nature biotechnology

LETTERS

https://doi.org/10.1038/s41587-020-00793-4

Deep diversification of an AAV capsid protein by machine learning

Drew H. Bryant[1,8], Ali Bashir[1,8], Sam Sinai[2,3,4,5,8], Nina K. Jain[2,3], Pierce J. Ogden[2,3,7], Patrick F. Riley[1], George M. Church[2,3] ✉, Lucy J. Colwell[1,6] ✉ and Eric D. Kelsic[2,3,4] ✉

Modern experimental technologies can assay large numbers of biological sequences, but engineered protein libraries rarely exceed the sequence diversity of natural protein families. Machine learning (ML) models trained directly on experimental data without biophysical modeling provide one route to accessing the full potential diversity of engineered proteins. Here we apply deep learning to design highly diverse adeno-associated virus 2 (AAV2) capsid protein variants that remain viable for packaging of a DNA payload. Focusing on a 28-amino acid segment, we generated 201,426 variants of the AAV2 wild-type (WT) sequence yielding 110,689 viable engineered capsids, 57,348 of which surpass the average diversity of natural AAV serotype sequences, with 12–29 mutations across this region. Even when trained on limited data, deep neural network models accurately predict capsid viability across diverse variants. This approach unlocks vast areas of functional but previously unreachable sequence space, with many potential applications for the generation of improved viral vectors and protein therapeutics.

Engineering of protein phenotypes is limited by our ability to mutate multiple positions in a protein sequence and predict the functional outcome. Despite outstanding progress in computational de novo protein design[1–3], simulation-based predictions are challenging for large natural protein complexes. Moreover, biophysical models falter when modifications affect conformation, since the physical interactions that determine protein function are not well understood[4–6]. Directed evolution is a powerful approach[7–9], with the repeated application of random mutation and artificial selection often being the default engineering strategy when mechanistic understanding is limited, as is the case for proteins like AAV capsids[10,11]. Recent high-throughput DNA sequencing-based assays allow large-scale mapping of fitness landscapes[12–14], while advances in DNA synthesis and ML technologies enable a completely data-driven workflow for accelerated directed evolution[15–22]. However, it is unknown to what extent ML models trained on and around natural sequences can generate functional sequences substantially different from any natural homolog. We applied ML-guided diversification to the AAV capsid, a complex multiprotein assembly, as a case study to test whether data collected from high-throughput experiments can yield ML models that successfully guide the design of functional and diverse sequence variants. We validated our approach with a massively parallel experimental study to directly test the utility of machine learning for biological sequence design and diversification (Fig. 1a).

Adeno-associated virus capsids show great promise as gene delivery vectors. The AAV2 capsid is a component of the first gene therapy to receive approval by the US Food and Drug Administration for use in humans[23,24], while other serotypes are in clinical trials[25]. However, new AAV designs could overcome the limitations of current vectors, such as the immunity of patients with previous AAV exposure[26]. Previous engineering strategies, such as error-prone mutagenesis[11], random shuffling between AAV serotypes to create chimeric capsids[10], and random mutation at structurally guided positions[27] have shown limited success in overcoming antibody neutralization, because the resultant sequences remain quite similar to natural isolates. Epitopes for neutralizing antibodies occur at many locations across the capsid surface[28], indicating that capsids capable of avoiding neutralizing serum will require changes to many positions, most probably approaching or exceeding the diversity of natural serotypes (that is, on the order of hundreds of sequence differences). To evaluate a purely data-driven approach to diversification, we directly generated synthetic sequences near the threefold symmetry axis of the icosahedral AAV2 VP1 protein. Specifically, we targeted positions 561–588, a region that encompasses buried, surface and interface regions and overlaps both known heparin- and antibody-binding sites[28].

Capsid production represents a bottleneck in the creation of diverse AAV capsids because the majority of sequence variants fail to either assemble or package their genome[19,27,29]. To generate large and diverse datasets for training ML models of capsid production, we employed two strategies—choosing multi-mutants randomly or based on predictions from simple additive models. For the latter, we first assayed all single amino acid substitutions and insertions within the target region (Fig. 1b), finding that 58% were viable, meaning that they assemble an integral capsid that packages the genome. In contrast, randomly chosen multi-mutant sequence variants with between 2 and 10 mutations (Levenshtein distance) were only 10% viable, with only 0.3% viability for variants with at least 6 mutations (3 of 1,154). The yield of viable multi-mutants was improved by stochastically sampling from additive models fit on single-site data (Methods). We used this baseline approach to design 56,372 variants with between 2 and 39 mutations in the target region, testing the limits of exploration made possible given our previous data: 62.5% were viable, although none of the 1,790 variants with >21 mutations were viable.

To assess different protocols for ML-guided sequence design, we examined the impact of (1) training set design and (2) ML model architecture. We compared three ML training datasets designed via

[1]Google Research, Mountain View, CA, USA. [2]Wyss Institute for Biologically Inspired Engineering, Boston, MA, USA. [3]Department of Genetics, Harvard Medical School, Boston, MA, USA. [4]Dyno Therapeutics, Cambridge, MA, USA. [5]Department of Organismic and Evolutionary Biology, Harvard University, Cambridge, MA, USA. [6]Deptartment of Chemistry, University of Cambridge, Cambridge, UK. [7]Present address: Manifold Biotechnologies, Allston, MA, USA. [8]These authors contributed equally: Drew H. Bryant, Ali Bashir, Sam Sinai. ✉e-mail: gchurch@genetics.med.harvard.edu; lcolwell@google.com; eric.kelsic@dynotx.com

