nature methods

ARTICLES

https://doi.org/10.1038/s41592-021-01100-y

Check for updates

Low-*N* protein engineering with data-efficient deep learning

Surojit Biswas[1,2,6], Grigory Khimulya[3,6], Ethan C. Alley[4,6], Kevin M. Esvelt[4] and George M. Church[1,5]✉

Protein engineering has enormous academic and industrial potential. However, it is limited by the lack of experimental assays that are consistent with the design goal and sufficiently high throughput to find rare, enhanced variants. Here we introduce a machine learning-guided paradigm that can use as few as 24 functionally assayed mutant sequences to build an accurate virtual fitness landscape and screen ten million sequences via in silico directed evolution. As demonstrated in two dissimilar proteins, GFP from *Aequorea victoria* (avGFP) and *E. coli* strain TEM-1 β-lactamase, top candidates from a single round are diverse and as active as engineered mutants obtained from previous high-throughput efforts. By distilling information from natural protein sequence landscapes, our model learns a latent representation of 'unnaturalness', which helps to guide search away from non-functional sequence neighborhoods. Subsequent low-*N* supervision then identifies improvements to the activity of interest. In sum, our approach enables efficient use of resource-intensive high-fidelity assays without sacrificing throughput, and helps to accelerate engineered proteins into the fermenter, field and clinic.

Protein engineering holds great promise for nanotechnology, agriculture and medicine. However, design is limited by our ability to search through the vastness of protein sequence space, which is only sparsely functional[1,2]. When searching for high-functioning sequences, engineers must be wary of the pervasive maxim 'you get what you screen for', which cautions against overoptimizing a protein's sequence using functional assays that may not be fully aligned with the final design objective[3–6]. However, in most resource-constrained real-world settings, including the design of protein therapeutics[7,8], agricultural proteins[9] and industrial biocatalysts[10,11], engineers must often compromise assay fidelity (careful endpoint-resembling measurements of a small number of variants) for assay throughput (high-throughput proxy measurements for a large number of variants)[12,13]. Consequently, the best candidates identified by early-stage high-throughput (>10^4 variants) proxy experiments[9,11,14] will often fail in validation under higher-fidelity, later-stage assays[13,15–17]. Moreover, high-throughput assays do not exist at all for many classes of proteins, making them inaccessible to screening and directed evolution[18–24].

Here we focus on enabling large-scale exploration of sequence space using only a small number ('low *N*') of functionally characterized training variants. We recently developed UniRep[25], a deep learning model trained on a large unlabeled protein sequence dataset. From scratch and from sequence alone, UniRep learned to distill the fundamental features of a protein, including biophysical, structural and evolutionary information, into a holistic statistical summary, or *representation*.

We reasoned that combining UniRep's global knowledge of functional proteins with just a few dozen functionally characterized mutants of the target protein might suffice to build a high-quality model of a protein's fitness landscape. Combined with in silico directed evolution, we hypothesized that we could computationally explore these landscapes at a scale of 10^7–10^8 variants, rivaling even the highest-throughput screens. Here, we test this paradigm in two fundamentally different proteins, eukaryotic avGFP and a prokaryotic β-lactam-hydrolyzing enzyme from *Escherichia coli* (TEM-1 β-lactamase). We demonstrate reliable production of substantially optimized designs with only 24 or 96 characterized sequence variants as training data.

Results

A paradigm for low-*N* protein engineering. To meet the enormous data requirement of supervised deep learning (typically greater than 10^6 labeled data points[26,27]) current machine learning-guided protein-design approaches must gather high-throughput experimental data[28–31] or abandon deep learning altogether[18,20,21,32–37]. We reasoned that we could leverage UniRep's existing knowledge of functional protein sequences to substantially reduce this prohibitive data requirement and enable low-*N* design.

For low-*N* engineering of a given target protein, our approach features five steps (Fig. 1).

1. Global unsupervised pre-training of UniRep on >20 million raw amino acid sequences to distill general features of all functional proteins as described previously[25] (Fig. 1a).
2. Unsupervised fine tuning of UniRep on sequences related to the target protein (evotuning) to learn the distinct features of the target family. We call this model, which combines features from both the global and local sequence landscape, evotuned UniRep or eUniRep (Fig. 1b).
3. Functional characterization of a low-*N* number of random mutants of the wild-type (WT) target protein to train a simple supervised top model that uses eUniRep's representation as input (Fig. 1c). Together, eUniRep and the top model define an end-to-end sequence-to-function model that serves as a surrogate of the protein's fitness landscape.
4. Markov chain Monte Carlo-based in silico directed evolution on this surrogate landscape (Fig. 1d,e).

[1]Wyss Institute for Biologically Inspired Engineering, Harvard University, Boston, MA, USA. [2]Nabla Bio, Inc., Boston, MA, USA. [3]Telis Bioscience Inc., Boston, MA, USA. [4]MIT Media Lab, Massachusetts Institute of Technology, Cambridge, MA, USA. [5]Department of Genetics, Harvard Medical School, Boston, MA, USA. [6]These authors contributed equally: Surojit Biswas, Grigory Khimulya, Ethan C. Alley. ✉e-mail: gchurch@genetics.med.harvard.edu

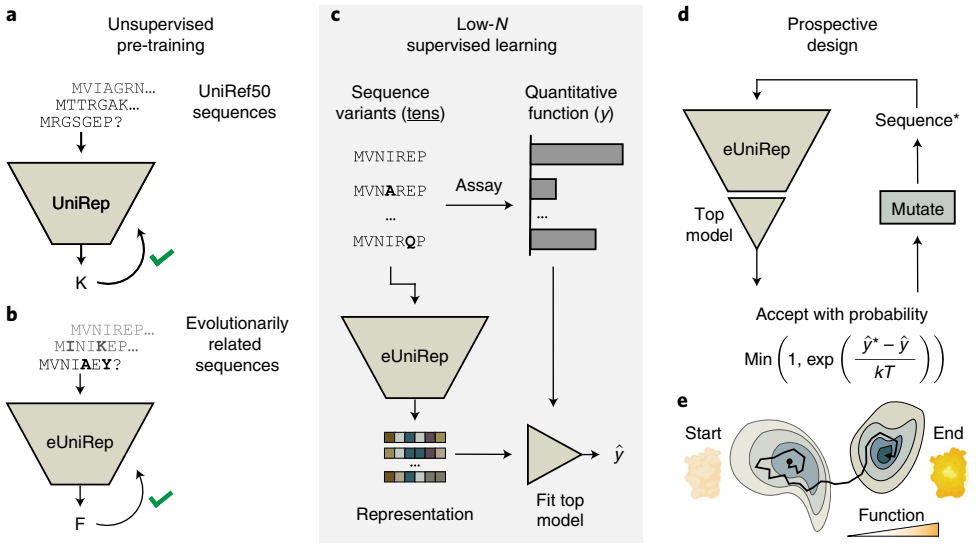

Fig. 1 | UniRep-guided in silico directed evolution for low-*N* protein engineering. a, UniRep is globally trained on a large sequence database (UniRef50) as described previously[25]. **b**, This trained, unsupervised model is further fine tuned to sequences that are related to the protein of engineering interest (eUniRep). **c**, A low-*N* number of mutants are obtained, characterized and used to train regularized linear regression 'on top' of eUniRep's representation. **d**, In silico directed evolution is used to navigate this virtual fitness landscape and propose putatively optimized designs, which are then experimentally characterized. This design loop may be repeated until desired functionality is reached. **e**, Illustration of the evolutionary process.

5. Experimental characterization of top sequence candidates that are predicted to have improved function relative to WT (>WT).

To understand the utility of eUniRep's global and local representation, we considered a control model that was trained de novo solely on the local sequence neighborhood[38–41] of the target protein (Local UniRep). Thus, Local UniRep lacks global information about all known sequence space. As an additional control, we included one-hot encoding, as an explicit and exact flattened binary matrix representation of the full amino acid sequence (Full AA), to contextualize the importance of any local sequence information (Methods).

We first evaluated our approach in retrospective experiments using pre-existing and newly designed datasets of characterized mutant proteins (Methods and Supplementary Fig. 1). We found that only globally pre-trained eUniRep enabled consistent low-*N* retrospective performance and that, with the right regularized top model, meaningful generalization required only 24 training mutants (Supplementary Fig. 2). Random selection of these 24 mutants from the output of error-prone PCR or single-mutation deep mutational scans worked as well as more tailored approaches (Methods). We note that these mutagenesis strategies most often produce variants with impaired activity. Thus, generalizing from these to >WT variants is non-trivial.

Low-*N* engineering of the fluorescent protein avGFP. To test our approach prospectively, we attempted low-*N* optimization of the fluorescence intensity of the original avGFP (Fig. 2a). The design process consisted of randomly sampling $N=24$ or $N=96$ training mutants from error-prone PCR[42], representing sequences, training a top model and performing in silico directed evolution to produce 300 putatively optimized designs within a 15 mutation 'trust radius' of WT (Methods). We replicated this process five times for each *N* and representation model, yielding a total of 12,000 sequence designs. The design window spanned an 81-amino acid region of avGFP that included the central chromophore-bearing helix and four straddling β-sheets (Fig. 2a, Methods and Supplementary Fig. 3).

Evotuning globally pre-trained UniRep was reproducible, and, in 19 of 20 replicates (95%), eUniRep enabled an overall 10% ± 2% (95% confidence interval) hit rate, defined as designs with activity greater than WT (>WT; eUniRep 1 and 2; Fig. 2b). For designs with three or fewer mutations, hit rates were 20–65% and were substantially higher than those from error-prone PCR mutagenesis (Supplementary Fig. 4b,d), a typical starting point for directed evolution. Unexpectedly, maximal activity improvements (nearly 10× WT levels) were observed for designs containing three to seven mutations, even though they had lower hit rates (5–25%). This reflects a risk–reward trade-off that eUniRep can exploit and would be challenging to achieve with directed evolution (Supplementary Fig. 4a,c).

Repeating prospective design while constraining in silico evolution to a seven-mutation trust radius improved eUniRep's overall hit rate to 18% without loss of quantitative fluorescence (Supplementary Fig. 5). Based on these numbers, '24-to-24 design' appeared tractable, in which the characterization of just 24 training mutants and 24 optimized designs would be sufficient to observe a >WT design 1.8 ± 0.8 (95% confidence interval) times (Supplementary Fig. 6). By contrast, prospective design on Full AA or Local UniRep was inconsistent and only enabled ~0% and ~2% hit rates, respectively, highlighting the importance of both global and local unsupervised training.

We clonally validated our best designs and compared them to sequences produced by ancestral sequence reconstruction (ASR)[43,44] and consensus sequence design[45,46] (Methods). While both consistently provided >WT variants, eUniRep designs were substantially more functional (Fig. 2c). Several, in fact, were on par with superfolder GFP (sfGFP; Fig. 2c), which is the result of a multi-year engineering effort that started with avGFP and benefits from mutations outside of our design window. Importantly, eUniRep designs were diverse and occupied a unique region of sequence space, different from evotuning, ASR and consensus sequences (median minimum number of mutations = 5, Fig. 2d).

