bioRxiv preprint doi: https://doi.org/10.1101/2021.11.18.468948; this version posted November 19, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

From sequence to yield: deep learning for protein production systems

Evangelos-Marios Nikolados,[1] Oisin Mac Aodha,[2,3] Guillaume Cambray,[4,5] and Diego A. Oyarzún[1,2,3,6]

[1)] *School of Biological Sciences, University of Edinburgh, Edinburgh EH9 3JH, UK*

[2)] *School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK*

[3)] *The Alan Turing Institute, London, NW1 2DB, UK*

[4)] *Diversité des Génomes et Interactions Microorganismes Insectes, University of Montpellier, INRAE UMR 1333, Montpellier, France*

[5)] *Centre de Biologie Structurale, University of Montpellier, INSERM U1054, CNRS UMR5048, Montpellier, France*

[6)] *Corresponding author: d.oyarzun@ed.ac.uk*

Abstract: Recent progress in laboratory automation has enabled rapid and large-scale characterization of strains engineered to express heterologous proteins, paving the way for the use of machine learning to optimize production phenotypes. The ability to predict protein expression from DNA sequence promises to deliver large efficiency gains and reduced costs for strain design. Yet it remains unclear which models are best suited for this task or what is the size of training data required for accurate prediction. Here we trained and compared thousands of predictive models of protein expression from sequence, using a large screen of *Escherichia coli* strains with varying levels of GFP expression. We consider models of increasing complexity, from linear regressors to convolutional neural networks, trained on datasets of variable size and sequence diversity. Our results highlight trade-offs between prediction accuracy, data diversity, and DNA encoding methods. We provide robust evidence that deep neural networks can outperform classic models with the same amount of training data, achieving prediction accuracy over 80% when trained on approximately 2,000 sequences. Using techniques from Explainable AI, we show that deep learning models capture sequence elements that are known to correlate with expression, such as the stability of mRNA secondary structure. Our results lay the groundwork for the more widespread adoption of deep learning for strain engineering across the biotechnology sector.

Keywords: synthetic biology; microbial engineering; deep learning; machine learning; explainable AI

I. INTRODUCTION

Microbial production systems have found applications across many sectors of the economy[1]. In protein manufacture systems, microbial hosts are transformed with heterologous genes that code for a target protein product. A key requirement for strain design is maximization of titers, productivity, and yield. Such optimisation normally relies on the design of bespoke DNA sequences that ensure high transcriptional and translational efficiency[2]. This requires the optimisation of various genetic elements such as codon usage, promoter sequence, and ribosome-binding sequences. But since prediction of protein expression is notoriously challenging, strain development suffers from costly rounds of prototyping and characterization using heuristic rules to navigate the sequence space towards increased production.

Progress in laboratory automation has fuelled the use of deep mutational scanning for the study of genotype-phenotype associations. Several works have combined high-throughput mutagenesis with a diverse range of measurable phenotypes, including protein expression[3–6], ribosome loading[7], and DNA methylation[8,9]. As a result, recent years have witnessed a substantial interest in machine learning methods that leverage such data for sequence-function prediction in natural systems[7,10–12]. For engineered systems, recent works have incorporated machine learning models into the design-build-test cycle of synthetic biology, *e.g.* for predictive modelling of ribosomal binding sequences[13], design of RNA constructs[14], or optimization of regulatory DNA elements[15,16]. Such models can be employed as *in silico* platforms for discovering new variants with improved expression properties, paving the way toward a new level of computer-aided design for production strains. Deep learning algorithms, in particular, can be used to infer relations between sequence and function on a scale that would be impossible to grasp by data inspection alone. Deep learning maximizes the benefits of big data owing to its ability to capture complex dependencies with minimal prior assumptions[17].

Although deep learning models can produce highly accurate predictions[10,18,19], they come at the cost of enormous data requirements for training, typically ranging from tens to hundreds of thousands of sequences. Little attention has been paid to the performance of deep learning in scenarios where the data available for training is far below the requirements of state-of-the-art models. Moreover, we have a poor grasp of what makes a good training dataset or the impact of different components of the model training pipeline, including *e.g.* the choice of DNA encoding and machine learning models.

Here we address these questions by training a large number of machine learning models on datasets of variable size and diversity. We make use of a large screen of GFP-producing strains in *Escherichia coli*[20] that was designed to ensure balanced coverage of the sequence space. These data are particularly useful for comparing machine learning models, because they can be sampled to construct training datasets of varying size and controlled sequence diversity. We considered models of increasing complexity, from linear regressors to deep neural networks, and tested their ability to predict protein expression from DNA sequence in a range of data scenarios. We first trained a large catalogue of classic "non-deep" models and found that these are sufficient for mildly accurate prediction ($R^2 > 50\%$) with fewer than 1,000 sequences for training. We further show that two of the common caveats of deep learn-

bioRxiv preprint doi: https://doi.org/10.1101/2021.11.18.468948; this version posted November 19, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

ing models, namely large data requirements and poor interpretability, can be overcome with careful experimental design and techniques from Explainable AI[21]. Specifically, we provide robust evidence that deep learning models can improve accuracy with the same amount of data as classic models. We routinely obtained near state-of-the-art accuracy ($R^2 > 80\%$) with convolutional neural networks trained on approximately 2,000 sequences. We further show that this high accuracy can be attributed to their ability to preferentially weigh sequence elements known to correlate with low protein expression. Our results provide a comprehensive characterisation of machine learning models for predicting protein expression, with implications for the widespread adoption of deep learning in strain design and optimization.

II. RESULTS

A. Size and diversity of the training data

We sought to compare various machine learning models using datasets of different size and diversity. To this end, we employed the genotype-phenotype association data from Cambray *et al*[20]. This dataset contains fluorescence measurements for over 220,000 GFP-coding sequences expressed in *Escherichia coli*, preceded by a variable 96nt upstream region to control translational efficiency and the resulting expression level. The library of upstream sequences was randomized with a rigorous design-of-experiments approach so as to achieve a balanced coverage of the sequence space and a controlled diversity of variants. Specifically, the 96nt sequences were designed from 56 seeds with maximal pairwise Hamming distances. Each seed was subject to further randomization using the D-Tailor framework[22], so as to generate a mutational series with a controlled coverage of eight biophysical properties. The considered properties describe sequence features at four levels of granularity: nucleotide sequence, codon sequence, amino acid sequence, and secondary mRNA structure (see Fig. 1A).

The complete dataset contains 56 mutational series that provide a global coverage of the sequence space, with each series containing ∼4,000 sequences for local exploration of the vicinity of the seed. The dataset is particularly well suited for our study because it provides access to controllable sequence diversity, as opposed to other screens that consider either random sequences that obscure the effect of few specific mutations, or specific mutants that lack diversity.

