

Protein structure-based machine learning for predicting viral gene delivery vehicles

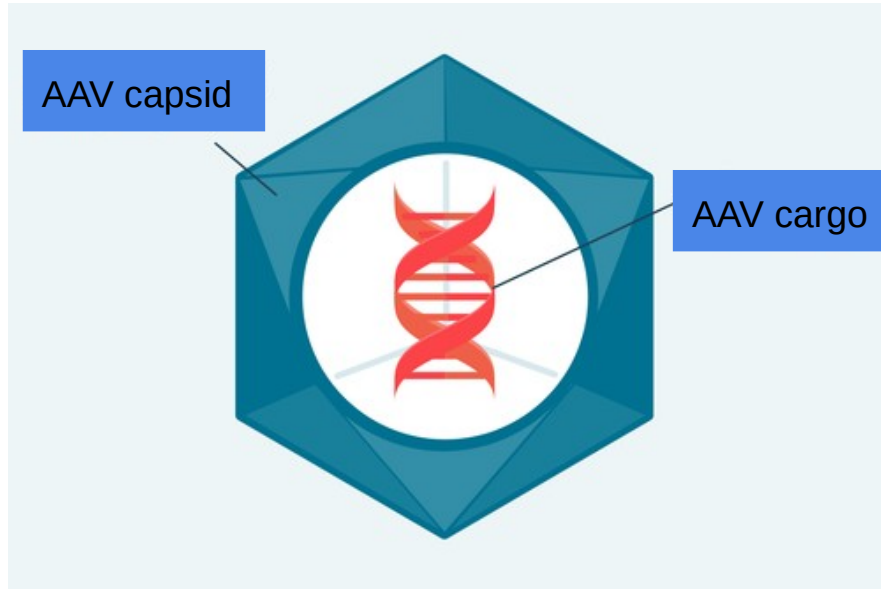
Gradinaru Lab Project
CMS 273

Final Presentation
3/18/2019

Xiaozhe Ding, David Brown, Florian Schaefer, Roshan Bal,
Sri (priya) Ravindra Kumar, Xinhong Chen

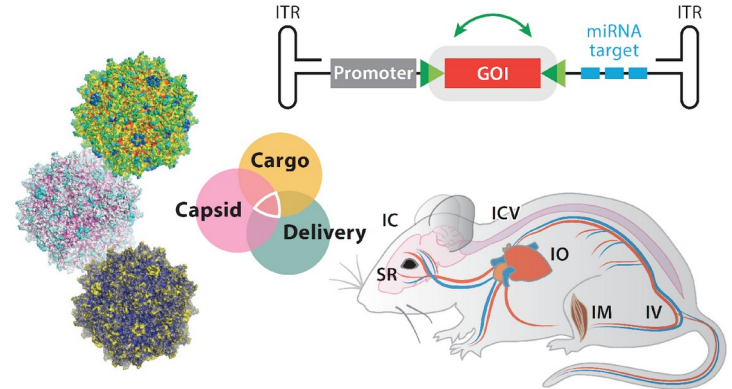
Adeno-associated viruses (AAVs) as vectors for gene delivery

AAVs have broad tropism (dividing and non-dividing cells) and safe for gene delivery.



Viral Tropism is mediated by:

- Capsid (outer protein shell)
- Cargo/Packaged genome (Regulatory elements)
- Route of delivery

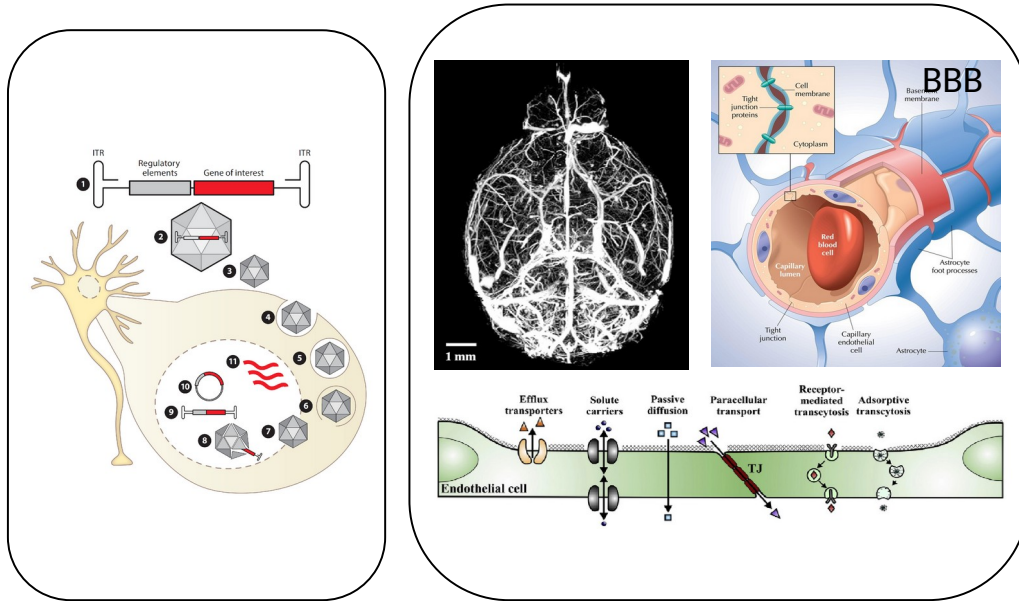


AAVs as vectors for gene delivery to central nervous system (CNS)

AAVs can target cells of the CNS.

Blood-brain barrier (BBB) limits access to CNS.

AAV9 can cross BBB, but inefficiently



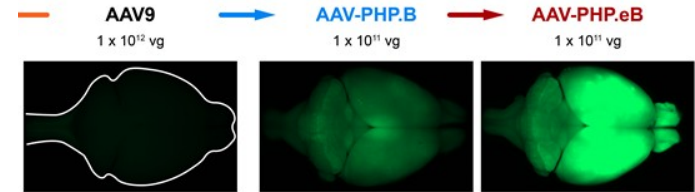
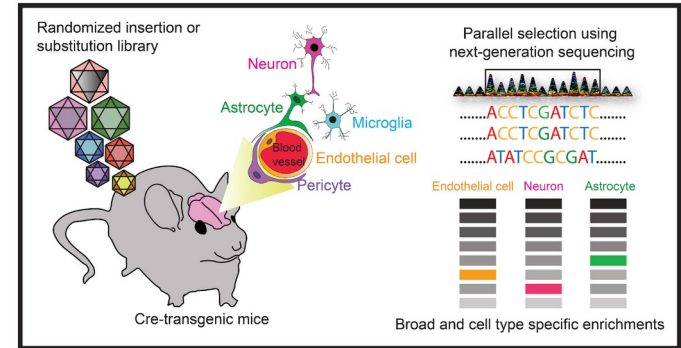
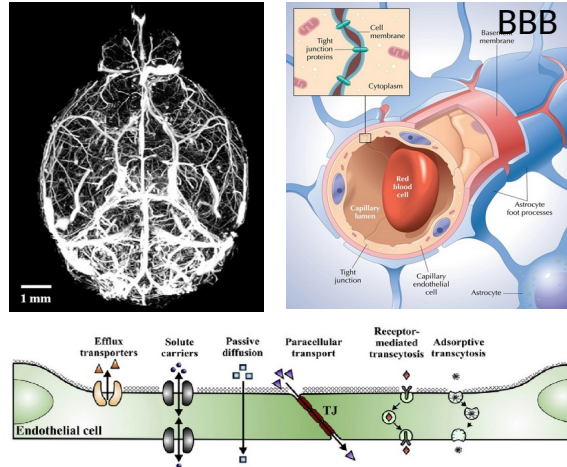
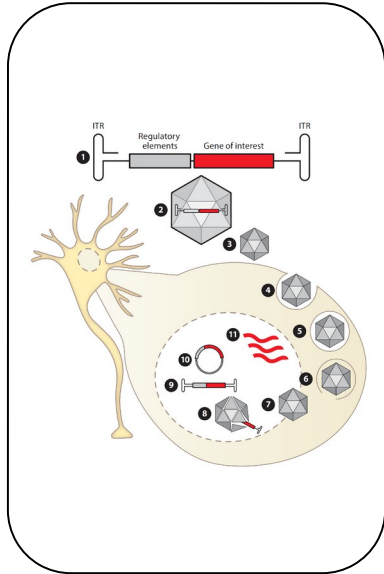
AAVs as vectors for gene delivery to central nervous system (CNS)

AAVs can target cells of the CNS.

Blood-brain barrier (BBB) limits access to CNS.

Engineering AAV9 capsid by directed evolution yields variants that can efficiently cross BBB and target the CNS.

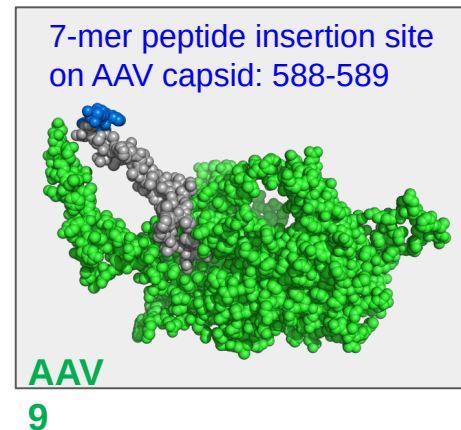
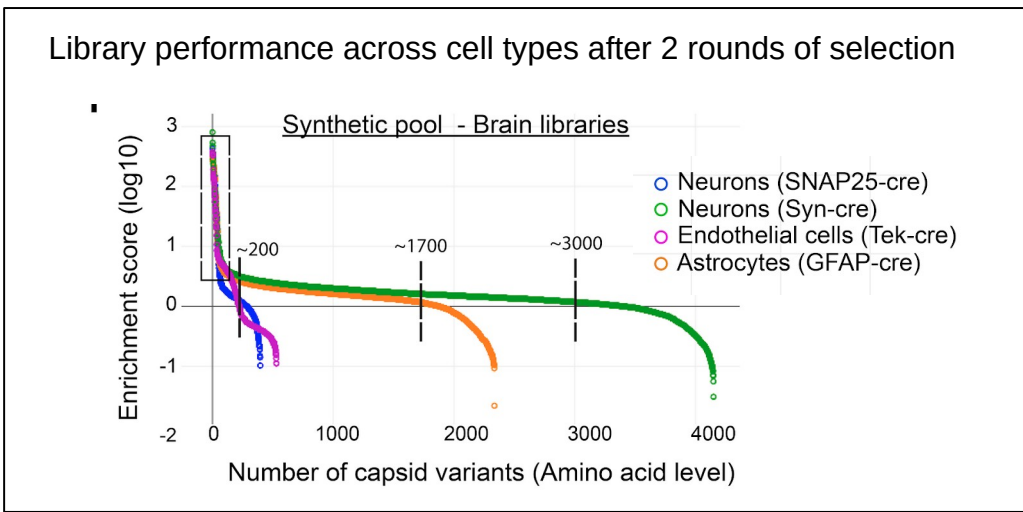
AAV9 can cross BBB, but inefficiently



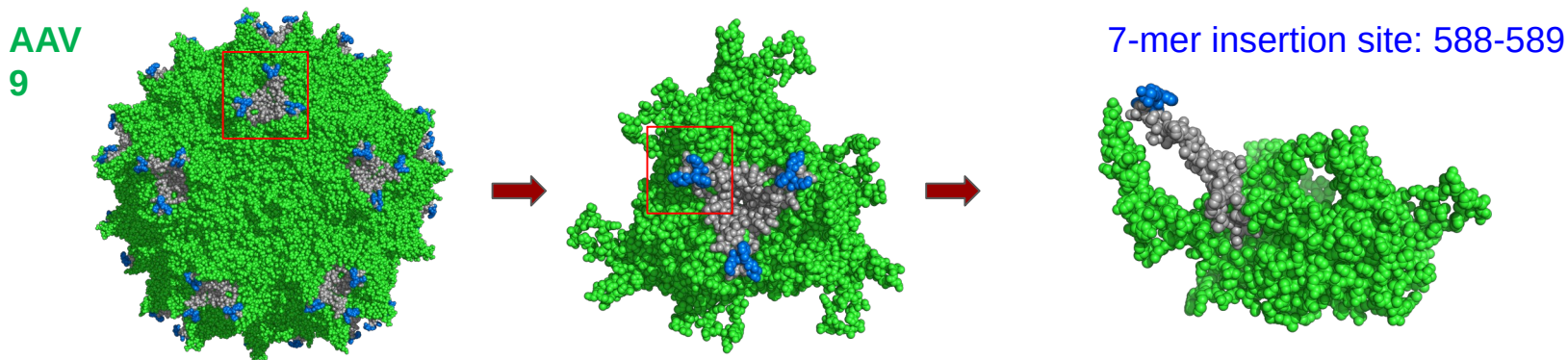
Can we use machine learning to further our chance of finding capsid variants uncovered through experiments?

COVERAGE OF EXPERIMENTS: ~9e3 covered variants out of ~1.3e9 theoretical variants

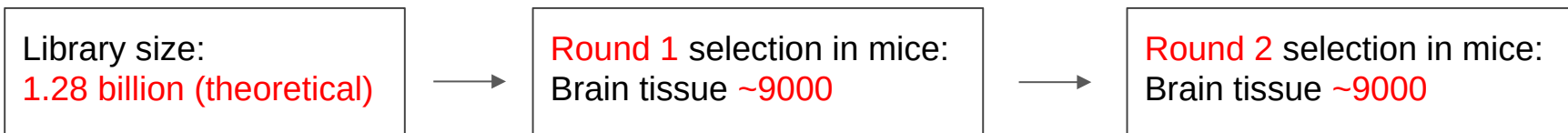
- Selection was performed across
 1. Endothelial cells (cells forming the BBB)
 2. Neurons
 3. Astrocytes
- ~600 out of 9000 were positively enriched in brain tissue across all cell types.



Backup slide: Directed evolution on the surface loop of AAV

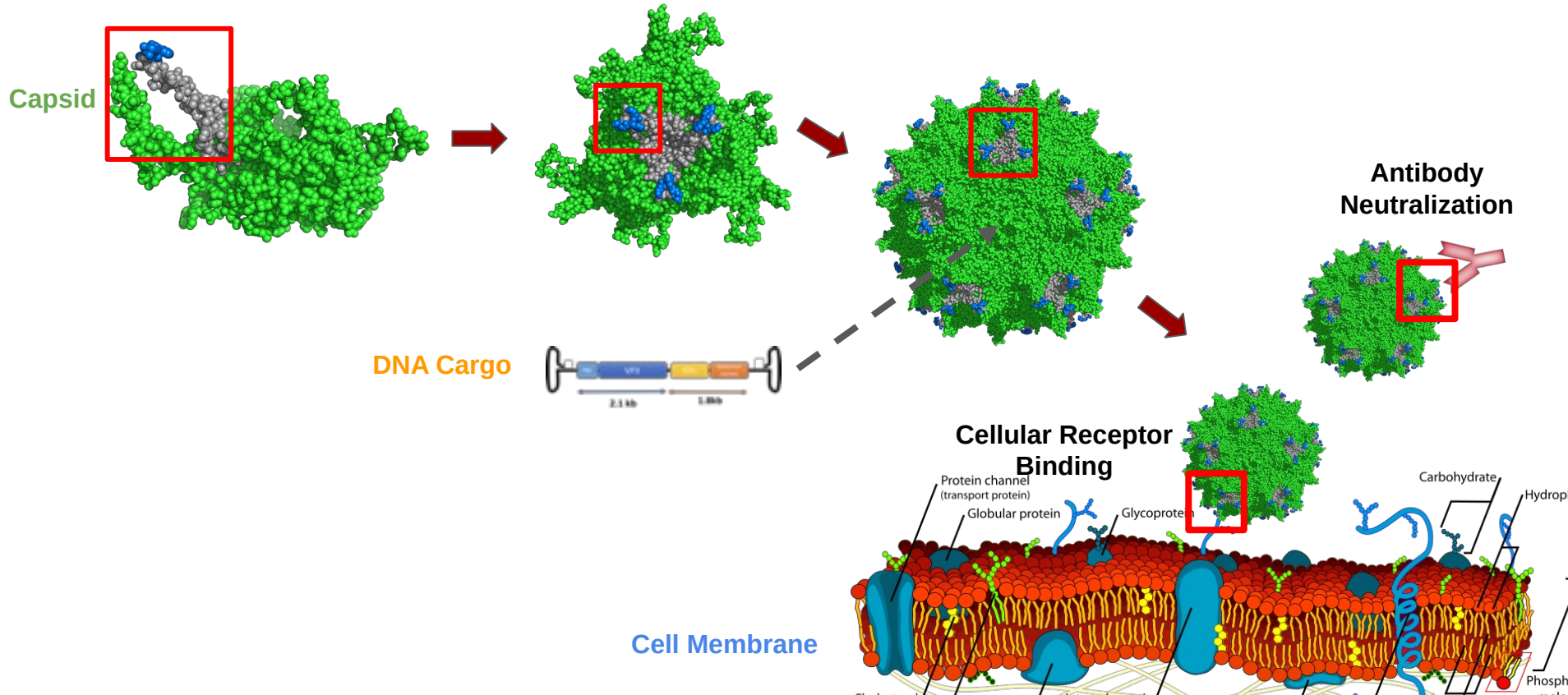


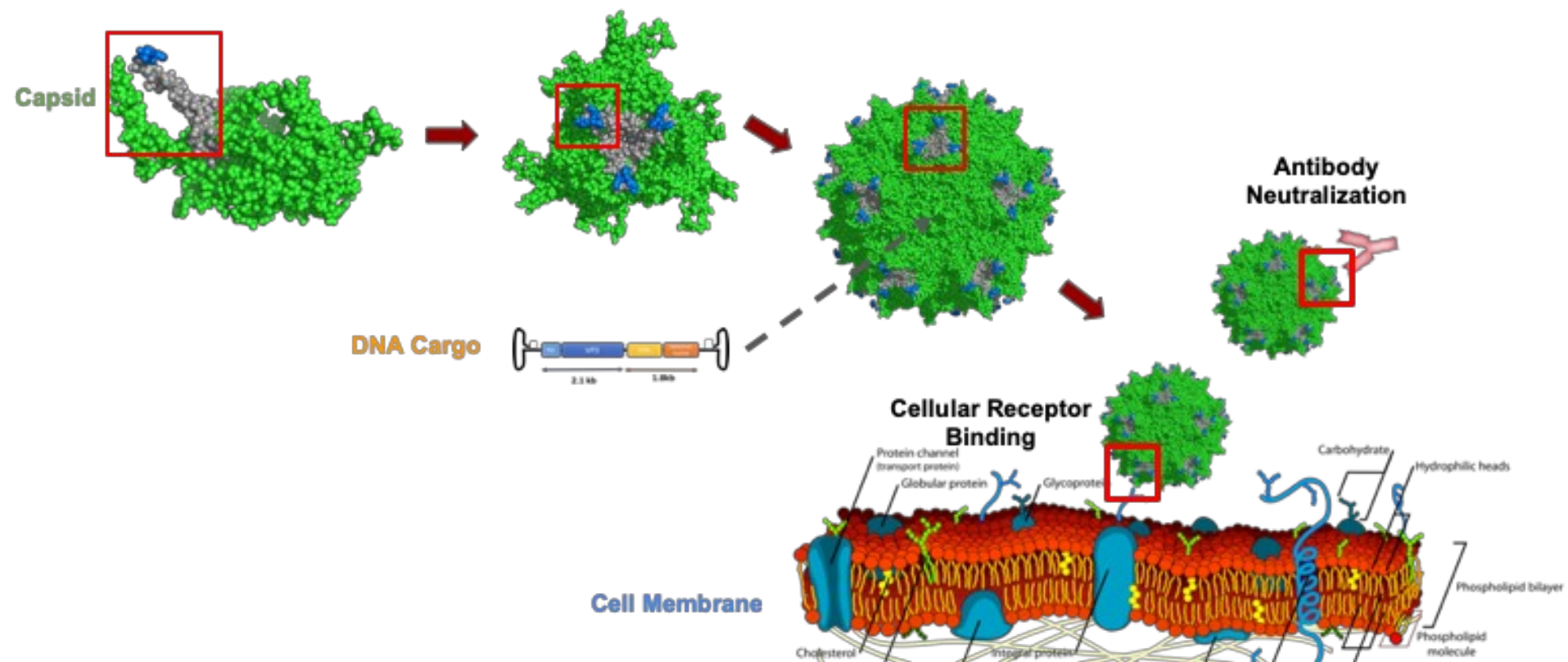
Randomized 7 amino acid peptide insertion on the surface loop of AAV9



Selection from large library size can suffer from insufficient sequence coverage.

