Massively parallel deep diversification of AAV capsid proteins by machine learning

Drew Bryant*{1}, Ali Bashir*{1}, Sam Sinai*{2,3,4,5}, Nina Jain{2,3}, Pierce Ogden{2,3}, Patrick Riley{1}, George Church^{2,3}, Lucy Colwell^{1,6}, Eric Kelsic^{2,3,4}

*Equal contribution
^Corresponding
Affiliations:

1. Google Research
2. Wyss Institute for Biologically Inspired Engineering, Boston, MA
3. Dept. of Genetics, Harvard Medical School, Boston, MA
4. Dyno Therapeutics, Cambridge, MA
5. Dept. of Organismic and Evolutionary Biology, Harvard University, Cambridge, MA
6. Dept. of Chemistry, University of Cambridge, Cambridge, UK

Email for all authors:
Drew: bryantd@google.com
Ali: bashira@google.com
Sam: sam.sinai@dynotx.com
Nina: ninajain@g.harvard.edu
Pierce: pierce.ogden@gmail.com
Patrick: pfr@google.com
George: gchurch@genetics.med.harvard.edu
Lucy: lcolwell@google.com
Eric: eric.kelsic@dynotx.com

Nature provides abundant examples of protein families with highly diverged sequences. The ability to design new protein homologs has many applications, yet synthetic approaches have been unable to generate similarly diverse protein sequences with functional activity in the lab [1, 2]. New technologies offer a solution: high-throughput DNA synthesis and sequencing technologies allow thousands of designed sequences to be assayed in parallel, enabling deep diversification guided by machine learning (ML) models that relate protein sequence to function without detailed biophysical or mechanistic modeling. Here we apply deep learning to design novel adeno-associated virus (AAV) capsid proteins, a challenging target of great utility for gene therapy. Focusing on a 28-amino acid segment spanning buried and exposed regions, we generated 201,426 highly diverse variants of the AAV2 wildtype (WT) sequence, yielding 110,689 viable synthetic 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 predicted capsid viability across highly diverse variants. Deep diversification enables the design of AAV capsids with completely synthetic sequences for the universal treatment of all patients regardless of prior exposure to natural AAV, while demonstrating a general approach that makes vast areas of functional but previously unreachable sequence space accessible.

Engineering 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 [3-5], 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 [6-8]. Directed evolution is a powerful approach [9-11], 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 [12, 13]. Recent high-throughput DNA sequencing-based assays allow large-scale mapping of fitness landscapes [14-16], while advances in DNA synthesis and ML technologies enable a completely data-driven workflow for accelerated directed evolution [2, 17-23]. 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 multi-protein 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).

AAV capsids hold tremendous promise as gene delivery vectors. The AAV2 capsid is a component of the first gene therapy to receive approval for sale by the U.S. Food and Drug Administration for use in humans [24, 25], while other serotypes are in clinical trials [26]. A major challenge is that immunity due to prior AAV exposure excludes 20-80% of the population from systemically administered therapies employing natural capsids [27]. Novel and diverse AAV vectors that retain the ability to package DNA payloads and transduce cells while evading the humoral immune system are urgently required [28]. Previous engineering strategies, such as targeted random peptide insertions, error-prone mutagenesis [13], random shuffling between AAV serotypes to create chimeric capsids [12], and random mutation at structurally-guided positions [29], have had limited success at 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 [30], indicating that capsids capable of avoiding neutralizing serum will require changes to many positions, most likely approaching or exceeding the diversity of natural serotypes (i.e. on the order of hundreds of sequence differences). To evaluate the utility of a purely data-driven approach to diversification, we directly generated synthetic sequences near the 3-fold symmetry axis of the icosahedral AAV2 capsid protein. Specifically, we targeted positions 561-588, a region that encompasses buried, surface and interface regions, and overlaps known heparin-binding as well as antibody binding sites [30].

Capsid production represents a bottleneck in the creation of diverse AAV capsids as the majority of sequence variants fail to assemble or to package their genome [21, 29, 31]. To generate large and diverse datasets for training machine learning 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 (i.e. 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 just 10% viable, with only 0.3% viability for variants with >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) to design 56,372 variants with between 2 and 39 mutations in the target region, with the goal of testing the limits of exploration made possible given our prior 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 (i) training set design and (ii) ML model architecture. We compared three ML training datasets designed via Complete (C), Random (R) or Additive (A) sampling strategies, splitting data from the prior experiment into three sets that vary in the number of sequence variants and their distribution and distance from WT. These splits enable assessment of how training data structure affected 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 2-10 mutations, while the R_{10}+A_{39} dataset contains R_{10} plus the 56,372 additive model-designed sequence variants with 2-39 mutations described above. A fixed set of 1,977 randomly chosen sequence variants with 2-10 mutations was held out for hyper-parameter 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 or not (Supplementary Fig 1), as defined by a threshold fit to best separate positive and negative controls (WT replicas and variants containing stop codons respectively).

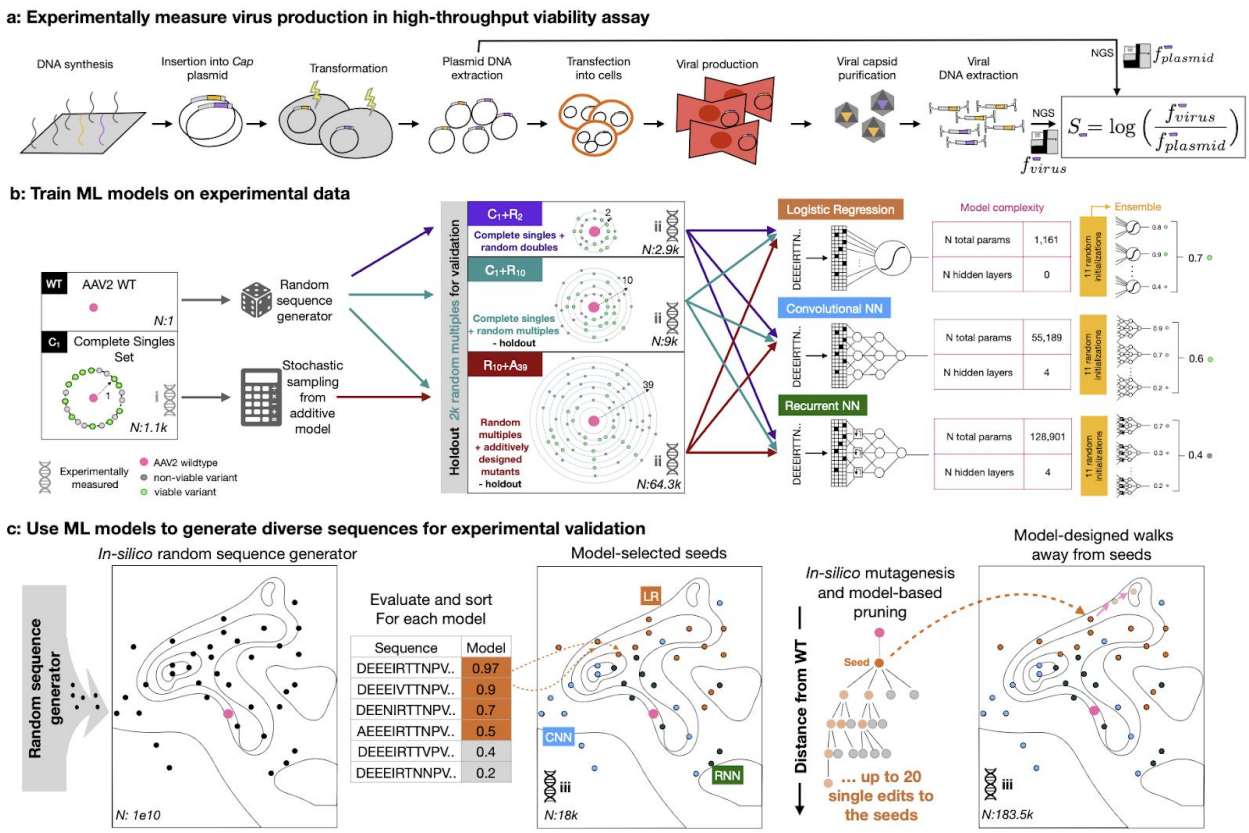


Figure 1 | Generation of diverse sequence variants guided by machine learning models trained on deep mutational libraries. a, *Experimental workflow:* Multiplexed measurement of viability for AAV capsid production. Three experiments (helix marker) were conducted to generate production data for: (i) all single mutants, (ii) ML training data, and (iii) ML validation data. **b**, *ML model training workflow*: Experimental data from mutants generated by complete (C), Random (R) or Additive model (A) design

strategies were assembled into three training data sets: C_1+R_2, C_1+R_{10}, R_{10}+A_{39}. Subscripts indicate the maximum number of mutations relative to WT. Each data set was used to train three machine learning models with varying architectures and increasing numbers of parameters: Logistic regression (LR), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN). **c**, *Sequence design workflow*: Randomly generated candidates were ranked by model ensemble score to yield model-selected sequences. Top candidates were subject to 20 iterative design cycles to obtain model-designed sequences.

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 the 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 5-29 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 re-tested 2000 sequences from the training set as controls, demonstrating strong experimental reproducibility (R=0.89, p<10^-20, Supplementary Fig 2).