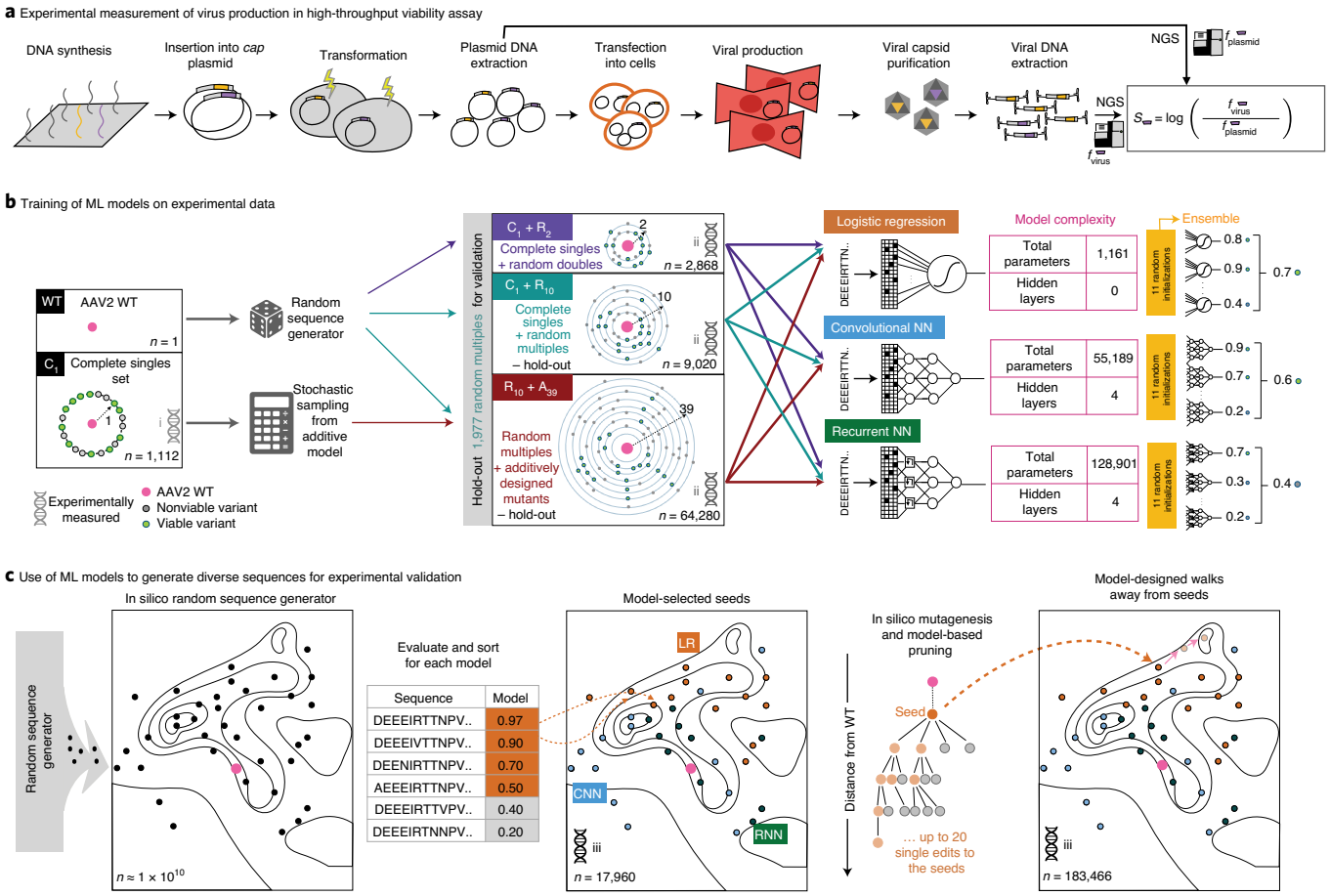

Fig. 1 | Generation of diverse sequence variants guided by ML models trained on deep mutational libraries. a, Experimental workflow: multiplexed measurement of viability for AAV capsid production. NGS, next-generation sequencing; *f*, frequency; *S*, computed viability score. Three experiments (helix markers (i, ii, iii) indicate separate experiments) were conducted to generate production data for (1) all single mutants, (2) ML training data and (3) ML validation data. **b**, ML model training workflow: experimental data from mutants generated by complete (C), random (R) or additive (A) sampling strategies were assembled into three training datasets: C_1+R_2, C_1+R_{10} and $R_{10}+A_{39}$. Subscripts indicate the maximum number of mutations relative to WT. Each dataset was used to train three ML models with varying architecture and increasing numbers of parameters: LR, CNN and RNN. **c**, Sequence design workflow: randomly generated candidates were ranked by model ensemble score to yield model-selected sequences. Top candidates were subjected to 20 iterative design cycles to obtain model-designed sequences.

complete (C), random (R) or additive (A) sampling strategies, splitting data from the previous experiment into three sets varying in the number of sequence variants and their distribution and distance from WT. These splits enabled assessment of how training data structure affects model performance (Fig. 1b). The smallest dataset, C_1+R_2, contains the complete set (C_1) of 1,112 possible single variants plus 1,756 randomly chosen sequence variants with 2 mutations. The C_1+R_{10} dataset contains C_1 together with R_{10}, 7,908 randomly chosen sequence variants with between 2 and 10 mutations, while the $R_{10}+A_{39}$ dataset contains R_{10} plus the 56,372 additive model-designed sequence variants with between 2 and 39 mutations, described above. A fixed set of 1,977 randomly chosen sequence variants with between 2 and 10 mutations was held out for hyperparameter tuning (Methods). To avoid overfitting to experimental noise, rather than predicting the quantitative production efficiency we used binary classification models to predict whether each sequence variant is viable (Supplementary Fig. 1), as defined by a threshold fit to best separate positive and negative controls (WT replicas, and variants containing stop codons, respectively).

Across each training set, we compared the performance of three model architectures: a simple logistic regression (LR) model, convolutional neural networks (CNNs) and recurrent neural networks (RNNs). For each of the nine resulting dataset–architecture combinations we trained an ensemble of 11 randomly initialized replica models and used the mean model score from each ensemble to rank 2.1 billion sequences (Fig. 1c), corresponding to 100 million sequences sampled uniformly at random at each distance from 5 to 25 steps from WT. For each ensemble, the 1,000 highest-scoring sequences at each distance were chosen as 'model-selected' seed sequences. However, in our random training dataset R_{10}, the proportion of viable capsid sequences drops rapidly as the distance from WT increases. Toward the goal of deep diversification, we therefore used the model ensembles to improve model-selected seed sequences. Briefly, to generate model-designed variants (Fig. 1c), we used the model ensembles to iteratively rank, filter and mutate (via single-residue edits) seed sequences for up to 20 rounds (Methods).

For each dataset–architecture combination, the highest-scoring model-selected and model-designed sequences at each distance between 5 and 29 from WT were synthesized and a total of 201,426 sequence variants were experimentally evaluated (Supplementary Tables 1–7). To verify reproducibility between the training and validation experiments we retested 2,000 sequences from the training set as controls, demonstrating strong experimental reproducibility ($R=0.89$, $P<10^{-20}$; Supplementary Fig. 2).

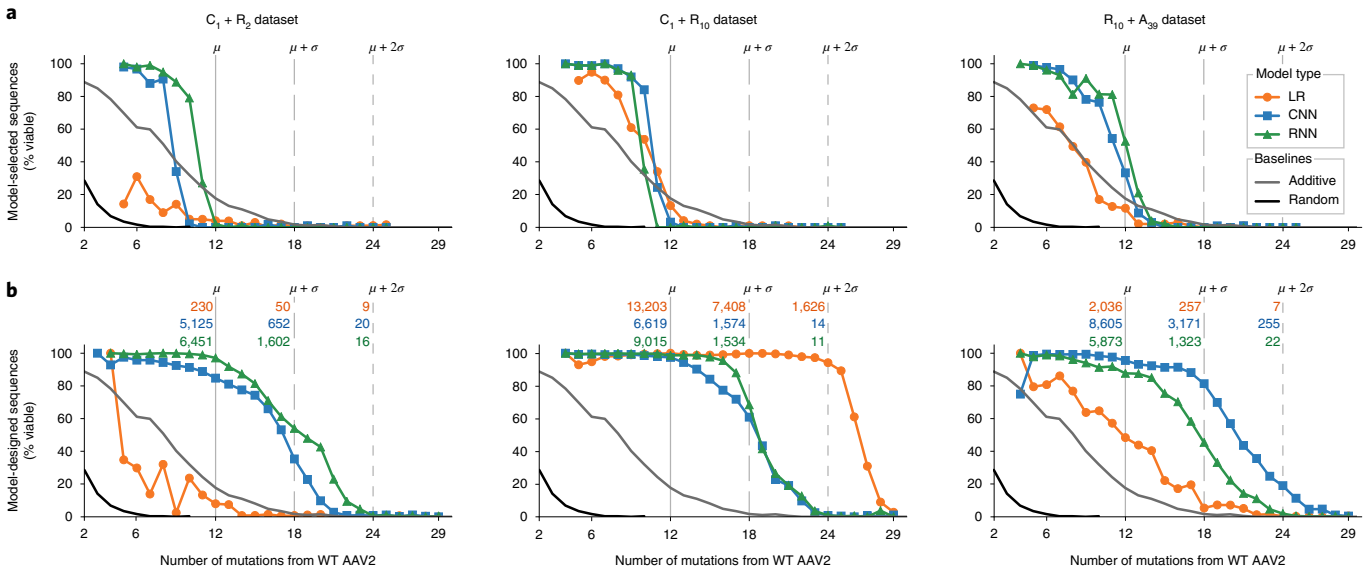

Fig. 2 | Experimental validation of synthetic sequences demonstrates high performance and robustness of NN models to training data composition. **a**, Performance of model-selected sequences. On each plot, black lines represent the randomly generated baseline ($n=10{,}997$ sequence variants) and gray lines represent the additive baseline ($n=56{,}372$ sequence variants). Vertical lines denote the number of mutations within natural AAV serotypes in the target region, on average ($\mu=12$), plus additional standard deviations ($\sigma=6$). **b**, Performance of model-designed sequences. Colored numbers denote viable capsids with at least the indicated number of mutations. Aggregated statistics are available in Supplementary Tables 1–7.

Model-guided design was dramatically successful at generating diverse viable sequence variants. Within this region, diverse natural AAV serotypes differ from AAV2 on average at $\mu=12\pm\sigma=6$ positions (s.d.). Model-selected sequences from CNN and RNN models showed close to 100% viability at 6 mutations from WT (Fig. 2a), the threshold at which randomly chosen sequence variants were largely nonviable. However model-selected viability dropped quickly beyond 12 mutations from WT, most probably because the randomly generated candidate sequences from which the models had to choose were overwhelmingly nonviable. In contrast, many model-designed sequences with >12 mutations from WT were viable (Fig. 2b). Overall, 58.1% of model-designed sequences (106,665 in total) formed viable capsids with up to 29 mutations from the WT sequence, including variants with up to 19 substitutions or 15 insertions within the 28-residue target segment. On average, the neural network (NN) model-designed sequences were 33 times more likely to be viable than those designed by the additive model at 18 mutations ($\mu+\sigma$) from WT, with even greater improvements at larger distances.

The performance of NN models was robust to variations in the amount and composition of training data. While the LR model trained with the medium-sized C_1+R_{10} dataset was >90% viable as far as 24 mutations ($\mu+2\sigma$) from WT, LR models trained on the smallest (C_1+R_2) and largest ($R_{10}+A_{39}$) datasets were unreliable (Fig. 2b). In contrast, CNN and RNN models trained on the smallest (C_1+R_2) dataset successfully designed many variants with >18 mutations ($\mu+\sigma$) from WT, comparable to those trained on the ~threefold larger C_1+R_{10} and ~22-fold larger $R_{10}+A_{39}$ datasets (Fig. 2b). We note that all models benefitted from the decision to use ensembles (Supplementary Fig. 3). Across all models, and the LR models most markedly, we observed that the inclusion of more training data does not guarantee better model performance. To better understand this observation, we turned to analyzing the diversity of designed variants.

