to proline are the most deleterious on average (presumably because they disrupt protein structure), followed by substitutions to the acidic amino acids. We found a site's average mutational effect is highly dependent on its location in the three-dimensional structure. Sites with large average mutational effects tend to be located in either the protein core or the IgG-binding interface (Figures 3B and 3C).

GB1 residues E27, K31, and W43 have the most negative average mutational effect, suggesting that many substitutions at these positions are highly deleterious. Consistent with the model results, these three residues are known to form key hydrogen bonds and salt bridges with the IgG ligand and make the largest contributions to the free energy of binding (Sauer-Eriksson et al, 1995; Sloan and Hellinga, 1999). Residues A24 and D40 have the largest positive average mutational effect, with many

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007



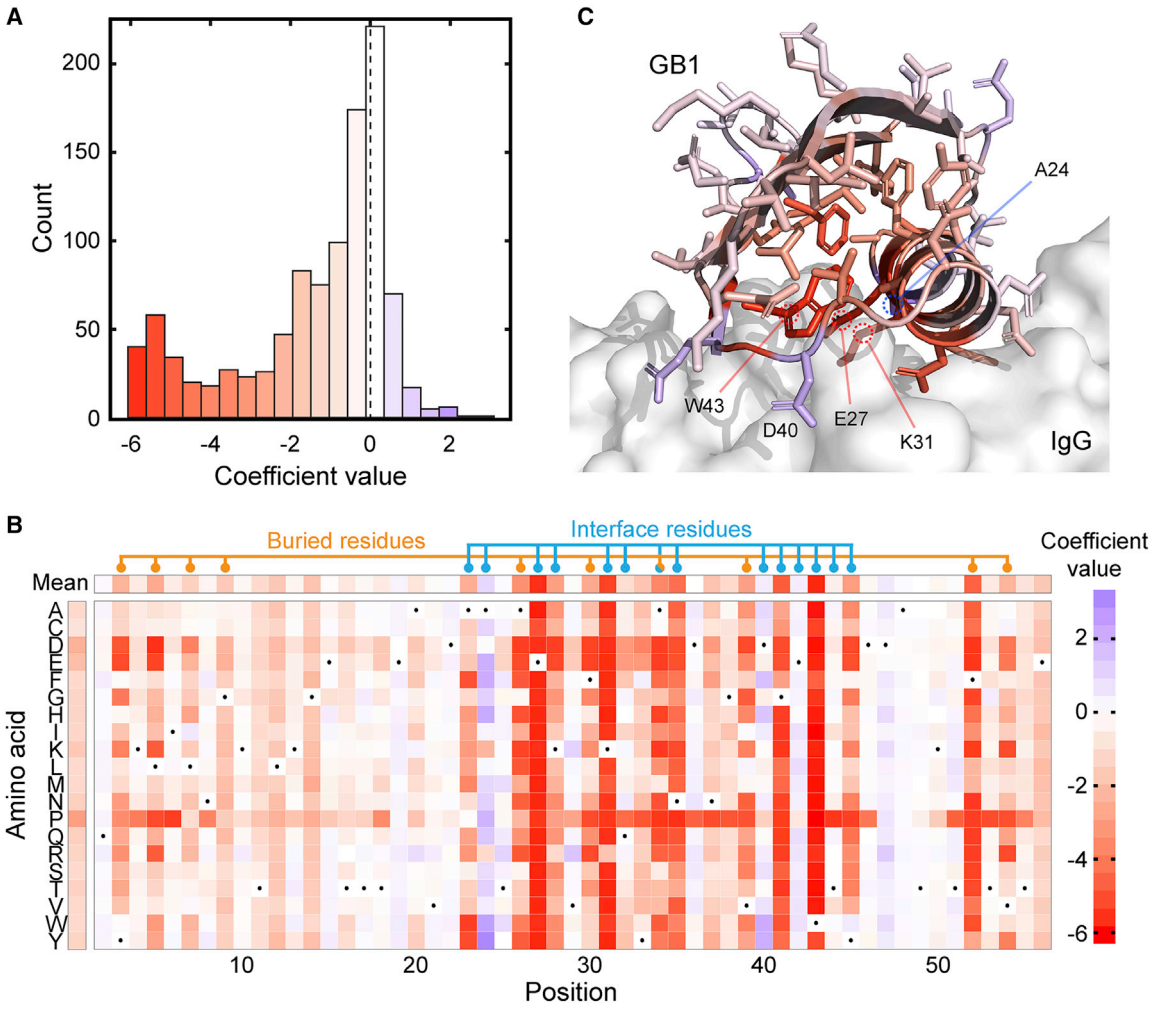



**Figure 3. Model Parameters Relate to GB1 Structure and Function**

(A) The distribution of model coefficients. Most coefficients have a relatively small magnitude, while a substantial fraction of coefficients have a large negative effect.

(B) A heatmap of the GB1 model coefficients. The wild-type amino acid is depicted with a black dot. Buried and interface residues tend to have larger magnitude coefficients, indicating their important role in GB1 function. Buried and interface residues were determined from the protein G crystal structure (PDB: 1FCC). Buried residues were defined as having a relative solvent accessibility less than 0.1. Interface residues were defined as having a heavy atom within 4 Å of IgG.

(C) The site-wise average model coefficients mapped onto the protein G crystal structure (PDB: 1FCC). The IgG-binding partner is depicted as a gray surface. Residues in the protein core and binding interface tend to have the largest average coefficients.

substitutions that are predicted to increase GB1 activity. Both of these sites are located in the IgG-binding interface. Previous studies have identified residue position 24 to play a key role in IgG binding, and substitutions from A24 can increase binding affinity through improved ionic interactions (Sauer-Eriksson et al, 1995). Furthermore, computationally designed high-affinity protein G variants have substitutions at position 24 (Jha et al, 2014). The model parameters suggest that residue D40 prefers substitution to aromatic amino acids (Figure 3B). Inspection of the crystal structure suggests these mutations could form potential interactions (pi-pi, cation-pi) with a nearby histidine in IgG (Figure S4).

## Statistics-Based Protein Design

Our PU learning method provides a quantitative description of the sequence-function mapping. The model also captures statistical uncertainties arising from undersampling and correlated sequence variables. Here, we develop a protein design framework that leverages this statistical sequence-function information.

We trained a PU model on a Bgl3 deep mutational scan that had been performed at an elevated temperature (Bgl3-HT, Table S1) (Romero et al, 2015). Under these experimental conditions, the positive class corresponds to Bgl3 sequences with a high thermal tolerance, and therefore the model should learn how amino acid substitutions affect thermostability. The learned PU model displayed excellent predictive ability (corrected AUC of 0.72, Figure S5A).

We applied the PU model to design Bgl3 variants based on either coefficient magnitudes or p values (Figure 4A). The coefficient-based design (Bgl.cf) contained ten amino acid substitutions corresponding to the ten largest positive coefficients. The

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007



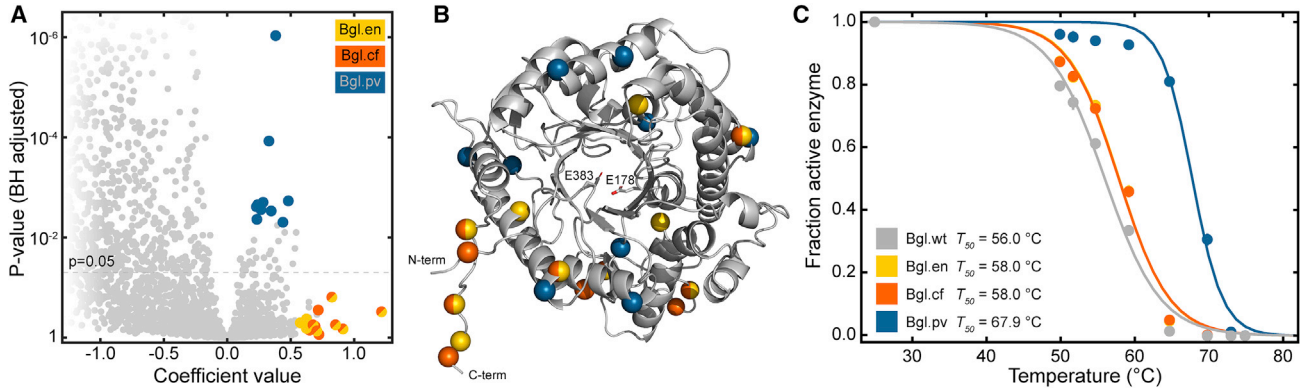



**Figure 4. Applying the PU Model to Design Enhanced Proteins**

(A) A plot of model coefficients versus p values. Sequences were designed to combine ten mutations with the largest coefficient values, smallest p values, or largest enrichment scores.

