# EngrAAV: A Novel Application of LSTM Neural Networks to Design Viable Recombinant-AAV Capsids

Nikhil Nayak, Akash Pai

Sunset High School, 13840 NW Cornell Rd, Portland, Oregon, 97229

**ABSTRACT**

Adeno-associated viruses (AAVs), being the most clinically successful viral vectors, are essential to gene therapy. To maximize the output of AAVs, capsids must be optimized via capsid engineering. Current solutions, like directed evolution and rational design, have several drawbacks: time, cost, throughput, and a high volume of sub-optimal capsids. In this project, we propose *EngrAAV*, a deep-learning-based AAV capsid design pipeline, that aims to maximize viral assembly. While deep-learning systems exist to design AAV capsids, the current state-of-the-art fails to predict capsid viability at high accuracy. In this paper, A Long-Short Term Memory (LSTM) Neural Network is leveraged to predict viral assembly on a dataset of mutated capsid sequences, augmented for increased data points. By applying Hyperparameter tuning and finetuning, we were able to increase the model's validation accuracy to 95%. Next, EngrAAV was validated on pre-screened datasets. Lastly, we applied the trained EngrAAV pipeline on a real-world application. By randomly mutagenizing capsid sequences and ranking them with EngrAAV, we were able to design new AAV capsids. These capsids were part of a library screened for viral assembly capability. We compared our capsids with their viral assembly metric, allowing us to determine the efficacy of EngrAAV with real sequences. Through this process, the model successfully designed two novel capsids with greater viral assembly than the wild-type.

## 1. INTRODUCTION

Adeno-associated viruses (AAV) are commonplace in gene therapy, and widely considered to be safe and effective. The AAV is a member of the Paroviridae family. Its base viral structure consists of 60 viral proteins (VP), produced in three types: VP1, VP2, and VP3. These VPs assemble in a 1:1:10 stoichiometric ratio, respectively, forming the icosahedral-shaped AAV (Naso et al., 2017). While AAVs have been extensively studied, optimizing various viral properties still remains challenging (FDA, 2017).

Directed evolution and rational design are the current widely-utilized solutions to optimize the recombinant-AAV (rAAV) or, for this study, its capsid (Wang et al., 2021; Lee et al., 2018). Directed evolution is an artificial, iterative process designed to mimic natural selection. Directed evolution requires a library of rAAV variants created through mutagenesis or the shuffling of an rAAV capsid gene among many serotypes. Each mutant is screened for fitness experimentally. Top-performing mutants are selected to undergo further mutagenesis or shuffling (Packer et al., 2015). Iterations continue until rAAV variants that satisfy benchmarks in the desired function are chosen. The other widely-used solution is rational design. In rational design, the scientist utilizes detailed knowledge of the protein structure to make pinpoint, desired changes (Korendovych, 2019).

The unfortunate drawbacks of directed evolution are the following. Firstly, directed evolution only utilizes experimental data of selected variants. Second, directed evolution often starts from an rAAV, with libraries built with few mutations per variant. This means that selected sequences are often close to the wild-type (WT) and, therefore, likely suboptimal. Next, due to the nature of directed evolution's iterative process, it is often very costly and time-consuming, limiting iterations and throughput (To et al., 2021). Additionally, the AAV capsids' complex

design causes rational design to fall short. Today, researchers have not been able to effectively predict the effects of mutations on AAV capsids (Korendovych, 2019). This lack of information limits the capability of rational design. To solve the aforementioned problems, we implement new methods, such as machine learning, to create a high-throughput, low-cost, and timely system that can accurately design capsids.

While recent efforts have been made to use ML to design AAV capsid sequences (Bryant et al., 2021), these models' ability to map sequence function is limited, and their ability to design diverse, viable capsid sequences is low. We aim to develop a machine-learning-based pipeline to design highly viable AAV capsid sequences based on datasets created in vivo.

In this study, we apply ML to design highly viable capsid sequences from AAV serotype two (AAV2). Specifically, we focus on optimizing amino acid (AA) position 560-588 in the VP3. AAV2 has a wide host range, good transduction efficacy, lack of pathogenicity, and capsid versatility (Genemedi, n.d.). Additionally, the 28 AA position chosen for optimization is a variable region in the capsid. (Bryant et al., 2021). Due to the promise of AAV2, by applying ML to optimize its capsid, we hope to further efforts in developing an ideal viral vector for human gene therapy. Further, to validate our pipeline, we use three datasets composed of capsid sequences with varying numbers of mutations. Here, we developed an ML approach that can be used to design highly viable de novo AAV2 capsids using datasets created in vivo.