Models differed in the levels of sequence diversity that they generated. The first two-thirds of the target region is more conserved across natural AAV sequences, probably because these positions are less surface exposed and are constrained by the oligomeric interface (Fig. 3a). While the performance of models trained on the C_1+R_{10} dataset was uniformly high, NN models successfully incorporated diverse residue substitutions at buried and interface sites much more frequently than the LR model (Fig. 3b and Supplementary Fig. 4). Additionally, NN models successfully incorporated many insertions into the buried part of the capsid, which is intolerant of insertions in general (Fig. 3b). While the LR(C_1+R_{10}) model had strong preferences for particular amino acids at each position (as seen by its low perplexity, Fig. 3b), RNN models exhibited preference for substituting amino acids with similar chemical properties while CNN models tended to be more selective among positions (Fig. 3b). Moreover, while all models were capable of mutating the later, surface-accessible, portion of the target region, NN models incorporated a greater diversity of amino acids at these positions (Fig. 3b). The LR($R_{10}+A_{39}$) model exhibited greater diversity (Fig. 3b) but relatively poor precision (Fig. 2b), indicating the importance of sequence context when mutating to more diverse sets of amino acids at each position. Conversely, while the LR(C_1+R_{10}) model had the highest precision of all models, the greater per-position diversity of the NN models suggested that their sequence proposals were distributed across a much larger region of sequence space.

To test this hypothesis, we quantified model diversity by calculating the number of clusters obtained when the viable sequences designed by each model were clustered using pairwise Levenshtein (edit) distance (Methods). For all datasets, CNN and RNN models identified viable sequences covering much larger volumes of sequence space than the LR models (Fig. 4a). The LR model with highest performance (C_1+R_{10}) was also the least diverse, primarily generating highly similar viable sequences. Pure maximization of precision or diversity can result in a trade-off: selecting only the highest-scoring sequence may be precise but results in no diversity, whereas randomly generated sequences have high diversity but low precision. Of course, models can also have low diversity and low precision (for example, LR(C_1+R_2)).

To quantitatively evaluate model performance in this respect, we further clustered all designed sequences using a radius of 12 edits (μ) and computed average viability within the resulting clusters. At all viability thresholds, NN models outperformed LR models. RNN performed best for the smallest (C_1+R_2) dataset, while CNN

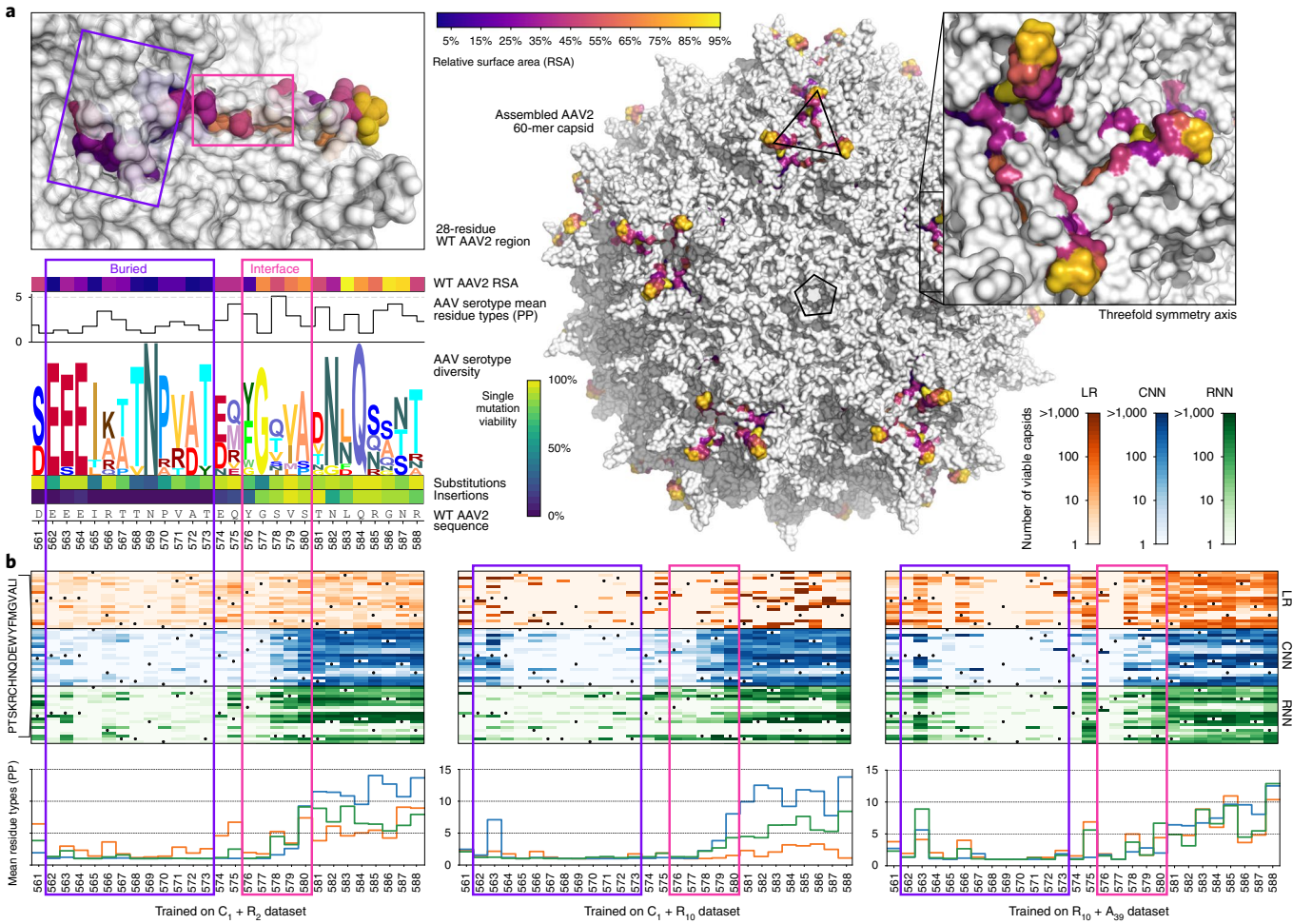

Fig. 3 | Neural network models generate greater diversity across positions. a, Three-dimensional structure of the 28-residue region with boxed buried (purple) and interface regions (pink) colored by relative surface area (RSA) for a single monomer, shown in context with interfacing monomers. Average tolerance to single substitutions and insertions (measured experimentally) is shown for each position, along with the perplexity (PP) and natural diversity across 12 common serotypes (logo plots created based on ref. [31]). **b**, Top: heatmaps showing successful substitutions within viable capsids (≥12 mutations) as designed by each model trained on each dataset. WT residues (dots) are masked. Bottom: mean number of residue type substitutions incorporated by position.

performed better for the larger datasets (Fig. 4b). Projecting viable sequences from the C_1+R_{10} models into two dimensions with ivis[30] provides visual intuition: the CNN model generated viable capsids across much larger regions of sequence space than the highly accurate LR(C_1+R_{10}) model, although we note that all models detected viable sequence variants that are highly distinct from natural AAV serotypes (Fig. 4c and Supplementary Fig. 4b–d). In summary, our CNN and RNN design strategies were more successful at deep diversification than LR at all precision levels and across all datasets, although better strategies are certainly possible and additional work will determine how these findings generalize to other contexts.

The success of these diversification strategies (1) addresses the immediate need for engineered AAV capsids with sequences distinct from natural isolates and (2) demonstrates that data-driven models can perform well on complex proteins without incorporating extensive domain knowledge or physical models, even with limited training data (as shown here by the success of models trained using <3,000 data points). For AAV, the diverse set of viable sequence variants discovered by the NN models are promising candidates to test for additional gain-of-function phenotypes, such as improved cell tropisms and manufacturability. More generally, models can be trained to simultaneously predict multiple phenotypes to jointly optimize variants for several desirable properties, a task that is substantially more challenging for traditional methods of directed evolution.