(B) The positions chosen by the three design methods are mapped onto the Bgl3 protein structure. The structure is based on the Bgl3 crystal structure (PDB: 1GNX) and missing termini/loops were built in using MODELLER (Sali and Blundell, 1993).

(C) Thermostability curves for wild-type Bgl3 and the three designed proteins. $T_{50}$ values were estimated by fitting a sigmoid function to the fraction of active enzyme. Note the curve for Bgl.en is shown in yellow and falls directly behind the orange Bgl.cf curve.

p-value-based design (Bgl.pv) contained ten substitutions corresponding to the ten positive coefficients with the smallest p values. We also designed a sequence that contained the ten substitutions with the largest enrichment scores (Bgl.en). The Bgl.cf and Bgl.en designs contained six common substitutions, while the substitutions in Bgl.pv were distinct from the other two (Figure S5B). The substitutions within these three designs are generally distributed throughout the protein structure (Figure 4B); however, there appears to be some bias for the coefficient/enrichment designs to choose substitutions in the termini.

We experimentally characterized the thermostability of wild-type Bgl3 and the three designed enzymes. All three designed sequences were stabilized relative to wild-type Bgl3 (Figure 4C). The coefficient- and enrichment-based designs displayed modest stability increases ($\sim 2^{\circ}$C), while the p-value-based design was almost 12°C more stable than wild-type Bgl3.

## DISCUSSION

We have presented a supervised learning framework to infer the mapping from protein sequence to function from large-scale sequence-function data. We applied a PU learning approach to address the lack of negative sequence examples typically encountered in DMS data. Our PU learning method models a protein's true functional response as an unobserved latent variable and then estimates how sequence maps to this latent response by maximizing the observed likelihood. Our approach leverages established statistical methods and hypothesis testing to evaluate the significance of sequence features and predictions. The PU models displayed excellent predictive ability and robustness across ten diverse protein datasets. The learned model parameters capture important aspects of protein structure and function and can be used to design new and enhanced proteins.

We compared the PU model's predictive ability to established structure-based and unsupervised learning methods including Rosetta, EVmutation, and DeepSequence. This is a rather unequal comparison because the PU model is trained directly on the DMS data, whereas the other methods are trained on peripherally related sequence/structure data. As expected, the PU model displayed substantially higher predictive performance than structure-based or unsupervised methods. These other methods have the distinct advantage that they can make reasonable predictions in the absence of DMS data, while our PU model requires DMS data for training. The relative performance of these various predictive methods is likely dependent on the particular protein activity that is being modeling. We expect Rosetta to capture protein activities related to folding and stability; while EVmutation and DeepSequence may capture preservation of native function and the associated biophysical properties. Along these lines, supervised methods that learn the mapping to a particular property are required to model and predict non-natural protein properties. The ability to predict non-natural properties is essential for designing new proteins with behaviors beyond naturally evolved biological function.

There is a close connection between our PU model's parameters and the enrichment scores commonly used to evaluate DMS experiments. Both methods estimate how amino acid substitutions affect a protein's functional response. However, there are two key differences in these estimates: (1) enrichment-based methods estimate the PU response, whereas our method directly estimates the PN response, and (2) enrichment-based methods estimate marginal amino acid effects and therefore make strong assumptions about the independence between sequence positions. In theory, the parameters estimated using our PU learning method should provide a more accurate and less biased estimate of how amino acid substitutions affect function. We found the PU model had greater predictive ability than enrichment on all ten protein datasets tested. While the differences in predictive performance were small (AUC differences <0.02), these differences were statistically significant ($p<10^{-9}$) in all cases. These results suggest that the learned PU model

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007



is better overall, but the predictions may not be much different from enrichment. The greatest advantage of the PU model over enrichment is the ability to perform statistical hypothesis testing to evaluate the significance of the model parameters. Hypothesis testing provides confidence in the parameter estimates and predictions and is thus essential for protein design.

We applied the learned PU models to design beta-glucosidases with improved thermal tolerance. We compared design strategies based on enrichment, PU model coefficients, and p values. We found that enrichment- and coefficient-based methods chose similar substitutions, and the resulting designs had modest increases in thermostability. In contrast, the p-value-based design contained a distinct set of substitutions and was significantly stabilized relative to the wild-type parent sequence. These results suggest that it is better to design sequences containing high-confidence substitutions rather than including uncertain substitutions with the largest magnitudes. In principle, this protein engineering strategy could be implemented iteratively, where a DMS dataset is generated from an initial parent sequence, these data are used to design an improved sequence, then a new DMS dataset is generated around this improved sequence, and the process is repeated. This iterative sequence optimization approach is similar to directed evolution; however, it fully leverages sequence-function information at each generation, allowing it to take larger jumps in protein sequence space.

There are several interesting extensions of the PU learning framework presented here. In this work we only considered a linear sequence-function mapping. In theory, our modeling framework could be extended to include pairwise or even higher order interactions between residues. These models would account for epistatic interactions between sites and could possibly be used to determine contacting residues in the protein's three-dimensional structure. We performed preliminary tests to evaluate whether we could model interactions in DMS data and found the massive increase in system variables made the computations intractable in most cases. Future work could explore more efficient algorithms for learning from high-dimensional interaction models. Another interesting area to explore is multi-response models that consider several protein properties simultaneously. For example, we could model the Bgl3 room-temperature and high-temperature datasets simultaneously to directly resolve the residues responsible for protein stability. Finally, our PU modeling approach used a point estimate for the hyperparameter $\pi$. A more integrated modeling framework could account for uncertainty in $\pi$ estimates and how this propagates to model coefficients and p values.

We applied our PU learning framework to model protein sequence-function relationships. In principle, similar approaches could be used to model genotype-phenotype mappings across any level of biological organization. PU data arise whenever a population of genetic variants (generated via mutagenesis, crossbreeding, etc.) is passed through a phenotypic screen/selection, and the genotypes from the before/after populations are determined using high-throughput DNA sequencing. This general format has been used to experimentally map genotype-phenotype relationships for promoters/regulatory sequences (Kosuri et al., 2013; Holmqvist et al., 2013), metabolic pathways (Ghosh and Landick, 2016), microbial and mammalian genomes (Ehrenreich et al., 2010; Robins et al, 2013; Price et al, 2018; Findlay et al., 2018), and microbial communities (Kehe et al, 2019; Hsu et al., 2019).

A quantitative understanding of the mapping between protein sequence and function is important for describing natural evolution, diagnosing and treating human disease, and designing customized proteins. Advances in experimental technology have enabled researchers to map sequence-function relationships on an unprecedented scale and resolution. The resulting data are challenging to analyze because they are typically massive, high-dimensional, contain missing sequence information, and lack negative sequence examples. Our PU learning framework provides a principled way of analyzing large-scale sequence-function data to yield biochemical insights and make quantitative predictions.

## STAR★METHODS

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

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead Contact
  - Materials Availability
  - Data and Code Availability
- METHOD DETAILS
  - Linear Sequence-Function Model
  - Site-Wise Enrichment Captures Marginal Effects, Which Are Biased
  - Enrichment-Based Methods Provide Attenuated Estimates of a Mutation's Effect due to Latent Positive Sequences
  - Data Preprocessing
  - PU Model Training
  - Aggregating Models from Multiple Replicates
  - Evaluating and Comparing Model Predictive Ability
  - Predictions Using Rosetta, EVmutation, and DeepSequence
  - Statistical Hypothesis Testing
  - Evaluating PU Model Stability
  - Beta-Glucosidase Cloning, Expression, and Characterization
- QUANTIFICATION AND STATISTICAL ANALYSIS

## SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.cels.2020.10.007.

## ACKNOWLEDGMENTS

We would like to acknowledge funding support from the NIH grants R35 GM119854 and R01 GM131381 and the NSF grant DMS-1811767.

