Supplementary Information
CLADE: Cluster learning-assisted directed evolution

Yuchi Qiu[1], Jian Hu[2,3] and Guo-Wei Wei *[1,3,4]

[1] *Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA*
[2] *Department of Chemistry, Michigan State University, MI, 48824, USA*
[3] *Department of Biochemistry and Molecular Biology, Michigan State University, MI, 48824, USA*
[4] *Department of Electrical and Computer Engineering, Michigan State University, MI 48824, USA*

Contents

*Corresponding author: weig@msu.edu

S1 Feature matrix

Suppose the combinatorial library consists of L mutation sites for mutagenesis. The feature matrix will encoding all possible variants (i.e. 20^L). Each amino acid at any mutation site is encoded as a vector with R-dimension (e.g. $R = 19$ for the Georgiev encoding [1, 2]). Then the embedding of the combinatory library leads to a tensor of shape $20^L \times L \times R$ (e.g. $160,000 \times 4 \times 19$). This three-dimensional feature tensor is standardized by function *StandardScalar*() in scikit-learn [3] to have centered mean and unit variance. We take this standardized tensor X as the input to both unsupervised clustering and supervised learning, and necessary reshape is needed.

In unsupervised clustering, X is reshaped into a two-dimensional matrix with shape $M \times LR$ (e.g. $160,000 \times 76$ for Georgiev on GB1). In supervised learning models, the same two-dimensional matrix was used for models except for the convolution neural networks. For one-dimensional convolution neural networks, the three-dimensional tensor X was taken as the input where M is the sample size, R is the dimension of the channel and L is the number of channels.

S2 Supervised learning model

The supervised learning model was implemented in the MLDE package [4]. It is an ensemble of 16 regression models optimized by Bayesian hyperparameter optimizations. Various regression architectures from scikit-learn [3], Keras [5], and XGBoost [6] were employed. Five-fold cross validation was used to evaluate the performance of each model in mean square errors. Bayesian hyperparameter optimizations were performed to find the best-performing hyperparameters, which is implement by package "hyperopt" [7]. After hyperparameter optimizations, the top three models were picked and averaged to predict the fitness of unlabeled variants. The list of 16 regression models was given in Table S6. Parameters that were optimized via hyperopt are listed with their default values and search space in Table S7. Other hyperparameters not optimized by hyperopt and supervised learning architectures are given below.

Keras models The architectures and the non-tunable hyperparameters are given here. In the three fully connected neural networks, a batch normalization layer and a "ReLu" activation layer were employed followed by each hidden layer. A dropout layer was used before the output layer. And a "ReLu" activation was used to obtain the scalar output. In the two convolution neural networks, the size and number of filters in the convolution layer were taken to be proportional to the dimension of the encoding for single amino acid and the number of mutation sites, respectively (Table S7). A batch normalization layer and a "ReLu" activation layer were employed followed by each convolutional layer. Then a single layer was used to flatten the convolution layer to the fully connected layer. This single layer can be taken as flatten, max pooling and, average pooling, given by the hyperparameter "flatten_choice" which was optimized by hyperopt (Table S7). Next, a dropout layer and a "ReLu" activation were employed to get the output layer. In all five neural network architectures, optimizer was taken as "adam" with a batch size of 32 and, "mse" loss was used. The number of epochs was taken as 1000 and the early stopping criterion for training was taken as 10 epochs if no further improvement.

XGBoost models The tree base model was employed with different loss functions. In Table S6, "Tree" used square errors, "reg:squarederror", and "Tree-Tweedie" used Tweedie loss function, "reg:tweedie". The Tweedie loss function is designed to learn imbalance data, like the fitness landscape enriched with low- or zero-fitness variants.

Scikit-learn models The default parameters from the scikit-learn package were used as the default hyperparameters in the MLDE model. They were then optimized.

S3 Simulations on cluster-learning sampling

Three simulated samplings with maximum hierarchy $N = 1$, $N = 2$ and $N = 3$ were presented to further illustrate the sampling process (Figure S1). The increments of clusters were taken to be 4 at any hierarchy and any N (e.g. $K_1 = K_2 = 4$ for $N = 2$). Variants were selected and screened in parallel at each batch and the batch size was taken as 96. In all cases, the sampling process selected and screened four batches of variants (i.e. 384 labeled variants). The first batch of variants (96 variants) was selected randomly and equally over clusters for any maximum hierarchy N. Starting from the second batch, variants were selected over clusters according to the sampling probabilities and the sampling probabilities were updated after each batch to allow oversampling in the cluster with high-fitness. With maximum hierarchy $N = 1$, the sampling was performed on 4 clusters from the second to fourth batch (Figure S1A). With maximum hierarchy $N = 2$, the sampling was performed on 4 clusters at the second batch. Then these 4 clusters were further divided into 8 clusters. The 4th cluster at hierarchy 1 having the highest average fitness was divided into 4 clusters at hierarchy 2. The 2nd cluster at hierarchy 1 having the second highest average fitness was divided into 2 clusters at hierarchy 2. The rest of the clusters, 1st and 3rd clusters, at hierarchy 1 were not further divided at hierarchy 2 because of their low average fitness. The third and fourth batches of variants were selected in the 8 clusters at hierarchy 2 (Figure S1B). With maximum hierarchy $N = 3$, the sampling was performed on 4 clusters at the second batch and 8 clusters at the third batch. The linkages of clusters at the first two hierarchies were identical to that in the case with maximum hierarchy $N = 2$. These 8 clusters at hierarchy 2 were further divided into 12 clusters at hierarchy 3. The 4th cluster at hierarchy 2 having the highest average fitness was divided into 4 clusters. The 7th cluster at hierarchy 2 having the second highest average fitness was divided into 2 clusters. The rest 6 clusters at hierarchy 2 with low average fitness were not further divided. The fourth batch of variants was selected in the 12 clusters at hierarchy 3 (Figure S1C). In all cases, more variants usually were selected in clusters with higher average sample fitness. Deep hierarchy structure can find the cluster with much higher average fitness. As a result, the deepest hierarchical structure generates the sample set with the highest average fitness, where the average fitness is 0.0120, 0.0113, and 0.0143 for $N = 1$, $N = 2$, and $N = 3$, respectively. The expected average fitness from multiple repeats with various hierarchies further supports this point (Table S2).

S4 CLADE using Louvain clustering

We also used Louvain clustering in CLADE. Louvain clustering is a robust community-based clustering method. It is widely used to detect cell heterogeneity in transcriptomic data. Here we incorporated Louvain clustering in CLADE. The Louvain clustering implemented in the Seurat package [8] requires two hyperparameters, *k.param*, and *resolution*. First, a shared nearest neighbor graph is calculated as the neighborhood overlap between every cell with *k.param* nearest neighbors. Then a modularity function is optimized to determine clusters. The parameter *resolution* sets the granularity of the downstream clustering, with increased values leading to a greater number of clusters.

We first performed Louvain clustering with various parameters and the followup cluster-learning sampling (Figure S2C-E). Multiple repeats were performed for a single simulation. In a single simulation, clusters were numbered by unique cluster ID, where cluster ID indicates the descending ranking of the average fitness for all variants within the corresponding cluster. In multiple repeats, expectations of quantities in cluster with identical cluster ID were calculated. Because the number of clusters in multiple repeats under the same parameters may be different, we only showed cluster ID exists for all repeats. Many similar observations have been found as we did in K-means (Figure 2 and Figure S2A-B). Louvain clustering reveals the heterogeneity in the fitness landscape, where the clusters with smaller cluster ID have significantly larger average fitness than the clusters with larger cluster ID, and the followup cluster-learning sampling successfully recapitulated such fitness heterogeneity (Figure S2C). Both expected cluster average fitness and expected number of selected variants in cluster show the polarized distributions (Figure S2C and E). Interestingly, a difference between Louvain and K-means was observed in the plot for the number of variants in clusters. The numbers of variants in clusters have similar sizes by using K-means (Figure S2A). But they fluctuate more by using Louvain (Figure S2D), especially, the first few clusters seem to contain a fewer number of variants than others. As a result, Louvain clustering may be less effective to select high-fitness variants than K-means, where it has lower expected max fitness and expected fitness in training data (Table S2).

Then we performed CLADE by using Louvain clustering and evaluated the outcome. Since Louvain requires two hyperparameters and it is not easy to precisely control the number of clusters, we only performed Louvain in the shallow hierarchy (i.e. $N = 1$). Compared to the CLADE using randomly sampled training data, CLADE with Louvain clustering always has improvement on expected max fitness and expected mean fitness in all explored parameters (Table S1). However, the global maximal fitness hitting rate sometimes can be lower than that using random sampling (e.g. *k.param* and *resolution* $= 0.8$). By picking up the best-performing parameters, CLADE using Louvain reaches almost 2-fold improvement on global maximal fitness hitting rate: 36.4% (Table 1). It can be seen that CLADE using Louvain clustering underperforms CLADE using K-means, and it is more sensitive to the selection of parameters (Table 1). The training data composition from Louvain clustering drew the consistent observations made in Section 2.4. Louvain clustering leads to a slightly reduced sequence diversity and increased fitness diversity in training data comparing to the random sampling. However, their sequence diversity and fitness diversity are generally higher and lower than that using K-means. Indeed, we observed CLADE using Louvain can outperform that using random sampling but underperformed CLADE using K-means in the shallow hierarchy in their best-performing parameters (Table 1).

S5 Supplementary Figures

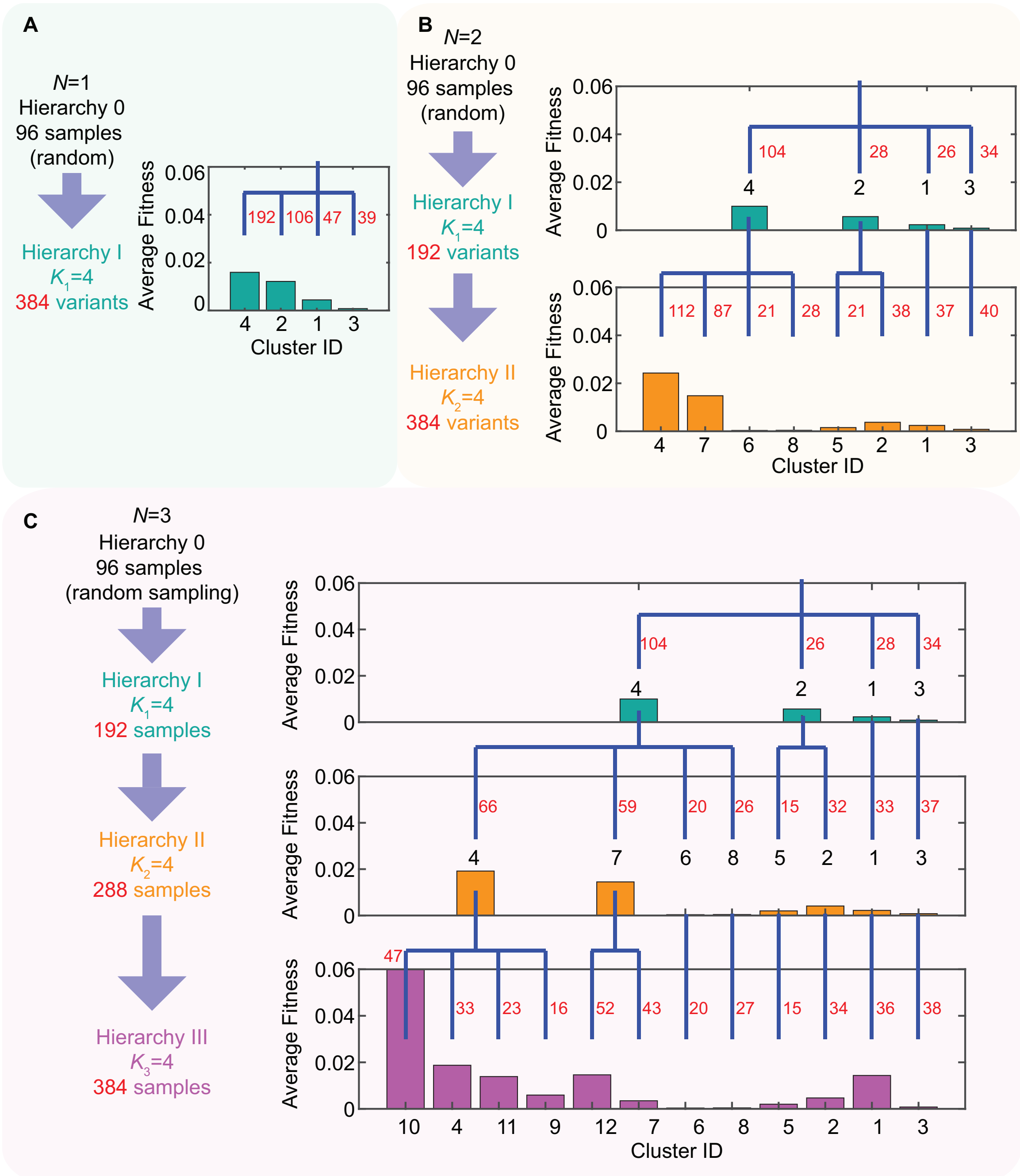

Figure S1: Cluster-learning sampling with various maximum hierarchies. Sampling is simulated in the medium-throughput systems with batch size 96. The first batch was randomly and uniformly sampled in $K_1 = 4$ clusters. Dendrograms show the linkage between clusters in different hierarchies. In each round of clustering, a parent cluster can divide into multiple clusters or remain unchanged depending on the average fitness of existing samples in it. Number with red font in the distal of the dendrogram shows the total number of samples obtained in the corresponding cluster and bar plots show the average fitness of samples obtained in the corresponding cluster before the cluster is further divided. **(A)** Maximum hierarchy $N = 1$; increment of clusters: $K_1 = 4$ clusters; total 4 clusters. **(B)** Maximum hierarchy $N = 2$; increments of clusters: $K_1 = K_2 = 4$; total 8 clusters. **(C)** Maximum hierarchy $N = 3$; increments of clusters: $K_1 = K_2 = K_3 = 4$; total 12 clusters.

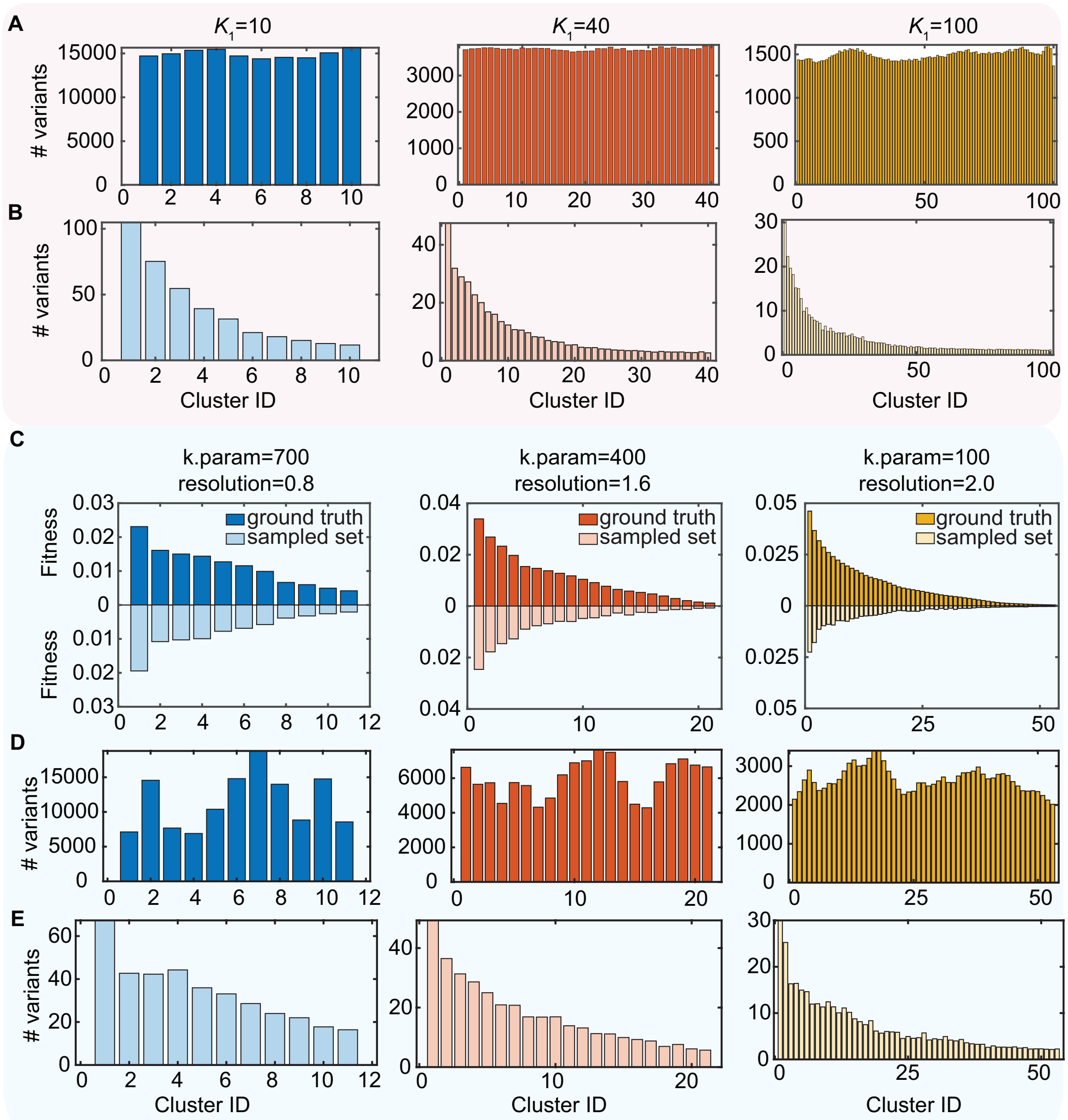

Figure S2: ***K*-means and Louvain clustering reveals fitness heterogeneity and cluster-learning sampling recapitulates the heterogeneity with maximum hierarchy $N = 1$.** GB1 dataset and AA encoding were used. **(A-B)** Supplementary to Figure 2 for K-means clustering with $K_1 = 10$ (blue), 40 (red), and 100 (yellow). **(A)** Bar plots show the expected number of variants in each cluster. **(B)** Bar plots show the expected number of variants selected from the cluster-learning sampling in each cluster. **(C-E)** Louvain clustering and the follow-up cluster-learning sampling are performed on the GB1 dataset with 500 independent repeats. Three sets of parameters are presented individually in different plots: 1) k.param=700, resolution=0.8; 2) k.param=400, resolution=1.6; 3) k.param=100, resolution=2.0. Range of cluster number from multiple repeats in each parameter set: 1) 11∼13; 2) 21∼27; 3) 53∼63. In a single simulation, each cluster is numbered by a unique cluster ID, where cluster ID indicates the descending ranking of the average fitness for all variants within the corresponding cluster. Bar plots only show clusters with cluster ID not greater than the minimum number of clusters obtained from multiple repeats. **(C)** Bar plots above the abscissa with dark color show the expected average ground-truth fitness for all variants contained in each cluster. Bar plots below the abscissa with light color show the expected average fitness for variants selected from the cluster-learning sampling in each cluster. **(D)** Bar plots show the expected number of variants in each cluster. **(E)** Bar plots show the expected number of variants selected from the cluster-learning sampling in each cluster.