While many ML studies are conducted on a single standardized dataset where differences in only model architecture choices are compared, our study highlights the value of optimizing training data distributions for improved predictive power. The fact that relatively small, simple and unbiased training sets enable viability predictions far from WT suggests that similar approaches can be used for proteins in which high-throughput screens are impractical. Importantly, after such models have been trained, the generation of new sequences requires only additional computing time, bringing a vast number of diverse and functional synthetic variants within reach. This study lays the foundation for the efficient model-guided exploration of deep sequence space, empowering both basic biology and protein engineering.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of

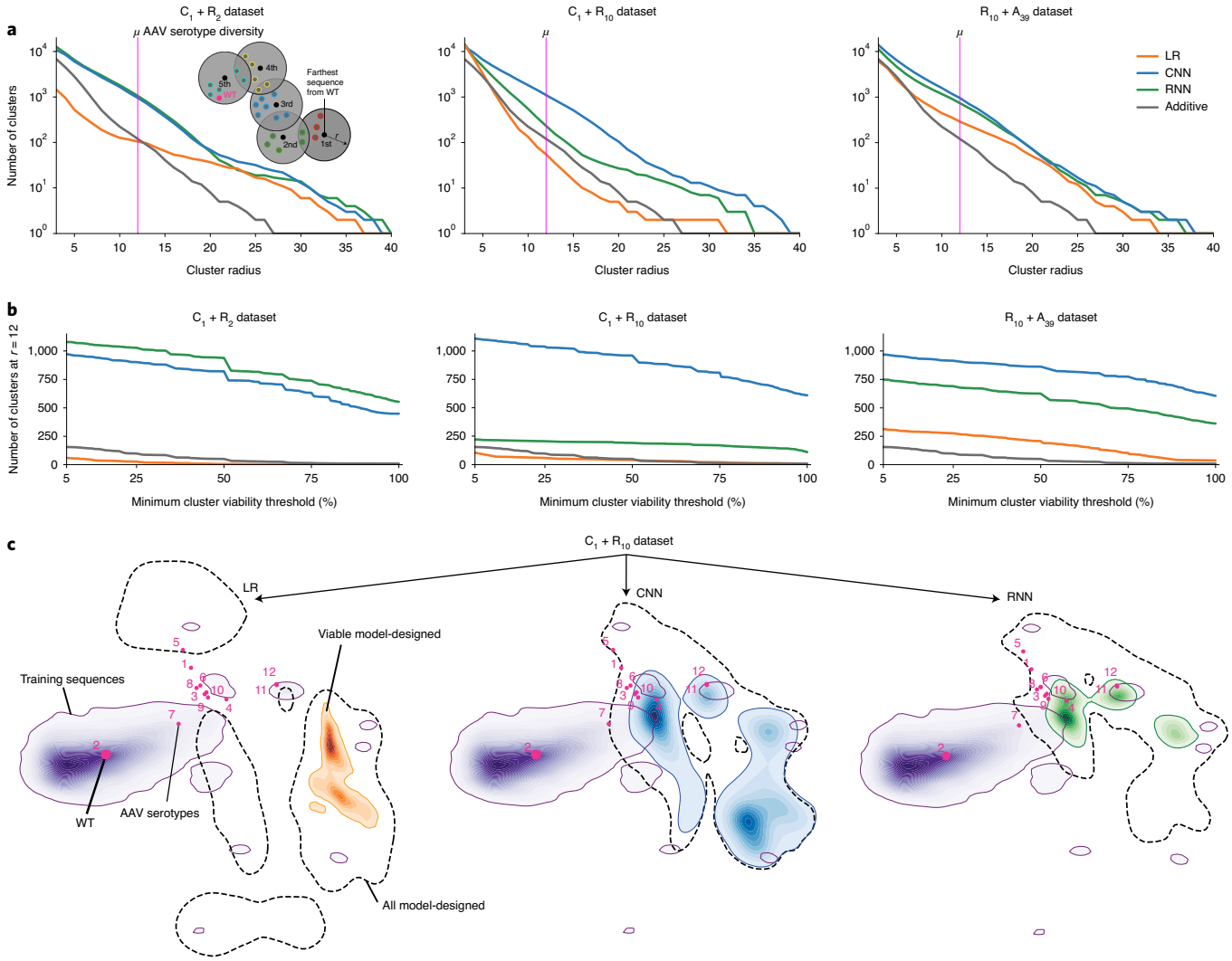

Fig. 4 | Neural networks generate greater functional diversity at equivalent levels of performance relative to additive and LR models. a, Number of distinct viable sequence clusters as a function of cluster radius. Inset: method for sequence clustering, radius r measured by Levenshtein distance (https://pypi.org/project/python-Levenshtein/). **b**, Number of clusters for which models predicted viable mutants at or above a minimum performance threshold. Cluster radius $r = 12$, the average AAV natural serotype diversity (μ) within the target region. **c**, Visualization of sampled diversity through ivis projections[30]. Purple: $C_1 + R_{10}$ training data (identical between panels); dashed outline: area containing all model-designed sequences; orange/blue/green: kernel density estimates for the viable capsid subset for LR, CNN and RNN models, respectively; magenta: natural AAV serotypes (1–12) embedded for reference.

author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41587-020-00793-4.

Received: 7 March 2020; Accepted: 8 December 2020;
Published online: 11 February 2021

References

1. Huang, P. S. et al. High thermodynamic stability of parametrically designed helical bundles. *Science* **346**, 481–485 (2014).
2. Butterfield, G. L. et al. Evolution of a designed protein assembly encapsulating its own RNA genome. *Nature* **552**, 415–420 (2017).
3. Langan, R. A. et al. De novo design of bioactive protein switches. *Nature* **572**, 205–210 (2019).
4. Weinreich, D. M., Delaney, N. F., DePristo, M. A. & Hartl, D. L. Darwinian evolution can follow only very few mutational paths to fitter proteins. *Science* **312**, 111–114 (2006).
5. Halabi, N., Rivoire, O., Leibler, S. & Ranganathan, R. Protein sectors: evolutionary units of three-dimensional structure. *Cell* **138**, 774–786 (2009).
6. Ferretti, L., Weinreich, D., Tajima, F. & Achaz, G. Evolutionary constraints in fitness landscapes. *Heredity* **121**, 466–481 (2018).
7. Stemmer, W. P. Rapid evolution of a protein in vitro by DNA shuffling. *Nature* **370**, 389–391 (1994).
8. Fox, R. J. et al. Improving catalytic function by ProSAR-driven enzyme evolution. *Nat. Biotechnol.* **25**, 338–344 (2007).
9. Davis, A. M., Plowright, A. T. & Valeur, E. Directing evolution: the next revolution in drug discovery? *Nat. Rev. Drug Discov.* **16**, 681–698 (2017).
10. Grimm, D. et al. In vitro and in vivo gene therapy vector evolution via multispecies interbreeding and retargeting of adeno-associated viruses. *J. Virol.* **82**, 5887–5911 (2008).
11. Dalkara, D. et al. In vivo-directed evolution of a new adeno-associated virus for therapeutic outer retinal gene delivery from the vitreous. *Sci. Transl. Med.* **5**, 189ra76 (2013).
12. Araya, C. L. et al. A fundamental protein property, thermodynamic stability, revealed solely from large-scale measurements of protein function. *Proc. Natl Acad. Sci. USA* **109**, 16858–16863 (2012).
13. Sarkisyan, K. S. et al. Local fitness landscape of the green fluorescent protein. *Nature* **533**, 397–401 (2016).
14. Poelwijk, F. J., Socolich, M. & Ranganathan, R. Learning the pattern of epistasis linking genotype and phenotype in a protein. *Nat. Commun.* **10**, 4213 (2019).
15. 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).

16. Wu, Z., Kan, S. J., 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).
17. 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).
18. Kelsic, E. D. & Church, G. M. Challenges and opportunities of machine-guided capsid engineering for gene therapy. *Cell Gene Ther. Insights* **5**, 523–536 (2019).
19. 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* **366**, 1139–1143 (2019).
20. Liu, G. et al. Antibody complementarity determining region design using high-capacity machine learning. *Bioinformatics* **36**, 2126–2133 (2020).
21. Brookes, D. H., Park, H. & Listgarten, J. 2019. Conditioning by adaptive sampling for robust design. *Proc. 36th Intl Conf. Machine Learning, PMLR* **97**, 773–782 (2019).
22. Yang, K. K., Wu, Z. & Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nat. Methods* **16**, 687–694 (2019).
23. Russell, S. et al. Efficacy and safety of voretigene neparvovec (AAV2-hRPE65v2) in patients with RPE65-mediated inherited retinal dystrophy: a randomised, controlled, open-label, phase 3 trial. *Lancet* **390**, 849–860 (2017).
24. Dunbar, C. E. et al. Gene therapy comes of age. *Science* **359**, eaan4672 (2018).
25. Mendell, J. R. et al. Single-dose gene-replacement therapy for spinal muscular atrophy. *New Engl. J. Med.* **377**, 1713–1722 (2017).
26. Calcedo, R., Vandenberghe, L. H., Gao, G., Lin, J. & Wilson, J. M. Worldwide epidemiology of neutralizing antibodies to adeno-associated viruses. *J. Infect. Dis.* **199**, 381–390 (2009).
27. Tse, L. V. et al. Structure-guided evolution of antigenically distinct adeno-associated virus variants for immune evasion. *Proc. Natl Acad. Sci. USA* **114**, E4812–E4821 (2017).
28. Tseng, Y. S. & Agbandje-McKenna, M. Mapping the AAV capsid host antibody response toward the development of second generation gene delivery vectors. *Front. Immunol.* **5**, 9 (2014).
29. Adachi, K., Enoki, T., Kawano, Y., Veraz, M. & Nakai, H. Drawing a high-resolution functional map of adeno-associated virus capsid by massively parallel sequencing. *Nat. Commun.* **5**, 3075 (2014).
30. Szubert, B. & Drozdov, I. ivis: dimensionality reduction in very large datasets using Siamese Networks. *J. Open Source Softw*. https://doi.org/10.21105/joss.01596 (2019).
31. Wheeler, T. J., Clements, J. & Finn, R. D. Skylign: a tool for creating informative, interactive logos representing sequence alignments and profile hidden Markov models. *BMC Bioinformatics* **15**, 7 (2014).