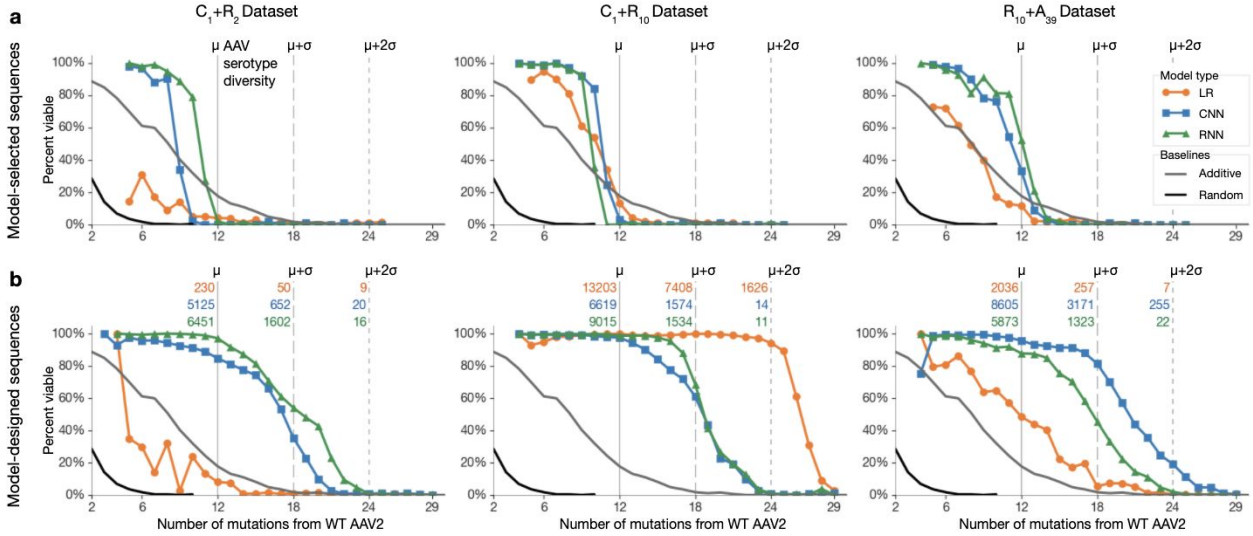


Figure 2 | Experimental validation of synthetic sequences demonstrates high performance and robustness of NN models to training data composition. a, Performance of model-selected sequences. On each plot, black is the randomly generated baseline (N=10,997), gray is the additive baseline (N=56,372). Vertical lines show the number of mutations within natural AAV serotypes in the target region on average (μ=12), plus additional standard deviations (σ=6). **b,** Performance of model-designed sequences. Colored numbers are viable capsids with at least the indicated number of mutations. Aggregated statistics 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 μ=12±6 positions. Model-selected sequences from CNN and RNN models had close to 100% viability at 6 mutations from WT (Fig 2a), the threshold at which randomly chosen sequence variants were largely non-viable. However model-selected viability dropped quickly beyond 12 mutations from WT, most likely because the randomly generated candidate sequences that the models had to choose from were overwhelmingly non-viable. 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 NN model-designed sequences were 33 times more likely to be viable than sequences designed by the additive model at 18 mutations (μ+σ) from WT, with even greater improvements at larger distances.

The performance of neural network 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 (μ+2σ) 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 (μ+σ) from WT, comparable to those trained on the ~3x larger C_1+R_{10} and ~22x 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 observe that more training data does not guarantee better model performance. To better understand this observation, we turned to analyze the diversity of designed variants.

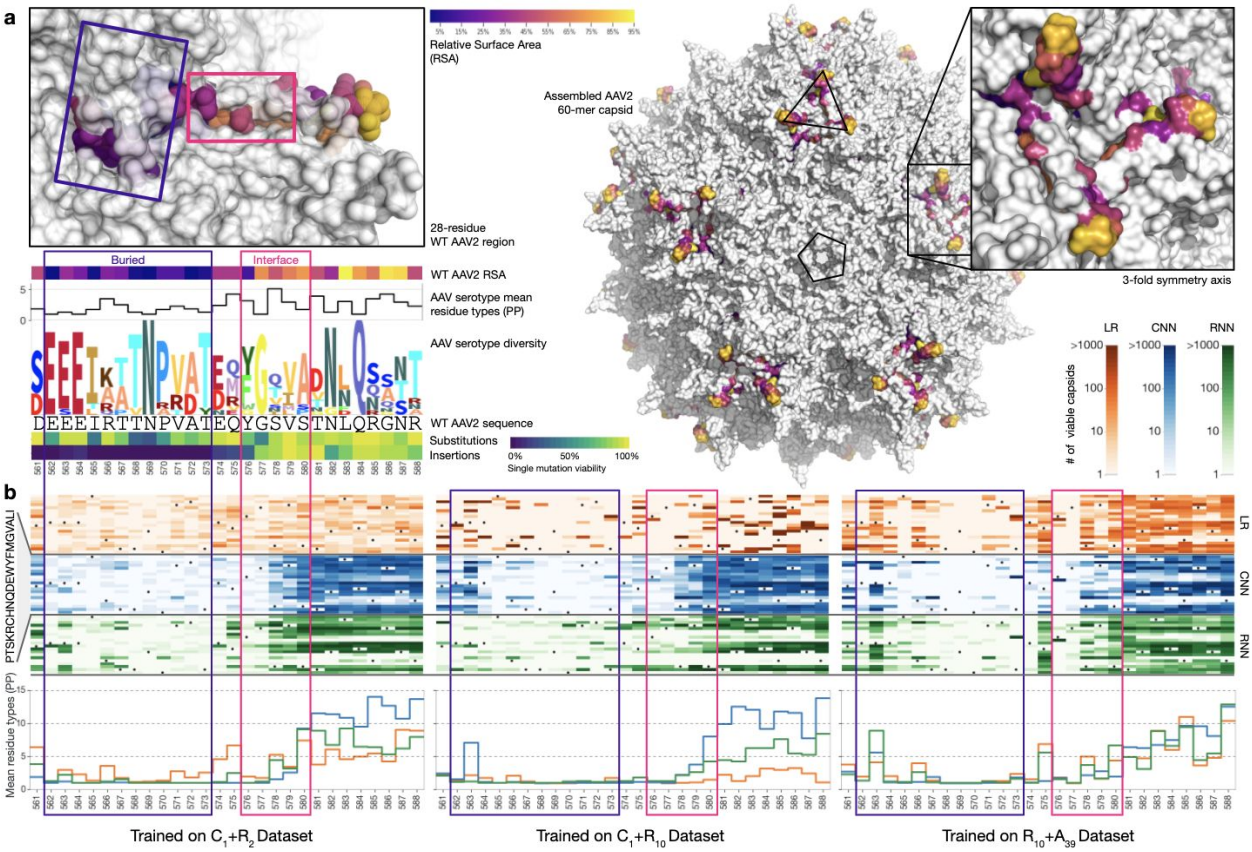


Figure 3 | Neural network models generated greater diversity across positions. a, 3D structure of the 28-residue region with boxed buried (purple) and interface regions (pink) colored by residue surface accessibility (RSA) for a single monomer, shown in context with interfacing monomers. Average tolerance to single substitutions and insertions (measured experimentally) shown for each position along with the perplexity and natural diversity across 12 common serotypes (logo plots made with [32]). **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 (PP=perplexity).

Models differed in the levels of sequence diversity that they generated. The first 2/3 of the target region is more conserved across natural AAV sequences, likely because these positions are less surface exposed, and 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, 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 in Fig 3b), RNN models exhibited preference for substituting amino-acids with similar chemical properties, while the 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) [33]. 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 tradeoff: picking only the highest scoring sequence may be precise, but has no diversity, whereas randomly generated sequences have high diversity but low precision. Of course, models can also have low diversity and low precision (e.g. in LR{C_1+R_2}).

To quantitatively evaluate model performance in this respect, we partitioned all designed sequences into clusters of radii 12 edits (μ), and computed the average viability within the resulting clusters. NN models outperformed LR models at all viability thresholds. The RNN performed best for the smallest C_1+R_2 dataset, while the CNN performed better for the larger datasets (Fig 4b). Projecting viable sequences from the C_1+R_{10} models into 2D with ivis [34] 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, though we note that all models discovered viable sequence variants that are highly distinct from natural AAV serotypes (Fig 4C, 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 data sets, although better strategies are certainly possible and additional work will determine how these findings generalize to other contexts.

The success of these diversification strategies (i) addresses the immediate need for synthetic AAV capsids with sequences distinct from natural isolates, and (ii) 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 <3000 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 significantly more challenging for traditional methods of directed evolution.