## AUTHOR CONTRIBUTIONS

H.S., G.R., and P.A.R. conceived the project. H.S. and G.R. developed the PU learning methods and code. H.S., E.C.H., and P.A.R. analyzed the data. B.J.B. performed the Rosetta, EVmutation, and DeepSequence analysis. B.J.B.

performed all experimental work and analysis. H.S. and P.A.R. wrote the paper with feedback from all other authors.

## DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: November 2, 2019
Revised: August 13, 2020
Accepted: October 22, 2020
Published: November 18, 2020

## REFERENCES

Abriata, L.A., Bovigny, C., and Dal Peraro, M. (2016). Detection and sequence/structure mapping of biophysical constraints to protein variation in saturated mutational libraries and protein sequence alignments with a dedicated server. BMC Bioinformatics *17*, 242.

Alford, R.F., Leaver-Fay, A., Jeliazkov, J.R., O'Meara, M.J., DiMaio, F.P., Park, H., Shapovalov, M.V., Renfrew, P.D., Mulligan, V.K., Kappel, K., et al. (2017). The Rosetta all-atom energy function for macromolecular modeling and design. J. Chem. Theor. Comput. *13*, 3031–3048.

Alvizo, O., Nguyen, L.J., Savile, C.K., Bresson, J.A., Lakhapatri, S.L., Solis, E.O.P., Fox, R.J., Broering, J.M., Benoit, M.R., Zimmerman, S.A., et al. (2014). Directed evolution of an ultrastable carbonic anhydrase for highly efficient carbon capture from flue gas. Proc. Natl. Acad. Sci. USA *111*, 16436–16441.

Bedbrook, C.N., Yang, K.K., Robinson, J.E., Mackey, E.D., Gradinaru, V., and Arnold, F.H. (2019). Machine learning-guided channelrhodopsin engineering enables minimally invasive optogenetics. Nat. Methods *16*, 1176–1184.

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Stat. Soc. B *57*, 289–300.

Bloom, J.D. (2015). Software for the analysis and visualization of deep mutational scanning data. BMC Bioinformatics *16*, 168.

Boël, G., Letso, R., Neely, H., Price, W.N., Wong, K.H., Su, M., Luff, J.D., Valecha, M., Everett, J.K., Acton, T.B., et al. (2016). Codon influence on protein expression in E. coli correlates with mRNA levels. Nature *529*, 358–363.

Boucher, J.I., Cote, P., Flynn, J., Jiang, L., Laban, A., Mishra, P., Roscoe, B.P., and Bolon, D.N. (2014). Viewing protein fitness landscapes through a next-gen lens. Genetics *198*, 461–471.

Bouckaert, R.R., and Frank, E. (2004). Evaluating the replicability of significance tests for comparing learning algorithms. In Advances in Knowledge Discovery and Data Mining (Springer), pp. 3–12.

Carpenter, J., and Kenward, M. (2013). Multiple Imputation and Its Application (John Wiley & Sons).

Dietterich, T.G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. *10*, 1895–1923.

Doud, M.B., and Bloom, J.D. (2016). Accurate measurement of the effects of all amino-acid mutations on influenza hemagglutinin. Viruses *8*, 155.

Ehrenreich, I.M., Torabi, N., Jia, Y., Kent, J., Martis, S., Shapiro, J.A., Gresham, D., Caudy, A.A., and Kruglyak, L. (2010). Dissection of genetically complex traits with extremely large pools of yeast segregants. Nature *464*, 1039–1042.

Elkan, C., and Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08), pp. 213–220.

Findlay, G.M., Daza, R.M., Martin, B., Zhang, M.D., Leith, A.P., Gasperini, M., Janizek, J.D., Huang, X., Starita, L.M., and Shendure, J. (2018). Accurate classification of BRCA1 variants with saturation genome editing. Nature *562*, 217–222.

Fowler, D.M., and Fields, S. (2014). Deep mutational scanning: a new style of protein science. Nat. Methods *11*, 801–807.

Ghosh, I.N., and Landick, R. (2016). OptSSeq: high-throughput sequencing readout of growth enrichment defines optimal gene expression elements for Homoethanologenesis. ACS Synth. Biol. *5*, 1519–1534.

Holmqvist, E., Reimegård, J., and Wagner, E.G.H. (2013). Massive functional mapping of a 5'-UTR by saturation mutagenesis, phenotypic sorting and deep sequencing. Nucleic Acids Res. *41*, e122.

Hopf, T.A., Ingraham, J.B., Poelwijk, F.J., Schärfe, C.P.I., Springer, M., Sander, C., and Marks, D.S. (2017). Mutation effects predicted from sequence co-variation. Nat. Biotechnol. *35*, 128–135.

Hsu, R.H., Clark, R.L., Tan, J.W., Ahn, J.C., Gupta, S., Romero, P.A., and Venturelli, O.S. (2019). Microbial interaction network inference in microfluidic droplets. Cell Syst. *9*, 229–242.e4.

Jain, S., White, M., and Radivojac, P. (2017). Recovering true classifier performance in positive-unlabeled learning. In AAAI'17: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 2066–2072.

Jha, R.K., Gaiotto, T., Bradbury, A.R., and Strauss, C.E. (2014). An improved Protein G with higher affinity for human/rabbit IgG Fc domains exploiting a computationally designed polar network. Protein Eng. Des. Sel. *27*, 127–134.

Kehe, J., Kulesa, A., Ortiz, A., Ackerman, C.M., Thakku, S.G., Sellers, D., Kuehn, S., Gore, J., Friedman, J., and Blainey, P.C. (2019). Massively parallel screening of synthetic microbial communities. Proc. Natl. Acad. Sci. USA *116*, 12804–12809.

Klesmith, J.R., and Hackel, B.J. (2019). Improved mutant function prediction via PACT: protein analysis and classifier toolkit. Bioinformatics *35*, 2707–2712.

Kosuri, S., Goodman, D.B., Cambray, G., Mutalik, V.K., Gao, Y., Arkin, A.P., Endy, D., and Church, G.M. (2013). Composability of regulatory sequences controlling transcription and translation in Escherichia coli. Proc. Natl. Acad. Sci. USA *110*, 14024–14029.

Lange, K., Hunter, D.R., and Yang, I. (2000). Optimization transfer using surrogate objective functions. J. Comput. Graph. Stat. *9*, 1–20.

Langmead, B., and Salzberg, S.L. (2012). Fast gapped-read alignment with Bowtie 2′. Nat. Methods *9*, 357–359.

Lee, L.-F. (1982). Specification error in multinomial logit models. Journal of Econometrics *20*, 197–209.

Leinonen, R., Sugawara, H., and Shumway, M.; International Nucleotide Sequence Database Collaboration (2011). The sequence read archive. Nucleic Acids Res. *39*, D19–D21.

Liao, J., Warmuth, M.K., Govindarajan, S., Ness, J.E., Wang, R.P., Gustafsson, C., and Minshull, J. (2007). Engineering proteinase K using machine learning and synthetic genes. BMC Biotechnol. *7*, 16.

Liu, B., Dai, Y., Li, X., Lee, W.S., and Yu, P.S. (2003). Building text classifiers using positive and unlabeled examples (Third IEEE International Conference on Data Mining), pp. 179–186.

Mazurenko, S., Prokop, Z., and Damborsky, J. (2020). Machine Learning in Enzyme Engineering. ACS Cat. *10*, 1210–1223.

Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D.S., Sander, C., Zecchina, R., Onuchic, J.N., Hwa, T., and Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. Proc. Natl. Acad. Sci. USA *108*, E1293–E1301.

Mordelet, F., and Vert, J.P. (2011). ProDiGe: prioritization of disease genes with multitask machine learning from positive and unlabeled examples. BMC Bioinformatics *12*, 389.

Nadeau, C., and Bengio, Y. (2003). Inference for the generalization error. Mach. Learn. *52*, 239–281.

Olson, C.A., Wu, N.C., and Sun, R. (2014). A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. Curr. Biol. *24*, 2643–2651.

Ortega, J.M., and Rheinboldt, W.C. (1970). Iterative Solution of Nonlinear Equations in Several Variables (Academic Press).