## 2. MATERIALS AND METHODS

***2.1 AAV2 Mutated Capsid Dataset.*** We utilize a library of AAV2 capsid sequences named c1r2.csv, which consists of sequences mutated from the specified 28 AA region to 30 AAs. No capsid sequence in this dataset had more than two mutations, which occur from substituted or inserted AAs. The dataset is in (x, y) format. The x represents the 2620 capsid sequences. The y

represents viral assembly: the ability of a capsid to assemble and package DNA. Viral assembly is a selection score, $S1 = log_2(\frac{f1^{virus}}{f1^{plasmid}})$, where $f1$ represents the frequency of a variant in plasmid or virus pool measured through next-generation sequencing (NGS), also known as viral titer. $S1$ is generated in decimal format but is normalized into two categories: viable and nonviable. A mutated capsid sequence is viable if it achieves a greater or equal selection score than the WT 28 AA capsid sequence. For reference, the WT 28 AA capsid sequence has an $S1$ of -2.8 (Sinai et al., 2021). As a proof-of-concept to show that machine learning can predict capsid viability, we apply this dataset to the model for training and pipeline development. Furthermore, we implement this pipeline on datasets with capsid sequences mutated more times and at different locations. Finally, to test real application, we create a dataset of randomly mutagenized capsid sequences. We apply this to the pipeline, and model-selected sequences are experimentally validated to determine if the model can design viable capsid sequences.

***2.2 Data Augmentation.*** Machine learning models require extensive datasets to perform well. The dataset in *section 2.1* consists of 2620 capsid sequences, which is not enough data for a model to predict capsid viability adequately. Therefore, we apply augmentation strategies to increase data points. Due to time and cost restraints, we could not generate mutated capsid sequences and screen them for viability. We therefore implement strategies to shuffle screened mutants, ensuring that while they are represented differently, the order of AAs did not change (Shen et al., n.d.). Firstly, we add AAs from the WT capsid sequence outside the target 28 AA region. We target AA positions 560-588, but add AAs to the left and right of this region from positions 550-559 and 589-598. Secondly, we utilize sequence reversal. For every sequence, we add a duplicate sequence to the dataset, but with the AAs reversed. These methods do not change

the order of the mutated capsid sequence but provide increased representations of the same sequences to help the model learn.

***2.3 Encoding of Dataset.*** We develop a trainable Embedding layer, which converts each AA into a form the model understands. Initially, "one-hot encoding," a method that places each amino acid on its dimension (i.e., Alanine would be [1, 0, 0, 0 …] while Arginine would be [0, 1, 0, 0…]), was implemented. However, this removes any correlation between similar Amino Acids. Embedding layers work similarly but decrease the dimensionality of the output, which constrains the Embedding optimizer to encode data better before it enters the model (Saxena, 2020). In this model, a dimensionality of 41*26 was chosen. It is also essential to consider that the Embedding layer is not trained individually but rather back-propagated with the final outputs from the model that follows.

***2.4 Machine Learning Analysis.*** We utilize a Long-Short Term Memory (LSTM) neural network for this dataset. After capsid sequences are encoded for the model, they pass through four stacked LSTM and dropout layers. Data proceeds through a dense and softmax layer, where a prediction is made on a mutated sequence's viability. To improve model accuracy, we implement hyperparameter tuning. We perform Keras Hyperband hyperparameter tuning. Hyperband uses adaptive resource allocation and early-stopping, randomly sampling all hyperparameter combinations. The model is trained for a few epochs with one combination of hyperparameter tuning and the best candidates are selected. Hyperband does this iteratively, running full training and evaluation on chosen hyperparameters (Wadekar, 2021). After EngrAAV is trained with optimized hyperparameters, finetuning is applied. Fine Tuning enables

the model to perform better on capsid sequences with a certain number of mutations. This proved inherently useful when ranking sequences from datasets with a consistent number of mutations. This fully developed model ranks capsid sequences on predicted viral assembly and selects sequences from a dataset of randomly mutagenized capsid sequences.