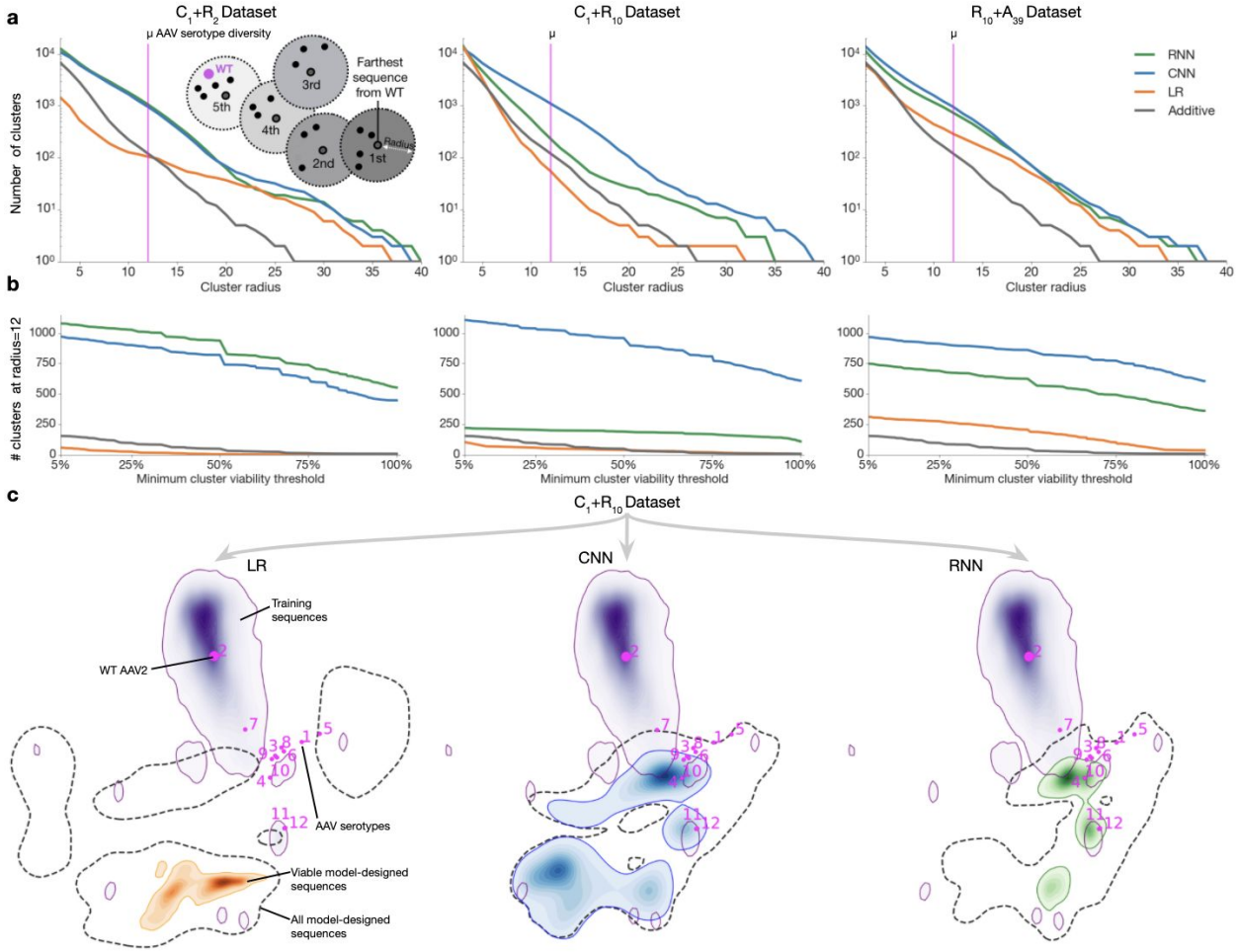


Figure 4 | Neural networks generated greater functional diversity at equivalent levels of performance relative to additive and LR models. Inset**:** method for sequence clustering. **a**, Number of distinct viable sequence clusters as a function of cluster radius. **b**, Number of clusters for which models predicted viable mutants at or above a minimum performance threshold. Cluster radius is at μ=12, the AAV natural serotype diversity within the target region. **c**, Visualization of sampled diversity through ivis projections [34]. 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/RNN models, respectively. Magenta: natural AAV serotypes (1-12) embedded for reference.

While many machine learning studies are conducted on a single standardized dataset where only

differences in 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 wildtype suggests that similar approaches can be used for proteins in which high-throughput screens are impractical. Importantly, after such models have been trained, generating new sequences requires only additional compute 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.

References

1. Trudeau, D.L., Smith, M.A. and Arnold, F.H., 2013. Innovation by homologous recombination. *Current opinion in chemical biology*, 17(6), pp.902-909.
2. Yang, K.K., Wu, Z. and Arnold, F.H., 2019. Machine-learning-guided directed evolution for protein engineering. *Nature methods*, 16(8), pp.687-694.
3. Huang, P.S., et al., 2014. High thermodynamic stability of parametrically designed helical bundles. *Science*, 346(6208), pp.481-485.
4. Butterfield, G.L., et al., 2017. Evolution of a designed protein assembly encapsulating its own RNA genome. *Nature*, 552(7685), pp.415-420.
5. Langan, R.A., et al., 2019. De novo design of bioactive protein switches. *Nature*, 572(7768), pp.205-210.
6. Weinreich, D.M., Delaney, N.F., DePristo, M.A. and Hartl, D.L., 2006. Darwinian evolution can follow only very few mutational paths to fitter proteins. *Science*, 312(5770), pp.111-114.
7. Halabi, N., Rivoire, O., Leibler, S. and Ranganathan, R., 2009. Protein sectors: evolutionary units of three-dimensional structure. *Cell*, 138(4), pp.774-786.
8. Ferretti, L., Weinreich, D., Tajima, F. and Achaz, G., 2018. Evolutionary constraints in fitness landscapes. *Heredity*, *121*(5), pp.466-481.
9. Stemmer, W.P., 1994. Rapid evolution of a protein in vitro by DNA shuffling. *Nature*, *370*(6488), pp.389-391.
10. Fox, R.J., et al., 2007. Improving catalytic function by ProSAR-driven enzyme evolution. *Nature biotechnology*, *25*(3), pp.338-344.
11. Davis, A.M., Plowright, A.T. and Valeur, E., 2017. Directing evolution: the next revolution in drug discovery?. *Nature Reviews Drug Discovery*, *16*(10), p.681.
12. Grimm, D., et al., 2008. In vitro and in vivo gene therapy vector evolution via multispecies interbreeding and retargeting of adeno-associated viruses. *Journal of virology*, *82*(12), pp.5887-5911.
13. Dalkara, D., et al., 2013. In vivo–directed evolution of a new adeno-associated virus for therapeutic outer retinal gene delivery from the vitreous. *Science translational medicine*, *5*(189), pp.189ra76-189ra76.
14. Araya, C.L., et al., 2012. A fundamental protein property, thermodynamic stability, revealed solely from large-scale measurements of protein function. *Proceedings of the National Academy of Sciences*, *109*(42), pp.16858-16863.
15. Sarkisyan, K.S., et al., 2016. Local fitness landscape of the green fluorescent protein. *Nature*, *533*(7603), pp.397-401.

16. Poelwijk, F.J., Socolich, M. and Ranganathan, R., 2019. Learning the pattern of epistasis linking genotype and phenotype in a protein. *Nature communications*, *10*(1), pp.1-11.
17. Romero, P.A., Krause, A. and Arnold, F.H., 2013. Navigating the protein fitness landscape with Gaussian processes. *Proceedings of the National Academy of Sciences*, *110*(3), pp.E193-E201.
18. Wu, Z., Kan, S.J., Lewis, R.D., Wittmann, B.J. and Arnold, F.H., 2019. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences*, *116*(18), pp.8852-8858.
19. Alley, E.C., Khimulya, G., Biswas, S., AlQuraishi, M. and Church, G.M., 2019. Unified rational protein engineering with sequence-based deep representation learning. Nature methods, 16(12), pp.1315-1322.
20. Kelsic, E.D. and Church, G.M., 2019. Challenges and opportunities of machine-guided capsid engineering for gene therapy. *Cell Gene Ther. Insights*, *5*, pp.523-536.
21. Ogden, P.J., Kelsic, E.D., Sinai, S. and Church, G.M., 2019. Comprehensive AAV capsid fitness landscape reveals a viral gene and enables machine-guided design. *Science*, *366*(6469), pp.1139-1143.
22. Liu, G., et al., 2019. Antibody Complementarity Determining Region Design Using High-Capacity Machine Learning. *bioRxiv*, p.682880.
23. Brookes, D.H., Park, H. and Listgarten, J., 2019. Conditioning by adaptive sampling for robust design. *arXiv preprint arXiv:1901.10060*.
24. Russell, S., et al., 2017. Efficacy and safety of voretigene neparvovec (AAV2-hRPE65v2) in patients with RPE65-mediated inherited retinal dystrophy: a randomised, controlled, open-label, phase 3 trial. *The Lancet*, *390*(10097), pp.849-860.
25. Dunbar, C.E., et al., 2018. Gene therapy comes of age. *Science* 359, eaan4672.
26. Mendell, J.R., et al., 2017. Single-dose gene-replacement therapy for spinal muscular atrophy. *New England Journal of Medicine*, *377*(18), pp.1713-1722.
27. Calcedo, R., Vandenberghe, L.H., Gao, G., Lin, J. and Wilson, J.M., 2009. Worldwide epidemiology of neutralizing antibodies to adeno-associated viruses. *The Journal of infectious diseases*, *199*(3), pp.381-390.
28. Kotterman, M.A. and Schaffer, D.V., 2014. Engineering adeno-associated viruses for clinical gene therapy. *Nature Reviews Genetics*, *15*(7), pp.445-451.
29. Tse, L.V., et al., 2017. Structure-guided iterative evolution of antigenically advanced AAV variants for therapeutic gene transfer. *Mol. Ther*, *25*(5S1), pp.232-232.
30. Tseng, Y.S. and Agbandje-McKenna, M., 2014. Mapping the AAV capsid host antibody response toward the development of second generation gene delivery vectors. *Frontiers in immunology*, *5*, p.9.
31. Adachi, K., Enoki, T., Kawano, Y., Veraz, M. and Nakai, H., 2014. Drawing a high-resolution functional map of adeno-associated virus capsid by massively parallel sequencing. *Nature communications*, *5*, p.3075.
32. Wheeler, T.J., Clements, J. and Finn, R.D., 2014. Skylign: a tool for creating informative, interactive logos representing sequence alignments and profile hidden Markov models. *BMC Bioinformatics*, *15*(1), p.7.
33. https://pypi.org/project/python-Levenshtein/
34. Szubert, B. and Drozdov, I., 2019. ivis: dimensionality reduction in very large datasets using Siamese Networks. *Journal of Open Source Software*, *4*(40), p.1596.

35. Pereira, Filipa, et al. "Pydna: a simulation and documentation tool for DNA assembly strategies using python." BMC bioinformatics 16.1 (2015): 142.
36. Zolotukhin, S., et al., 1999. Recombinant adeno-associated virus purification using novel methods improves infectious titer and yield. *Gene therapy*, 6(6), pp.973-985.
37. Zhang, J., Kobert, K., Flouri, T. and Stamatakis, A., 2014. PEAR: a fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics*, *30*(5), pp.614-620.
38. Thompson, J.D., Higgins, D.G. and Gibson, T.J., 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, *22*(22), pp.4673-4680.

Acknowledgements

The authors would like to thank Kai Kohlhoff, Steven Kearnes, David Belanger, Eli Bixby, and Jeffrey Gerold for helpful discussions. The authors thank the Wyss Institute for funding.

Authors contributions

EK, LJC, AB, GMC, PR conceived the study. EK, NJ, and PJO performed in-vitro experiments. DHB, AB, LJC designed, implemented and used ML models to generate variants with input from EK, SS. DHB, SS, AB, LJC, EK analyzed the data. DHB, SS, AB, LJC, EK wrote the paper with input from all authors. AB, PR, GMC, LJC, EK, supervised the project and secured funding.

Competing interests

EK, PJO, NJ, SS, GMC performed research while at Harvard University and EK, SS also performed research while at Dyno Therapeutics. EK, SS, and GMC hold equity at Dyno Therapeutics. A full list of GMC's tech transfer, advisory roles, and funding sources can be found on the lab's website: http://arep.med.harvard.edu/gmc/tech.html. Harvard University has filed a provisional patent application for inventions related to this work. DHB, AB, LJC, PR performed research as part of their employment at Google LLC. Google is a technology company that sells machine learning services as part of its business.