Price, M.N., Wetmore, K.M., Waters, R.J., Callaghan, M., Ray, J., Liu, H., Kuehl, J.V., Melnyk, R.A., Lamson, J.S., Suh, Y., et al. (2018). Mutant phenotypes for thousands of bacterial genes of unknown function. Nature *557*, 503–509.

Riesselman, A.J., Ingraham, J.B., and Marks, D.S. (2018). Deep generative models of genetic variation capture the effects of mutations. Nat. Methods *15*, 816–822.

Robins, W.P., Faruque, S.M., and Mekalanos, J.J. (2013). Coupling mutagenesis and parallel deep sequencing to probe essential residues in a genome or gene. Proc. Natl. Acad. Sci. USA *110*, E848–E857.

Romero, P.A., Krause, A., and Arnold, F.H. (2013). Navigating the protein fitness landscape with Gaussian processes. Proc. Natl. Acad. Sci. USA *110*, E193–E201.

Romero, P.A., Tran, T.M., and Abate, A.R. (2015). Dissecting enzyme function with microfluidic-based deep mutational scanning. Proc. Natl. Acad. Sci. USA *112*, 7159–7164.

Sali, A., and Blundell, T.L. (1993). Comparative protein modelling by satisfaction of spatial restraints. J. Mol. Biol. *234*, 779–815.

Sauer-Eriksson, A.E., Kleywegt, G.J., Uhlén, M., and Jones, T.A. (1995). Crystal structure of the C2 fragment of streptococcal protein G in complex with the Fc domain of human IgG. Structure *3*, 265–278.

Sloan, D.J., and Hellinga, H.W. (1999). Dissection of the protein G B1 domain binding site for human IgG Fc fragment. Protein Sci. *8*, 1643–1648.

Song, H., and Raskutti, G. (2019). PUlasso: high-dimensional variable selection with presence-only data. J. Am. Stat. Assoc. *115*, 334–347.

Song, H., Dai, R., Raskutti, G., and Barber, R.F. (2020). Convex and Non-Convex Approaches for Statistical Inference with Class-Conditional Noisy Labels. J. Mach Learn. Res *21*, 1–58.

Song, Y., Dimaio, F., Wang, R.Y.R., Kim, D., Miles, C., Brunette, T., Thompson, J., and Baker, D. (2013). High-resolution comparative modeling with RosettaCM. Structure *21*, 1735–1742.

Suzek, B.E., Wang, Y., Huang, H., McGarvey, P.B., Wu, C.H., and UniProt, Consortium. (2015). UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinformatics *31*, 926–932.

Ward, G., Hastie, T., Barry, S., Elith, J., and Leathwick, J.R. (2009). Presence-only data and the em algorithm. Biometrics *65*, 554–563.

Weile, J., and Roth, F.P. (2018). Multiplexed assays of variant effects contribute to a growing genotype–phenotype atlas. Hum. Genet. *137*, 665–678.

Weile, J., Sun, S., Cote, A.G., Knapp, J., Verby, M., Mellor, J.C., Wu, Y., Pons, C., Wong, C., van Lieshout, N., et al. (2017). A framework for exhaustively mapping functional missense variants. Mol. Syst. Biol. *13*, 957.

Wheeler, T.J., and Eddy, S.R. (2013). Nhmmer: DNA homology search with profile HMMs. Bioinformatics *29*, 2487–2489.

Wrenbeck, E.E., Bedewitz, M.A., Klesmith, J.R., Noshin, S., Barry, C.S., and Whitehead, T.A. (2019). An automated data-driven pipeline for improving heterologous enzyme expression. ACS Synth. Biol. *8*, 474–481.

Wrenbeck, E.E., Faber, M.S., and Whitehead, T.A. (2017). Deep sequencing methods for protein engineering and design. Curr. Opin. Struct. Biol. *45*, 36–44.

Yang, K.K., Wu, Z., and Arnold, F.H. (2019). Machine-learning-guided directed evolution for protein engineering. Nat. Methods *16*, 687–694.

Yi, J., Hsieh, C.-J., Varshney, K.R., Zhang, L., and Li, Y. (2017). Scalable demand-aware recommendation. In Advances in Neural Information Processing Systems 30, I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds. (Curran Associates, Inc.), pp. 2412–2421.

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007





## STAR★METHODS

### KEY RESOURCES TABLE

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---|---|---|
| Bacterial and Virus Strains | | |
| E coli BL21(DE3) | Lucigen | Cat# 60401-1 |
| Chemicals, Peptides, and Recombinant Proteins | | |
| Golden Gate Assembly Kit | New England Biolabs | Cat#: E1601 |
| LB Broth | Fisher Scientific | Cat#: BP9723 |
| LB Agar | Fisher Scientific | Cat#: BP1425 |
| QIAprep Spin Miniprep Kit | Qiagen | Cat#: 27104 |
| MagicMedia *E. coli* Expression Medium | ThermoFisher Scientific | Cat#: K6803 |
| BugBuster® 10X Protein Extraction Reagent | Millipore | Cat#: 70921 |
| rLysozyme Solution | Millipore | Cat#: 71110 |
| DNase I | New England Biolabs | Cat#: M0303 |
| Potassium phosphate | Sigma Aldrich | Cat#: P5655 |
| 4-Methylumbelliferyl β-D-glucopyranoside | Sigma Aldrich | Cat#: M3633 |
| Deposited Data | | |
| Protein G domain B1 data set | Olson et al., 2014 | SRR2054943,SRR2054944 |
| Pyrrolidine Ketide Synthase data set | Wrenbeck et al., 2019 | SRR8238223,SRR8238221, SRR8238219,SRR8238217, SRR8238222,SRR8238220, SRR8238218,SRR8238216 |
| Levoglucosan kinase data set | Wrenbeck et al., 2019 | SRR8238225,SRR8238229, SRR8238231,SRR8238227, SRR8238224,SRR8238228, SRR8238230,SRR8238226 |
| Small ubiquitin-related modifier 1 data set | Weile et al., 2017 | SRR5680627,SRR5680629, SRR5680640,SRR5680625, SRR5680634,SRR5680638 |
| SUMO-conjugating enzyme UBC9 data set | Weile et al., 2017 | SRR5680697,SRR5680703, SRR5680713,SRR5680715, SRR5680731,SRR5680699, SRR5680705,SRR5680707, SRR5680727,SRR5680733 |
| Thiamin pyrophosphokinase 1 data set | Weile et al., 2017 | SRR5680677,SRR5680635, SRR5680674,SRR5680718, SRR5680717,SRR5680693, SRR5680694,SRR5680669, SRR5680632,SRR5680722, SRR5680719,SRR5680689, SRR5680692,SRR5680670 |
| Hemagglutinin data set | Doud and Bloom 2016 | SRR3113656,SRR3113660 |
| Beta-glucosidase data set | Romero et al., 2015 | SRR11472224, SRR11472223, SRR11472221, SRR11472220, SRR11472219, SRR11472215, SRR11472218, SRR11472216, SRR11472214, SRR11472217, SRR11472222 |
| 1-deoxy-D-xylulose-5-phosphate synthase data set | Romero Lab, UW-Madison | SRR11467347, SRR11467346 |

(*Continued on next page*)

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007



*Continued*

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---|---|---|
| Rocker data set | Romero Lab, UW-Madison | SRR12767727, SRR12767726 |
| Protein G domain B1 protein structure | Protein Data Bank | PDB ID: 1FCC |
| Beta-glucosidase protein structure | Protein Data Bank | PDB ID: 1GNX |
| Recombinant DNA | | |
| Beta-glucosidase gene (bgl3) from Streptomyces sp. strain QM-B814 | GenBank | GenBank: CAA82733.1 |
| Software and Algorithms | | |
| Bowtie2 | Langmead and Salzberg, 2012 | http://bowtie-bio.sourceforge.net/bowtie2 |
| Rosetta | Alford et al., 2017 | https://www.rosettacommons.org/software/ |
| EVmutation | Hopf et al., 2017 | https://marks.hms.harvard.edu/evmutation/ |
| DeepSequence | Riesselman et al., 2018 | https://github.com/debbiemarkslab/DeepSequence |
| HMMER | Wheeler and Eddy, 2013 | http://hmmer.org/ |
| PU learning code | This Paper | https://github.com/RomeroLab/pudms |
| scripts used to analyze the data sets and generate the figures | This Paper | https://github.com/RomeroLab/PU-learning-paper-analysis |

## RESOURCE AVAILABILITY

### Lead Contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the Lead Contact, Philip Romero (promero2@wisc.edu).

### Materials Availability

Plasmids generated in this are available from the Lead Contact.