Random peptide insertion changes capsid surface structure, which determines virus-host interactions before entering a target cell





Rosetta, a protein modeling software based on physical and statistical knowledge allows prediction of possible conformations of an AAV variant

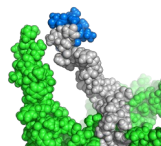
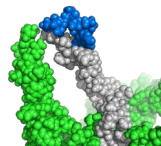
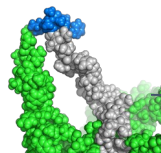
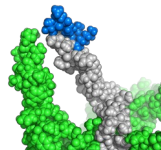
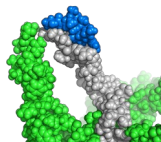
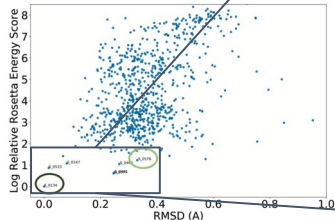
Insertion sequence

AQ**TLAVPFKA**Q

Rosetta



RMSD vs. Log Energy Score



Coordinates of all atoms

1	ATOM	1	N	ASP	A	1	-58.685	33.924	73.962	1.00	0.00
2	ATOM	2	CA	ASP	A	1	-59.234	34.811	75.031	1.00	0.00
3	ATOM	3	C	ASP	A	1	-60.416	34.122	75.728	1.00	0.00
4	ATOM	4	O	ASP	A	1	-61.448	34.751	75.975	1.00	0.00
5	ATOM	5	CB	ASP	A	1	-59.704	36.135	74.408	1.00	0.00
6	ATOM	6	CG	ASP	A	1	-59.505	37.337	75.338	1.00	0.00
7	ATOM	7	OD1	ASP	A	1	-58.362	37.840	75.440	1.00	0.00
8	ATOM	8	OD2	ASP	A	1	-60.491	37.780	75.970	1.00	0.00
9	ATOM	9	1H	ASP	A	1	-57.916	34.379	73.513	1.00	0.00
10	ATOM	10	2H	ASP	A	1	-58.372	33.066	74.368	1.00	0.00
11	ATOM	11	3H	ASP	A	1	-59.398	33.733	73.288	1.00	0.00
12	ATOM	12	HA	ASP	A	1	-58.447	34.999	75.762	1.00	0.00
13	ATOM	13	1HB	ASP	A	1	-59.157	36.316	73.483	1.00	0.00
14	ATOM	14	2HB	ASP	A	1	-60.762	36.062	74.154	1.00	0.00
15	ATOM	15	N	GLY	A	2	-60.270	32.836	76.049	1.00	0.00
16	ATOM	16	CA	GLY	A	2	-61.359	32.111	76.688	1.00	0.00
17	ATOM	17	C	GLY	A	2	-61.013	31.375	77.968	1.00	0.00
18	ATOM	18	O	GLY	A	2	-60.023	31.685	78.629	1.00	0.00
19	ATOM	19	H	GLY	A	2	-59.405	32.352	75.853	1.00	0.00
20	ATOM	20	1HA	GLY	A	2	-62.167	32.803	76.926	1.00	0.00
21	ATOM	21	2HA	GLY	A	2	-61.761	31.374	75.994	1.00	0.00
22	ATOM	22	N	VAL	A	3	-61.834	30.387	78.314	1.00	0.00

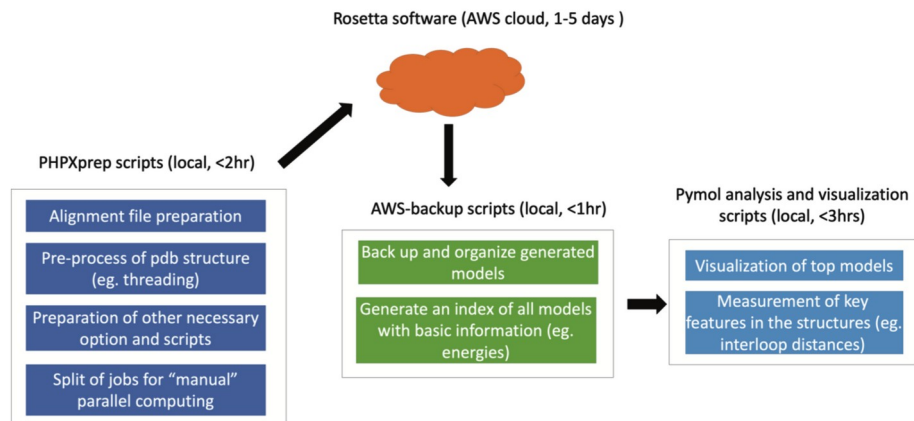
+

Energy scores (total and compositions)

[illegible]

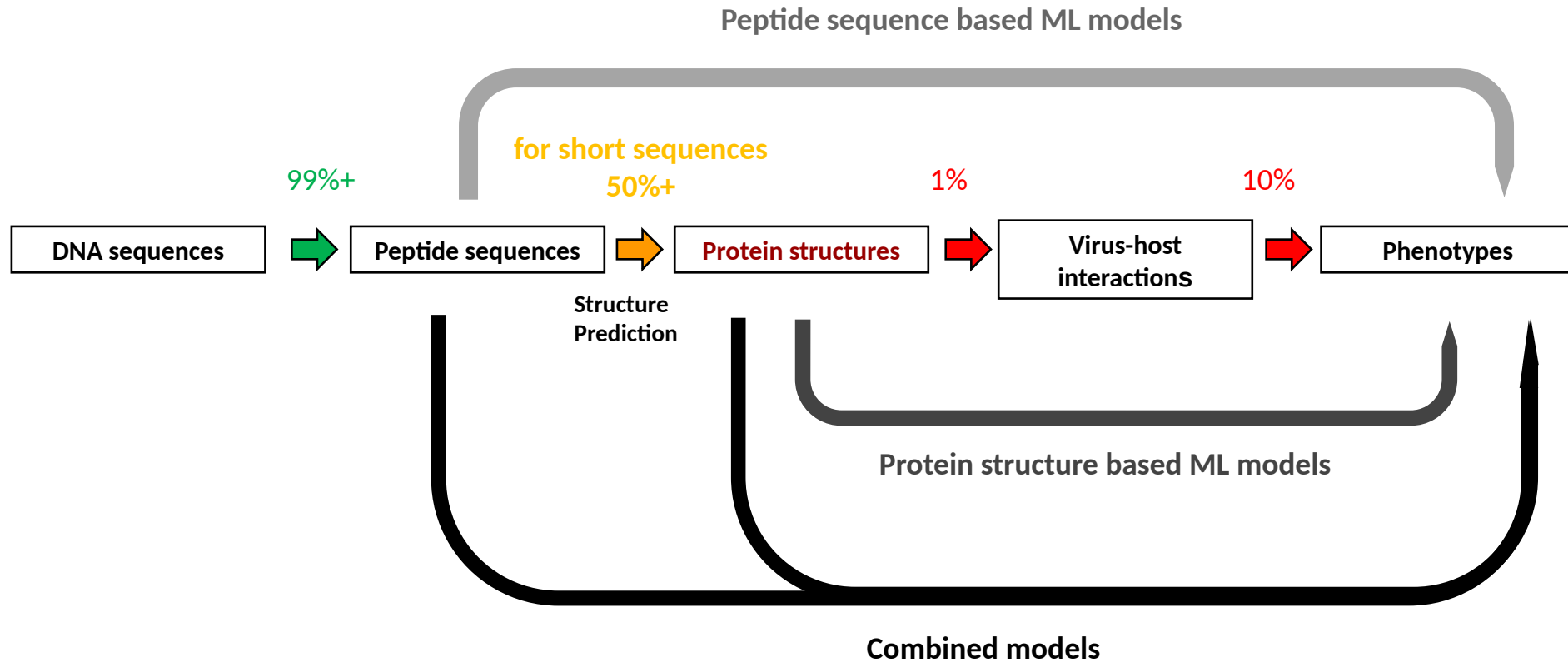
The structure prediction method can be scaled up and generate structural information for thousands of AAV variants

A pipeline for high-throughput structure prediction on the cloud

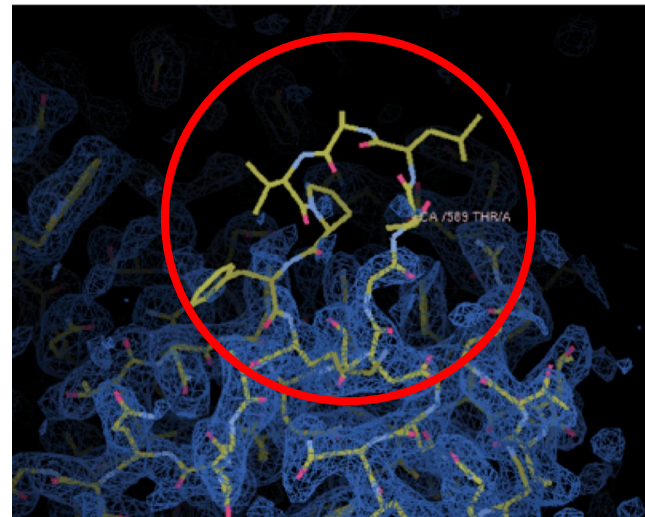
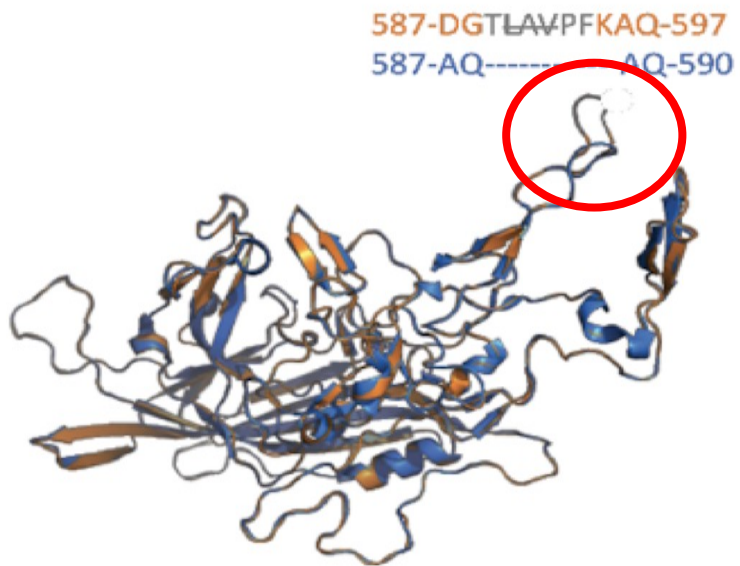


Average computational cost: 1-3 CPU*hr / pose

Exploring incorporation of structural information into machine learning models

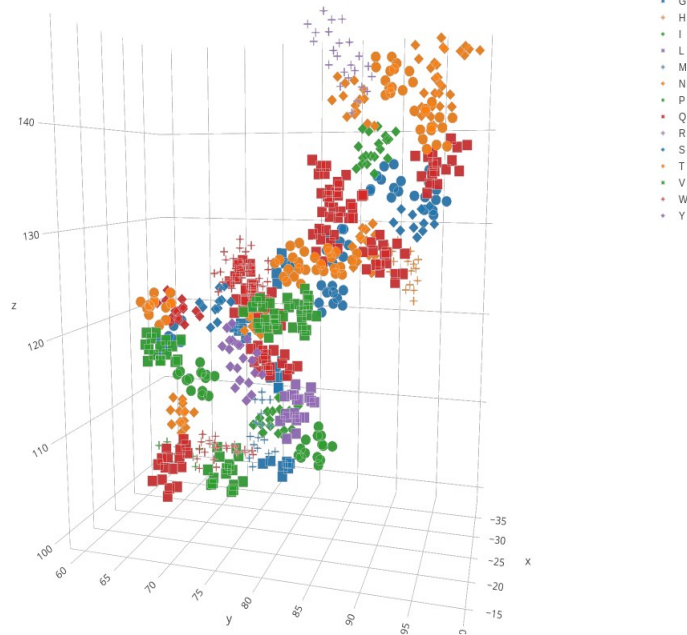


Backup slide: Structural changes introduced in our AAV libraries are restricted to a small, local region that is feasible for relatively accurate structure prediction



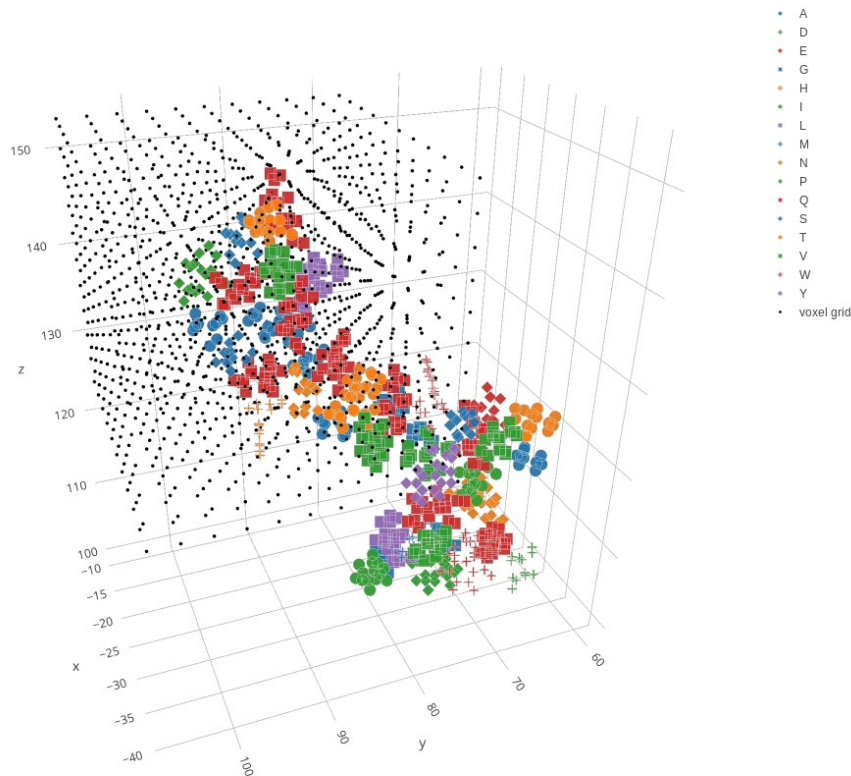
How to incorporate structural information

- For each variant (sequence): multiple poses (conformations) and their energies
- Position of each of the atoms, their amino-acid membership, and chemical properties
- Only (neighbourhood of) modified region is assumed to vary

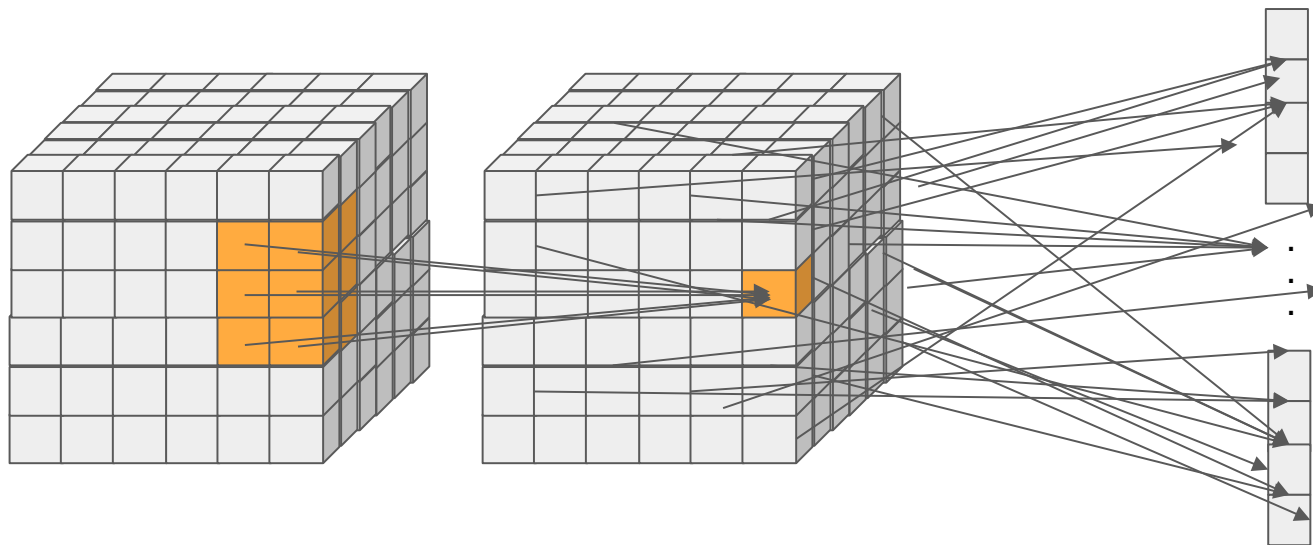


Voxelization

- Treat geometry as a 3-d image.
- $n_x * n_y * n_z$
“3d-pixel” (voxel)
- $n_c = 26$ different “colors” (channels),
 - Abundance of atoms of the 20 different AA
 - Abundance of atoms of AA with 6 different physical properties



A simple CNN



$(n_x, n_y, n_z, 26)$

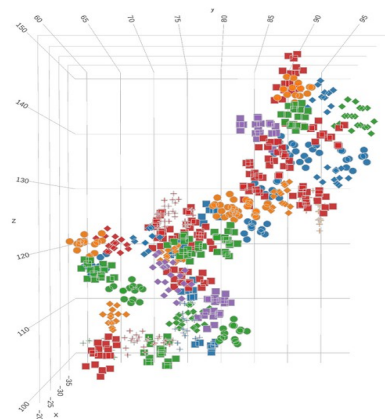
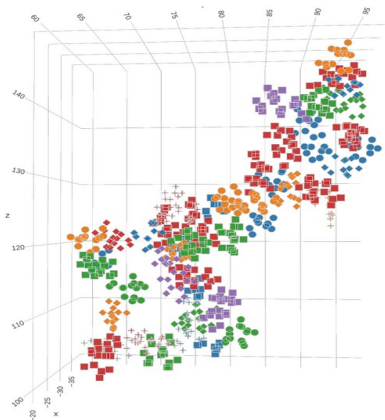
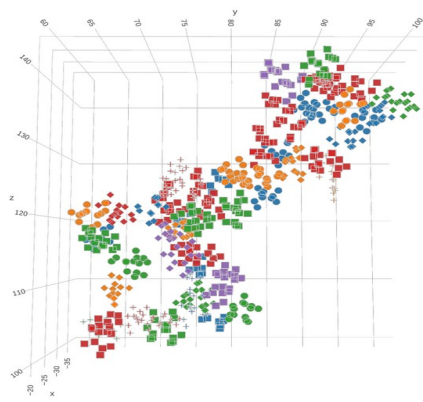


$(n_x, n_y, n_z, 20)$



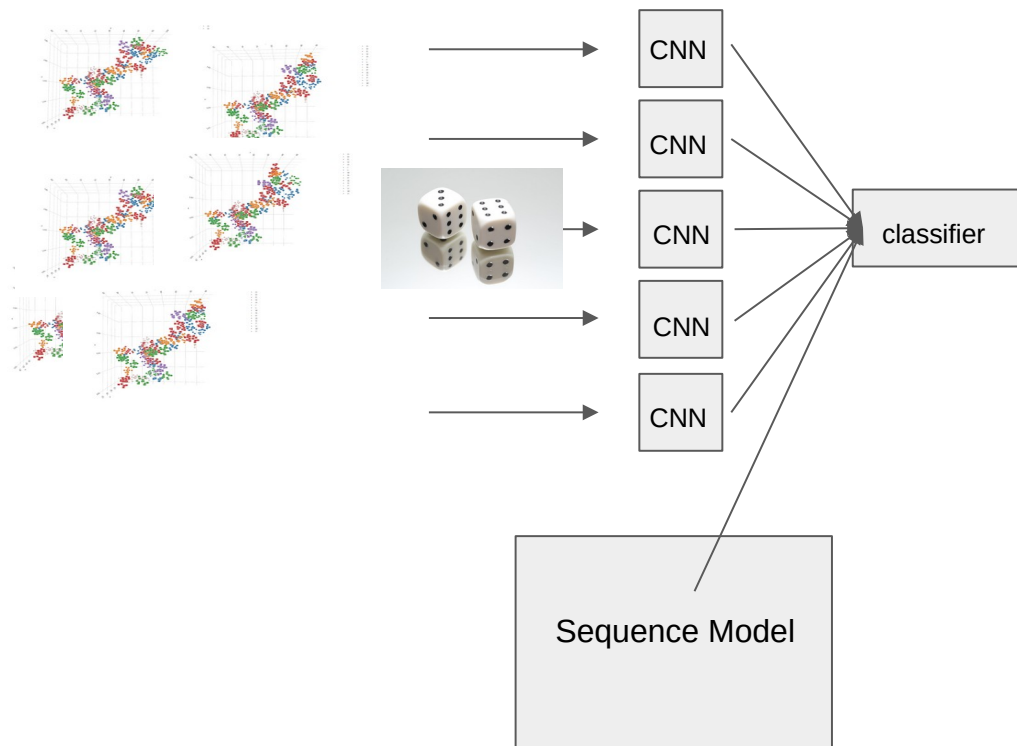
(N)

How to deal with different poses?