Data availability

Experimental data for all 3 experiments will be deposited on a public repository (NCBI SRA (https://www.ncbi.nlm.nih.gov/sra), id: SUB7629680) by publication date.

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 for creating each model are available as part of the experimental dataset released as described in the preceding section. The code required to construct the A_{39} training data and also to synthesize, process, and analyze the experimental data is provided for download, together with ipython notebooks that reproduce the analysis figures from the main text.

Methods

AAV mutant sequence library generation and production assay

Libraries were constructed using a method similar to that previously described in Ogden et al. [21]. For the final validation experiments, 184mer 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 choosing 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'), an 84+ nucleotide mutant coding sequence, 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 prior production experiments). PyDNA[35] 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.

An example oligo with WT coding region:
5'-GGGTCACGCGTAGGAGAAGACATTACAGACGAAGAGGAAATCAGGACAACCAATCCCGTGGCTA
CGGAGCAGTATGGTTCTGTATCTACCAACCTCCAGAGAGGCAACAGACAAGCGAGACCGATATCGG
TCTCACGCTGTAATGCGGTCTGAGCCGCGCTAAGTCTTCGTGTGGCTGCGGAAC-3'.

Cloning was carried out in three steps. First, oligos were PCR amplified using Q5 high-fidelity DNA polymerase (NEB M0492) and an annealing temperature of 60°C (fwd: 5'-GGGTCACGCGTAGGA-3', rev: 5'-GTTCCGCAGCCACAC-3'). A backbone plasmid containing the WT AAV2 *cap* gene was also amplified with Q5 and an annealing temperature of 72°C (fwd:
5'-TTGGTCTCA|CGCTAGAGACGGTGTGGCTGCGGAAC-3', rev:
5'-AAGGTCTCC|TGTAATCATGACCTTTTCAATGTCCACATTTG-3'). This PCR was used to add BsaI sites with overhangs complementary to the BbsI overhangs in the oligos (as indicated by cut sites in primer sequences). Amplified oligos were digested with BbsI-HF (NEB R3539) and amplified plasmid was digested with BsaIHF-v2 (NEB 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 x 10^6 U/mL, NEB M0202). Ligation products were ethanol precipitated and transformed into 50 uL of electrocompetent cells (Lucigen 10G SUPREME 60081). Following a 1 hour recovery at 37°C, cells were added to 4 mL of selection media (2x YT with kanamycin) and grown at 37°C overnight. The following morning, step one library plasmids were mini-prepped by alkaline lysis (Qiagen 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 (fwd:
5'-TTGGTCTCA|CAAGCAGCTACCGCAGATGTCA-3', rev:

5’-AAGGTCTCA|AGCGAGAGACGTCCTACGCGTGACCC-3’). This amplification step was also used to add BsaI sites complementary to those in the oligos. Step one 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 R3195) in a “kill cutting” step to remove step one plasmids that did not incorporate the 3’ WT amplicon. EcoRV digest products were ethanol precipitated, transformed, and mini-prepped as in step one. In the third cloning step, a destination ITR-containing plasmid was digested with HindIII-HF (NEB R3104) and SpeI-HF (NEB R3133). Complete mutant *cap* gene sequences were amplified from the step two plasmid library using Q5 and an annealing temperature of 70°C (fwd: 3’-AGGTCTCA|AGCTTCGATCAACTACGCAGACAG-5’, rev: 3’-AGGTCTCA|CTAGATGAGCTCGTCGACGTTCC-5’). 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 two 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 were 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 10566016) supplemented with 10% FBS (ThermoFisher 10082147) and seeded in 5-layer cell stacks (Corning 353144) two days prior to 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 minutes, and added to cells. Media was 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 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 hours. 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 431098). 40% PEG-8000 was then added to a final concentration of 8% and mixtures were incubated at 4°C for 3 hours. Samples were centrifuged at 3000 x *g* for 20 minutes 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 of benzonase (Millipore Sigma 1.01695.0001) was added to resuspended pellets and samples were incubated at 37°C for 45 minutes. Encapsidated genomes were separated from the remaining cellular debris using iodixanol ultracentrifugation and concentration via size exclusion spin filters as described previously [21, 36]. Briefly, benzonase-treated samples were underlaid with an iodixanol gradient (Sigma D1556) in polypropylene tubes (Beckman Coulter 362183) and centrifuged at 242,000 x *g* for 1 hour at 16°C. Capsids were collected from the 40% iodixanol fraction and concentrated using a spin concentrator (Millipore Sigma 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 minutes and PCR was run with Q5 and an annealing temperature of 65°C to amplify the mutant region of the *cap* gene (fwd:

5’-GCTCAGAGAAAACAAATGTGGAC-3’, rev: 5’-GAACGCCTTGTGTGTTGACATC-3’). PCR reactions were carried out in the presence of EvaGreen (Biotium 31000) and run on a BioRad CFX96 qPCR 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 an Illumina NextSeq. Paired-end reads were merged to generate a consensus read using PEAR [37], 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 2 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 A39 data set was done via amplicon sequencing from purified vector genomes, with PCR protocols as described above. Three separate batches of virus were prepared and 3.5e10vg (batch 1), 2.3e10vg (batch 2) and 3.5e10vg (batch 3) of the C_1 virus library was diluted in 200uL PBS and injected into a mouse, 4 mice per batch, for 12 mice in total. Mice were all 8 weeks old, male, and C57BL/6J. For each of the 3 batches, 2 mice were injected retro-orbitally and 2 intraperitoneally. 30uL blood was drawn after 1 hour, 5 hours and 24 hours by facial bleed from and frozen on dry ice, then at -80C. After 8 days, mice were dissected and tissue samples from liver, kidney, heart, lung, brain, spleen, muscle, skin, stomach, and testes were frozen on dry ice, then at -80C. Approximately 150 mg of each organ was ground using disposable mortar and pestle (Kimble Chase 749625-0010). DNA was purified from tissues using alkaline lysis (Qiagen 27104) and from blood using Qiagen MinElute Virus Spin Kit (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 4 mice from batch 2 for the remainder of organs.

Random sampling of AAV2 mutants around wildtype

To generate a sequence at mutation distance *k* steps from WT AAV2, first a uniform random draw from the set of {28 WT positions + 28 insertion positions} was 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 then 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 (7908 variants) + tune (1977 variants) random multi-mutant sequence set was generated by sampling 1732-1756 sequences at each distance of 2-6 steps away from WT, inclusive, and 288-290 sequences from 7-10 steps, inclusive. In total, the random multi-mutant sequence baseline experiment tested 9885 unique sequences between 2-10 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, testes, and blood samples was used to compute selection scores (the relative enrichment of variants in

the tissue vs. the original plasmid library) for each sample. This data all contained information about production ability, as viral production is a necessary requirement for viruses to be observed in each tissue, and was therefore a common contributor to variance across all models. We generated mutants for the A_{39} set using data from biodistribution data rather than simply production data so as 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: i) allowing substitutions across the region, ii) allowing substitutions and insertions (but no more than one amino-acid between two positions) and iii) allowing substitutions but restricting the same insertions to the second half of the tile.

To design variants from single mutants data with the additive model, we employed three flavors 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 was the tropism for amino-acid *i* in that position as measured in the singles library (for different tissues) and *Z* ensured that the sum of probabilities across the position equaled 1. The temperature parameter T, controlled 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). The parameter T was fixed during the generation of each variant. However to produce a diverse library we varied T between $10^{[-2, 0]}$ (with 0.18 increments 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 the t_s between [-1, 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 only accept the variant if the sum of effects from the individual mutations were above the threshold t_m. We varied t_m between [0, 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 introducing the mutations included in the variant one at a time. The order in which these mutations were introduced was either greedy (meaning better mutations introduced first) or at 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 that each contain different numbers of sequence variants that were sampled from a constrained interval of sequence space around the wildtype AAV2 sequence using three distinct sampling strategies. The additive dataset (A_{39}) provides a baseline training data set in which mutants were generated first by measuring the complete set of single mutants

and then generating diverse mutants using additive models (see section above). In contrast, the other two libraries (R_2 and R_{10}) exploit the power of random sampling to choose sequences with multiple mutations sampled uniformly at random from the sequence space around the wildtype 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: (i) the complete set of single site mutants, C_1 (N=1,112, 58% viable); and (ii) a <1% random subset of the 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; i.e., 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: (i) the complete set of single site mutants, C_1 (N=1,112, 58% viable); and (ii) a set of 7,908 randomly generated mutants with 2-10 mutations (10% viable). Note that the randomly generated mutants are fully disjoint from the validation set discussed in the ML model training Methods section. While many of the 7,908 randomly generated mutants are non-viable, these negative examples still provided valuable information about the sequence space to aid ML models during training.

The R_{10}+A_{39} dataset (N=64,280, 56% viable) contains: (i) A_{39}, the 56,372 mutants generated by the baseline additive single site fitness model described in Methods, (62% viable); and (ii) R_{10}, the same 7,908 sequences with 2-10 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). The comparison between C_1+R_{10} and R_{10}+A_{39} explicitly tests the effect of training on a dataset that explicitly includes all the single mutant variants, vs a dataset that includes a large number of higher order variants designed using the 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 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 smaller of our three training libraries. In addition, we also trained both convolutional and recurrent neural network 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, multi-layered 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 (see Supplementary Fig 1).

Training ML models

Our cross product of three model architectures (LR, CNN, RNN) and three distinct training datasets (C_1+R_2, C_1+R_{10}, R_{10}+A_{39}) resulted in nine categories of trained machine learning models (LR{C_1+R_2}, LR{C_1+R_{10}}, ..., 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 terminating 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 approaches.

The CNN and RNN were optimized using the Adam algorithm (with learning rates of 0.001 and 0.01, respectively) while the TensorFlow logistic regression implementation utilized 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 on 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 the same for every architecture. Early stopping halted training after the model's precision on the validation set did not increase for 10 evaluation periods. An evaluation period occurs every 500 steps in our setup, with a batch size of 25 examples per step. The mean and max wall-times were 20.3, and 85.3 minutes, respectively.