To characterise the diversity of the sequence space, we visualised the distribution of overlapping 4-mers using Uniform Manifold Approximation and Projection (UMAP) algorithm for dimensionality reduction[23]. The resulting two-dimensional distribution of sequences (Fig. 1B) shows a clear structure of 56 clusters, each one corresponding to a mutational series. Moreover, as shown in Fig. 1C the protein expression data displays marked qualitative differences across mutational series, including near-Gaussian distributions, as well as left- and right-tailed distributions, bimodal, and near-uniform distributions. This indicates that the dataset is diverse in both genotype and phenotype space, and thus ideally suited for benchmarking different machine learning models.

B. Impact of encoding and sample size of the training sequences

To first understand the baseline performance achievable with classic (*i.e.* "non-deep") machine learning models, we trained a number of models on datasets of varying size with different strategies for DNA encoding (Fig. 2A). Sequence encoding is needed to featurize nucleotide strings into numerical vectors that can be processed by a downstream machine learning regressor. As shown in Table I and Fig. 2A, we considered encodings on three levels: global resolution (biophysical properties), subsequence resolution (overlapping k-mers), and single base resolution (one-hot encoding). Specifically, we considered a global encoding where each sequence is described by the eight biophysical properties in Fig. 1A. At a subsequence resolution, we considered two versions of overlapping k-mer encodings: an ordinal version where each k-mer is assigned a unique integer value between 1 and 4^k, and k-mer counts that contain the number of occurrences of each unique k-mer along the sequence. For base-resolution encodings, we employed two variants of one-hot encoding: binary one-hot where a sequence of length L is encoded as a binary matrix of size $4 \times L$, with each column having a one at a position corresponding to the base in the sequence, and zeros elsewhere; ordinal one-hot encoding assigns a unique integer value to each of the four bases, resulting in encoded vectors of length L. To test the impact of combinations of DNA encodings at different resolutions, we also considered binary one-hot encoding augmented with the biophysical properties from Fig. 1A. Each of the considered encodings produces vector representations of different lengths, as shown in Table I.

DNA encoding	Resolution	Dimension
biophysical properties	global	8
k-mer counts	subsequence	4^k
k-mer ordinal	subsequence	$L-k+1$
one-hot binary	single base	$4L$
one-hot ordinal	single base	L
mixed	global and single base	$4L+8$

TABLE I. **Strategies for encoding DNA sequences.** We considered sequence encodings at three resolutions, which result in encoded vectors of varying dimension; L is sequence length in nucleotides. The biophysical properties are described in Fig. 1A.

We first trained classical machine learning models on five mutational series chosen on the basis of their markedly different expression distributions (shown in Fig. 1C), and using an increasing number of sequences for training (5%, 10%, 25%, 50% and 75% of sequences per series). Given the variation in phenotype distributions, we stratified training samples to ensure that their distribution is representative of the full series. We considered four non-deep models, namely a ridge regressor[24] (a type of penalised linear model) and a multilayer perceptron[25] (MLP, a feed-forward neural network), plus two

bioRxiv preprint doi: https://doi.org/10.1101/2021.11.18.468948; this version posted November 19, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

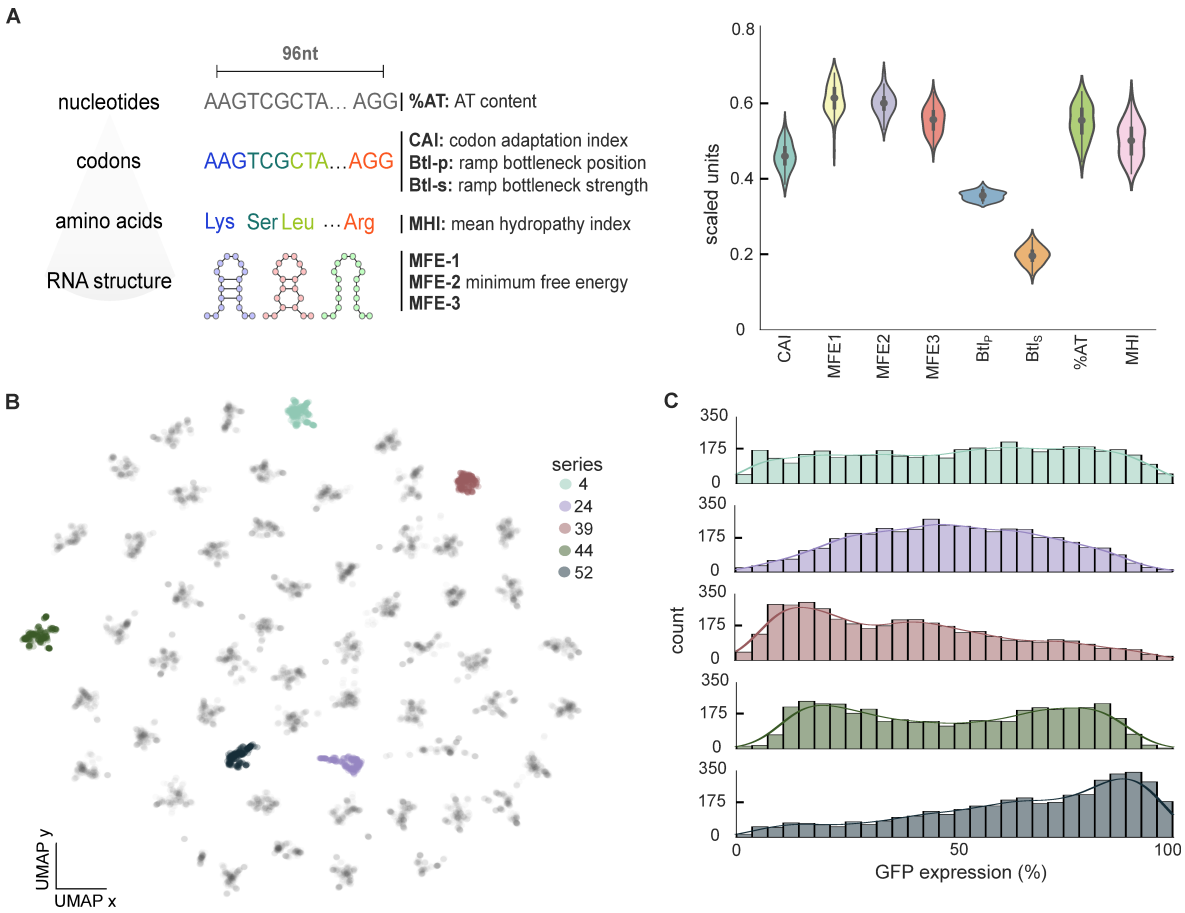

FIG. 1. **Characterization of the training data.** **(A)** We employed a large phenotypic screen in *Escherichia coli*[20] of a GFP coding gene preceded by a variable 96nt sequence. The variable region was designed on the basis of eight sequence properties that affect translational efficiency: nucleotide content (%AT), patterns of codon usage (codon adaptation index, CAI, codon ramp bottleneck position, Btl_P, and strength, Btl_S), hydrophobicity of the polypeptide (mean hydrophobicity index, MHI) and stability of three secondary structures tiled along the transcript (MFE-1, MFE-2, and MFE-3). A total of 56 seed sequences were designed to provide a broad coverage of the sequence space, and then subjected to controlled randomization to create 56 mutational series of ∼4,000 sequences each. Violin plots show the distribution of the average value of the eight properties, for each of the 56 mutational series; values were normalized to the range $[0, 1]$. **(B)** Two dimensional visualization (UMAP[23]) of the overlapping 4-mers computed for the 228,000 sequences in the dataset; this representation reveals 56 clusters, each one corresponding to a mutational series that locally explores the sequence space around its seed. **(C)** Phenotype distributions of five mutational series as measured by FACS-sequencing of GFP fluorescence normalized to its maximal measured value; solid lines correspond to a Gaussian kernel density estimate of the fluorescence distribution.

models that operate by partitioning the feature space: a support vector regressor[26] (SVR, based on linear separation of the feature space) and a random forest regressor[27] (RF, based on a rectangular partition of the feature space). We chose this array of models because they differ in their principle of operation and underlying assumptions on the shape of the feature space.