### Data and Code Availability

This paper analyzes existing, publicly available data. These datasets' accession numbers are provided in the Key Resources Table and Table S1. The PU learning code is publicly available on GitHub: https://github.com/RomeroLab/pudms The scripts used to analyze the data sets and generate the figures reported in this paper are available on GitHub: https://github.com/RomeroLab/PU-learning-paper-analysis Any additional information required to reproduce this work is available from the Lead Contact."

## METHOD DETAILS

### Linear Sequence-Function Model

We model the sequence-function mapping $f$ as a linear function of amino acid sequence. Suppose we have a protein of length $L$ and a protein's activity can be described as the sum of individual amino acid contributions:

$$A_i = \sum_{j \in [L]} \sum_{aa \in \mathcal{S}} x_{i,j,aa} a_{j,aa}, \quad \text{(Equation 6)}$$

where $x_{i,j,aa} \in \{0, 1\}$ is a binary variable that specifies whether sequence $i$ has amino acid $aa$ at position $j$, $a_{j,aa}$ specifies the contribution of amino acid $aa$ at position $j$ to protein activity, $[L]$ is the set of all positions in the sequence $(1,\ldots,L)$, and $\mathcal{S}$ is the set of all 20 amino acids and the stop codon $(A,V,\ldots,*)$. We note that each sequence position takes on one and only one-amino acid option ($\sum_{aa \in \mathcal{S}} x_{i,j,aa} = 1$ for all $i,j$) and therefore the model is over-parameterized. We introduce a reduced model:

$$A_i = \sum_{j \in [L]} a_{j,\mathrm{ref}_j} + \sum_{j \in [L]} \sum_{aa \neq \mathrm{ref}_j} x_{i,j,aa} a'_{j,aa}, \quad \text{(Equation 7)}$$

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007



where $\text{ref}_j$ is a reference amino acid state for the $j$th position. This reference sequence is typically the wild-type parent sequence that was used to make the DMS library. It can be shown with simple algebra that $a\prime_{j,aa} = a_{j,aa} - a_{j,ref_j}$, and therefore $a'_{j,aa}$ parameterizes the effect of making an amino acid substitution from the reference state $ref_j$ to $aa$ at position $j$.

Our linear sequence-function model can be specified in vector notation as:

$$A_i = x_i^\top \theta + t \qquad \text{(Equation 8)}$$

where $x_i$ is a one-hot encoded vector that specifies a protein's amino acid sequence:

$$x_i = \left[1, (x_{i,1,aa})_{aa \neq ref_1}, \ldots, (x_{i,L,aa})_{aa \neq ref_L}\right] \qquad \text{(Equation 9)}$$

and the vector $\theta$ contains the model parameters:

$$\theta = \left[\underbrace{\left(\sum_{j \in [L]} a_{j,\text{ref}_j} - t\right)}_{\text{intercept}}, \underbrace{(a\prime_{1,aa})_{aa \neq \text{ref}_1}, \ldots, (a\prime_{L,aa})_{aa \neq \text{ref}_L}}_{\text{coefficients}}\right]$$

Here we see that the intercept term parameterizes the activity of the reference sequence relative to the activity threshold $t$.

## Site-Wise Enrichment Captures Marginal Effects, Which Are Biased

Here we demonstrate that site-wise enrichment scores capture marginal mutational effects that are in general biased for estimating an amino acid substitution's true effect $a'_{j,aa}$. Suppose the true positive-negative responses $(y_i)$ were available. Enrichment scores can be calculated from this positive-negative (PN) response as:

$$E^Y_{j,aa} = \log\left(\frac{\sum_{i \in P} x_{i,j,aa}/n_p}{\sum_{i \in N} x_{i,j,aa}/n_n}\right) - \log\left(\frac{\sum_{i \in P} x_{i,j,\text{ref}_j}/n_p}{\sum_{i \in N} x_{i,j,\text{ref}_j}/n_n}\right). \qquad \text{(Equation 10)}$$

where $P$ and $N$ are the positive and negative sets of sequences. We demonstrate that $E^Y_{j,aa}$ are in general biased for $a'_{j,aa}$.

Recall the logistic model from Equation 2 in the main text:

$$\mathbb{P}(Y_i = 1 | x_i) = \frac{\exp\left(\theta_0 + \sum_{j \in [L]} \sum_{aa \neq \text{ref}_j} x_{i,j,aa} a\prime_{j,aa}\right)}{1 + \exp\left(\theta_0 + \sum_{j \in [L]} \sum_{aa \neq \text{ref}_j} x_{i,j,aa} a\prime_{j,aa}\right)} \qquad \text{(Equation 11)}$$

where $\theta_0 = \sum_{j \in [L]} a_{j,\text{ref}_j} - t$ and $x_i = (x_{i,j,aa})_{j \in [L], aa \in \mathcal{S}}$ is a one-hot encoded vector of the sequence $i$. Maximizing the likelihood function $\ell(\theta) = \log(\prod_{i=1}^{n} \mathbb{P}(Y_i = 1 | (x_{i,j,aa})_{j \in [L], aa \in \mathcal{S}}))$ will produce consistent estimates for $a'_{j,aa}$. Consistent estimates will approach the true value of $a'_{j,aa}$ as the number of data points increases. It is a well-known result in statistics that the maximum likelihood estimator provides a consistent estimate.

Site-wise enrichment scores calculated from the true PN response $(y_i)$ are equivalent to the maximum likelihood estimates (MLE) from a logistic model that considers only one site at a time:

$$\mathbb{P}(Y_i = 1 | x_{ij}) = \frac{\exp\left(\gamma_{j,0} + \sum_{aa \neq \text{ref}_j} x_{i,j,aa} \gamma_{j,aa}\right)}{1 + \exp\left(\gamma_{j,0} + \sum_{aa \neq \text{ref}_j} x_{i,j,aa} \gamma_{j,aa}\right)}, \qquad \text{(Equation 12)}$$

where $x_{ij} = (x_{i,j,aa})_{aa \in \mathcal{S}}$. Note how this equation only includes terms related to the $j$th position, in contrast to Equation 11, which sums over all $j \in [L]$.

We demonstrate that the MLE of $\gamma_{j,aa}$ is equal to the enrichment score calculated from the true positive-negative responses $E^Y_{j,aa}$. The maximum likelihood estimate $\hat{\gamma} := (\hat{\gamma}_{j,0}, (\hat{\gamma}_{j,aa})_{aa \neq ref_j})$ solves likelihood equations

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \frac{\exp\left(\gamma_{j,0} + \sum_{aa \neq \text{ref}_j} \gamma_{j,aa} x_{i,j,aa}\right)}{1 + \exp\left(\gamma_{j,0} + \sum_{aa \neq \text{ref}_j} \gamma_{j,aa} x_{i,j,aa}\right)},$$

and for all $aa \neq \text{ref}_j$,

$$\sum_{i=1}^{n} y_i x_{i,j,aa} = \sum_{i=1}^{n} \frac{\exp\left(\gamma_{j,0} + \sum_{aa \neq \text{ref}_j} \gamma_{j,aa} x_{i,j,aa}\right)}{1 + \exp\left(\gamma_{j,0} + \sum_{aa \neq \text{ref}_j} \gamma_{j,aa} x_{i,j,aa}\right)} x_{i,j,aa}$$

$$= \sum_{i;x_{i,j,aa}=1} \frac{\exp(\gamma_{j,0} + \gamma_{j,aa})}{1 + \exp(\gamma_{j,0} + \gamma_{j,aa})}.$$