Architecture selection and hyperparameter tuning for neural network models

The complexity of the architectures tested varied from the simplest logistic regression (LR) model with only 1,161 params and 0 hidden layers, to the CNN model with 55,189 params and 4 hidden layers {ConvPool, ConvPool, FC, FC} and finally to the most complex RNN (LSTM) model with 128,901 params and 2 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 2-10x 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 2-10x mutant set incorporated into the R_{10} dataset.

The Convolutional Neural Network (CNN) model uses 55,189 params and 4 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 Recurrent Neural Network (RNN) model, a multilayer LSTM, uses 128,901 params with 2 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 3x larger C_1+R_{10} dataset (N=9,020) was comparable to that obtained using the smaller C_1+R_2 dataset (N=2,868), which only included 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.

Generating 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 5-25, 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 wildtype, *n* indices between [0, 58) (2 * the number of positions in tile21) were drawn at random (note, prefix insertions were not permitted). For each index a random non-wild-type 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 sequences 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 wildtype were also tested empirically for viability (before any model-guided sequence design); the performance of these top 100 sets are shown in Fig 2 for each model, and are referred to as the model-based selection set throughout this work.

Generating 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 3k of the 100 Million random candidates 15 steps from WT will be viable, diminishing to just 100 of 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 could utilize their internal representation of the AAV2 fitness landscape to “evolve” seed sequences in promising directions by exploring local neighborhoods around randomly sampled candidate sequences that the model predicts 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 (see Training ML Models). The highest-scoring 50 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 were 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, only permitting the addition of a candidate sequence if it was at least 3 mutations away from a higher scoring sequence already included in the set. These candidate sets were aggregated across all model-selected seed sequences, with the top 900 sequences at each distance between 5-25, and the top 500 sequences at distances 26-29 selected. As most model-designed, viable sequences originated from viable seeds (Supplementary Fig 5), viable sequences even further from wild-type may be possible 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 enough to significantly 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 the randomly chosen training libraries to be compared 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 labelled viable or non-viable by calculating the ratio of mutant read counts in the viral library over the counts in the plasmid DNA library, and comparing to this ratio calculated for the WT sequence. To confirm that our results were robust to noise due to small absolute counts, we excluded sequence variants with <10 plasmid counts from the reported results. 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 in addition 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 non-viable. This dataset yielded 100,929 viable sequences out of 172,664 model designed sequences that met this criteria. In a second analysis, we removed the 2,055 viable sequences that had <50 viral counts, and verified that the reported trends still hold using this slightly smaller set of viable sequences.

A surprising outcome of this experiment was that although the R_{10}+A_{39} dataset contained 5x and 25x more sequences than the C_1+R_{10} and C_1+R_2 datasets, 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 significant improvement in performance, in particular in the ability to

identify sequences that were further away from wildtype.

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 significantly less likely to be viable around 8-12 steps from WT, this decline is at least partly explained by the choice of a 20-iteration maximum model-design iterations mentioned above.

Sequence clustering

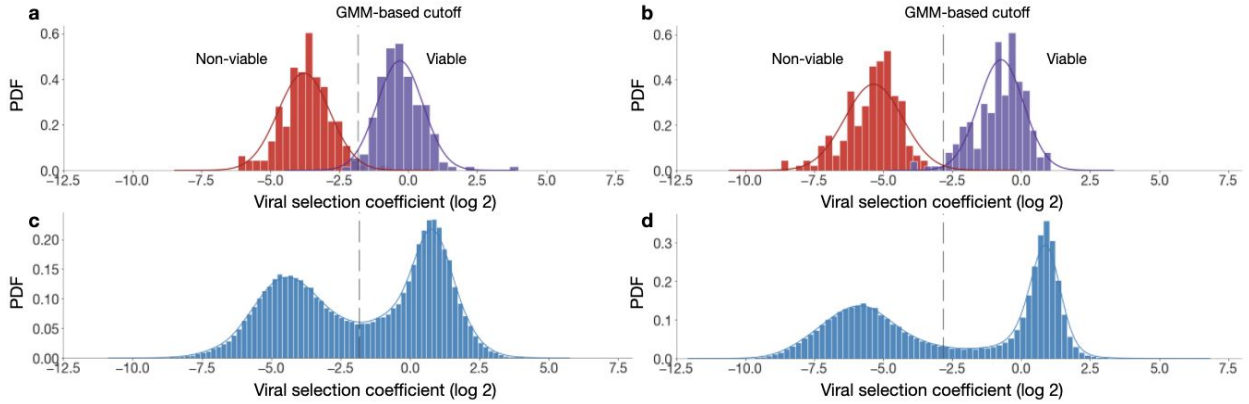
To cluster each set of model designed sequences, we first sorted them in descending order by the number of mutations from WT AAV2 (i.e. farthest first). For a given cluster radius, *R*, we start with the first sequence in the list and use it as a founder, then build a cluster around it by including every as-yet unclustered sequence <*R* edit distance from the founder sequence. 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 (e.g., 2.5x 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 to present 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 of the downsampled sequences (viable + non-viable) for the statistics presented in Fig 4b, which quantifies the volume of sequence space covered versus capsid design success rates (% viable) at a fixed radius of 12 (μ AAV serotype diversity).

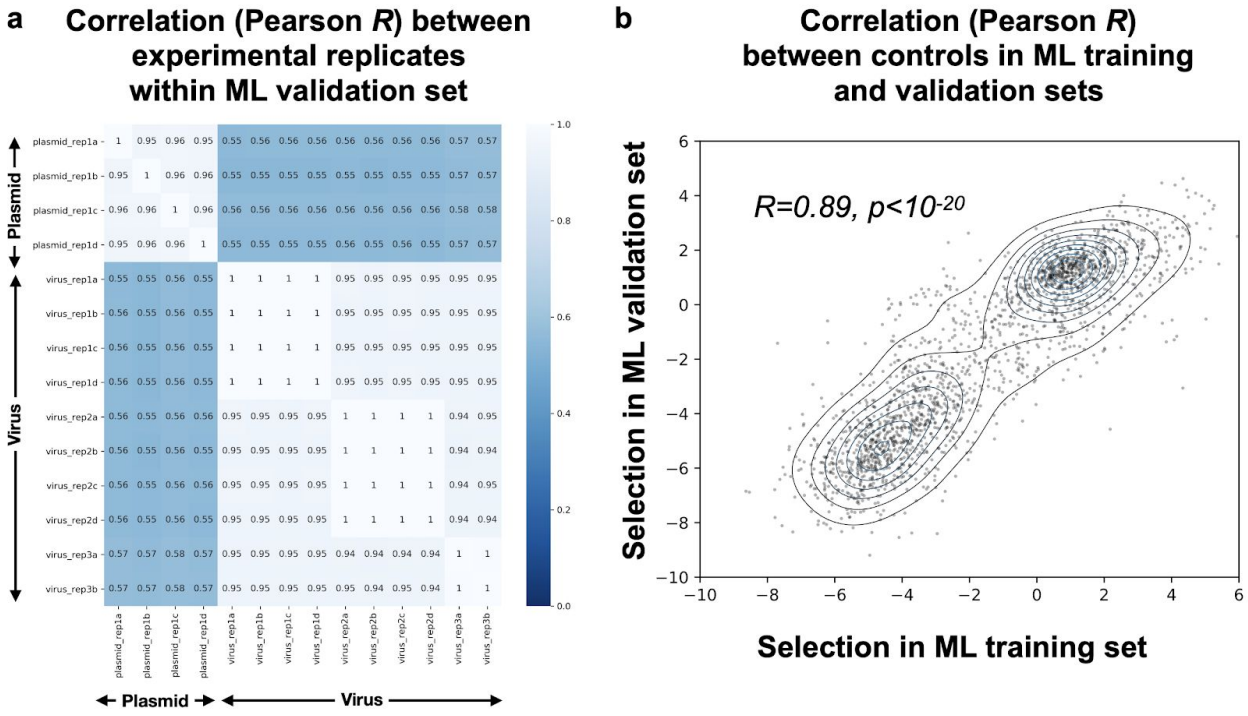
AAV2 homolog selection and alignment construction

In order to compare our diversification approach with natural diversity, we investigated the available sequences on NCBI for dependoparvoviruses. We found 415 complete coding sequence records (~1000 gene products) for dependoparvoviruses (txid 10803) which contained a structural or VP protein. This data was parsed to extract structural and VP1 proteins. Records without a complete structural or VP protein were discarded. We also ensured that we included all common AAV VP1 sequences (12 sequences). This processing results in 310 unique sequences which we aligned using ClustalW [*38*]. We then extracted the corresponding sequence to 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, the dependoviruses show slightly *less* diversity than the common 12 serotypes (i.e. a lot of sequences are quite similar to each other, and AAV2). Therefore, for comparisons in the paper, we used the common 12 serotypes as a stricter benchmark.