The training results shown in Fig. 2B reveal a number of insights on the relation between the training data and encoding strategies. We first observe that the phenotype distribution has a minor impact on model accuracy, indicating the benefits of stratified sampling for training. The results show that models trained on a small number of sequences are generally poor ($R^2 \leq 40\%$) irrespective of encoding method and type regressor. Moreover, linear models (ridge) display exceptionally poor accuracy (see example in Fig. 2C) and are insensitive to the size of training set. In contrast, substantial gains in accuracy were achieved with the multilayer perceptron for larger training sets (see example in Fig. 2C), possibly owing to its ability to capture nonlinear relationships in the data. A key finding is that approximately 1000 sequences appear to be sufficient to train mildly accurate models ($R^2 \geq 50\%$), in particular when using regressors that rely on partitioning the feature space (SVR and RF), regardless of the encoding method. We found that random forest regressors are the most accurate among the considered models, consistently achieving $R^2 > 50\%$ for datasets above 1000 samples in the five mutational series (see example in Fig. 2C).

The results in Fig. 2B also underscore the impact of DNA encodings on predictive accuracy. We found that subsequence-resolution encodings achieve varying accuracy that is highly dependent on the specific mutational series and

bioRxiv preprint doi: https://doi.org/10.1101/2021.11.18.468948; this version posted November 19, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

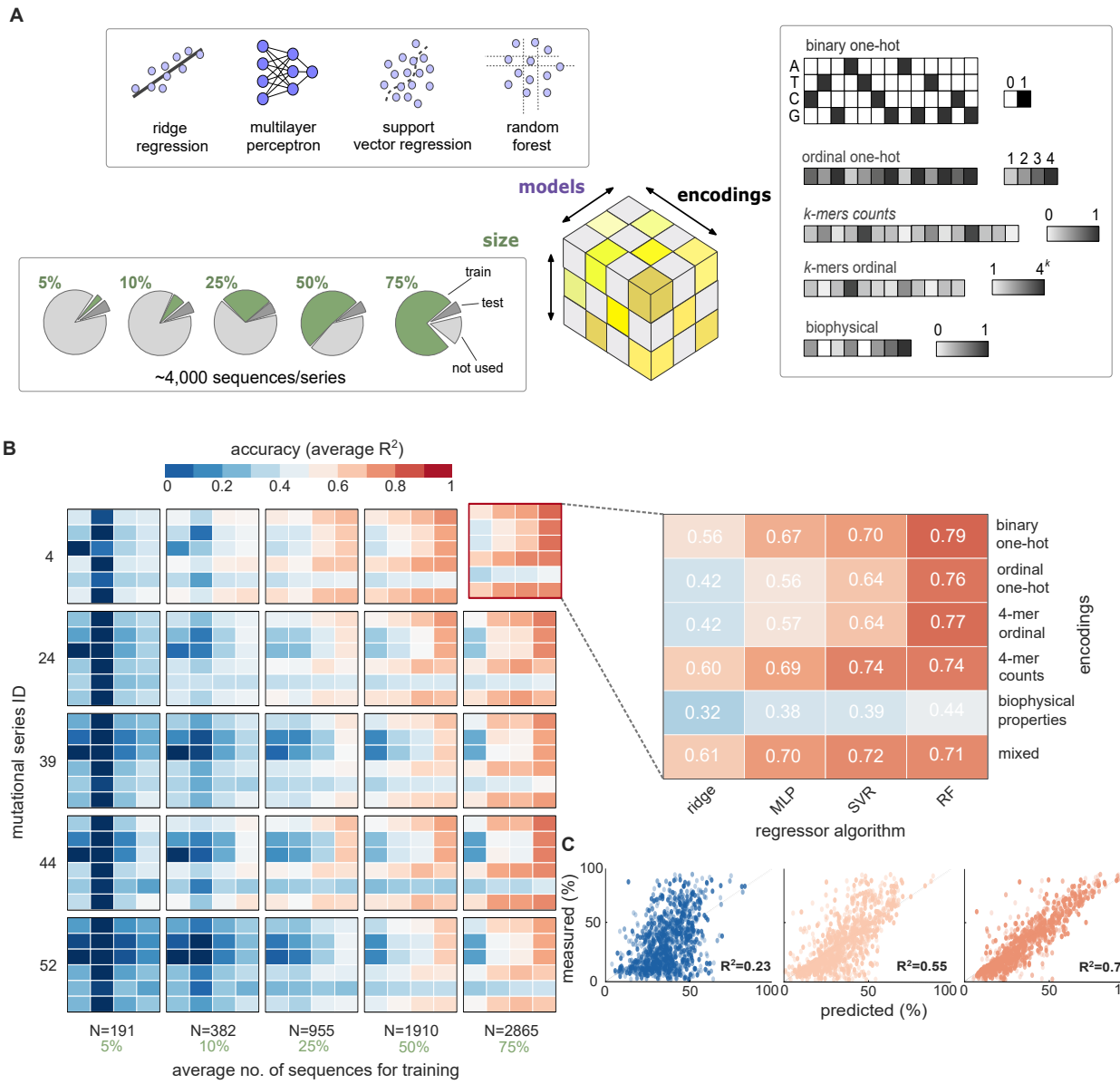

FIG. 2. **Impact of data size and sequence encoding on the accuracy of classic machine learning models.** **(A)** Schematic of our model training strategy. We trained various models using datasets of variable size and with different strategies for DNA encoding. Sequences were converted to numerical vectors with six encoding strategies (see Table I). We considered four classic, non-deep, models trained on an increasing number of sequences from five mutational series with different phenotype distributions (see Fig. 1C); details on model training can be found in the Methods. **(B)** Training results reveal a sizeable impact of DNA encoding and data size on model accuracy. Overall we found that random forest regressors and binary one-hot encodings provide the best accuracy. We quantified model accuracy with the coefficient of determination (R^2) between predicted and measured GFP fluorescence, computed on a fixed held-out dataset with 10% of sequences from each series and averaged across five repeats. **(C)** Examples of poor, mild and good accuracy predictors; plots show predictions on held-out sequences for mutational series 39 with 4-mer ordinal encoding using ridge regression, support vector regression, and random forest models.