A
D
E
G
H
I
L
M
N
O
P
Q
S
T
V
W
Y

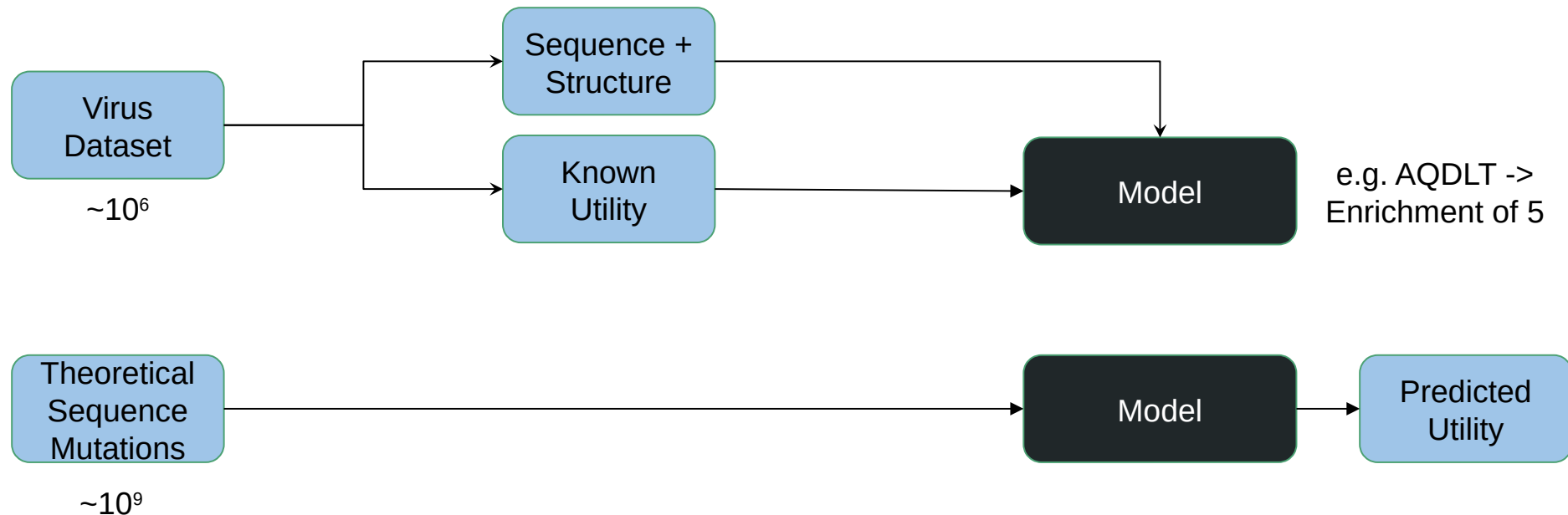
How to deal with different poses?



- Poses are independently sampled with replacement
- An identical CNN is applied individually to each pose
- The outputs are combined with the sequence based model and fed into a classifier
- Advantage: Network is trained on many “realistic” sample poses.

Metrics

The Framework



Metrics - Accuracy is not a good estimator

Goal: to generate a list of the **best** sequences

Sequence	Score
Sequence 1	0.99
Sequence 2	0.98
Sequence 3	0.95
Sequence 4	0.94
Sequence 5	0.92
...	...
Sequence 9999995	0.34
Sequence 9999996	0.33
Sequence 9999997	0.33
Sequence 9999998	0.32
Sequence 9999999	0.31

Metrics - Accuracy is not a good estimator

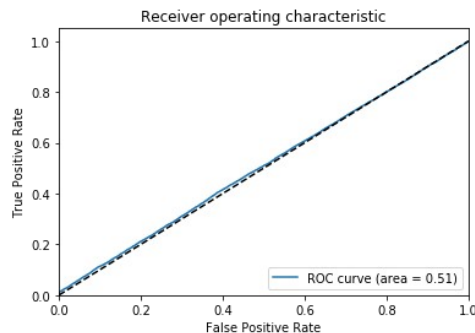
Goal: to generate a list of the **best** sequences

Sequence	Score
Sequence 1	0.99
Sequence 2	0.98
Sequence 3	0.95
Sequence 4	0.94
Sequence 5	0.92
...	...
Sequence 9999995	0.34
Sequence 9999996	0.33
Sequence 9999997	0.33
Sequence 9999998	0.32
Sequence 9999999	0.31

If we go make **just** these, how will they do?

ROC curves and accuracy can be misleading

- 1,000,000 sequences
- Model predicts the top 100 sequences with 100% accuracy
- Predictions on the rest are random



Metrics - Precision at K

Goal: to generate a list of the **best** sequences

Method:

- Split the data into high vs low performers (imbalanced classes)
- Look at precision at k, where $k/\text{test_size} \ll 0.5$
- Analogous to:
 - Amazon product suggestions
 - Recommended search results
 - Etc...

Metrics - Precision at K

Goal: to generate a list of the **best** sequences

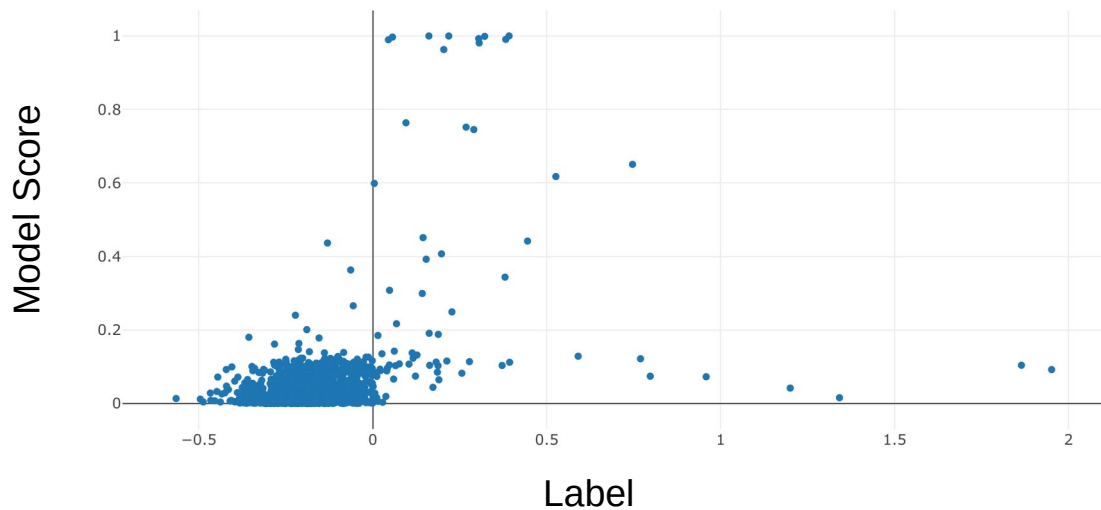
Method:

- Split the data into high vs low performers (imbalanced classes)
- Look at precision at k, where $k/\text{test_size} \ll 0.5$
- Analogous to:
 - Amazon product suggestions
 - Recommended search results
 - Etc...

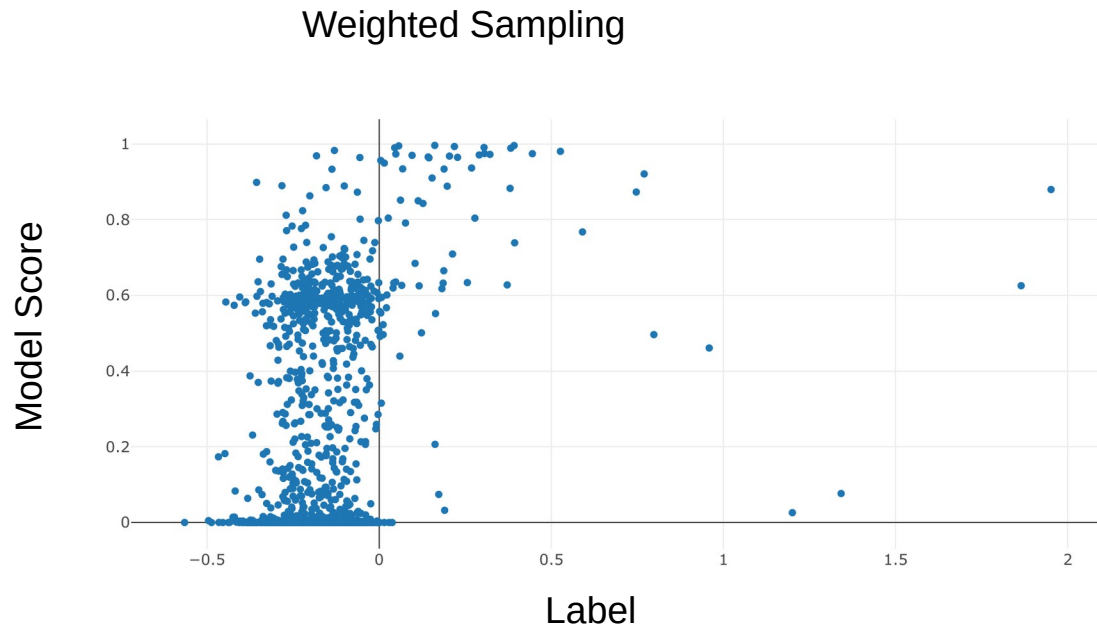
Sequence	Score	True Utility
Sequence 1	0.99	2.42
Sequence 2	0.98	-1.21
Sequence 3	0.95	1.1
Sequence 4	0.94	0.5
Sequence 5	0.92	0.9
...	...	
Sequence 9995	0.34	-1.9
Sequence 9996	0.33	0.1

Weighting Training Examples

Unweighted Sampling



Weighting Training Examples

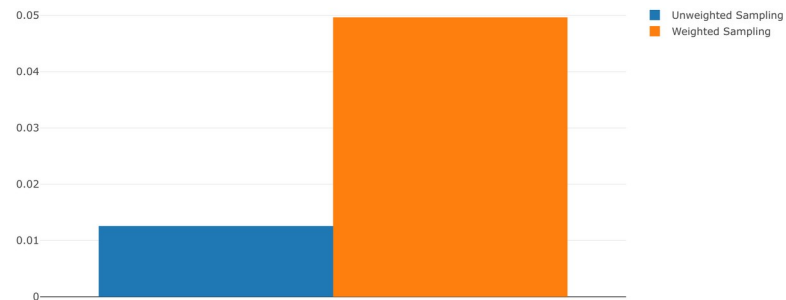


Effect of weighting training examples

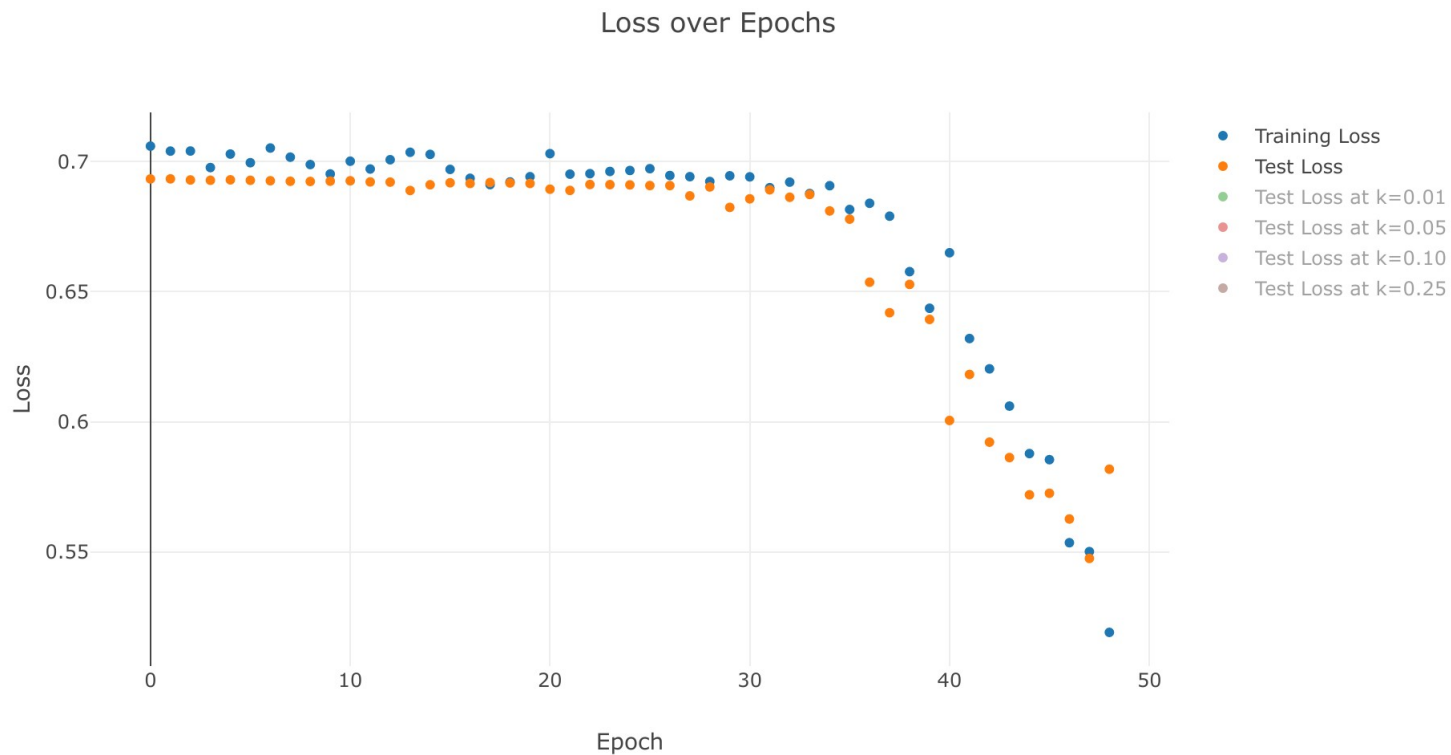
Precisions at 0.05



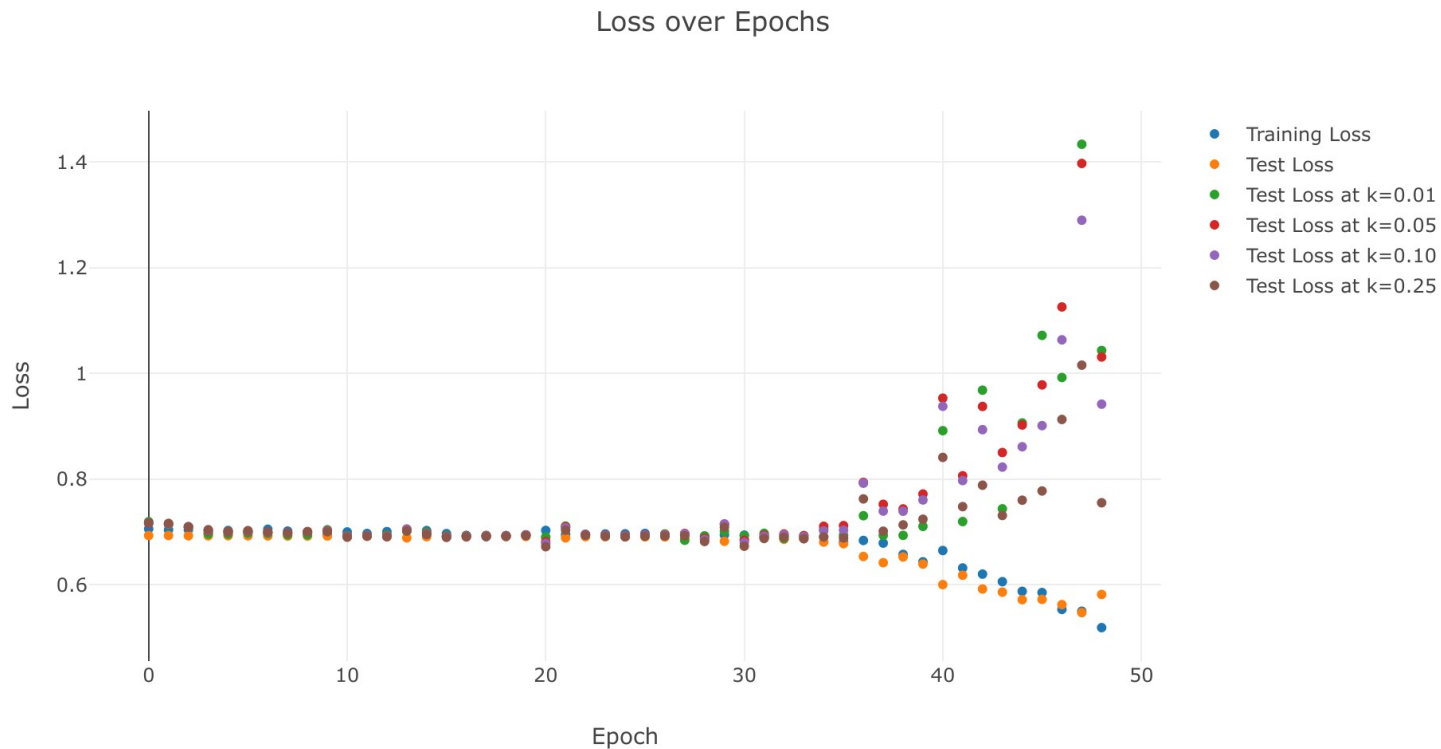
Mean Label at 0.05



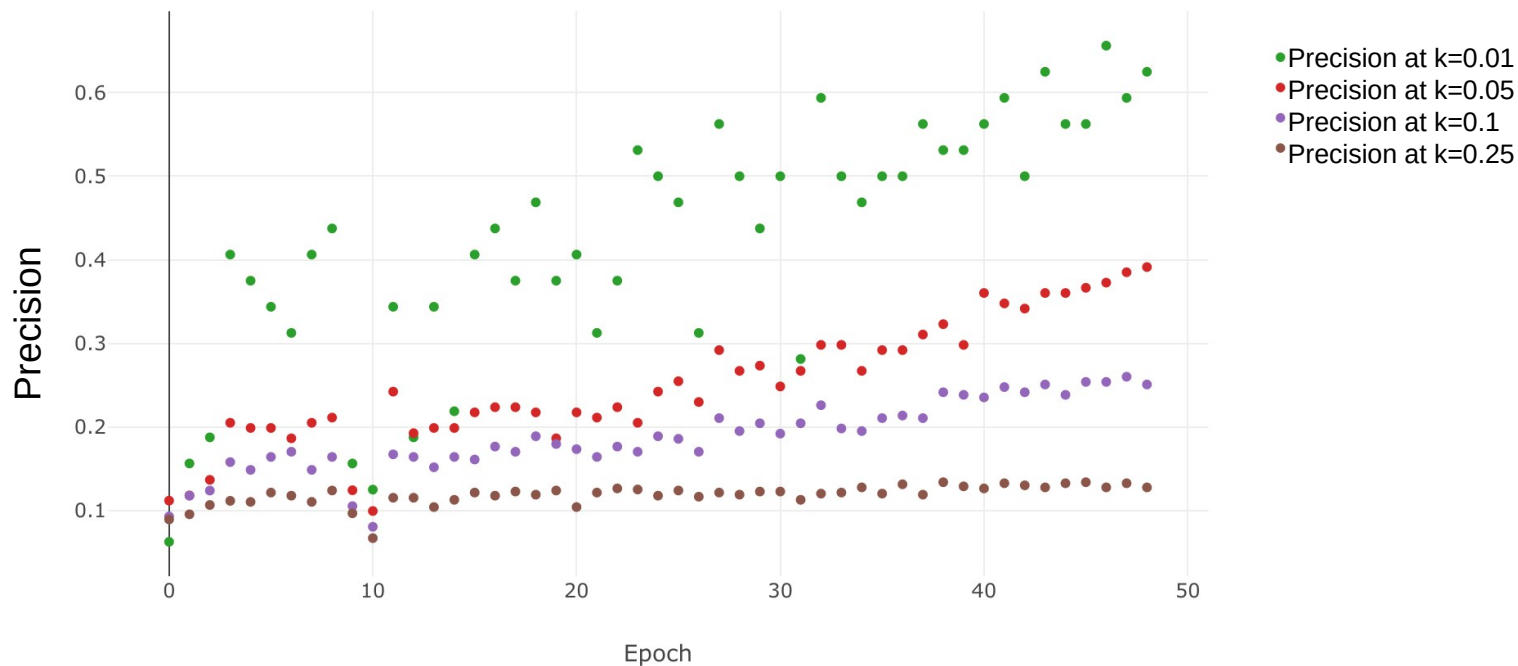
Loss can be misleading...