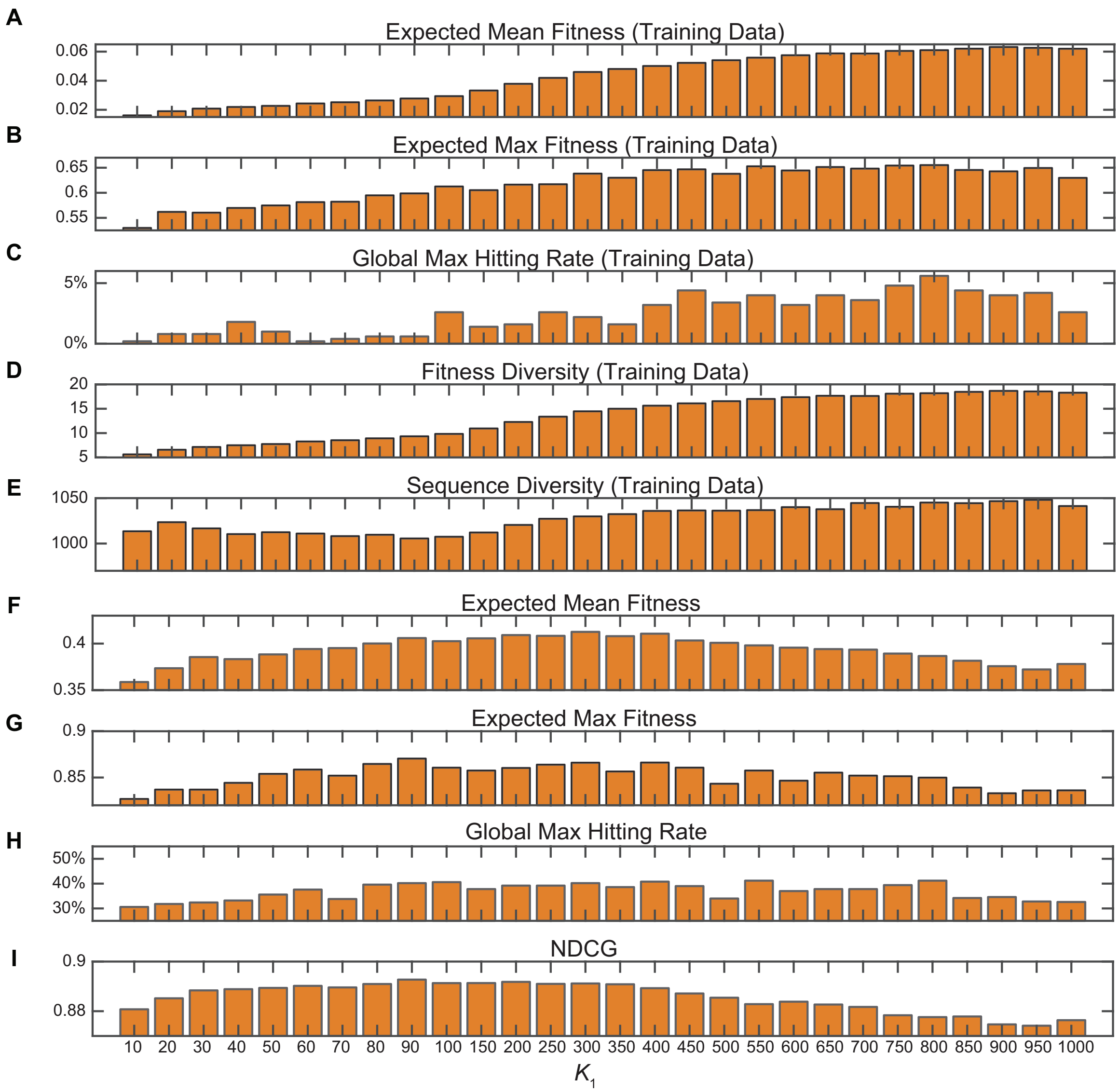

Figure S3: Performance of CLADE on GB1 dataset with maximum hierarchy $N = 1$ by using AA encoding. Various K_1 were explored, and 500 independent repeats were performed for each parameter set. Training data composition were quantified by **(A)** expected mean fitness, **(B)** expected max fitness, **(C)** global max hitting rate, **(D)** fitness diversity measured by MFAD, and **(E)** sequence diversity measured by MFAD. Top-96 predicted variants from CLADE were evaluated by **(F)** expected mean fitness, **(G)** expected max fitness, **(H)** global max hitting rate. The ranking quality of CLADE predictions on all unlabeled variants was evaluated by **(I)** expected NDCG.

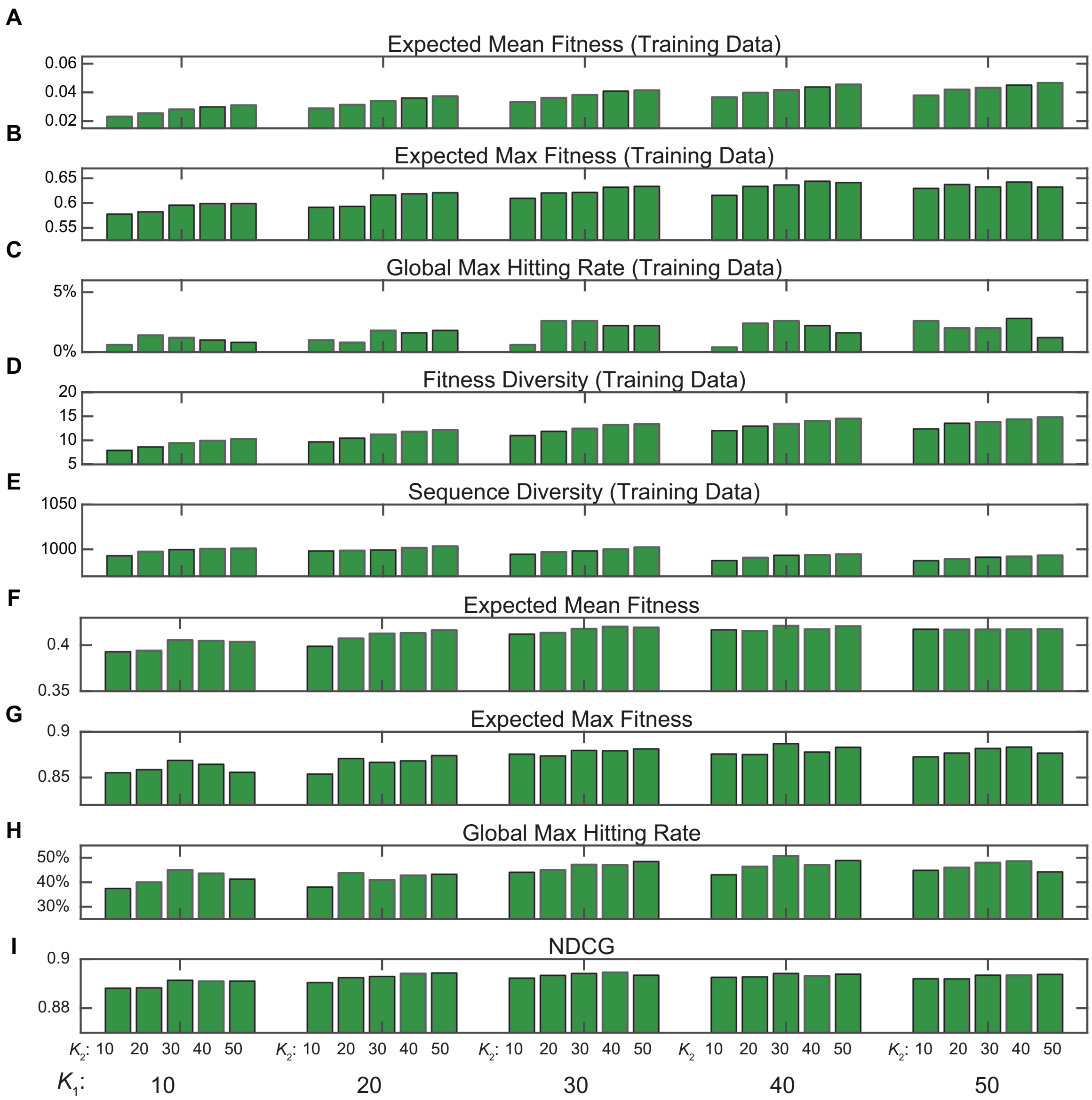

Figure S4: Performance of CLADE on GB1 dataset with maximum hierarchy $N = 2$ by using AA encoding. Various K_1 and K_2 were explored, and 500 independent repeats were performed for each parameter set. Training data composition were quantified by **(A)** expected mean fitness, **(B)** expected max fitness, **(C)** global max hitting rate, **(D)** fitness diversity measured by MFAD, and **(E)** sequence diversity measured by MFAD. Top-96 predicted variants from CLADE were evaluated by **(F)** expected mean fitness, **(G)** expected max fitness, **(H)** global max hitting rate. The ranking quality of CLADE predictions on all unlabeled variants was evaluated by **(I)** expected NDCG.

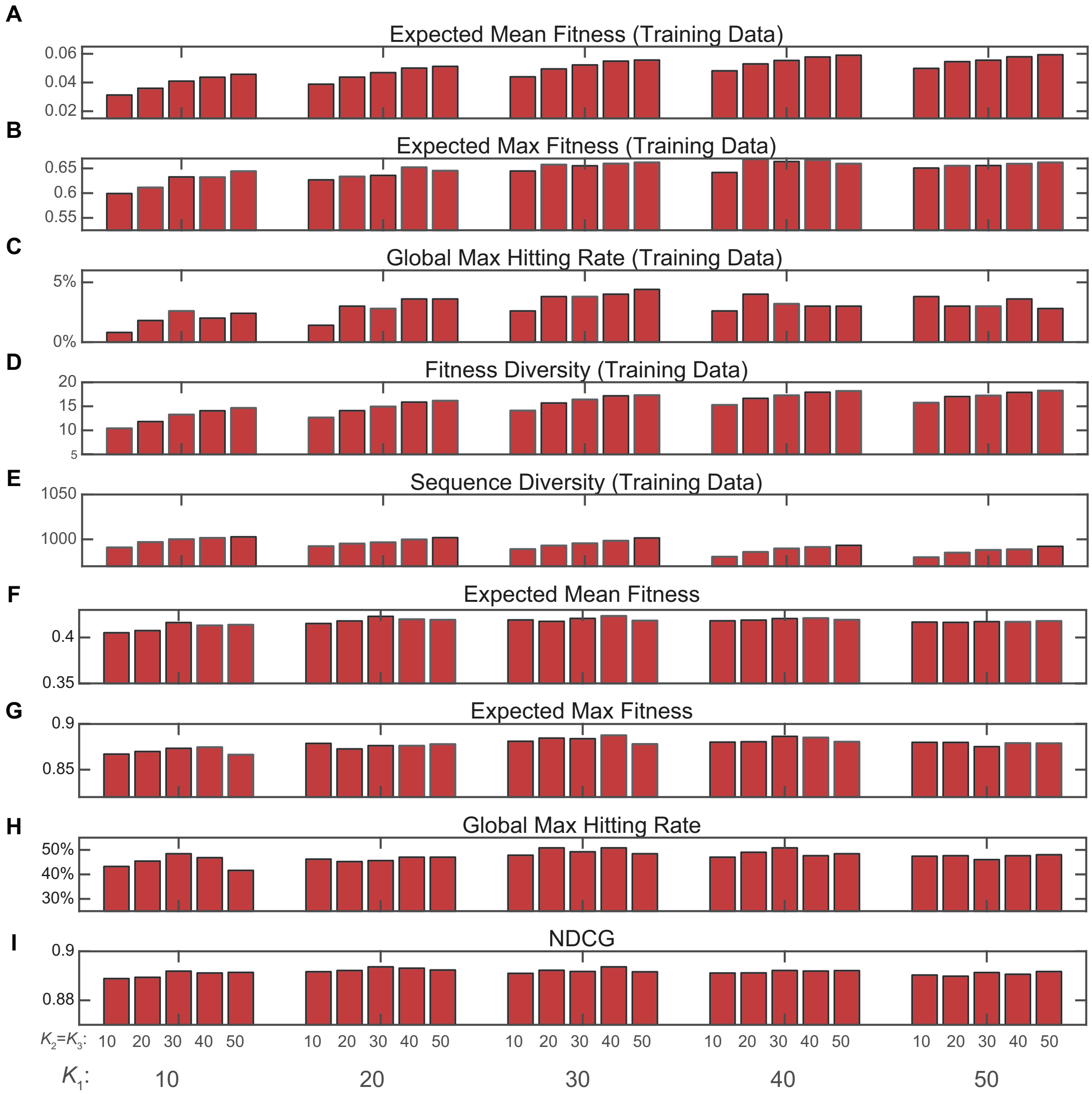

Figure S5: Performance of CLADE on GB1 dataset with maximum hierarchy $N = 3$ by using AA encoding. K_3 was taken to be identical to K_2. Various K_1 and K_2 were explored, and 500 independent repeats were performed for each parameter set. Training data composition were quantified by **(A)** expected mean fitness, **(B)** expected max fitness, **(C)** global max hitting rate, **(D)** fitness diversity measured by MFAD, and **(E)** sequence diversity measured by MFAD. Top-96 predicted variants from CLADE were evaluated by **(F)** expected mean fitness, **(G)** expected max fitness, **(H)** global max hitting rate. The ranking quality of CLADE predictions on all unlabeled variants was evaluated by **(I)** expected NDCG.

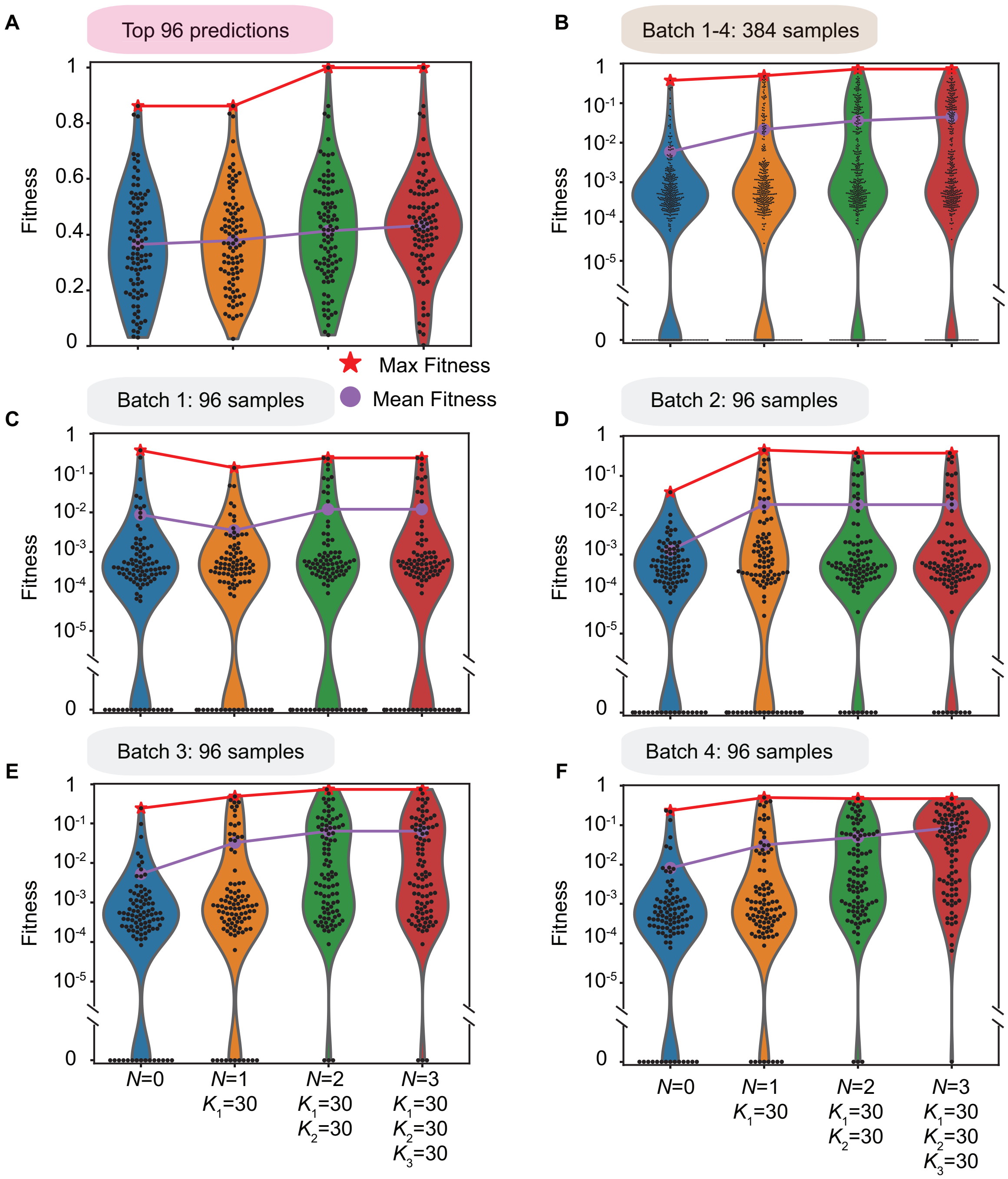

Figure S6: Fitness distributions for variants selected by CLADE. GB1 dataset and AA encoding were used. Single CLADE simulation with various maximum hierarchies: 1) $N = 0$ (random sampling); 2) $N = 1$ ($K_1 = 30$); 3) $N = 2$ ($K_1 = K_2 = 30$); 4) $N = 3$ ($K_1 = K_2 = K_3 = 30$). The violin plot outlines illustrate kernel probability density where the width of the shaded area represents the proportion of the data located there. The black dots show the fitness data. The red line shows the maximum fitness and the purple line shows the mean fitness. Variants selected from different stages are shown: **(A)** top 96 predicted variants; **(B)** entire training data (batch 1-4 with 96 variants); **(C)** first batch of training data; **(D)** second batch of training data; **(E)** third batch of training data; **(F)** fourth batch of training data. Each batch contains 96 variants.

