Cell Systems, Volume *12*

Supplemental information

Informed training set design enables efficient machine learning-assisted directed protein evolution

Bruce J. Wittmann, Yisong Yue, and Frances H. Arnold

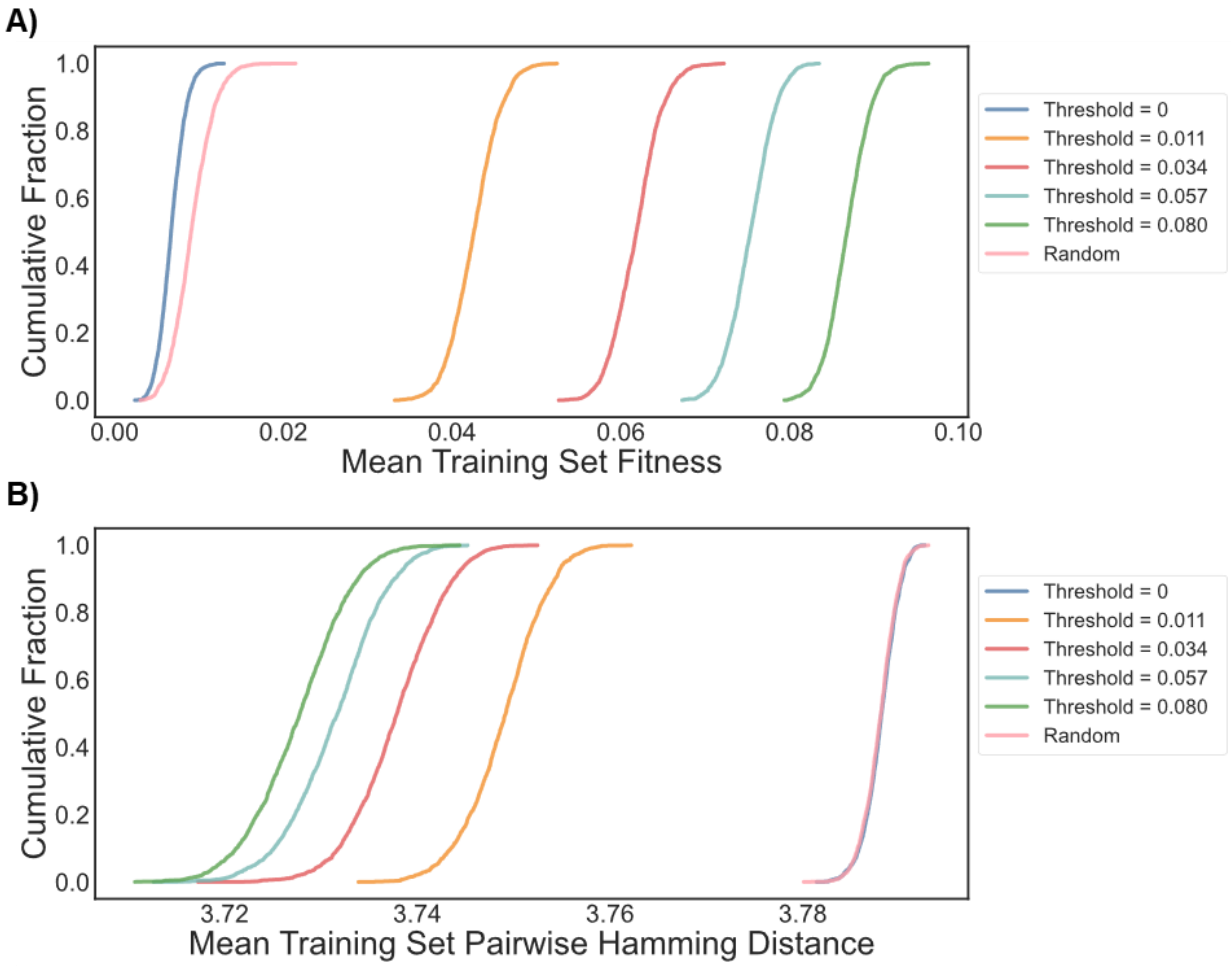

Figure S1. Summary statistics (shown as empirical cumulative distribution functions) for the 2000 training sets (each consisting of 384 samples) designed to be enriched in fit variants, related to Figure 3. For a given threshold, 50% of variants in the training data have fitness greater than or equal to the threshold and the remainder have fitness less than or equal to the threshold. When the threshold = 0, 100% of variants in the training data have fitness greater than or equal to 0. For all thresholds (including the one with a threshold at 0), the maximum allowed fitness in the training data is 0.34. The random sample is equivalent to the data with a threshold at 0, but does not have the upper bound on training fitness. (A) Plots of the mean fitness of all variants in a training set for all 2000 training sets designed to be enriched in fitness. As the fitness threshold increases, the mean fitness rises as expected. (B) Plots of the mean pairwise hamming distance between all members of a training set for all 2000 training sets designed to be enriched in fitness. A higher mean pairwise hamming distance indicates greater sequence diversity in the training data. As the fitness threshold increases, the mean pairwise hamming distance decreases. This is because the training data is increasingly restricted to the narrow regions of sequence space that contain higher-fitness variants.