Loss can be misleading...

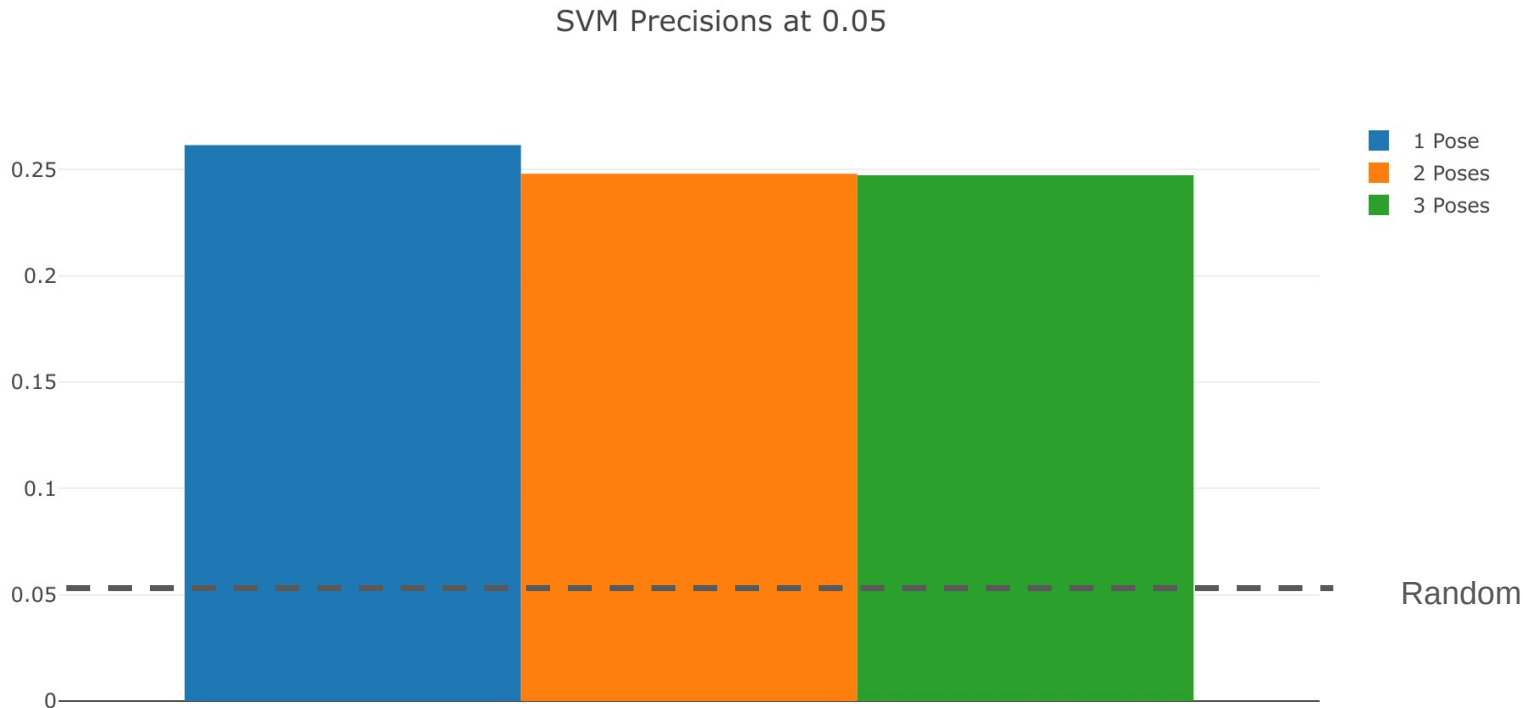


Loss and precision at k can be opposing

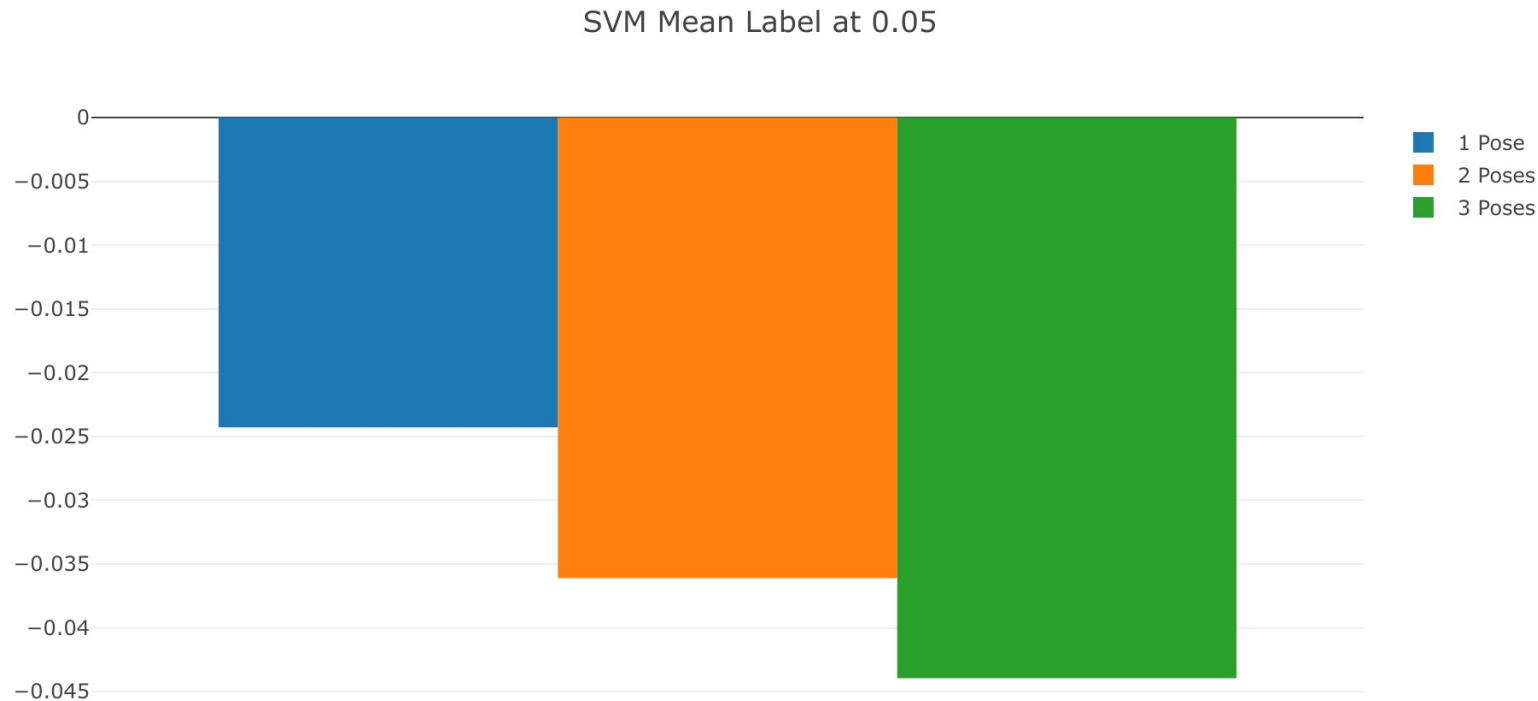


Model Comparisons

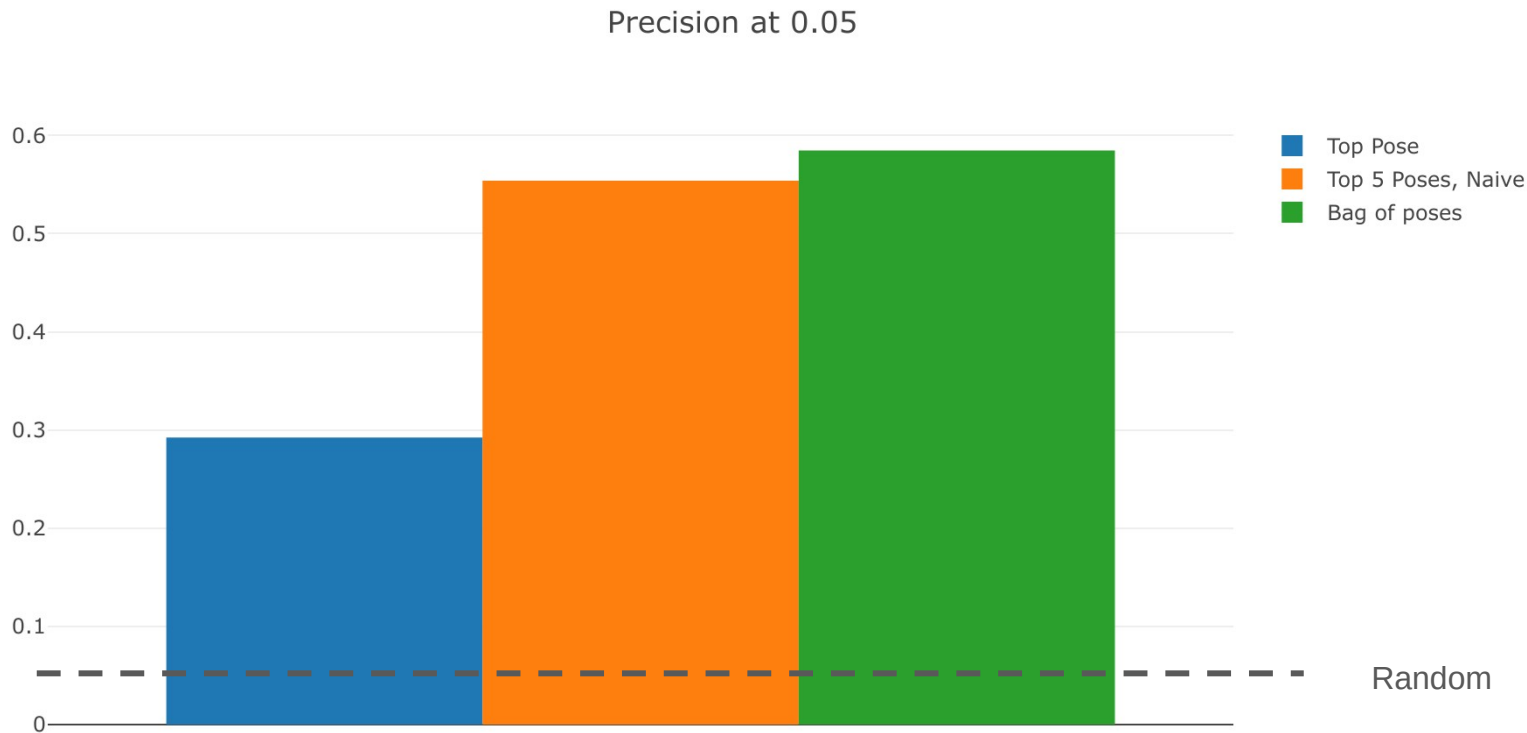
Naively adding poses hurts performance in SVM



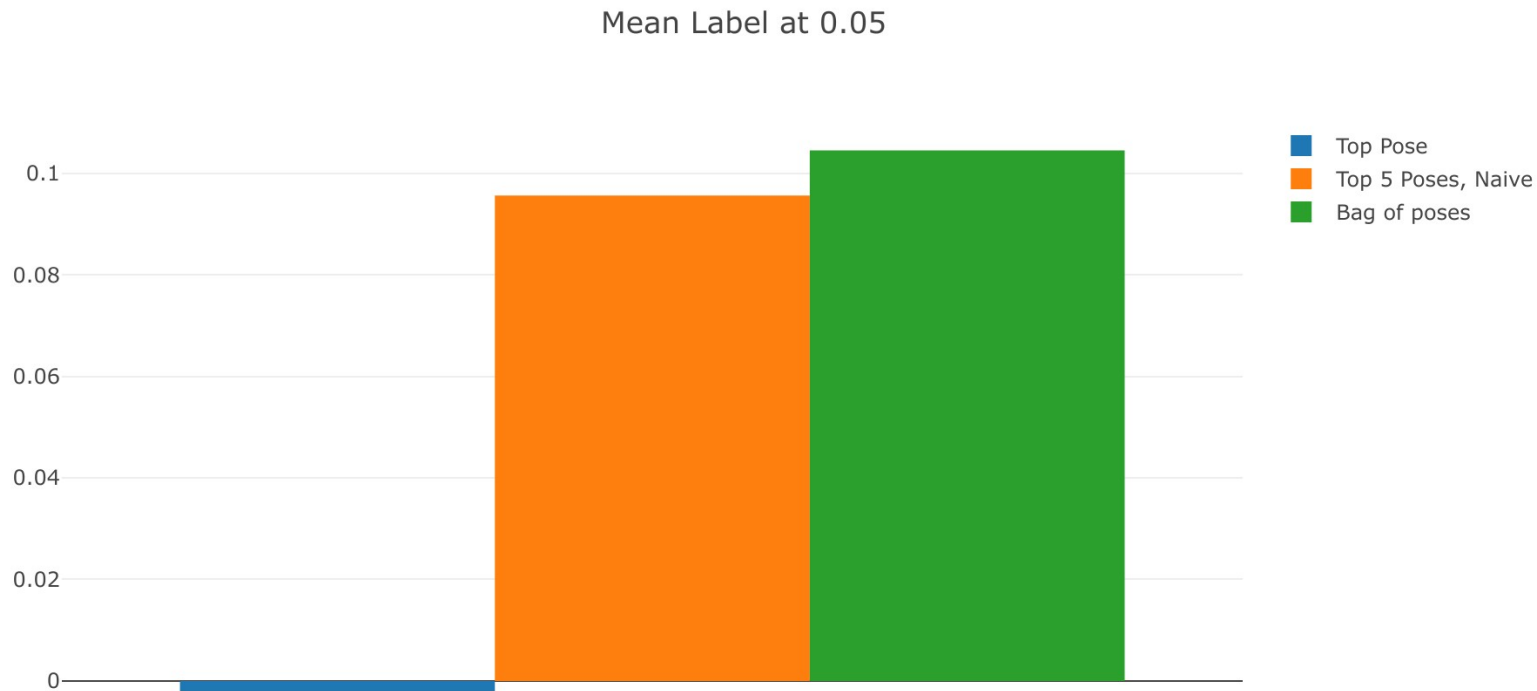
Naively adding poses hurts performance in SVM



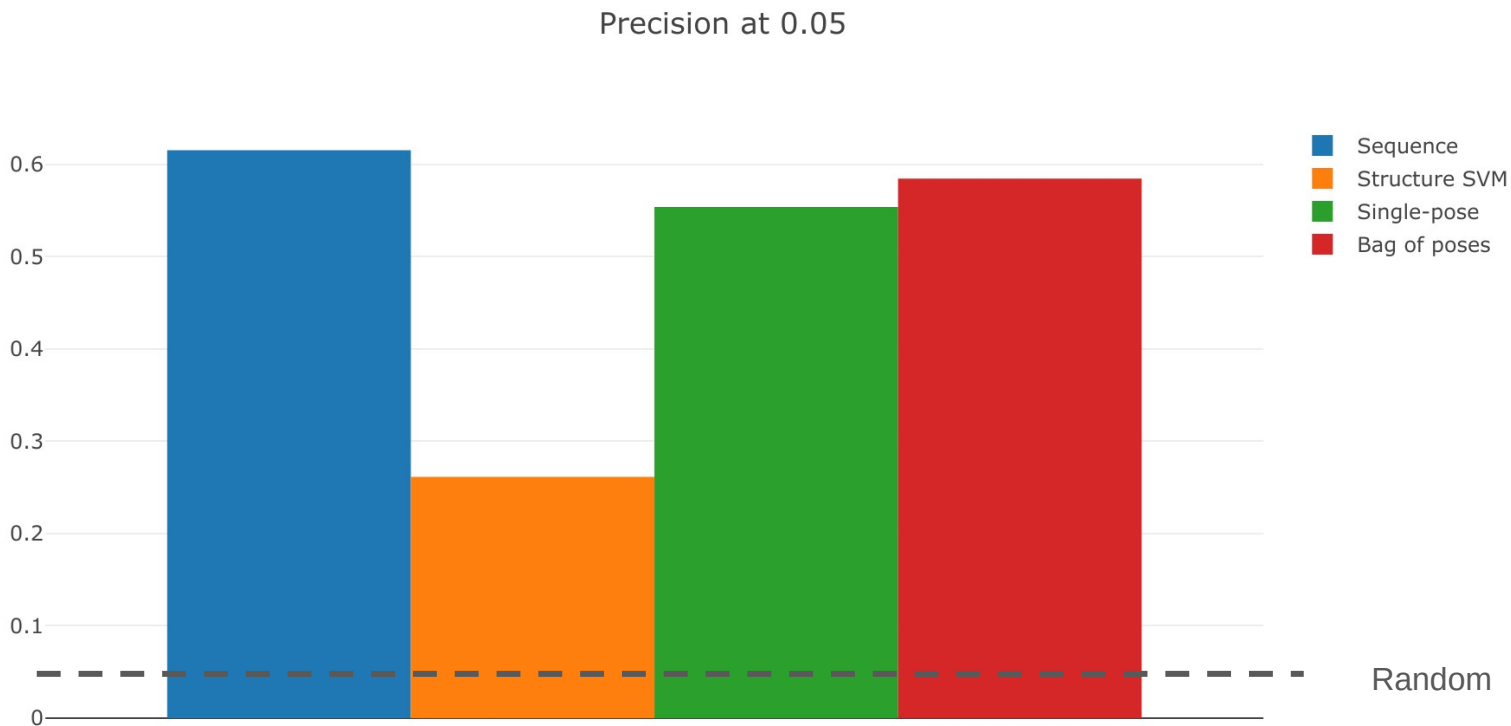
Adding poses helps performance in Conv Net



Adding poses helps performance in Conv Net



Model Comparison



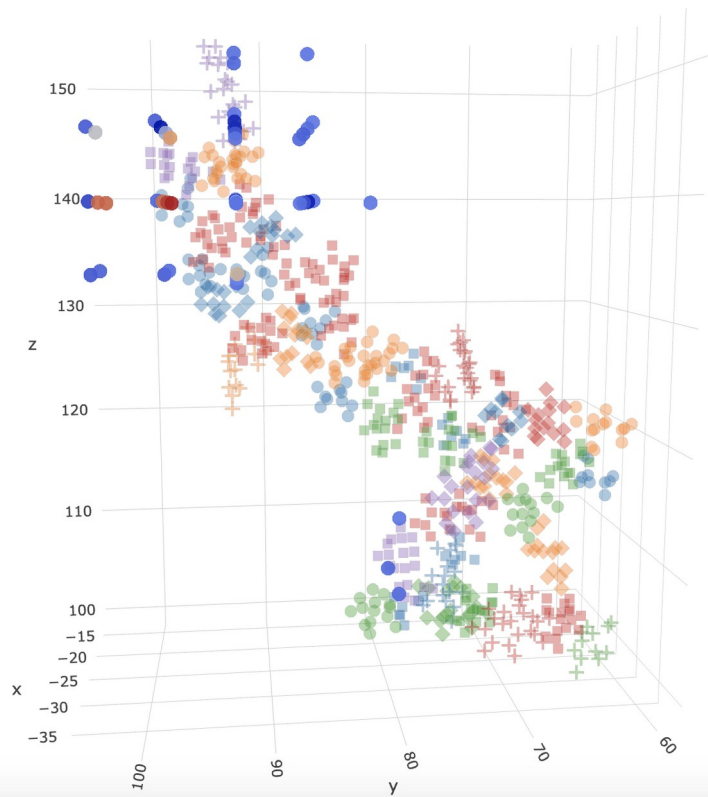
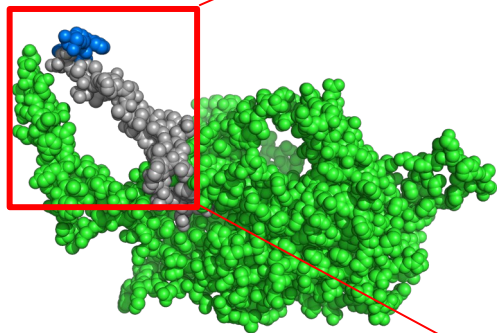
Model Comparison

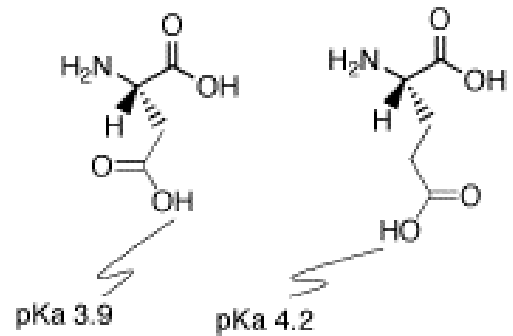
Mean Label at 0.05



Visualization of interpretable model (Roshan)