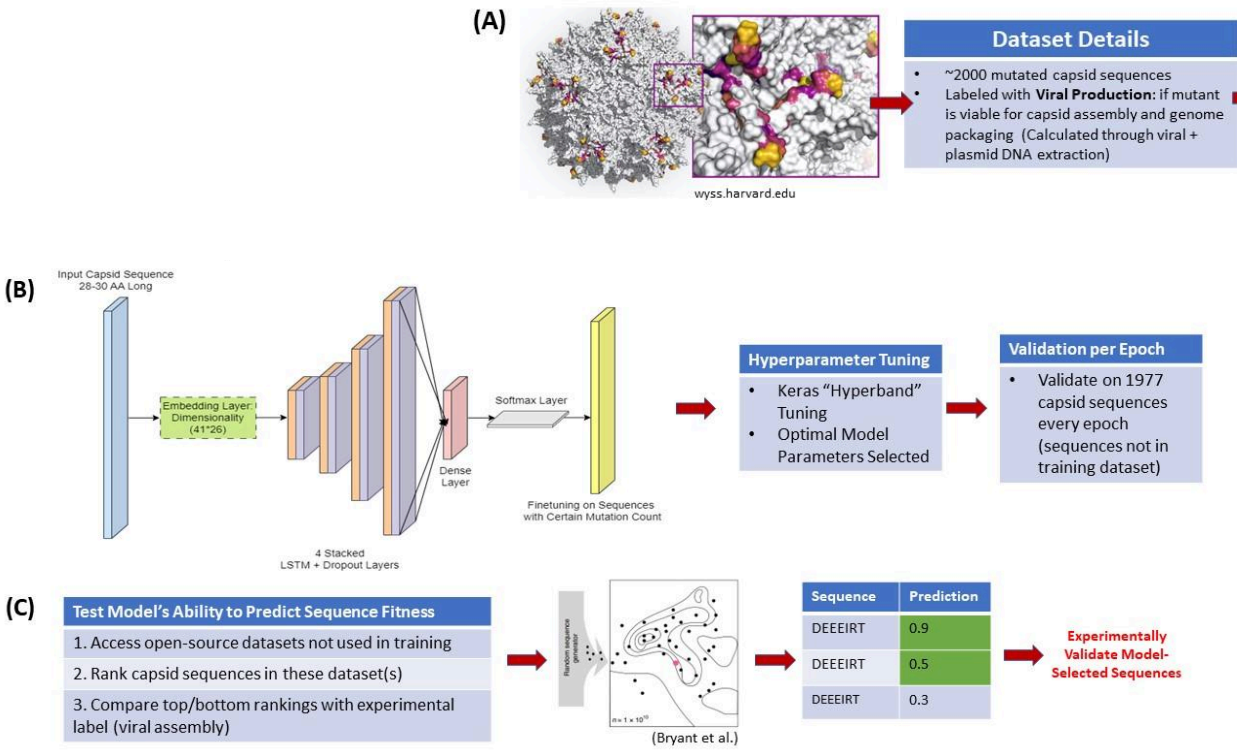


**Figure 1.** Schematic describing the methods followed to develop this project. (A) describes dataset creation, preprocessing and augmentation. (B) model development, and (C) validation of pipeline by testing it on multiple datasets and experimentally validating model-designed sequences.

| **Hyperparameter** | **Value** |
|---|---|
| Embedding_Dimension | 26 |
| LSTM_Units | 96 |
| Learning Rate | 1e-3 |

| | |
|---|---|
| LSTM_Layers | 4 |
| Dropout | 0.85 |

**Figure 2.** Final optimized hyperparameters.

***2.5 Selecting Randomly Mutagenized Capsid Sequences and Experimentally Validating Them.***

Once the model has reached a target accuracy of >90% on the training dataset for the proof-of-concept: predicting capsid viability, EngrAAV has accurately learned what features in a sequence entail high or low viability. It must now be determined if EngrAAV can design highly viable sequences. This is done by randomly mutagenizing sequences of a particular mutation count and using EngrAAV to select the "most viable" mutants. Using python scripts, we randomly mutagenize the WT 28 AA capsid region to have 12 mutations, doing this 10,000 times, creating a dataset of 10,000 unique mutated capsid sequences with 12 mutations each. The dataset is named r12mutants.csv. EngrAAV selects sequences based on predicted capsid viability. The library of 10,000 is sent to the lab, where mutants are screened for viability using NGS. We compare EngrAAV's designed sequences to their actual viral assembly number.

## 3. RESULTS

Results are grouped into five sections. *3.1* displays model performance on predicting capsid viability. *3.2-3.4* analyzes top-ranked sequences predicted by EngrAAV from datasets *allseqs_20191230* and *ValidationChipWithModelScores*. *3.5* discusses EngrAAV's ability to design de novo sequences, compares EngrAAV to the SOTA, and discusses the potential limitations of EngrAAV.