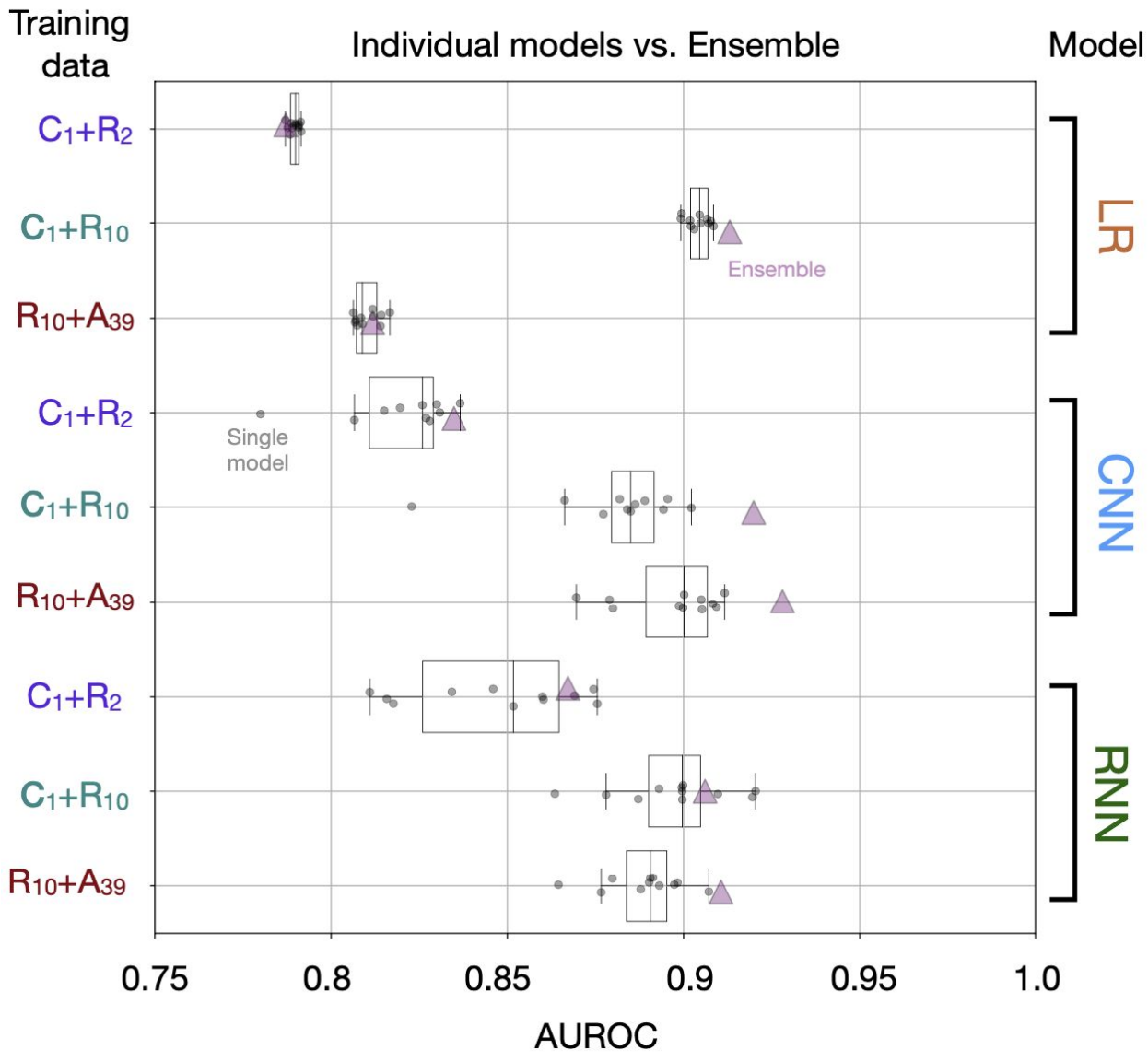
Supplementary Information



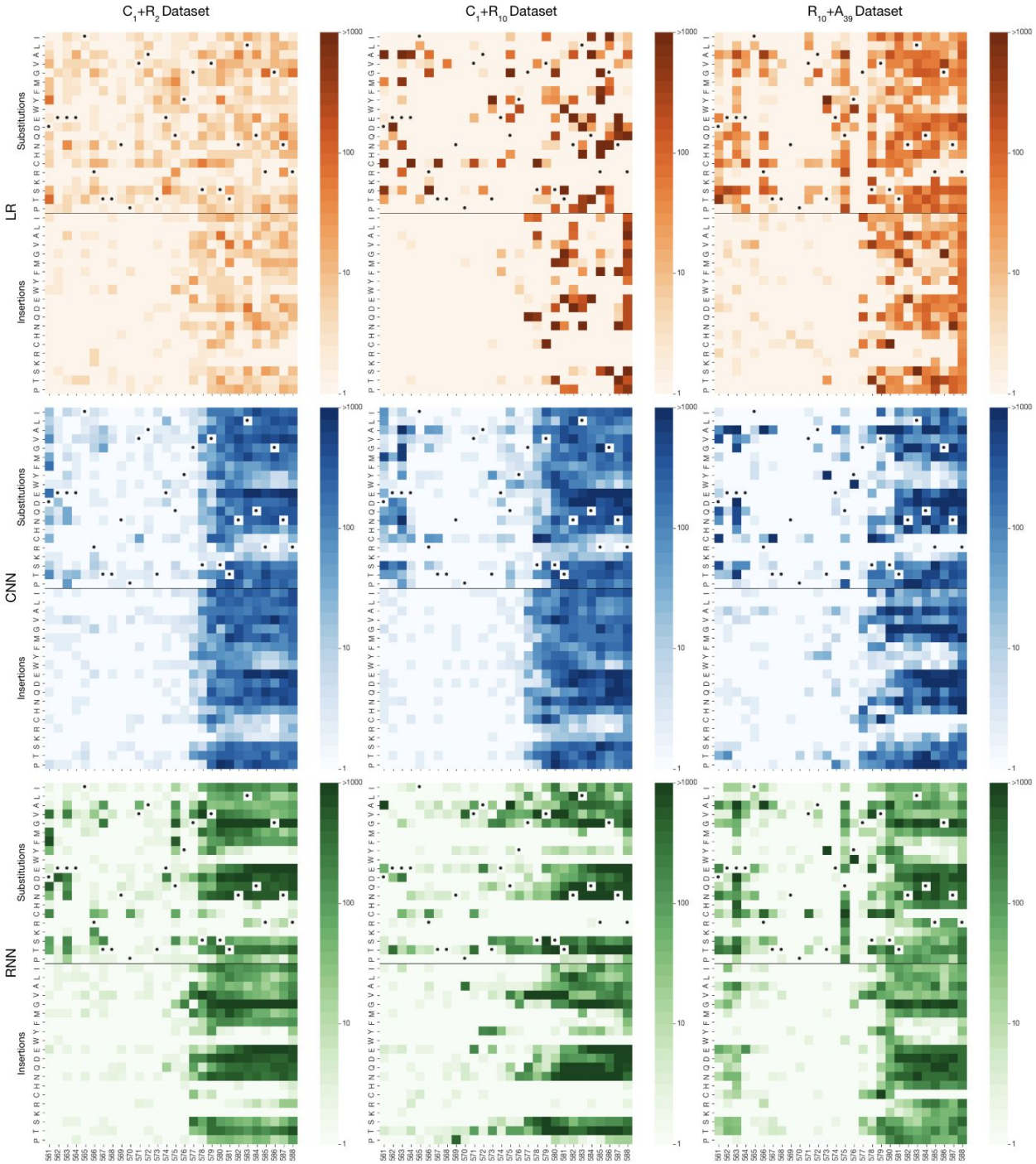
Supplementary Figure 1 | Bimodal packaging viral selection coefficient distribution. a, Viral selection coefficients for 168 sequences known to produce successfully (WT AAV2 alternate codon variants) and 162 sequences known to fail at production (capsid variants truncated via stop codon insertions) from the initial experiment. The viral selection threshold for the viable/non-viable classes was determined by fitting the 2-component GMM shown (red and purple lines), on a log2 scale. **b**, Viral selection coefficients for 200 sequences known to produce successfully (WT AAV2 alternate codon variants) and 171 sequences known to fail at production (capsid variants truncated via stop codon insertions) from the final experiment. **c, d** Distributions of all >70k variants from the initial experiment and all >240k variants from the final experiment are bimodal, motivating our use of categorical prediction models. These distributions illustrate the binary nature of the packaging assay outcomes and the intrinsic measurement variance associated with the assay.



Supplementary Figure 2 | Reproducibility within and across experiments, a, Pearson correlation between plasmid replicates and virus replicates in ML validation set. For each of the four biologically replicated virus experiment (numbered), we have at least two technical (PCR) replicates (denoted by letters). **b**, We replicated the measurements for 2000 unique samples with a range of selection scores from our ML training data as control on the validation chip designed by the classifiers to calibrate our comparison with the additive model and ensure reproducibility of results.