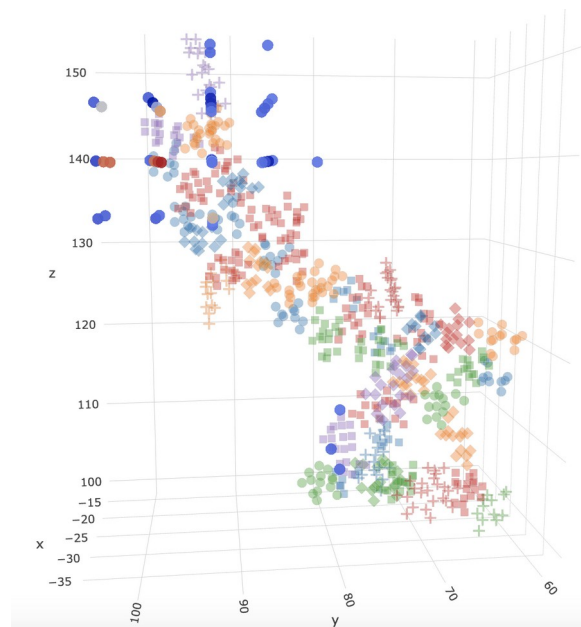
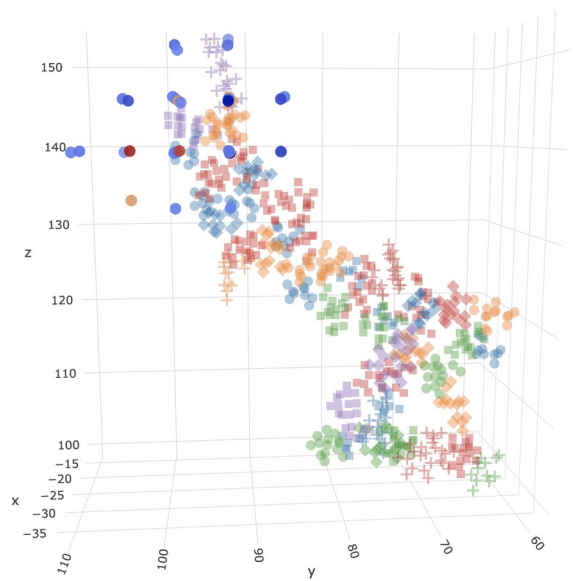
1. Visualization of interpretable model: SVM (One way to use the results) —> most protruding part

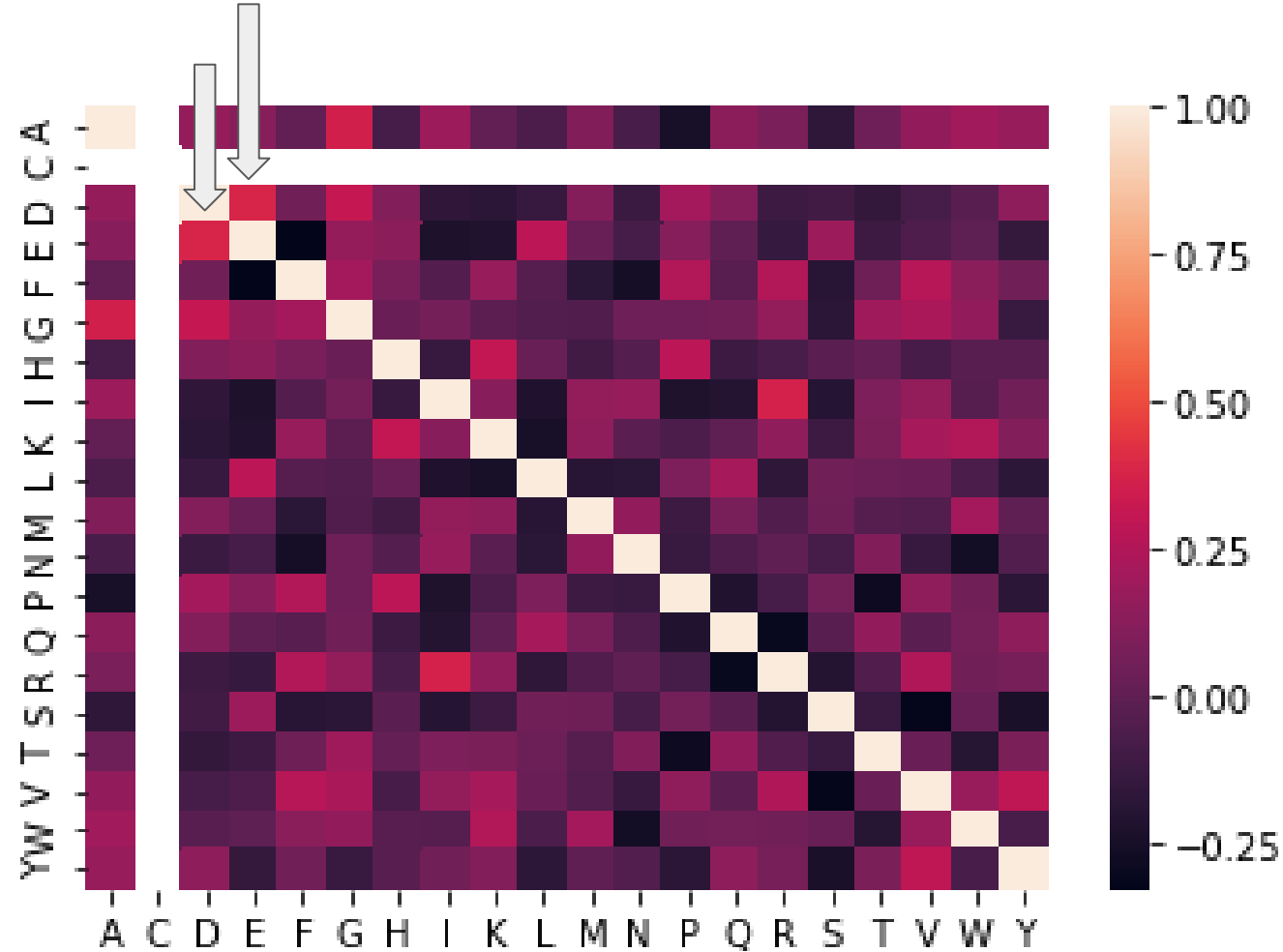




aspartic acid
Asp
D

glutamic acid
Glu
E





Conclusions

- Models *can* learn the AAV structure-utility function
- Neural networks dramatically outperform linear models
- We need more structural data!
- Top- k performance requires special treatment
- Handling multiple energy poses is an open and interesting problem

Future Directions

- Top- k loss functions
- Add energy to available features
 - Let the network find the importance of different energy poses
- Improve fusion between sequence and structure network
 - Potentially improve diversity of high predictions
 - Attention mechanisms to learn the residual of the sequence network
 - “Adversarial” network to drive learning different subsets of sequences
- More + better data from Rosetta performance improvements
- Explore all labels to find most learnable datasets
- Multi-output network
- Hyperparameter tuning :(

Protein structure-based machine learning for predicting viral gene delivery vehicles

Final project report

Caltech, Gradinaru Lab

Project authors

Xiaozhe Ding, David Brown, Florian Schaefer, Roshan Bal,
Sri (Priya) Ravindra Kumar, Xinhong Chen

Names and order follow the final presentation title slide.

Project development: December 2018-March 2019

Course: Winter 2019 (January-March)

Final presentation: March 18, 2019

Original report PDF generated: March 21, 2019

Protein structure-based machine learning for predicting viral gene delivery vehicles

ABSTRACT

Directed evolution has been widely used for engineering proteins for desired properties. One such example is the evolution of a viral vector for efficient gene delivery to areas that are otherwise inaccessible or limited through naturally existing viruses. Adeno-associated viruses (AAVs) are non-pathogenic, and their existence in host cell as episomal DNA without random integration into host chromosome makes them a safe vector for gene delivery, and hence, have been widely used in both gene therapy and biomedical research. The research in Dr. Viviana Gradinaru's laboratory has focused on using *in vivo* directed evolution to improve the efficiency of AAV towards crossing the blood-brain barrier and targeting the central nervous system. This yielded novel variants AAV-PHP.B and AAV-PHP.eB exhibiting enhanced brain wide transduction in adult rodent upon intravenous delivery.

Despite numerous successes achieved with directed evolution, the efficiency of the method is limited by the inability to cover the entire sequence space of a large randomized library. One possible solution to improve the efficiency of directed evolution is by performing a selection *in silico* using machine learning (ML) models to predict high-quality libraries with sequences with higher probability of having the desired properties. In the past, the Gradinaru Lab has explored training neural network models with data generated from next-generation sequencing experiments. With architecture design and hyperparameter fine tuning, the models were able to predict whether an AAV variant is among the most efficient to cross the BBB with decent precision.

In this report we present a novel method that allows us to incorporate more physical knowledge into the ML models. As the structure of the capsid plays a central role in determining the virus-host interaction and thus the tropism of a virus variant, we hypothesized that feeding structural information into the ML model would improve its prediction power. Structural

predictions for 6600 AAV variants are generated with RosettaRemodeling software, with an average of 10-25 conformations (poses) for each variant. After discussing different approaches to featurizing the structural data we decided to use voxelization, which converts 3D structures to 4th order tensors. We compare the results obtained by either treating the poses of each variant as independent data points, or as defining a probability distribution that the learning algorithm accesses by sampling.

We examined two major types of models -- Linear Support Vector Classifier (LinearSVC) and Neural Networks for their prediction power after training with this dataset. Both types of models were able to predict the experimental enrichment score of an AAV variant in the brain, with Neural Networks showing superior precision. LinearSVC showed its own strength in interpretability and provided useful information that might guide rational design of AAV variants with desired functions.

We expect that with more fine tuning of the model architectures and hyperparameters, more careful processing and selection of experimental datasets, and more data points generated with faster and more affordable structure modeling, the ML models will achieve greater precision. *In vivo* experiments will be used to validate the prediction power of the models and the utility of these models in improving efficiency of directed evolution of AAV capsids.

Part 1: Machine learning for directed evolution

Engineering proteins to make them useful or “fit” for a desired application is a large area of research in bioengineering. Proteins are subjected to continuous evolution for optimal performance, and directed evolution with and without rational mutation strategies has yielded greater success than a rational engineering approach, likely due to our limited understanding of the structure-function relationship of most proteins under study¹⁻⁴. Directed evolution involves generation of a large library of mutated proteins, and depending on the mutation size, the library sequence space (amino acid) can range from as low as 10^3 mutants to 10^{10} or beyond.

While the ability to select a “better-fit” variant is higher in this process of randomization, it can suffer from incomplete sequence coverage, especially for larger libraries in a given experimental setup, leading to potential loss of identifying the “best-fit” variant for a given evolution study, despite the cost of time and resources involved in these studies.

Machine learning (ML) can potentially help this problem by mapping the sequence-functionality relationship from given data, and accelerate our ability to identify “best-fit” or “equivalent better-fit” variants from the unexplored sequence space of the library^{5,6}. Performing selection *in silico* can be orders of magnitude faster than what could be achieved with experimental procedures, hence building a ML model with greater accuracy could potentially become an invaluable tool for candidate selection. While this sequence-functionality based ML is currently being explored by researchers in the field, to date the exploration is naive towards the use of protein structures as part of the ML pipeline. Extracting information merely from the sequence would lead to the loss of information for the direct impact of complex 3D structure on functionality. Hence, if significant learning can be made on sequence-functionality, our hypothesis is that we may be able to obtain improved learning and outcome by incorporating structural information.

Part 2: Evolution of adeno-associated viruses (AAVs) for efficient crossing of blood-brain barrier.

Evolving the outer protein shell (capsid) of a virus whose interaction with a host cell can mediate infectivity, is one example where structure-functionality can add more confidence in ML over sequence-functionality. Here we describe the evolution of adeno-associated virus

(AAV), a non-pathogenic single stranded DNA virus where the capsid is assembled from 60 monomeric (identical) units⁷⁻⁹. They are widely used as vectors for gene delivery applications due to their ability to target both dividing and non-dividing cells (neurons), which makes them versatile for broad applications in science and in gene therapies¹⁰⁻¹². A specific serotype, AAV9, is known to cross the blood-brain barrier (BBB) and can target the central nervous system (CNS)¹³. This is of great interest for noninvasive systemic gene delivery; however there is need for significant improvement on their efficiency and specificity to the delivery target.

Viral Tropism is mediated by:

- Capsid (outer protein shell)
- Cargo/Packaged genome (Regulatory elements)
- Route of delivery

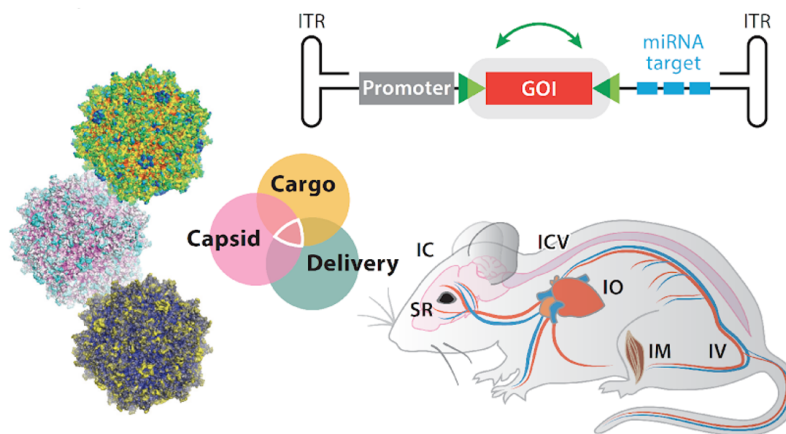


Figure 1. The AAV capsid and the route of delivery can regulate the interaction of virus with a host cell and the design of AAV cargo can control the expression of the delivered protein in a specific cell type of interest¹⁴.

Part 3: *In vivo* selection: experimental design

a. AAV variant generation, selection, and quantification

Using a directed evolution method called CREATE (Cre-recombination based AAV targeted evolution)^{15,16}, the Gradinaru Lab evolved AAV9 capsid to cross the BBB more efficiently in order to target distinct cell types of the CNS. This method relies on functional capsid recovery from cell types of interest by Cre recombination, and rounds of positive selection. A randomized 7-amino acid long peptide is inserted on the monomeric unit (between 588-589 amino acid positions) of the protein which folds into a loop that is exposed to the outer surface, and a mutated virus will contain 60 copies of the mutated surface loop. This mutagenesis strategy can generate a theoretical library size of ~1.3 billion variants (at the level of amino acid). This variation in the capsid sequence, and potential reflection on altered structure can lead to a dramatic change in their targeted host cell interaction.

We performed the capsid library selection across three cells types of the adult mouse brain - endothelial cells, which form the BBB, neurons and astrocytes - which exist as the major cell types of the CNS. We intravenously administered the virus library to adult transgenic mice that specifically express a protein (named Cre) in a specific cell type (Transgenic mice: SNAP-cre and Syn-cre for neurons, GFAP-cre for astrocytes, and Tek-cre for endothelial cells). With a next-generation DNA sequencing (Illumina) based capsid library recovery from brain tissue, we identified the performance of 9000 out of ~1.3 billion variants in one round of selection and further validated this in a second round. We also observed that out of 9000 variants, around 600 were positively enriched in brain tissue across all cell types.

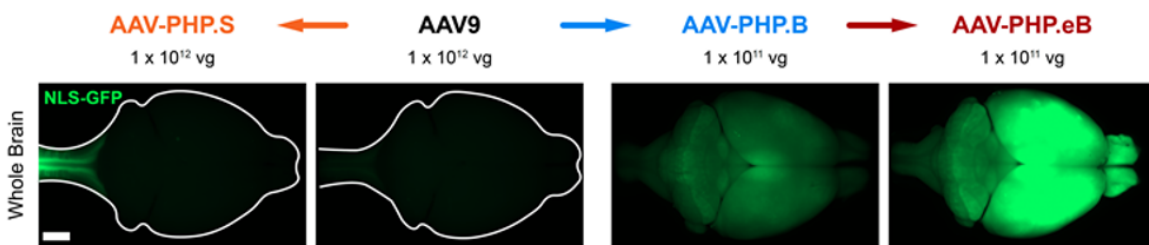
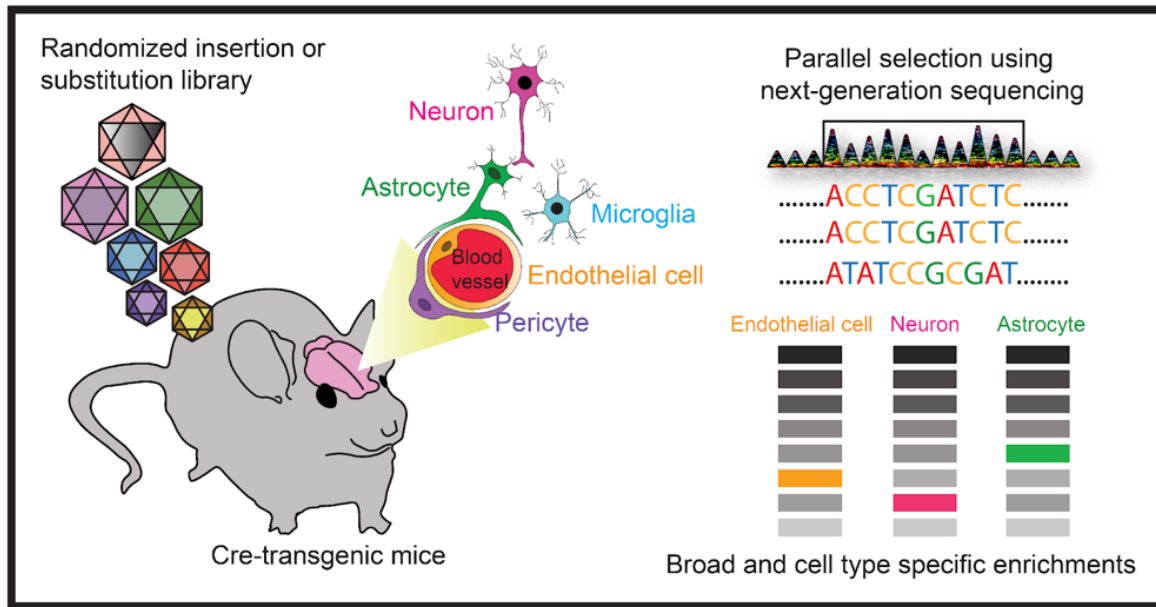


Figure 2. A multiplexed Cre-recombination based AAV targeted evolution (CREATE) method is optimized to use deep sequencing based AAV capsid recovery in order to perform positive and negative selections across multiple targets (Ravindra Kumar et al, unpublished work). Using a positive selection approach with CREATE, AAV9 was first evolved to AAV-PHP.B variant that has enhanced brain wide CNS transduction in adult mouse¹⁵. AAV-PHP.B was further evolved to AAV-PHP.eB variant with better transduction than the parent AAV-PHP.B. AAV9 was also evolved to AAV-PHP.S variant that has better transduction of peripheral nervous system¹⁶.

Library performance across cell types after 2 rounds of selection

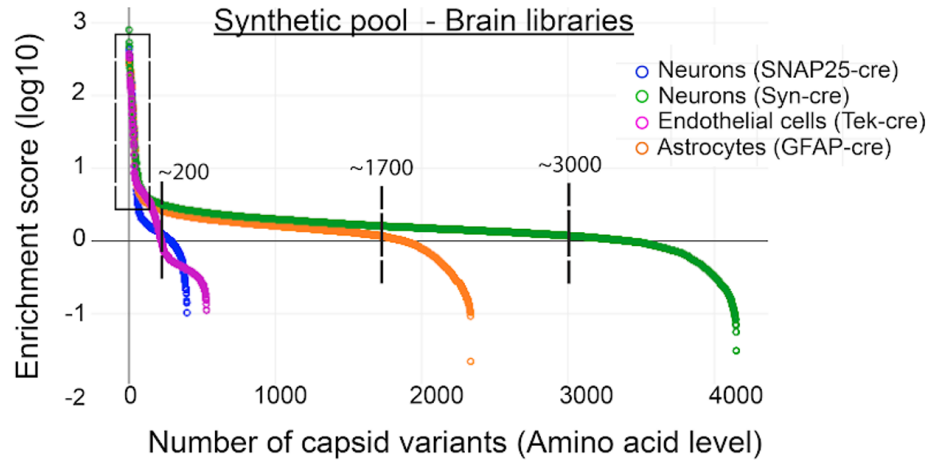


Figure 3. Using the Multiplexed CREATE method, AAV capsid libraries were selected across different transgenic mice expressing Cre in different cell types, where SNAP and Syn are for neurons, GFAP for astrocytes and Tek for endothelial cells. The plot shows the enrichment score (relative abundance of a capsid in cell type of interest) of AAV libraries after the 2nd round of selection. The number of positively enriched variants across libraries are marked by dotted lines (Ravindra Kumar et al, unpublished work).