Methods

AAV mutant sequence library generation and production assay. Libraries were constructed using a method similar to that previously described by Ogden et al.[19]. For the final validation experiments, 184-mer DNA oligonucleotides (oligos) were synthesized as single-stranded DNA by Agilent. Designed amino acid sequences were back-translated to nucleotide sequences by choosing any possible codon (generally keeping the WT codon and selecting mutant amino acid codons with no bias, but disallowing codon choices that created restriction enzyme sites used in cloning). From 5′ to 3′, each oligo contained a forward primer binding site, a BbsI restriction site (5′-GAAGACAT|TACA-3′), a nucleotide mutant coding sequence (with at least 84 nucleotides), a BsaI restriction site (5′-CAAG|CGAGACC-3′), an EcoRV 'kill-cutter' restriction site, a BsaI restriction site in the opposite orientation (5′-GGTCTCA|CGCT-3′), an 18-nucleotide barcode sequence, a BbsI site in the opposite orientation (5′-CGCT|AAGTCTTC-3′) and a reverse primer binding site. Note that barcodes were included in the synthesized oligos but were not used for downstream sequencing or analysis (rather, the mutant coding region was directly amplified and sequenced, matching the amplification method used in the previous production experiments). PyDNA[32] was used for in silico testing of the cloning process, ensuring sequence compatibility with the cloning strategy. The code for executing this process is fully provided in the synthesis pipeline component of the bioinformatics pipeline code (Code availability and Nature Research Reporting Summary).

An example oligo with WT coding region is 5′-GGGTCACGCGTAGG AGAAGACATTACAGACGAAGAGGAAATCAGGACAACCAATCCCGTGG CTACGGAGCAGTATGGTTCTGTATCTACCAACCTCCAGAGAGGCAA CAGACAAGCGAGACCGATATCGGTCTCACGCTGTAATGCGGTCTGA GCCGCGCTAAGTCTTCGTGTGGCTGCGGAAC-3′.

Cloning was carried out in three steps. First, oligos were PCR amplified using Q5 high-fidelity DNA polymerase (NEB, no. M0492) and an annealing temperature of 60 °C (forward: 5′-GGGTCACGCGTAGGA-3′, reverse: 5′-GTTCCGCAGCCACAC-3′). A backbone plasmid containing the WT AAV2 *cap* gene was also amplified with Q5 and an annealing temperature of 72 °C (forward: 5′-TTGGTCTCA|CGCTAGAGACGGTGTGGCTGCGGAAC-3′, reverse: 5′-AAGGTCTCC|TGTAATCATGACCTTTTCAATGTCCACATTTG-3′). This PCR was used to add BsaI sites with overhangs complementary to BbsI overhangs in the oligos (as indicated by cut sites in primer sequences). Amplified oligos were digested with BbsI-HF (NEB, no. R3539), and amplified plasmid was digested with BsaIHF-v2 (NEB, no. R3733) in separate 50-µl reactions. Digest products were purified using homemade SPRI beads, mixed at a 3/1 molar ratio (oligos/plasmid) and ligated using T4 DNA ligase (2×10^6 U ml^{-1}; NEB, no. M0202). Ligation products were ethanol precipitated and transformed into 50 µl of electrocompetent cells (Lucigen 10G SUPREME, no. 60081). Following 1-h recovery at 37 °C, cells were added to 4 ml of selection medium (2× YT with kanamycin) and grown at 37 °C overnight. The following morning, step 1 library plasmids were mini-prepped by alkaline lysis (Qiagen, no. 27104). In this first cloning step, oligo sequences replaced the corresponding 84-base-pair WT sequence and the 3′ region of the *cap* gene in the backbone plasmid.

In the second cloning step, an amplicon containing the 3′ WT region of the *cap* gene was generated from the initial backbone plasmid using Q5 and an annealing temperature of 72 °C (forward: 5′-TTGGTCTC A|CAAGCAGCTACCGCAGATGTCA-3′, reverse: 5′-AAGGTCTCA|A GCGAGAGACGTCCTACGCGTGACCC-3′). This amplification step was also used to add BsaI sites complementary to those in the oligos. Step 1 library plasmids and the 3′ WT amplicon were separately digested with BsaI-HFv2, bead purified and ligated as above. Ligation products were digested with EcoRV-HF (NEB, no. R3195) in a kill-cutting step to remove step 1 plasmids that did not incorporate the 3′ WT amplicon. EcoRV digest products were ethanol precipitated, transformed and mini-prepped as in step 1. In the third cloning step, a destination ITR-containing plasmid was digested with HindIII-HF (NEB, no. R3104) and SpeI-HF (NEB, no. R3133). Complete mutant *cap* gene sequences were amplified from the step 2 plasmid library using Q5 and an annealing temperature of 70 °C (forward: 5′-AGGTCTCA|AGCTTCGATCAACTACGCAGACAG-3′, reverse: 5′-AGGTCTCA|CTAGATGAGCTCGTCGACGTTCC-3′). This amplification step was also used to add BsaI sites and overhangs complementary to the HindIII and SpeI sites in the ITR plasmid. Amplicons were digested with BsaI-HFv2 as above. Digested ITR plasmid and step 2 library amplicons were bead purified, ligated, transformed and mini-prepped to generate the final plasmid library. For the earlier rounds of library cloning, creation of mutant *cap* genes was accomplished in a single cloning step since oligos did not contain BsaI sites, EcoRV sites or barcode sequences, enabling cloning directly into the corresponding position in the WT *cap* gene. Ligation sites and oligo and *cap* PCR primers for this single-step cloning were the same as above. Similarly, the final library cloning step to move the mutant *cap* gene sequences into the ITR plasmid remained the same.

The final plasmid library was transfected into HEK293T cells to produce viral particles. Cells were grown in DMEM (ThermoFisher, no. 10566016) supplemented with 10% fetal bovine serum (ThermoFisher, no. 10082147) and seeded in five-layer cell stacks (Corning, no. 353144) 2 d before transfection. Polyethylenimine (PEI) was used for transfection at a mass ratio of 3/1; 125 ug of adenovirus pHelper plasmid, 75 ug of an AAV *rep* plasmid and 1 ug of library plasmids were mixed with PEI, incubated for 20 min and added to cells. Media were changed completely at the time of transfection and replicate transfections were carried out in separate cell stacks. Here, the lower levels of library plasmid were chosen to reduce the number of plasmids transfected into individual cells, such that the potential for mosaic capsid formation and cross-packaging was minimized. Three days post-transfection, 5 M NaCl was added to the cultures for a final concentration of 0.5 M and cultures were incubated at 37 °C for 3 h. Following incubation, mixtures were transferred to fresh containers and incubated at 4 °C overnight. The next day, the resulting supernatants were run through 0.22-µm PES filters (Corning, no. 431098); 40% PEG-8000 was then added to give a final concentration of 8%, and mixtures were incubated at 4 °C for 3 h. Samples were centrifuged at 3,000*g* for 20 min to pellet the PEG precipitate, and pellets were resuspended in 7 ml of DPBS. Viral genomes external to the capsid and carryover plasmid DNA were degraded with benzonase; a 10,000-fold dilution in benzonase (Millipore Sigma, no. 1.01695.0001) was added to resuspended pellets, and samples were incubated at 37 °C for 45 min. Encapsidated genomes were separated from the remaining cellular debris using iodixanol ultracentrifugation and concentration via size-exclusion spin filters as described previously[19,33]. Briefly, benzonase-treated samples were underlaid with an iodixanol gradient (Sigma, no. D1556) in polypropylene tubes (Beckman Coulter, no. 362183) and centrifuged at 242,000*g* for 1 h at 16 °C. Capsids were collected from the 40% iodixanol fraction and concentrated using a spin concentrator (Millipore Sigma, no. UFC910024) to generate the final purified pool.