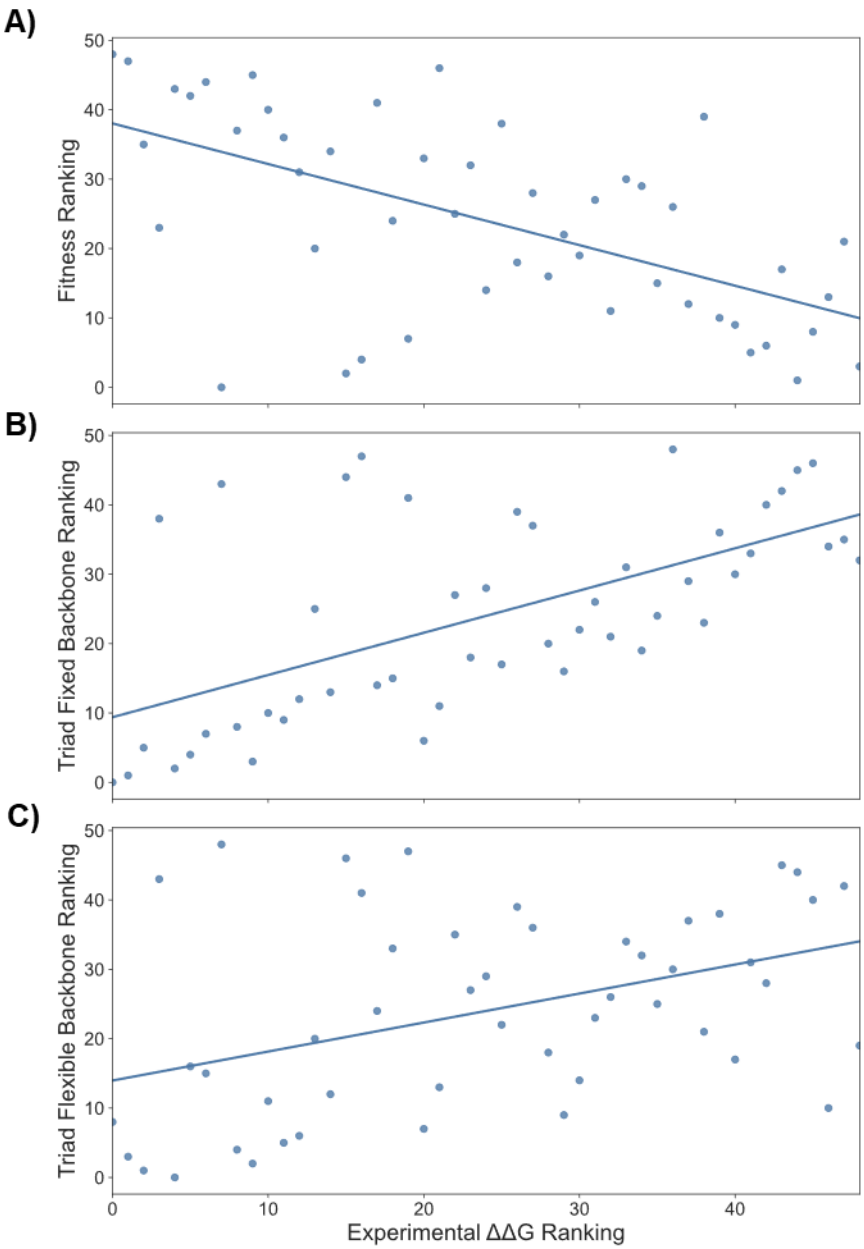

Figure S2. (A) Relationship between experimentally determined ΔΔG and GB1 fitness for single mutants at positions V39, D40, G41, and V54; both metrics are ranked from lowest value to highest value; related to Figure 4. We define a lower ΔΔG to be stabilizing and a higher ΔΔG to be destabilizing. The fitness of GB1 (at least at the considered positions) is loosely correlated with ΔΔG, but is clearly not the only determinant, with some lower-fitness variants having low ΔΔG and some higher-fitness variants having high ΔΔG. (B) Comparison of predicted ΔΔG upon mutation for GB1 variants using fixed backbone calculations to experimentally measured values of ΔΔG for single mutants at positions V39, D40, G41, and V54. (C) Comparison of predicted ΔΔG for GB1 variants using flexible backbone calculations to experimentally measured values of ΔΔG for single mutants at positions V39, D40, G41, and V54.

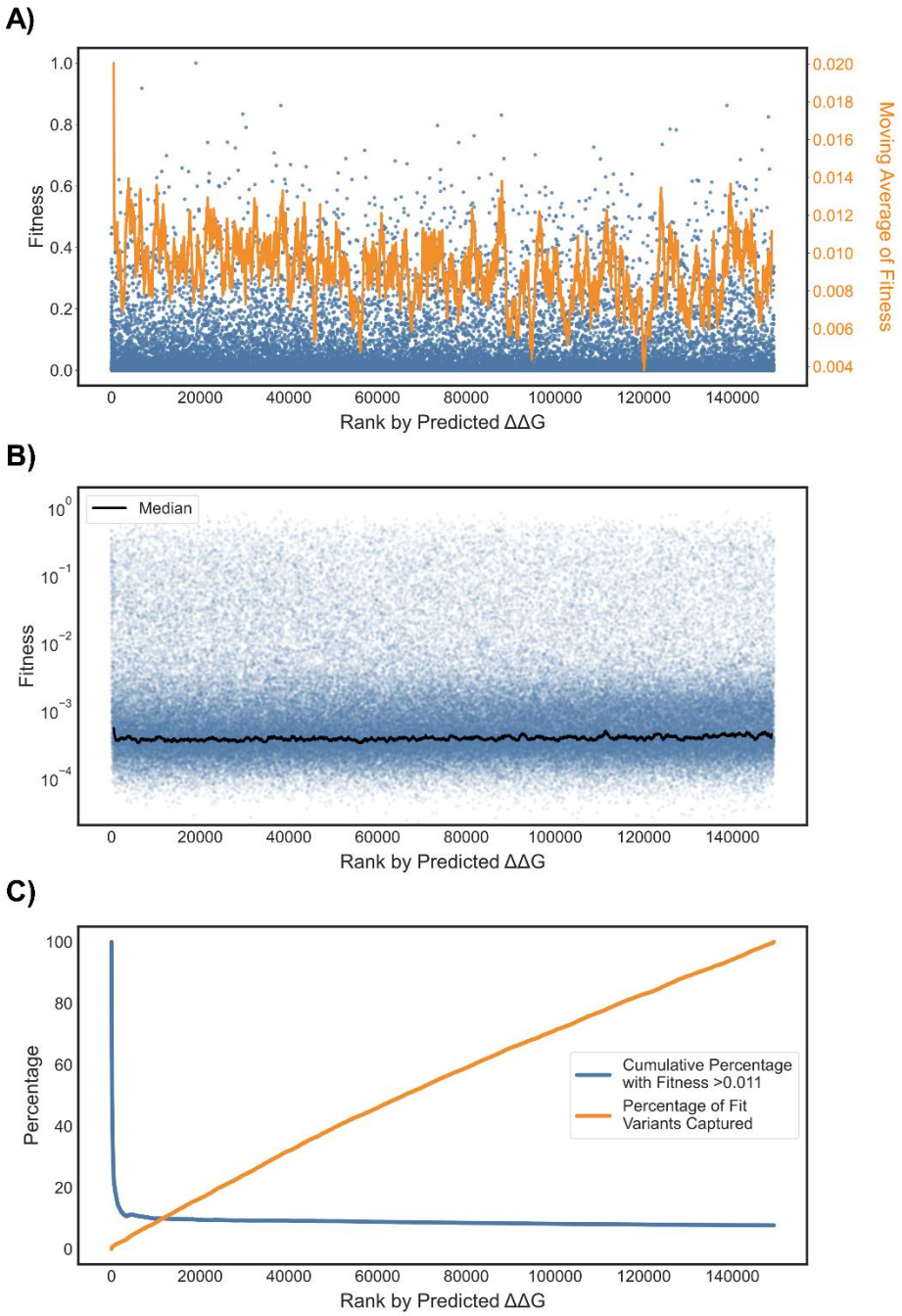

Figure S3. Results of zero-shot prediction using flexible backbone Triad ΔΔG calculations, related to Figure 4. (A) The fitness of all GB1 variants plotted against the rank (from lowest to highest ΔΔG) given by Triad calculations. Blue dots are all individual variants while the orange line is the sliding mean (window size = 1000) of fitness. (B) The log-fitness of all GB1 variants plotted against the rank given by Triad calculations. Blue dots are all individual variants while the black line is the sliding median (window size = 1000) of fitness. (C) Cumulative fitness metrics for all GB1 variants ranked by Triad score. The blue curve gives the percentage of variants ranked up to and including a given Triad rank that have fitness greater than 0.011. The orange curve gives the percentage of all "fit" (defined as fitness greater than 0.011) variants encompassed in the set up to and including a given Triad rank.