***3.1 Model performance on validation dataset.*** We test our model's ability to predict capsid viability with accuracy. EngrAAV validates on dataset *holdout.csv*. Compared to the experimentally validated results, the EngrAAV pipeline had a validation accuracy of ~95%. Similar to accuracy, we conduct the same process, evaluating AUROC, a standard metric in testing machine learning models. AUROC is the measure of the ability of a classifier to distinguish between classes. The EngrAAV model achieved a 0.988 AUROC.

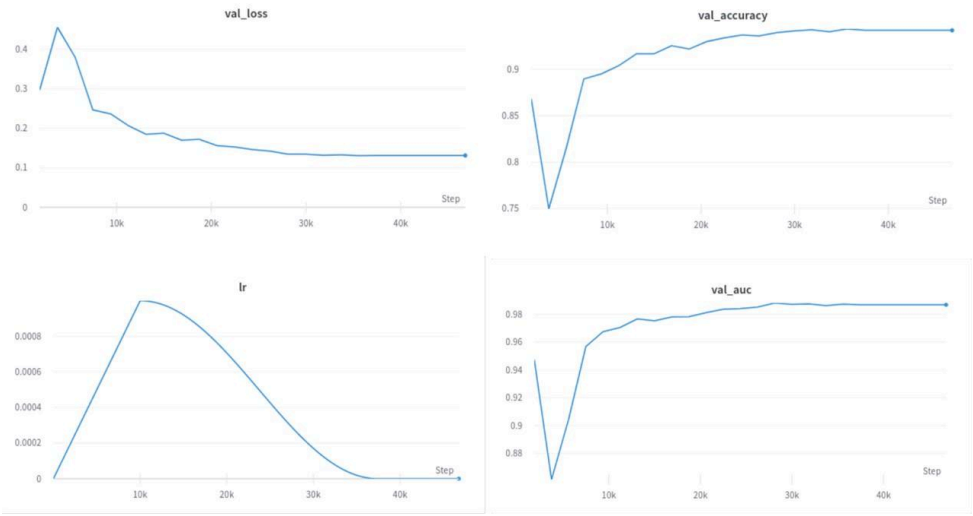


**Figure 3.** Model performance per training step progression. Top left: validation loss; top right: validation accuracy; bottom left: learning rate; bottom right: validation AUROC.

***3.2 Analysis of sequence logos.*** EngrAAV ranked sequences in two datasets: allseqs_20191230.csv and ValidationChipWithModelScores.csv. Two sequence logos were created per dataset to analyze ranked sequences. Multi-Sequence Alignment (MSA) was utilized, which requires all analyzed sequences to be of the same length. Due to EngrAAV-ranked sequences predominantly consisting of 32 AAs, sequence logos were created on that length. Sequence logos provide a visual for the frequency of amino acid type, per position, in mutated

sequences. Capsid sequences with high predicted viability had a slight variance of AA-type earlier in the sequence but a significant variance of AA-type in later positions.

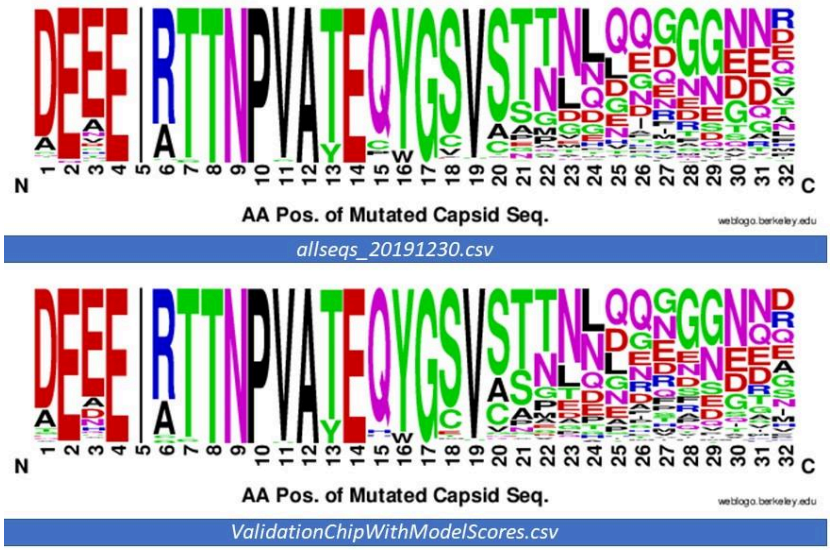


**Figure 4.** Logos showing the top-ranked sequences from two datasets by EngrAAV. Sequence logos show amino acid usage within viable mutant capsids with length, 32. Top: sequence logo for ranked-sequences from *allseqs_20191230.csv*. Bottom: sequence logo for ranked-sequences from *ValidationChipWithModelScores.csv*.