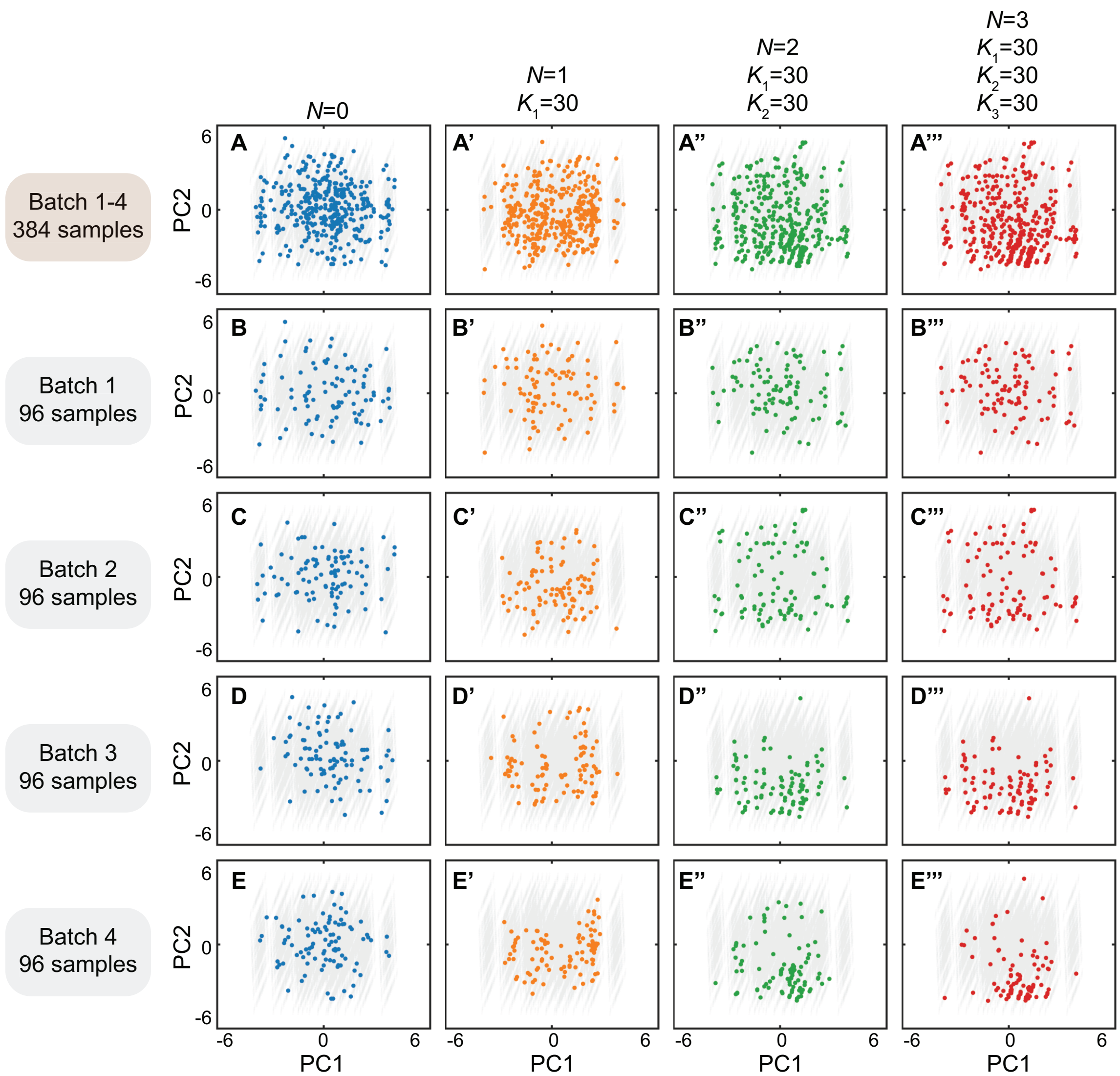

Figure S7: Sequence distributions for variants selected by CLADE. GB1 dataset and AA encoding were used. Selected variants are shown in sequence space in the two-dimensional space spanned by the first two principal components. Gray dots show all variants in the combinatorial library. Distributions of variants selected from **(A-A"')** entire training data (batch 1-4 with 384 variants); **(B-B"')** first batch; **(C-C"')** second batch; **(D-D"')** third batch; **(E-E"')** fourth batch. Different clustering hierarchies are shown: **(A-E)** N=0 with random sampling; **(A'-E')** $N = 1$ with $K_1 = 30$; **(A"-E")** $N = 2$ with $K_1 = K_2 = 30$; **(A"'-E"')** $N = 3$ with $K_1 = K_2 = K_3 = 30$.

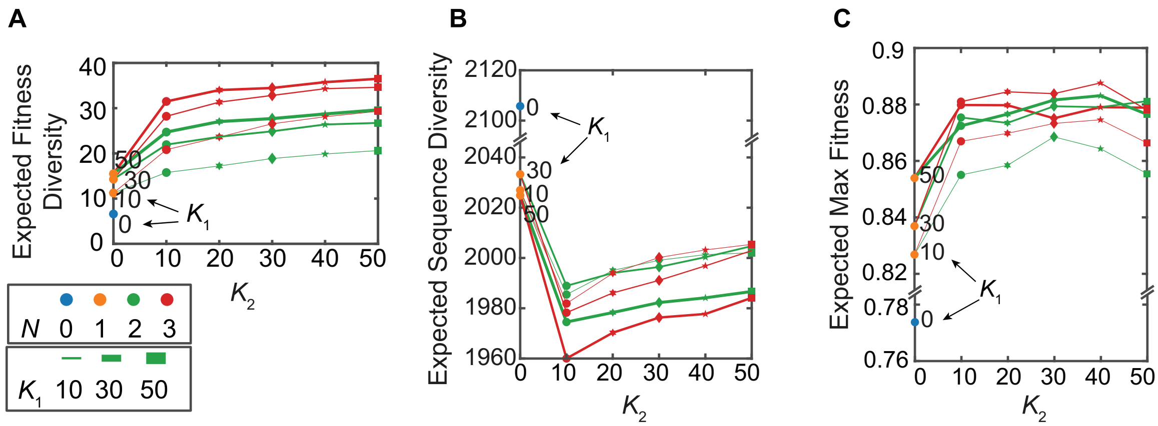

Figure S8: CLADE performance on GB1 by using AA encoding. For each set of hyperparameters, CLADE was repeated independently 500 times. **(A)** Expected values of training data fitness diversity, **(B)** training data sequence diversity, and **(C)** expected maximum fitness from CLADE are plotted against the cluster increment at the second hierarchy (K_2). Each dot represents CLADE results under a clustering architecture. For shallow hierarchy, dots are plotted at $K_2 = 0$.

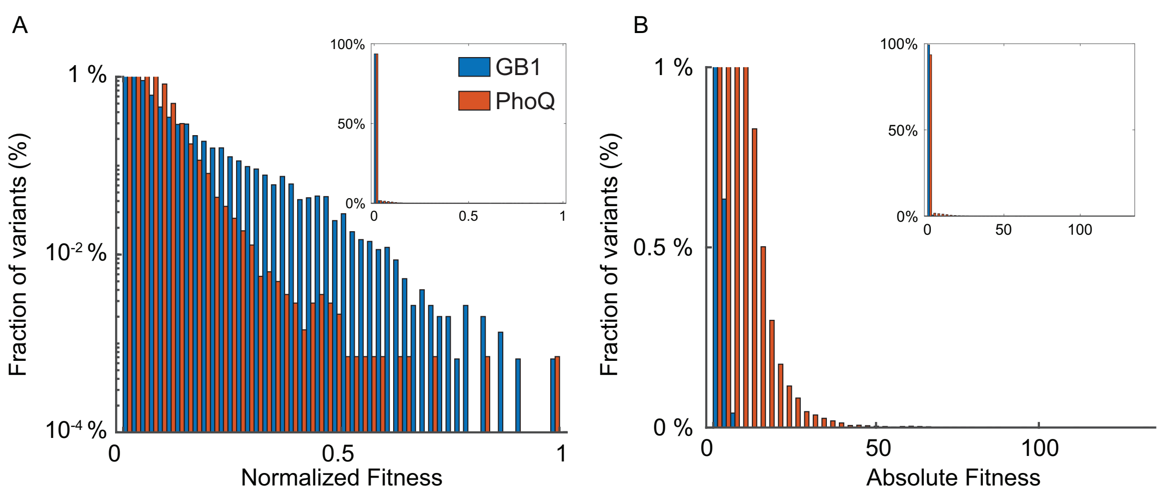

Figure S9: Fitness distributions of GB1 and PhoQ datasets. (A) Normalized fitness. **(B)** Absolute fitness. Absolute fitness for wildtype protein is 1 for both datasets.

S6 Supplementary Tables

CLADE Performance						
GB1; AA; Medium throughput; $N = 0$ (Random sampling); K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
0	0	0.774 (0.151)	0.305 (0.074)	17.80%	18.60%	0.860 (0.021)
GB1; AA; Medium throughput; $N = 1$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
10	0	0.827 (0.142)	0.359 (0.065)	30.40%	30.60%	0.881 (0.019)
20	0	0.842 (0.136)	0.377 (0.062)	32.80%	33.40%	0.886 (0.018)
30	0	0.837 (0.137)	0.386 (0.059)	31.60%	32.40%	0.888 (0.016)
40	0	0.844 (0.133)	0.383 (0.062)	31.40%	33.20%	0.889 (0.018)
50	0	0.854 (0.129)	0.388 (0.060)	34.60%	35.60%	0.889 (0.019)
60	0	0.859 (0.133)	0.394 (0.066)	37.40%	37.60%	0.890 (0.019)
70	0	0.852 (0.129)	0.395 (0.064)	33.40%	33.80%	0.890 (0.020)
80	0	0.865 (0.129)	0.400 (0.063)	39.00%	39.60%	0.891 (0.019)
90	**0**	**0.870 (0.126)**	**0.406 (0.060)**	**39.60%**	**40.20%**	**0.893 (0.017)**
100	0	0.861 (0.132)	0.403 (0.060)	38.00%	40.60%	0.891 (0.018)
150	0	0.858 (0.131)	0.406 (0.064)	36.40%	37.80%	0.891 (0.020)
200	0	0.860 (0.133)	0.409 (0.063)	37.60%	39.20%	0.892 (0.020)
250	0	0.864 (0.127)	0.408 (0.063)	36.60%	39.20%	0.891 (0.020)
300	0	0.866 (0.127)	0.413 (0.062)	38.00%	40.20%	0.891 (0.021)
350	0	0.857 (0.130)	0.408 (0.061)	37.00%	38.60%	0.891 (0.020)

400	0	0.866 (0.127)	0.411 (0.063)	37.40%	40.80%	0.889 (0.021)
450	0	0.861 (0.124)	0.403 (0.062)	34.60%	39.00%	0.887 (0.022)
500	0	0.843 (0.130)	0.401 (0.066)	30.60%	34.00%	0.885 (0.023)
550	0	0.858 (0.132)	0.398 (0.067)	37.20%	41.20%	0.883 (0.023)
600	0	0.847 (0.135)	0.396 (0.069)	33.80%	37.00%	0.884 (0.024)
650	0	0.855 (0.128)	0.394 (0.069)	33.80%	37.80%	0.883 (0.024)
700	0	0.852 (0.133)	0.394 (0.067)	34.20%	37.80%	0.882 (0.025)
750	0	0.851 (0.132)	0.389 (0.070)	34.60%	39.40%	0.878 (0.024)
800	0	0.850 (0.137)	0.387 (0.068)	35.60%	41.20%	0.878 (0.025)
850	0	0.839 (0.134)	0.382 (0.074)	29.80%	34.20%	0.878 (0.026)
900	0	0.833 (0.139)	0.376 (0.073)	30.60%	34.60%	0.875 (0.026)
950	0	0.836 (0.132)	0.372 (0.073)	28.60%	32.80%	0.874 (0.025)
1000	0	0.836 (0.134)	0.378 (0.069)	30.00%	32.60%	0.876 (0.023)
GB1; AA; Medium throughput; $N = 2$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
10	10	0.855 (0.134)	0.393 (0.058)	36.80%	37.40%	0.888 (0.019)
10	20	0.858 (0.137)	0.394 (0.058)	38.60%	40.00%	0.888 (0.020)
10	30	0.869 (0.136)	0.406 (0.058)	43.80%	45.00%	0.891 (0.020)
10	40	0.864 (0.138)	0.405 (0.057)	42.60%	43.60%	0.891 (0.020)
10	50	0.856 (0.142)	0.404 (0.061)	40.40%	41.20%	0.891 (0.020)
20	10	0.854 (0.136)	0.399 (0.063)	37.00%	38.00%	0.890 (0.020)
20	20	0.871 (0.132)	0.407 (0.062)	43.00%	43.80%	0.892 (0.019)

20	30	0.866 (0.129)	0.413 (0.058)	39.20%	41.00%	0.893 (0.019)
20	40	0.868 (0.131)	0.413 (0.059)	41.20%	42.80%	0.894 (0.019)
20	50	0.874 (0.127)	0.416 (0.057)	41.40%	43.20%	0.894 (0.020)
30	10	0.875 (0.129)	0.412 (0.057)	43.40%	44.00%	0.892 (0.020)
30	20	0.873 (0.129)	0.414 (0.059)	42.40%	45.00%	0.893 (0.018)
30	30	0.879 (0.128)	0.418 (0.057)	44.60%	47.20%	0.894 (0.019)
30	40	0.879 (0.129)	0.420 (0.058)	44.80%	47.00%	0.895 (0.020)
30	50	0.881 (0.130)	0.419 (0.059)	46.20%	48.40%	0.893 (0.020)
40	10	0.876 (0.127)	0.417 (0.059)	42.60%	43.00%	0.893 (0.019)
40	20	0.875 (0.131)	0.416 (0.064)	44.00%	46.40%	0.893 (0.020)
40	**30**	**0.887 (0.128)**	**0.421 (0.058)**	**48.20%**	**50.80%**	**0.894 (0.018)**
40	40	0.878 (0.129)	0.417 (0.060)	44.80%	47.00%	0.893 (0.019)
40	50	0.883 (0.130)	0.421 (0.060)	47.20%	48.80%	0.894 (0.020)
50	10	0.872 (0.131)	0.417 (0.064)	42.20%	44.80%	0.892 (0.021)
50	20	0.877 (0.131)	0.417 (0.067)	44.00%	46.00%	0.892 (0.022)
50	30	0.882 (0.128)	0.417 (0.063)	46.00%	48.00%	0.893 (0.020)
50	40	0.883 (0.126)	0.417 (0.062)	45.80%	48.60%	0.893 (0.021)
50	50	0.877 (0.126)	0.418 (0.063)	43.00%	44.20%	0.894 (0.022)
GB1; AA; Medium throughput; $N = 3$; $K_2 = K_3$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
10	10	0.867 (0.134)	0.405 (0.058)	42.40%	43.20%	0.889 (0.019)
10	20	0.870 (0.136)	0.407 (0.057)	43.60%	45.40%	0.889 (0.021)

10	30	0.873 (0.137)	0.416 (0.058)	45.80%	48.40%	0.892 (0.021)
10	40	0.875 (0.134)	0.413 (0.057)	44.80%	46.80%	0.891 (0.021)
10	50	0.866 (0.128)	0.414 (0.056)	39.20%	41.60%	0.891 (0.021)
20	10	0.879 (0.129)	0.415 (0.059)	44.80%	46.20%	0.892 (0.020)
20	20	0.873 (0.129)	0.418 (0.057)	42.20%	45.20%	0.892 (0.020)
20	30	0.876 (0.127)	0.423 (0.060)	42.80%	45.60%	0.894 (0.021)
20	40	0.876 (0.129)	0.420 (0.062)	43.40%	47.00%	0.893 (0.021)
20	50	0.878 (0.126)	0.419 (0.057)	43.40%	47.00%	0.892 (0.020)
30	10	0.881 (0.126)	0.419 (0.057)	45.20%	47.80%	0.891 (0.020)
30	20	0.884 (0.128)	0.417 (0.057)	47.00%	50.80%	0.892 (0.019)
30	30	0.884 (0.125)	0.421 (0.057)	45.40%	49.20%	0.892 (0.021)
30	**40**	**0.888 (0.124)**	**0.423 (0.056)**	**46.80%**	**50.80%**	**0.894 (0.021)**
30	50	0.878 (0.128)	0.418 (0.060)	44.00%	48.40%	0.892 (0.021)
40	10	0.880 (0.127)	0.418 (0.060)	44.40%	47.00%	0.891 (0.021)
40	20	0.880 (0.126)	0.419 (0.059)	45.00%	49.00%	0.891 (0.021)
40	30	0.886 (0.125)	0.421 (0.057)	47.60%	50.80%	0.892 (0.019)
40	40	0.885 (0.122)	0.421 (0.059)	44.60%	47.60%	0.892 (0.019)
40	50	0.881 (0.127)	0.419 (0.060)	45.40%	48.40%	0.892 (0.021)
50	10	0.880 (0.126)	0.417 (0.063)	43.60%	47.40%	0.890 (0.022)
50	20	0.880 (0.129)	0.416 (0.064)	44.60%	47.60%	0.890 (0.022)
50	30	0.875 (0.129)	0.417 (0.061)	43.00%	46.00%	0.891 (0.022)
50	40	0.879 (0.127)	0.417 (0.063)	44.00%	47.60%	0.891 (0.023)
50	50	0.879 (0.130)	0.418 (0.064)	45.20%	48.00%	0.892 (0.023)
GB1; AA; Low throughput;						