Cap gene sequences remaining in the purified pool represent mutants viable for capsid assembly and genome packaging. Purified capsids were heat denatured at 98 °C for 10 min, and PCR was run with Q5 and an annealing temperature of 65 °C to amplify the mutant region of the *cap* gene (forward: 5′-GCTCAGAGAAAACAAATGTGGAC-3′, reverse: 5′-GAACGCCTTGTGTGTTGACATC-3′). PCR reactions were carried out in the presence of EvaGreen (Biotium, no. 31000) and run on a BioRad CFX96 quantitative PCR machine to ensure that reactions were stopped during the exponential phase. Illumina sequencing adapters and indices were added in a subsequent PCR. These PCR amplicons were sequenced with overlapping paired-end reads using Illumina NextSeq. Paired-end reads were merged to generate a consensus read using PEAR[34], and read counts were calculated for every member of the designed library. Reads with a minimum Q score of 20 were selected for four technical plasmid replicates and three biological virus replicates (each with at least two technical replicates; Supplementary Fig. 2). Mutant fitness in the viral production assay was calculated by taking the ratio of mutant read counts in the viral library over the counts in the original DNA library, normalizing by the ratio of the WT sequence.

Measurement of viral genome abundances from tissues for the design of the A_{39} dataset was done via amplicon sequencing from purified vector genomes, with PCR protocols as described above. Three separate batches of virus were prepared and 3.5×10^{10} (batch 1), 2.3×10^{10} (batch 2) and 3.5×10^{10} viral genomes (batch 3) of the C_1 virus library were diluted in 200 µl of PBS and injected into mouse, four mice per batch, for 12 mice in total. Mice were all 8 weeks old, male and C57BL/6J. For each of the three batches, two mice were injected retro-orbitally and two intraperitoneally. After 1, 5 and 24 h, 30 µl of blood was drawn by facial bleed and frozen on dry ice, then at −80 °C. After 8 d, mice were dissected and tissue samples from liver, kidney, heart, lung, brain, spleen, muscle, skin, stomach and testis were frozen on dry ice, then at −80 °C. Approximately 150 mg of each organ was ground using a disposable mortar and pestle (Kimble Chase, no. 749625-0010). DNA was purified from tissues using alkaline lysis (Qiagen, no. 27104), and from blood using the Qiagen MinElute Virus Spin Kit (no. 57704). Biodistribution was similar across both routes of administration. The overall effect on biodistribution of viral genomes for each organ and blood sample was calculated in R using deseq2 across measurements from multiple mice, combining 12 mice for blood and liver and four mice from batch 2 for the remaining organs. The animal protocol was approved by the Harvard Medical School Institutional Animal Care and Use Committee.

Random sampling of AAV2 mutants around WT. To generate a sequence at mutation distance k steps from WT AAV2, a uniform random draw from the set of 28 WT + 28 insertion positions was first made. This mutation was then removed from the consideration set, and $k-1$ subsequent draws without replacement from the remaining set of unsampled positions were made until k distinct mutation positions were selected. For each of the k positions selected, a residue type was selected uniformly at random: for insertion positions, all 20 standard amino acids were available; for substitution positions, the 19 amino acids distinct from WT were available. The set of k mutations relative to the WT AAV2 sequence then fully defines a mutant sequence at distance k from WT.

Baseline random sequence set generation. The train (7,908 variants) + tune (1,977 variants) random multi-mutant sequence sets were generated by sampling 1,732–1,756 sequences at each distance of two to six steps away from WT, inclusive, and 288–290 sequences from seven to ten steps, inclusive. In total, the random multi-mutant sequence baseline experiment tested 9,885 unique sequences between two and ten steps, inclusive.

Baseline additive model sequence set generation. The biodistribution of the C_1 library across liver, kidney, heart, lung, brain, spleen, muscle, skin, stomach, testis and blood samples was used to compute selection scores (relative enrichment of variants in tissue versus the original plasmid library) for each sample. All data contained information about production ability, because viral production is a necessary requirement for observation of viruses in each tissue and is therefore a common contributor to variance across all models. We generated mutants for the A_{39} set using biodistribution data rather than simply production data, to facilitate enrichment of variants with diverse biodistribution phenotypes within the additive set; however, we focused on training ML models using the production assay measurements because these higher-accuracy measurements enabled us to better assess the predictive power of our models during the final round of validation experiments.

We generated random mutants in three ways, by allowing the following: (1) substitutions across the region, (2) substitutions and insertions (but no more than one amino acid between two positions) and (3) substitutions but restricting the same insertions to the second half of the tile.

To design variants from single-mutant data with the additive model, we employed three types of Monte Carlo sampling, as follows:

1. For each position along the region of interest, we constructed a Boltzmann distribution defined as $2^{s_i/T}/Z$, where s_i is the tropism for amino acid i in that position as measured in the singles library (for different tissues) and Z ensures that the sum of probabilities across the position equals 1. The temperature parameter. T, controls the degree of fidelity to the best-proposed mutation according to the additive model, with higher T resulting in more diverse choices but lower expected fitness gain. We then combined mutations by scanning across the region of interest and sampling amino acids probabilistically for each position (potentially WT); T was fixed during the generation of each variant. However, to produce a diverse library we varied T between ~10^{-2} and ~10^{0} (with increments of 0.18 in the exponent) for different variants. The A_{39} dataset contains 18,155 unique sequence variants generated using this process.
2. For each position along the region of interest, we sampled uniformly from a subset of amino acids that had selective advantage above threshold t_s. We varied t_s between −1 and 2 to induce further variation. The A_{39} dataset contains 23,420 unique sequence variants generated using this process.
3. For each variant, we would randomly sample multiple single edits and accept the variant only if the sum of effects from individual mutations was above the threshold, t_m. We varied t_m between 0 and 2.33 to induce variation. The A_{39} dataset contains 14,797 unique sequence variants generated using this process.

For variants with multiple mutations against WT reference, we would sometimes also sample related variants by individually introducing mutations included in the variant. The order in which these mutations was introduced was either greedy (meaning better mutations introduced first) or random; hence these sets of mutations would entail a stepwise 'path' from WT to the target variant. Additionally, we sampled around 11,000 unique variants randomly.

Construction of ML training datasets C_1+R_2, C_1+R_{10} and $R_{10}+A_{39}$. Our experimental design compares three libraries of training data, each containing different numbers of sequence variants that were sampled from a constrained interval of sequence space around the WT AAV2 sequence, using three distinct sampling strategies. The additive dataset (A_{39}) provides a baseline training dataset in which mutants were generated first by measuring the complete set of single mutants and then generating diverse mutants using additive models (see preceding section). In contrast, the other two libraries (R_2 and R_{10}) exploit the power of random sampling to select sequences with multiple mutations sampled uniformly at random from the sequence space around the WT sequence, and are more efficient in that they require only one experiment to generate training data.

The C_1+R_2 dataset ($n=2{,}868$, 40% viable) contains (1) the complete set of single-site mutants, C_1 ($n=1{,}112$, 58% viable) and (2) a <1% random subset of possible double mutants, R_2 ($n=1{,}756$, 29% viable). The types of single mutants allowed in this study included all possible substitutions at the 28 residue positions considered and all possible single-residue insertions between and surrounding the 28 positions—that is, 29 possible insertion positions, resulting in 29×20 (insertions) + 28×19 (non-WT substitutions) = 1,112 single-site mutants.

The C_1+R_{10} dataset ($n=9{,}020$, 16% viable) contains (1) the complete set of single-site mutants, C_1 ($n=1{,}112$, 58% viable) and (2) a set of 7,908 randomly generated mutants with two to ten mutations (10% viable). Note that the randomly generated mutants are fully disjoint from the validation set discussed in the ML model training (Methods). While many of the 7,908 randomly generated mutants were nonviable, these negative examples still provided valuable information about the sequence space to aid in ML modeling during training.

The $R_{10}+A_{39}$ dataset ($n=64{,}280$, 56% viable) contains (1) A_{39}, the 56,372 mutants generated by the baseline additive single-site fitness model described in Methods (62% viable) and (2) R_{10}, the same 7,908 sequences (with two to ten randomly generated mutants) as the C_1+R_{10} dataset (10% viable).

Note that the $R_{10}+A_{39}$ dataset does not contain the 1,112 single-site mutants (C_1). Comparison between C_1+R_{10} and $R_{10}+A_{39}$ explicitly tests the effect of training on a dataset that explicitly includes all single-mutant variants, versus a dataset that includes a large number of higher-order variants designed using single-variant data and tested in an additional round of data collection experiments.

Across all three libraries of training data, for each sequence variant we required a plasmid count >100 to provide some insulation from noisy mutant fitness measurements caused by low plasmid counts for specific variants. The resulting dataset contained at least four synonymous nucleotide sequences for each amino acid sequence variant present, and for each unique amino acid sequence present we took the highest observed fitness measurement across the synonymous nucleotide sequences that each had a plasmid count >100.