Given the high confidence data from our selection experiment, we are interested in exploring the possibility of feeding the sequence and structure information of these variants into ML to further our chance of finding variants with better and interesting phenotypes that were uncovered in our experiments.

b. Computational problem formulation

By the end of the AAV selection experimental pipeline, we have a list of mutant sequences present at different levels in tissues of interest. The abundance of each of these mutants corresponds to the mutant's ability to deliver cargo to the recovered cell type, relative to the other mutants present in the library. Thus, for each mutant sequence, we have a utility metric, one for each cell type extracted. From the machine learning perspective, the sequence serves as the source of the features, and their relative abundances become the labels. From here, we can formulate this problem as either a classification or a regression problem - given a certain mutant sequence, we aim to predict the degree to which this mutant is able to achieve delivery to the target of interest (regression), or whether this mutant is among the top performers (binary classification, where the two classes are "top performer" and "not top performer"). The goal, once we have a model that can predict this relationship, is to feed new sequences into the model to obtain predictions on which of these are the best performers. For this work, we focus on the binary classification case, but the methods throughout could easily be extended to the regression case, and used to predict not only whether a sequence is a top performer, but how much of an improvement one could expect from model-predicted mutants.

Part 4: *In silico* selection: Sequence-based ML

a. Protein sequence featurization

Previous work on machine learning for predicting viral variants has operated exclusively on sequence space, considering only the sequence of amino acids that differ between variants. We

used these sequence-based models

as both a benchmark to compare

structural models against, and as a

starting point for deciding on a

protein featurization scheme. In this

work, the mutation region is fixed in

size, preventing the need for more complex or learned variable-length feature representations,

as have been explored in other protein engineering domains¹⁷. Thus, we opted for 3 feature

encoding schemes that could operate

on fixed length sequences: 1) one-hot

encoded amino acids 2) one-hot

encoded amino acid motifs 3)

encoding physical properties of

amino acids (Figure 4). The

motivation behind choosing the physical properties was that it could provide a better measure

of distance between sequences, since not all amino acids are uniformly different from one

another; they are often grouped into subclasses by properties such as charge, size, and

reactivity. We chose 6 properties: charge, phosphorylation, average flexibility, presence of an

ionic bond, molecular weight, and hydrophobicity. All values were obtained from BioPython¹⁸.

Of the different amino acid encoding schemes, amino acid characteristics consistently improved

performance over amino acid identity, so we opted to use a combined vector of both

featurizations going forward (Figure 5). While the amino acid motifs also performed well, their

Amino Acid by Position

Sequence	A ₁	D ₁	...	Y ₁	A ₂	...	A _N	...	Y _N
DATY		X			X				X

Amino Acid Presence

Sequence	A	...	Y	AA	AD	...	YY	...	YYY
ADTY	X		X		X				

Amino Acid Characteristics

Sequence	Charge ₁	...	Hydro ₁	Charge ₂	...	Hydro ₂	...	Hydro _N
NQTW	0.00		-3.5	0.00		-3.5		-0.9

Figure 4. Example encodings of an amino acid sequence into a one-hot encoded vector (top), and amino acid motif vector (middle), and a sequence of physical properties (bottom)

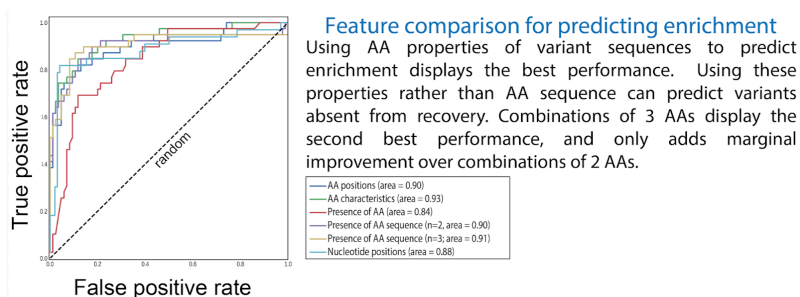


Figure 5. Comparison of encoding schemes reveals physical characteristics outperform others

feature count quickly grew (with $20^3 = 8,000$ features at a motif size of $n=3$), making this encoding scheme less appealing for scalability reasons.

b. Sequence-based models

Prior to the present work, a number of sequence-based architectures had been explored, and the architecture that seemed to perform the best on the most number of datasets was to perform a 1D convolution along the sequence dimension, with a moving window size of 3, which feeds into a series of fully connected layers that also receive input from the original input data (a skip-layer connection).

c. Baseline performance

Based on a classification scheme, sequence-based models perform significantly above random at predicting the viral variants that are highly enriched in some regions of the brain, but not others (Figure 6). This baseline provided confidence that learning on viral mutant data is possible, and might be enhanced by augmenting the sequence information with structural data.

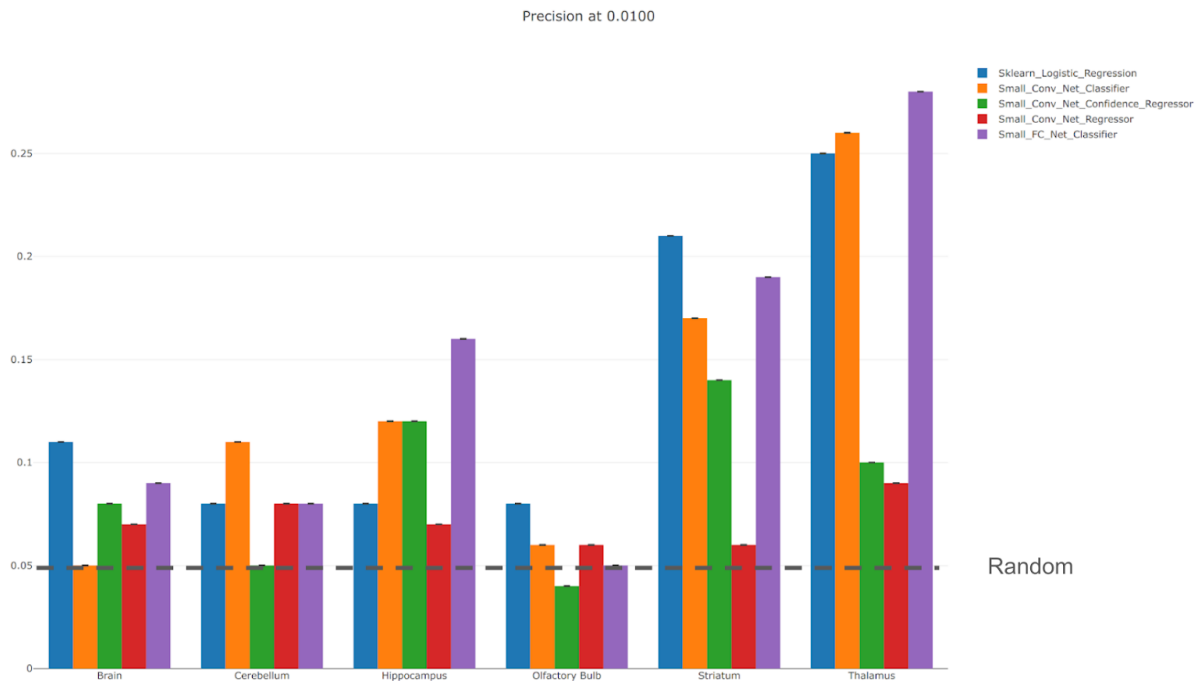


Figure 6. Sequence-based models perform significantly above random on several brain targets.

Part 5: Structure data - Rosetta modeling

a. Capsid surface structure is a determinant for its tropism

The molecular basis underlying diverse tropisms (and thus gene delivery profiles) of AAV variants is the interaction between the exterior surface of the capsids and host body proteins. These interactions can include complicate processes such neutralization by antibodies in the bloodstream, binding with proteins on the surface of target cells etc.

In our AAV library, we introduced random peptide insertions in a surface spike (blue site in the figure 7) that is regarded as a major site for virus-host interaction. Therefore, the variation in

different tropism is mostly determined by the structure of the inserted peptide i.e. 3D organization and physical chemical properties of the amino acid residues composing the peptide.

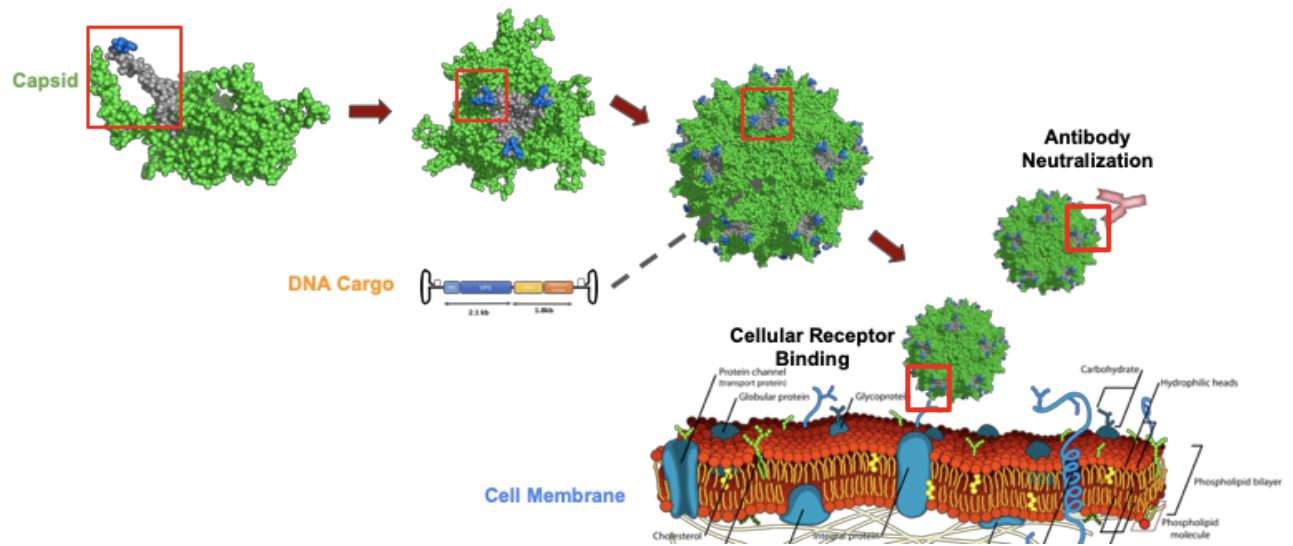


Figure 7. The gene delivery profile of an AAV variant is determined by the interaction between the exterior surface of the capsid and host body proteins. In our directed evolution method, random peptide insertions are introduced in the blue sites. Schematics for cell membrane and antibody are adapted from Wikipedia.

Given the role that the structure of capsid protein plays in host-virus interaction, we hypothesize that incorporation of structural information into the machine learning models (“protein structure based models” and “combined models” in Figure 8) can improve the prediction power of these models.

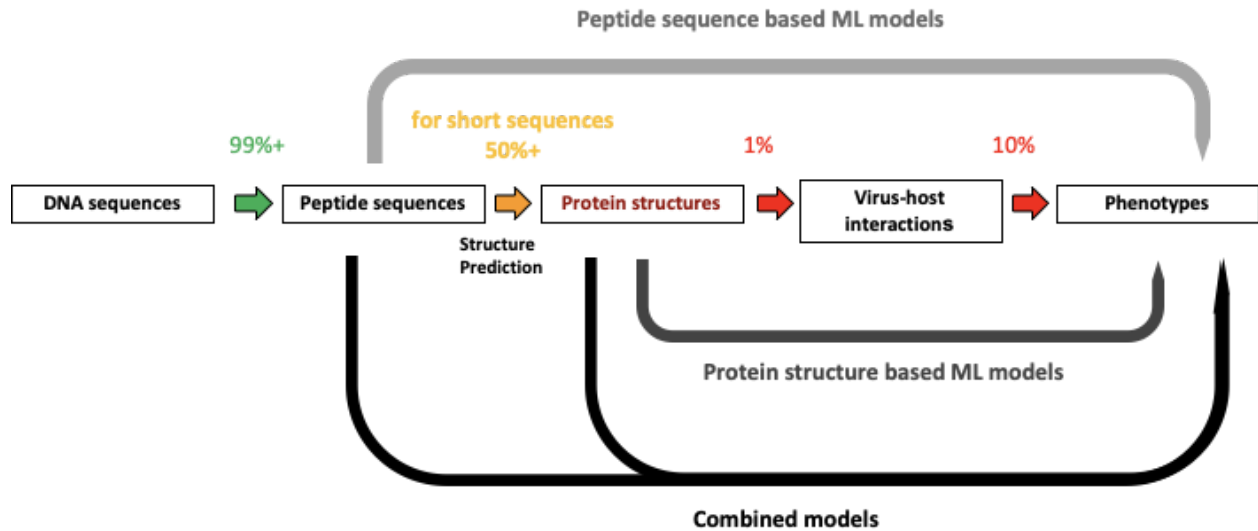


Figure 8. Incorporation of protein structure information might improve the prediction power of machine learning models

b. Feasibility of using computational structure prediction tools

Traditionally the atomic structure of proteins were determined with X-ray crystallography or single-particle CryoEM. However these methods suffer from low throughput and high cost and are thus not suitable for generating datasets to a scale that is suitable to be used as machine learning training sets. What is more, the peptide insertions on AAV surface usually take multiple possible conformations, making the structures hardly solvable by X-ray crystallography or CryoEM (Figure 3x, right panel).

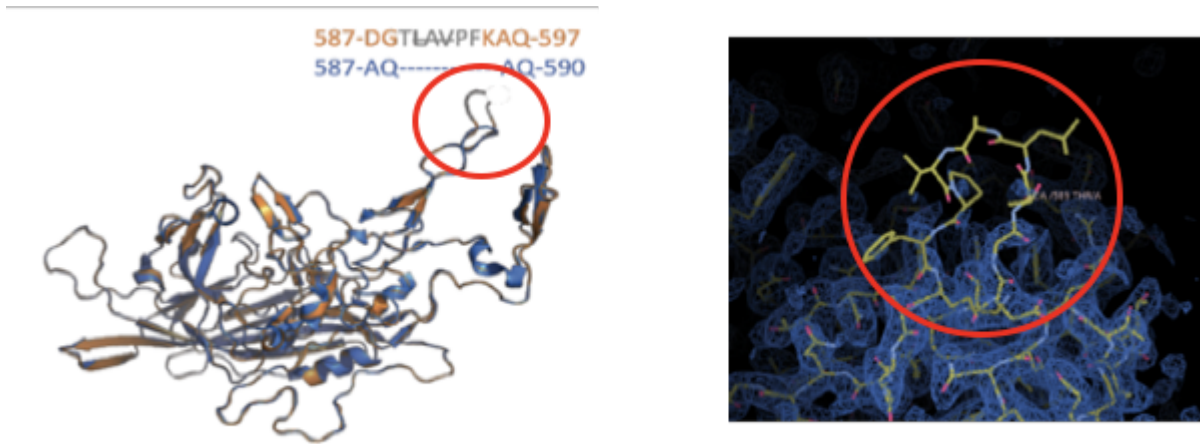


Figure 9. Single particle CryoEM study of an AAV variant PHP.eB shows that the structural change is limited to the site of insertion. Note in the right panel that the peptide insertion cannot be resolved at atomic resolution, possibly caused by existence of multiple possible conformations.

Experimental results indicate that the structural change introduced by a 7-residues peptide insertion at the specific spike is restricted to the site of insertion. In the meanwhile, recent tools in computational structural biology have robustly achieved accurate (<1Å resolution) determination of structures of peptide loops smaller than 12 residues¹⁹. Therefore, accurate prediction of the conformations of the peptide insertions take with computational tools is feasible.

a protein modeling software based on physical and statistical knowledge allows prediction of possible conformations of an AAV variant

c. High-throughput Rosetta and output

We used Rosetta, a state-of-the-art protein modeling software based on physical and statistical knowledge. The software generates random conformations of input protein sequence following certain algorithms (in this specific case we used RosettaRemodel²⁰ protocol with Next Generation Kinematic Closure with symmetric setting) and picks the conformations with a Monte Carlo sampler. Usually an optimization of details of the conformation (relaxation) follows the Monte Carlo sampling.

Typically, a number of independent Monte Carlo runs are conducted for each query protein, and each run would yield a plausible conformation, stored as a so-called “pose”. Each pose file contains information including coordinates of all atoms and energy scores of the pose (including the total energy score and compositional scores). In a typical loop modeling aiming for structures with atomic accuracy, 500 or more poses with different energy scores are generated for each query protein.

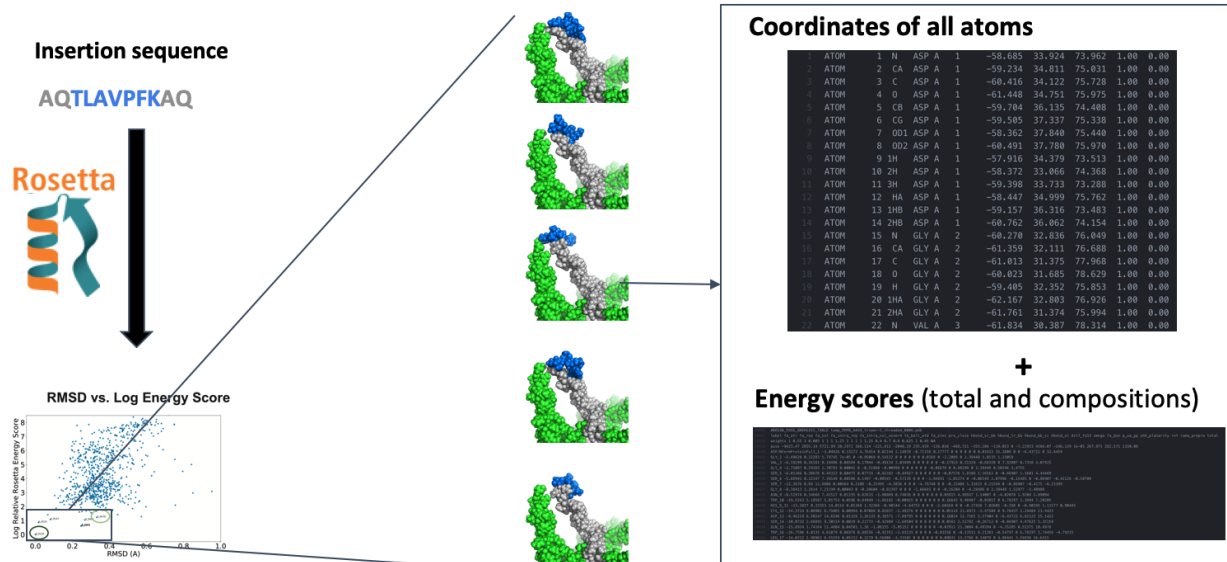


Figure 10. Output format of Rosetta structure prediction software.

As a proof-of-concept test to validate the prediction power of Rosetta on AAV variants with loop insertions, we used Rosetta to predict structures of three AAV variants -- PHP.B, PHP.eB and PHP.V1 from the library, each with 500+ poses. All three variants showed converging conformations for poses with lowest energy scores. As expected, PHP.B and PHP.eB, the two variants that showed highest enrichment in the brain, shared almost identical conformations in a few key amino acids (i.e. a few amino acids that appeared frequently in brain-enriched AAV variants), but not the rest amino acids that were inserted. PHP.V1, an AAV variant that does not show enrichment in the brain, do not share a similar conformation.