$N = 3$; $K_2 = K_3$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
10	10	0.8561 (0.1393)	0.4030 (0.0580)	39.60%	41.00%	0.8890 (0.0197)
10	20	0.8747 (0.1359)	0.4126 (0.0533)	45.60%	46.80%	0.8923 (0.0199)
10	30	0.8724 (0.1350)	0.4167 (0.0567)	44.80%	47.00%	0.8930 (0.0211)
10	40	0.8794 (0.1339)	0.4200 (0.0560)	47.20%	48.60%	0.8946 (0.0196)
10	50	0.8740 (0.1349)	0.4194 (0.0585)	45.60%	48.20%	0.8942 (0.0203)
20	10	0.8734 (0.1274)	0.4188 (0.0573)	41.60%	43.80%	0.8937 (0.0194)
20	20	0.8857 (0.1199)	0.4250 (0.0544)	44.20%	47.40%	0.8955 (0.0187)
20	30	0.8843 (0.1234)	0.4228 (0.0543)	44.80%	47.40%	0.8955 (0.0183)
20	40	0.8781 (0.1246)	0.4284 (0.0542)	42.60%	45.80%	0.8972 (0.0176)
20	50	0.8919 (0.1192)	0.4307 (0.0518)	47.00%	50.40%	0.8975 (0.0183)
30	10	0.9017 (0.1175)	0.4248 (0.0531)	51.20%	55.20%	0.8942 (0.0174)
30	20	0.8974 (0.1199)	0.4261 (0.0503)	51.20%	54.40%	0.8951 (0.0176)
30	30	0.8870 (0.1279)	0.4256 (0.0547)	48.80%	51.80%	0.8952 (0.0187)
30	40	0.8970 (0.1207)	0.4230 (0.0534)	49.80%	55.40%	0.8956 (0.0176)
30	**50**	**0.9043 (0.1161)**	**0.4313 (0.0489)**	**52.00%**	**55.60%**	**0.8959 (0.0176)**
40	10	0.8799 (0.1256)	0.4229 (0.0601)	44.00%	47.20%	0.8924 (0.0195)
40	20	0.8867 (0.1266)	0.4249 (0.0597)	47.60%	50.60%	0.8946 (0.0202)
40	30	0.8805 (0.1264)	0.4232 (0.0582)	44.60%	48.80%	0.8947 (0.0190)
40	40	0.8829 (0.1254)	0.4250 (0.0607)	45.60%	50.00%	0.8950 (0.0193)
40	50	0.8838 (0.1267)	0.4255 (0.0551)	45.80%	49.40%	0.8955 (0.0193)
50	10	0.8879 (0.1224)	0.4221 (0.0581)	46.20%	49.20%	0.8929 (0.0200)

50	20	0.8925 (0.1175)	0.4270 (0.0568)	46.60%	51.80%	0.8951 (0.0187)
50	30	0.8934 (0.1189)	0.4278 (0.0573)	47.40%	51.00%	0.8953 (0.0198)
50	40	0.8846 (0.1231)	0.4285 (0.0578)	44.20%	50.20%	0.8954 (0.0210)
50	50	0.8982 (0.1187)	0.4269 (0.0561)	50.60%	53.40%	0.8960 (0.0192)
GB1; AA; Medium throughput; $N = 1$; Seurat (Louvain)						
k.param	resolution	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
100	0.4	0.795 (0.139)	0.330 (0.066)	19.20%	19.60%	0.865 (0.021)
100	0.8	0.786 (0.137)	0.332 (0.064)	17.80%	18.00%	0.868 (0.021)
100	1.2	0.804 (0.137)	0.350 (0.066)	22.00%	22.20%	0.872 (0.022)
100	1.6	0.810 (0.135)	0.355 (0.067)	21.20%	21.80%	0.872 (0.022)
100	2	0.808 (0.134)	0.349 (0.069)	21.60%	22.40%	0.870 (0.022)
200	0.4	0.791 (0.140)	0.327 (0.069)	20.80%	21.00%	0.860 (0.023)
200	0.8	0.788 (0.133)	0.327 (0.066)	16.00%	16.20%	0.864 (0.022)
200	1.2	0.806 (0.139)	0.337 (0.069)	22.40%	23.20%	0.865 (0.023)
200	1.6	0.809 (0.121)	0.350 (0.070)	16.80%	17.00%	0.868 (0.022)
200	2	0.818 (0.130)	0.355 (0.067)	22.40%	22.80%	0.870 (0.021)
300	0.4	0.792 (0.140)	0.329 (0.070)	20.00%	20.20%	0.861 (0.022)
300	0.8	0.804 (0.134)	0.334 (0.065)	20.20%	20.80%	0.866 (0.022)
300	1.2	0.826 (0.140)	0.352 (0.070)	28.80%	29.40%	0.871 (0.022)
300	1.6	0.810 (0.130)	0.347 (0.071)	20.60%	21.20%	0.869 (0.022)
300	2	0.817 (0.121)	0.361 (0.064)	19.20%	20.00%	0.873 (0.021)
400	0.4	0.790 (0.139)	0.322 (0.069)	20.60%	20.60%	0.858 (0.023)

400	0.8	0.810 (0.136)	0.331 (0.066)	23.60%	23.80%	0.865 (0.022)
400	1.2	0.829 (0.139)	0.352 (0.067)	28.80%	29.00%	0.872 (0.020)
400	1.6	0.825 (0.134)	0.356 (0.066)	26.00%	27.20%	0.873 (0.021)
400	2	0.816 (0.129)	0.355 (0.066)	21.20%	21.60%	0.873 (0.021)
500	0.4	0.799 (0.149)	0.324 (0.076)	25.60%	26.20%	0.860 (0.023)
500	0.8	0.799 (0.133)	0.331 (0.066)	20.00%	20.20%	0.866 (0.022)
500	**1.2**	**0.846 (0.137)**	**0.357 (0.070)**	**34.80%**	**36.40%**	**0.872 (0.021)**
500	1.6	0.832 (0.131)	0.360 (0.066)	27.80%	29.00%	0.875 (0.020)
500	2	0.828 (0.132)	0.358 (0.066)	25.80%	26.60%	0.872 (0.022)
600	0.4	0.801 (0.147)	0.326 (0.072)	26.20%	26.40%	0.860 (0.023)
600	0.8	0.797 (0.133)	0.332 (0.065)	18.60%	19.20%	0.864 (0.024)
600	1.2	0.844 (0.141)	0.357 (0.073)	35.20%	36.40%	0.872 (0.020)
600	1.6	0.840 (0.128)	0.366 (0.069)	29.40%	29.80%	0.878 (0.020)
600	2	0.837 (0.135)	0.360 (0.067)	30.60%	31.60%	0.875 (0.020)
700	0.4	0.816 (0.145)	0.329 (0.075)	28.40%	29.60%	0.862 (0.021)
700	0.8	0.802 (0.137)	0.332 (0.067)	22.20%	22.40%	0.864 (0.023)
700	1.2	0.845 (0.142)	0.353 (0.071)	35.40%	36.80%	0.871 (0.020)
700	1.6	0.845 (0.127)	0.368 (0.068)	29.80%	30.20%	0.877 (0.020)
700	2	0.835 (0.139)	0.359 (0.069)	31.80%	33.00%	0.877 (0.021)
GB1; Georgiev; Medium throughput; $N = 0$ (Random sampling); K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
0	0	0.7471 (0.1356)	0.2719 (0.0766)	7.40%	8.20%	0.8327 (0.0263)
GB1; Georgiev; Medium throughput;						

$N = 3$; $K_2 = K_3$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
10	10	0.8184 (0.1151)	0.3430 (0.0946)	15.60%	17.20%	0.8402 (0.0292)
10	20	0.8302 (0.1171)	0.3535 (0.0928)	20.80%	22.80%	0.8425 (0.0275)
10	30	0.8260 (0.1194)	0.3442 (0.0930)	19.00%	22.80%	0.8390 (0.0282)
10	40	0.8325 (0.1220)	0.3458 (0.0969)	21.80%	24.20%	0.8407 (0.0296)
10	50	0.8378 (0.1111)	0.3501 (0.0914)	20.20%	23.40%	0.8410 (0.0286)
20	10	0.8315 (0.1088)	0.3498 (0.0861)	16.80%	18.60%	0.8434 (0.0270)
20	20	0.8286 (0.1120)	0.3492 (0.0859)	17.20%	19.60%	0.8442 (0.0278)
20	30	0.8308 (0.1187)	0.3467 (0.0861)	20.40%	23.80%	0.8440 (0.0270)
20	40	0.8386 (0.1126)	0.3529 (0.0868)	20.80%	24.00%	0.8452 (0.0279)
20	50	0.8430 (0.1136)	0.3599 (0.0875)	24.40%	26.60%	0.8472 (0.0280)
30	10	0.8436 (0.1048)	0.3609 (0.0774)	19.80%	22.60%	0.8464 (0.0279)
30	20	0.8500 (0.1075)	0.3610 (0.0830)	22.20%	25.80%	0.8468 (0.0284)
30	30	0.8497 (0.1052)	0.3605 (0.0800)	22.60%	25.60%	0.8462 (0.0289)
30	**40**	**0.8595 (0.1056)**	**0.3674 (0.0799)**	**26.00%**	**30.60%**	**0.8493 (0.0284)**
30	50	0.8584 (0.1062)	0.3713 (0.0802)	25.40%	28.80%	0.8510 (0.0289)
40	10	0.8367 (0.1188)	0.3539 (0.0798)	21.40%	24.40%	0.8480 (0.0277)
40	20	0.8430 (0.1176)	0.3497 (0.0806)	24.40%	27.00%	0.8456 (0.0294)
40	30	0.8461 (0.1111)	0.3537 (0.0817)	22.60%	26.20%	0.8485 (0.0295)
40	40	0.8512 (0.1084)	0.3538 (0.0823)	24.20%	28.20%	0.8493 (0.0291)
40	50	0.8492 (0.1090)	0.3583 (0.0814)	22.60%	25.60%	0.8504 (0.0291)
50	10	0.8292 (0.1184)	0.3369 (0.0845)	19.20%	22.60%	0.8452 (0.0287)

50	20	0.8399 (0.1157)	0.3493 (0.0814)	21.80%	24.60%	0.8474 (0.0299)
50	30	0.8400 (0.1137)	0.3488 (0.0858)	21.20%	24.40%	0.8490 (0.0287)
50	40	0.8477 (0.1135)	0.3523 (0.0866)	23.20%	27.40%	0.8498 (0.0296)
50	50	0.8372 (0.1173)	0.3548 (0.0843)	21.40%	24.80%	0.8507 (0.0284)
PhoQ; AA; Medium throughput; $N = 0$ (Random sampling); K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
0	0	0.2985 (0.1321)	0.0719 (0.0205)	1.00%	1.00%	0.7859 (0.0189)
PhoQ; AA; Medium throughput; $N = 3$; $K_2 = K_3$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
10	10	0.3476 (0.1541)	0.0915 (0.0149)	2.60%	2.60%	0.8060 (0.0147)
10	20	0.3415 (0.1234)	0.0935 (0.0144)	0.80%	0.80%	0.8071 (0.0142)
10	30	0.3573 (0.1404)	0.0934 (0.0146)	1.40%	1.60%	0.8071 (0.0151)
10	40	0.3429 (0.1390)	0.0925 (0.0151)	1.80%	2.20%	0.8073 (0.0143)
10	50	0.3479 (0.1377)	0.0932 (0.0148)	1.40%	2.00%	0.8062 (0.0155)
20	10	0.3278 (0.1298)	0.0912 (0.0163)	1.40%	1.40%	0.8048 (0.0156)
20	20	0.3441 (0.1334)	0.0923 (0.0166)	1.20%	1.40%	0.8050 (0.0149)
20	30	0.3449 (0.1418)	0.0926 (0.0142)	1.80%	1.80%	0.8051 (0.0143)
20	40	0.3404 (0.1355)	0.0923 (0.0147)	1.40%	1.60%	0.8054 (0.0149)
20	50	0.3366 (0.1289)	0.0912 (0.0160)	1.40%	1.60%	0.8051 (0.0158)
30	10	0.3325 (0.1237)	0.0917 (0.0161)	1.00%	1.20%	0.8040 (0.0164)

30	20	0.3417 (0.1393)	0.0930 (0.0150)	1.60%	1.80%	0.8049 (0.0157)
30	30	0.3414 (0.1413)	0.0917 (0.0161)	1.80%	2.00%	0.8036 (0.0173)
30	40	0.3488 (0.1273)	0.0925 (0.0154)	1.00%	1.60%	0.8043 (0.0156)
30	50	0.3514 (0.1509)	0.0939 (0.0159)	2.60%	3.00%	0.8048 (0.0163)
40	10	0.3528 (0.1479)	0.0926 (0.0178)	2.20%	2.60%	0.8037 (0.0194)
40	20	0.3533 (0.1479)	0.0925 (0.0173)	2.00%	2.00%	0.8038 (0.0193)
40	**30**	**0.3572 (0.1579)**	**0.0934 (0.0160)**	**2.80%**	**3.00%**	**0.8047 (0.0153)**
40	40	0.3533 (0.1471)	0.0935 (0.0163)	2.00%	2.00%	0.8045 (0.0170)
40	50	0.3528 (0.1434)	0.0930 (0.0164)	2.00%	2.20%	0.8045 (0.0177)
50	10	0.3284 (0.1193)	0.0930 (0.0160)	0.80%	1.20%	0.8039 (0.0171)
50	20	0.3481 (0.1475)	0.0934 (0.0159)	2.20%	2.80%	0.8037 (0.0173)
50	30	0.3351 (0.1310)	0.0930 (0.0167)	1.20%	1.80%	0.8034 (0.0177)
50	40	0.3398 (0.1393)	0.0932 (0.0161)	1.80%	2.00%	0.8031 (0.0183)
50	50	0.3365 (0.1226)	0.0930 (0.0161)	0.80%	1.20%	0.8038 (0.0167)
PhoQ; Georgiev; Medium throughput; $N = 0$ (Random sampling); K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
0	0	0.3705 (0.2116)	0.0769 (0.0220)	7.20%	7.60%	0.7873 (0.0193)
PhoQ; Georgiev; Medium throughput; $N = 3$; $K_2 = K_3$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
10	10	0.4700 (0.2363)	0.0959 (0.0221)	13.60%	15.40%	0.7928 (0.0225)

10	20	0.4842 (0.2447)	0.0967 (0.0216)	15.20%	17.80%	0.7931 (0.0222)
10	**30**	**0.5028 (0.2538)**	**0.0960 (0.0230)**	**17.60%**	**20.60%**	**0.7919 (0.0232)**
10	40	0.4963 (0.2588)	0.0963 (0.0228)	17.60%	20.20%	0.7921 (0.0234)
10	50	0.4818 (0.2521)	0.0970 (0.0224)	16.20%	21.60%	0.7929 (0.0226)
20	10	0.4684 (0.2386)	0.0961 (0.0207)	13.40%	15.00%	0.7943 (0.0207)
20	20	0.4629 (0.2329)	0.0947 (0.0212)	12.80%	14.80%	0.7943 (0.0215)
20	30	0.4806 (0.2426)	0.0945 (0.0226)	15.00%	17.80%	0.7942 (0.0221)
20	40	0.4682 (0.2348)	0.0950 (0.0199)	13.40%	16.60%	0.7942 (0.0217)
20	50	0.4723 (0.2371)	0.0943 (0.0225)	13.40%	17.60%	0.7945 (0.0220)
30	10	0.4669 (0.2384)	0.0925 (0.0227)	13.20%	14.80%	0.7937 (0.0222)
30	20	0.4563 (0.2389)	0.0925 (0.0213)	13.20%	16.00%	0.7944 (0.0227)
30	30	0.4732 (0.2502)	0.0924 (0.0230)	15.40%	18.80%	0.7926 (0.0223)
30	40	0.4535 (0.2350)	0.0947 (0.0197)	12.60%	15.20%	0.7947 (0.0211)
30	50	0.4716 (0.2449)	0.0933 (0.0223)	14.40%	17.80%	0.7947 (0.0217)
40	10	0.4492 (0.2368)	0.0905 (0.0213)	12.40%	14.00%	0.7926 (0.0226)
40	20	0.4609 (0.2456)	0.0921 (0.0208)	14.00%	16.00%	0.7935 (0.0226)
40	30	0.4550 (0.2354)	0.0918 (0.0213)	12.60%	14.40%	0.7940 (0.0224)
40	40	0.4658 (0.2449)	0.0915 (0.0223)	14.80%	18.00%	0.7932 (0.0225)
40	50	0.4693 (0.2482)	0.0928 (0.0206)	15.40%	18.00%	0.7952 (0.0214)
50	10	0.4728 (0.2434)	0.0912 (0.0223)	14.40%	16.40%	0.7923 (0.0231)
50	20	0.4615 (0.2333)	0.0914 (0.0218)	12.60%	15.40%	0.7930 (0.0221)
50	30	0.4675 (0.2413)	0.0933 (0.0208)	14.00%	17.00%	0.7941 (0.0226)
50	40	0.4761 (0.2559)	0.0933 (0.0213)	16.20%	20.60%	0.7945 (0.0212)