ML model experimental design overview. We use all three datasets to train classification models that predict whether a distant variant of the AAV2 capsid sequence is functional, as illustrated in Fig. 1b. To provide a baseline approach in which interactions between different mutations are not captured by the learned model, we trained logistic regression models. Although these models are restrained by their inability to capture higher-order interactions, they have the advantage that the number of free parameters is comparatively small, potentially avoiding the issue of overfitting that might temper the predictive ability of more complex models, in particular when trained using the smallest of our three training libraries. In addition, we also trained both convolutional and recurrent NN models using each of the three datasets. The CNN architecture was selected to assess the value of providing contiguous windows of local mutations as raw feature inputs to a deep NN, allowing it to assemble small local windows into larger aggregated receptive fields at deeper layers of the model. The RNN architecture was selected to assess the utility of having a stateful deep NN with aggregated knowledge of the mutations-incorporated N-terminal to a given mutation; specifically, a unidirectional, multilayered long short-term memory (LSTM) architecture was used. For all model architectures we used a simple one-hot representation of the input sequence data and supervised the model using binary labels for packaging, derived from the experimental measurements after taking into account the experimental noise present in the assay (Supplementary Fig 1).

Training ML models. Our cross-product of three model architectures (LR, CNN and RNN) and three distinct training datasets (C_1+R_2, C_1+R_{10} and $R_{10}+A_{39}$) resulted in nine categories of trained ML model (LR(C_1+R_2), LR(C_1+R_{10}) and RNN($R_{10}+A_{39}$)). Within each (architecture, dataset) category we trained 11 replica models, using distinct random initializations, to yield an ensemble model. Replica performance, specifically classification precision as a function of distance from WT, on a held-out random mutant validation set was used for termination of training via early stopping for each replica. To evaluate a given sequence, the mean model replica score of the ensemble was used; these 11-replica mean model scores were used for ranking sequences generated by both the model-based selection and model-guided design approach.

The CNN and RNN were optimized using the Adam algorithm (with learning rates of 0.001 and 0.010, respectively), while the TensorFlow logistic regression implementation used the FTRL algorithm with a learning rate of 0.01. The learning rates were selected via a hyperparameter sweep—selecting the learning rate with the best validation performance using the C_1+R_{10} training set for each model. All models were trained using a binary softmax cross entropy loss.

Models were regularized via early stopping using the implementation provided within 'train_utils.EarlyStopper'. Model training progression was monitored using the hold-out validation set of sequences that was identical for every architecture. Early stopping halted training after the model's precision on the validation set did not increase for ten evaluation periods. An evaluation period occurs every 500 steps in our set-up, with a batch size of 25 examples per step. The mean and maximum wall times were 20.3 and 85.3 min, respectively.

Architecture selection and hyperparameter tuning for NN models. The complexity of the architectures tested varied from the simplest LR model, with only 1,161 parameters and no hidden layers, to the CNN model with 55,189 parameters and four hidden layers (ConvPool, ConvPool, FC, FC) and, finally, to the most complex RNN (LSTM) model with 128,901 parameters and two hidden layers (FC, FC). All hyperparameter tuning was done while training on the fully random C_1+R_{10} dataset ($n=9{,}020$) and evaluating against a single validation set comprised of randomly sampled 2–10× mutant sequences ($n=1{,}977$); the validation set was also used for early stopping of training. Note that this validation set is fully disjoint from the random 2–10× mutant set incorporated into the R_{10} dataset.

The CNN model uses 55,189 parameters and four hidden layers:

- Input shape: (58, 20)
- Conv1d-relu-BN< Width=7, depth=12>
- Pool1d<width=2, stride=2>
- Conv1d-relu-BN< Width=7, depth=24>
- Pool1d<width=2, stride=2>
- FC1-relu-BN
- FC2-relu-BN

The RNN model, a multilayer LSTM, uses 128,901 parameters with two hidden layers, each having 100 units:

- Input token shape: (20)
- LSTM cell layer 1 (100 units)
- LSTM cell layer 2 (100 units)

Retrospective model validation. Before using the trained ML model ensembles to propose new diverse sequence variants predicted to be viable, we used the baseline additive set of 56,372 multi-mutant variants (A_{39}) as a held-out test set with which to compare the models trained using either the C_1+R_2 or C_1+R_{10} dataset, both of which excluded the A_{39} set of sequences. We were surprised to find that while all models exhibited a degree of lift in their ability to accurately predict viable mutants far from WT, compared to the additive model used to select the baseline set, the performance of the NN models trained using the threefold larger C_1+R_{10} dataset ($n=9{,}020$) was comparable to that obtained using the smallest C_1+R_2 dataset ($n=2{,}868$), which included only single and double mutants. We note that while the R_2 and R_{10} datasets are randomly generated, they contain multiple examples of sequence variants for which the measured phenotype does not reflect an additive model given the C_1 data. These cases are more difficult for the LR model to fit, providing a potential explanation for this performance difference compared to the NN models.

Generation of sequences via ML model-based selection. The pool of AAV2 mutant sequences from which ML models were allowed to rank and select was created by randomly sampling sequences 100 million times for each mutation distance between five and 25, inclusive, thereby generating a total of 2.1 billion candidate sequences as shown in Fig. 1c. To randomly sample a sequence at distance n from WT, n indices between 0 and 58 (2× the number of positions in tile 21) were drawn at random (note, prefix insertions were not permitted). For each index a random non-WT amino acid residue was selected (for indices corresponding to insertion positions, any residue was permitted). Each of the ML models compared in this work was then used to evaluate the entire pool of 2.1 billion sequences, selecting the top 1,000 at each mutation distance. These top 1,000 sequences at each mutation distance were then used as 'seed' sequences for the model-guided sequence design process described below. The top 100 of the 1,000 seed sequences at each distance from WT were also tested empirically for viability (before any model-guided sequence design); the performance of these top 100 sets is shown in Fig. 2 for each model, and these are referred to as the model-based selection set throughout this work.

Generation of sequences via ML model-guided design. To go beyond model-based selection, we developed a model-guided sequence design strategy to follow model gradients and find sequences with higher scores. Our previous experiment suggests that roughly 3,000 of the 100 million random candidates 15 steps from WT will be viable, diminishing to just 100 for those 20 steps from WT. This is supported by a marked decrease in model confidence for sequences that are far from WT. We next asked whether the trained models' internal representation of the AAV2 fitness landscape could be used to 'evolve' seed sequences in promising directions by exploring local neighborhoods around randomly sampled candidate sequences predicted by the model to be viable. The 1,000 highest-scoring sequences at each distance from WT were selected by ML models (specifically in each case by an ensemble of 11 replica models with the same architecture, trained on the same training data) as seed sequences (with the top 100 at each distance experimentally evaluated).

Starting from these model-selected seed sequence variants, we performed an iterative mutation process. First, a random set of 250 single-mutation steps (disallowing movements towards WT) were scored using the model ensemble mean probability (Training ML models). The 50 highest-scoring candidates were passed forward for the next iteration of mutation. We terminated this process after 20 iterations because the resulting variants exceeded the most distant viable sequence variant discovered by the additive baseline strategy (21 steps from WT). After 20 iterations had been completed, the total set of evaluated sequences across all mutation levels was ranked by consensus score. This model-ranked set was greedily filtered for diversity, permitting the addition of a candidate sequence if it was at least three mutations away from a higher-scoring sequence already included in the set. These candidate sets were aggregated across all model-selected seed sequences, selecting the top 900 sequences at each distance between five and 25 and the top 500 sequences at distances 26–29. Most model-designed, viable sequences originated from viable seeds (Supplementary Fig 5), suggesting that viable sequences even further from WT may be discovered by increasing the number of iterative mutation rounds.

Prospective model validation. To truly test the hypothesis that a small amount of double-mutant data is sufficient to improve the models over the additive model trained using all single-site variants, we experimentally validated sequences proposed using multiple training strategies. This framework has the advantage that it also allows comparison of the randomly chosen training libraries with the much larger set of sequences designed using the simple additive model, as an alternative, information-rich training dataset.

The model-selected and model-designed sequences were labeled as either viable or nonviable by calculating the ratio of mutant read counts in the viral library over counts in the plasmid DNA library, and comparison to the ratio calculated for the WT sequence. To confirm that our results were robust to noise resulting from small absolute counts, we excluded from the reported results sequence variants with fewer than ten plasmid counts. Note that this threshold is more permissive than that used for the training data, reflecting our desire to avoid training the models on data with noisy labels. Furthermore, we confirmed that the reported trends—in terms of the performance as a function of distance from the WT sequence, and the diversity of model-designed sequences—were maintained if we imposed more stringent criteria. We first restricted to those sequences for which the viral counts from each of the three biological replicates agree that the sequence is either viable or nonviable. This dataset yielded 100,929 viable sequences out of 172,664 model-designed sequences that met this criterion. In a second analysis, we removed the 2,055 viable sequences with <50 viral counts and verified that the reported trends still held using this slightly smaller set of viable sequences.