Importantly, eUniRep's design performance could not be explained by a simple tendency to guide search toward the evotuning or low-*N* training sequences. First, the vast majority (>99%) of evotuning sequences had less than 28% sequence similarity with avGFP (>170 mutations; Supplementary Fig. 7a,b). Furthermore, of all mutations present in >WT eUniRep designs, approximately

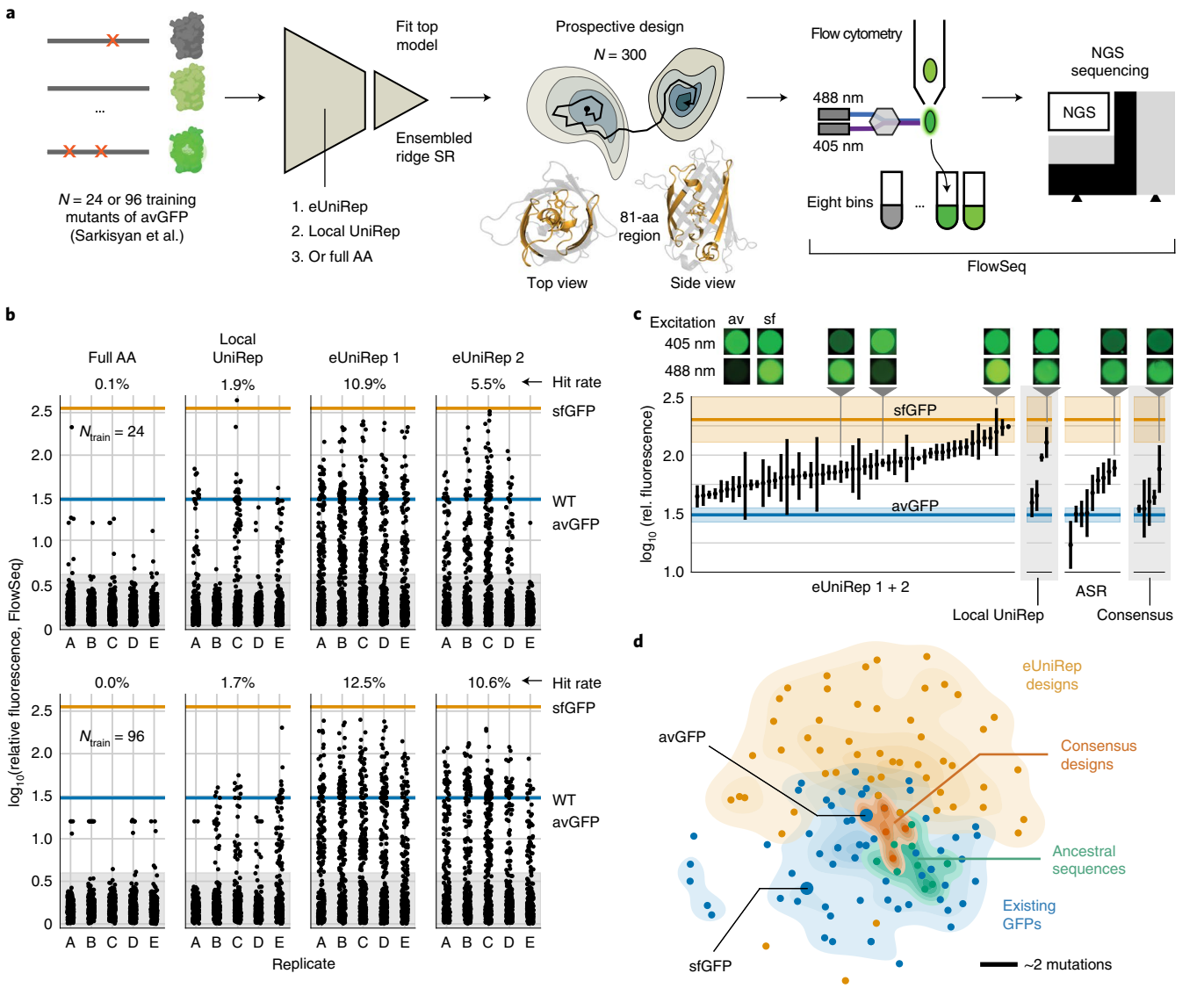

Fig. 2 | eUniRep enables low-*N* engineering of avGFP. a, Experimental workflow describing training mutant acquisition, sequence-to-function modeling, in silico directed evolution and the use of FlowSeq to quantitatively characterize designs in multiplex. aa, amino acid. **b**, Low-*N* engineering results for 24 (top) and 96 (bottom) training mutants. eUniRep 1 and 2 correspond to two replicate evotunings initialized from the same globally pre-trained UniRep. **c**, Quantitative flow cytometric measurements of top eUniRep and Local UniRep designs, as well as those of ASR and consensus sequence designs. Shown above are false-colored images of *E. coli* expressing avGFP (av), sfGFP (sf) and a subset of the designs under excitation at 405 nm or 488 nm, read with a 525/50-nm emission filter. Rel., relative. **d**, Distance-preserving multidimensional scaling plot illustrating the diversity of eUniRep designs compared to existing GFPs, ASRs and consensus sequence designs. A scale bar of two mutations is shown.

25% were new (defined to be neither found among the evotuning sequences nor the low-*N* training sequences) (Supplementary Fig. 8a). For the remaining 75% of shared mutations, abundance among evotuning or low-*N* training sequences was a poor predictor of abundance among >WT eUniRep designs (Spearman $\rho = -0.24$). Finally, 89% of all >WT eUniRep designs contained at least one new mutation, with many of the most active designs containing 33–66% new mutations (Supplementary Fig. 8b,d). As these analyses only consider simple 'first-order' mutational overlap, they provide a lower bound on non-triviality. Indeed, due to epistasis, even recombining existing mutations among homologs to produce functional proteins is a difficult challenge[47].

Through a retrospective analysis (Methods), we found evotuning to be robust to the size of the evotuning sequence set as well as to the number of model updates performed during training (Supplementary Fig. 9a). Specifically, our models were equally performant even with 30% of the full sequence data used for evotuning and half of the model updates.

Low-*N* engineering of the enzyme TEM-1 β-lactamase. We next challenged our approach to generalize to the enzyme TEM-1 β-lactamase and optimize protein function training only on single mutants, which lack epistatic information[48]. Not only is this an arduous task due to the essential role of epistasis in proteins[49,50], but, also, TEM-1 β-lactamase is dissimilar to avGFP both evolutionarily (eukaryotic versus prokaryotic) and functionally (fluorescence versus hydrolysis). Additionally, unlike GFP, our measure of TEM-1 β-lactamase function is only observable through organism-level fitness (Methods), which is an indirect, endpoint measure that depends on the activity of other proteins (for example, peptidoglycan-forming DD-carboxypeptidases and peptidoglycan transpeptidases). Finally, we note that low-*N*

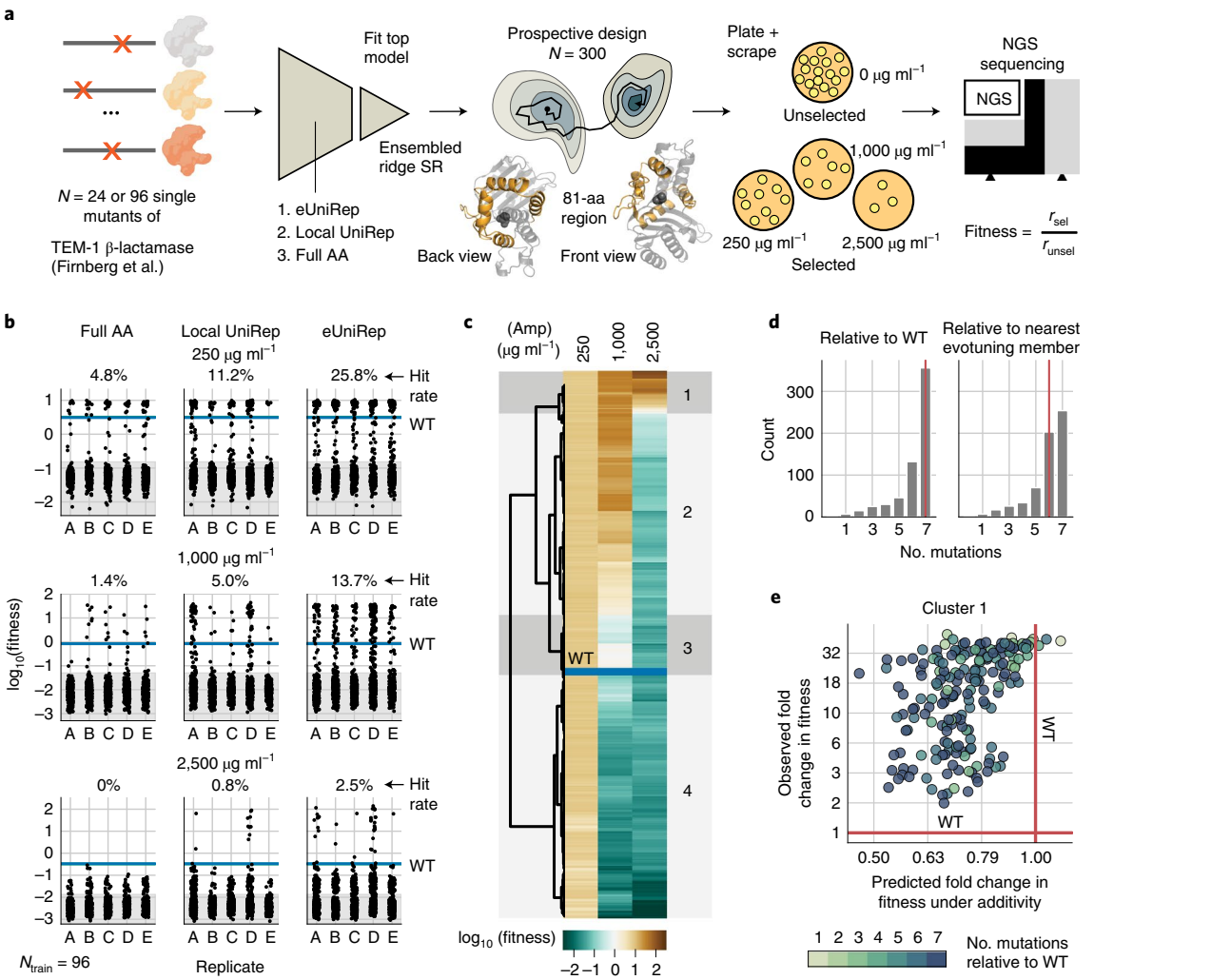

Fig. 3 | eUniRep enables low-*N* engineering of the enzyme TEM-1 β-lactamase using only single mutants as training data. a, Experimental workflow describing training mutant acquisition, sequence-to-function modeling, in silico directed evolution and plate-based antibiotic selection combined with next-generation sequencing (NGS) to characterize designs. r_{sel}, relative abundance under selection; r_{unsel}, relative abundance under no selection. **b**, Low-*N* engineering results using $N = 96$ training mutants for three different antibiotic selection conditions. **c**, Heatmap illustrating $\log_{10}$(fitness) of all >WT eUniRep designs. Four clusters are annotated. [Amp], concentration of ampicillin. **d**, Bar plots illustrating the number of mutations of eUniRep designs relative to WT (left) and to the nearest member of the evotuning sequence set (right). **e**, Scatterplot of eUniRep cluster 1 (highly >WT) designs illustrating observed fold change in fitness (relative to WT) versus predicted fold change in fitness under additivity.

engineering is particularly desirable for enzyme biocatalysts[18], of which β-lactamase is a model. Here, high-throughput assays are frequently intractable due to the difficulty of intracellularly reporting on enzyme activity.

We performed low-*N* optimization of TEM-1 β-lactamase fitness at three concentrations of the antibiotic ampicillin (250, 1,000 or 2,500 μg ml^{-1}) using single mutants as training data (Fig. 3a, Methods and Supplementary Fig. 10)[48]. We designed an 81-amino acid region spanning four helices that straddle but do not include the central helix bearing the catalytic serine S70 (Fig. 3a). Designs were proposed with a seven-mutation trust radius (Methods). As carried out for GFP, we generated 300 designs for each N_{train} and representation model and replicated this process five times.

eUniRep consistently enabled a 5–10× and 2–3× higher hit rate than Full AA and Local UniRep, respectively (Fig. 3b). eUniRep's relative performance improved to a 5–9× gain over Local UniRep for training sets of size $N = 24$ (Supplementary Fig. 11), and, except at the most stringent antibiotic concentration, eUniRep's performance was robust and consistent across training sets.

Importantly, eUniRep designs were diverse both in function and in sequence (Fig. 3c,d). As observed for GFP, eUniRep >WT designs diverged substantially from WT (median number of mutations = 7) and from any evotuning set sequences (median minimum number of mutations = 6) (Fig. 3d).

We note that the majority (>89%) of evotuning sequences had less than 28% sequence identity with WT TEM-1 β-lactamase (>204 mutations; Supplementary Fig. 7c). On average, 18% of the mutations found in eUniRep >WT designs were new (Supplementary Fig. 12a), and of those that were not, abundance among the evotuning sequences was not a strong predictor of abundance among designs (Spearman $\rho = 0.1$). Additionally, 97% of >WT designs contained at least one new mutation, and many of the most active designs contained 30–70% new mutations (Supplementary Fig. 12b). Therefore, as with GFP, it is unlikely that the higher hit rates of eUniRep are explained by a simple tendency to guide search toward sequences in the evotuning or low-*N* training sequence sets.