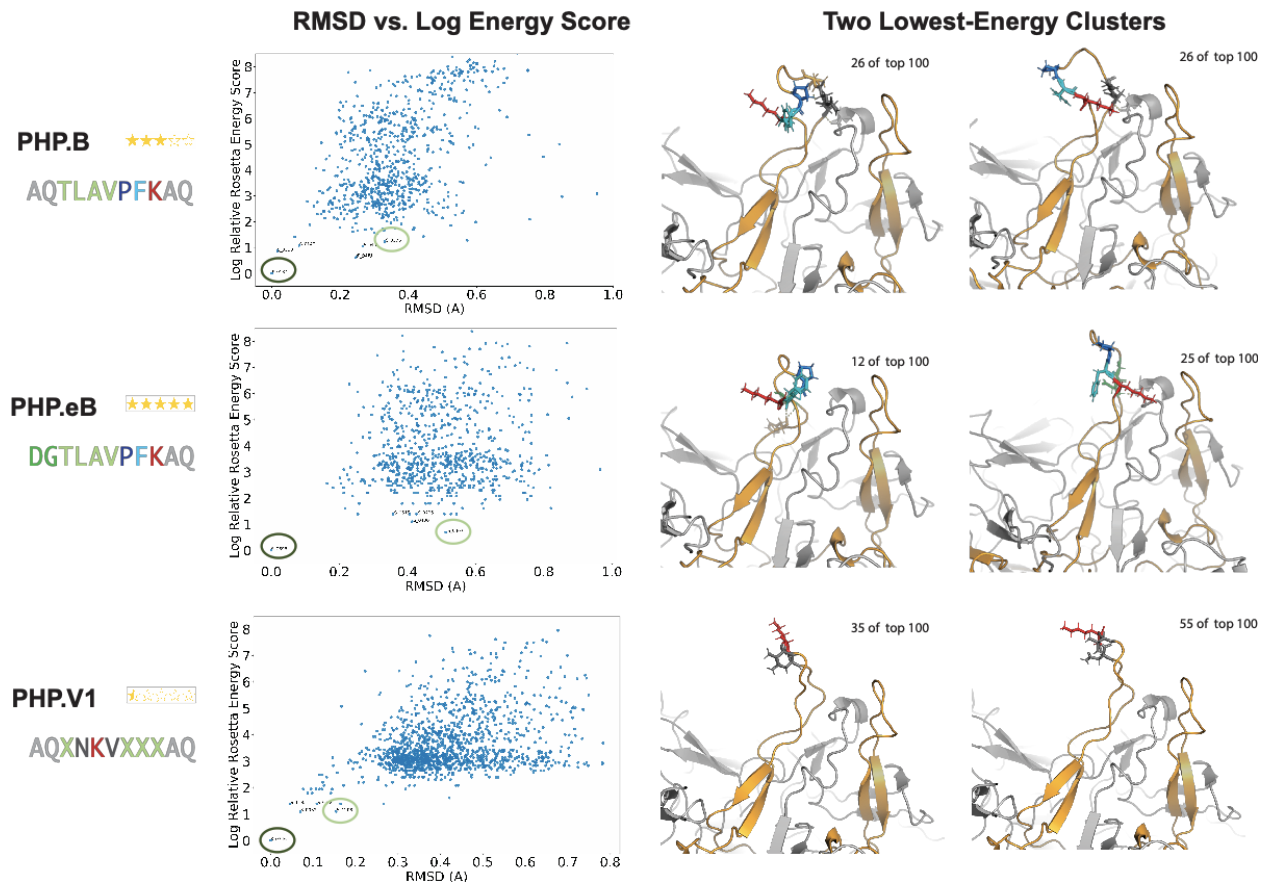


Figure 11. Some common structural features are found in variants that transduce the brain efficiently (PHP.B and PHP.eB) but not in the one that does not transduce the brain.

With cloud computing platforms such as AWS, the structure prediction method can be scaled up and used to generate structural information for a large number of AAV variants. We wrote a series of scripts for preparation, backup and analysis for Rosetta modeling on AWS, which allowed us to model thousands of variants within a few days.

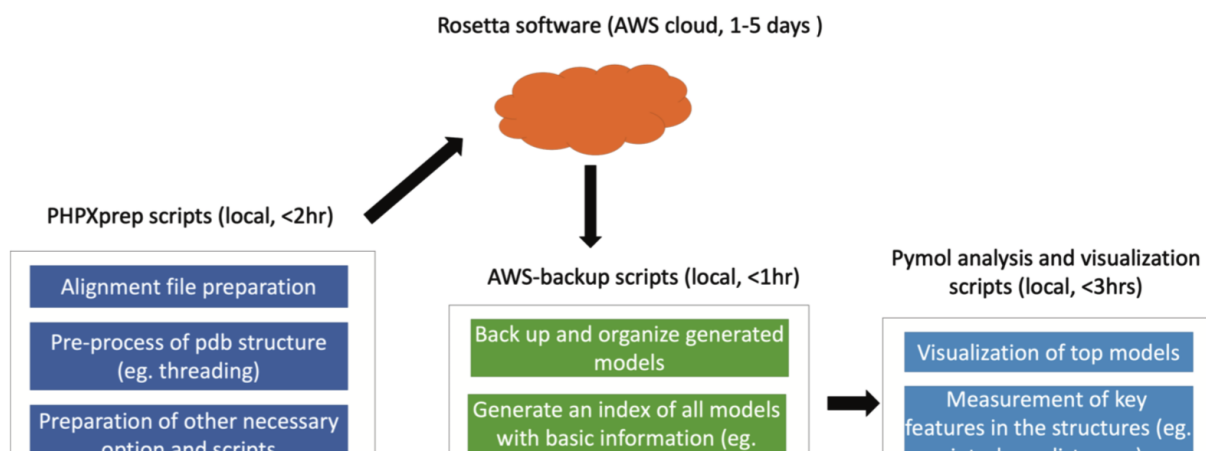


Figure 12. A pipeline for high-throughput structure prediction on the cloud.

With this pipeline for high-throughput structure prediction, we predicted structures for 6600 AAV variants with RosettaRemodel (NGK method with symmetry setting), each computed with 33 CPU*hrs. This yielded an average of 10-25 poses for each variant. To be noted here is that the number of AAV variants and the number of poses for each variant are not optimal, which is mainly limited by the computation speed of the Rosetta protocol (1-3 poses/CPU*hr).

Part 6: Featurizing protein structure

a. Possible feature schemes

In order to incorporate protein structure into machine learning methods, we have to find ways to derive features from the data provided by Rosetta. At the outset of the project we discussed different variants including the use of the pairwise distances between atoms, the use of pairwise distances between the centroids of different amino acids, or the angles formed by subsequent amino acids. Those representations have the property of being invariant to distance-preserving transformations, which is useful in many settings. In our case however we are only modelling a small part of a much bigger molecule that is assumed to stay unchanged, thus providing a natural frame of reference. Therefore it is undesirable to choose a feature map that enforces rotational invariance. Furthermore, the above mentioned schemes do not provide a natural way of combining chemical and structural information. It would be possible in

principle to directly use individual coordinates of atoms or centroids of amino acids as features. When using atom coordinates, this creates problems due to the different sizes of the different amino acids. Using the coordinates of the amino acid centroids, on the other hand, greatly reduces the amount of structural information encoded in the features.

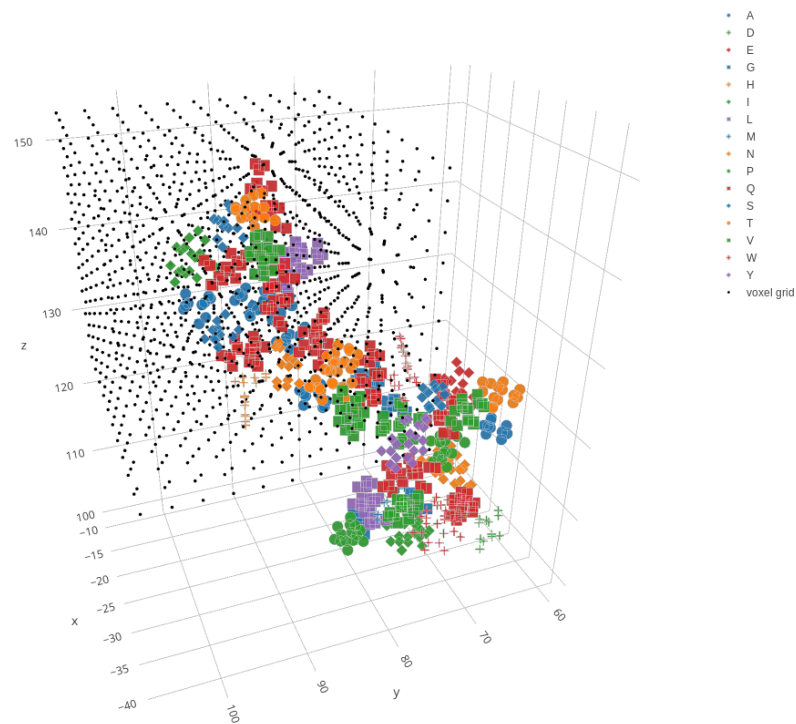


Figure 13. Visualization of the site of peptide insertion on AAV capsid. The black grids shows the voxels.

b. Voxelization

Based on the above reasoning, we decided to incorporate the protein structure by means of voxelization. In the coordinate system of the full molecule, we define a cubic grid of $n_x \times n_y \times n_z$ cubic volumes (voxels) centered at the part of the molecule that we modify. We then choose n_c attributes (channels), in our case the one-hot encoded membership to a type of amino acid and physical properties of amino acids, such as the hydrophobicity. To each voxel, we then assign an n_c -dimensional vector of real numbers that we obtain by adding the respective attributes of all atoms contained in that voxel. The resulting object can be thought of as a three-dimensional image, with resolution $n_x \times n_y \times n_z$ and an n_c -dimensional color space. As desired for our application, voxelization is not invariant to rotation or translation and can easily incorporate chemical information by adding additional channels. Furthermore it allows us to directly apply proven tools from image analysis, such as convolutional neural nets (see section 7 b).

c. Multi-pose handling

A particular challenge in the present dataset is the fact that a given pose does not have only one structure, but rather can be found in many different conformations at different times. Rosetta provides us with a sample of possible conformations (poses) together with an estimate of their potential energy (which is inversely correlated to the likelihood to find the molecule in this pose, at a given time). Naively, one could add multiple data points of the same sequence, with different structure. This however might cause confusion in the learning algorithm, since poses

that are not relevant for the desired function might be labeled with high enrichment score.

Furthermore, it raises the question which of the poses of a new variant to use when performing prediction.

Another option is to simply concatenate all the poses (or the poses with the lowest energy) into one feature vector. The problem with this approach is that the distribution of the lowest energy poses can still be highly variable and we might furthermore have different numbers of relevant (low enough energy) poses for different sequences. In order to circumvent this problem we decided to interpret Rosetta as outputting a probability distribution, given by the empirical distribution of the lowest energy poses. Any learning algorithm assessing these “bag of poses”-features then does so by drawing a fixed number of samples from the empirical distribution of the lowest energy poses as provided by Rosetta.

Part 7: *In silico* selection: Models for Structure based ML

a. Linear SVC

Although neural networks performed the best overall (see Results), we wanted to better interpret the learning of this data. To create a more visualizable result, we trained a linear support vector classifier on the voxelized data using the sklearn library. Then, we extracted the voxels with the highest weights and then plotted them on a three dimensional visualization of a variant using the plotly visualization library in python.

In order to determine whether the model was learning relevant information, we decided to specifically focus on two structurally and chemically similar amino acids. So, we compared the

weights obtained from the similar amino acids aspartic acid (D) and glutamic acid (E). The intuition behind this was that if two amino acids share chemical and structural properties, then they should also have a similar influence in determining if a variant is successful at its utility of passing through the blood brain barrier or not. As we can see from the visualizations below, both aspartic acid and glutamic acid, both have their most important weights in orange and red at relatively similar locations. To compare the weights more quantitatively, we calculated the correlation of weights between each of the amino acids. We found that aspartic acid and glutamic acid had a higher correlation with each other than with any of the other amino acids. So, we we successfully found that a structural model was able to determine this relationship of similarity between alike amino acids from our data.

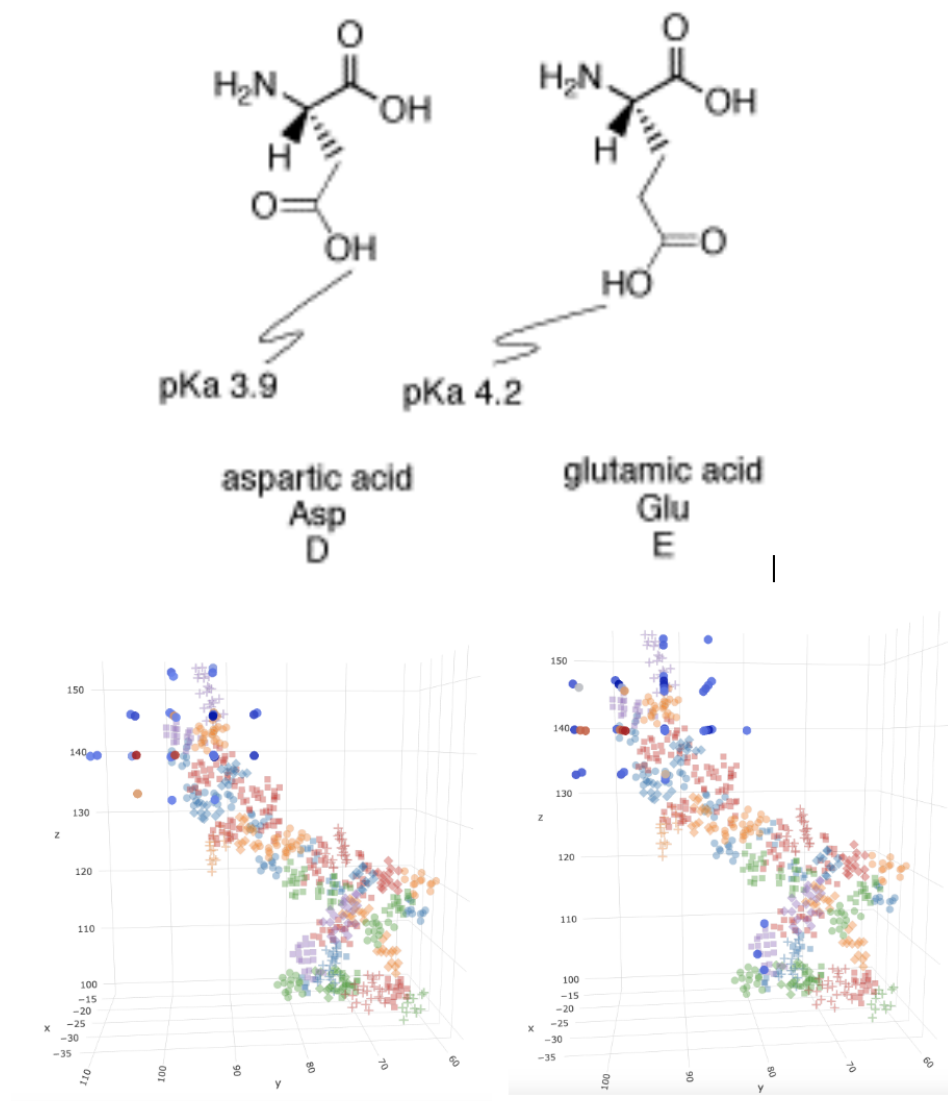


Figure 14. Structural and chemical properties of aspartic acid and glutamic acid along with their corresponding visualizations obtained with weights from the Linear Support Classifier Model

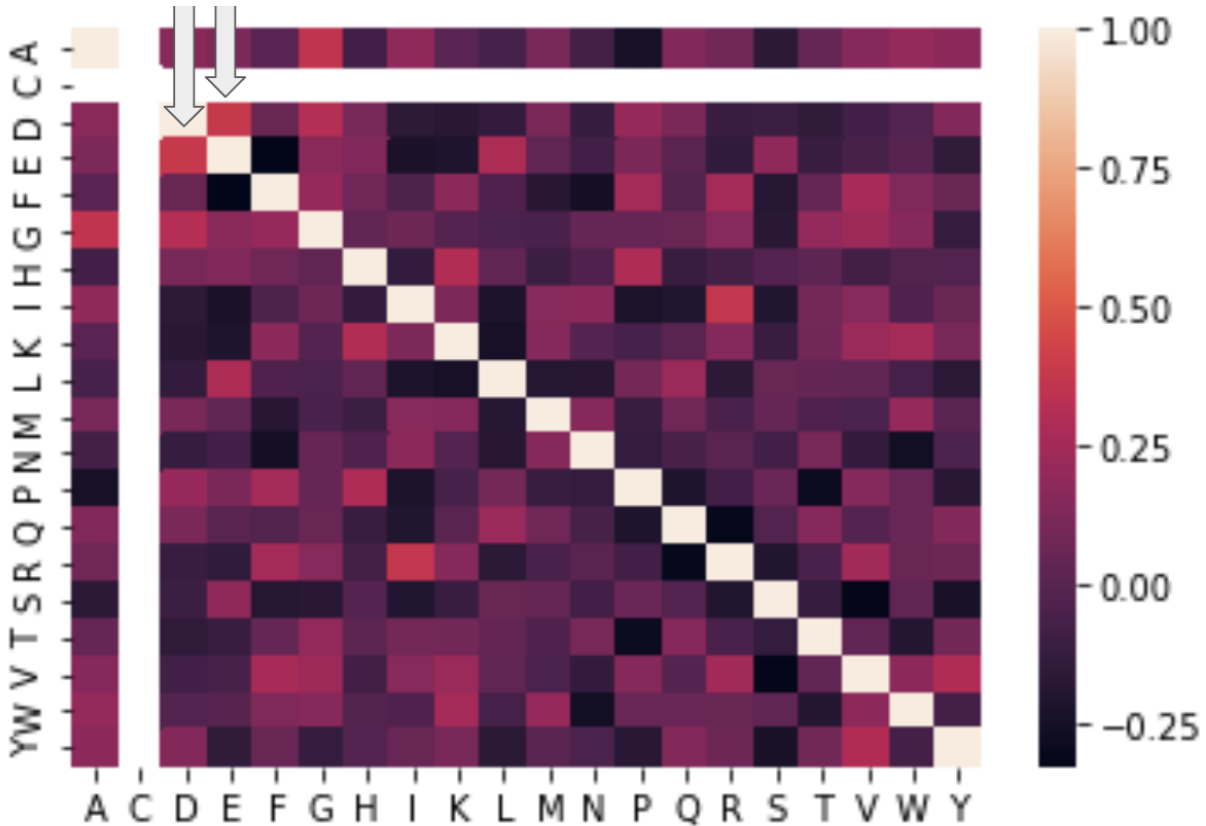


Figure 15. Heatmap showing correlation between all of the amino acids. Arrows indicate correlations between amino acids Aspartic Acid (D) and glutamic acid (E). As we can see, they have a higher correlation with each other than with any other amino acid.

b. Neural Networks

We tried a variety of neural network architectures, all implemented in PyTorch. Our initial class of models took as input a single pose of a viral variant coupled with its label. This feeds into a single 3D convolution layer (Figure 16), followed by multiple downstream fully connected layers. However, one limitation of this method is that it does not take explicit advantage of the fact that single variants can be thought of as a distribution of different poses that they can adopt. Thus, we opted for an alternative model, a bag-of-poses (BoP), that simultaneously receives several

poses, randomly sampled, for one variant, feeds each of these poses through a single shared weight 3D CNN layer, and then feeds the output of all poses into downstream fully connected layers (Figure 17). By randomly sampling the poses, the model sees the same variant with different pose configurations, fed into different copies of the CNN, yielding some pose invariance and pose order invariance. Finally, for the BoP model, we combined it with sequence information by adding a parallel stream of input containing the 1D convolution network described earlier.

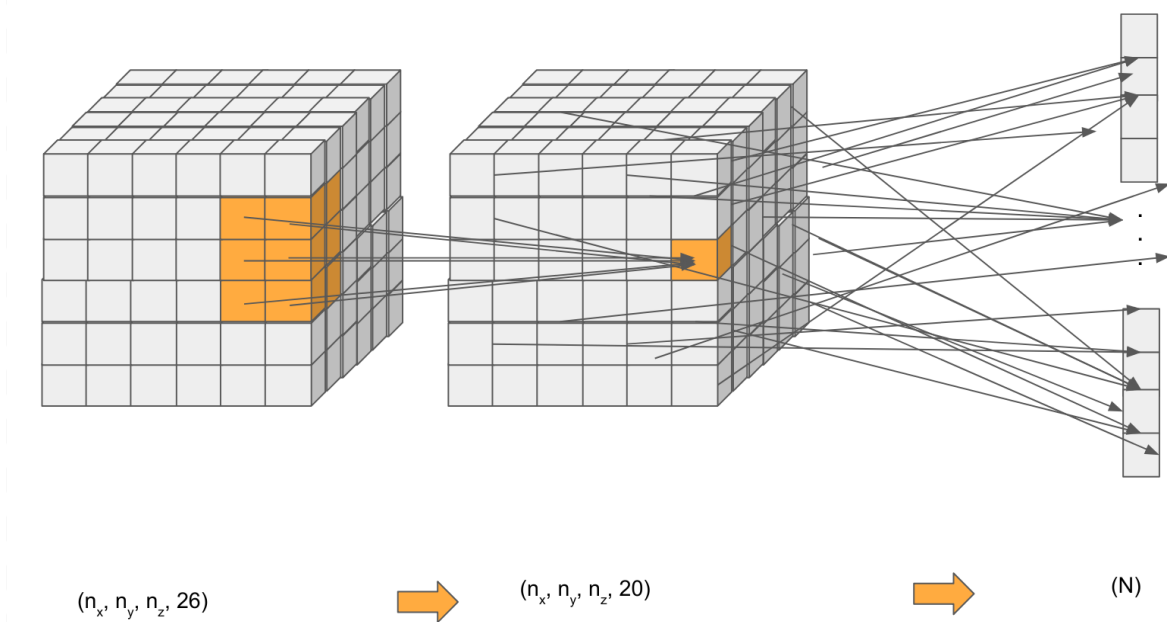


Figure 16. A simple 3D CNN layer. Each voxel and its associated 26 channels (20 amino acid identities, and 6 physical properties) are fed through a convolution filter into 20 channels.