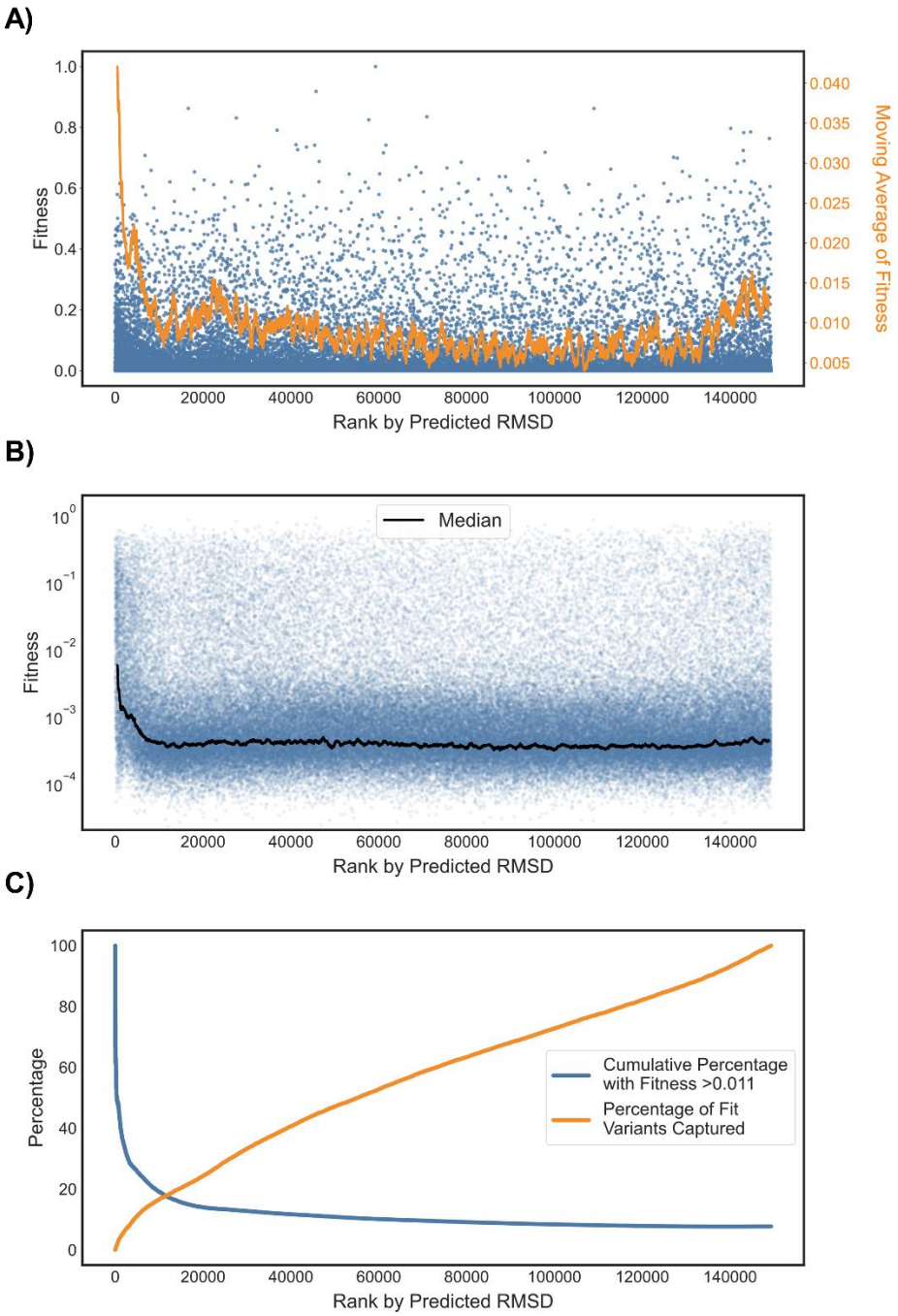

Figure S4. Results of zero-shot prediction using flexible backbone Triad root mean squared deviation (RMSD) calculations, related to Figure 4. (A) The fitness of all GB1 variants plotted against the rank (from lowest to highest RMSD) given by Triad calculations. Blue dots are all individual variants while the orange line is the sliding mean (window size = 1000) of fitness. (B) The log-fitness of all GB1 variants plotted against the rank given by Triad calculations. Blue dots are all individual variants while the black line is the sliding median (window size = 1000) of fitness. (C) Cumulative fitness metrics for all GB1 variants ranked by Triad score. The blue curve gives the percentage of variants ranked up to and including a given Triad rank that have fitness greater than 0.011. The orange curve gives the percentage of all "fit" (defined as fitness greater than 0.011) variants encompassed in the set up to and including a given Triad rank.

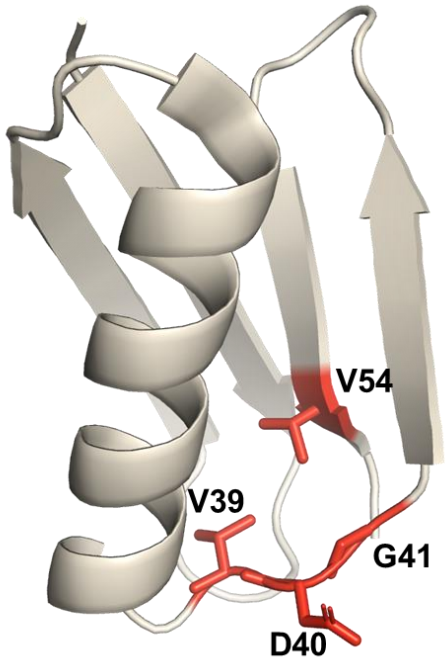

Figure S5. GB1 crystal structure (PDB: 2GI9) with the positions mutated in the GB1 combinatorial landscape highlighted in red, related to Figure 4.