chosen model. We observed a consistently poor accuracy in models trained on the sequence biophysical properties; this is surprising because some of them, such as codon usage and the minimum free energy of mRNA secondary structure, are known to correlate with expression[20,28,29]. Moreover, models trained on the one-hot encodings alone performed better than those trained on one-hot encodings augmented with the biophysical properties. These results suggest that such biophysical properties can be deleterious and not sufficiently informative for model training. A possible explanation is that the small number of biophysical features limits are insufficient to account for the diversity of the sequence space. Overall we observed a strong preference for base-resolution encodings, with binary one-hot representations achieving the most the most accurate models, possibly due to their incorporation of positional information.

bioRxiv preprint doi: https://doi.org/10.1101/2021.11.18.468948; this version posted November 19, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

To identify which combinations of encodings and regressors maximize accuracy across a broader space of training sequences, we trained a large catalogue of models on all 56 mutational series. We excluded the ridge regressor due to its poor performance in Fig. 2B and focused on training sets with at least 25% of sequences in each mutational series, corresponding to ∼1,000 sequences per series. We thus trained 1,008 models models (56 series × 6 encodings × 3 models) for training datasets of three different sizes (25%, 50% and 75% of each series), totalling 3,024 models shown in the swarm plots in Fig. 3. Some of the resulting R^2 distributions are heavy tailed and contain a number of outliers; this highlights the importance of contextual effects in the training set, whereby specific combinations of regressors and encodings can lead to exceptionally poor models in specific mutational series. We generally found that the random forest regressor trained on binary one-hot encodings consistently outperformed other models and encodings across most conditions. Moreover, the results also indicate that binary one-hot encoded sequences lead to a more consistent accuracy across the 56 mutational series, as reflected by a much narrower distribution of R^2 values. In agreement with the results of Fig. 2B, these larger scale results reinforce the observation that the biophysical properties are a poor description of DNA sequences for model training. The results achieved with the mixed encoding consistently show that inclusion of biophysical properties leads to poorer models than those trained on one-hot encodings alone.

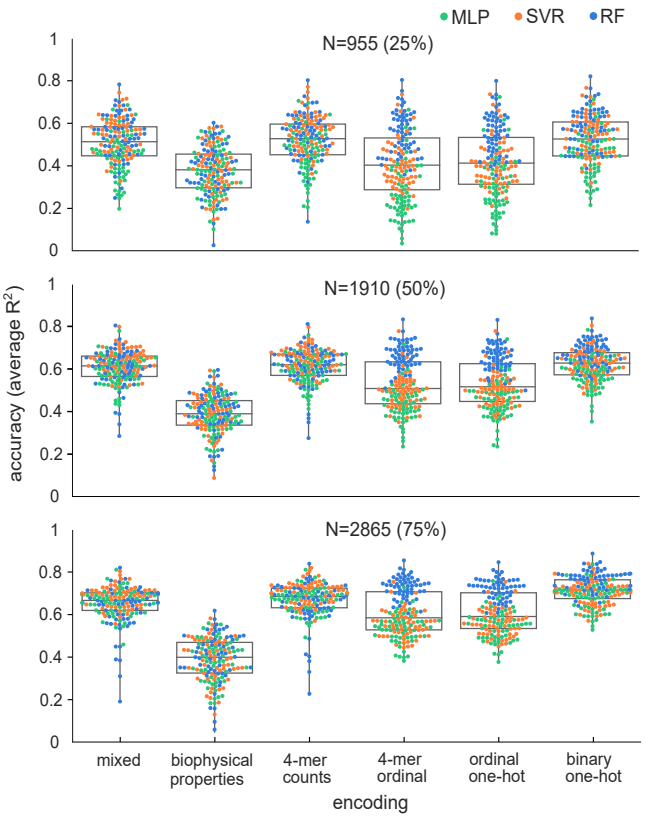

FIG. 3. **Model accuracy across the whole sequence space.** For each mutational series, we trained a total of 54 models (3 regressors × 3 data sizes × 6 encodings). Each dot in the swarm plots represents one of the 3,024 models. The results show that random forests in combination with binary one-hot encoding provide the best overall accuracy, and the least sensitivity to the shape of the sequence space in each series. The biophysical properties from Fig. 1A lead to particularly poor models. As in Fig. 2B, accuracy is reported as the average R^2 across 5 repeats, and computed on a fixed held-out dataset containing 10% of sequences of each mutational series.

C. Deep learning improves accuracy without more data

Prior work has demonstrated that deep learning can produce far more accurate predictions than classic methods[13,15]. An important limitation of current deep learning models, however, is their reliance on extremely large datasets. The latest and most powerful deep learning predictors, such as DeepBIND[10], Optimus 5-prime[7], ExpressionGAN[16], and Enformer[12] were trained with tens and even hundreds of thousands of sequence variants. Data of such size is unlikely to be available in many synthetic biology laboratories and, moreover, it is unclear if the accuracy of deep learning models is a result of their architecture or the size of the training data. We therefore sought to determine the capacity of convolutional neural networks (CNN), a common type of deep learning model, to produce accurate models from much smaller datasets than previously considered.

We designed a convolutional neural network that reads a binary one-hot encoded matrix of dimension 4×96 nucleotides and processes it through three convolutional layers. These layers can be regarded as positional weight matrices acting on an input sequence. By stacking several convolutional layers together, the network can capture interactions between different components of the input sequence. We also added dropout layers for regularization, followed by a multilayer perceptron to integrate information across all positions of the input sequence. The output of the final layer is the predicted level of protein expression. We employed Bayesian Optimization for hyperparameter tuning[30] in conjunction with stochastic gradient descent for training. More details on the network architecture, training and hyperparameters can be found in the Methods.

To perform a thorough comparison between CNNs and other non-deep models, we trained a total of 56 CNNs (one for each mutational series) on 25%, 50% and 75% of all sequences in each series. We benchmarked the accuracy of the resulting CNNs against two classic models: multilayer perceptrons because they are an example of non-deep neural networks, and random forest regressors because we found them to be the most accurate in Fig. 2B–3. The results in Fig. 4A show that CNNs are consistently more accurate than the other models, regardless of the size of the training data. In particular, we observe that some CNNs achieve accuracy over 60% for ∼1000 sequences for training, and in some cases they reach state-of-the-art accuracy ($R^2 > 90\%$) when trained with fewer than 3000 sequences, as shown in the inset of Fig. 4A. These results strongly suggest that deep learning models not only outperform classic methods, but they can do so without the need for additional training data.

bioRxiv preprint doi: https://doi.org/10.1101/2021.11.18.468948; this version posted November 19, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