***3.3 Analysis of heatmaps.*** Heatmaps demonstrate the relationship between a single mutation and viral assembly. These heatmaps were created from EngrAAV's top-ranked sequences with only one mutation per dataset. Single mutations on AA positions late in the sequence have a positive effect on viral assembly, seen in the cluster of green at the top right of each figure.

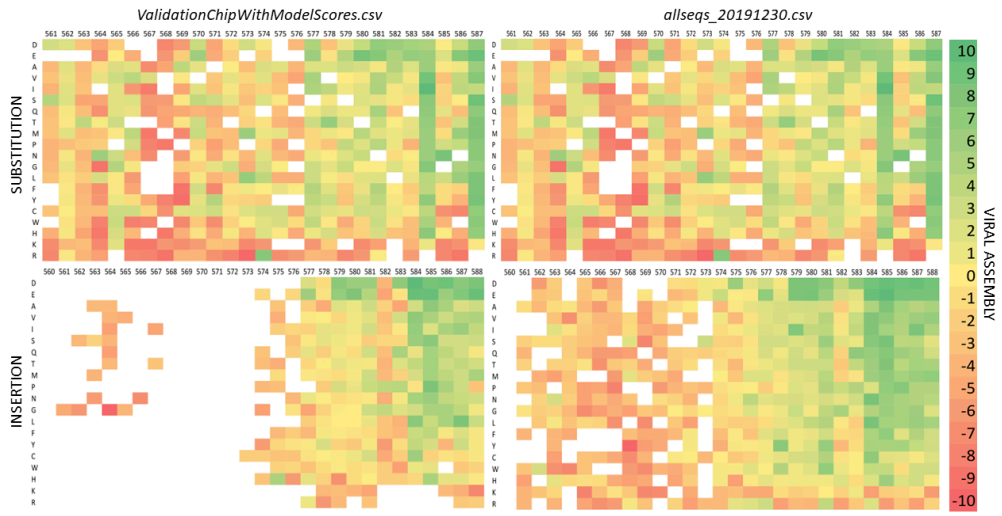


**Figure 5.** Heatmaps show correlation between location/type of single mutations and viral assembly for EngrAAV's top 1000 ranked sequences per dataset. Top: substitution mutations. Bottom: insertion mutations. X-axis label is the amino acid position, the y-axis label is the amino acid.

***3.4 Density of top-ranked sequences compared to all sequences in each dataset.*** EngrAAV's top-1000 ranked sequences in each dataset, on average, have higher viral assembly than the average viral assembly of all the sequences. This can be seen in both datasets, distinctly showing that EngrAAV can select capsids with high viral selection.

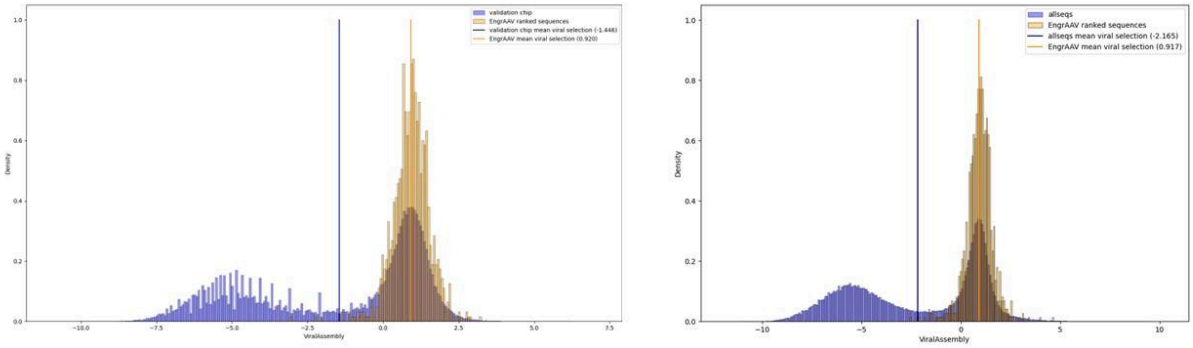


**Figure 6.** Left: Histogram of viral assembly for sequences in the *ValidationChipWithModelScores.csv* dataset. Blue shows the baseline (*ValidationChipWithModelScores)* while orange shows the 1000 sequences ranked highest by the EngrAAV model. Bright blue line depicts the average viral assembly for all mutants, while the bright orange line depicts the average viral assembly for EngrAAV selected mutants. Bottom: Same figure, but baseline and ranked sequences are from *ValidationChipWithModelScores.csv* dataset.