Notably, despite being generated from single-mutant training data, eUniRep's >WT designs were epistatically non-trivial

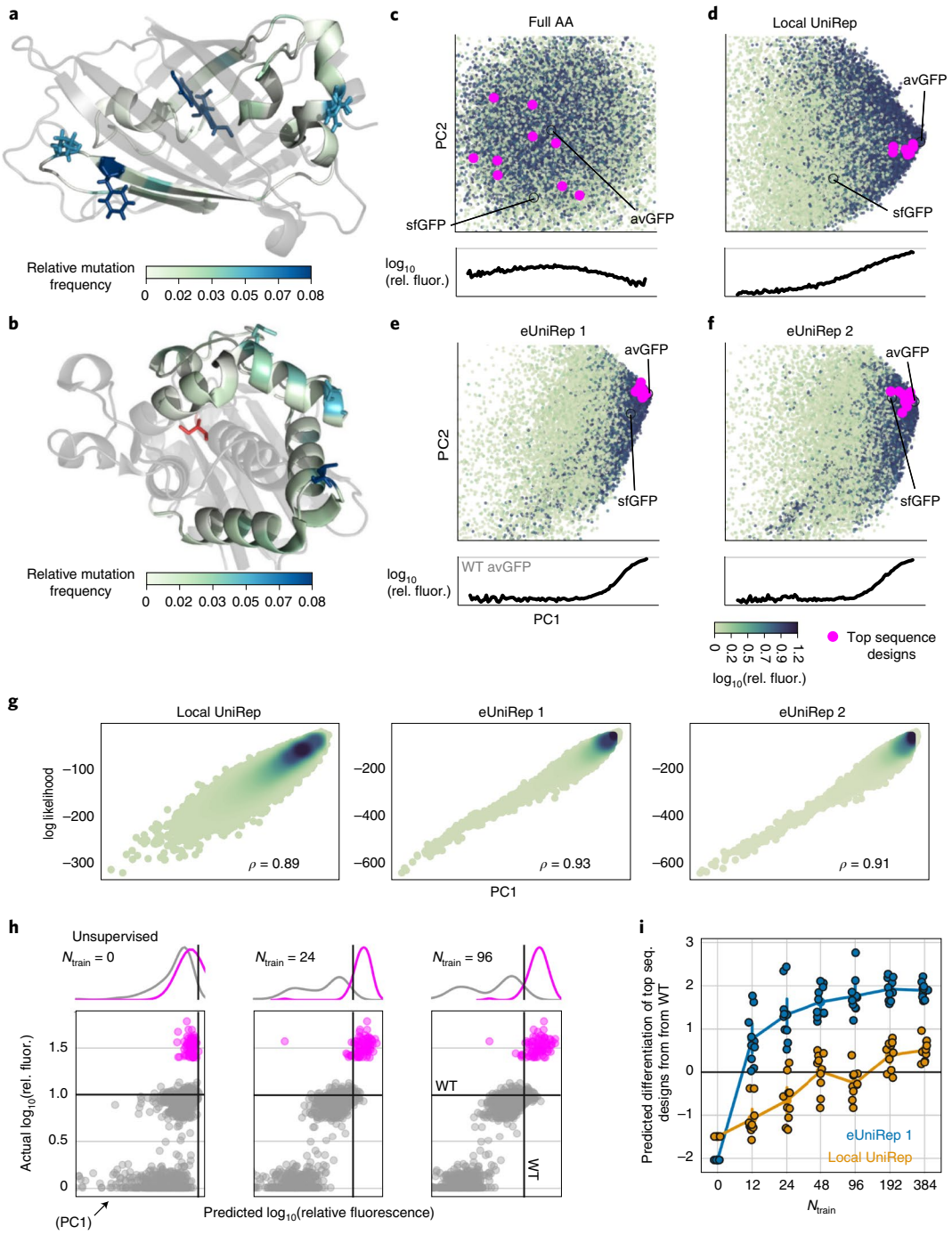

Fig. 4 | eUniRep designs are structurally non-trivial and require both unsupervised training and low-*N* supervised training to discover >WT variants. **a**, Structural visualization of avGFP (Protein Data Bank (PDB), 2WUR). Mutations are colored by relative frequency in >WT designs. The top three residues by mutation count are shown as sticks. The chromophore is colored by count of mutations made to any of the chromophore residues. Fluor., fluorescence. **b**, As in **a** but for the TEM-1 β-lactamase structure (PDB, 1ZG4), in which the catalytic serine (S70) is highlighted in red. Principal component analyses of Full AA (**c**), Local UniRep (**d**), eUniRep 1 (**e**) and eUniRep 2 (**f**) representations of sequences from the local fitness landscape of avGFP, colored by $\log_{10}$(relative fluorescence). Magenta points show the top ten sequence designs produced by each model. Below each plot, $\log_{10}$(relative fluorescence) is shown as a function of PC1; Pearson $r = 0.02$ (Full AA), $r = 0.52$ (Local UniRep), $r = 0.52$ (eUniRep 1), $r = 0.51$ (eUniRep 2). **g**, Sequence log-likelihood versus PC1 for Local UniRep (left), eUniRep 1 (middle) and eUniRep 2 (right) with Spearman correlations noted. **h**, Scatterplots of actual versus predicted $\log_{10}$(relative fluorescence), ordered by varying amounts of supervision. $N_{train} = 0$ corresponds to a purely unsupervised case, and thus the *x* axis corresponds to PC1. Gray circles are examples from the training distribution from which low-*N* training mutants are sampled. Magenta points represent the top 82 designed GFP sequences. Kernel density estimates of each population are shown above each scatterplot. **i**, Jitter plot depicting the degree to which top sequence designs can be differentiated from WT on the basis of predicted activity as a function of the number of low-*N* training mutants used (Methods). At a given N_{train} value, each data point represents a prediction replicate, which involves an independently sampled low-*N* training sequence (seq.) set.

(Fig. 3e). For cluster 1 designs, which were >WT under all antibiotic conditions, we calculated predicted fitness, assuming each mutation contributed additively, and compared this to the experimentally observed fitness of the fully mutated design. Surprisingly, most of these designs were substantially >WT despite being predicted as loss of function under additivity (Fig. 3e). Additionally, their in silico evolutionary trajectories were consistent with the navigation of a rugged, epistatic fitness landscape[51] (Supplementary Fig. 13). These results suggest that, via transfer of epistatic information from unsupervised learning, eUniRep can exploit epistasis even when no higher-order mutation combinations have been observed in the training data.

As with GFP, we found evotuning to be robust to the size of the evotuning sequence set as well as to the number of model updates performed during training (Methods and Supplementary Fig. 9b). Our models were equally performant even with 10% of the full sequence data used for evotuning and 15% of the model updates.

Unsupervised training serves to guide search away from loss-of-function sequences, while low-*N* supervision enables the discovery of >WT sequences. We next attempted to explain eUniRep's unique ability to enable low-*N* engineering (Fig. 4 and Supplementary Figs. 14–16). While mutations in eUniRep proposals and >WT designs were biased toward solvent-exposed residues, a substantial fraction (40% for GFP and 28% for β-lactamase) were targeted to buried positions including the avGFP chromophore (Fig. 4a and Supplementary Fig. 14). This suggested that eUniRep could make non-trivial, beneficial rearrangements to the hydrophobic core, which previous work suggested is difficult[29]. Additionally, we observed that the most functional β-lactamase designs were not preferentially mutated near the catalytic serine (S70), which ran counter to the typical engineering heuristic of targeting mutations around the enzyme's active site[19]. This result also suggested that eUniRep can exploit non-local epistatic interactions (Fig. 4b and Supplementary Fig. 14). Unsurprisingly, eUniRep's mutational preference could not be explained by first-order position-wise mutational tolerance, suggesting that eUniRep enabled more than consensus sequence design despite both methods drawing on local sequence information (Fig. 2c and Supplementary Fig. 15).

Not finding a clear explanation for eUniRep's performance among these structural and evolutionary analyses, we examined the eUniRep sequence representation. Strikingly, we found a strong correlation between its primary axis of variation (principal component (PC)1) and protein function (Fig. 4c–f, avGFP, Pearson $r=0.51$, 0.52; Supplementary Fig. 17a, β-lactamase, Pearson $r=0.44$), which was not observed for PC1 of the Full AA representation (avGFP, Pearson $r=0.02$; β-lactamase, Pearson $r=0.05$). However, while PC1 could differentiate nonfunctional sequences from functional ones, it could not differentiate functional sequences with WT or greater levels of activity (Fig. 4c–f and Supplementary Fig. 17a). For example, our most active GFP designs and sfGFP had PC1 scores that were similar to those of WT avGFP (Fig. 4e,f). Further examination revealed that PC1 was highly correlated with sequence likelihood under each UniRep model, with the highest such correlations observed for eUniRep (Spearman $\rho=0.93$ and 0.91 for eUniRep 1 and 2, respectively; Fig. 4g and Supplementary Fig. 17b). Given that global unsupervised pre-training and evotuning of these models are performed on natural sequences, this suggests that the primary utility of unsupervised learning as performed here is to guide search away from unpromising sequences in the fitness landscape based on a (semantically meaningful) sense of their unnaturalness.

However, this also suggests that unsupervised training alone does not enable the discovery of better-than-natural variants. Indeed, we observed that only with low-*N* supervised learning could >WT designs be differentiated from those with WT or lower levels of activity (Fig. 4h,i and Supplementary Fig. 17c,d). Thus, we propose a two-part model to explain eUniRep's ability to enable low-*N* protein engineering: First, unsupervised learning greatly simplifies search by eliminating the vast majority of the nonfunctional fitness landscape on the basis of unnaturalness. 'On top' of this information, supervised learning with a small number of low-*N* mutants then distills the critical information needed to discover better-than-natural variants.

Discussion

This work demonstrates a generalizable and scalable paradigm for low-*N* protein engineering. By distilling information from both the global and local sequence landscape, we reproducibly leveraged $N=24$ random training mutants and one round of in silico screening into over 1,000 new >WT designs. This is the strongest case of generalization and data efficiency in machine learning-guided protein function optimization to date (Supplementary Fig. 18). Additionally, our two-part mechanism to explain this performance provides context for and extends previous unsupervised protein function modeling and design work. While unsupervised methods trained on natural sequence data perform well at predicting or avoiding loss-of-function variants during modeling and design, they have also been unable to reliably model or design better-than-natural variants[38,39,41,47,52]. Our findings suggest that a small amount of labeled data and additional supervised learning in addition to unsupervised pre-training may be necessary to find enhanced variants.

We took advantage of robust, high-fidelity multiplexed assays to extensively characterize our approach on avGFP and TEM-1 β-lactamase. While low-*N* design is intended for proteins for which such assays are not available, both proteins have a rich history of being studied or engineered with them. As such, we consider existing >WT variants to be a high bar. Here, with only 24 random mutants of avGFP as training data, we designed new fluorescent proteins (FPs) that rivaled sfGFP, the product of many years of high-throughput, high-fidelity protein engineering.

Nevertheless, unlike GFP and TEM-1 β-lactamase, most proteins do not have assays that are both high throughput and high fidelity. In many therapeutic and industrial projects, high-fidelity experimental measurements of endpoint functions, such as crop yield or biologic efficacy, are scarce and come at the end of long test cycles. In theory, generating high-throughput proxy assays of these endpoints should improve engineering success rates. However, empirically this is often not the case as evidenced, for example, by Eroom's law in drug development[13,15]. Here efforts to use high-throughput proxy assays for the endpoint in question may in fact generate worse candidates for later-stage development[13,15] by overoptimizing a biased metric[53]. In sum, this suggests that generalizing from low-*N* high-fidelity measurements may be more important than learning from high-*N* low-fidelity measurements.

Indeed, several previous efforts successfully engineered valuable proteins using high-fidelity assays and low-*N* design[19,23,24,54–58]. However, these (semi-)rational protein engineering approaches intensively rely on hand-crafted structural or (co)evolutionary priors to narrow the search space of potential mutations[8,19,59,60]. Additionally, they often require expert judgment to learn from data, which may include modifying energy functions for biophysical design[61] and iteratively designing and testing structure-guided mutation combinations[19,62–65]. Together, these modeling and design choices introduce biases that could manifest as a mismatch between optimization metric and endpoint. By contrast, UniRep and our low-*N* approach are paradigmatically empirical and sequence based, improving with the exponential growth of sequence databases to minimize bias[25] and leaving open the possibility of discovering new principles of protein folding and activity that extend beyond our current mental models. Indeed, when combining data-driven digital fitness landscapes with in silico evolution to both measure well

and search far, we find that there may be surprising diversity and function in the vastness of sequence space.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-021-01100-y.