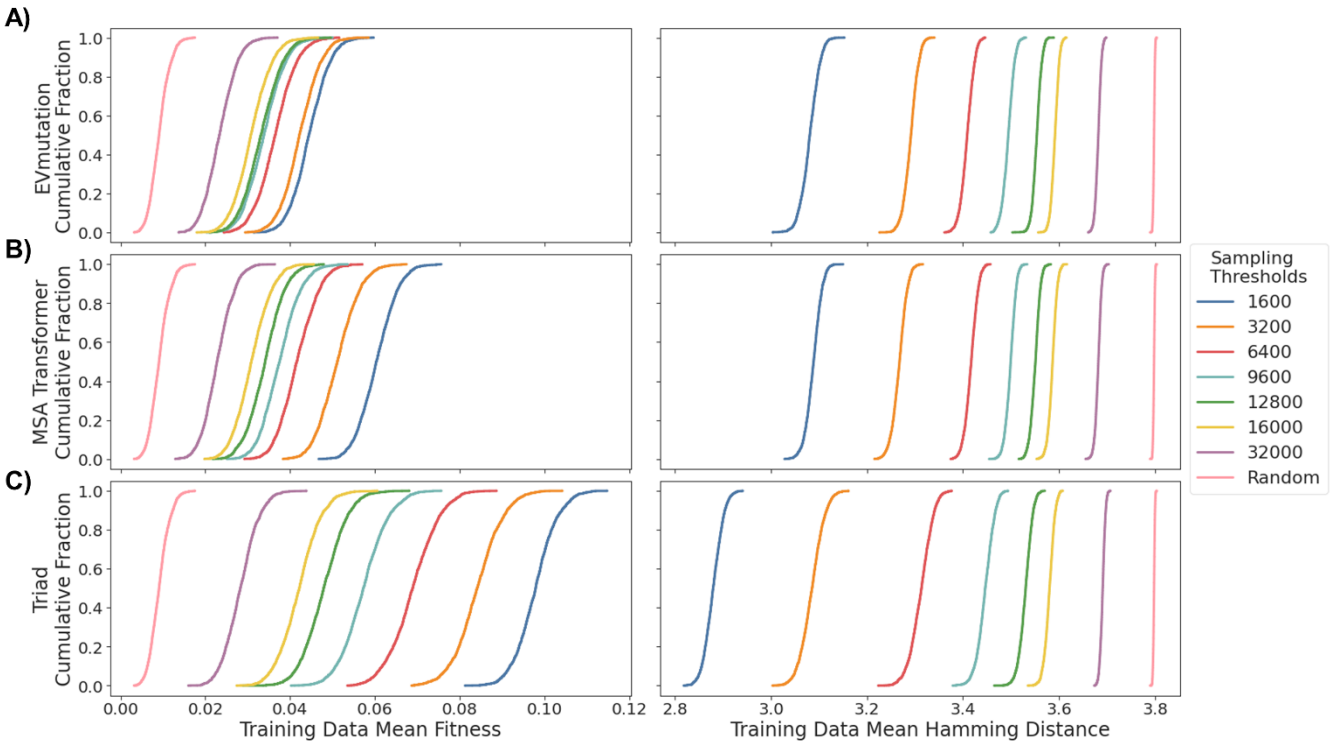

Figure S6. Summary statistics (shown as empirical cumulative distribution functions) for the 384-sample training sets generated using all zero-shot predictors, related to Figures 5 and 6. For each subplot (A–C), the left panel shows the mean fitness of all variants in a training set for all 2000 training sets derived for each sampling threshold; the right panel shows the mean pairwise hamming distance between all members of a training set for all 2000 training sets derived from each sampling threshold. In all subplots, as the threshold increases, the mean fitness decreases as more low-fitness variants have the potential to be included in the training data. Likewise, as the threshold increases, the mean pairwise hamming distance also increases. This is because predictive algorithms will tend to group similar sequences as having similar properties, and so sequences close in rank-order (as given by the zero-shot predictors) will be similar. By increasing the range of the rank from which we sample, the range of sequences sampled from is thus also increased. (A) The summary statistics for training data derived from EVmutation zero-shot predictions. (B) The summary statistics for training data derived from zero-shot predictions made using the MSA Transformer for masked token prediction. (C) The summary statistics for training data derived from zero-shot predictions made using predicted ΔΔG with a fixed backbone.

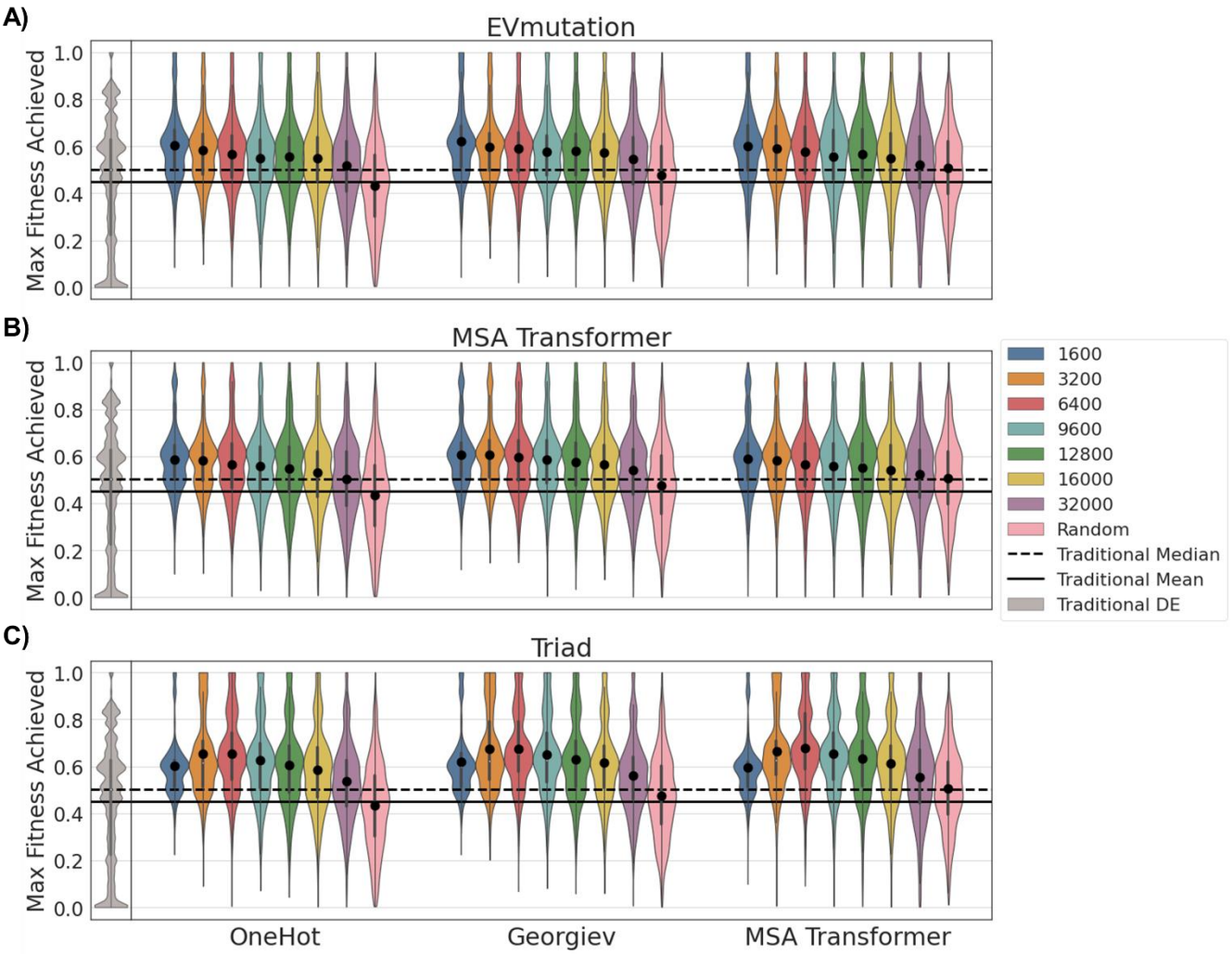

Figure S7. Zero-shot prediction for training set design enables highly effective ftMLDE on the GB1 landscape, as measured by maximum fitness achieved in simulated experiments, related to Figure 5. Each subplot (A–C) shows the effect of different zero-shot predictors on the maximum fitness achieved in simulated ftMLDE experiments. Each violin (except for the grey ones corresponding to simulated traditional DE) represents data from 2000 simulated experiments where 48 variants were used for training and the top 32 predictions were tested. The major groupings of violins within each subplot correspond to different encoding strategies (one-hot, Georgiev parameters, or learned embeddings from the MSA Transformer). The color of each violin corresponds to the zero-shot sampling threshold (i.e., the number of best-ranked variants according to a zero-shot predictor from which random samples were drawn to generate training data). Results of ftMLDE are compared to the results of simulated traditional DE (at the left of each plot, in grey) and standard MLDE (the three pink violins in each plot). (A) The maximum fitness achieved by simulated ftMLDE when EVmutation was used as the zero-shot predictor for training set design. (B) The maximum fitness achieved by simulated ftMLDE when a mask-filling protocol using the MSA Transformer was used as the zero-shot predictor for training set design. (C) The maximum fitness achieved by simulated ftMLDE when predicted ΔΔG was used as the zero-shot predictor for training set design.