That is:

$$|\{i; y_i = 1\}| = |\{i; x_{i,j,\text{ref}_j} = 1\}| \frac{\exp(\gamma_{j,0})}{1 + \exp(\gamma_{j,0})} + \sum_{aa \neq \text{ref}_j} |\{i; x_{i,j,aa} = 1\}| \frac{\exp(\gamma_{j,0} + \gamma_{j,aa})}{1 + \exp(\gamma_{j,0} + \gamma_{j,aa})}, \quad \text{(Equation 13)}$$

and for all $aa \neq ref_j$,

$$|\{i; y_i = 1, x_{i,j,aa} = 1\}| = |\{i; x_{i,j,aa} = 1\}| \frac{\exp(\gamma_{j,0} + \gamma_{j,aa})}{1 + \exp(\gamma_{j,0} + \gamma_{j,aa})}, \quad \text{(Equation 14)}$$

where $|A|$ denotes the size of $A$. Solving Equations 13 and 14 for $\gamma_{j,aa}$,

$$\begin{aligned}\widehat{\gamma}_{j,aa} &= \log\left(\frac{|\{i; y_i = 1, x_{i,j,aa} = 1\}|}{|\{i; y_i = 0, x_{i,j,aa} = 1\}|}\right) - \log\left(\frac{|\{i; y_i = 1, x_{i,j,\text{ref}_j} = 1\}|}{|\{i; y_i = 0, x_{i,j,\text{ref}_j} = 1\}|}\right) \\ &= \log\left(\frac{\sum_{i \in P} x_{i,j,aa}}{\sum_{i \in N} x_{i,j,aa}}\right) - \log\left(\frac{\sum_{i \in P} x_{i,j,\text{ref}_j}}{\sum_{i \in N} x_{i,j,\text{ref}_j}}\right) \\ &= \log\left(\frac{\sum_{i \in P} x_{i,j,aa}/n_p}{\sum_{i \in N} x_{i,j,aa}/n_n}\right) - \log\left(\frac{\sum_{i \in P} x_{i,j,\text{ref}_j}/n_p}{\sum_{i \in N} x_{i,j,\text{ref}_j}/n_n}\right) = E^Y_{j,aa}.\end{aligned}$$

Thus $\widehat{\gamma}_{j,aa} = E^Y_{j,aa}$.

In addition, $\gamma_{j,aa} \neq a\prime_{j,aa}$ in general because the estimate for $\gamma_{j,aa}$ does not consider other sites and is therefore subject to omitted variable bias (e.g. see Lee (1982)). From this we can conclude that site-wise enrichment scores calculated from the true positive-negative response ($E^Y_{j,aa}$) provide biased estimates for $a'_{j,aa}$.

There exists one exception when sequence positions are independent given each response class. In this case, $\gamma_{j,aa} = a\prime_{j,aa}$ and enrichment scores from the positive-unlabeled response provide consistent estimates $a'_{j,aa}$. Class conditional independence between positions is unlikely to hold for any DMS data set due to biophysical interactions between sites, correlated sequence variables in multi-mutant sequences, and the nonlinear threshold in the response.

### Enrichment-Based Methods Provide Attenuated Estimates of a Mutation's Effect due to Latent Positive Sequences

Here we demonstrate that enrichment scores calculated from positive-unlabeled (PU) responses provide *biased* estimates of a mutation's effect. Consider the population enrichment score $\mathcal{E}^Y_{j,aa}$ calculated from the positive-negative (PN) responses:

$$\mathcal{E}^Y_{j,aa} := \log\left(\frac{\mathbb{P}(X_{j,aa} = 1 | Y = 1)}{\mathbb{P}(X_{j,aa} = 1 | Y = 0)}\right) - \log\left(\frac{\mathbb{P}(X_{j,\text{ref}_j} = 1 | Y = 1)}{\mathbb{P}(X_{j,\text{ref}_j} = 1 | Y = 0)}\right). \quad \text{(Equation 15)}$$

which correspond to an amino acid substitution's true marginal effect, i.e. the effect of changing site $j$ from $ref_j$ to amino acid $aa$ while allowing all other positions to vary. Also consider the population enrichment score $\mathcal{E}^Z_{j,aa}$ calculated from PU responses

$$\mathcal{E}^Z_{j,aa} := \log\left(\frac{\mathbb{P}(X_{j,aa} = 1 | Z = 1)}{\mathbb{P}(X_{j,aa} = 1 | Z = 0)}\right) - \log\left(\frac{\mathbb{P}(X_{j,\text{ref}_j} = 1 | Z = 1)}{\mathbb{P}(X_{j,\text{ref}_j} = 1 | Z = 0)}\right). \quad \text{(Equation 16)}$$

We consider $\mathcal{E}^Y_{j,aa}$ to be the true marginal effect, and show that enrichment calculated from the PU response $\mathcal{E}^Z_{j,aa}$ is not equivalent (i.e. $\mathcal{E}^Z_{j,aa} \neq \mathcal{E}^Y_{j,aa}$). We also demonstrate that $\mathcal{E}^Z_{j,aa}$ provides an attenuated estimate of the true mutational effect due to latent positive sequences in the unlabeled set.

Since labeled sequences are positive, for any $k \in (A, V, \ldots, *)$ we have

$$\mathbb{P}(X_{j,k} = 1 | Z = 1) = \mathbb{P}(X_{j,k} = 1 | Y = 1).$$

On the other hand,

$$\begin{aligned}\mathbb{P}(X_{j,k} = 1 | Z = 0) &= \frac{\mathbb{P}(X_{j,k} = 1, Y = 1, Z = 0) + \mathbb{P}(X_{j,k} = 1, Y = 0, Z = 0)}{\mathbb{P}(Z = 0)} \\ &= \mathbb{P}(X_{j,k} = 1 | Y = 1)\mathbb{P}(Y = 1 | Z = 0) + \mathbb{P}(X_{j,k} = 1 | Y = 0)\mathbb{P}(Y = 0 | Z = 0)\end{aligned}$$

Therefore,

$$\frac{\mathbb{P}(X_{j,k}=1|Z=0)}{\mathbb{P}(X_{j,k}=1|Z=1)}=\frac{\mathbb{P}(X_{j,k}=1|Y=1)\mathbb{P}(Y=1|Z=0)+\mathbb{P}(X_{j,k}=1|Y=0)\mathbb{P}(Y=0|Z=0)}{\mathbb{P}(X_{j,k}=1|Y=1)}$$
$$=\pi\cdot 1+(1-\pi)\frac{\mathbb{P}(X_{j,k}=1|Y=0)}{\mathbb{P}(X_{j,k}=1|Y=1)}. \quad \text{(Equation 17)}$$

where $\pi := \mathbb{P}(Y=1|Z=0)=\mathbb{P}(Y=1)$. Applying Equations 17 to 16 with $k\in\{ref_j, aa\}$,

$$\mathcal{E}^{Z}_{j,aa} := \log\left(\frac{\mathbb{P}\left(X_{j,\mathrm{ref}_j}=1|Z=0\right)}{\mathbb{P}\left(X_{j,\mathrm{ref}_j}=1|Z=1\right)}\right)-\log\left(\frac{\mathbb{P}(X_{j,aa}=1|Z=0)}{\mathbb{P}(X_{j,aa}=1|Z=1)}\right)$$
$$=\log\left(\frac{\pi\cdot 1+(1-\pi)\frac{\mathbb{P}\left(X_{j,\mathrm{ref}_j}=1\middle|Y=0\right)}{\mathbb{P}\left(X_{j,\mathrm{ref}_j}=1\middle|Y=1\right)}}{\pi\cdot 1+(1-\pi)\frac{\mathbb{P}\left(X_{j,aa}=1\middle|Y=0\right)}{\mathbb{P}\left(X_{j,aa}=1\middle|Y=1\right)}}\right)$$

To ease notation, let

$$R^{Y}_{k} := \frac{\mathbb{P}(X_{j,k}=1|Y=1)}{\mathbb{P}(X_{j,k}=1|Y=0)}.$$

Then,

$$\exp\left(\mathcal{E}^{Z}_{j,aa}\right) := \frac{\pi+(1-\pi)\frac{1}{R^{Y}_{ref_j}}}{\pi+(1-\pi)\frac{1}{R^{Y}_{aa}}}=\left(\frac{\pi R^{Y}_{ref_j}+(1-\pi)}{\pi R^{Y}_{aa}+(1-\pi)}\right)\frac{R^{Y}_{aa}}{R^{Y}_{ref_j}}=\left(\frac{\pi R^{Y}_{ref_j}+(1-\pi)}{\pi R^{Y}_{aa}+(1-\pi)}\right)\exp\left(\mathcal{E}^{Y}_{j,aa}\right)$$

Note $R^{Y}_{aa}>R^{Y}_{ref_j}$ if and only if $\exp\left(\mathcal{E}^{Z}_{j,aa}\right)<\exp\left(\mathcal{E}^{Y}_{j,aa}\right)$. In other words, $\mathcal{E}^{Y}_{j,aa}>0$ if and only if $\mathcal{E}^{Z}_{j,aa}<\mathcal{E}^{Y}_{j,aa}$. Since $\mathcal{E}^{Y}_{j,aa}=\log\left(R^{Y}_{aa}\right)-\log\left(R^{Y}_{ref_j}\right)$ by definition, $R^{Y}_{aa}>R^{Y}_{ref_j}$ is equivalent to $\mathcal{E}^{Y}_{j,aa}>0$. That is, $\mathcal{E}^{Y}_{j,aa}>0$ if and only if $\mathcal{E}^{Z}_{j,aa}<\mathcal{E}^{Y}_{j,aa}$. Therefore, $\mathcal{E}^{Z}_{j,aa}$ underestimates the effect amino acid substitutions with positive effects ($\mathcal{E}^{Y}_{j,aa}>0$) and overestimates substitutions with negative effects.