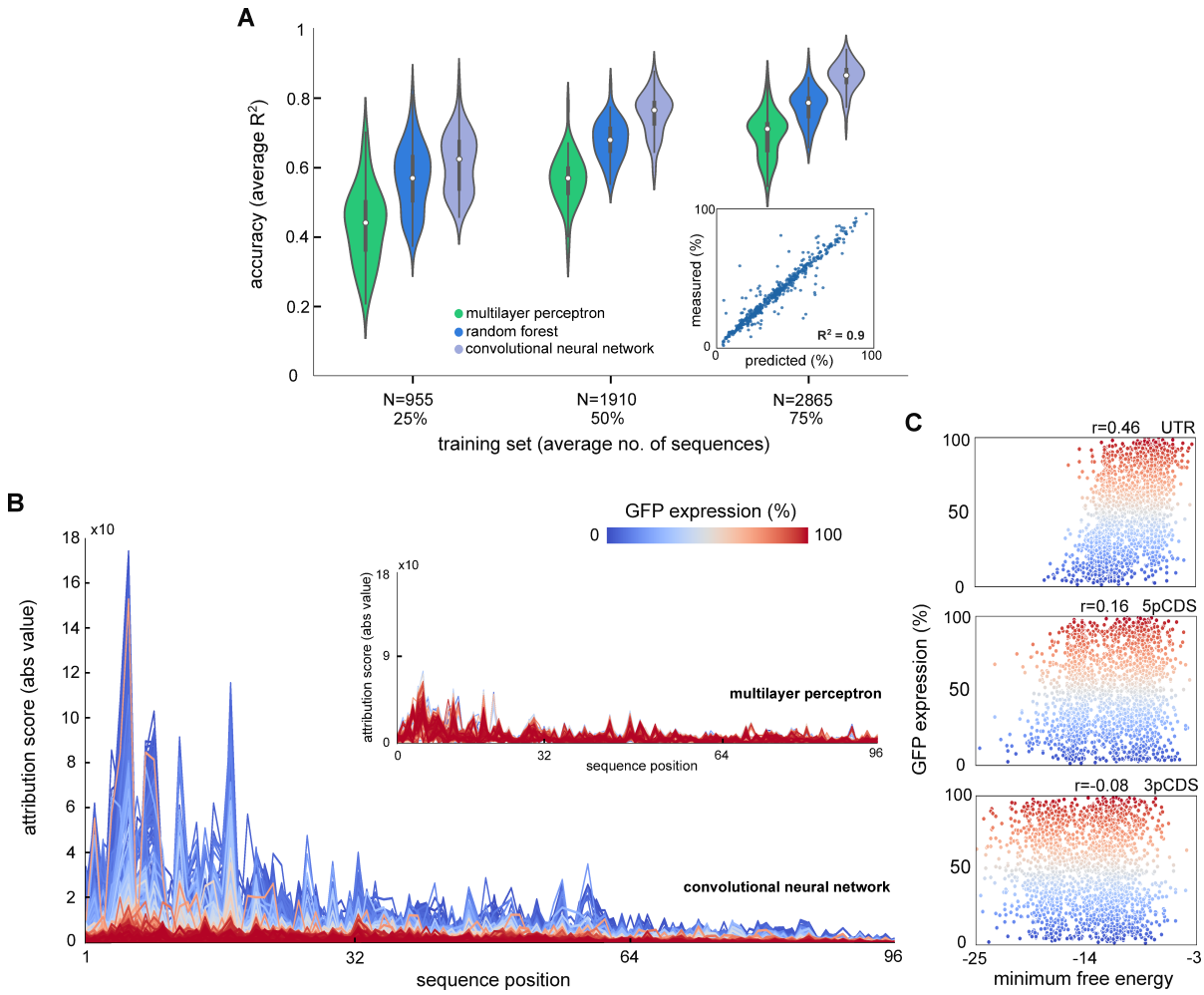

FIG. 4. **Accuracy and interpretability of convolutional neural networks.** **(A)** Comparison of CNNs against multilayer perceptrons and random forest regressors trained on each of the 56 mutational series. Models were trained with binary one-hot encoding on data increasing size. Violin plots show the distribution of average R^2 in 5 training repeats, computed for each of the 56 mutational series on a held-out dataset with 10% of sequences per series. Inset shows predictions of a CNN trained on 75% of mutational series 39 (Fig. 1C), computed on 424 held-out sequences. The CNN consistently displays improved predictive accuracy without the need for more training data. **(B)** Attribution scores computed with DeepLIFT[21] for a CNN trained on 75% of the aggregate of series 2, 27, 32, 50, and 55. Plots show the absolute value of the attribution scores for each position of ∼2120 sequences in a test set; color denotes measured GFP expression. The CNN assigns importance to the translation initiation region (first third of the sequence) of low producers. High producers do not display such localization of attribution scores. We also did not observe any localization for the attribution scores of a multilayer perceptron trained and tested on the same sequences (inset). **(C)** Pearson correlation (r) between GFP expression and the minimum free energy of RNA secondary structures for three tiled regions (see Fig. 1A) of the held-out sequences. The MFE for the first third of the sequence displays much higher correlation than the other two regions. This suggests that the CNN selectively weighs sequence elements that impact expression.

To further understand why CNNs deliver such gains in accuracy, we employed methods from Explainable AI[31] to quantify how different regions of the input sequence are processed by the models. Poor interpretability is a major caveat of deep learning models, as it is challenging to examine the model parameters in a way that they can be meaningfully related to the input sequence. A number of methods have been developed for this task, typically consisting of feature analyses based on activation-maximization and saliency maps[32] to reveal sequence motifs that the models considers relevant. Other approaches employ *in silico* mutagenesis[15] to quantify the sensitivity of predictions to point mutations in the input sequence, yet computational costs limit the number and type of mutations that can be simulated. Here we instead employed DeepLIFT[21], a computationally efficient method that uses backpropagation to produce "attribution scores" for each feature of the input data (details in Methods). Such scores represent the importance that the network assigns to the input features. When applied to one-hot encoded sequences, DeepLIFT produces scores at the resolution of single nucleotides for each sequence employed for training.

We aggregated five mutational series chosen at random, and constructed a new training dataset with 75% of the total number of aggregated sequences, corresponding to ∼19,000 se-

bioRxiv preprint doi: https://doi.org/10.1101/2021.11.18.468948; this version posted November 19, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

quences. We trained a CNN based on the same architecture as in Fig. 4A, as well as a multilayer perceptron for comparison. The resulting CNN achieved an excellent prediction accuracy, with $R^2 \approx 79\%$ on a held-out dataset consisting of 2,037 sequences; the accuracy for the perceptron was 71% when trained and tested on the same data. We ran DeepLIFT on both models and detected stark differences in their attribution scores. As shown in Fig. 4B, the CNN displays high attribution scores localized on the first 32 positions of low producers. High producers, on the contrary, showed consistently low attributions across the whole sequence length. We found no evidence of such localization in the attribution scores of the multilayer perceptron (inset of Fig. 4B). The contrast in positional distributions of scores suggests that the CNN selectively weighs the first third of the sequence in low producing strains. This observation is in agreement with the previous finding from Cambray *et al*[20] that low protein expression correlates with the stability of the mRNA secondary structure of the first third of the sequence. As shown in Fig. 4C, the minimum free energy of the first third of the sequence displays a higher correlation with expression than downstream regions of the sequence. This indicates that the CNN captures the importance of sequence regions that strongly affect protein expression.