***3.5 Model-designed sequences; Comparison of EngrAAV to SOTA; Run time, throughput, cost.*** r12mutants.csv was experimentally validated using NGS. Two of EngrAAV's designed sequences were viable, certifying EngrAAV's ability to design viable de novo capsids.

EngrAAV generated a 0.988 AUROC, when trained on dataset *c1r2* and validated on dataset, *holdout*. The SOTA, using a Recurrent Neural Network, when trained and validated on the same datasets, features a ~0.84 AUROC. The SOTA was also tested on *r12mutants.csv*, ranking sequences in this library. All of their top-ranked sequences were found to be not viable.

EngrAAV has no limit on throughput; any amount of AAV capsid sequences can be computationally screened for viability. Any number of randomly mutagenized sequences can be generated and selected: the process for designing de novo sequences. The average training time for EngrAAV to achieve its accuracy, as noted in *3.1*, is 90.1 minutes. Additionally, EngrAAV can process up to 100,000 sequences per minute. Finally, the development of EngrAAV was of no cost. The only process in this paper of potential financial burden was experimentally validating *r12mutants.csv*.

## 4. DISCUSSION

Optimizing viral vectors for human gene therapy is essential for providing the best possible outcomes. This study focuses on the optimization of the viral capsid. Current capsid engineering strategies include directed evolution and rational design. However, these design capsids in limited throughput, do not make full use of experimental information, design sub-optimal capsid sequences due to limited mutations, and are costly and time-consuming (To et al., 2021). Here, we attempt to apply ML to design AAV2 capsids. So far, few ML approaches have been applied to this task, including a Convolutional Neural Network (CNN), Recurrent

Neural Network (RNN), and Logistic Regression (LR) (Bryant et al., 2021; Sinai et al., 2021). We develop a novel LSTM architecture that can take into account sequential information. The model predicts capsid viability and designs viable capsids in high throughput.

We used a dataset of 2620 mutated capsid sequences to train our ML model. While initial accuracy was subpar, two main strategies were implemented to increase accuracy. Firstly, data augmentation, to increase training data points from 2620 to nearly 1,000,000. Secondly, hyperparameter optimization is applied. Upon implementing these strategies, a validation accuracy of ~95% was achieved. AUROC was also tested on the model, and a value of 0.988 was achieved.

With a fully-trained model, we rank datasets on predicted capsid viability and determine if the top-ranked sequences are of better viral assembly than the average sequence. To do this, we utilize density plots, which show the average sequence viral assembly distribution, and the top-selected sequences from EngrAAV. For datasets *allseqs_20191230* and *ValidationChipWithModelScores*, EngrAAV's selected sequences had greater average viral selection than both the WT viral selection and the average viral selection of all of the sequences.

We developed sequence logos and heatmaps to analyze top-ranked sequences in the mentioned datasets. Sequence logos display AA frequency per position over multiple mutated sequences. Heatmaps show a correlation between pinpoint mutations and viral assembly rating. From the sequence logos, we learned that in high-performing sequences, earlier AA positions mainly consisted of the same type of AAs. In contrast, later AA positions had significant variability in the types of AAs. From the heatmap, we noticed that mutations in the later end of the sequence often led to higher viral selection, which correlates with information garnered in the sequence logos.

Finally, to apply EngrAAV to a real application, we built a dataset of capsid sequences with 12 mutations by randomly mutagenizing the WT using python scripts. EngrAAV selected sequences from this dataset. When the dataset was screened for viability using NGS, we found that two of EngrAAV's designed sequences returned viable. The current SOTA found no viable sequences when tested on the screened dataset.

While EngrAAV proved successful in measured metrics, there are still many desirable improvements. Firstly, the data augmentation strategies we used failed to diversify the dataset, only increasing data points. Sequence reversal and WT AA addition don't create new mutated capsid sequences, but rather, generate different representations of the same mutated sequence. Secondly, while the accuracy on the sequences with 12 mutations in the validation dataset was nearly 97%, the model only found 2 viable sequences with this mutation count during the testing phase. This is most likely because the library EngrAAV was tested on was not designed for the model, meaning many of the sequences were dissimilar to the types of sequences that EngrAAV would predict as viable.