## Data Preprocessing

We obtained the ten large-scale sequence-function data sets from previously published work (Olson et al, 2014. Wrenbeck et al, 2019; Weile et al, 2017. Doud and Bloom, 2016; Romero et al., 2015) and the Sequence Read Archive (SRA) (Leinonen et al., 2011). The details of each data set and their SRA accession numbers are available in Table S1, and an overview of the data processing is summarized in Figure S2. For each data set, we obtained raw FASTQ files for both unlabeled and positive sequences, and mapped these reads to a reference sequence using Bowtie2 (Langmead and Salzberg, 2012). We translated the aligned gene sequences to amino acid sequences, and filtered the data sets to remove any amino acid substitutions that were observed less than ten times.

We used mode imputation to fill in any missing sequence information. Many of the analyzed data sets consisted of partial sequencing fragments (either tiled or random) because the entire gene was too long to cover with a paired-end Illumina read. The remainder of the sequence positions were unobserved. We used mode imputation to replace this missing sequence information. For nearly all DMS data sets, mode imputation simply replaces unobserved positions with the wild-type amino acid.

We converted protein sequence observations to a design matrix **X** using one-hot encoding. Each row of the design matrix **X** has the form in (Equation 9), where the reference amino acid sequence is taken to be the most frequent amino acid at each position (usually corresponding to the wild-type sequence). The DXS data set consisted of eight recombined gene fragments from one of the four DXS parent sequences (*E. coli*, *B. subtilis*, *Z. mobilis*, *P. trichocarpa*). These chimeric DXS sequences can be represented as an ordered sequence of "blocks" that indicates which parent the gene fragment was inherited from. We chose the *E. coli* DXS as the reference and generated dummy variables for each block change from the reference. Each data set resulted in two design matrices corresponding to unlabeled and positive sequences.

## PU Model Training

We trained PU models on the unlabeled and positive sequence sets for each protein data set. We computed the observed likelihood (Equation 3) for a given data set $(x_i, z_i)_{i=1}^{n}$ and a hyperparameter $\pi$. We found a stationary point of the negative observed log-likelihood using a Majorization-Minimization (MM) algorithm (Ortega and Rheinboldt, 1970; Lange et al., 2000). Specifically, starting from an initial parameter value which corresponds to the null model (no features in the model), we obtained a quadratic majorizer of the negative log-likelihood function at the current parameter value and updated the current parameter with a minimizer of the quadratic majorizer function. Since the majorizer function is greater than the negative likelihood at all points, the minimizer decreases the function value of the negative likelihood compared to the function value evaluated at the current parameter value, i.e. the minimizer increases the likelihood value. This process was repeated until convergence. For the implementation of this process, we have used the

**PUlasso** R package from the Comprehensive R Archive Network (CRAN) (Song and Raskutti, 2019), setting the regularization parameter $\lambda = 0$ to fit the un-penalized model.

The hyperparameter $\pi$ was either determined experimentally or tuned to maximize the model's classification performance. For hyperparameter tuning, we used twenty log-spaced $\pi$ values ranging from $10^{-3}$ to 0.5. For each $\pi$ value, we trained a model on 90% of the data set, used the model to make predictions on the remaining 10%, and generated a receiver operating characteristic (ROC) curve for the predictions using the labeled/unlabeled response $Z$. $\pi$ sets an upper limit on this labeled/unlabeled ROC curve (Figure S1), and in some instances the observed ROC curve exceeded this upper limit. These values of $\pi$ were determined to be infeasible because they resulted in true positive rates greater than the oracle classifier. We selected the $\pi$ value which resulted in the highest ROC-AUC value among feasible $\pi$ values.

## Aggregating Models from Multiple Replicates

In some instances we had data from multiple replicates that needed to be combined. For example, the Bgl3 high-temp data set had two experimental replicates ($R = 2$). We trained models on each individual replicate and then aggregated these results into a single model with estimated coefficients $\widehat{\theta}_{AGG}$ and variance-covariance matrix $\widehat{V}_{AGG}$. Let $(\widehat{\theta}^{(i)}, \widehat{V}^{(i)})$ be the estimated coefficients and the variance-covariance matrix of the coefficients from the $i$th replicate. Here $\widehat{V}^{(i)}$ was computed as an inverse of the estimated Fisher information at $\widehat{\theta}^{(i)}$. The aggregated coefficient and variance-covariance matrix were calculated as follows:

$$\widehat{\theta}_{AGG} = \frac{1}{R}\sum_{i=1}^{R}\widehat{\theta}^{(i)} \quad \text{and} \quad \widehat{V}_{AGG} = \widehat{W} + \left(1 + \frac{1}{R}\right)\widehat{B} \qquad \text{(Equation 18)}$$

where

$$\widehat{W} = \frac{1}{R}\sum_{i=1}^{R}\widehat{V}^{(i)} \quad \text{and} \quad \widehat{B} = \frac{1}{R-1}\sum_{i=1}^{R}\left(\widehat{\theta}^{(i)} - \widehat{\theta}_{AGG}\right)\left(\widehat{\theta}^{(i)} - \widehat{\theta}_{AGG}\right)^{\top}.$$

We note that the aggregated variance matrix $\widehat{V}_{AGG}$ has two components: $\widehat{W}$ for the variation in $\widehat{\theta}^{(i)}$ within each replicate and $\widehat{B}$ for the variation across different replicates, i.e. $(\widehat{\theta}^{(i)})_{i=1}^{R}$. Thus the form of $\widehat{V}_{AGG}$ in Equation 18 is a classical variance decomposition with the extra $1/R$ factor to account for the finite $R$ (see e.g. Carpenter and Kenward (2013)).

## Evaluating and Comparing Model Predictive Ability

We used the area under the receiver operating characteristic curve (ROC-AUC) to evaluate the predictive ability of each model. With PU data, we don't have negative examples and therefore we can't directly calculate a model's false positive rate (FPR). Instead, we used the labeled-unlabeled response ($Z$) to calculate the false positive rate ($FPR^{PU}$) and true positive rate ($TPR^{PU}$), and then performed a correction (Jain et al, 2017) to obtain ROC curves and ROC-AUC values:

$$FPR = \frac{FPR^{PU} - \pi TPR^{PU}}{1-\pi} \quad \text{and} \quad TPR = TPR^{PU} \quad \text{and} \quad ROC-AUC = \frac{ROC-AUC^{PU} - \pi/2}{1-\pi}.$$

This PU ROC curve correction is illustrated in Figure S1. We obtained corrected ROC curves and ROC-AUC values for each of the ten cross-validation folds, and then averaged over folds to obtain the model's corrected ROC curve and ROC-AUC value.