D. Impact of sequence diversity on model accuracy and generalization

We finally sought to establish the ability of CNNs to generalize predictions to regions of the sequence space that were not included in training. We found that the CNNs from Fig. 4A, which were trained on a single mutational series each, performed poorly when tested on sequences from other mutational series (not shown). To further understand how to improve cross-series accuracy, we performed a computational experiment designed to test the impact of sequence diversity on the ability of CNNs to produce accurate predictions across different mutational series.

We trained CNN models on datasets of increasing diversity but constant in size. To this end, we trained an initial model on 5,800 sequences sampled from the aggregate of two series chosen at random (series 6 and 49, shown at the bottom row of Fig. 5A). We repeated this strategy so as to successively increase the number of series in the aggregate, while keeping a constant number of total sequences for training. As shown in Fig. 4E, we trained a total of 27 models, each one on an increasingly diverse sequence space, but with a decreasing fraction of sequences per series. For example, model 1 in Fig. 5A, was trained on 2900 sequences per series, out of a total of 2 series, while the last model (top row of Fig. 5A) was trained on as few as $\sim$107 sequences per series, from a total of 54 mutational series.

We compared the accuracy of the 27 models on 10% of held-out sequences from the aggregate employed for training, against 10% of sequences from each series not included in the aggregate. The results (Fig. 5A) show that models trained on aggregate series perform well when tested in-series, even in cases when the training set contains just over a hundred sequences per series; since all these models were trained on the same number of sequences, this suggests that sequence diversity, and not the size of the dataset, is a key factor for accurate predictions. We also found, however, that increasing the diversity of the training set had no impact on model generalization, as we obtained an extremely low accuracy ($R^2 < 10\%$) when testing the models cross-series for all aggregates. This result suggests that model generalization is a key limiting factor when using CNNs to predict protein expression from sequence. The data requirements per series for each model, and the resulting diversity of each aggregate are shown in Fig. 5B.

In agreement with the swarm plots of Fig. 3, in this case we also observed substantial heterogeneity in prediction scores across mutational series. We overall identified three salient patterns: (i) series that are consistently predicted well even under severe data limitations. For example, series 6 and 51 retain $R^2 \approx 70\%$ in most cases, falling to $R^2 \approx 40\%$ only in the last three models, which were trained on fewer than 116 sequences from them. (ii) Series that display good prediction scores only when they are well represented in the training; for example, series 49 is well predicted in the first three models, which were trained on at least 725 sequences from that series, but accuracy drops sharply in subsequent models. (iii) We observed multiple cases of series that appear particularly hard to predict, across all models and irrespective of the number of sequences employed for training; for example, series 21, 27 and 54 display poor R^2 scores in all models. Altogether, these results suggest that the shape of the sequence space can prescribe our ability to train accurate models on them, thus highlighting the importance of thorough use of design-of-experiments when acquiring data for training.

III. DISCUSSION

The ability to predict protein expression from DNA sequence can substantially accelerate the design cycle for many biotechnology applications. An increasing number of works have reported the construction of deep learning models with excellent phenotypic predictions for both natural genes[7,10–12] and synthetic constructs[13–16]. Yet such models have been trained on extremely large datasets that are unrealistic for most design scenarios. In this study we have examined the impact of size and diversity of training data on a large panel of machine learning models. We made extensive use of a genotype-phenotype screen[20] that employed a careful design-of-experiments approach to cover the sequence space with a controlled sequence diversity. By sampling and combining measurements from this large screen, we evaluated the joint effect of models, DNA encodings, and size of the training set on the predictive accuracy.

We found that with such a balanced coverage of the sequence space, accurate predictions can be obtained with training sets of a couple of thousand samples, by using simple random forest regressors trained on one-hot encoded sequences. We demonstrated that such accuracy can be further improved with convolutional neural networks, without the need for more

bioRxiv preprint doi: https://doi.org/10.1101/2021.11.18.468948; this version posted November 19, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

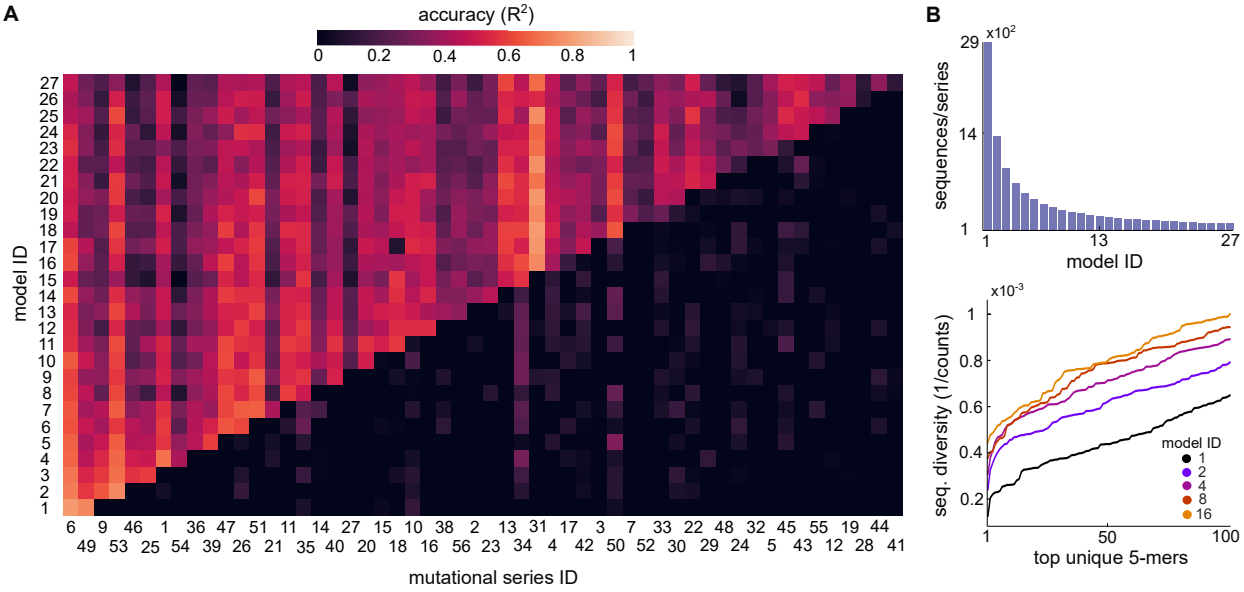

FIG. 5. **Impact of sequence diversity on model generalization. (A)** We trained CNNs on datasets of constant size and increasing diversity. We trained a total of 27 models by successively aggregating fractions of different mutational series into a new dataset for training; the total size of the training was kept constant at 5,800 sequences. Training with aggregated sequences achieves good accuracy for series in the training set, but extremely poor predictions for series not included in the aggregate for training. This suggests poor generalization of the considered CNN models. Accuracy is reported as the R^2 computed on 10% held-out sequences sampled from the series aggregate, and on 10% of each of the series not employed for training. **(B)** Top: data requirements per series for each of the 27 models in panel A. Bottom: increasing diversity of training sets for five models in panel A; diversity was quantified as the inverse of the top unique 5-mers of all sequences in each training set.

sequences for training. This challenges the notion that deep learning models necessarily require extremely large datasets for training. We also demonstrated that methods from Explainable AI[21,31] can provide mechanistic understanding on why deep learning models can produce such gains in prediction accuracy.