50	50	0.4639 (0.2445)	0.0937 (0.0217)	14.40%	17.80%	0.7949 (0.0232)
PhoQ; Georgiev; Low throughput; $N = 3$; $K_2 = K_3$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate (Top 96 predictions)	Global Max Hitting Rate (Top 96 predictions & training data)	NDCG
10	10	0.4844 (0.2480)	0.0978 (0.0215)	15.60%	17.60%	0.7935 (0.0226)
10	20	0.4810 (0.2499)	0.0981 (0.0208)	16.00%	18.20%	0.7939 (0.0222)
10	30	0.4928 (0.2517)	0.0967 (0.0217)	16.60%	20.80%	0.7930 (0.0235)
10	**40**	**0.5005 (0.2543)**	**0.0985 (0.0222)**	**17.60%**	**21.00%**	**0.7937 (0.0222)**
10	50	0.4835 (0.2443)	0.0979 (0.0215)	15.60%	20.20%	0.7940 (0.0222)
20	10	0.4754 (0.2448)	0.0956 (0.0214)	14.40%	16.40%	0.7949 (0.0209)
20	20	0.4593 (0.2217)	0.0952 (0.0213)	11.60%	16.40%	0.7937 (0.0211)
20	30	0.4988 (0.2470)	0.0967 (0.0214)	16.40%	20.20%	0.7956 (0.0208)
20	40	0.4852 (0.2426)	0.0969 (0.0218)	15.00%	19.00%	0.7960 (0.0218)
20	50	0.4796 (0.2458)	0.0977 (0.0204)	15.20%	19.40%	0.7963 (0.0206)
30	10	0.4600 (0.2355)	0.0945 (0.0201)	13.00%	15.20%	0.7952 (0.0216)
30	20	0.4648 (0.2353)	0.0937 (0.0211)	12.80%	16.20%	0.7951 (0.0216)
30	30	0.4736 (0.2452)	0.0963 (0.0206)	14.80%	16.40%	0.7968 (0.0206)
30	40	0.4485 (0.2225)	0.0947 (0.0202)	11.00%	14.80%	0.7965 (0.0212)
30	50	0.4759 (0.2467)	0.0967 (0.0203)	15.20%	18.60%	0.7976 (0.0211)
40	10	0.4730 (0.2487)	0.0948 (0.0202)	15.40%	17.40%	0.7954 (0.0216)
40	20	0.4464 (0.2336)	0.0930 (0.0213)	12.40%	14.40%	0.7949 (0.0220)
40	30	0.4726 (0.2475)	0.0946 (0.0203)	15.20%	17.80%	0.7954 (0.0218)
40	40	0.4701 (0.2472)	0.0956 (0.0201)	15.20%	18.20%	0.7967 (0.0213)

40	50	0.4805 (0.2578)	0.0963 (0.0197)	17.00%	19.80%	0.7970 (0.0210)
50	10	0.4747 (0.2515)	0.0928 (0.0224)	15.80%	17.80%	0.7930 (0.0212)
50	20	0.4788 (0.2414)	0.0921 (0.0235)	14.60%	18.00%	0.7925 (0.0231)
50	30	0.4667 (0.2365)	0.0935 (0.0220)	13.20%	17.40%	0.7941 (0.0235)
50	40	0.5022 (0.2583)	0.0955 (0.0215)	18.20%	21.40%	0.7942 (0.0222)
50	50	0.4913 (0.2449)	0.0959 (0.0212)	15.60%	20.40%	0.7956 (0.0220)

Table S1: Predictive performance of CLADE. Two datasets, GB1 and PhoQ, were tested. Two encoding methods, AA and Georgiev, were used. Medium- and low-throughput systems were simulated. Two clustering methods, K-means and Louvain, were used. 384 variants training data and 96 top predicted variants were evaluated. All metrics were evaluated on 500 independent repeats. Both expectations and standard deviations (in parenthesis) were shown for max fitness, mean fitness, and NDCG. Max fitness and mean fitness were evaluated on the top 96 predicted variants. NDCG was evaluated on all predicted variants. Global maximum hitting rate in top 96 predictions, and variants from top 96 predictions and training data were shown. In each architecture, the case with best expected max fitness was highlighted by bold font.

Training Data						
GB1; AA; Medium throughput; $N = 0$ (Random sampling); K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate	MFAD (Sequence)	MFAD (Fitness)
0	0	0.4753 (0.1352)	0.0092 (0.0023)	0.80%	2.11×10^3 (21.67)	6.60 (1.68)
GB1; AA; Medium throughput; $N = 1$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate	MFAD (Sequence)	MFAD (Fitness)
10	0	0.5298 (0.1264)	0.0161 (0.0042)	0.20%	2.03×10^3 (40.14)	11.25 (2.85)
20	0	0.5591 (0.1213)	0.0190 (0.0047)	0.60%	2.05×10^3 (53.68)	13.15 (3.13)
30	0	0.5603 (0.1220)	0.0208 (0.0051)	0.80%	2.03×10^3 (57.91)	14.29 (3.38)
40	0	0.5697 (0.1285)	0.0219 (0.0058)	1.80%	2.02×10^3 (61.15)	15.00 (3.84)
50	0	0.5747 (0.1189)	0.0227 (0.0062)	1.00%	2.02×10^3 (66.61)	15.50 (4.05)
60	0	0.5813 (0.1261)	0.0243 (0.0071)	0.20%	2.02×10^3 (69.19)	16.55 (4.64)
70	0	0.5821 (0.1133)	0.0252 (0.0074)	0.40%	2.02×10^3 (66.06)	17.06 (4.76)
80	0	0.5948 (0.1208)	0.0264 (0.0078)	0.60%	2.02×10^3 (71.76)	17.85 (4.99)
90	**0**	**0.5987 (0.1183)**	**0.0277 (0.0080)**	**0.60%**	$\mathbf{2.01\times10^3}$ **(71.85)**	**18.68 (5.12)**
100	0	0.6125 (0.1333)	0.0294 (0.0091)	2.60%	2.01×10^3 (74.86)	19.66 (5.77)
150	0	0.6052 (0.1247)	0.0332 (0.0105)	1.40%	2.02×10^3 (72.17)	21.89 (6.47)
200	0	0.6162 (0.1228)	0.0378 (0.0115)	1.60%	2.04×10^3 (70.34)	24.55 (6.84)
250	0	0.6170 (0.1305)	0.0419 (0.0138)	2.60%	2.05×10^3 (71.02)	26.73 (7.99)
300	0	0.6382 (0.1244)	0.0460 (0.0147)	2.20%	2.06×10^3 (75.49)	28.94 (8.28)
350	0	0.6299 (0.1318)	0.0480 (0.0163)	1.60%	2.06×10^3 (76.98)	29.99 (9.21)
400	0	0.6452 (0.1306)	0.0501 (0.0154)	3.20%	2.07×10^3 (81.56)	31.24 (8.62)

450	0	0.6467 (0.1406)	0.0523 (0.0179)	4.40%	2.07×10^3 (82.47)	32.19 (9.87)
500	0	0.6378 (0.1319)	0.0541 (0.0183)	3.40%	2.07×10^3 (88.07)	33.09 (10.04)
550	0	0.6528 (0.1406)	0.0558 (0.0187)	4.00%	2.07×10^3 (85.10)	34.00 (10.25)
600	0	0.6444 (0.1291)	0.0575 (0.0183)	3.20%	2.08×10^3 (89.13)	34.74 (9.84)
650	0	0.6513 (0.1365)	0.0588 (0.0188)	4.00%	2.08×10^3 (83.06)	35.30 (9.98)
700	0	0.6482 (0.1336)	0.0587 (0.0178)	3.60%	2.09×10^3 (89.36)	35.22 (9.65)
750	0	0.6543 (0.1411)	0.0605 (0.0197)	4.80%	2.08×10^3 (97.92)	36.17 (10.56)
800	0	0.6552 (0.1426)	0.0610 (0.0204)	5.60%	2.09×10^3 (92.29)	36.36 (10.84)
850	0	0.6454 (0.1354)	0.0620 (0.0194)	4.40%	2.09×10^3 (93.34)	36.92 (10.37)
900	0	0.6427 (0.1366)	0.0632 (0.0216)	4.00%	2.09×10^3 (87.00)	37.31 (11.39)
950	0	0.6494 (0.1363)	0.0626 (0.0192)	4.20%	2.10×10^3 (97.52)	37.09 (10.21)
1000	0	0.6297 (0.1239)	0.0620 (0.0197)	2.60%	2.08×10^3 (96.69)	36.57 (10.38)
GB1; AA; Medium throughput; $N = 2$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate	MFAD (Sequence)	MFAD (Fitness)
10	10	0.5773 (0.1178)	0.0231 (0.0060)	0.60%	1.99×10^3 (50.88)	15.77 (3.90)
10	20	0.5820 (0.1235)	0.0255 (0.0070)	1.40%	2.00×10^3 (56.12)	17.22 (4.47)
10	30	0.5954 (0.1206)	0.0281 (0.0076)	1.20%	2.00×10^3 (57.47)	18.87 (4.76)
10	40	0.5985 (0.1126)	0.0298 (0.0086)	1.00%	2.00×10^3 (58.60)	19.87 (5.34)
10	50	0.5986 (0.1191)	0.0311 (0.0089)	0.80%	2.00×10^3 (58.03)	20.62 (5.56)
20	10	0.5911 (0.1278)	0.0288 (0.0078)	1.00%	2.00×10^3 (63.24)	19.30 (4.93)
20	20	0.5929 (0.1173)	0.0314 (0.0086)	0.80%	2.00×10^3 (64.21)	20.83 (5.35)
20	30	0.6161 (0.1244)	0.0340 (0.0095)	1.80%	2.00×10^3 (62.83)	22.43 (5.81)

20	40	0.6184 (0.1229)	0.0360 (0.0103)	1.60%	2.00×10^3 (64.47)	23.63 (6.25)
20	50	0.6206 (0.1232)	0.0373 (0.0104)	1.80%	2.01×10^3 (65.95)	24.36 (6.33)
30	10	0.6093 (0.1163)	0.0332 (0.0098)	0.60%	1.99×10^3 (72.11)	21.93 (5.97)
30	20	0.6201 (0.1286)	0.0362 (0.0105)	2.60%	1.99×10^3 (72.81)	23.67 (6.33)
30	30	0.6212 (0.1297)	0.0382 (0.0113)	2.60%	2.00×10^3 (75.72)	24.88 (6.77)
30	40	0.6318 (0.1245)	0.0408 (0.0115)	2.20%	2.00×10^3 (78.03)	26.39 (6.82)
30	50	0.6334 (0.1254)	0.0415 (0.0120)	2.20%	2.00×10^3 (75.39)	26.73 (7.12)
40	10	0.6153 (0.1194)	0.0366 (0.0110)	0.40%	1.97×10^3 (74.27)	23.97 (6.60)
40	20	0.6334 (0.1331)	0.0398 (0.0120)	2.40%	1.98×10^3 (78.91)	25.85 (7.22)
40	**30**	**0.6363 (0.1293)**	**0.0417 (0.0115)**	**2.60%**	$\mathbf{1.99\times10^3}$ **(79.23)**	**26.91 (6.85)**
40	40	0.6438 (0.1196)	0.0437 (0.0119)	2.20%	1.99×10^3 (82.21)	28.06 (7.05)
40	50	0.6410 (0.1276)	0.0455 (0.0128)	1.60%	1.99×10^3 (80.42)	29.03 (7.58)
50	10	0.6293 (0.1360)	0.0379 (0.0115)	2.60%	1.97×10^3 (74.99)	24.71 (6.99)
50	20	0.6371 (0.1314)	0.0419 (0.0127)	2.00%	1.98×10^3 (77.65)	27.04 (7.55)
50	30	0.6325 (0.1310)	0.0433 (0.0125)	2.00%	1.98×10^3 (79.46)	27.70 (7.44)
50	40	0.6422 (0.1347)	0.0450 (0.0127)	2.80%	1.98×10^3 (81.68)	28.72 (7.46)
50	50	0.6322 (0.1248)	0.0466 (0.0129)	1.20%	1.99×10^3 (83.14)	29.62 (7.54)
GB1; AA; Medium throughput; $N = 3$; $K_2 = K_3$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate	MFAD (Sequence)	MFAD (Fitness)
10	10	0.5990 (0.1134)	0.0312 (0.0084)	0.80%	1.98×10^3 (56.78)	20.82 (5.25)
10	20	0.6113 (0.1240)	0.0359 (0.0102)	1.80%	1.99×10^3 (61.50)	23.61 (6.26)
10	30	0.6326 (0.1239)	0.0409 (0.0113)	2.60%	2.00×10^3 (65.27)	26.55 (6.75)

10	40	0.6322 (0.1232)	0.0436 (0.0119)	2.00%	2.00×10^3 (65.61)	28.11 (6.97)
10	50	0.6443 (0.1291)	0.0457 (0.0121)	2.40%	2.01×10^3 (64.03)	29.31 (7.12)
20	10	0.6267 (0.1275)	0.0388 (0.0101)	1.40%	1.98×10^3 (66.98)	25.31 (6.19)
20	20	0.6335 (0.1259)	0.0437 (0.0111)	3.00%	1.99×10^3 (68.44)	28.16 (6.68)
20	30	0.6357 (0.1268)	0.0468 (0.0120)	2.80%	1.99×10^3 (68.85)	29.89 (7.04)
20	40	0.6522 (0.1316)	0.0500 (0.0130)	3.60%	2.00×10^3 (70.83)	31.71 (7.55)
20	50	0.6453 (0.1293)	0.0512 (0.0130)	3.60%	2.00×10^3 (71.65)	32.30 (7.61)
30	10	0.6446 (0.1282)	0.0439 (0.0128)	2.60%	1.98×10^3 (74.93)	28.21 (7.58)
30	20	0.6576 (0.1306)	0.0494 (0.0134)	3.80%	1.99×10^3 (78.17)	31.33 (7.77)
30	30	0.6551 (0.1333)	0.0522 (0.0141)	3.80%	995.55 (40.33)	16.41 (4.04)
30	**40**	**0.6598 (0.1297)**	**0.0549 (0.0139)**	**4.00%**	$\mathbf{2.00\times10^3}$ **(82.32)**	**34.31 (7.94)**)
30	50	0.6621 (0.1310)	0.0556 (0.0145)	4.40%	2.00×10^3 (81.72)	34.61 (8.24)
40	10	0.6415 (0.1259)	0.0481 (0.0133)	2.60%	1.96×10^3 (77.77)	30.56 (7.70)
40	20	0.6685 (0.1343)	0.0529 (0.0143)	4.00%	1.97×10^3 (83.68)	33.26 (8.24)
40	30	0.6637 (0.1275)	0.0553 (0.0142)	3.20%	1.98×10^3 (84.10)	34.54 (8.15)
40	40	0.6669 (0.1291)	0.0577 (0.0140)	3.00%	1.98×10^3 (86.66)	35.82 (7.95)
40	50	0.6596 (0.1260)	0.0590 (0.0152)	3.00%	1.99×10^3 (87.14)	36.36 (8.50)
50	10	0.6506 (0.1380)	0.0498 (0.0141)	3.80%	1.96×10^3 (80.11)	31.49 (8.27)
50	20	0.6553 (0.1310)	0.0545 (0.0154)	3.00%	1.97×10^3 (82.87)	34.01 (8.79)
50	30	0.6556 (0.1339)	0.0555 (0.0146)	3.00%	1.98×10^3 (83.64)	34.46 (8.33)
50	40	0.6594 (0.1310)	0.0579 (0.0149)	3.60%	1.98×10^3 (85.32)	35.75 (8.40)
50	50	0.6620 (0.1297)	0.0593 (0.0145)	2.80%	1.98×10^3 (87.71)	36.50 (8.16)
GB1; AA; Low throughput; $N = 3$**;** $K_2 = K_3$**;** K**-means**						

K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate	MFAD (Sequence)	MFAD (Fitness)
110	10	0.6084 (0.1242)	0.0315 (0.0081)	1.40%	1.99×10^3 (50.36)	21.01 (5.11)
10	20	0.6158 (0.1182)	0.0365 (0.0097)	1.20%	2.00×10^3 (55.21)	23.94 (5.85)
10	30	0.6359 (0.1208)	0.0415 (0.0107)	2.20%	2.00×10^3 (56.83)	26.90 (6.34)
10	40	0.6366 (0.1167)	0.0447 (0.0110)	1.40%	2.01×10^3 (57.18)	28.74 (6.48)
10	50	0.6436 (0.1230)	0.0474 (0.0114)	2.60%	2.01×10^3 (60.59)	30.31 (6.68)
20	10	0.6227 (0.1261)	0.0397 (0.0097)	2.20%	1.99×10^3 (62.79)	25.89 (5.93)
20	20	0.6394 (0.1268)	0.0451 (0.0106)	3.20%	2.00×10^3 (61.95)	29.03 (6.34)
20	30	0.6429 (0.1253)	0.0494 (0.0115)	2.60%	2.00×10^3 (64.63)	31.40 (6.75)
20	40	0.6481 (0.1248)	0.0524 (0.0116)	3.20%	2.01×10^3 (65.31)	33.00 (6.81)
20	50	0.6601 (0.1305)	0.0536 (0.0120)	3.40%	2.01×10^3 (65.12)	33.67 (6.93)
30	10	0.6491 (0.1317)	0.0450 (0.0113)	4.00%	1.98×10^3 (68.48)	28.84 (6.67)
30	20	0.6584 (0.1277)	0.0514 (0.0125)	3.20%	1.99×10^3 (67.36)	32.34 (7.15)
30	30	0.6588 (0.1288)	0.0548 (0.0132)	3.00%	1.99×10^3 (70.29)	34.17 (7.50)
30	40	0.6715 (0.1396)	0.0571 (0.0135)	5.60%	2.00×10^3 (73.43)	35.46 (7.64)
30	**50**	**0.6672 (0.1243)**	**0.0586 (0.0130)**	**3.60%**	$\mathbf{2.00\times10^3}$ **(72.38)**	**36.26 (7.28)**
40	10	0.6550 (0.1259)	0.0490 (0.0119)	3.20%	1.97×10^3 (74.34)	31.15 (6.96)
40	20	0.6509 (0.1338)	0.0551 (0.0135)	3.00%	1.98×10^3 (74.60)	34.39 (7.75)
40	30	0.6679 (0.1323)	0.0583 (0.0139)	4.20%	1.99×10^3 (76.22)	36.14 (7.93)
40	40	0.6718 (0.1304)	0.0599 (0.0134)	4.40%	1.99×10^3 (77.16)	36.94 (7.64)
40	50	0.6717 (0.1282)	0.0620 (0.0140)	3.60%	1.99×10^3 (79.08)	38.00 (7.89)
50	10	0.6555 (0.1278)	0.0513 (0.0128)	3.00%	1.97×10^3 (73.14)	32.38 (7.43)