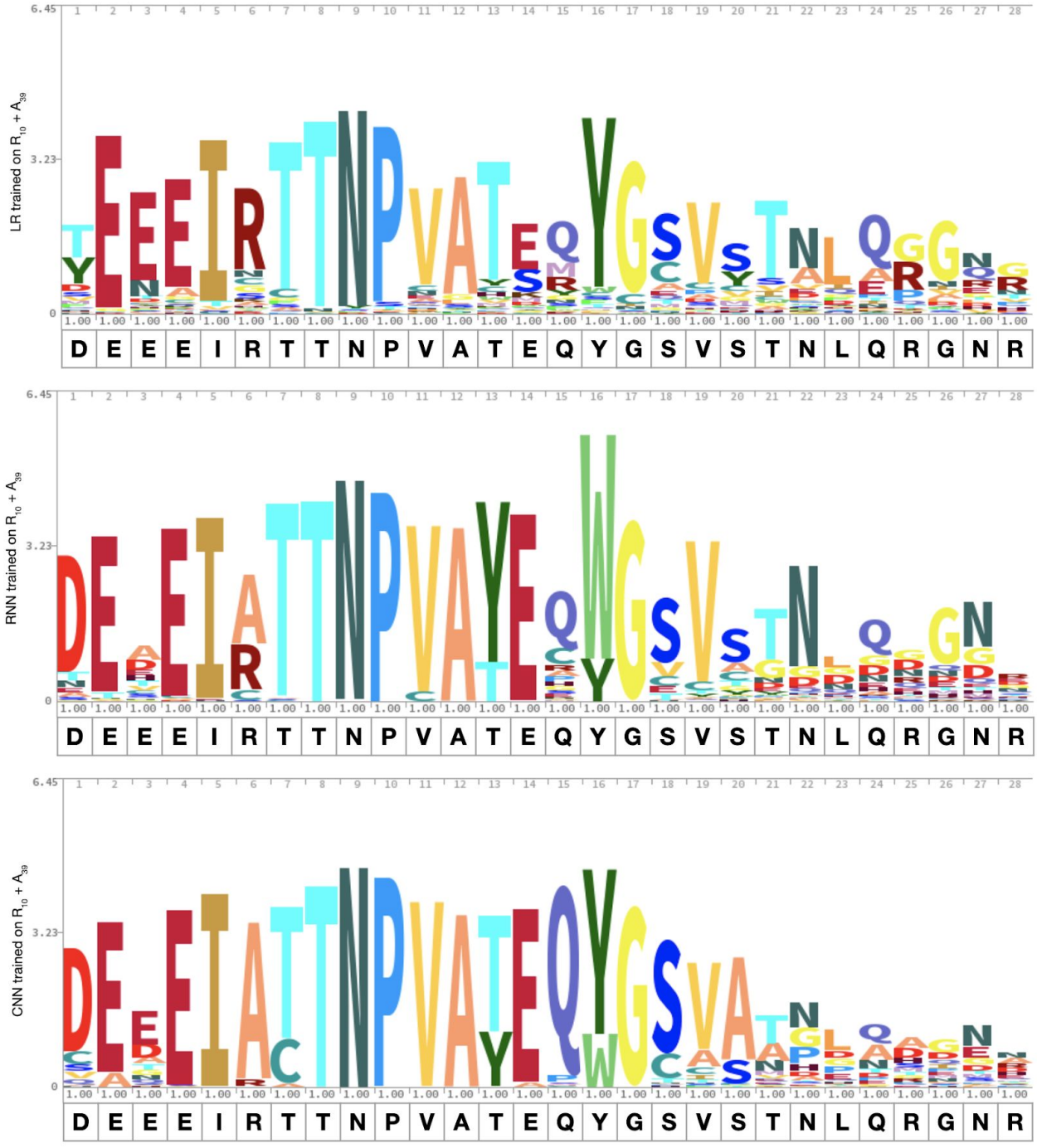


Supplementary Figure 3 | Comparison of individual and ensemble model performance. Evaluation of the performance of both the single (black dots) and ensemble (pink triangles) models built for each architecture/training set combination using the area under the receiver operating characteristic (AUROC). For the ensemble, we average the scores of each eleven individual models before computing the AUROC. Overall we find that the ensembles consistently outperform the median performance of individual models, in some cases outperforming the best individual model as well. Note that logistic regression replicate models tend to display highly similar performance regardless of initialization, while the effects of random initializations can be quite significant for the neural networks. As a result, the performance gain due to ensembling is particularly notable for the neural network models.

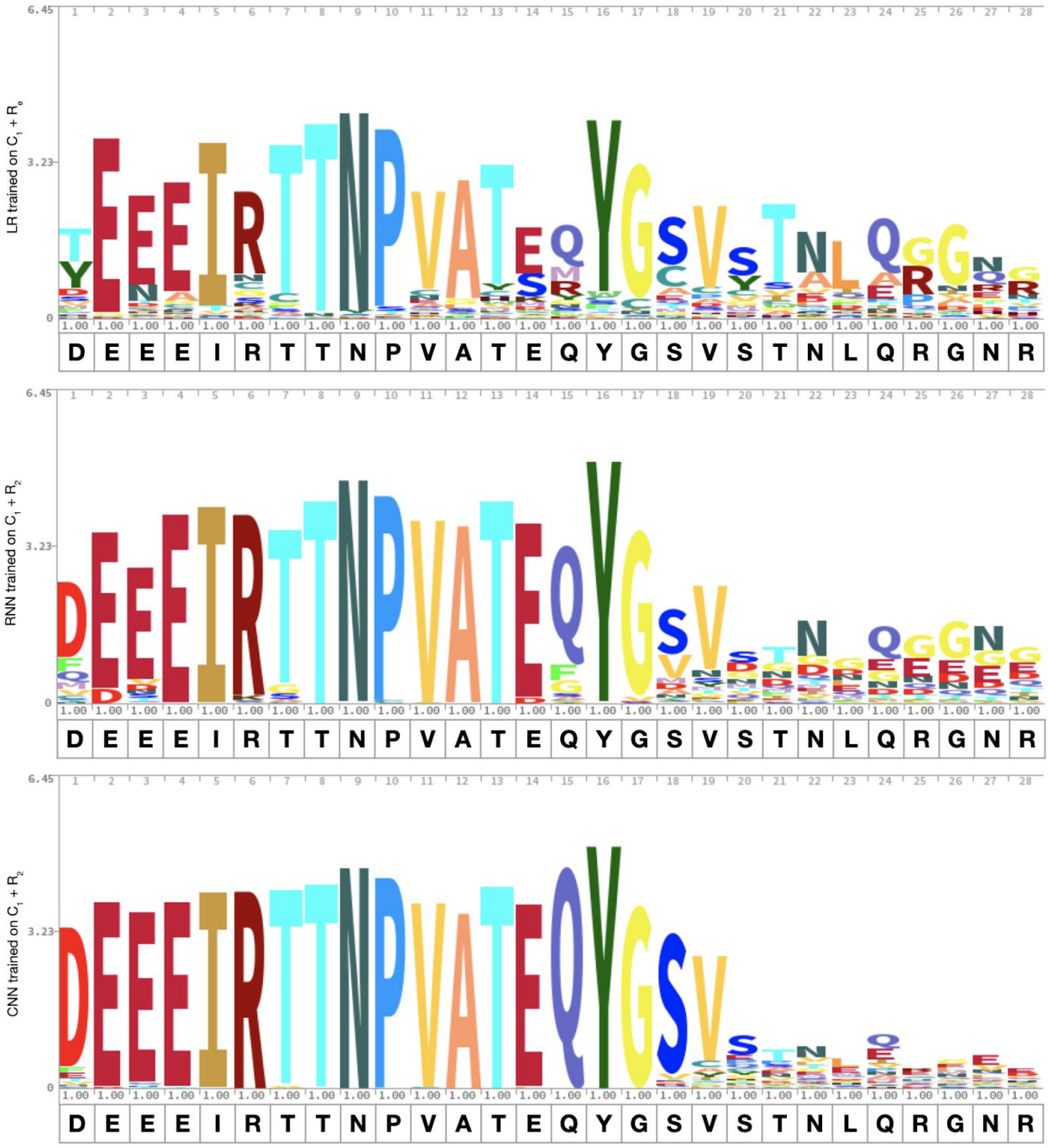


Supplementary Figure 4 | (a) Mutation preference distribution for all ML models. Heatmaps showing counts of substitutions (top) and insertions (bottom) within viable mutant capsids with ≥12 mutations as designed by each model architecture (LR, CNN, RNN), trained on each dataset.

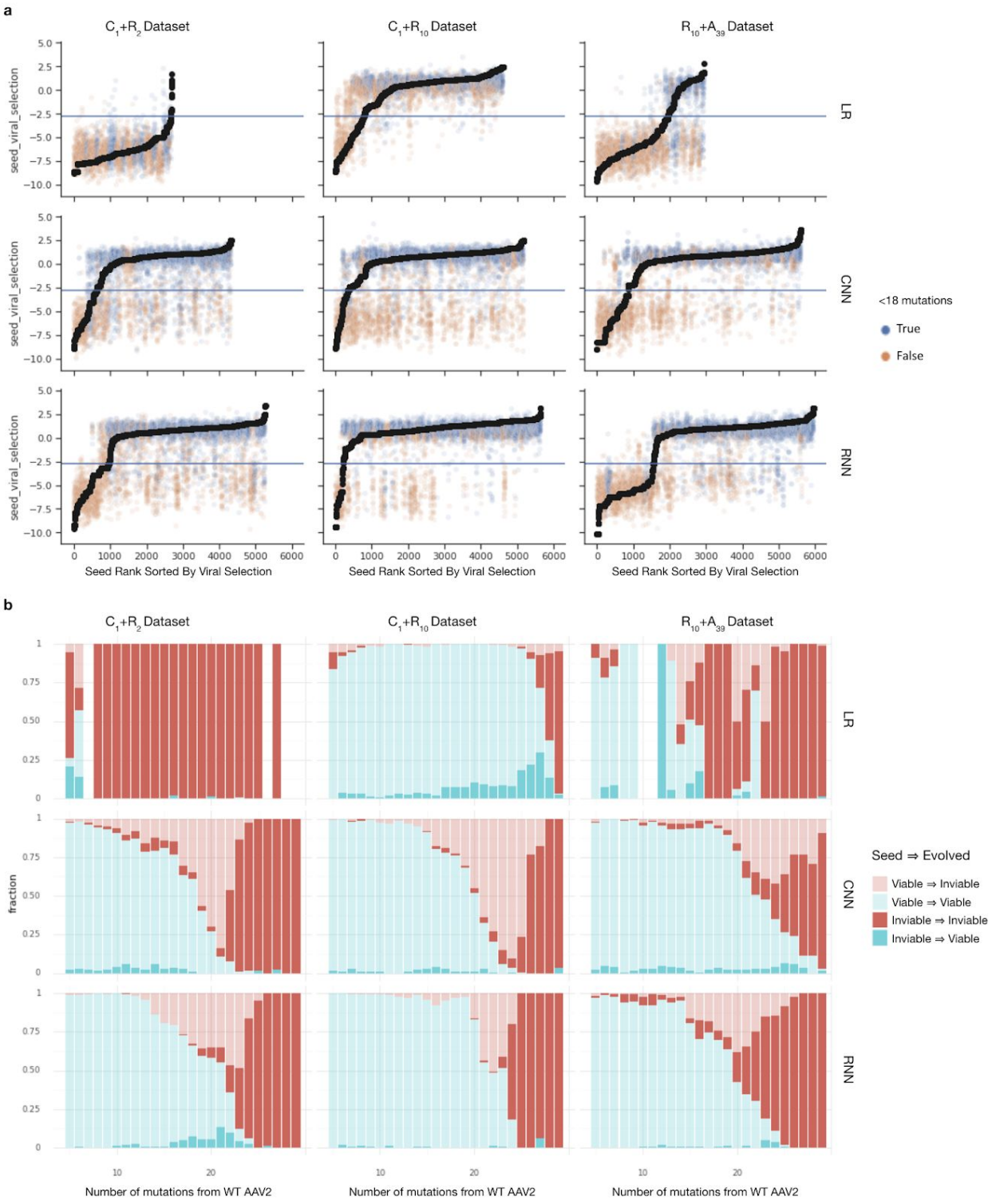
Supplementary Figure 4 | (b) Logos showing viable model-designed sequences for ML models trained on the C_1 + R_{10} dataset. Sequence logos showing amino acid usage within viable mutant capsids with ≥12 mutations from the AAV2 wildtype sequence as designed by each model architecture (LR, CNN, RNN). The wildtype AAV2 sequence is shown in black below each logo.



Supplementary Figure 4 | (c) Logos showing viable model-designed sequences for ML models trained on the R_{10} + A_{39} dataset. Sequence logos showing amino acid usage within viable mutant capsids with ≥12 mutations from the AAV2 wildtype sequence as designed by each model architecture (LR, CNN, RNN). The wildtype AAV2 sequence is shown in black below each logo.