Received: 21 August 2020; Accepted: 22 February 2021;
Published online: 7 April 2021

References

1. Romero, P. A. & Arnold, F. H. Exploring protein fitness landscapes by directed evolution. *Nat. Rev. Mol. Cell Biol.* **10**, 866–876 (2009).
2. Packer, M. S. & Liu, D. R. Methods for the directed evolution of proteins. *Nat. Rev. Genet.* **16**, 379–394 (2015).
3. Lutz, S. & Patrick, W. M. Novel methods for directed evolution of enzymes: quality, not quantity. *Curr. Opin. Biotechnol.* **15**, 291–297 (2004).
4. Goldsmith, M. & Tawfik, D. S. Directed enzyme evolution: beyond the low-hanging fruit. *Curr. Opin. Struct. Biol.* **22**, 406–412 (2012).
5. Zhao, H. & Arnold, F. H. Combinatorial protein design: strategies for screening protein libraries. *Curr. Opin. Struct. Biol.* **7**, 480–485 (1997).
6. You, L. & Arnold, F. H. Directed evolution of subtilisin E in *Bacillus subtilis* to enhance total activity in aqueous dimethylformamide. *Protein Eng.* **9**, 77–83 (1996).
7. Lagassé, H. A. D. et al. Recent advances in (therapeutic protein) drug development. *F1000Res.* **6**, 113 (2017).
8. Marshall, S. A., Lazar, G. A., Chirino, A. J. & Desjarlais, J. R. Rational design and engineering of therapeutic proteins. *Drug Discov. Today* **8**, 212–221 (2003).
9. Rao, A. G. The outlook for protein engineering in crop improvement. *Plant Physiol.* **147**, 6–12 (2008).
10. Schmid, A. et al. Industrial biocatalysis today and tomorrow. *Nature* **409**, 258–268 (2001).
11. Sheldon, R. A. & Pereira, P. C. Biocatalysis engineering: the big picture. *Chem. Soc. Rev.* **46**, 2678–2691 (2017).
12. Mullard, A. Better screening and disease models needed. *Nat. Rev. Drug Discov.* **15**, 751–769 (2016).
13. Scannell, J. W. & Bosley, J. When quality beats quantity: decision theory, drug discovery, and the reproducibility crisis. *PLoS ONE* **11**, e0147215 (2016).
14. Hughes, J. P., Rees, S., Kalindjian, S. B. & Philpott, K. L. Principles of early drug discovery. *Br. J. Pharmacol.* **162**, 1239–1249 (2011).
15. Scannell, J. W., Blanckley, A., Boldon, H. & Warrington, B. Diagnosing the decline in pharmaceutical R&D efficiency. *Nat. Rev. Drug Discov.* **11**, 191–200 (2012).
16. Laverty, H. et al. How can we improve our understanding of cardiovascular safety liabilities to develop safer medicines? *Br. J. Pharmacol.* **163**, 675–693 (2011).
17. Silver, L. L. Challenges of antibacterial discovery. *Clin. Microbiol. Rev.* **24**, 71–109 (2011).
18. Wu, Z., Jennifer Kan, S. B., Lewis, R. D., Wittmann, B. J. & Arnold, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proc. Natl Acad. Sci. USA* **116**, 8852–8858 (2019).
19. Lutz, S. Beyond directed evolution—semi-rational protein engineering and design. *Curr. Opin. Biotechnol.* **21**, 734–743 (2010).
20. Bedbrook, C. N., Yang, K. K., Rice, A. J., Gradinaru, V. & Arnold, F. H. Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization. *PLoS Comput. Biol.* **13**, e1005786 (2017).
21. Bedbrook, C. N. et al. Machine learning-guided channelrhodopsin engineering enables minimally invasive optogenetics. *Nat. Methods* **16**, 1176–1184 (2019).
22. Romney, D. K., Murciano-Calles, J., Wehrmüller, J. E. & Arnold, F. H. Unlocking reactivity of TrpB: a general biocatalytic platform for synthesis of tryptophan analogues. *J. Am. Chem. Soc.* **139**, 10769–10776 (2017).
23. Silva, D. A., Yu, S., Ulge, U. Y., Spangler, J. B. & Jude, K. M. De novo design of potent and selective mimics of IL-2 and IL-15. *Nature* **565**, 186–191 (2019).
24. Marcandalli, J., Fiala, B., Ols, S. & Perotti, M. Induction of potent neutralizing antibody responses by a designed protein nanoparticle vaccine for respiratory syncytial virus. *Cell* **176**, 1420–1431 (2019).
25. Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods* **16**, 1315–1322 (2019).
26. Halevy, A., Norvig, P. & Pereira, F. The unreasonable effectiveness of data. In *IEEE Intelligent Systems* (IEEE, 2009).
27. Hénaff, O. J. et al. Data-efficient image recognition with contrastive predictive coding. In *Proc. 37th Int. Conf. Machine Learning* **119**, 4182–4192 (2020).
28. Ogden, P. J., Kelsic, E. D., Sinai, S. & Church, G. M. Comprehensive AAV capsid fitness landscape reveals a viral gene and enables machine-guided design. *Science* **336**, 1139–1143 (2019).
29. Biswas, S. et al. Toward machine-guided design of proteins. Preprint at *bioRxiv* https://doi.org/10.1101/337154 (2018).
30. Brookes, D. H., Park, H. & Listgarten, J. Conditioning by adaptive sampling for robust design. Preprint at https://arxiv.org/abs/1901.10060 (2019).
31. Gupta, A. & Zou, J. Feedback GAN for DNA optimizes protein functions. *Nat. Mach. Intell.* **1**, 105–111 (2019).
32. Cadet, F., Fontaine, N., Li, G., Sanchis, J. & Chong, M. N. F. A machine learning approach for reliable prediction of amino acid interactions and its application in the directed evolution of enantioselective enzymes. *Sci. Rep.* **8**, 16757 (2018).
33. Saito, Y., Oikawa, M., Nakazawa, H. & Niide, T. Machine-learning-guided mutagenesis for directed evolution of fluorescent proteins. *ACS Synth. Biol.* **7**, 2014–2022 (2018).
34. Musdal, Y., Govindarajan, S. & Mannervik, B. Exploring sequence–function space of a poplar glutathione transferase using designed information-rich gene variants. *Protein Eng. Des. Sel.* **30**, 543–549 (2017).
35. Romero, P. A., Krause, A. & Arnold, F. H. Navigating the protein fitness landscape with Gaussian processes. *Proc. Natl Acad. Sci. USA* **110**, E193–E201 (2013).
36. Liao, J. et al. Engineering proteinase K using machine learning and synthetic genes. *BMC Biotechnol.* **7**, 16 (2007).
37. Fox, R. J. et al. Improving catalytic function by ProSAR-driven enzyme evolution. *Nat. Biotechnol.* **25**, 338–344 (2007).
38. Riesselman, A. J., Ingraham, J. B. & Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nat. Methods* **15**, 816–822 (2018).
39. Hopf, T. A., Ingraham, J. B., Poelwijk, F. J. & Schärfe, C. P. I. Mutation effects predicted from sequence co-variation. *Nature* **35**, 128–135 (2017).
40. Sinai, S., Kelsic, E., Church, G. M. & Nowak, M. A. Variational auto-encoding of protein sequences. Preprint at https://arxiv.org/abs/1712.03346 (2017).
41. Shin, J.-E. et al. Protein design and variant prediction using autoregressive generative models. Preprint at *bioRxiv* https://doi.org/10.1101/757252 (2019).
42. Sarkisyan, K. S. et al. Local fitness landscape of the green fluorescent protein. *Nature* **533**, 397–401 (2016).
43. Ashkenazy, H. & Penn, O. FastML: a web server for probabilistic reconstruction of ancestral sequences. *Nucleic Acids Res.* **40**, W580–W584 (2012).
44. Gumulya, Y. & Gillam, E. M. J. Exploring the past and the future of protein evolution with ancestral sequence reconstruction: the 'retro' approach to protein engineering. *Biochem. J.* **474**, 1–19 (2017).
45. Sternke, M., Tripp, K. W. & Barrick, D. Consensus sequence design as a general strategy to create hyperstable, biologically active proteins. *Proc. Natl Acad. Sci. USA* **116**, 11275–11284 (2019).
46. Porebski, B. T. & Buckle, A. M. Consensus protein design. *Protein Eng. Des. Sel.* **29**, 245–251 (2016).
47. Russ, W. P. et al. An evolution-based model for designing chorismate mutase enzymes. *Science* **369**, 440–445 (2020).
48. Firnberg, E., Labonte, J. W. & Gray, J. J. A comprehensive, high-resolution map of a gene's fitness landscape. *Mol. Biol. Evol.* **31**, 1581–1592 (2014).
49. Breen, M. S., Kemena, C., Vlasov, P. K., Notredame, C. & Kondrashov, F. A. Epistasis as the primary factor in molecular evolution. *Nature* **490**, 535–538 (2012).
50. Povolotskaya, I. S. & Kondrashov, F. A. Sequence space and the ongoing expansion of the protein universe. *Nature* **465**, 922–926 (2010).
51. Schenk, M. F., Szendro, I. G., Salverda, M. L. M., Krug, J. & de Visser, J. A. G. M. Patterns of epistasis between beneficial mutations in an antibiotic resistance gene. *Mol. Biol. Evol.* **30**, 1779–1787 (2013).
52. Repecka, D. et al. Expanding functional protein sequence spaces using generative adversarial networks. *Nat. Mach. Intell.* https://doi.org/10.1038/s42256-021-00310-5 (2021).
53. Manheim, D. & Garrabrant, S. Categorizing variants of Goodhart's Law. Preprint at https://arxiv.org/abs/1803.04585 (2018).
54. Dou, J. et al. De novo design of a fluorescence-activating β barrel. *Nature* **561**, 485–491 (2018).
55. Lu, P., Min, D., DiMaio, F., Wei, K. Y. & Vahey, M. D. Accurate computational design of multipass transmembrane proteins. *Science* **359**, 1042–1046 (2018).
56. Bick, M. J. et al. Computational design of environmental sensors for the potent opioid fentanyl. *eLife* **6**, e28909 (2017).
57. Zhang, R. K., Chen, K., Huang, X. & Wohlschlager, L. Enzymatic assembly of carbon–carbon bonds via iron-catalysed sp^3 C–H functionalization. *Nature* **565**, 67–72 (2019).

58. Bornscheuer, U. T. & Pohl, M. Improved biocatalysts by directed evolution and rational protein design. *Curr. Opin. Chem. Biol.* **5**, 137–134 (2001).
59. Huang, P.-S., Boyken, S. E. & Baker, D. The coming of age of de novo protein design. *Nature* **537**, 320–327 (2016).
60. Chen, R. Enzyme engineering: rational redesign versus directed evolution. *Trends Biotechnol.* **19**, 13–14 (2001).
61. Alford, R. F. et al. The Rosetta all-atom energy function for macromolecular modeling and design. *J. Chem. Theory Comput.* **13**, 3031–3048 (2017).
62. Pédelacq, J.-D., Cabantous, S., Tran, T., Terwilliger, T. C. & Waldo, G. S. Engineering and characterization of a superfolder green fluorescent protein. *Nat. Biotechnol.* **24**, 79–88 (2006).
63. Dror, A., Shemesh, E. & Dayan, N. Protein engineering by random mutagenesis and structure-guided consensus of *Geobacillus stearothermophilus* lipase T6 for enhanced stability in methanol. *Appl. Environ. Microbiol.* **80**, 1515–1527 (2014).
64. Rocklin, G. J., Chidyausiku, T. M., Goreshnik, I. & Ford, A. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science* **357**, 168–175 (2017).
65. Wannier, T. M. et al. Monomerization of far-red fluorescent proteins. *Proc. Natl Acad. Sci. USA* **115**, E11294–E11301 (2018).

Methods

Evolutionary fine tuning (evotuning). We reasoned that, by fine tuning UniRep's existing knowledge of all protein sequences to the local sequence neighborhood of the target sequence (evotuning), we may be able to reduce the prohibitive data requirements of supervised deep learning and thereby enable low-*N* design. Indeed, impressive gains in data efficiency were obtained through similar means in other machine learning domains including vision[27,66,67] and language[66,68,69]. We began with model weights that had been globally pre-trained on UniRef50 as described previously[25]. To evotune, we select a subset of public sequences that are closer to the target protein and then fine tune the globally pre-trained weights on the UniRep multiplicative long short-term memory (mLSTM) model on this local sequence neighborhood. We note that the evotuning procedure does not strictly require sequences that are evolutionarily related. Indeed, sequence search tools such as JackHMMER, which we use to source related sequences to the protein of interest, return sequences that are only statistically related at a sequence level (as measured by the *E* value). Although they are often evolutionarily related homologs of the protein of interest, they need not necessarily be.