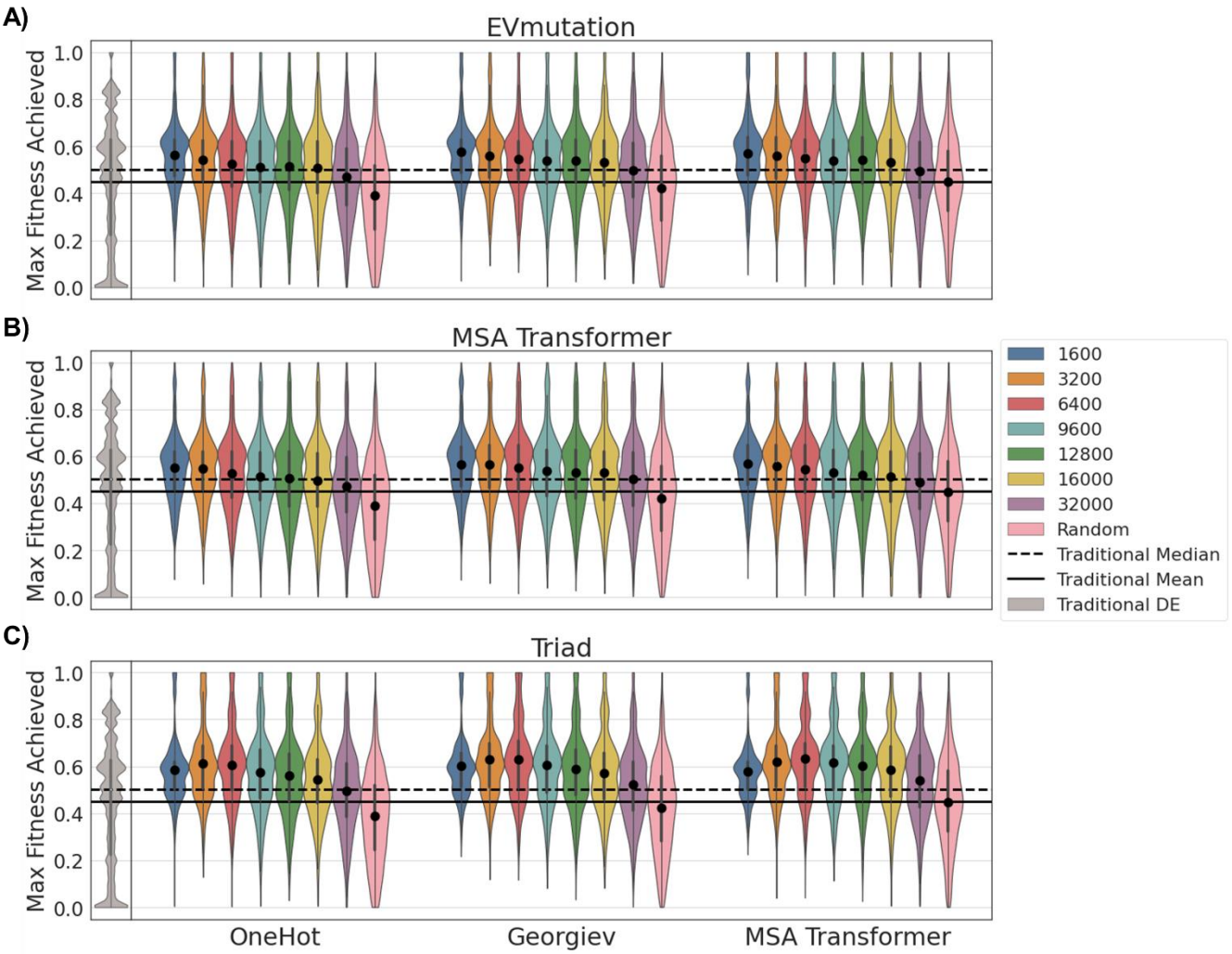

Figure S8. Zero-shot prediction for training set design enables highly effective ftMLDE on the GB1 landscape, as measured by maximum fitness achieved in simulated experiments, related to Figure 5. Each subplot (A–C) shows the effect of different zero-shot predictors on the maximum fitness achieved in simulated ftMLDE experiments. Each violin (except for the grey ones corresponding to simulated traditional DE) represents data from 2000 simulated experiments where 24 variants were used for training and the top 56 predictions were tested. The major groupings of violins within each subplot correspond to different encoding strategies (one-hot, Georgiev parameters, or learned embeddings from the MSA Transformer). The color of each violin corresponds to the zero-shot sampling threshold (i.e., the number of best-ranked variants according to a zero-shot predictor from which random samples were drawn to generate training data). Results of ftMLDE are compared to the results of simulated traditional DE (at the left of each plot, in grey) and standard MLDE (the three pink violins in each plot). (A) The maximum fitness achieved by simulated ftMLDE when EVmutation was used as the zero-shot predictor for training set design. (B) The maximum fitness achieved by simulated ftMLDE when a mask-filling protocol using the MSA Transformer was used as the zero-shot predictor for training set design. (C) The maximum fitness achieved by simulated ftMLDE when predicted ΔΔG was used as the zero-shot predictor for training set design.