We used a corrected repeated cross-validation test to compare the predictive ability between the PU model and site-wise enrichment (Bouckaert and Frank, 2004). This test controls inflated type 1 error caused by data overlaps in cross-validation folds (Dietterich, 1998; Nadeau and Bengio, 2003) and also has high replicability (Bouckaert and Frank, 2004). The test involves running $K$-fold cross validation for $R$ independent runs and comparing models using a corrected test statistic. For each run $i = 1,\ldots,R$, we split the data randomly into $K$ sub-samples and fit one model for each fold $j = 1,\ldots,K$. We used $R = K = 10$ as recommended by the authors (Bouckaert and Frank, 2004). Let $ROC-AUC(M)_{ij}$ be the corrected ROC-AUC value for model M from the $i$th run and $j$th cross-validation fold. For the enrichment-based predictions, we used an additive model that summed all individual enrichment scores. We define the difference between the models $d_{ij} := \text{ROC} - \text{AUC}(\text{PU model})_{ij} - \text{ROC} - \text{AUC}(\text{enrich})_{ij}$ and the standard deviation of this difference $\widehat{\sigma}^2 = \frac{1}{K\cdot R-1}\sum_{ij}(d_{ij} - \overline{d})^2$. The test statistic $t$ was calculated as follows

$$t = \frac{\frac{1}{K\cdot R}\sum_{ij} d_{ij}}{\sqrt{\left(\frac{1}{K\cdot R} + \frac{1}{9}\right)\widehat{\sigma}^2}},$$

and was compared with the $t$ distribution with $K\cdot R - 1$ degrees of freedom.

## Predictions Using Rosetta, EVmutation, and DeepSequence

We made protein function predictions using established structure-based and unsupervised learning methods including Rosetta, EVmutation, and DeepSequence (Alford et al, 2017; Hopf et al, 2017; Riesselman et al, 2018). For Rosetta modeling, we searched

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007



the Protein Data Bank to identify the structure most similar to the DMS data's reference sequence and used this as a template for Rosetta comparative modeling using the default options (Song et al, 2013). We sampled 500 random sequences from both the unlabeled and positive sequence sets, built Rosetta models for these 1000 sequences, and calculated the Rosetta energy for each. We used these calculated Rosetta energies to create ROC curves classifying unlabeled and positive sets, and corrected these ROC curves to account for PU data.

We made predictions using EVmutation and DeepSequence models for each protein data set. We created family multiple sequence alignments (MSAs) using jackhmmer (Wheeler and Eddy, 2013) to query the DMS data set's reference sequence against the UniRef90 sequence database (Suzek et al, 2015). For DXS, we chose the *E. Coli* parent as the jackhmmer reference sequence. For Rocker, we had to relax the inclusion threshold (jackhmmer domE option set to 10000) to include additional sequences because Rocker is a *de novo* designed protein. We filtered the jackhmmer results to remove amino acid insertions relative to the reference sequence and also removed any resulting sequences that had less than 50% coverage over the reference sequence. We trained EVmutation and DeepSequence models on these curated MSAs using the default options. For EVmutation, we scored all sequences in each data set, except for GB1, where we sampled $10^6$ random sequences and DXS, where we sampled $10^4$ random sequences. For DeepSequence, we scored $10^4$ random sequences from each data set. We used the EVmutation and DeepSequence scores to create corrected ROC curves for each data set.

### Statistical Hypothesis Testing

We performed hypothesis tests to determine which features "significantly" affect protein function. We calculated the Z statistic $z_j$ to test whether a feature $j$ affects protein function or not (i.e. $H_0 : \theta_j = 0$):

$$z_j := \widehat{\theta}_j \Big/ \sqrt{\widehat{V}_{jj}},$$

where $\widehat{V}$ is the estimated variance-covariance matrix of $\widehat{\theta}$, computed as the inverse of the estimated Fisher information. We obtained p-values under the null hypothesis that $\theta_j = 0$ and computing tail probabilities:

$$\text{p value for the feature } j = 2 \cdot \mathbb{P}(Z \geq |z_j|),$$

where $Z$ is a standard normal variable. These p-values were then adjusted using the Benjamini-Hochberg (BH) procedure to account for multiple hypothesis testing and control the false discovery rate (Benjamini and Hochberg, 1995). We considered a feature to be significant if its BH-adjusted p-value was less than 0.05.

### Evaluating PU Model Stability

We used 10-fold cross-validation to evaluate the stability of the fitted PU model's parameter estimates and selected features. We calculated the coefficient of variation for each feature $j$ across different cross-validation folds:

$$\mathrm{CV}_j = \frac{\sqrt{\sum_{i=1}^{10} \left(\widehat{\theta}_j^{(i)} - \overline{\widehat{\theta}_j}\right)^2 \Big/ (10-1)}}{\sum_{i=1}^{10} \widehat{\theta}_j^{(i)} \Big/ 10} \qquad \text{(Equation 19)}$$

where $\widehat{\theta}_j^{(i)}$ is an estimated coefficient for the $j$th feature from the $i$th cross-validation fold. The absolute value of this coefficient of variation is a measure of a coefficient's relative variability (Figure 3D).

We evaluated feature selection stability by comparing the set of selected features across different cross-validation folds. We defined a selection stability measure (SS) as the average fraction of common selected features:

$$SS_k := \frac{1}{\binom{10}{2}} \sum_{i<j} \frac{\left|F_k^{(i)} \cap F_k^{(j)}\right|}{k}.$$

where $k$ is the selection size, $I$ and $j$ are different cross-validation folds, and $F_k^{(i)}$ is the set containing features with the $k$ smallest p-values from the $i$th fold. We computed $SS_k$ for $k = 1, \ldots, K$, where $K$ was chosen to be the number of the significant features (BH-adjusted $p<0.05$). We then averaged $SS_k$ over $k$ to obtain the average feature selection stability $\overline{SS}$. A value of $\overline{SS}$ that approaches one indicates that many common features are selected across different cross-validation folds (Figure S3D).

### Beta-Glucosidase Cloning, Expression, and Characterization

We designed the genes encoding the Bgl3 variants by making codon substitutions into a base Bgl3 gene sequence. If there were multiple codon options for an amino acid, we chose the particular codon randomly from a set of 31 codons that are optimized for expression in E coli (Boël et al, 2016). We ordered the designed genes as gBlocks from IDT and cloned them into a protein expression vector (pET22b) using GoldenGate cloning. We verified the sequences of all genes using Sanger sequencing with the T7 promoter and T7 terminator primers.

Please cite this article in press as: Song et al., Inferring Protein Sequence-Function Relationships with Large-Scale Positive-Unlabeled Learning, Cell Systems (2020), https://doi.org/10.1016/j.cels.2020.10.007



We expressed the Bgl3 variants shaking at 30°C in a 5-mL MagicMedia (Invitrogen) culture overnight. We then pelleted the expression culture by centrifugation and froze at -20°C. We resuspended the cell pellets in lysis buffer [0.3× BugBuster (Novagen), 30 kU/mL rLysozyme (Novagen), and 50 U/mL DNase I (New England Biolabs) in 100 mM potassium phosphate, pH 7.2] and performed serial dilutions to determine the linear range of the enzyme assay. We then diluted all samples in 100 mM potassium phosphate, pH 7.2 to be within the linear range and have similar end-point activities.

We arrayed the diluted cell extracts into 96-well PCR plates and heated the samples over multiple temperatures (50–75°C) for 15 min using a gradient thermocycler. After the heat step, we quantified the remaining functional enzyme by adding the fluorogenic substrate 4-methylumbelliferyl-$\beta$-D-glucopyranoside (Sigma) to a final concentration of 1 mM. We monitored the reaction progress by fluorescence spectroscopy (372 nm excitation/445 nm emission), and determined the rate by fitting a linear function to the progress curves. We normalized all rates to enzyme samples that had been incubated at room temperature (25°C). The $T_{50}$ (temperature where 50% of the protein is inactivated in 15 min) was determined by fitting a shifted sigmoid function to the thermal inactivation curves. All measurements were performed in at least duplicate with the mean $T_{50}$ values reported.

## QUANTIFICATION AND STATISTICAL ANALYSIS

In Figure 2B, the PU model's Corrected AUC error bars were calculated by taking the standard deviation of AUCs across the ten cross-validation folds. In Figure 2C, we used a corrected repeated cross-validation test to compare the predictive ability between the PU model and site-wise enrichment (Bouckaert and Frank, 2004). The full details of this test are given in the Methods Details section. Statistical significance for PU model coefficients was determined using Benjamini-Hochberg (BH) adjusted p-values (Benjamini and Hochberg, 1995). We considered a coefficient to be significant if its BH-adjusted p-value was less than 0.05.