50	20	0.6699 (0.1375)	0.0575 (0.0138)	5.20%	1.97×10^3 (75.26)	35.75 (7.86)
50	30	0.6649 (0.1310)	0.0594 (0.0142)	3.60%	1.98×10^3 (80.44)	36.63 (7.99)
50	40	0.6787 (0.1337)	0.0608 (0.0141)	6.00%	1.99×10^3 (79.12)	37.38 (7.89)
50	50	0.6725 (0.1237)	0.0623 (0.0143)	2.80%	1.99×10^3 (79.91)	38.08 (7.95)
GB1; AA; Medium throughput; $N=1$; Seurat (Louvain)						
k.param	resolution	Max Fitness	Mean Fitness	Global Max Hitting Rate	MFAD (Sequence)	MFAD (Fitness)
100	0.4	0.5043 (0.1201)	0.0121 (0.0032)	0.40%	2.08×10^3 (69.99)	8.56 (2.27)
100	0.8	0.4886 (0.1081)	0.0143 (0.0040)	0.20%	2.10×10^3 (68.17)	10.02 (2.73)
100	1.2	0.5135 (0.1234)	0.0162 (0.0044)	0.20%	2.11×10^3 (69.85)	11.30 (3.05)
100	1.6	0.5378 (0.1329)	0.0172 (0.0052)	0.60%	2.13×10^3 (73.31)	12.02 (3.58)
100	2	0.5390 (0.1334)	0.0174 (0.0051)	0.80%	2.10×10^3 (74.06)	12.15 (3.54)
200	0.4	0.5089 (0.1204)	0.0130 (0.0035)	0.20%	2.00×10^3 (44.70)	9.15 (2.48)
200	0.8	0.5013 (0.1145)	0.0135 (0.0035)	0.20%	2.03×10^3 (55.73)	9.52 (2.44)
200	1.2	0.5217 (0.1291)	0.0141 (0.0041)	0.80%	2.03×10^3 (58.30)	9.95 (2.85)
200	1.6	0.5289 (0.1242)	0.0157 (0.0048)	0.20%	2.05×10^3 (54.08)	11.00 (3.34)
200	2	0.5470 (0.1295)	0.0163 (0.0052)	0.40%	2.06×10^3 (55.55)	11.47 (3.59)
300	0.4	0.5006 (0.1160)	0.0120 (0.0031)	0.20%	2.03×10^3 (41.59)	8.54 (2.22)
300	0.8	0.5189 (0.1254)	0.0133 (0.0035)	0.60%	2.05×10^3 (49.05)	9.36 (2.44)
300	1.2	0.5420 (0.1231)	0.0155 (0.0041)	0.60%	2.03×10^3 (55.73)	10.86 (2.87)
300	1.6	0.5419 (0.1325)	0.0161 (0.0047)	0.60%	2.03×10^3 (55.95)	11.30 (3.29)
300	2	0.5586 (0.1274)	0.0174 (0.0054)	0.80%	2.05×10^3 (52.11)	12.23 (3.74)
400	0.4	0.4936 (0.1207)	0.0118 (0.0031)	0.00%	2.02×10^3 (41.51)	8.36 (2.23)

400	0.8	0.5214 (0.1302)	0.0135 (0.0034)	0.20%	2.04×10^3 (47.90)	9.50 (2.38)
400	1.2	0.5485 (0.1265)	0.0159 (0.0042)	0.20%	2.03×10^3 (52.23)	11.13 (2.91)
400	1.6	0.5338 (0.1265)	0.0159 (0.0043)	1.20%	2.02×10^3 (50.12)	11.11 (2.97)
400	2	0.5326 (0.1209)	0.0159 (0.0046)	0.40%	2.03×10^3 (51.14)	11.12 (3.17)
500	0.4	0.5184 (0.1291)	0.0116 (0.0029)	0.60%	2.05×10^3 (38.84)	8.22 (2.11)
500	0.8	0.5091 (0.1227)	0.0138 (0.0035)	0.20%	2.03×10^3 (47.29)	9.74 (2.43)
500	**1.2**	**0.5563 (0.1367)**	**0.0160 (0.0044)**	**1.60%**	$\mathbf{2.05\times10^3}$ **(48.16)**	**11.27 (3.03)**
500	1.6	0.5550 (0.1272)	0.0169 (0.0042)	1.20%	2.03×10^3 (49.11)	11.78 (2.89)
500	2	0.5438 (0.1302)	0.0164 (0.0042)	0.80%	2.02×10^3 (52.24)	11.46 (2.88)
600	0.4	0.5156 (0.1265)	0.0115 (0.0029)	0.20%	2.04×10^3 (37.93)	8.19 (2.09)
600	0.8	0.5041 (0.1250)	0.0131 (0.0033)	0.60%	2.02×10^3 (45.34)	9.27 (2.30)
600	1.2	0.5593 (0.1333)	0.0159 (0.0046)	1.20%	2.04×10^3 (48.04)	11.19 (3.18)
600	1.6	0.5578 (0.1215)	0.0167 (0.0044)	0.40%	2.05×10^3 (43.80)	11.71 (3.06)
600	2	0.5528 (0.1318)	0.0162 (0.0042)	1.00%	2.03×10^3 (47.92)	11.35 (2.93)
700	0.4	0.5353 (0.1364)	0.0116 (0.0029)	1.20%	2.07×10^3 (29.19)	8.25 (2.10)
700	0.8	0.5130 (0.1183)	0.0132 (0.0033)	0.20%	2.01×10^3 (43.31)	9.34 (2.30)
700	1.2	0.5577 (0.1427)	0.0154 (0.0043)	1.40%	2.04×10^3 (49.69)	10.82 (2.99)
700	1.6	0.5544 (0.1232)	0.0170 (0.0045)	0.40%	2.04×10^3 (45.95)	11.87 (3.08)
700	2	0.5522 (0.1276)	0.0165 (0.0043)	1.20%	2.03×10^3 (48.30)	11.56 (2.95)
GB1; Georgiev; Medium throughput; $N = 0$ (Random sampling); K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate	MFAD (Sequence)	MFAD (Fitness)
0	0	0.4753 (0.1352)	0.0092 (0.0023)	0.80%	4.71×10^3 (1.12)	6.60 (1.68)
GB1; Georgiev; Medium throughput;						

$N = 3$; $K_2 = K_3$; K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate	MFAD (Sequence)	MFAD (Fitness)
10	10	0.6118 (0.1384)	0.0324 (0.0172)	1.60%	4.55×10^3 (73.02)	21.50 (10.38)
10	20	0.6309 (0.1422)	0.0395 (0.0238)	2.00%	4.55×10^3 (76.35)	25.51 (13.41)
10	30	0.6468 (0.1471)	0.0436 (0.0260)	3.80%	4.55×10^3 (75.01)	27.83 (14.61)
10	40	0.6575 (0.1509)	0.0477 (0.0264)	2.40%	4.55×10^3 (73.61)	30.21 (14.87)
10	50	0.6606 (0.1456)	0.0480 (0.0256)	3.20%	4.55×10^3 (71.31)	30.41 (14.40)
20	10	0.6327 (0.1399)	0.0368 (0.0210)	1.80%	4.55×10^3 (63.44)	23.95 (12.07)
20	20	0.6449 (0.1396)	0.0449 (0.0256)	2.40%	4.54×10^3 (65.24)	28.53 (14.26)
20	30	0.6618 (0.1428)	0.0478 (0.0252)	3.40%	4.54×10^3 (60.75)	30.24 (14.05)
20	40	0.6620 (0.1415)	0.0495 (0.0251)	3.20%	4.54×10^3 (59.18)	31.18 (13.99)
20	50	0.6683 (0.1434)	0.0519 (0.0256)	2.20%	4.54×10^3 (57.13)	32.48 (14.27)
30	10	0.6473 (0.1495)	0.0408 (0.0246)	2.80%	4.55×10^3 (62.76)	26.21 (13.83)
30	20	0.6646 (0.1510)	0.0483 (0.0280)	3.60%	4.54×10^3 (60.11)	30.42 (15.41)
30	30	0.6742 (0.1471)	0.0519 (0.0283)	3.00%	4.54×10^3 (56.77)	32.40 (15.50)
30	**40**	**0.6833 (0.1432)**	**0.0539 (0.0270)**	**4.60%**	$\mathbf{4.54\times10^3}$ **(56.97)**	**33.61 (14.77)**
30	50	0.6829 (0.1471)	0.0539 (0.0266)	3.40%	4.55×10^3 (54.20)	33.56 (14.69)
40	10	0.6407 (0.1522)	0.0410 (0.0262)	3.00%	4.55×10^3 (63.07)	26.19 (14.46)
40	20	0.6676 (0.1483)	0.0490 (0.0277)	2.60%	4.54×10^3 (60.13)	30.81 (15.20)
40	30	0.6682 (0.1500)	0.0506 (0.0274)	3.60%	4.54×10^3 (59.18)	31.68 (14.97)
40	40	0.6695 (0.1494)	0.0517 (0.0265)	4.00%	4.55×10^3 (56.67)	32.39 (14.60)
40	50	0.6767 (0.1438)	0.0529 (0.0262)	3.00%	4.55×10^3 (55.73)	33.03 (14.38)
50	10	0.6448 (0.1504)	0.0427 (0.0244)	3.40%	4.54×10^3 (68.14)	27.16 (13.71)

50	20	0.6583 (0.1456)	0.0489 (0.0269)	2.80%	4.54×10^3 (65.49)	30.64 (14.79)
50	30	0.6585 (0.1437)	0.0500 (0.0255)	3.20%	4.54×10^3 (60.70)	31.36 (14.11)
50	40	0.6709 (0.1504)	0.0516 (0.0254)	4.20%	4.55×10^3 (60.37)	32.20 (14.11)
50	50	0.6688 (0.1458)	0.0517 (0.0257)	3.40%	4.55×10^3 (53.75)	32.27 (14.25)
PhoQ; AA; Medium throughput; $N = 0$ (Random sampling); K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate	MFAD (Sequence)	MFAD (Fitness)
0	0	0.2256 (0.0945)	0.0051 (0.0011)	0.00%	2.10×10^3 (23.87)	3.70 (0.82)
PhoQ; AA; Medium throughput; $N = 3$; $K_2 = K_3$; K-means						
10	10	0.2888 (0.1117)	0.0142 (0.0037)	0.00%	1.95×10^3 (55.37)	9.38 (2.21)
10	20	0.2899 (0.1109)	0.0154 (0.0042)	0.00%	1.95×10^3 (61.30)	10.04 (2.43)
10	30	0.2941 (0.1122)	0.0160 (0.0040)	0.20%	1.96×10^3 (61.20)	10.39 (2.29)
10	40	0.2937 (0.1137)	0.0163 (0.0042)	0.40%	1.96×10^3 (63.06)	10.53 (2.39)
10	50	0.2957 (0.1171)	0.0167 (0.0041)	0.60%	1.96×10^3 (60.82)	10.77 (2.31)
20	10	0.2889 (0.1077)	0.0158 (0.0041)	0.00%	1.95×10^3 (62.64)	10.25 (2.35)
20	20	0.2877 (0.1057)	0.0168 (0.0044)	0.20%	1.96×10^3 (67.68)	10.79 (2.45)
20	30	0.2923 (0.1049)	0.0172 (0.0044)	0.00%	1.95×10^3 (65.91)	11.02 (2.43)
20	40	0.2950 (0.1103)	0.0178 (0.0044)	0.20%	1.95×10^3 (68.93)	11.32 (2.41)
20	50	0.2949 (0.1121)	0.0179 (0.0044)	0.20%	1.95×10^3 (71.86)	11.41 (2.39)
30	10	0.2864 (0.0990)	0.0169 (0.0044)	0.20%	1.93×10^3 (72.50)	10.84 (2.48)
30	20	0.2852 (0.0980)	0.0178 (0.0045)	0.20%	1.93×10^3 (74.42)	11.32 (2.45)
30	30	0.2954 (0.1016)	0.0184 (0.0044)	0.20%	1.93×10^3 (75.48)	11.66 (2.41)
30	40	0.2993 (0.1118)	0.0187 (0.0046)	0.60%	1.93×10^3 (75.35)	11.78 (2.52)

30	50	0.2956 (0.1054)	0.0191 (0.0048)	0.40%	1.93×10^3 (74.61)	11.98 (2.60)
40	10	0.2951 (0.1115)	0.0178 (0.0047)	0.40%	1.90×10^3 (78.49)	11.30 (2.61)
40	20	0.3003 (0.1032)	0.0185 (0.0047)	0.00%	1.91×10^3 (79.41)	11.72 (2.60)
40	**30**	**0.3010 (0.1084)**	**0.0189 (0.0047)**	**0.20%**	$\mathbf{1.91\times10^3}$ **(83.04)**	**11.94 (2.56)**
40	40	0.3073 (0.1114)	0.0193 (0.0047)	0.00%	1.90×10^3 (84.00)	12.17 (2.60)
40	50	0.2993 (0.1062)	0.0195 (0.0048)	0.20%	1.90×10^3 (83.99)	12.24 (2.61)
50	10	0.2911 (0.1027)	0.0181 (0.0048)	0.40%	1.91×10^3 (81.69)	11.47 (2.62)
50	20	0.2992 (0.1094)	0.0191 (0.0047)	0.60%	1.91×10^3 (80.65)	12.02 (2.55)
50	30	0.2998 (0.1121)	0.0196 (0.0049)	0.60%	1.91×10^3 (85.13)	12.25 (2.69)
50	40	0.3003 (0.0978)	0.0199 (0.0048)	0.20%	1.91×10^3 (86.03)	12.41 (2.56)
50	50	0.3020 (0.1048)	0.0200 (0.0048)	0.40%	1.91×10^3 (87.09)	12.49 (2.57)
PhoQ; Georgiev; Medium throughput; $N = 0$ (Random sampling); K-means						
K_1	K_2	Max Fitness	Mean Fitness	Global Max Hitting Rate	MFAD (Sequence)	MFAD (Fitness)
0	0	0.2272 (0.1074)	0.0051 (0.0012)	0.40%	4.71×10^3 (1.17)	3.72 (0.87)
PhoQ; Georgiev; Medium throughput; $N = 3$; $K_2 = K_3$; K-means						
10	10	0.3382 (0.1606)	0.0192 (0.0051)	1.80%	4.52×10^3 (75.63)	12.18 (2.92)
10	20	0.3428 (0.1644)	0.0212 (0.0057)	2.60%	4.52×10^3 (75.68)	13.18 (3.18)
10	**30**	**0.3569 (0.1734)**	**0.0224 (0.0060)**	**3.00%**	$\mathbf{4.52\times10^3}$ **(73.80)**	**13.80 (3.35)**
10	40	0.3625 (0.1715)	0.0227 (0.0058)	2.60%	4.52×10^3 (72.46)	13.98 (3.18)
10	50	0.3783 (0.1987)	0.0231 (0.0062)	5.40%	4.52×10^3 (72.27)	14.16 (3.45)
20	10	0.3285 (0.1514)	0.0188 (0.0049)	1.60%	4.55×10^3 (55.54)	11.98 (2.81)
20	20	0.3369 (0.1581)	0.0205 (0.0053)	2.00%	4.55×10^3 (53.77)	12.86 (2.97)

20	30	0.3510 (0.1634)	0.0215 (0.0052)	2.80%	4.55×10^3 (53.05)	13.40 (2.88)
20	40	0.3612 (0.1684)	0.0221 (0.0053)	3.20%	4.55×10^3 (52.47)	13.70 (2.88)
20	50	0.3778 (0.1849)	0.0229 (0.0054)	4.20%	4.55×10^3 (52.37)	14.14 (2.97)
30	10	0.3305 (0.1462)	0.0193 (0.0055)	1.60%	4.55×10^3 (56.81)	12.21 (3.08)
30	20	0.3420 (0.1635)	0.0209 (0.0058)	2.80%	4.55×10^3 (55.49)	13.04 (3.19)
30	30	0.3538 (0.1681)	0.0216 (0.0056)	3.40%	4.56×10^3 (53.51)	13.45 (3.05)
30	40	0.3491 (0.1599)	0.0218 (0.0055)	2.60%	4.56×10^3 (51.19)	13.53 (3.00)
30	50	0.3617 (0.1737)	0.0224 (0.0055)	3.40%	4.56×10^3 (51.62)	13.85 (2.98)
40	10	0.3381 (0.1495)	0.0195 (0.0056)	1.60%	4.55×10^3 (60.96)	12.31 (3.14)
40	20	0.3422 (0.1534)	0.0209 (0.0054)	2.00%	4.55×10^3 (57.10)	13.09 (3.02)
40	30	0.3461 (0.1494)	0.0217 (0.0053)	1.80%	4.55×10^3 (55.50)	13.47 (2.89)
40	40	0.3641 (0.1668)	0.0223 (0.0054)	3.20%	4.55×10^3 (55.01)	13.81 (2.99)
40	50	0.3492 (0.1583)	0.0220 (0.0052)	2.60%	4.56×10^3 (53.95)	13.65 (2.84)
50	10	0.3477 (0.1593)	0.0202 (0.0055)	2.00%	4.53×10^3 (63.37)	12.70 (3.04)
50	20	0.3562 (0.1697)	0.0214 (0.0057)	2.80%	4.54×10^3 (61.04)	13.37 (3.16)
50	30	0.3536 (0.1670)	0.0218 (0.0055)	3.00%	4.54×10^3 (61.01)	13.51 (2.96)
50	40	0.3696 (0.1821)	0.0225 (0.0056)	4.40%	4.54×10^3 (60.24)	13.89 (3.04)
50	50	0.3607 (0.1700)	0.0229 (0.0056)	3.40%	4.54×10^3 (59.12)	14.08 (3.01)
PhoQ; Georgiev; Low throughput; $N = 3$; $K_2 = K_3$; K-means						
10	10	0.3351 (0.1549)	0.0196 (0.0051)	2.00%	4.51×10^3 (77.83)	12.40 (2.90)
10	20	0.3453 (0.1601)	0.0217 (0.0056)	2.20%	4.51×10^3 (75.20)	13.44 (3.09)
10	30	0.3698 (0.1860)	0.0224 (0.0057)	4.20%	4.51×10^3 (74.08)	13.83 (3.19)
10	**40**	**0.3713 (0.1765)**	**0.0234 (0.0058)**	**3.40%**	$\mathbf{4.51\times10^3}$ **(73.10)**	**14.35 (3.20)**