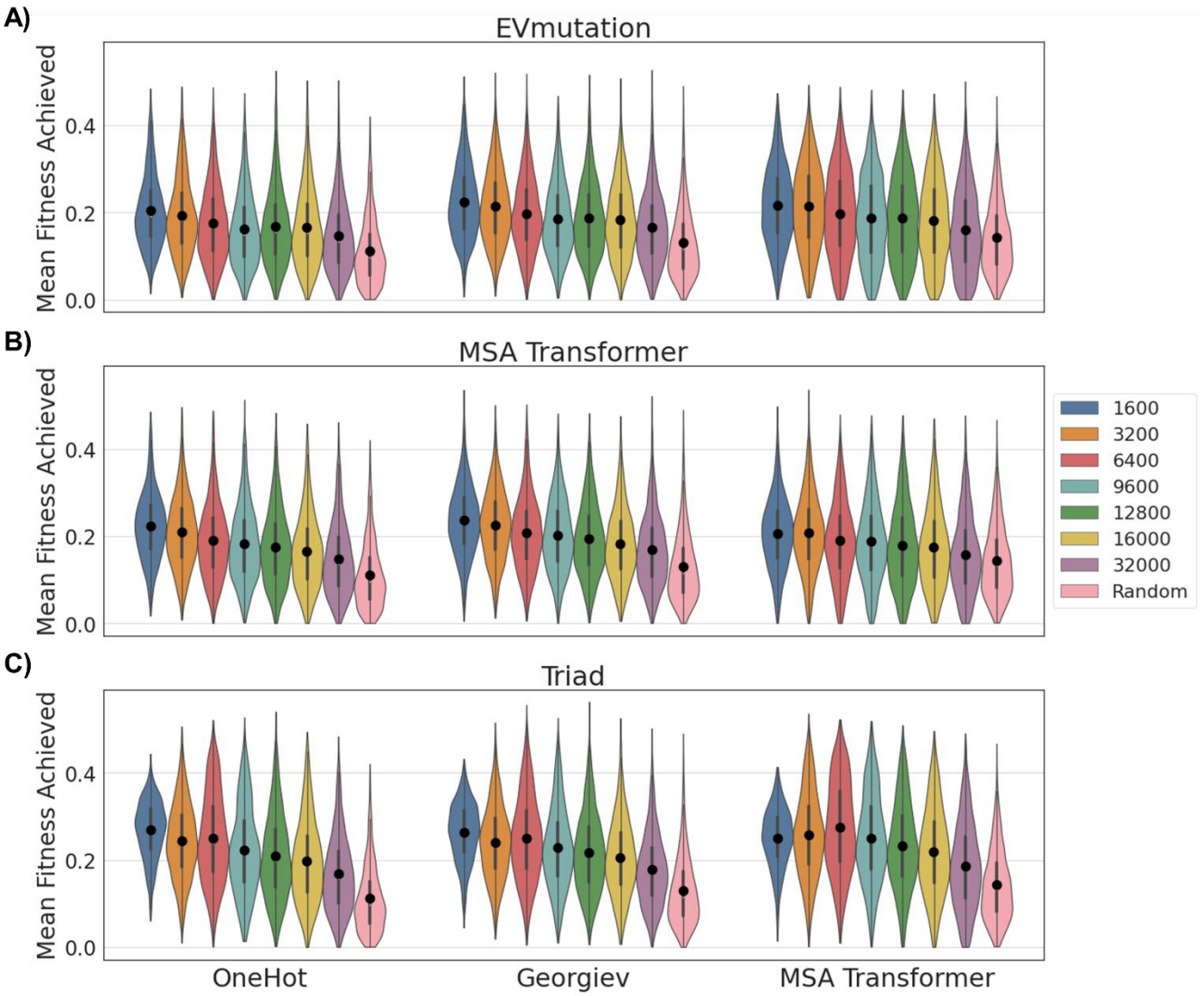

Figure S9. Zero-shot prediction for training set design enables highly effective ftMLDE on the GB1 landscape, as measured by mean fitness achieved in simulated experiments, related to Figure 6. Each subplot (A–C) shows the effect of different zero-shot predictors on the mean fitness achieved in simulated ftMLDE experiments. Each violin represents data from 2000 simulated experiments where 48 variants were used for training and the top 32 predictions were tested. The major groupings of violins within each subplot correspond to different encoding strategies (one-hot, Georgiev parameters, or learned embeddings from the MSA Transformer). The color of each violin corresponds to the zero-shot sampling threshold (i.e., the number of best-ranked variants according to a zero-shot predictor from which random samples were drawn to generate training data). Results of ftMLDE are compared to the results of standard MLDE (the three pink violins in each plot). (A) The mean fitness achieved by simulated ftMLDE when EVmutation was used as the zero-shot predictor for training set design. (B) The mean fitness achieved by simulated ftMLDE when a mask-filling protocol using the MSA Transformer was used as the zero-shot predictor for training set design. (C) The mean fitness achieved by simulated ftMLDE when predicted ΔΔG was used as the zero-shot predictor for training set design.

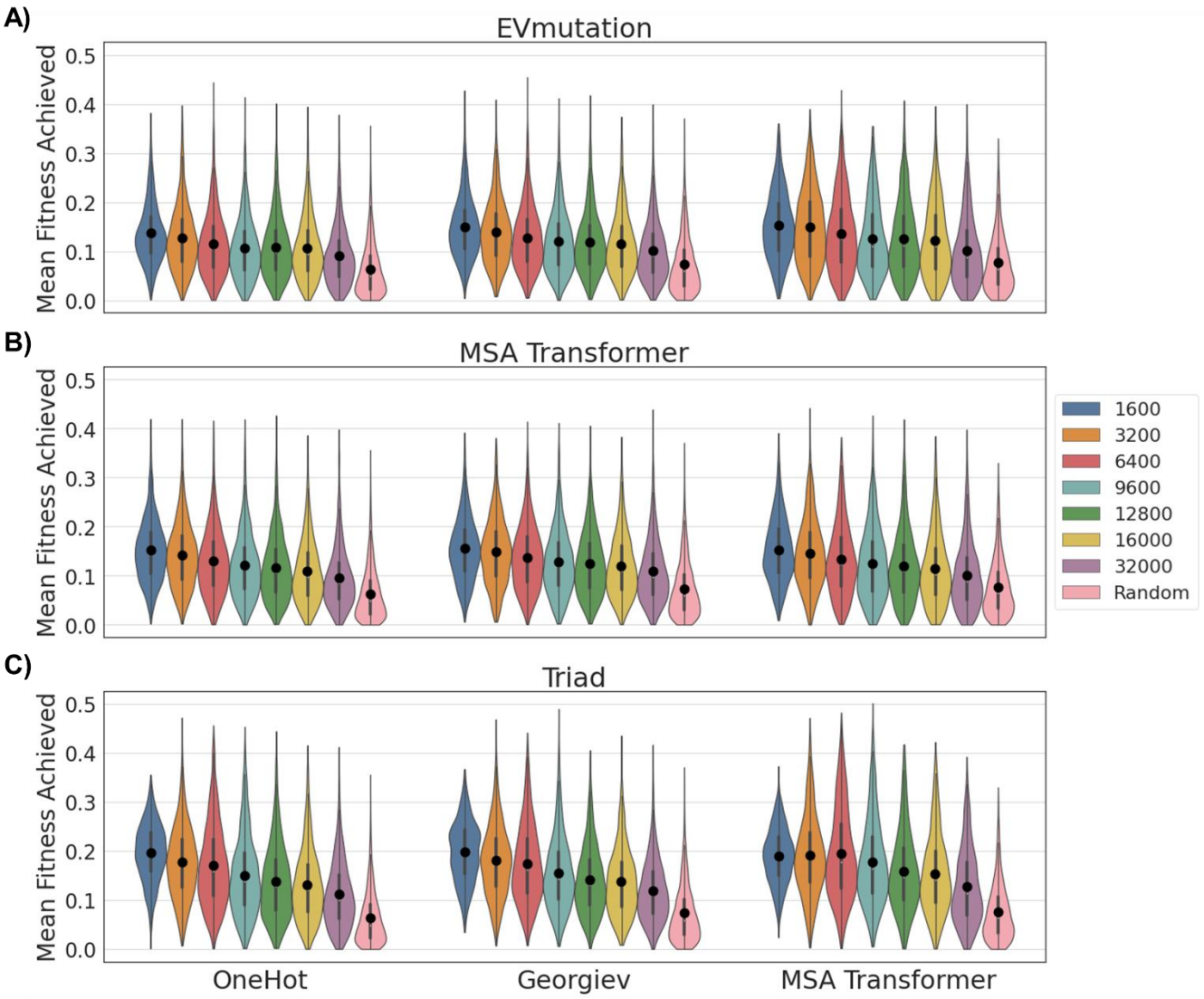

Figure S10. Zero-shot prediction for training set design enables highly effective ftMLDE on the GB1 landscape, as measured by mean fitness achieved in simulated experiments, related to Figure 6. Each subplot (A–C) shows the effect of different zero-shot predictors on the mean fitness achieved in simulated ftMLDE experiments. Each violin represents data from 2000 simulated experiments where 24 variants were used for training and the top 56 predictions were tested. The major groupings of violins within each subplot correspond to different encoding strategies (one-hot, Georgiev parameters, or learned embeddings from the MSA Transformer). The color of each violin corresponds to the zero-shot sampling threshold (i.e., the number of best-ranked variants according to a zero-shot predictor from which random samples were drawn to generate training data). Results of ftMLDE are compared to the results of standard MLDE (the three pink violins in each plot). (A) The mean fitness achieved by simulated ftMLDE when EVmutation was used as the zero-shot predictor for training set design. (B) The mean fitness achieved by simulated ftMLDE when a mask-filling protocol using the MSA Transformer was used as the zero-shot predictor for training set design. (C) The mean fitness achieved by simulated ftMLDE when predicted ΔΔG was used as the zero-shot predictor for training set design.

Table S1. The frequency with which the 2-layer 1D convolutional neural network (1D CNN) architecture appeared in the top 3 models (as ranked by cross-validation error) over 2000 rounds of simulated MLDE for each encoding type by training data size, related to Figure 2. The 2-layer 1D CNN was particularly effective when trained on 384 points, especially for higher-dimensional encodings. The same could not be said for the model when trained on 24 or 48 training points.