A salient result from our analysis is that sequence properties commonly employed for strain optimization, such as the CAI, %AT and other metrics, are particularly poor for model training. This is particularly striking in light of studies that have shown that some of these correlate strongly with expression[28]. In our case, we reasoned that the eight considered properties are insufficient to describe the diversity of the sequence space employed for training, hence leading to poorly predictive models. Other mitigating factors are the limitations on how such metrics are computed. For example, CAI is computed as a geometric average across codons that loses positional information. Likewise, the scores on RNA secondary structures are the result of possibly imprecise free energy calculations on windows of fixed length. Richer descriptions of secondary structures, such as ensemble free energies or probabilities of base pairing at each position, may improve the predictive power of the models.

Although our results suggest that model accuracy depends critically on sequence diversity, and not the number of sequences themselves, we report data requirements above 1,000 sequences that are still too large for most applications. Further work is required on DNA encodings that are maximally informative for model training, as well as model architectures that can deliver high accuracy. Such strategies have provided important gains in data-efficiency for protein design[33], where unsupervised models can be trained on large databases to produce embeddings suitable for prediction. A second important challenge is model generalization, as we consistently observed that models are extremely weak at predicting expression for sequences that are sufficiently divergent from those employed for training. This limitation is particularly relevant for forward-engineering, where designers may utilize model predictions to navigate the sequence space beyond the coverage of the training data. One avenue to address this problem is via design-of-experiments approaches that ensure a coverage of the sequence space that is broad enough for the range of constructs at hand. Such approach can potentially lead to general rules on how to best construct a training set that leads to accurate predictors of expression.

Deep learning models promise to deliver large gains in efficiency in many synthetic biology applications. But this inevitably requires the acquisition of strain characterization data for training. The risk is that the cost of such experimental work solely for the purpose of model training may outweigh the perceived benefits of highly predictive models. Here we have comprehensively mapped the relation between data size, diversity and the choice of machine learning models, laying the foundations for more data-efficient approaches that can promote the adoption of deep learning as a platform technology for strain design.

bioRxiv preprint doi: https://doi.org/10.1101/2021.11.18.468948; this version posted November 19, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

ACKNOWLEDGEMENTS

EMN was supported by a doctoral studentship from the Darwin Trust of Edinburgh.

CONFLICTS OF INTEREST

The authors have no conflicts of interest.

AUTHOR CONTRIBUTIONS

EMN and and DAO designed the research. EMN performed model training and data analysis. GC, OMA and DAO provided input on data analysis and model training. DAO provided overall supervision of the work.

IV. METHODS

A. Data processing

We obtained the full dataset from the OpenScience Framework[34]. After removing sequences with missing values for GFP fluorescence, the dataset contains ∼228,000 sequences with measured GFP fluorescence and the eight biophysicial properties in Fig. 1A. In all trained models, we employed the arithmetic mean of GFP fluorescence across replicates for the case of normal translational initiation[20]. To visualize sequences in a two dimensional space (Fig. 1B), we employed the UMAP algorithm[23] on sequences featurized on counts of overlapping k-mers. We found that the UMAP projection improved for larger k, and chose $k = 4$ to achieve a good trade-off between computation time and quality of projection; k-mer counting was done with custom Python scripts.

B. Model training

Classic machine learning models. Non-deep models in Fig. 2–4 were trained using the scikit-learn package. In all cases, we stratified the protein expression data to ensure that the phenotypic distributions are preserved in the samples employed for training and testing. Stratification was done with the verstack package, which employs binning for continuous variables; we further customised the code to gain control of the binning resolution. For model training, we used varying fractions of the stratified samples, and held-out 10% of sequences for model testing. In all cases except Fig. 5, we did five training repeats by resampling the mutational series; the CNNs in Fig. 5 were trained only once. The accuracy reported in Fig. 2–4 correspond to the average coefficient of determination (R^2) between predicted and measured fluorescence in the test set, averaged across the five training repeats. All DNA encoding strategies (Table I) were implemented with custom Python code. Model hyperparameters were determined via a grid search and 10-fold cross validation. We performed the optimization multiple times and chose a family of configurations that worked equally well across libraries. We repeat this process for all encodings, and finally chose a single best configuration for all mutational series and encoding, see Table II.

Regressor	Hyperparameter	Value
ridge regressor	regularization (α)	$[10^{-1}, 10^{2}]$
support vector regressor	kernel method	RBF
	regularization (C)	30
	margin tolerance (ϵ)	0.5
multilayer perceptron	activation function	ReLU
	hidden layers	3
	no. of neurons	100
	optimizer	Adam
	early stopping	Yes
	learning rate	10^{-3}
random forest	no. of estimators	25
	maximum depth	30
	min samples per leaf	3
	min samples to split	2

TABLE II. **Hyperparameter for the four non-deep regressors.** In all models, except the ridge regressors, we employed the same hyperparameters for all combinations of mutational series and DNA encodings. The regularization strength of the ridge regressor had to be optimized on a case-by-case basis in the range above.

Convolutional neural networks. All CNN models were trained on Tesla K80 GPUs from Google Colaboratory[35]. To design the CNN architectures, we use the Sequential class of the Keras package with the TensorFlow backend[36,37]. Hyperparameters were designed with Bayesian optimization implemented in the HyperOpt package[30]. Since our goal was to use a fixed architecture throughout this paper, we identified a small family of architectures and choose the one that performs best when trained on individual libraries, as well as on aggregates of libraries. We thus performed multiple iterations of the HyperOpt routine using stratified sets, and tested combinations of various hyperparameters. The resulting model architecture is outlined in Table III. All CNNs were trained on binary one-hot encoded sequences with mean squared error as the loss function. We used a batch size of 64, set the learning rate to 10^{-3}, and employ the Adam optimizer[38]. Since ADAM computes adaptive learning rates for each weight of the neural network, we found that the default options were adequate and hence did not specify a learning rate schedule. We set the maximum number of epochs to 100, and used the number of epochs without loss improvement over the validation set as our early stopping criterion (15 epochs) to prevent overfitting.