10	50	0.3728 (0.1880)	0.0236 (0.0057)	4.60%	4.51×10^3 (72.17)	14.44 (3.12)
20	10	0.3433 (0.1598)	0.0194 (0.0049)	2.00%	4.55×10^3 (57.16)	12.30 (2.74)
20	20	0.3718 (0.1904)	0.0210 (0.0051)	4.80%	4.55×10^3 (55.87)	13.16 (2.85)
20	30	0.3616 (0.1792)	0.0222 (0.0052)	3.80%	4.55×10^3 (54.16)	13.75 (2.89)
20	40	0.3686 (0.1797)	0.0230 (0.0048)	4.00%	4.55×10^3 (53.13)	14.12 (2.60)
20	50	0.3680 (0.1799)	0.0233 (0.0050)	4.20%	4.55×10^3 (52.80)	14.31 (2.71)
30	10	0.3431 (0.1598)	0.0201 (0.0053)	2.20%	4.55×10^3 (56.06)	12.66 (2.97)
30	20	0.3566 (0.1719)	0.0213 (0.0054)	3.40%	4.56×10^3 (54.30)	13.30 (3.00)
30	30	0.3491 (0.1478)	0.0224 (0.0051)	1.60%	4.56×10^3 (52.25)	13.81 (2.80)
30	40	0.3639 (0.1778)	0.0228 (0.0052)	3.80%	4.56×10^3 (52.72)	14.07 (2.85)
30	50	0.3725 (0.1750)	0.0234 (0.0052)	3.40%	4.56×10^3 (52.79)	14.36 (2.81)
40	10	0.3382 (0.1542)	0.0203 (0.0051)	2.00%	4.54×10^3 (62.56)	12.76 (2.87)
40	20	0.3404 (0.1492)	0.0216 (0.0053)	2.00%	4.55×10^3 (58.07)	13.43 (2.85)
40	30	0.3457 (0.1573)	0.0222 (0.0052)	2.60%	4.55×10^3 (56.12)	13.74 (2.87)
40	40	0.3520 (0.1646)	0.0229 (0.0052)	3.00%	4.55×10^3 (54.76)	14.08 (2.81)
40	50	0.3589 (0.1610)	0.0233 (0.0050)	2.80%	4.56×10^3 (53.41)	14.30 (2.69)
50	10	0.3493 (0.1555)	0.0211 (0.0056)	2.00%	4.53×10^3 (64.40)	13.21 (3.10)
50	20	0.3696 (0.1734)	0.0225 (0.0056)	3.40%	4.53×10^3 (63.02)	13.95 (3.06)
50	30	0.3763 (0.1861)	0.0229 (0.0057)	4.20%	4.54×10^3 (63.54)	14.16 (3.11)
50	40	0.3707 (0.1715)	0.0235 (0.0053)	3.20%	4.54×10^3 (60.59)	14.44 (2.89)
50	50	0.3828 (0.1880)	0.0239 (0.0053)	4.80%	4.54×10^3 (57.60)	14.66 (2.87)

Table S2: Compositions of CLADE training data. Two datasets, GB1 and PhoQ, were tested. Two encoding methods, AA and Georgiev, were used. Medium- and low-throughput systems were simulated. Two clustering methods, K-means and Louvain, were used. The training data consists of 384 variants. All metrics were evaluated on 500 independent repeats (Note: A few repeats (less than 10/500) in deep hierarchical clustering ($N = 3$) with Georgiev encoding may have issue that a cluster contains fewer variants than the number of its clusters at next hierarchy. These simulations were discarded and more simulations with different random seeds were performed until 500 successful simulations were collected.). Both expectations and standard deviations (in parenthesis) were shown for max fitness, mean fitness, sequence MFAD, and fitness MFAD in training data. The sequence MFAD and the fitness MFAD quantify sequence diversity and fitness diversity, respectively. In each clustering architecture, the case with best expected max fitness shown in Table 1 was highlighted by bold font.

Training Errors					
GB1; AA; Medium throughput; $N = 0$ (Random sampling); K-means					
K_1	K_2	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
0	0	0.940	4.04×10^{-4}	0.541	1.50×10^{-3}
GB1; AA; Medium throughput; $N = 1$; K-means					
K_1	K_2	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
10	0	0.955	4.77×10^{-4}	0.602	1.33×10^{-3}
20	0	0.957	5.40×10^{-4}	0.618	1.29×10^{-3}
30	0	0.957	5.49×10^{-4}	0.623	1.27×10^{-3}
40	0	0.958	5.59×10^{-4}	0.624	1.27×10^{-3}
50	0	0.960	5.60×10^{-4}	0.625	1.27×10^{-3}
60	0	0.960	5.78×10^{-4}	0.627	1.26×10^{-3}
70	0	0.965	5.45×10^{-4}	0.624	1.26×10^{-3}
80	0	0.966	5.56×10^{-4}	0.628	1.26×10^{-3}
90	**0**	**0.966**	$\mathbf{5.50\times10^{-4}}$	**0.632**	$\mathbf{1.24\times10^{-3}}$
100	0	0.967	5.63×10^{-4}	0.627	1.26×10^{-3}
150	0	0.972	5.28×10^{-4}	0.626	1.26×10^{-3}
200	0	0.976	5.07×10^{-4}	0.627	1.26×10^{-3}
250	0	0.977	4.90×10^{-4}	0.623	1.27×10^{-3}
300	0	0.981	4.59×10^{-4}	0.622	1.29×10^{-3}
350	0	0.981	4.48×10^{-4}	0.622	1.28×10^{-3}
400	0	0.983	4.31×10^{-4}	0.616	1.30×10^{-3}
450	0	0.984	4.02×10^{-4}	0.610	1.32×10^{-3}
500	0	0.985	3.93×10^{-4}	0.605	1.33×10^{-3}
550	0	0.985	3.94×10^{-4}	0.598	1.35×10^{-3}
600	0	0.987	3.49×10^{-4}	0.601	1.36×10^{-3}
650	0	0.987	3.52×10^{-4}	0.597	1.35×10^{-3}
700	0	0.987	3.63×10^{-4}	0.594	1.38×10^{-3}
750	0	0.988	3.42×10^{-4}	0.584	1.39×10^{-3}
800	0	0.988	3.40×10^{-4}	0.582	1.41×10^{-3}
850	0	0.988	3.34×10^{-4}	0.584	1.41×10^{-3}
900	0	0.989	3.20×10^{-4}	0.575	1.43×10^{-3}
950	0	0.989	3.04×10^{-4}	0.574	1.44×10^{-3}
1000	0	0.989	3.01×10^{-4}	0.581	1.41×10^{-3}
GB1; AA; Medium throughput; $N = 2$; K-means					
K_1	K_2	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
10	10	0.961	5.47×10^{-4}	0.622	1.27×10^{-3}
10	20	0.962	5.70×10^{-4}	0.622	1.27×10^{-3}
10	30	0.965	5.61×10^{-4}	0.630	1.24×10^{-3}
10	40	0.968	5.56×10^{-4}	0.630	1.24×10^{-3}
10	50	0.968	5.38×10^{-4}	0.631	1.24×10^{-3}

20	10	0.967	5.42×10^{-4}	0.628	1.25×10^{-3}
20	20	0.971	5.23×10^{-4}	0.634	1.23×10^{-3}
20	30	0.972	5.28×10^{-4}	0.635	1.23×10^{-3}
20	40	0.974	5.13×10^{-4}	0.638	1.22×10^{-3}
20	50	0.976	5.06×10^{-4}	0.638	1.22×10^{-3}
30	10	0.970	5.56×10^{-4}	0.631	1.24×10^{-3}
30	20	0.974	5.31×10^{-4}	0.634	1.23×10^{-3}
30	30	0.974	5.38×10^{-4}	0.635	1.22×10^{-3}
30	40	0.978	5.02×10^{-4}	0.637	1.22×10^{-3}
30	50	0.978	5.09×10^{-4}	0.634	1.23×10^{-3}
40	10	0.974	5.12×10^{-4}	0.630	1.24×10^{-3}
40	20	0.977	5.10×10^{-4}	0.631	1.24×10^{-3}
40	**30**	**0.978**	$\mathbf{5.11\times10^{-4}}$	**0.635**	$\mathbf{1.23\times10^{-3}}$
40	40	0.980	4.85×10^{-4}	0.632	1.23×10^{-3}
40	50	0.980	4.86×10^{-4}	0.634	1.23×10^{-3}
50	10	0.975	5.12×10^{-4}	0.628	1.25×10^{-3}
50	20	0.979	4.84×10^{-4}	0.628	1.24×10^{-3}
50	30	0.980	4.54×10^{-4}	0.633	1.24×10^{-3}
50	40	0.981	4.64×10^{-4}	0.633	1.23×10^{-3}
50	50	0.982	4.36×10^{-4}	0.634	1.23×10^{-3}
GB1; AA; Medium throughput; $N = 3$; $K_2 = K_3$; K-means					
10	10	0.969	5.38×10^{-4}	0.622	1.26×10^{-3}
10	20	0.974	5.30×10^{-4}	0.624	1.25×10^{-3}
10	30	0.978	4.98×10^{-4}	0.631	1.23×10^{-3}
10	40	0.979	5.01×10^{-4}	0.629	1.23×10^{-3}
10	50	0.981	4.77×10^{-4}	0.630	1.23×10^{-3}
20	10	0.978	4.68×10^{-4}	0.629	1.24×10^{-3}
20	20	0.980	4.63×10^{-4}	0.631	1.23×10^{-3}
20	30	0.982	4.52×10^{-4}	0.635	1.21×10^{-3}
20	40	0.983	4.68×10^{-4}	0.633	1.22×10^{-3}
20	50	0.985	4.11×10^{-4}	0.632	1.23×10^{-3}
30	10	0.980	4.77×10^{-4}	0.625	1.25×10^{-3}
30	20	0.982	4.72×10^{-4}	0.629	1.24×10^{-3}
30	30	0.983	4.68×10^{-4}	0.628	1.24×10^{-3}
30	**40**	**0.985**	$\mathbf{4.42\times10^{-4}}$	**0.633**	$\mathbf{1.23\times10^{-3}}$
30	50	0.985	4.41×10^{-4}	0.627	1.25×10^{-3}
40	10	0.982	4.51×10^{-4}	0.625	1.25×10^{-3}
40	20	0.984	4.58×10^{-4}	0.625	1.25×10^{-3}
40	30	0.985	4.31×10^{-4}	0.629	1.24×10^{-3}
40	40	0.987	4.14×10^{-4}	0.628	1.24×10^{-3}
40	50	0.987	4.18×10^{-4}	0.628	1.24×10^{-3}
50	10	0.983	4.43×10^{-4}	0.622	1.26×10^{-3}
50	20	0.984	4.42×10^{-4}	0.621	1.26×10^{-3}
50	30	0.985	4.29×10^{-4}	0.626	1.25×10^{-3}
50	40	0.986	4.23×10^{-4}	0.623	1.26×10^{-3}
50	50	0.988	3.89×10^{-4}	0.628	1.25×10^{-3}

GB1; AA; Low throughput; $N = 3$; $K_2 = K_3$; K-means					
K_1	K_2	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
10	10	0.971	5.29×10^{-4}	0.624	1.26×10^{-3}
10	20	0.973	5.37×10^{-4}	0.632	1.23×10^{-3}
10	30	0.977	5.39×10^{-4}	0.634	1.22×10^{-3}
10	40	0.980	4.94×10^{-4}	0.639	1.20×10^{-3}
10	50	0.981	5.01×10^{-4}	0.637	1.21×10^{-3}
20	10	0.976	5.02×10^{-4}	0.636	1.22×10^{-3}
20	20	0.980	4.91×10^{-4}	0.640	1.20×10^{-3}
20	30	0.982	4.75×10^{-4}	0.641	1.20×10^{-3}
20	40	0.983	4.79×10^{-4}	0.645	1.19×10^{-3}
20	50	0.984	4.61×10^{-4}	0.646	1.19×10^{-3}
30	10	0.978	5.25×10^{-4}	0.634	1.22×10^{-3}
30	20	0.982	4.91×10^{-4}	0.637	1.21×10^{-3}
30	30	0.983	4.98×10^{-4}	0.637	1.21×10^{-3}
30	40	0.985	4.77×10^{-4}	0.639	1.20×10^{-3}
30	**50**	**0.985**	$\mathbf{4.81\times10^{-4}}$	**0.638**	$\mathbf{1.21\times10^{-3}}$
40	10	0.981	4.94×10^{-4}	0.628	1.24×10^{-3}
40	20	0.984	4.74×10^{-4}	0.635	1.22×10^{-3}
40	30	0.985	4.65×10^{-4}	0.635	1.22×10^{-3}
40	40	0.986	4.48×10^{-4}	0.637	1.22×10^{-3}
40	50	0.987	4.30×10^{-4}	0.638	1.21×10^{-3}
50	10	0.983	4.58×10^{-4}	0.629	1.24×10^{-3}
50	20	0.985	4.60×10^{-4}	0.636	1.22×10^{-3}
50	30	0.986	4.46×10^{-4}	0.637	1.22×10^{-3}
50	40	0.986	4.67×10^{-4}	0.638	1.22×10^{-3}
50	50	0.987	4.40×10^{-4}	0.639	1.21×10^{-3}
GB1; AA; Medium throughput; $N = 1$; Seurat (Louvain)					
k.param	resolution	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
100	0.4	0.949	4.06×10^{-4}	0.555	1.46×10^{-3}
100	0.8	0.953	3.90×10^{-4}	0.564	1.43×10^{-3}
100	1.2	0.957	4.04×10^{-4}	0.573	1.40×10^{-3}
100	1.6	0.958	4.33×10^{-4}	0.573	1.41×10^{-3}
100	2	0.957	4.48×10^{-4}	0.567	1.42×10^{-3}
200	0.4	0.954	3.94×10^{-4}	0.535	1.50×10^{-3}
200	0.8	0.952	3.85×10^{-4}	0.548	1.46×10^{-3}
200	1.2	0.953	3.97×10^{-4}	0.549	1.46×10^{-3}
200	1.6	0.958	4.09×10^{-4}	0.558	1.45×10^{-3}
200	2	0.961	4.06×10^{-4}	0.564	1.44×10^{-3}
300	0.4	0.946	4.34×10^{-4}	0.537	1.49×10^{-3}
300	0.8	0.948	4.33×10^{-4}	0.553	1.45×10^{-3}
300	1.2	0.957	4.19×10^{-4}	0.567	1.42×10^{-3}
300	1.6	0.954	4.44×10^{-4}	0.561	1.44×10^{-3}

300	2	0.960	4.44×10^{-4}	0.570	1.43×10^{-3}
400	0.4	0.948	4.06×10^{-4}	0.529	1.52×10^{-3}
400	0.8	0.952	4.11×10^{-4}	0.550	1.47×10^{-3}
400	1.2	0.954	4.60×10^{-4}	0.571	1.42×10^{-3}
400	1.6	0.955	4.51×10^{-4}	0.573	1.41×10^{-3}
400	2	0.956	4.41×10^{-4}	0.572	1.41×10^{-3}
500	0.4	0.949	4.10×10^{-4}	0.533	1.51×10^{-3}
500	0.8	0.954	3.97×10^{-4}	0.554	1.45×10^{-3}
500	**1.2**	**0.952**	$\mathbf{4.73\times10^{-4}}$	**0.568**	$\mathbf{1.43\times10^{-3}}$
500	1.6	0.956	4.80×10^{-4}	0.580	1.39×10^{-3}
500	2	0.955	4.53×10^{-4}	0.569	1.41×10^{-3}
600	0.4	0.947	4.16×10^{-4}	0.534	1.51×10^{-3}
600	0.8	0.951	4.00×10^{-4}	0.548	1.47×10^{-3}
600	1.2	0.957	4.38×10^{-4}	0.569	1.43×10^{-3}
600	1.6	0.956	4.75×10^{-4}	0.587	1.37×10^{-3}
600	2	0.957	4.61×10^{-4}	0.581	1.39×10^{-3}
700	0.4	0.949	4.39×10^{-4}	0.543	1.49×10^{-3}
700	0.8	0.950	4.14×10^{-4}	0.545	1.47×10^{-3}
700	1.2	0.951	4.85×10^{-4}	0.567	1.43×10^{-3}
700	1.6	0.956	4.89×10^{-4}	0.585	1.38×10^{-3}
700	2	0.953	4.93×10^{-4}	0.585	1.38×10^{-3}
GB1; Georgiev; Medium throughput; $N = 0$ (Random sampling); K-means					
K_1	K_2	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
0	0	0.945	4.84×10^{-4}	0.46	1.66×10^{-3}
GB1; Georgiev; Medium throughput; $N = 3$; $K_2 = K_3$; K-means					
K_1	K_2	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
10	10	0.979	4.25×10^{-4}	0.467	1.73×10^{-3}
10	20	0.981	4.27×10^{-4}	0.472	1.74×10^{-3}
10	30	0.982	4.24×10^{-4}	0.464	1.77×10^{-3}
10	40	0.982	4.38×10^{-4}	0.468	1.77×10^{-3}
10	50	0.984	4.18×10^{-4}	0.468	1.78×10^{-3}
20	10	0.980	4.39×10^{-4}	0.476	1.72×10^{-3}
20	20	0.983	4.20×10^{-4}	0.478	1.73×10^{-3}
20	30	0.983	4.39×10^{-4}	0.476	1.74×10^{-3}
20	40	0.985	4.09×10^{-4}	0.480	1.74×10^{-3}
20	50	0.986	4.18×10^{-4}	0.485	1.73×10^{-3}
30	10	0.982	4.06×10^{-4}	0.485	1.68×10^{-3}
30	20	0.985	3.87×10^{-4}	0.486	1.68×10^{-3}
30	30	0.985	4.01×10^{-4}	0.484	1.70×10^{-3}
30	**40**	**0.986**	$\mathbf{4.15\times10^{-4}}$	**0.493**	$\mathbf{1.69\times10^{-3}}$
30	50	0.986	4.18×10^{-4}	0.497	1.68×10^{-3}
40	10	0.982	3.79×10^{-4}	0.493	1.63×10^{-3}
40	20	0.985	3.89×10^{-4}	0.486	1.68×10^{-3}