For avGFP, we used the same evotuned weights as previously described, called eUniRep 1 (ref. [25]) above, and additionally repeated the evotuning process to ensure its robustness. As with eUniRep 1 (ref. [25]), the avGFP target sequence together with a selection of related FPs was searched with JackHMMER[70] until convergence. Edit distance was computed between the search result sequences and the avGFP target sequence. The sequence set was filtered for length (keeping all with <500 amino acids) and Levenshtein distance from avGFP (keeping all <400), and sequences with non-standard amino acids were removed, yielding 79,482 sequences. We note that this number is larger than the 32,225 sequences used to train eUniRep 1 obtained in ref. [25]. The difference is due to the stochasticity of JackHMMER and updates to the JackHMMER web server between runs for eUniRep 1 and eUniRep 2, as well as running JackHMMER to convergence for eUniRep 2. We note that the downstream design performance enabled by these two evotuning models was similar, despite this 2× difference in the number of sequences in the dataset.

To determine when to stop training, we selected a 10% 'out-of-distribution set' by sampling each sequence with a probability proportional to the fourth power of the edit distance. A 10% in-distribution set was selected uniformly randomly. We initialized the weights of the 1,900 dimensional UniRep mLSTM values with the globally pre-trained weights and trained for 13,500 iterations with early stopping[71,72], until the outer validation set loss began to increase. This model was used to produce the representations for eUniRep 2 as named above.

The evotuning for TEM-1 β-lactamase proceeded similarly, seeding the JackHMMER search with the WT TEM-1 β-lactamase sequence together with related β-lactamase sequences. The results were filtered for length (<600 amino acids) and Levenshtein distance from TEM-1 β-lactamase (<286), and sequences with non-standard amino acids were removed, yielding 76,735 results. Training, initialized with the global weights as above, proceeded for 13,500 iterations.

For Local UniRep, we used the same dataset and training procedure as described above, but instead of using the globally pre-trained UniRep weights as initialization, we generated a random weight initialization from the same distribution that was used to initialize the original UniRep model. This is analogous to retraining the original UniRep model but only on the local sequence landscape, leading to the name Local UniRep.

We expect that using JackHMMER for sourcing evotuning sequences will be a reliable approach for most proteins. Indeed, previous work that trained unsupervised models on local sequence landscapes to successfully predict protein function for 28 different proteins employed JackHMMER[38,39].

Retrospective experiments for low-*N* engineering. The purpose of our retrospective experiments was to evaluate the possibility of low-*N* engineering. Toward this end, we tested the abilities of different sequence-to-function models to meaningfully generalize in terms of predictive performance from a 'local' region of the fitness landscape to more 'distant' regions using only a small number, *N*, of (sequence, function) pairs from the local fitness landscape.

Our retrospective experiments took the following steps.

1. Dataset creation and processing. Here we established three datasets, the generation and/or processing of which is described in detail below and the properties of which are summarized in Supplementary Fig. 1.
   a. 'Sarkisyan', which is comprised of functionally characterized sequences from the local fitness landscape of avGFP. This dataset was publicly available and was processed from Sarkisyan et al.[42]. In our experiments, this dataset was used for sampling training sequences.
   b. 'SynNeigh', which is comprised of functionally characterized sequences from the local fitness landscape of sfGFP and the local fitness landscapes of related variants of sfGFP that were obtained through simple machine learning-guided exploration strategies. Thus, this dataset represents a collection of many local fitness landscapes for different avGFPs. These data were generated from variants obtained from Biswas et al.[29] and will be made publicly available upon peer-reviewed publication. In our experiments, this dataset was used for evaluating generalization.
   c. 'FP Homologs', which is comprised of functionally characterized sequences from the global fitness landscape of known Aequorean FPs. This dataset was generated by molecularly shuffling the DNA of 65 extant Aequorean FPs and thus represents a global, albeit sparse, sampling of the global fitness landscape substantially beyond that explored in the local fitness landscape of avGFP (Sarkisyan). This dataset was generated and processed for this work and has been made publicly available. In our experiments, this dataset was used for evaluating generalization.
2. Each dataset was then randomly split three ways to produce splits 0, 1 and 2. Model prototyping and evaluation as described in subsequent steps below were entirely performed on split 0. After prototyping, a final list of models, hyperparameters and procedural parameters was fixed, and the performance of each approach was evaluated on split 1, the results of which are reported in Supplementary Fig. 2. Split 2 was used for all prospective experiments as reported in the main text.
3. We systematically evaluated the impact of several factors on generalization on split 0. We defined good generalization to be accurate rank ordering of sequences in a generalization set, such that, if we were to select the top ranked sequences for experimental characterization, they would be highly functional. The factors examined are as follows:
   a. Number of training sequences (*N*).
   b. Acquisition policy. This defines how the *N* training sequences are selected. A complete list of policies and their descriptions is below.
   c. Sequence representation. This defines how the amino acid sequence is numerically encoded to the top model. Full AA or eUniRep are examples of encodings. A complete list of representations and their descriptions is below.
   d. Top model. This is a simple, low-parameter supervised model that is trained on training sequence representations to predict quantitative function. Ridge regression is an example top model. A complete list of top models examined and their descriptions is below.
4. Once we were able to determine how these variables affected retrospective generalization, especially in low-*N* settings, we fixed a final list of *N* training sequences, sequence representations, top models and reporting criteria and reproduced the retrospective experiments again on split 1. This was to ensure that we did not overfit to split 0. A summary of these results is reported in Supplementary Fig. 2.

Retrospective experimental result summary. Supplementary Fig. 2 summarizes the results of our retrospective generalization experiments, in which the task was to rank order members of the generalization set such that, if we were to select the top 96 for characterization, as many as possible should be >WT 'hits'. To contextualize performance, this metric can be normalized as a ratio to the performance obtained by a random ordering of generalization set members.

Sequence representation was the most influential variable that affected performance. One-hot Full AA, Doc2Vec[73], UniRep (globally trained but not evotuned) generally did not show improvements over random ordering for training sets of any size from the local fitness landscape of avGFP (Sarkisyan). By contrast, evotuned models showed a performance gain greater than 20× over random models when generalizing to members of SynNeigh and a performance gain 2–5× over random models when generalizing to members of FP Homologs. In particular, eUniRep 1 and eUniRep 2 were superior to Local UniRep, which lacks knowledge of global sequence space, showing highly data-efficient performance with as few as $N=8$ training sequences (Supplementary Fig. 2).

Choice of top model played a less substantial but nonetheless important role. In particular, we noticed a marked performance difference between L1- (Lasso–least-angle regression (LARS)) and L2-penalized (Ridge) top models, with L2 variants performing substantially better. We suspect that this is likely because the meaningful information contained in the mLSTM representations is entangled, and hence the representation as a whole is non-sparse. This violates the assumptions of L1-penalized regression. Among L2 models, we noticed that choosing a more stringent regularization with the same (statistically) inner cross-validation performance yielded a slight gain in performance (Ridge 'sparse refit' (SR)). Finally, ensembling this approach (Ens Ridge SR) neither hurt nor improved performance but yielded an empirical uncertainty estimate.

Interestingly, how training sequences were acquired did not matter greatly (data not shown). For real-world technical simplicity, we therefore chose to acquire training points randomly from the output of error-prone PCR or single-mutation deep mutational scans. Additionally, the models that worked the best (eUniRep powered models) were surprisingly robust to the number of training points sampled ($N=8$, 24 or 96), which are all small enough that they can be feasibly collected for a variety of proteins and applications.

Dataset creation. Three datasets were used for our retrospective low-*N* engineering experiments. The Sarkisyan dataset was also used for the prospective design experiment illustrated in Fig. 2 in the main text. A detailed description of their generation and/or processing follows.

Sarkisyan. This dataset was obtained from Sarkisyan et al.[42]; it is publicly available. Briefly, the authors used error-prone PCR to mutate the sequence encoding WT avGFP and then measured the fluorescence of approximately 50,000 variants using

FlowSeq in a manner similar to how it was performed in this work (FlowSeq). We further processed their dataset by

1. Minimum–maximum scaling $\log_{10}$(relative fluorescence) values according to the formula $(x - \text{min\_val})(\text{wt\_val} - \text{min\_val})^{-1}$, where min_val is the fluorescence of the least fluorescent sequence, and wt_val is the fluorescence of the WT sequence. Thus, after transformation, WT fluorescence corresponds to a value of 1, whereas an entirely nonfunctional sequence has a fluorescence value of 0. This minimum–maximum scaling was performed to ensure consistency with the other datasets.
2. Random splitting of the dataset into three splits as described above.

The distribution of transformed fluorescence values, edit distances (number of mutations) to avGFP and edit distances between members of this dataset are shown in Supplementary Fig. 1a.

SynNeigh. The purpose of this dataset was to serve as a generalization set to evaluate model generalizability. This dataset was generated from variants discovered in Biswas et al.[29]. Here the authors used a variety of simple machine learning-guided approaches to propose diverse but functional sequence variants of sfGFP. This included model-guided exploration under a three-layer fully connected feed-forward neural network and under a composite-residues neural network. The goals of these explorations were varied and included attempts to improve fluorescence, to diversify the sequence while maintaining function and to diversify the sequence while maintaining function and only mutating combinations of residues otherwise difficult to singly mutate. In total, 286 'parent' variants were proposed in this manner.

In this work, after pooling plasmid DNA for all 286 parent variants, we performed error-prone PCR (GeneMorph II Random Mutagenesis kit, Agilent Technologies) over the full length of the gene encoding GFP, aiming for an average of two mutations per template. This library was cloned and transformed into DH5α *E. coli* (see "Library cloning and transformation"), with an estimated library size of 150,000 members. The relative fluorescence of each variant in the library was then measured with FlowSeq (FlowSeq). In total, we obtained high-quality fluorescence measurements for 104,285 variants.

Because many of the 286 parent variants were highly functional and we were mostly measuring minorly mutated variants thereof, much of the dataset comprised functional variants. In practice, in low-*N* engineering, our task is to find rare high-functioning sequences among a sea of nonfunctional sequences in the distant or non-local fitness landscape. To better incorporate this intuition into our retrospective experiments, we therefore filtered out variants with intermediate fluorescence ($\geq$0.7 and $\leq$1.5), leaving only nonfunctional and highly functional variants. After filtering, we retained 52,512 variants, 52,416 of which were nonfunctional and 96 of which were highly functional.

This final dataset, which we refer to as 'SynNeigh', was minimum–maximum scaled (using avGFP fluorescence as wt_val) as described above and then split into three parts. Because all measured variants were derivatives from one of the 286 parents, the three-way split was created by first randomly splitting the 286 parent variants three ways and then assigning derivative variants to one of the three splits according to the parent variant from which they had the fewest mutations.

FP Homologs. The purpose of this dataset was to serve as an additional generalization set to evaluate model generalizability and was generated for this work. While SynNeigh is inherently 'centered' around sfGFP and samples several local fitness landscapes densely, FP Homologs sparsely samples the global fitness landscape of known Aequorean FPs.

To accomplish this, we first mined an October 2018 download of the FPbase database[74] for FP sequences of Aequorean origin. Of these 132 sequences, 70 were mutually different with at least three mutations. After manual curation of the 70 sequences, which involved stripping away His tags and manually adjusting the N and C termini of the sequences, which were sometimes modified for crystallization purposes, 65 sequences remained that were mutually different by at least one mutation. The median and maximum numbers of amino acid mutations between these 65 'parent' sequences were 15 and 63, respectively. Note that the full length of each sequence was 238 amino acids. These parents also encompassed a variety of spectral properties, with some of them fluorescing blue or yellow in addition to green. Nucleotide sequences of these 65 parents were obtained by choosing an *E. coli* codon optimization and were subsequently ordered as separate Gene Fragments from Twist Biosciences.

Each Gene Fragment was individually cloned and transformed into DH5α *E. coli* using Golden Gate assembly, and the coding sequence and spectral phenotypes were individually confirmed. Plasmid DNA for each parent was mini-prepped (Qiagen), and all parent plasmid DNA was subsequently pooled. To generate a sparse but broad sampling of sequences in the global fitness landscape spanned by these parents, we performed DNA shuffling[75] followed by error-prone PCR (GeneMorph II Random Mutagenesis kit, Agilent Technologies). The library prepared by DNA shuffling and error-prone PCR is hereafter referred to as the 'shuffled library.'