Encoding	Amino Acid Encoding Dimensionality	Percentage 2-Layer CNN in Top 3, 384 Training Points	Percentage 2-Layer CNN in Top 3, 48 Training Points	Percentage 2-Layer CNN in Top 3, 24 Training Points
Georgiev	19	14.55%	2.60%	6.30%
OneHot	20	20.55%	3.45%	5.45%
Bepler	100	18.30%	2.15%	3.75%
ResNet	256	38.05%	6.70%	6.05%
TAPE-Transformer	512	57.80%	9.35%	7.15%
MSA Transformer	768	32.80%	3.70%	4.85%
ProtBert-BFD	1024	36.85%	3.65%	3.55%
Esm1b	1280	32.80%	3.00%	4.00%
UniRep	1900	51.65%	4.55%	5.30%
LSTM	2048	62.15%	5.20%	5.20%

Table S2. The frequency with which the 1-layer 1D convolutional neural network (1D CNN) architecture appeared in the top 3 models (as ranked by cross-validation error) over 2000 rounds of simulated MLDE for each encoding type by training data size, related to Figure 2. The 1-layer 1D CNN was generally not as effective as the 2-layer 1D CNN.

Encoding	Amino Acid Encoding Dimensionality	Percentage 1-Layer CNN in Top 3, 384 Training Points	Percentage 1-Layer CNN in Top 3, 48 Training Points	Percentage 1-Layer CNN in Top 3, 24 Training Points
Georgiev	19	4.20%	3.85%	6.50%
OneHot	20	9.05%	3.50%	5.90%
Bepler	100	1.90%	1.05%	2.40%
ResNet	256	2.20%	3.80%	4.25%
TAPE-Transformer	512	19.50%	9.75%	10.75%
MSA Transformer	768	11.85%	3.05%	5.45%
ProtBert-BFD	1024	6.55%	1.85%	4.90%
Esm1b	1280	11.00%	1.35%	5.90%
UniRep	1900	5.70%	1.40%	5.75%
LSTM	2048	7.95%	1.40%	6.50%

Table S3. The frequencies with which XGBoost models with a tree base model and trained with the Tweedie regression objective achieved a greater than or equal to MLDE outcome than the same models trained with the standard regression objective, related to Figure 2. MLDE outcome is measured by max fitness achieved, mean fitness achieved, and NDCG. Frequencies are calculated over 2000 simulated MLDE experiments for each combination of encoding and number of training points. Instances where Tweedie gave a greater than or equal to result compared to standard regression are bolded.

Encoding	Training Points	Max, Percent Tweedie ≥ Standard	Mean, Percent Tweedie ≥ Standard	NDCG, Percent Tweedie ≥ Standard
Bepler	24	**60.85%**	**63.40%**	**67.80%**
ESM1b	24	**69.50%**	**74.85%**	**80.60%**
Georgiev	24	**63.40%**	**67.60%**	**68.60%**
LSTM	24	**69.55%**	**75.15%**	**81.15%**
MSA Transformer	24	**67.30%**	**71.70%**	**75.50%**
OneHot	24	**55.30%**	**51.80%**	42.05%
ProtBert-BFD	24	**68.65%**	**72.95%**	**77.15%**
ResNet	24	**66.95%**	**69.05%**	**72.40%**
TAPE-Transformer	24	**66.35%**	**72.00%**	**78.25%**
UniRep	24	**69.90%**	**73.95%**	**80.25%**
Bepler	48	**69.70%**	**75.95%**	**87.90%**
ESM1b	48	**77.90%**	**87.60%**	**97.60%**
Georgiev	48	**71.45%**	**75.05%**	**86.05%**
LSTM	48	**74.90%**	**84.15%**	**96.40%**
MSA Transformer	48	**73.35%**	**84.20%**	**95.40%**
OneHot	48	**65.20%**	**62.80%**	**58.05%**
ProtBert-BFD	48	**77.50%**	**87.00%**	**97.55%**
ResNet	48	**71.05%**	**77.70%**	**88.85%**
TAPE-Transformer	48	**72.75%**	**82.55%**	**94.95%**
UniRep	48	**79.30%**	**86.60%**	**97.45%**
Bepler	384	**80.50%**	**96.55%**	**100.00%**
ESM1b	384	**71.80%**	**90.40%**	**99.00%**
Georgiev	384	**72.60%**	**76.05%**	**98.80%**
LSTM	384	**71.70%**	**88.95%**	**99.90%**
MSA Transformer	384	**68.35%**	**84.30%**	**99.20%**
OneHot	384	**74.75%**	**76.15%**	**94.35%**
ProtBert-BFD	384	**75.35%**	**92.60%**	**99.50%**
ResNet	384	**81.65%**	**95.05%**	**99.80%**
TAPE-Transformer	384	**73.65%**	**91.60%**	**99.65%**
UniRep	384	**72.40%**	**95.50%**	**100.00%**

Table S4. The frequencies with which XGBoost models with a linear base model and trained with the Tweedie regression objective achieved a greater than or equal to MLDE outcome than the same models trained with the standard regression objective, related to Figure 2. MLDE outcome is measured by max fitness achieved, mean fitness achieved, and NDCG. Frequencies are calculated over 2000 simulated MLDE experiments for each combination of encoding and number of training points. Instances where Tweedie gave a greater than or equal to result compared to standard regression are bolded.

Encoding	Training Points	Max, Percent Tweedie ≥ Standard	Mean, Percent Tweedie ≥ Standard	NDCG, Percent Tweedie ≥ Standard
Bepler	24	**58.85%**	**50.05%**	**73.80%**
ESM1b	24	45.40%	44.20%	**61.85%**
Georgiev	24	**87.70%**	**63.65%**	**81.35%**
LSTM	24	42.90%	42.30%	**62.60%**
MSA Transformer	24	**54.30%**	**53.55%**	**73.80%**
OneHot	24	**96.00%**	**55.10%**	**61.30%**
ProtBert-BFD	24	45.05%	43.85%	**62.90%**
ResNet	24	**67.60%**	**59.35%**	**75.95%**
TAPE-Transformer	24	49.00%	49.10%	**66.60%**
UniRep	24	46.90%	43.00%	**67.40%**
Bepler	48	**52.10%**	42.85%	**83.50%**
ESM1b	48	40.50%	41.15%	**62.80%**
Georgiev	48	**80.25%**	**67.45%**	**90.75%**
LSTM	48	30.30%	30.95%	**50.25%**
MSA Transformer	48	55.50%	**54.70%**	**80.10%**
OneHot	48	**94.45%**	**55.60%**	**67.00%**
ProtBert-BFD	48	48.35%	47.70%	**72.80%**
ResNet	48	**69.85%**	**57.20%**	**87.80%**
TAPE-Transformer	48	**57.70%**	**52.60%**	**84.45%**
UniRep	48	35.15%	31.60%	**57.95%**
Bepler	384	34.75%	17.30%	**70.60%**
ESM1b	384	46.35%	24.80%	**96.05%**
Georgiev	384	**72.10%**	42.90%	**96.75%**
LSTM	384	**64.10%**	**70.75%**	**99.90%**
MSA Transformer	384	**69.35%**	**63.15%**	**99.95%**
OneHot	384	**95.90%**	46.95%	**90.60%**
ProtBert-BFD	384	43.15%	25.10%	**99.50%**
ResNet	384	**66.00%**	**55.70%**	**92.85%**
TAPE-Transformer	384	**65.45%**	46.80%	**98.00%**
UniRep	384	49.25%	48.40%	**99.55%**

Table S5. Expected max of the top 96 predictions, mean of the top 96 predictions, and normalized discounted cumulative gain (NDCG) for the 2000 ftMLDE simulations performed using training data designed to be enriched in fit variants, related to Figure 3.