40	30	0.985	3.84×10^{-4}	0.494	1.66×10^{-3}
40	40	0.986	3.86×10^{-4}	0.497	1.67×10^{-3}
40	50	0.986	4.00×10^{-4}	0.499	1.67×10^{-3}
50	10	0.984	3.61×10^{-4}	0.487	1.66×10^{-3}
50	20	0.985	3.78×10^{-4}	0.492	1.66×10^{-3}
50	30	0.985	3.80×10^{-4}	0.495	1.65×10^{-3}
50	40	0.986	3.77×10^{-4}	0.498	1.64×10^{-3}
50	50	0.986	3.84×10^{-4}	0.500	1.64×10^{-3}
PhoQ; AA; Medium throughput; $N = 0$ (Random sampling); K-means					
K_1	K_2	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
0	0	0.825	2.30×10^{-4}	0.381	4.54×10^{-4}
PhoQ; AA; Medium throughput; $N = 3$; $K_2 = K_3$; K-means					
K_1	K_2	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
10	10	0.865	4.16×10^{-4}	0.437	4.26×10^{-4}
10	20	0.865	4.37×10^{-4}	0.439	4.25×10^{-4}
10	30	0.867	4.49×10^{-4}	0.440	4.25×10^{-4}
10	40	0.869	4.43×10^{-4}	0.440	4.24×10^{-4}
10	50	0.867	4.57×10^{-4}	0.436	4.27×10^{-4}
20	10	0.875	4.14×10^{-4}	0.433	4.30×10^{-4}
20	20	0.875	4.26×10^{-4}	0.433	4.29×10^{-4}
20	30	0.866	4.57×10^{-4}	0.433	4.29×10^{-4}
20	40	0.866	4.72×10^{-4}	0.434	4.29×10^{-4}
20	50	0.872	4.66×10^{-4}	0.434	4.29×10^{-4}
30	10	0.871	4.31×10^{-4}	0.430	4.30×10^{-4}
30	20	0.875	4.37×10^{-4}	0.432	4.29×10^{-4}
30	30	0.871	4.66×10^{-4}	0.429	4.32×10^{-4}
30	40	0.873	4.67×10^{-4}	0.431	4.31×10^{-4}
30	50	0.869	4.84×10^{-4}	0.432	4.29×10^{-4}
40	10	0.874	4.46×10^{-4}	0.429	4.32×10^{-4}
40	20	0.87	4.75×10^{-4}	0.429	4.30×10^{-4}
40	**30**	**0.87**	$\mathbf{4.83\times10^{-4}}$	**0.432**	$\mathbf{4.30\times10^{-4}}$
40	40	0.871	4.99×10^{-4}	0.431	4.31×10^{-4}
40	50	0.870	4.96×10^{-4}	0.431	4.31×10^{-4}
50	10	0.872	4.53×10^{-4}	0.430	4.30×10^{-4}
50	20	0.869	4.83×10^{-4}	0.428	4.31×10^{-4}
50	30	0.868	4.90×10^{-4}	0.427	4.32×10^{-4}
50	40	0.870	4.92×10^{-4}	0.426	4.33×10^{-4}
50	50	0.872	4.98×10^{-4}	0.429	4.32×10^{-4}
PhoQ; Georgiev; Medium throughput; $N = 0$ (Random sampling); K-means					
K_1	K_2	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
0	0	0.899	1.88×10^{-4}	0.378	4.56×10^{-4}

PhoQ; Georgiev; Medium throughput; $N = 3$; $K_2 = K_3$; K-means					
K_1	K_2	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
10	10	0.909	4.80×10^{-4}	0.384	4.74×10^{-4}
10	20	0.910	4.93×10^{-4}	0.386	4.70×10^{-4}
10	**30**	**0.907**	$\mathbf{5.60\times10^{-4}}$	**0.383**	$\mathbf{4.77\times10^{-4}}$
10	40	0.900	5.89×10^{-4}	0.383	4.75×10^{-4}
10	50	0.902	6.07×10^{-4}	0.384	4.76×10^{-4}
20	10	0.900	4.89×10^{-4}	0.387	4.68×10^{-4}
20	20	0.900	5.34×10^{-4}	0.388	4.70×10^{-4}
20	30	0.894	5.68×10^{-4}	0.388	4.70×10^{-4}
20	40	0.892	6.06×10^{-4}	0.387	4.70×10^{-4}
20	50	0.892	6.38×10^{-4}	0.388	4.75×10^{-4}
30	10	0.901	4.87×10^{-4}	0.390	4.61×10^{-4}
30	20	0.896	5.46×10^{-4}	0.392	4.62×10^{-4}
30	30	0.893	5.81×10^{-4}	0.387	4.66×10^{-4}
30	40	0.887	5.93×10^{-4}	0.392	4.61×10^{-4}
30	50	0.895	5.89×10^{-4}	0.392	4.66×10^{-4}
40	10	0.894	5.25×10^{-4}	0.388	4.61×10^{-4}
40	20	0.893	5.51×10^{-4}	0.390	4.62×10^{-4}
40	30	0.889	5.73×10^{-4}	0.391	4.63×10^{-4}
40	40	0.888	6.15×10^{-4}	0.39	4.66×10^{-4}
40	50	0.892	5.72×10^{-4}	0.394	4.60×10^{-4}
50	10	0.894	5.56×10^{-4}	0.387	4.61×10^{-4}
50	20	0.890	5.99×10^{-4}	0.388	4.63×10^{-4}
50	30	0.886	5.95×10^{-4}	0.390	4.63×10^{-4}
50	40	0.884	6.44×10^{-4}	0.390	4.66×10^{-4}
50	50	0.882	6.25×10^{-4}	0.392	4.65×10^{-4}
PhoQ; Georgiev; Low throughput; $N = 3$; $K_2 = K_3$; K-means					
K_1	K_2	R_p (CV)	RMSE (CV)	R_p (testing)	RMSE (testing)
10	10	0.912	4.69×10^{-4}	0.386	4.72×10^{-4}
10	20	0.907	5.23×10^{-4}	0.386	4.75×10^{-4}
10	30	0.904	5.72×10^{-4}	0.384	4.77×10^{-4}
10	**40**	**0.895**	$\mathbf{6.27\times10^{-4}}$	**0.384**	$\mathbf{4.77\times10^{-4}}$
10	50	0.893	6.37×10^{-4}	0.385	4.76×10^{-4}
20	10	0.906	4.94×10^{-4}	0.388	4.69×10^{-4}
20	20	0.896	5.87×10^{-4}	0.385	4.74×10^{-4}
20	30	0.899	5.74×10^{-4}	0.390	4.73×10^{-4}
20	40	0.897	5.98×10^{-4}	0.392	4.70×10^{-4}
20	50	0.892	6.25×10^{-4}	0.391	4.71×10^{-4}
30	10	0.900	5.39×10^{-4}	0.393	4.61×10^{-4}
30	20	0.895	5.75×10^{-4}	0.393	4.66×10^{-4}
30	30	0.888	6.02×10^{-4}	0.397	4.60×10^{-4}
30	40	0.883	6.64×10^{-4}	0.396	4.62×10^{-4}

30	50	0.888	6.51×10^{-4}	0.398	4.59×10^{-4}
40	10	0.894	5.46×10^{-4}	0.394	4.57×10^{-4}
40	20	0.891	5.77×10^{-4}	0.394	4.59×10^{-4}
40	30	0.891	5.78×10^{-4}	0.394	4.59×10^{-4}
40	40	0.886	6.21×10^{-4}	0.398	4.58×10^{-4}
40	50	0.887	6.23×10^{-4}	0.397	4.62×10^{-4}
50	10	0.894	5.65×10^{-4}	0.388	4.64×10^{-4}
50	20	0.888	6.38×10^{-4}	0.386	4.66×10^{-4}
50	30	0.889	6.55×10^{-4}	0.390	4.66×10^{-4}
50	40	0.883	6.78×10^{-4}	0.390	4.65×10^{-4}
50	50	0.888	6.66×10^{-4}	0.394	4.64×10^{-4}

Table S3: Training errors of CLADE. All metrics were evaluated on 500 independent repeats. Two datasets, GB1 and PhoQ, were tested. Two encoding methods, AA and Georgiev, were used. Medium- and low-throughput systems were simulated. Two clustering methods, K-means and Louvain, were used. Root mean square errors (RMSE) and Pearson correlations (R_p) were used to evaluate the performance of CLADE models on both 5-fold cross validation (CV) and testing set (i.e. Full dataset excluding training data). In each clustering architecture, the case with best expected max fitness shown in Table 1 was highlighted by bold font.

Dataset	Encoding method	Architecture	Expected max fitness	Expected mean fitness	Global max hitting rate
GB1	AA	$N = 0$	0.774	0.305	18.6%
GB1	AA	$N = 3$	0.888	0.423	50.8%
GB1	AA	$N = 3$ (low throughput)	0.904	0.431	55.6%
GB1	Georgiev	$N = 0$	0.747	0.272	8.2%
GB1	Georgiev	$N = 3$	0.860	0.367	30.6%
PhoQ	AA	$N = 0$	0.299	0.072	1.0%
PhoQ	AA	$N = 3$	0.357	0.093	3.0%
PhoQ	Georgiev	$N = 0$	0.371	0.077	7.2%
PhoQ	Georgiev	$N = 3$	0.503	0.096	20.6%
PhoQ	Georgiev	$N = 3$ (low throughput)	0.501	0.099	21.0%

Table S4: CLADE performance with different encoding methods on different datasets. Two datasets, GB1 and PhoQ, were performed. Two encoding methods, AA and Georgiev, were used to compare their effects on CLADE. Clustering architectures $N = 0$ and $N = 3$ were shown. $N = 0$ indicates randomly sampled training data which is equivalent to the MLDE approach. Parameters for deep CLADE with $N = 3$ were explored (Table S1-S3), and the case with highest expected max fitness was presented in this table. All cases used K-means for clustering method. TUnless explicitly indicated, the batch size is taken as 96 to simulate the medium-throughput systems. All statistics were obtained from 500 independent repeats including sampling and training. Expected max fitness and expected mean fitness were evaluated on top 96 variants from supervised learning model. The global maximum hitting rate was evaluated on the union of the top 96 variants from supervised learning model and the 384 variants in training data.

Amino Acid	Molecular Mass (Da)	Hydropathy index	Surface Area (Å^2)	Volume (Å^3)
A	89.094	1.8	115	88.6
R	174.203	-4.5	225	173.4
D	133.104	-3.5	150	111.1
N	132.119	-3.5	160	114.1
C	121.154	2.5	135	108.5
E	147.131	-3.5	190	138.4
Q	146.146	-3.5	180	143.8
G	75.067	-0.4	75	60.1
H	115.156	-3.2	195	153.2
I	131.175	4.5	175	166.7
L	131.175	3.8	170	166.7
K	146.189	-3.9	200	168.6
M	149.208	1.9	185	162.9
F	165.192	2.8	210	189.9
P	115.132	-1.6	145	112.7
S	105.093	-0.8	115	89.0
T	119.120	-0.7	140	116.1
W	204.228	-0.9	255	227.8
Y	181.191	-1.3	230	193.6
V	117.148	4.2	155	140.0
AAIndex Entry [9] or Reference	FASG760101	KYTJ820101	CHOC760101	[10]

Table S5: Physicochemical descriptors for sequence encoding. Amino acids are given by the one letter codes.

ModelClass	SpecificModel	Rounds of Hyperopt
Keras	NoHidden	10
Keras	OneHidden	10
Keras	TwoHidden	10
Keras	OneConv	10
Keras	TwoConv	10
XGBoost	Tree	100
XGBoost	Tree-Tweedie	100
sklearn	GradientBoostingRegressor	100
sklearn	RandomForestRegressor	100
sklearn	ARDRegression	100
sklearn	BaggingRegressor	100
sklearn	LassoLarsCV	100
sklearn	DecisionTreeRegressor	100
sklearn	SGDRegressor	100
sklearn	KNeighborsRegressor	100
sklearn	ElasticNet	100

Table S6: List of models used in the ensemble supervised learning model. NHyperopt shows the rounds of hyper-parameter optimizations performed for the regression model.

Model Class	Specific Model	Tunable parameter	Default value	Hyperopt Search Space	Hyperopt Search Method	Description
Keras	NoHidden	dropout	0.2	0∼0.5	uniform	Dropout value
Keras	OneHidden	dropout	0.2	0∼0.5	uniform	Dropout value
		size1	0.25	0.25∼0.75	uniform	Size of the hidden layer: fraction of the encoding dimension
Keras	TwoHidden	dropout	0.2	0∼0.5	uniform	Dropout value
		size1	0.25	0.25∼0.75	uniform	Size of the first hidden layer: fraction of the encoding dimension
		size2	0.0625	0.03125∼0.25	uniform	Size of the second hidden layer: fraction of the encoding dimension
Keras	OneConv	dropout	0.2	0∼0.5	uniform	Dropout value
		filter_choice	0.5	0.25; 0.5; 0.75; 1	choice	Width of the 1D convolutional window: fraction of the number of mutation sites
		n_filters1	0.0625	0.0625∼0.25	uniform	First layer; Number of filters: fraction of the encoding dimensionality of an amino acid
		flatten_choice	Average	Flatten; Max; Average	choice	Flatten or pooling layer to obtain fully connected layer. Flatten: Flatten layer. M
Keras	TwoConv	dropout	0.2	0∼0.5	uniform	Dropout value
		filter_arch	(0.5,0.5)	(1,0.25); (0.75,0.5); (0.75,0.25); (0.5,0.5); (0.5,0.25)	choice	Widths of the two 1D convolutional windows: fraction of the number of mutation sites, given as a tuple
		n_filters1	0.0625	0.0625∼0.25	uniform	First layer; Number of filters used in the first convolutional layer: fraction of the encoding dimensionality of an amino acid
		n_filters2	1/128	1/256∼0.0625	uniform	Second layer; Number of filters used in the first convolutional layer: fraction of the encoding dimensionality of an amino acid
		flatten_choice	Average	Flatten; Max; Average	choice	The method of flattening post convolution
XGBoost	Tree	eta	0.3	0.01∼0.5	uniform	
		max_depth	6	2∼10	quniform; q=1	
		lambda	1	0∼10	uniform	
		alpha	0	0∼10	uniform	
XGBoost	Tree-Tweedie	eta	0.3	0.01∼0.5	uniform	
		max_depth	6	2∼10	quniform; q=1	
		Lambda	1	0∼10	uniform	
		alpha	0	0∼10	uniform	
sklearn	Gradient-Boosting-Regressor	learning_rate	0.1	0.01∼1	uniform	
		n_estimators	100	10∼500	quniform; q=1	
		min_samples_split	2	0.005∼0.03	uniform	
		min_samples_leaf	1	0.002∼0.03	uniform	
		max_depth	3	3∼10	quniform; q=1	
sklearn	Random-Forest	n_estimators	100	10∼500	quniform; q=1	
		min_samples_split	2	0.005∼0.03	uniform	
		min_samples_leaf	1	0.002∼0.03	uniform	
		max_depth	None	3∼10	quniform; q=1	
sklearn	ARD-Regression	tol	10^{-3}	10^{-5} ∼10^{-3}	uniform	
		alpha_1	10^{-6}	10^{-7}∼10^{-5}	uniform	
		alpha_2	10^{-6}	10^{-7}∼10^{-5}	uniform	
		lambda_1	10^{-6}	10^{-7}∼10^{-5}	uniform	
		lambda_2	10^{-6}	10^{-7}∼10^{-5}	uniform	
sklearn	Bagging-Regression	n_estimators	10	10∼500	quniform; q=1	
		max_samples	1.0	0.1∼1	uniform	
sklearn	Lasso-LarsCV	max_iter	500	10∼1000	quniform; q=1	
		cv	5	2∼10	quniform; q=1	
		max_n_alphas	1000	10∼2000	quniform; q=1	
sklearn	Decision-Tree-Regressor	max_depth	None	3∼10	quniform; q=1	
		min_samples_split	2	0.005∼0.03	uniform	
		min_samples_leaf	1	0.002∼0.03	uniform	

sklearn	SGD-Regressor	alpha	10^{-4}	$10^{-4} \sim 10$	uniform	
		l1_ratio	0.15	$0 \sim 1$	uniform	
		tol	10^{-3}	$10^{-5} \sim 10^{-3}$	uniform	
sklearn	KNeighbors-Regressor	n_neighbors	5	$1 \sim 30$	quniform; q=1	
		weights	uniform	uniform; distance	choice	
		leaf_size	30	$1 \sim 50$	quniform; q=1	
		p	2	$1 \sim 2$	uniform	
sklearn	ElasticNet	l1_ratio	0.5	$0 \sim 1$	uniform	
		alpha	1.0	$10^{-4} \sim 10$	uniform	

Table S7: Tunable parameters in regression models and their search ranges in hyperopt. Hyperopt Search Method indicates the sampling method in the given search space. “uniform” is a value from uniform random sampling in the given range low high. “quiniform” is a round-off value from uniformly random sampling in the given range low high with parameter q: if the sampled value is a, then the round-off value is round(a/q)*q. “choice” returns one of the options in the list given in the search range.

References

[1] Dan Ofer and Michal Linial. ProFET: Feature engineering captures high-level protein functions. *Bioinformatics*, 31(21):3429–3436, 2015.

[2] Alexander G Georgiev. Interpretable numerical descriptors of amino acid space. *Journal of Computational Biology*, 16(5):703–723, 2009.

[3] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[4] Bruce J Wittmann, Yisong Yue, and Frances H Arnold. Machine learning-assisted directed evolution navigates a combinatorial epistatic fitness landscape with minimal screening burden. *bioRxiv*, 2020.

[5] Nikhil Ketkar. Introduction to keras. In *Deep learning with Python*, pages 97–111. Springer, 2017.

[6] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[7] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR, 2013.

[8] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36 (5):411–420, 2018.

[9] Shuichi Kawashima, Hiroyuki Ogata, and Minoru Kanehisa. AAindex: amino acid index database. *Nucleic Acids Research*, 27(1):368–369, 1999.

[10] AA Zamyatnin. Protein volume in solution. *Progress in Biophysics and Molecular Biology*, 24:107–123, 1972.