Because the shuffled library contained mutations throughout the full length of the FP gene, it was not immediately compatible with our FlowSeq protocol, which cannot sequence amplicons longer than 600 bp. We next therefore performed a 'stitching PCR', in which we added a random 20-bp DNA barcode (BHVDBHVDBHVDBHVDBHVD) to the 3′ end of the DNA in both the parent pool and the shuffled library after the stop codon of the gene. The then barcoded parent pool and shuffled library were separately cloned and transformed into DH5α *E. coli* (see "Library cloning and transformation"), with an estimated library size of approximately 100,000 members. Through simulation, we confirmed that it would be overwhelmingly statistically likely that one barcode would 'point to' only one template and not more than one template. Barcodes did not affect translation but were likely transcribed. We nonetheless assumed that this would have a negligible impact on the expression level of the resulting protein.

We next spiked in the transformed parent pool at 0.5% into the transformed shuffled library and performed FlowSeq (see "FlowSeq"). Because this pool contained a collection of spectrally diverse variants, we excited with two different laser combinations (488 nm only, 405 nm and 488 nm) and sorted in four different emission channels (FL1 = 450/50 nm, three bins; FL2 = 525/50 nm, eight bins; FL3 = 600/60 nm, six bins; and FL4 = 665/30 nm, two bins). Instead of sequencing the coding region, we sequenced the 20-bp barcode. Barcode sequencing was performed using a 2 × 75-bp NextSeq mid-output sequencing run.

Examining a heatmap of variant log abundances across all samples, we observed clear structure, indicating groups of variants that were clearly enriched or depleted from sort bins representing different fluorescence intensities under different excitation (lasers) and emission (filters) conditions. However, we also observed what we suspected to be higher frequency noise, in which certain variants would be abundant in one condition but would have zero counts in a highly related condition. We suspected that this was an artifact of under-sorting and possibly under-sequencing our library. To remedy this, we performed imputation of these missing measurements with MAGIC[76], which was originally developed to perform the same kind of imputation for drop-out measurements in single-cell RNA-seq data. We confirmed that imputations were likely high fidelity by artificially dropping out measurements of high-confidence variants (the highly abundant parent sequences) and examining the accuracy of their imputed values (Pearson $r = 0.89$). Considering these imputed counts as 'final', we proceeded with fluorescence inference as we would for a normal FlowSeq experiment. At this point, we obtained $\log_{10}$(relative fluorescence) values associated with each barcode and, for consistency, specifically used those associated with 405-nm and 488-nm excitation and emission in FL2 (525/50 nm).

To determine the identity of the variant that each barcode represented, we performed long-read amplicon sequencing. The sequenced amplicon included both the coding sequence of the FP as well as the 3′ barcode. Two independent PacBio Sequel II runs were performed. The first was for the parent pool and the shuffled library (input into FlowSeq). The second was for all functional members of the parent pool and the shuffled library, which was deemed to be all variants that did not sort into the nonfunctional bin during the flow cytometry step of FlowSeq. The second run was performed to increase the chance that we could successfully decode barcodes for functional library members.

After performing a number of sanity checks, we could reliably associate barcodes with their respective FP variants. The number of instances when a given barcode pointed to multiple variants that were not explainable by sequencing noise was extremely low ($<1 \times 10^{-25}$). In total, we could make 40,581 high-confidence barcode associations, representing 37,582 unique variant sequences. In total, these 37,582 variants (and their 40,581 associated barcodes) accounted for 58% of the NextSeq barcode sequencing data after basic processing (read pair merging, amplicon extraction and basic length filtering on the barcodes). This suggested that, while it is likely that a small-to-moderate portion of the transformed library might have been missed by using this barcode association procedure, we could still capture a large fraction of it.

To make the generalization task more challenging, we further filtered this data to include only parents that were highly functional (10× brighter than avGFP) and variants that bore any of their sequences. To do this, we first identified a set of 16 parent sequences that were highly functional (>10× brighter than avGFP) and confirmed their qualitative improvement over avGFP from the literature. We then analyzed the protein sequence of every variant and assigned any variant with any subsequence that could be unambiguously attributed to one of these 16 parents to be in the filtered list of variants. In total, 27,050 variants met these criteria.

Finally, as performed for SynNeigh, we removed variants with intermediate fluorescence, minimum–maximum scaled the fluorescence values as described above and split the data randomly into three splits.

Acquisition policies. We considered several acquisition policies for sampling training set (sequence, function) pairs. These could be broadly classified into three categories, sequence only, structural and evolutionary, based on the primary source of information they need. For sequence-only methods, we considered randomly sampling mutants from the output of error-prone PCR and randomly sampling single mutants (for example, as the output of a deep mutational scan). For structural and evolutionary approaches, we considered several policies that would sample mutations based on their structural and evolutionary conservation properties to build epistatically dynamic training sets. We found the sequence-only policies of random sampling from error-prone PCR or from single mutants to be as performant as structural and evolutionary policies.

Sequence representations. We considered several different methods to convert sequences into a numerical representation suitable for use in supervised modeling:

1. Full AA. One-hot encoding of the full amino acid sequence is a simple representation method that exactly represents the information contained in an amino acid sequence (no more, no less). Procedurally, to one-hot encode a sequence of length L, a $20 \times L$ matrix, $\mathbf{O}$, is constructed such that $\mathbf{O}(i,j) = 1$ if amino acid i occurs in position j of the sequence (for some predetermined ordering of the 20 amino acids). The final encoding of the sequence is a 'flattened' or 'unrolled' version of $\mathbf{O}$, that is, a vector of dimension $1 \times (20 \times L)$.
2. Doc2Vec. Here we used a previously state-of-the-art approach for representing protein sequences[73], based on the popular Doc2Vec natural-language processing paradigm for generating vector representations of entire documents[77]. In previous work in which we developed UniRep, we compared extensively to this 'Doc2Vec for proteins' approach[25].
3. UniRep (the sequence representation obtained from the globally trained (on UniRef50) UniRep mLSTM). Specifically, the representation is the average hidden state taken across the length of the sequence as reported by Alley et al.[25]. We also refer to this representation as 'avg_hidden'.
4. Local UniRep. The avg_hidden representation obtained from training a randomly initialized mLSTM, the architecture of which is the same as that of UniRep on the same local sequence dataset used for evotuning.
5. eUniRep. The avg_hidden representation obtained from evotuning the UniRep mLSTM that had already been globally trained on UniRef50. The additional suffixes '1' or '2' refer to replicates of the evotuning process.

Top models. We considered several top models. Although, in principle, any supervised model could be used here, for the purposes of low-N engineering, we reasoned that only simple low-parameter models would be reliably fit and have a lower risk of overfitting. Additionally, if the sequence representation is truly semantically rich, then only a simple top model should be needed to make accurate quantitative predictions about function. We therefore restricted our attention to single-layer models, that is, various forms of linear regression as follows:

1. Lasso–LARS. This is L1-penalized linear regression implemented using the LARS algorithm[78]. We used the Python 'sklearn.linear_model.LassoLarsCV' implementation to perform tenfold cross-validation (on the input training data) to select a level of regularization (the parameter α) that minimizes held-out mean squared error. The schedule of regularization strengths is known upfront by use of the LARS algorithm.
2. Ridge. This is L2-penalized linear regression. We used the Python 'sklearn.linear_model.RidgeCV' implementation to perform tenfold cross-validation (on the input training data) to select a level of regularization (the parameter α) that minimizes held-out mean squared error. The schedule of regularization strengths was set to be logarithmically spaced from 1×10^{-6} to 1×10^{6}. Features were normalized upfront by subtracting the mean and dividing by the L2 norm.
3. Ridge SR. This is the same as the 'Ridge' procedure above, except that we additionally perform a post hoc SR procedure. The 'Ridge' top model above chooses a level of regularization that optimizes for model generalizability if the ultimate test distribution (that is, distant regions of the fitness landscape) resembles the training distribution. However, this is not likely the case. Therefore, we performed a post hoc procedure to choose the strongest regularization such that the cross-validation performance was still statistically equal (by t-test) to the level of regularization we would select through normal cross-validation. This procedure selects a stronger regularization than what would be obtained using the 'Ridge' procedure as defined above.
4. Ensembled Ridge SR. This is the same as the 'Ridge SR' procedure above, except that the final top model is an ensemble of Ridge SR top models. The ensemble is composed of 100 members. Each member (a Ridge SR top model) is fit to a bootstrap of the training data (N training points are resampled N times with replacement) and a random subset of 50% of the features. The final prediction is an average of all members in the ensemble. The rationale for this approach is that it is based on consensus of many different Ridge SR models that have different 'hypotheses' for how sequence might influence function. Differences in these 'hypotheses' are driven by the fact that every bootstrap represents a different plausible instantiation of the training data and that every random subsample of features represents different variables that could influence function.

Training datasets for prospective low-N engineering. For prospective design of GFP, we relied on sampling random subsets of size $N = 24$ or $N = 96$ from the Sarkisyan dataset (see dataset descriptions in "Retrospective experiments for low-N engineering" above). This corresponded to virtually picking random mutants (for example, colonies) from the error-prone PCR-generated library. This would be straightforward to implement experimentally, and, indeed, error-prone PCR is a common starting point for many protein engineering efforts. A shortcoming of error-prone PCR is that, because changes of only a few nucleotide (usually at a rate of 0.1–0.5%) are made per gene, it is difficult to observe amino acid substitutions that require multiple mutations to the same codon. However, it is a simple and tunable way to sample higher-order mutation combinations.

For prospective design of TEM-1 β-lactamase, we relied on sampling random subsets of size $N = 24$ or $N = 96$ from the single-mutation scanning mutagenesis (deep mutational scan) dataset generated by Firnberg et al.[48]. Briefly, Firnberg et al. performed scanning mutagenesis of the *E. coli* TEM-1 β-lactamase protein and profiled the activity of 95.6% (5,212 of 5,453) of single amino acid substitutions. Unlike the output error-prone PCR, scanning mutagenesis as performed here can explore any amino acid substitution. However, higher-order mutation combinations were not explored. The authors used a tunable bandpass genetic selection assay[79] to measure the resistance of a variant to different concentrations of ampicillin, up to 1,024 μg ml^{-1}. The output of their assay was highly correlated with the minimum inhibitory concentration of ampicillin at which a variant could no longer confer resistance. We note that this is a different measure of fitness than that used in this work, which is based on logarithmic fold enrichments. Nevertheless, we would expect a gain- or loss-of-function variant in their system to be a gain- or loss-of-function variant in our system, and thus we felt that it was a suitable pool of training mutants for our prospective design experiments.

Prospective design: sequence proposal via in silico directed evolution. We wished to use an algorithm that would, on average, seek more functional variants but was not deterministically forced to do so. We therefore used a Metropolis–Hastings Markov chain Monte Carlo algorithm to stochastically sample from the non-physical Boltzmann distribution defined by

$$p_i = \frac{1}{Z} \exp\left(-\frac{\hat{y}_i}{kT}\right),$$

where $\hat{y}_i$ is the model-predicted fitness for sequence i, k is a constant that was set to 1, T is the temperature, and Z is an unknown normalization constant.

Our in silico directed evolution algorithm was performed as follows:

1. Input:
   a. An initial sequence.
   b. A sequence-to-function model that predicts the quantitative function or fitness of an amino acid sequence.
   c. Temperature, T.
   d. Trust radius (the number of mutations relative to WT allowed in proposed designs).
2. Initialize (set state sequence s equal to a provided initial sequence).
3. Propose a new sequence, s^*, by randomly adding $m \sim \text{Poisson}(\mu - 1) + 1$ mutations to s, where μ is the sequence proposal mutation rate.
4. Accept the proposal and update the state sequence $s \leftarrow s^*$, with probability equal to

$$\min\left(1, \exp\left(\frac{\hat{y}^* - \hat{y}}{T}\right)\right),$$

   where $\hat{y}^*$ and $\hat{y}$ are the predicted fitness of the proposed sequence and state sequence, respectively. Otherwise, the proposal is rejected (and the state sequence is kept as it is). Note that, if the sequence proposal has more mutations than the input trust radius, its predicted fitness is set, post hoc, to negative infinity, thereby forcing rejection of the proposal.
5. Iterate steps 3 and 4 for a predetermined number of iterations.

For the prospectively designed GFP and TEM-1 β-lactamase libraries, for a given sequence-to-function model (the combination of sequence representation method and a low-N trained top model), 3,500 evolutionary trajectories were run in parallel for 3,000 iterations. The initial sequence for each trajectory was obtained by making Poisson(2) + 1 random mutations to the WT sequence. The sequence proposal mutation rate μ for each trajectory was set to be a random draw from a uniform(1, 2.5) distribution.