Thresh	Max Fitness Achieved	Mean Fitness Achieved	NDCG
0.000	0.656	0.210	0.821
0.011	0.778	0.313	0.885
0.034	0.795	0.331	0.894
0.057	0.810	0.343	0.899
0.080	0.806	0.345	0.901

Table S6. Effectiveness of zero shot strategies that did not rely on a mask filling protocol, related to Figure 4. A more positive Spearman ρ indicates a more effective prediction. The entry "Triad_FlexibleBb_ΔΔG" refers to zero-shot prediction made using predicted ΔΔG with a flexible protein backbone; the entry "Triad_FlexibleBb_RMSD" refers to zero-shot prediction made using predicted RMSD with a flexible protein backbone; the entry "Triad_FixedBb_ΔΔG" refers to zero-shot prediction made using predicted ΔΔG with a fixed protein backbone.

ZeroShotStrategy	Spearman ρ
Triad_FlexibleBb_ΔΔG	-0.02
DeepSequence	0.05
Triad_FlexibleBb_RMSD	0.06
EVmutation	0.21
Triad_FixedBb_ΔΔG	0.27

Table S7. Effectiveness of different transformer models for zero-shot predictions of GB1 fitness using a masked token prediction protocol, related to Figure 4. A more positive Spearman ρ indicates a more effective prediction. All models except the MSA Transformer showed poor predictive performance when used for zero-shot prediction; the mask filling rankings provided by many models showed negative correlation with GB1 fitness, indicating a prediction that was worse than a random guess.

Model Name	Parameters (Millions)	Training Data Source	Conditional Spearman ρ	Naïve Spearman ρ
esm1_t34_670M_UR50S	670	UniRef50	-0.12	-0.09
esm1_t34_670M_UR50D	670	UniRef100 Sampled Evenly Across UniRef50 Clusters	-0.11	-0.08
esm1_t12_85M_UR50S	85	UniRef50	-0.08	-0.06
ESM1b	650	UniRef50	-0.06	-0.03
esm1_t6_43M_UR50S	43	UniRef50	-0.05	-0.03
ProtBert-BFD	420	Big Fat Database (BFD)	-0.05	0.00
ProtBert	420	UniRef100	-0.02	0.00
esm1_t34_670M_UR100	670	UniRef100	0.03	0.04
MSA Transformer	100	MSAs of Each UniRef50 Sequence Against UniClust30	0.20	0.24

Table S8. The tunable parameters with their default values for the different neural network architectures used in MLDE, related to STAR Methods.

Architecture	Parameter	Description	Default
NoHidden	dropout	Dropout value for model	0.2
OneHidden	dropout	Dropout value for model	0.2
OneHidden	size1	Size of the hidden layer as a fraction of the encoding dimensionality	0.25
TwoHidden	dropout	Dropout value for model	0.2
TwoHidden	size1	Size of the hidden layer as a fraction of the encoding dimensionality	0.25
TwoHidden	size2	Size of the second hidden layer as a fraction of the encoding dimensionality	0.0625
OneConv	dropout	Dropout value for model	0.2
OneConv	filter_choice	The width of the 1D convolutional window as a fraction of the number of positions in the combinatorial space	0.5
OneConv	n_filters1	The number of filters used in the convolution as a fraction of the encoding dimensionality of a single position	0.0625
OneConv	flatten_choice	The method of flattening post convolution	"Average"
TwoConv	dropout	Dropout value for model	0.2
TwoConv	filter_arch	The widths of the two 1D convolutional windows as a fraction of the number of positions in the combinatorial space, given as a tuple	(0.5, 0.5)
TwoConv	n_filters1	The number of filters used in the first convolution as a fraction of the encoding dimensionality of a single position	0.0625
TwoConv	n_filters2	The number of filters used in the second convolution as a fraction of the encoding dimensionality of a single position	0.007813
TwoConv	flatten_choice	The method of flattening post convolution	"Average"

Table S9. The tunable parameters with their default values for the base models used in the XGBoost models of MLDE, related to STAR Methods. “See XGBoost docs” indicates that the default values provided in XGBoost were used.

Base Model	Parameter	Default Value
Linear	lambda	1
Linear	alpha	See XGBoost docs
Tree	eta	See XGBoost docs
Tree	max_depth	See XGBoost docs
Tree	lambda	See XGBoost docs
Tree	alpha	See XGBoost docs

Table S10. The tunable parameters with their default values for the scikit-learn models used in MLDE, related to STAR Methods. “See sklearn docs” indicates that the default values provided in scikit-learn were used.

Model	Parameter	Default Value
Linear	N/A	N/A
GradientBoostingRegressor	learning_rate	See sklearn docs
GradientBoostingRegressor	n_estimators	See sklearn docs
GradientBoostingRegressor	min_samples_split	See sklearn docs
GradientBoostingRegressor	min_samples_leaf	See sklearn docs
GradientBoostingRegressor	max_depth	See sklearn docs
RandomForestRegressor	n_estimators	See sklearn docs
RandomForestRegressor	min_samples_split	See sklearn docs
RandomForestRegressor	min_samples_leaf	See sklearn docs
RandomForestRegressor	max_depth	See sklearn docs
LinearSVR	tol	See sklearn docs
LinearSVR	C	See sklearn docs
LinearSVR	dual	See sklearn docs
ARDRegression	tol	See sklearn docs
ARDRegression	alpha_1	See sklearn docs
ARDRegression	alpha_2	See sklearn docs
ARDRegression	lambda_1	See sklearn docs
ARDRegression	lambda_2	See sklearn docs
KernelRidge	alpha	See sklearn docs
KernelRidge	kernel	See sklearn docs
BayesianRidge	tol	See sklearn docs
BayesianRidge	alpha_1	See sklearn docs
BayesianRidge	alpha_2	See sklearn docs
BayesianRidge	lambda_1	See sklearn docs
BayesianRidge	lambda_2	See sklearn docs
BaggingRegressor	n_estimators	See sklearn docs
BaggingRegressor	max_samples	See sklearn docs
LassoLarsCV	max_iter	See sklearn docs
LassoLarsCV	cv	5
LassoLarsCV	max_n_alphas	See sklearn docs
DecisionTreeRegressor	max_depth	See sklearn docs
DecisionTreeRegressor	min_samples_split	See sklearn docs
DecisionTreeRegressor	min_samples_leaf	See sklearn docs
SGDRegressor	alpha	See sklearn docs
SGDRegressor	l1_ratio	See sklearn docs
SGDRegressor	tol	See sklearn docs
KNeighborsRegressor	n_neighbors	See sklearn docs
KNeighborsRegressor	weights	See sklearn docs
KNeighborsRegressor	leaf_size	See sklearn docs
KNeighborsRegressor	p	See sklearn docs
ElasticNet	l1_ratio	See sklearn docs
ElasticNet	alpha	See sklearn docs
AdaBoostRegressor	n_estimators	See sklearn docs
AdaBoostRegressor	learning_rate	See sklearn docs