C. Interpretability analysis

We used DeepLIFT[21], a back-propagation based approach that produces attribution scores of inputs based on a reference input. For both the CNN and MLP in Fig. 4B, we chose a blank sequence as a reference, *i.e.* a 4×96 matrix filled with zeroes. We used the *GenomicsDefault* option that im-

bioRxiv preprint doi: https://doi.org/10.1101/2021.11.18.468948; this version posted November 19, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Blocks	Hyperparameter	Value
Convolutional (1-3)	number of filters	256
	filter width	13
	activation	ReLU
	dropout prob.	0.15
	max-pooling	(2,2)
Dense (4-7)	hidden units	256
	activation	ReLU
	dropout prob.	0.1
Dense (final)	unit	1

TABLE III. **Hyperparameters for the CNN architecture.** We used the same architecture and hyperparameters in the models trained in Fig. 4–5.

plements the *Rescale* and *RevealCancel* rules for convolutional and dense layers, respectively. This process outputs 2,120 (424 test sequences for five mutational series) vectors of length 96, containing an attribution score per base. Finally, we use absolute values for Fig. 4B, since we are interested only in the size of the resulting attribution.

D. Impact of sequence diversity

The models in Fig. 5 were trained on data of constant size and increasing sequence diversity. We successively aggregated fractions of mutational series to create new training sets with improved diversity. We employed the same training strategy as in Fig. 4A with the hyperparameters outlined in Table III for all 27 models. To ensure a comparison solely on the basis of diversity, we fixed the size of the training set to $5,800$ sequences. To increase diversity, for successive models we sampled training sequences from two additional series, as shown in Fig. 5. The specific series for the aggregates were randomly chosen. Model accuracy was evaluated on 10% of held-out sequences from the aggregates employed for training, and 10% of each series not employed for training.

REFERENCES

[1] K. Terpe, Applied Microbiology and Biotechnology 2006 72:2 **72**, 211 (2006).
[2] H. P. Sørensen and K. K. Mortensen, Journal of biotechnology **115**, 113 (2005).
[3] J. B. Kinney, A. Murugan, C. G. Callan, and E. C. Cox, Proceedings of the National Academy of Sciences **107**, 9158 (2010).
[4] E. Sharon, Y. Kalma, A. Sharp, T. Raveh-Sadka, M. Levo, D. Zeevi, L. Keren, Z. Yakhini, A. Weinberger, and E. Segal, Nature biotechnology **30**, 521 (2012).
[5] S. Kosuri, D. B. Goodman, G. Cambray, V. K. Mutalik, Y. Gao, A. P. Arkin, D. Endy, and G. M. Church, Proceedings of the National Academy of Sciences **110**, 14024 (2013).
[6] C. G. de Boer, E. D. Vaishnav, R. Sadeh, E. L. Abeyta, N. Friedman, and A. Regev, Nature biotechnology **38**, 56 (2020).
[7] P. J. Sample, B. Wang, D. W. Reid, V. Presnyak, I. J. McFadyen, D. R. Morris, and G. Seelig, Nature biotechnology **37**, 803 (2019).
[8] M. d. Raad, C. Modavi, D. J. Sukovich, and J. C. Anderson, ACS chemical biology **12**, 191 (2017).
[9] E. Yus, J.-S. Yang, A. Sogues, and L. Serrano, Nature communications **8**, 1 (2017).
[10] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, Nature biotechnology **33**, 831 (2015).
[11] J. A. Valeri, K. M. Collins, P. Ramesh, M. A. Alcantar, B. A. Lepe, T. K. Lu, and D. M. Camacho, Nature communications **11**, 1 (2020).
[12] Ž. Avsec, V. Agarwal, D. Visentin, J. R. Ledsam, A. Grabska-Barwinska, K. R. Taylor, Y. Assael, J. Jumper, P. Kohli, and D. R. Kelley, Nature Methods 2021 18:10 **18**, 1196 (2021).
[13] S. Höllerer, L. Papaxanthos, A. C. Gumpinger, K. Fischer, C. Beisel, K. Borgwardt, Y. Benenson, and M. Jeschek, Nature communications **11**, 1 (2020).
[14] N. M. Angenent-Mari, A. S. Garruss, L. R. Soenksen, G. Church, and J. J. Collins, Nature communications **11**, 1 (2020).
[15] J. T. Cuperus, B. Groves, A. Kuchina, A. B. Rosenberg, N. Jojic, S. Fields, and G. Seelig, Genome research **27**, 2015 (2017).
[16] J. Zrimec, X. Fu, A. S. Muhammad, C. Skrekas, V. Jauniskis, N. K. Speicher, C. S. Börlin, V. Verendel, M. H. Chehreghani, D. Dubhashi, V. Siewers, F. David, J. Nielsen, and A. Zelezniak, bioRxiv , 2021.07.15.452480 (2021).
[17] D. M. Camacho, K. M. Collins, R. K. Powers, J. C. Costello, and J. J. Collins, Cell **173**, 1581 (2018).
[18] J. Zhou and O. G. Troyanskaya, Nature methods **12**, 931 (2015).
[19] D. R. Kelley, J. Snoek, and J. L. Rinn, Genome research **26**, 990 (2016).
[20] G. Cambray, J. C. Guimaraes, and A. P. Arkin, Nature biotechnology **36**, 1005 (2018).
[21] A. Shrikumar, P. Greenside, and A. Kundaje, 34th International Conference on Machine Learning, ICML 2017 **7**, 4844 (2017), 1704.02685.
[22] J. C. Guimaraes, M. Rocha, A. P. Arkin, and G. Cambray, Bioinformatics **30**, 1087 (2014).
[23] L. McInnes, J. Healy, and J. Melville, arXiv preprint arXiv:1802.03426 (2018).
[24] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction* (Springer Science & Business Media, 2009).
[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," Tech. Rep. (California Univ San Diego La Jolla Inst for Cognitive Science, 1985).
[26] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik, *et al.*, Advances in neural information processing systems **9**, 155 (1997).
[27] L. Breiman, Machine learning **45**, 5 (2001).
[28] G. Kudla, A. W. Murray, D. Tollervey, and J. B. Plotkin, science **324**, 255 (2009).
[29] T. E. Quax, N. J. Claassens, D. Söll, and J. van der Oost, Molecular Cell **59**, 149 (2015).
[30] J. Bergstra, D. Yamins, and D. Cox, in *International conference on machine learning* (2013) pp. 115–123.
[31] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: interpreting, explaining and visualizing deep learning*, Vol. 11700 (Springer Nature, 2019).
[32] J. Lanchantin, R. Singh, B. Wang, and Y. Qi, Pacific Symposium on Biocomputing **0**, 254 (2017), 1608.03644.
[33] S. Biswas, G. Khimulya, E. C. Alley, K. M. Esvelt, and G. M. Church, Nature methods **18**, 389 (2021).
[34] https://osf.io/a56vu/.
[35] https://colab.research.google.com/.
[36] F. Chollet *et al.*, https://keras.io **7** (2015).
[37] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous systems," (2015).
[38] D. P. Kingma and J. Ba, arXiv preprint arXiv:1412.6980 (2014).