We investigated a number of different temperature parameters spanning six orders of magnitude. We found that, for GFP and TEM-1 β-lactamase models, a temperature of 0.01 yielded good trajectory behavior. We qualitatively ascertained this by visualizing how predicted fitness varied across the trajectory. High temperatures, which increase acceptance probabilities, produced overly explorative trajectories that mostly dwelled in low predicted fitness regions. Low temperatures, which decrease acceptance probabilities, produced overly exploitative trajectories that had monotonically increasing fitness traces. A temperature of 0.01 produced trajectories with fitness traces that improved on average but were not monotonic, suggesting a qualitatively good exploration–exploitation balance.

For the prospective GFP designs presented in the main text we used a trust radius of 15 mutations, and for a smaller-scale experiment presented in Supplementary Fig. 4, we used a trust radius of seven mutations. For the prospective TEM-1 β-lactamase designs, we used a trust radius of seven mutations. We reduced the trust radius relative to GFP because only single mutants were used as low-N training data for the TEM-1 β-lactamase experiments.

From here, final sequence proposals were obtained by filtering the $3{,}500 \times 3{,}000 = \sim 10$ million sequences explored for each independently trained sequence-to-function model. This was performed by finding the best sequence in each trajectory and then selecting the top P sequences among these

best-in-trajectory selections, where $P=300$ was the design budget. We did not perform any further filtering to ensure mutual diversity, as the selected sequences were already diverse in terms of the number of pairwise mutations separating them.

Library cloning and transformation. For library cloning and transformation, we assumed that we had available as input the output of a PCR reaction, in which the 5′ and 3′ ends contain type (T)IIS restriction sites compatible with Golden Gate assembly. For SynNeigh and FP Homologs, this corresponded to error-prone PCR product made with primers with appropriate TIIS flanking sequences. For each prospectively designed GFP and TEM-1 β-lactamase variant, corresponding DNA oligonucleotides contained 5′ and 3′ primer sequences such that their corresponding oligonucleotide pools could be amplified. Internal to these priming sequences were TIIS restriction sites that would be cut internally into the oligonucleotide containing the coding sequence of the variant and would consequently be 'clipped off' the priming sequences.

All library cloning and transformation was performed using the following general steps: (1) PCR of the vector backbone, (2) Golden Gate assembly of the insert and vector, (3) ethanol precipitation of the ligated plasmid and (4) electroporation into electrocompetent DH5α *E. coli*, recovery and subsequent outgrowth under selection.

PCR of vectors was performed with primers adjacent to the insert region that extended into the vector backbone. Vector primers were also adapted with TIIS restriction sites (either BsaI or BbsI), such that complementarity of 4 bp would be achieved with the library ('insert') on both the 5′ and 3′ end after digestion with the appropriate TIIS enzyme. PCR of vectors was performed using Q5 High-Fidelity 2X Master Mix (New England Biolabs). All GFP-related libraries were cloned using BsaI sites. The prospectively designed TEM-1 β-lactamase library was cloned using BbsI sites. PCR products of both inserts and vectors were bead purified using home-made SPRI beads[80].

PCR-amplified vector and library inserts were then cloned using a one-pot Golden Gate assembly reaction that contained TIIS restriction enzyme (BsaI-HF version 2 or BbsI-HF), T4 DNA ligase and DpnI. Reactions were cycled between 37 °C and 23 °C to encourage iterative cutting and ligation. All enzymes were ordered from New England Biolabs. Reactions were then precipitated with ethanol to purify the ligated plasmid in a form suitable for high-efficiency electroporation and then electroporated into DH5α *E. coli* (Lucigen, 10G Elite) cells using 0.1-cm electroporation cuvettes (Gene Pulser cuvettes, Bio-Rad) and a Bio-Rad MicroPulser. After electroporation, cultures were recovered in 1 ml recovery medium (Lucigen) for 1 h and subsequently grown overnight in LB with selection.

FlowSeq. Our FlowSeq procedure was adapted from Kosuri et al.[81]. For every FlowSeq experiment, we followed the steps described below.

Set-up

1. The night before, we grew 1-ml cultures of the following control strains: DH5α *E. coli*, DH5α *E. coli* expressing avGFP and DH5α *E. coli* expressing sfGFP.
2. The library (500 µl) (either frozen stock or outgrown transformation from the night before) was diluted 1:100 into 50 ml LB with selection and shaken at 37 °C. Control strains were handled similarly at a smaller scale.
3. Once cells for both the library and control strains reached an OD_{600} of 0.1–0.4, cultures were washed twice with 1× ice-cold PBS buffer.
4. Control avGFP and sfGFP strains were 'spiked' into the library at a representation of 0.1% to serve as internal standards.
5. Cells were passed through a 100-µm cell strainer and were kept on ice for 2 h.

Flow cytometry

6. All flow cytometry experiments were performed on a Sony SH800S cell sorter. Unless otherwise noted, all excitation lasers (405 nm, 488 nm, 561 nm, 638 nm) were turned on, and readings were taken and gates were drawn with respect to filter FL2 (525/25 nm). Thus, only the 405-nm and 488-nm lasers were relevant. We note that the FL2 measurement represents the emission induced by joint excitation with the 405-nm and 488-nm lasers.
7. We first flowed DH5α *E. coli* to determine FSC and SSC sensor gains and trigger thresholds. Using additional information from FSC and SSC area and height measurements, we drew a polygon gate to capture ~90% of singlet events, excluding likely doublets.
8. We next flowed the avGFP and sfGFP control strains to adjust the FL2 sensor gain such that there was good dynamic range between the non-fluorescent DH5α and the fluorescent avGFP- and sfGFP-expressing strains, without saturating the upper detection range. We confirmed that avGFP and sfGFP showed about one $\log_{10}$ difference in relative fluorescence. Finally, we flowed the library to confirm that its range of fluorescence values was well captured under these sensor settings.
9. We next drew B perfectly adjacent but non-overlapping gates or 'bins' to partition the entire range of fluorescence values observed across FL2 for the library. For generating the SynNeigh dataset, $B=17$. For FP Homologs, $B=8$, and for the prospectively designed GFP library (Fig. 2 in main text), $B=8$. The uppermost bin was always set such that it captured the upper tail of the fluorescence distribution. Bin minima and maxima were noted.
10. Library variants in each bin were then collected using two-way sorts. Sorts were made into polystyrene tubes filled with 1 ml LB with selection medium, and we noted the number of events that were sorted into each bin.
11. Sorted cells for each bin were then added to 10 ml LB with selection medium and grown overnight. Unused library (input into the flow cytometer) was pelleted and frozen at −20 °C. NGS
12. Cultures of each bin as well as the input library (hereafter 'input') were mini-prepped (Qiagen).
13. Illumina sequencing-ready amplicons of the library region (SynNeigh and prospectively designed GFP library) or barcode region (FP Homologs) of each sample were prepared using a two-stage PCR strategy. Sample multiplexing and pooling was accomplished with a standard dual-indexing strategy.
14. The amplicon pool was then purified with home-made SPRI beads, and quality control was performed with TapeStation analysis and with qPCR to ensure that the final pool was properly indexed, of the right length and accurately quantified.
15. To generate the SynNeigh dataset, we used a MiSeq 2×300-bp version 3 run to directly sequence the ~500-bp library region of the sequence encoding GFP. To generate the FP Homologs dataset, we used a NextSeq 2×75-bp mid-output run to sequence variant barcodes. When sequencing the prospectively designed GFP library, we sequenced the ~280-bp library region using a NextSeq 2×150-bp mid-output run.

Data processing and $\log_{10}$(relative fluorescence) inference

16. After sample demultiplexing, if multiple lanes were used during sequencing (NextSeq runs), their corresponding FastQ files were pooled.
17. For each sample, read pairs were merged using FLASH version 1.2.11 (refs. [82,83]).
18. For each merged read in each sample, the library region or variant barcode was extracted using a regular expression that identified delimiting constant primer sequences used for preparing the amplicon sequencing pools.
19. For each extracted region in each sample, protein sequences were determined by translating the directly sequenced or associated (in the case of variant barcodes as performed for FP Homologs) nucleotide sequence.
20. For each sample, the count of every unique protein sequence was then determined. And the total collection of unique protein sequences across all samples was used to create a variants×bins count table, **C**.
21. Using the metadata collected during flow cytometry, we could then infer the $\log_{10}$(relative fluorescence) values of each variant using the following procedure:
    a. Compute a relative abundance table, **R**, by dividing the columns of **C** by their sums. The columns of **R** sum to 1.
    b. Divide each column of **R** element-wise by the input relative abundance vector (relative abundance of variants in the library before flow cytometry) to obtain a fold-change table, **F**.
    c. Divide each row of **F** by its sum to obtain a table of adjusted abundances, **A**. Each row of **A** sums to 1.
    d. Each row of **A**, which corresponds to data for a particular protein variant, defines a discrete probability mass function, the flow cytometry bins over which the variant will appear. We therefore set the inferred $\log_{10}$(relative fluorescence) of variant i to be the median of the distribution $\mathbf{A}_i$.

Ancestral sequence reconstruction. We used the FastML web server to perform ASR[43]. A version or release was not available, but the tool was used on 21 October 2019. As input, we provided a multiple sequence alignment of Aequorean FPs. Otherwise, default FastML parameters were used: phylogenetic tree reconstruction method, RAxML; model of substitution, JTT; use gamma distribution, yes; probability cutoff to prefer ancestral indel over character, 0.5.

When examining the reconstructed phylogenetic tree, we isolated two interesting ancestral nodes, N1 and N11. N1 was the ancestor for all sequences, whereas N11 was an ancestor that excluded the *Aequorea macrodactyla* sequences TagCFP, OFPxm and TagGFP, which contain a large number of mutations relative to avGFP. From each node, we generated the top five most likely ancestral sequences at both N1 and N11. Because we were comparing ASR to model-guided approaches, ASR mutations outside of the 81-amino acid library regions were converted back to WT sequences. These designs were submitted as a Gene Fragments order to Twist Biosciences and cloned individually with Gibson assembly (reagents were from New England Biolabs).

Consensus sequence designs. Consensus sequence design attempts to sample the most probable sequences given a position weight matrix (PWM). We generated a PWM using the same sequence alignment that we used for ASR. To sample the highest-probability sequences from the PWM, we used a Metropolis–Hastings sampler to explore 180,000 sequences from which we filtered the top

five highest-probability sequences. Repeated runs of this procedure as well as multiple rarefaction analyses showed that we consistently captured the top two most probable sequences (manually derived) and that, beyond 180,000 explored sequences, no further improvements in sequence probabilities would be observed. The top five consensus sequence designs were submitted as a Gene Fragments order to Twist Biosciences and cloned individually with Gibson assembly (reagents were from New England Biolabs).

Fitness determination for TEM-1 β-lactamase variants. For each concentration of ampicillin (0, 250, 1,000, 2,500 μg ml^{-1}) and for each biological replicate, we prepared three large 150-mm plates of LB agar with ampicillin. We then prepared overnight starter cultures of two biological replicates expressing the cloned designed library and WT TEM-1 β-lactamase. On the day of the experiment, we back-diluted starter cultures 1:100 and allowed them to grow to $OD_{600}=0.5$, at which point we placed them on ice. Cells were then washed twice with ice-cold 1× PBS, and the WT strain was spiked into the library cultures at 0.1%. Cells (250 μl, about 6×10^8) were spread onto each prepared plate. Plates were incubated at 37 °C overnight.

The next day, plates were 'scraped' by adding 1 ml 1× PBS and 5–10 cell spreader beads. Plates were shaken laterally so that beads could dislodge colonies and mix cells into the PBS. This cell mixture was pooled for the three replicate plates for each antibiotic condition and biological replicate. These samples were then pelleted, mini-prepped and sequenced by NGS in the same manner as performed for FlowSeq. A 2×150-bp NextSeq run was used to sequence the library region. A design's fitness at a particular strength of antibiotic selection was determined to be the ratio of its relative abundance under selection to its relative abundance under no selection.

Exploration of evolutionary, structural and principal component mutational patterns in designs. In our examination of the mutational patterns in proposed and successful designs, we began by gathering high-quality position-specific scoring matrices (PSSMs) from the ProteinNet database[84] for both avGFP (PDB, 2WUR) and TEM-1 β-lactamase (PDB, 1ZG4) structures. These PSSMs are without gaps. We computed the 'effective number of mutations' per residue within our design window by taking the exponent of the per-position Shannon entropy, for example, $\exp\left(-\sum_i p_i \log(p_i)\right)$. For residues for which only one amino acid was observed in the multiple sequence alignment, the PSSM had 1 in that amino acid's position and 0 elsewhere, such that the effective number of mutations was 1. Likewise, if all amino acids were observed with equal frequency at that position, the effective number of mutations was 20.