Supplementary Figure 4 | (d) Logos showing viable model-designed sequences for ML models trained on the C_1 + R_2 dataset. Sequence logos showing amino acid usage within viable mutant capsids with ≥12 mutations from the AAV2 wildtype sequence as designed by each model architecture (LR, CNN, RNN). The wildtype AAV2 sequence is shown in black below each logo.



Supplementary Figure 5 | Relationship between model-designed sequences and their model-selected starting seeds. a, The set of model-designed sequences with experimentally tested seeds are shown within each facet. Model-designed sequences for a particular seed are rendered at the same x-axis position and colored by whether they were <18 (blue) or >=18 (orange) mutations from wildtype. The seeds are sorted by their viral selection value (y-axis). The horizontal blue line corresponds

to the viability cutoff. Most models show a strong preference for viable model-designed sequences from viable seeds. **b,** The relative fraction of viable (blue) and non-viable (red) model-designed sequences that came from viable seeds (dark alpha) and non-viable seeds (light alpha). Most models start from viable seeds and identify viable children close to WT. Far from WT, models become less reliable and more likely to start from non-viable seeds.that came from viable seeds (dark alpha) and non-viable seeds (light alpha). Most models start from viable seeds and identify viable children close to WT. Far from WT, models become less reliable and more likely to start from non-viable seeds.

Supplementary Table 1 | ML-generated AAV2 capsid statistics by mutation count. Cumulative viable capsid generation statistics across all machine learning models (LR, CNN and RNN), including both model-designed and model-selected sequences across all training datasets (C_1+R_2, C_1+R_{10}, and R_{10}+A_{39}), for a range of mutations-from-WT thresholds. The bolded row corresponds to the mutation distance at which the models first exceed the additive model in % viable capsids.

Min Mutations Threshold	# Generated Capsids	# Viable Capsids	% Viable Capsids	%Viable Capsides (Additive Model)
2	201,426	110,689	55.00%	62.50%
3	201,426	110,689	55.00%	59.50%
4	**201,424**	**110,687**	**55.00%**	**53.70%**
5	201,368	110,633	54.90%	46.10%
6	193,413	103,403	53.50%	36.40%
7	184,424	95,422	51.70%	21.30%
8	175,443	87,571	49.90%	17.30%
9	166,361	79,628	47.90%	13.70%
10	157,294	72,180	45.90%	10.70%
11	148,167	64,678	43.70%	8.30%
12	138,815	57,348	41.30%	6.30%
13	129,433	50,330	38.90%	4.70%
14	119,469	43,236	36.20%	3.50%
15	109,474	36,173	33.00%	2.40%
16	99,137	29,326	29.60%	1.60%
17	88,694	22,901	25.80%	1.00%
18	78,951	17,588	22.30%	0.60%
19	69,612	13,233	19.00%	0.40%
20	60,049	9,710	16.20%	0.30%
21	51,164	7,048	13.80%	0.10%
22	42,202	4,952	11.70%	0.00%
23	33,500	3,301	9.90%	0.00%
24	24,879	1,983	8.00%	0.00%
25	16,977	1,038	6.10%	0.00%
26	11,089	484	4.40%	0.00%
27	7,350	196	2.70%	0.00%
28	4,094	52	1.30%	0.00%
29	1,489	10	0.70%	0.00%

Supplementary Table 2 | ML-designed AAV2 capsid statistics by mutation count. Cumulative viable capsid generation statistics across all machine learning models (LR, CNN and RNN), for only model-designed sequences (i.e., excludes model-selected) across all training datasets (C_1+R_2, C_1+R_{10}, and R_{10}+A_{39}). The bolded row corresponds to the mutation distance at which the models first exceed the additive model in % viable capsids.

Min Mutations Threshold	# Generated Capsids	# Viable Capsids	% Viable Capsids	%Viable Capsides (Additive Model)
2	183,466	106,665	58.10%	62.50%
3	183,466	106,665	58.10%	59.50%
4	**183,464**	**106,663**	**58.10%**	**53.70%**
5	183,411	106,612	58.10%	46.10%
6	176,351	100,150	56.80%	36.40%
7	168,231	92,923	55.20%	21.30%
8	160,096	85,766	53.60%	17.30%
9	151,805	78,411	51.70%	13.70%
10	143,464	71,416	49.80%	10.70%
11	135,099	64,267	47.60%	8.30%
12	126,589	57,157	45.20%	6.30%
13	118,046	50,243	42.60%	4.70%
14	108,965	43,188	39.60%	3.50%
15	99,868	36,138	36.20%	2.40%
16	90,448	29,299	32.40%	1.60%
17	80,932	22,879	28.30%	1.00%
18	72,082	17,571	24.40%	0.60%
19	63,657	13,217	20.80%	0.40%
20	55,026	9,698	17.60%	0.30%
21	47,032	7,039	15.00%	0.10%
22	38,986	4,946	12.70%	0.00%
23	31,190	3,297	10.60%	0.00%
24	23,441	1,980	8.40%	0.00%
25	16,395	1,037	6.30%	0.00%
26	11,089	484	4.40%	0.00%
27	7,350	196	2.70%	0.00%
28	4,094	52	1.30%	0.00%
29	1,489	10	0.70%	0.00%

Supplementary Table 3 | NN-designed AAV2 capsid statistics by mutation count. Cumulative viable capsid generation statistics across all neural network models (CNN and RNN) for only model-designed sequences across all training datasets (C_1+R_2, C_1+R_{10}, and R_{10}+A_{39}) for a range of mutations-from-WT thresholds (i.e., excludes model-selected sequences). The bolded row corresponds to the mutation distance at which the models first exceed the additive model in % viable capsids.

Min Mutations Threshold	# Generated Capsids	# Viable Capsids	% Viable Capsids	%Viable Capsides (Additive Model)
2	**123,331**	**79,837**	**64.70%**	**62.50%**
3	123,331	79,837	64.70%	59.50%
4	123,329	79,835	64.70%	53.70%
5	123,280	79,788	64.70%	46.10%
6	117,855	74,431	63.20%	36.40%
7	112,376	69,020	61.40%	21.30%
8	106,907	63,624	59.50%	17.30%
9	101,326	58,145	57.40%	13.70%
10	95,698	52,658	55.00%	10.70%
11	90,035	47,192	52.40%	8.30%
12	84,291	41,688	49.50%	6.30%
13	78,449	36,219	46.20%	4.70%
14	72,332	30,635	42.40%	3.50%
15	65,960	24,953	37.80%	2.40%
16	59,277	19,247	32.50%	1.60%
17	52,702	13,997	26.60%	1.00%
18	46,774	9,856	21.10%	0.60%
19	41,028	6,559	16.00%	0.40%
20	35,300	4,092	11.60%	0.30%
21	30,053	2,482	8.30%	0.10%
22	24,771	1,385	5.60%	0.00%
23	19,712	670	3.40%	0.00%
24	15,145	338	2.20%	0.00%
25	10,854	165	1.50%	0.00%
26	7,306	72	1.00%	0.00%
27	4,887	47	1.00%	0.00%
28	2,666	15	0.60%	0.00%

29	942	4	0.40%	0.00%

Supplementary Table 4 | Model-selected AAV2 capsid statistics per ML model.

Model	# Generated Capsids	# Viable Capsids	% Viable Capsids
LR{C_1+R_2}	2,071	114	5.5%
LR{C_1+R_{10}}	1,989	486	24.4%
LR{R_{10}+A_{39}}	2,030	340	16.7%
CNN{C_1+R_2}	2,022	381	18.8%
CNN{C_1+R_{10}}	1,924	476	24.7%
CNN{R_{10}+A_{39}}	1,898	529	27.9%
RNN{C_1+R_2}	2,045	575	28.1%
RNN{C_1+R_{10}}	1,916	412	21.5%
RNN{R_{10}+A_{39}}	2,065	711	34.4%

Supplementary Table 5 | Model-designed AAV2 capsid statistics per ML model.

Model	# Generated Capsids	# Viable Capsids	% Viable Capsids
LR{C_1+R_2}	19,999	1,483	7.4%
LR{C_1+R_{10}}	20,456	19,211	93.9%
LR{R_{10}+A_{39}}	19,680	6,134	31.2%
CNN{C_1+R_2}	20,454	11,229	54.9%
CNN{C_1+R_{10}}	20,395	13,086	64.2%
CNN{R_{10}+A_{39}}	20,759	14,968	72.1%
RNN{C_1+R_2}	20,154	13,056	64.8%
RNN{C_1+R_{10}}	20,838	15,525	74.5%
RNN{R_{10}+A_{39}}	20,731	11,973	57.8%

Supplementary Table 6 | Additive model (A_{39}) capsid statistics. Cumulative across edit distance thresholds.

Min Mutations Threshold	# Generated Capsids	# Viable Capsids	% Viable Capsids
2	56,372	35,217	62.5%
3	50,572	30,068	59.5%
4	41,232	22,129	53.7%
5	31,561	14,551	46.1%
6	22,407	8,159	36.4%
7	13,892	2,953	21.3%
8	12,603	2,181	17.3%
9	11,387	1,561	13.7%
10	10,245	1,101	10.7%
11	9,171	757	8.3%
12	8,160	511	6.3%
13	7,195	340	4.7%
14	6,312	224	3.5%
15	5,495	134	2.4%
16	4,757	74	1.6%
17	4,102	42	1.0%
18	3,522	22	0.6%
19	2,994	13	0.4%
20	2,541	8	0.3%
21	2,148	2	0.1%
22	1,790	0	0.0%
...	...	...	...
37	30	0	0.0%
38	16	0	0.0%
39	3	0	0.0%

Supplementary Table 7 | Randomly generated (R_{10}) capsid statistics. Cumulative across edit distance thresholds.

Min Mutations Threshold	# Generated Capsids	# Viable Capsids	% Viable Capsids
2	9,885	964	9.80%
3	8,129	461	5.70%
4	6,378	213	3.30%
5	4,631	93	2.00%
6	2,883	32	1.10%
7	1,154	3	0.30%
8	866	2	0.20%
9	576	1	0.20%
10	284	1	0.40%