We optimized capsids for the function of viral assembly, but, in future experiments, we hope to apply EngrAAV to optimize other functions of viral vectors, such as immune response evasion or liver de-targeting. Overall, it is clear that EngrAAV can predict capsid viability excellently and can design viable AAV capsids. These results were achieved promptly, and EngrAAV was developed without cost.

**Author Contribution:** AP and NN both contributed equally to EngrAAV development, analysis, and the manuscript.

**Acknowledgement:** The authors would like to thank Dr. Hiroyuki Nakai for providing guidance and resources. They would also like to thank Oregon Health Science University for providing research experience and garnering the knowledge needed to complete this project.

## 5. REFERENCES


*Aav serotypes and AAV tissue-specific tropism*. GeneMedi. (n.d.). Retrieved January 20, 2023, from https://www.genemedi.net/i/aav-serotypes-tissue-specific-tropism

Bryant, D. H., Bashir, A., Sinai, S., Jain, N. K., Ogden, P. J., Riley, P. F., Church, G. M., Colwell, L. J., & Kelsic, E. D. (2021, February 11). *Deep diversification of an AAV capsid protein by machine learning*. Nature News. Retrieved January 20, 2023, from https://www.nature.com/articles/s41587-020-00793-4

Center for Biologics Evaluation and Research. (n.d.). *Luxturna Home Page*. U.S. Food and Drug Administration. Retrieved January 20, 2023, from https://www.fda.gov/vaccines-blood-biologics/cellular-gene-therapy-products/luxturna

Commissioner, O. of the. (n.d.). *FDA approves novel gene therapy to treat patients with a rare form of inherited vision loss*. U.S. Food and Drug Administration. Retrieved January 20, 2023, from https://www.fda.gov/news-events/press-announcements/fda-approves-novel-gene-therapy-treat-patients-rare-form-inherited-vision-loss#:~:text=Luxturna%20is%20approved%20for%20the,complete%20blindness%20in%20certain%20patients.

Crooks, G. E. (n.d.). · *about* · *CREATE* · *examples* ·. WebLogo. Retrieved January 20, 2023, from https://weblogo.berkeley.edu/logo.cgi

*Directed evolution: Methodologies and applications*. (n.d.). Retrieved January 21, 2023, from https://pubs.acs.org/doi/10.1021/acs.chemrev.1c00260

*Heatmap*. Optimizely. (n.d.). Retrieved January 20, 2023, from https://www.optimizely.com/optimization-glossary/heatmap/

*Improving generalizability of protein sequence models with data ...* (n.d.). Retrieved January 21, 2023, from https://ml4molecules.github.io/papers2020/ML4Molecules_2020_paper_14.pdf

Key, M. (2012, November 5). *A tutorial in displaying mass spectrometry-based proteomic data using heat maps - BMC Bioinformatics*. BioMed Central. Retrieved January 20, 2023, from https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-13-S16-S10

Korendovych, I. V. (2018). *Rational and Semirational Protein Design*. Methods in molecular biology (Clifton, N.J.). Retrieved January 20, 2023, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5912912/

Lee, E. J., Guenther, C. M., & Suh, J. (2018, September). *Adeno-associated virus (AAV) vectors: Rational design strategies for Capsid Engineering*. Current opinion in biomedical engineering. Retrieved January 20, 2023, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6516759/

Mahmood, H. (2018, November 26). *Softmax function, simplified*. Medium. Retrieved January 20, 2023, from https://towardsdatascience.com/softmax-function-simplified-714068bf8156

Minot, M., & Reddy, S. T. (2022, January 1). *Nucleotide augmentation for machine learning-guided protein engineering*. bioRxiv. Retrieved January 20, 2023, from https://www.biorxiv.org/content/10.1101/2022.03.08.483422v1.full

Morgunov, A. (2022, November 14). *Keras Tuner: Lessons learned from tuning hyperparameters of a real-life deep learning model*. neptune.ai. Retrieved January 20, 2023, from https://neptune.ai/blog/keras-tuner-tuning-hyperparameters-deep-learning-model

*Multiple sequence alignment*. Multiple Sequence Alignment - an overview | ScienceDirect Topics. (n.d.). Retrieved January 20, 2023, from https://www.sciencedirect.com/topics/medicine-and-dentistry/multiple-sequence-alignment