For each position in the design window, we computed the relative frequency of mutation for the proposed and functional eUniRep designs. We counted the number of times a position was mutated to any residue outside the WT and divided it by the total number of mutations for each set.

We computed a least-squares regression between the mutation tolerance and relative mutation frequency using Scipy (https://docs.scipy.org/), including the r value and the P value (Fig. 4a,b, left). We also visualized the scatterplot of relative mutation frequency in proposed and gain-of-function designs along with the effective number of mutations (Fig. 4a,b, right).

Next, we used the experimentally determined crystal structures for both proteins to analyze relationships between mutation frequency and structural features. We first examined the Euclidean distance in three-dimensional space between the positions in the design window of avGFP and the centroid of the chromophore of avGFP (S65, Y66, G67). Likewise, we computed distances of positions within the design window of TEM-1 β-lactamase with the side-chain oxygen of catalytic serine S70. Instead of examining the per-position distance, we took all bright designs and computed the distribution of distances of all the mutated positions within each design and visualized the relationship between the quantitative function score ($\log_{10}$(relative fluorescence) and $\log_{10}$(fitness)) and the mean distance of mutated residues from the active site along with fifth and 95th percentile distances, computing a least-squares regression, r value and P value as above.

Using DSSP[85], we inferred per-position secondary structure annotations and relative solvent accessibility. For small residues without a DSSP annotation, we manually examined the crystal structure and classified the residue's secondary structure by eye. All positions with relative solvent accessibility less than 0.2 were classified as buried, and all others were exposed[86]. We visualized the frequency of mutations in our design window into each secondary structure category if we were to mutate uniformly randomly, the null expectation, and compared it to the mutation frequency we observed in proposed and >WT eUniRep designs (Fig. 4c,d, bottom). We colored the crystal structures of each protein by the relative per-position mutation frequency in >WT designs (Fig. 4c,d, upper center).

Lastly, we examined the relationship between function and the Euclidean space defined by eUniRep's vector representation. We sampled sequences with a random number of mutations ~ Poisson(4) + 1 (uniform across the sequence length) relative to WT for both proteins. eUniRep representations were computed for each, along with one-hot encoded matrices. We performed principal component analysis on the representations of this collection of random sequences and subsequently projected representations of the experimentally characterized random mutant sequences of avGFP from Sarkisyan et al.[42] and the single mutants of TEM-1 β-lactamase from Firnberg et al.[48] onto the first and second PCs of both eUniRep (avGFP and TEM-1 β-lactamase) and Full AA (avGFP and TEM-1 β-lactamase). Projected sequences points were colored by their quantitative function. We computed Pearson's correlation between the measured quantitative function and eUniRep PC1, as well as between the measured quantitative function and Full AA PC1.

Each model's ability to differentiate top >WT designed sequences from WT on the basis of predicted function (Fig. 4h,i), was defined to be the (signed) number of standard deviations that the predicted WT function was from the median of the top sequence design-predicted functions. For robustness, standard deviation was estimated using the median absolute deviation.

Assessing the robustness of evotuning to sequence set size and training time. To assess the robustness of the evotuning process, we examined how the quality of the UniRep representation varied with the size of the evotuning sequence set and training time. The quality of a particular representation was assessed by its ability to enable good supervised generalization. For GFP (Supplementary Fig. 9a), the Sarkisyan dataset, which is composed of functionally characterized sequences from the local fitness landscape of avGFP[42], was randomly partitioned into a 'training pool' of 41,715 mutants and a 'test pool' of 10,000 mutants. For each evotuned model, we randomly sampled $N=96$ mutants, built a top model as described previously, predicted the function of held-out mutants in the test pool and calculated the Spearman correlation between predicted and actual functions of test pool mutants. Low-N mutant sampling, top-model training and test pool evaluation was replicated ten times to assess the impact of low-N mutant selection. We performed the same analysis for TEM-1 β-lactamase, except that the Firnberg et al.[48] dataset was used (Supplementary Fig. 9b). The training pool contained 4,199 mutants, and the test pool contained 1,000 mutants.

Evotuning sequence set sizes were adjusted by randomly downsampling the full evotuning sequence sets for GFP and TEM-1 β-lactamase to 10%, 30% or 50%. Training times, as measured by the number of weight updates performed, were 0, ~2,000, ~7,000 and ~13,500 updates. Approximately 13,500 updates were used for the 'full' training of the model.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

Data required to reproduce all analyses in this work are provided or can be found at https://github.com/churchlab/low-N-protein-engineering. All referenced PDB structures were obtained from https://www.rcsb.org/. The Sarkisyan dataset was obtained from https://figshare.com/articles/dataset/Local_fitness_landscape_of_the_green_fluorescent_protein/3102154.

Code availability

Code for UniRep model training and inference with trained weights along with links to all necessary data is available at https://github.com/churchlab/UniRep. Code required to reproduce all analyses in this work is provided at https://github.com/churchlab/low-N-protein-engineering.

References

66. Xie, Q., Dai, Z., Hovy, E., Luong, M.-T. & Le, Q. V. Unsupervised data augmentation for consistency training. Preprint at https://arxiv.org/abs/1904.12848 (2019).
67. Berthelot, D. et al. MixMatch: a holistic approach to semi-supervised learning. Preprint at https://arxiv.org/abs/1905.02249 (2019).
68. Radford, A., Jozefowicz, R. & Sutskever, I. Learning to generate reviews and discovering sentiment. Preprint at https://arxiv.org/abs/1704.01444 (2017).
69. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. Preprint at https://arxiv.org/abs/1810.04805 (2018).
70. Potter, S. C., Luciani, A., Eddy, S. R. & Park, Y. HMMER web server: 2018 update. *Nucleic Acids Res.* **46**, W200–W204 (2018).
71. Caruana, R., Lawrence, S. & Giles, C. L. Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems* (NIPS, 2001).
72. Maclaurin, D., Duvenaud, D. & Adams, R. P. Early stopping is nonparametric variational inference. Preprint at https://arxiv.org/abs/1504.01344 (2015).
73. Yang, K. K., Wu, Z., Bedbrook, C. N. & Arnold, F. H. Learned protein embeddings for machine learning. *Bioinformatics* **34**, 2642–2648 (2018).
74. Lambert, T. J. FPbase: a community-editable fluorescent protein database. *Nat. Methods* **16**, 277–278 (2019).
75. Arnold, F. H. & Georgiou, G. (eds) *Directed Evolution Library Creation: Methods and Protocols*. (Humana Press, 2010).
76. van Dijk, D. et al. Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**, 716–729 (2018).
77. Le, Q. & Mikolov, T. Distributed representations of sentences and documents. In *Proc. 31st Int. Conf. Machine Learning* **32**, 1188–1196 (PMLR, 2014).
78. Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. Least angle regression. *Ann. Stat.* **32**, 407–499 (2004).

79. Sohka, T. et al. An externally tunable bacterial band-pass filter. *Proc. Natl Acad. Sci. USA* **106**, 10135–10140 (2009).
80. Oberacker, P. et al. Bio-On-Magnetic-Beads (BOMB): open platform for high-throughput nucleic acid extraction and manipulation. *PLoS Biol.* **17**, e3000107 (2019).
81. Kosuri, S. et al. Composability of regulatory sequences controlling transcription and translation in *Escherichia coli*. *Proc. Natl Acad. Sci. USA* **110**, 14024–14029 (2013).
82. Magoč, T. & Salzberg, S. L. FLASH: fast length adjustment of short reads to improve genome assemblies. *Bioinformatics* **27**, 2957–2963 (2011).
83. Stiffler, M. A., Hekstra, D. R. & Ranganathan, R. Evolvability as a function of purifying selection in TEM-1 β-lactamase. *Cell* **160**, 882–892 (2015).
84. AlQuraishi, M. ProteinNet: a standardized data set for machine learning of protein structure. *BMC Bioinformatics* **20**, 311 (2019).
85. Kabsch, W. & Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22**, 2577–2637 (1983).
86. Chen, H. & Zhou, H. X. Prediction of solvent accessibility and sites of deleterious mutations from protein sequence. *Nucleic Acids Res.* **33**, 3193–3199 (2005).

Acknowledgements

We thank M. AlQuraishi, C. Bakerlee, A. Chiappino-Pepe, A. Eremina, K. Fish, S. Gosai, X. Guo, E. Kelsic, S. Kosuri, P. Ogden, S. Sinai, M. Schubert, A. Taylor-Weiner, D. Thompson and A. Tucker for feedback on earlier drafts of this manuscript. We thank members of the Esvelt and Church laboratories for valuable discussion. S.B. was supported by an NSF GRFP Fellowship under grant number DGE1745303. G.K. was supported by a grant from the Center for Effective Altruism. E.C.A. was supported by a scholarship from the Open Philanthropy Project. This material is based upon work supported by the US Department of Energy, Office of Science under award number DE-FG02-02ER63445. Computational resources were, in part, generously provided by the AWS Cloud Credits for Research Program and Lambda Labs, Inc.

Author contributions

S.B., G.K. and E.C.A. conceived the study. S.B. performed wet-lab experiments and managed data. S.B., G.K. and E.C.A. performed machine learning modeling and data analyses. K.M.E. and G.M.C. supervised the project. S.B., G.K. and E.C.A. wrote the manuscript with help from all authors.

Competing interests

A full list of G.M.C.'s technology transfer, advisory roles and funding sources can be found on the laboratory's website at http://arep.med.harvard.edu/gmc/tech.html. S.B. is employed by and holds equity in Nabla Bio, Inc. G.K. is employed by and holds equity in Telis Bioscience Inc. E.C.A. and K.M.E. declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-021-01100-y.

Correspondence and requests for materials should be addressed to G.M.C.

Peer review information *Nature Methods* thanks Gabriel Rocklin, Guillaume Lamoureux, and the other, anonymous reviewer, for their contribution to the peer review of this work. Arunima Singh was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Reprints and permissions information is available at www.nature.com/reprints.

nature research

Corresponding author(s): George M. Church

Last updated by author(s): Jan 12, 2021

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☐	☒	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☐	☒	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☐	☒	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Code (including environment specifications/version numbers) required reproduce all analyses in this work are linked or provided at https://github.com/churchlab/low-N-protein-engineering.
Data analysis	Code (including environment specifications/version numbers) required to reproduce all analyses in this work are linked or provided at https://github.com/churchlab/low-N-protein-engineering.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

Data required to reproduce all analyses in this work are linked or provided at https://github.com/churchlab/low-N-protein-engineering.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Wherever possible sample sizes were chosen before hand such that the detectable effect size was an order of magnitude smaller than what we would reasonably have expected to observe in the experiment. For example, we evaluated N=300 prospective designs in order to be able to detect single digit percent differences in model effectiveness.
Data exclusions	N/A
Replication	Low-N hit rates and quantitative functions reached were obtained from two end-to-end biological replicates. Within each biological replicate, 5 prospective design replicates were performed. Low-N results were broadly replicated across two distinct proteins. Finally, hits obtained from these prospective design experiments were validated using single-plex/clonal characterization.
Randomization	Since our experiments consisted of prospective design (as opposed to observational study), randomization was not relevant. Any hidden covariates (e.g. researcher bias) were controlled for by randomizing samples during processing and blinding the researcher to the identity of each sample (see "Blinding" below).
Blinding	All experiments were done by coding samples before hand by an independent researcher unrelated to the project. The researcher/author on the project would then proceed and process these samples in a blinded manner.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☐	☒ Flow cytometry
☒	☐ MRI-based neuroimaging

Flow Cytometry

Plots

Confirm that:

☒ The axis labels state the marker and fluorochrome used (e.g. CD4-FITC).

☒ The axis scales are clearly visible. Include numbers along axes only for bottom left plot of group (a 'group' is an analysis of identical markers).

☒ All plots are contour plots with outliers or pseudocolor plots.

☒ A numerical value for number of cells or percentage (with statistics) is provided.

Methodology

Sample preparation	Described in detail in the Methods section of the manuscript
Instrument	Described in detail in the Methods section of the manuscript

Software	Described in detail in the Methods section of the manuscript
Cell population abundance	Described in detail in the Methods section of the manuscript
Gating strategy	Described in detail in the Methods section of the manuscript

☒ Tick this box to confirm that a figure exemplifying the gating strategy is provided in the Supplementary Information.