A surprising outcome of this experiment was that although the $R_{10}+A_{39}$ dataset contained 5× and 25× more sequences than the C_1+R_{10} and C_1+R_2 dataset, respectively, this abundance of training data did not always improve its ability to accurately identify viable sequences. In particular, the LR and RNN models trained using the $R_{10}+A_{39}$ dataset were outperformed by their respective variants trained using the smaller C_1+R_{10} dataset, and in the RNN case also by the C_1+R_2 dataset. Only in the case of the CNN did the larger $R_{10}+A_{39}$ training set result in a substantial improvement in performance, in particular in the ability to identify sequences further away from WT.

As a post hoc observation, across all models we empirically observed a decline in performance above 18 steps from WT. Since Fig. 2a shows that the model-selected seeds become notably less likely to be viable around eight to 12 steps from WT, this decline is at least partly explained by the choice of 20 maximum model-design iterations mentioned above.

Sequence clustering. To cluster each set of model-designed sequences, we first sorted them in descending order by number of mutations from WT AAV2 (that is, farthest first). For a given cluster radius, R, we start with the first sequence in the list and use that as a founder, then build a cluster around it by including every sequence $<R$ edit distance from the founder sequence (we compute the edit distance using the method described in https://pypi.org/project/python-Levenshtein/). We then repeat this process with the next remaining farthest-from-WT sequence in the yet-to-be-clustered set, and so forth, until all sequences have been placed into clusters.

Each set of designed sequences was of a different cardinality (for example, 2.5× additive sequences versus ML-designed sequences) and, to make the sets comparable as presented in Fig. 4a,b, we downsampled all sequence sets uniformly at random to the smallest common size: 19,680 sequences. We then clustered the viable subset of these partitions as shown in Fig. 4, which quantifies the volume of sequence space successfully covered for various clustering radii. To provide an additional perspective, we separately clustered all downsampled sequences (viable + nonviable) for the statistics presented in Fig. 4b, which quantifies the volume of sequence space covered versus capsid design success rates (percentage viable) at a fixed cluster radius of 12 (mean AAV serotype diversity μ).

AAV2 homolog selection and alignment construction. To compare our diversification approach with natural diversity, we investigated the available sequences on NCBI for dependoparvoviruses. We found 415 complete coding sequence records (~1,000 gene products) for dependoparvoviruses (txid 10803) containing a structural or VP protein. These data were parsed to extract structural and VP1 proteins. Records lacking a complete structural or VP protein were discarded. We also ensured that we had included all common AAV VP1 sequences (12 sequences). This processing resulted in 310 unique sequences, which we aligned using Clustal Omega 1.2.4 (ref. [35]). We then extracted the corresponding sequence to the AAV2 VP1 region of interest for each record to compute the statistics for natural diversity. We found that, for the 28-amino acid region of interest, dependoviruses show slightly less diversity than the 12 common serotypes (that is, many sequences are quite similar to each other, and to AAV2). Therefore, for comparison we used the 12 common serotypes as a stricter benchmark. All logo plots were generated using Skylign[31].

Statistical methods. In Supplementary Fig. 2a we provide estimates of the Pearson correlation between replicates of the plasmid and virus libraries for the experimental tests of the prospective ML validation libraries. In each case, the Pearson correlation was calculated over 243,481 replicate sequence variants. In Supplementary Fig. 2b we provide an estimate of the Pearson correlation between the experiment in which the ML training data were generated and the ML validation experiment for $n=2{,}000$ sequence variants, together with a P value that reflects the likelihood that this correlation would be seen by chance. The P value is calculated using a two-sided t-test with $n-2$ degrees of freedom. In Supplementary Fig. 3 the box plots for each model type are derived from the area under the receiver operating characteristics computed for each of the 11 individual models.

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

Data availability

Experimental data for all three experiments are available at NCBI SRA under accession code PRJNA673640).

Code availability

The TensorFlow 1.3 API was used to implement and train all models using the architectures described in Methods. The training and validation datasets used to create each model are available as part of the experimental dataset released as described in Data availability. The code required to construct the A_{39} training data, and also to synthesize, process and analyze the experimental data, is provided for download (https://github.com/churchlab/Deep_diversification_AAV), as well as the ipython notebooks that reproduced the analysis figures from the main text (https://github.com/google-research/google-research/tree/master/aav).

References

32. Pereira, F. et al. Pydna: a simulation and documentation tool for DNA assembly strategies using python. *BMC Bioinformatics* **16**, 142 (2015).
33. Zolotukhin, S. et al. Recombinant adeno-associated virus purification using novel methods improves infectious titer and yield. *Gene Ther.* **6**, 973–985 (1999).
34. Zhang, J., Kobert, K., Flouri, T. & Stamatakis, A. PEAR: a fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics* **30**, 614–620 (2014).
35. Sievers, F. et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.* **7**, 539 (2011).

Acknowledgements

The authors thank K. Kohlhoff, S. Kearnes, D. Belanger, E. Bixby and J. Gerold for helpful discussions. The authors thank the Wyss Institute for funding. L.J.C. gratefully acknowledges support from the Simons Foundation.

Author contributions

E.D.K., L.J.C., A.B., G.M.C. and P.F.R. conceived the study. E.D.K., N.K.J. and P.J.O. performed in vitro experiments. D.H.B., A.B. and L.J.C. designed, implemented and used ML models to generate variants, with input from E.D.K. and S.S. D.H.B., S.S., A.B., L.J.C. and E.D.K. analyzed the data. D.H.B., S.S., A.B., L.J.C. and E.D.K. wrote the paper, with input from all authors. A.B., P.F.R., G.M.C., L.J.C. and E.D.K. supervised the project and secured funding.

Competing interests

E.D.K., P.J.O., N.K.J., S.S. and G.M.C. performed research while at Harvard University, and E.D.K. and S.S. also performed research while at Dyno Therapeutics. E.D.K., S.S. and G.M.C. hold equity at Dyno Therapeutics. A full list of G.M.C.'s tech transfer, advisory roles and funding sources can be found on the website: http://arep.med.harvard.edu/gmc/tech.html. Harvard University has filed a provisional patent application for inventions related to this work. D.H.B., A.B., L.J.C. and P.F.R. performed research as part of their employment at Google LLC. Google is a technology company that sells ML services as part of its business.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41587-020-00793-4.

Correspondence and requests for materials should be addressed to G.M.C., L.J.C. or E.D.K.

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

nature research

Corresponding author(s): Lucy Colwell, Eric Kelsic

Last updated by author(s): Nov 16, 2020

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 Authors & Referees 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	Relevant excerpt from Methods: "PCR amplicons were sequenced with overlapping paired end reads using an Illumina Nextseq. Paired-end reads were merged to generate a consensus read using PEAR [35]"
Data analysis	Our manuscript uses the open source sklearn 0.20.4 and TensorFlow 1.3 APIs (including all algorithms e.g. FTRL contained therein) together with previously published neural network and logistic regression model architectures trained via the standard optimization protocols described in 'Methods'. The code required to construct the A39 training data and also to synthesize, process, and analyze the experimental data is provided at https://github.com/churchlab/Deep_diversification_AAV. The code used to derive the results described in the manuscript is provided as python notebooks hosted at https://github.com/google-research/google-research/tree/master/aav. The data and ~300k AAV capsid sequences described within the manuscript will be provided alongside the code for easier reproducibility. Additionally the data is hosted on the NCBI SRA as mentioned below. To align natural AAV sequences we used Clustal Omega 1.2.4

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/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

Experimental data for all 3 experiments is available on the public repository NCBI SRA (https://www.ncbi.nlm.nih.gov/sra), with accession id: PRJNA673640

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	To evaluate the predictive power of the machine learning models we tested thousands of model-selected and model-designed sequence variants. We also tested thousands of sequence variants that were chosen at random, and that were designed using a baseline approach.
Data exclusions	No data was excluded from analysis.
Replication	We confirmed that the experimental results were reproducible by replicating measurements for 2000 sequence variants with a range of selection scores from the ML training data set in the ML validation experiment. Results are reported in Fig. S2B. We also carried out PCR and transfection replicates as described in 'Methods' to confirm the reproducibility within the ML validation experiment. Results shown in Fig. S2a.
Randomization	Randomization was not used because there was no part of this study for which randomization would have been appropriate.
Blinding	Blinding was not used because there was no part of this study for which blinding would have been appropriate.

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
☐	☒ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data

Methods

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

Eukaryotic cell lines

Policy information about cell lines

Cell line source(s)	ATCC CRL-3216
Authentication	No methods were used for cell line authenticaion
Mycoplasma contamination	Separate from the AAV production experiment, aliquots of the HEK293T cells were confirmed negative for mycoplasma using the Lonza Mycoalert Plus kit.
Commonly misidentified lines (See ICLAC register)	*Name any commonly misidentified cell lines used in the study and provide a rationale for their use.*

Animals and other organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research

Laboratory animals	8 weeks old, male, C57BL/6J
Wild animals	n/a
Field-collected samples	n/a

Ethics oversight	Animal protocol was reviewed and approved by HMS IACUC

Note that full information on the approval of the study protocol must also be provided in the manuscript.