Naso, M. F., Tomkowicz, B., Perry, W. L., & Strohl, W. R. (2017, August). *Adeno-associated virus (AAV) as a vector for gene therapy*. BioDrugs : clinical immunotherapeutics, biopharmaceuticals and gene therapy. Retrieved January 20, 2023, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5548848/

Packer, M. S., & Liu, D. R. (2015, June 9). *Methods for the directed evolution of proteins*. Nature News. Retrieved January 20, 2023, from https://www.nature.com/articles/nrg3927

Qu, Y., Liu, Y., Noor, A. F., Tran, J., & Li, R. (2019, June). *Characteristics and advantages of adeno-associated virus vector-mediated gene therapy for Neurodegenerative Diseases*. Neural regeneration research. Retrieved January 20, 2023, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6404499/

*Random mutagenesis*. Random Mutagenesis - an overview | ScienceDirect Topics. (n.d.). Retrieved January 20, 2023, from https://www.sciencedirect.com/topics/engineering/random-mutagenesis

Saxena, S. (2021, February 6). *Understanding embedding layer in Keras*. Medium. Retrieved January 20, 2023, from https://medium.com/analytics-vidhya/understanding-embedding-layer-in-keras-bbe3ff1327ce

Shen, H., Price, L. C., Bahadori, M. T., & Seeger, F. (2020, September 28). *Improving generalizability of protein sequence models via data...* OpenReview. Retrieved January 20, 2023, from https://openreview.net/forum?id=Kkw3shxszSd

Shen, H., Price, L. C., Bahadori, M. T., & Seeger, F. (n.d.). *Improving generalizability of protein sequence models with data augmentations*. Amazon Science. Retrieved January 20, 2023, from https://www.amazon.science/publications/improving-generalizability-of-protein-sequence-models-with-data-augmentations

Sinai, S., Jain, N., Church, G. M., & Kelsic, E. D. (2021, January 1). *Generative Aav capsid diversification by Latent Interpolation*. bioRxiv. Retrieved January 20, 2023, from https://www.biorxiv.org/content/10.1101/2021.04.16.440236v1

*Softmax function*. DeepAI. (2019, May 17). Retrieved January 20, 2023, from https://deepai.org/machine-learning-glossary-and-terms/softmax-layer

*A systematic capsid evolution approach performed in vivo for the ... - PNAS*. (n.d.). Retrieved January 21, 2023, from https://www.pnas.org/doi/10.1073/pnas.1910061116

Team, K. (n.d.). *Keras documentation: Embedding layer*. Keras. Retrieved January 20, 2023, from https://keras.io/api/layers/core_layers/embedding/

Team, K. (n.d.). *Keras Documentation: Hyperband Tuner*. Keras. Retrieved January 20, 2023, from https://keras.io/api/keras_tuner/tuners/hyperband/

U.S. National Library of Medicine. (n.d.). *Guide to using the multiple sequence alignment viewer*. National Center for Biotechnology Information. Retrieved January 20, 2023, from https://www.ncbi.nlm.nih.gov/tools/msaviewer/tutorial1/

Wadekar, S. (2021, January 15). *Hyperparameter tuning in Keras: Tensorflow 2: With Keras Tuner: RandomSearch, Hyperband...* Medium. Retrieved January 20, 2023, from https://medium.com/swlh/hyperparameter-tuning-in-keras-tensorflow-2-with-keras-tuner-random search-hyperband-3e212647778f

Wang, D., Tai, P. W. L., & Gao, G. (2019, February 1). *Adeno-associated virus vector as a platform for gene therapy delivery*. Nature News. Retrieved January 20, 2023, from https://www.nature.com/articles/s41573-019-0012-9

*What are heat maps? guide to heatmaps/how to use them*. Hotjar. (n.d.). Retrieved January 20, 2023, from https://www.hotjar.com/heatmaps/

*Word embeddings : text : tensorflow*. TensorFlow. (n.d.). Retrieved January 20, 2023, from https://www.tensorflow.org/text/guide/word_embeddings#:~:text=The%20Embedding%20layer%20takes%20the,batch%2C%20sequence%2C%20embedding)%20.

Yoon, D. S., Lee, K.-M., Cho, S., Ko, E. A., Kim, J., Jung, S., Shim, J.-H., Gao, G., Park, K. H., & Lee, J. W. (2021, July 25). *Cellular and tissue selectivity of AAV serotypes for gene delivery to chondrocytes and cartilage*. International journal of medical sciences. Retrieved January 20, 2023, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8436087/