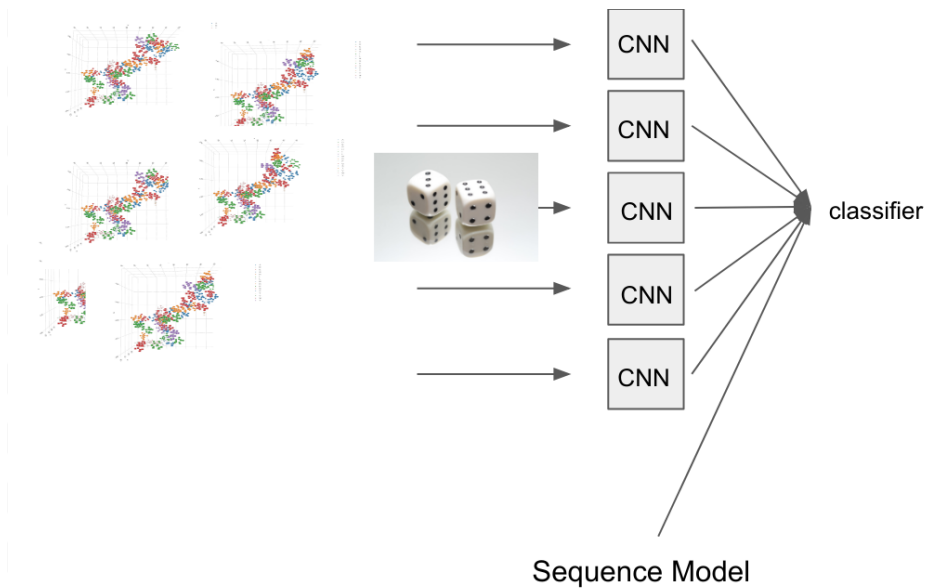


Figure 17. A bag-of-poses model for feeding in multiple poses in parallel to a shared-weight 3D CNN to simulate the physical ground truth of a protein adopting different poses at different times

Part 8: Results

a. Metrics

Since we are interested in building models that can take in an exploration set of theoretical sequences, and then output a short list of sequences that are predicted to be the top performers in this set, the performance of the model on the low score samples is irrelevant. Thus, our primary performance metric for evaluating model performance is the precision of the model at the top k th percentile of the test set. By looking at the precision at top k , we can get an estimate for how many samples in the exploration set, if tested in the lab, would be expected to be high performers.

Unfortunately, although precision at k is the ideal metric, it cannot directly used as a loss function, as it is not differentiable. Thus, for our binary classification case, we use the binary cross-entropy loss function to drive the learning process. It is interesting to note that on many occasions, the performance of a model based on cross-entropy loss was uncorrelated with, or even directly in contrast to, the performance of the model on the top k th percentile (Figure 18). However, in general, the precision at k increased over epochs as the training loss went down, and this was the metric we were ultimately interested in.

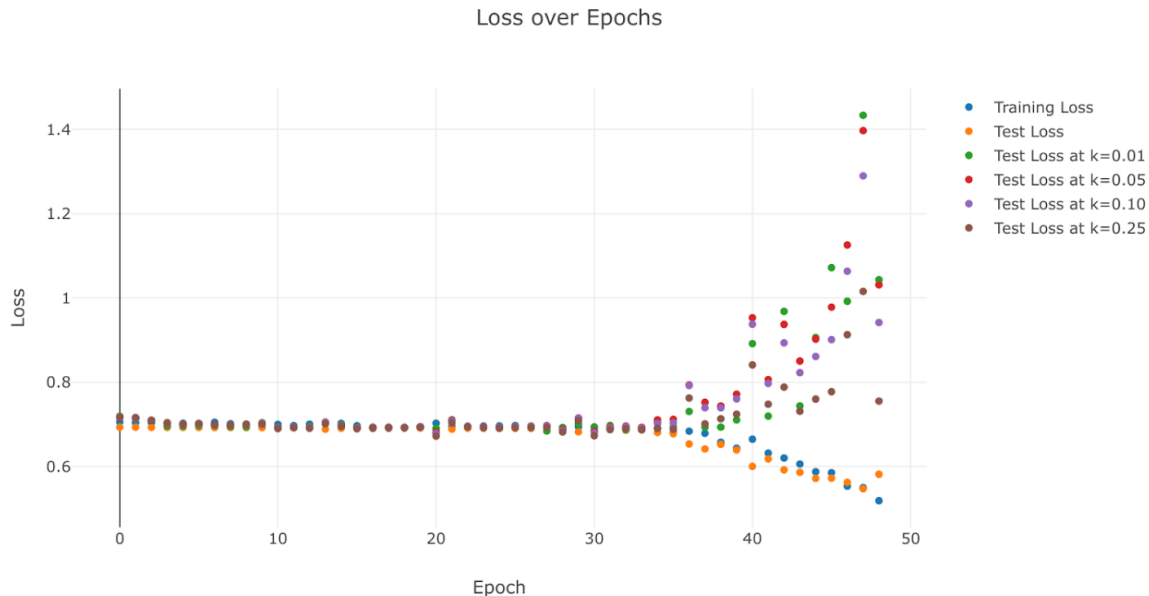


Figure 18. In some iterations, the training and test loss over epochs were at odds with the test loss on the k th percentile.

b. Effects of Pose

As the inclusion of multiple poses in ML on proteins is an unexplored problem, we investigated the effects of different pose incorporation schemes on model performance. For each of our 6,600 mutant sequences, we had at least 5 poses and their associated energy level. Since the energy level of a pose is correlated with its probability of appearing, we started by training a model with only the lowest energy (most likely) pose for each variant. Interestingly, we discovered that for a linear model such as a linear SVC, adding more poses appeared to hurt the model's performance (Figure 19). Intuitively, this makes sense - the model was now receiving multiple 3D configurations for each training data point, reducing the signal of any particular amino acids in particular positions. However, adding poses seems to increase the performance

of neural network models (Figure 20). For the naive case, we simply fed in each of the 5 poses as separate training examples, with the same label, which helped performance significantly over just feeding in the top pose (normalized to the number of total samples the network sees). Lastly, we looked at the bag-of-poses architecture, where each variant is fed in as a training example with 5 of its poses randomly sampled simultaneously. This architecture seemed to increase performance by a small, but consistent, amount.

c. Model comparison

Finally, we performed a comprehensive comparison of all model classes (sequence-based, linear structure, sequence+structure, and structure-only), and found that, as expected, non-linear structural models dramatically outperform linear structure models (Figure 21). However, despite the bag-of-poses representation, structure+sequence models underperformed sequence-only models. This strongly suggests that we have not sufficiently explored the architecture and hyperparameter space for the structural settings, since a sequence+structural model should necessarily be able to always perform at least as good as a sequence-only model, since it has all the same information, and more.

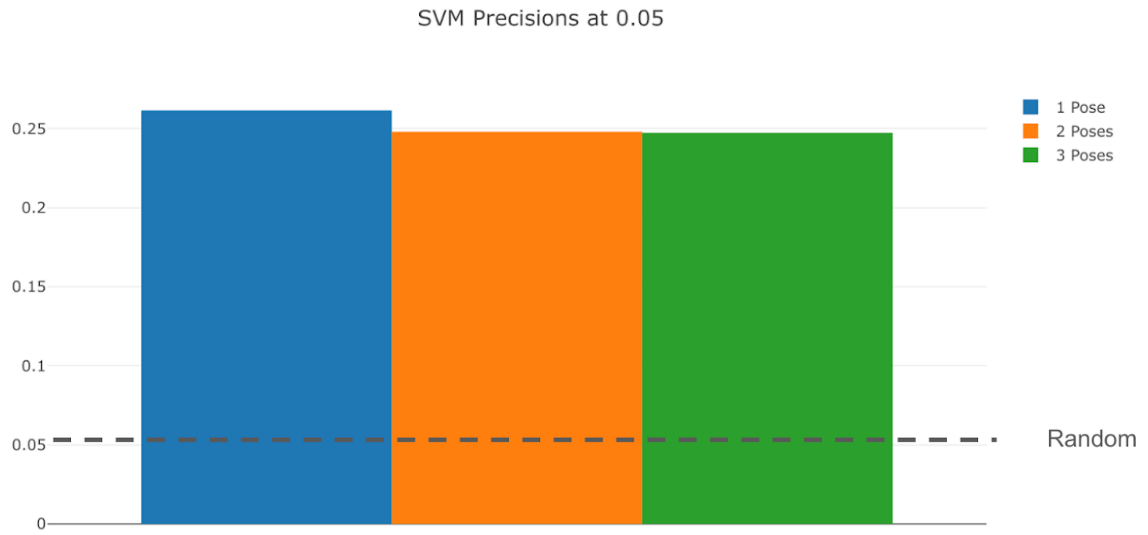


Figure 19. Increasing the number of poses hurts the performance of linear models

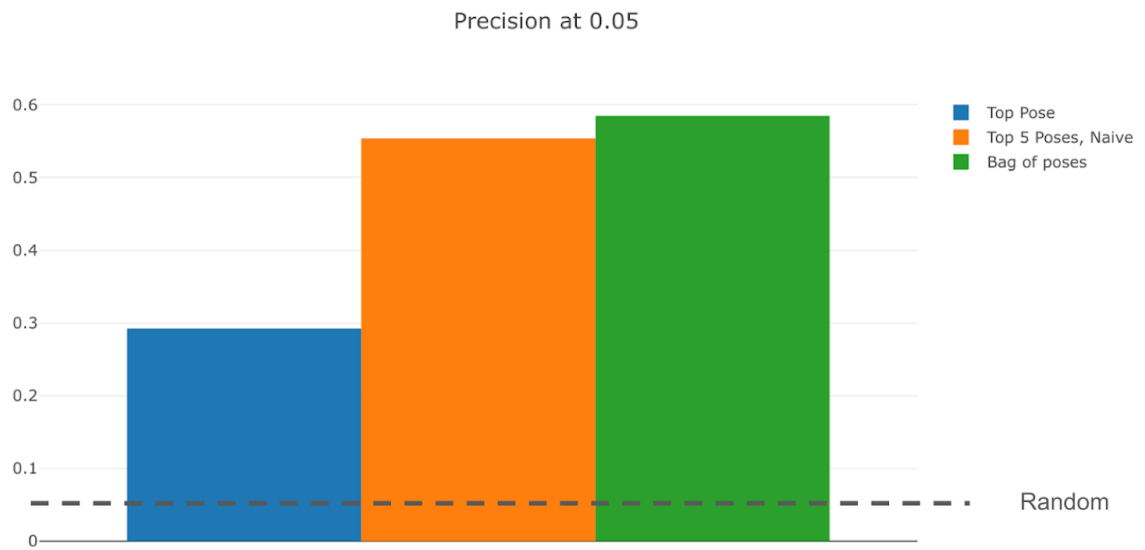


Figure 20. Adding additional poses increases the performance of 3D CNN models

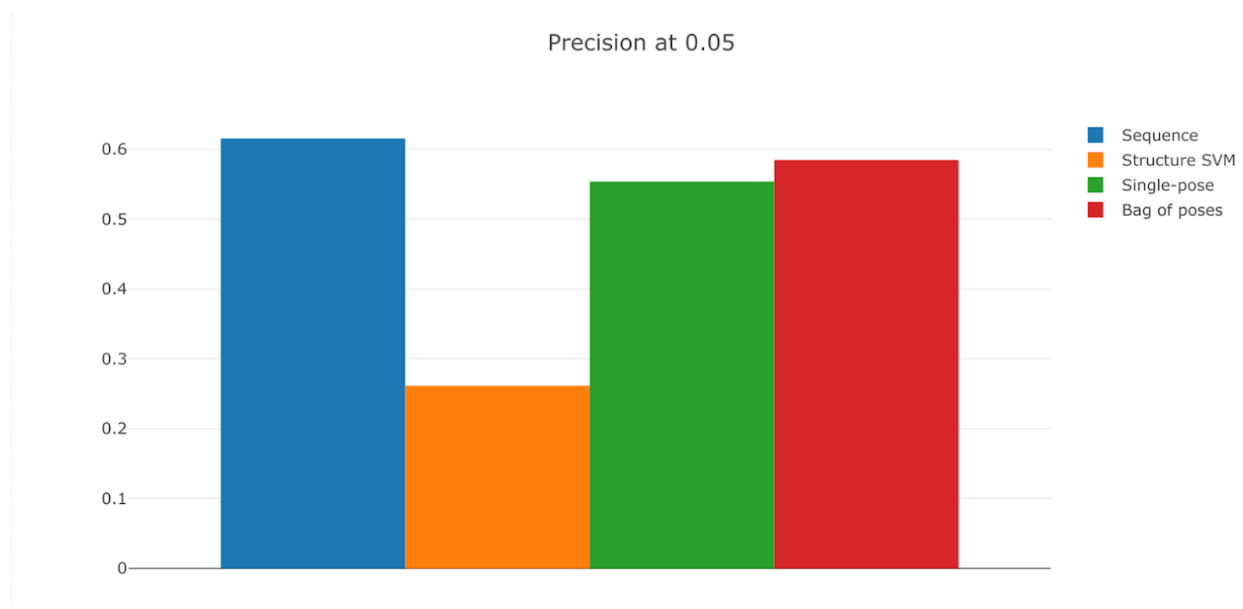


Figure 21. Comparing performance across models shows that structural models could perform as well as, but not better than, sequence models.

Part 9: Discussion

Adding computationally-predicted structural information to the machine learning pipeline of protein engineering is a new and unsolved problem, and has the potential to increase the throughput of sequencing-based directed evolution efforts by orders of magnitude. It has an even higher impact on protein engineering applications where the utility readout requires work with animal models, where experiments can take months for single rounds of screening or selection. AAV engineering in particular, which has both basic research and therapeutic applications, and where our knowledge of the mechanism of action is extremely limited, is a prime candidate for benefiting from data-driven acceleration of mutant selection and screening.

We have devised and implemented several schemes for processing and learning on multi-pose AAV variant deep mutational scanning experiments, and have explored the effects of multiple poses on learning models. Based on the results of this work so far, we cannot draw any conclusions about whether computationally-predicted structure can provide information beyond what sequencing-based models alone can provide; however, we have established that learning a complex utility function such as viral blood-brain barrier crossing and cell infection from low-fidelity structural estimates is possible, and can garner insight into the amino acids and positions important for this utility.

Perhaps most importantly, we have discovered that there is a unique and interesting challenge to doing machine learning on computationally-predicted protein structure, which is that protein structures can be thought of as distributions of poses adopted at different frequencies, and that architectures that learn on samplings of such distributions can yield improved performance.

There are many directions to follow up on, some possibilities being: incorporating the energy levels to guide pose sampling, incorporating the energy levels into the model, and using the number of poses generated per unit time as a proxy for confidence or protein stability.

Our next steps for validating both the sequence and structure-based ML models will be to produce, in the lab, some viral variants that are predicted by each of the models (an exploration set), and compare their performance to the variants present in the test set. This will be the ultimate test of whether the models are learning true utility function mappings, or whether they are overfitting to some artifacts in the data.

Finally, we look forward to future versions of Rosetta that will have increased throughput of structural predictions. The results presented in this report were based on 6600 AAV variants,

each with around 20 poses generated by Rosetta, which is far less than the optimal number of poses (500+) needed for typical atomic-level modeling of short loops with 12 residues. Given that feeding 5 poses to the structure-based ML models led to higher prediction scores than a single pose, it is very possible that structure-based ML models require significantly more data than their sequence-based counterparts, and that, much like the field of image classification, larger models and increased dataset sizes could be the enabling factors that transition structure-based ML from a sequence-based alternative to the state-of-the-art in protein engineering predictions.

Part 10: Supplemental

Preprocessing Pipeline

One of the first steps required in this pipeline was to aggregate the data in a consolidated form that was suitable for training and testing. The information in pdb files outputted by rosetta needed to be matched with experimental findings of the utility of each variant, and then placed into a singular struct with all the relevant information that we wanted to learn on. This preprocessing workflow was done in python using the open source library Biopython for computational biology. From the pdb file, for each pose of the variants, we extracted the estimated 3-D coordinates, atomic identities, amino acid identities, and pose energies provided by rosetta, and then included the corresponding sequence as well as the utility label of that

pose from an excel library spreadsheet containing information about the performance of each variant. Then, using the pickle python library, this collected structure was serialized into a binary pickle file that could be loaded with all the appropriate information we desired.

REFERENCES:

1. Packer, M. S. & Liu, D. R. Methods for the directed evolution of proteins. *Nat. Rev. Genet.* **16**, 379–394 (2015).
2. Davis, A. M., Plowright, A. T. & Valeur, E. Directing evolution: the next revolution in drug discovery? *Nat. Rev. Drug Discov.* **16**, 681–698 (2017).
3. Arnold, F. H. Directed Evolution: Bringing New Chemistry to Life. *Angew. Chem. Int. Ed.* **57**, 4143–4148 (2018).
4. Cobb, R. E., Chao, R. & Zhao, H. Directed Evolution: Past, Present and Future. *AIChE J. Am. Inst. Chem. Eng.* **59**, 1432–1440 (2013).
5. Yang, K. K., Wu, Z. & Arnold, F. H. Machine learning in protein engineering. *ArXiv181110775 Q-Bio* (2018).
6. Wu, Z., Kan, S. B. J., Lewis, R. D., Wittmann, B. J. & Arnold, F. H. Machine-Learning-Assisted Directed Protein Evolution with Combinatorial Libraries. *ArXiv190207231 Q-Bio* (2019).
7. Weitzman, M. D. & Linden, R. M. Adeno-Associated Virus Biology. in *Adeno-Associated Virus* (eds. Snyder, R. O. & Moullier, P.) **807**, 1–23 (Humana Press, 2012).
8. Vance, M. A., Mitchell, A. & Samulski, R. J. AAV Biology, Infectivity and Therapeutic Use from Bench to Clinic. *Gene Ther. - Princ. Chall.* (2015). doi:10.5772/61988
9. DiMattia, M. A. *et al.* Structural Insight into the Unique Properties of Adeno-Associated Virus

- Serotype 9. *J. Virol.* **86**, 6947–6958 (2012).
10. Naso, M. F., Tomkowicz, B., Perry, W. L. & Strohl, W. R. Adeno-Associated Virus (AAV) as a Vector for Gene Therapy. *Biodrugs* **31**, 317–334 (2017).
 11. Daya, S. & Berns, K. I. Gene Therapy Using Adeno-Associated Virus Vectors. *Clin. Microbiol. Rev.* **21**, 583–593 (2008).
 12. Murlidharan, G., Samulski, R. J. & Asokan, A. Biology of adeno-associated viral vectors in the central nervous system. *Front. Mol. Neurosci.* **7**, (2014).
 13. Foust, K. D. *et al.* Intravascular AAV9 preferentially targets neonatal neurons and adult astrocytes. *Nat. Biotechnol.* **27**, 59–65 (2009).
 14. Bedbrook, C. N., Deverman, B. E. & Gradinaru, V. Viral Strategies for Targeting the Central and Peripheral Nervous Systems. *Annu. Rev. Neurosci.* **41**, 323–348 (2018).
 15. Deverman, B. E. *et al.* Cre-dependent selection yields AAV variants for widespread gene transfer to the adult brain. *Nat. Biotechnol.* **34**, 204–209 (2016).
 16. Chan, K. Y. *et al.* Engineered AAVs for efficient noninvasive gene delivery to the central and peripheral nervous systems. *Nat. Neurosci.* **20**, 1172–1179 (2017).
 17. Yang, K. K., Wu, Z., Bedbrook, C. N. & Arnold, F. H. Learned protein embeddings for machine learning. *Bioinformatics* **34**, 4138–4138 (2018).
 18. Cock, P. J. A. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinforma. Oxf. Engl.* **25**, 1422–1423 (2009).
 19. Stein, A. & Kortemme, T. Improvements to Robotics-Inspired Conformational Sampling in Rosetta. *PLOS ONE* **8**, e63090 (2013).
 20. Huang, P.-S. *et al.* RosettaRemodel: A Generalized Framework for Flexible Backbone Protein Design. *PLOS ONE* **6**, e24109 (2011).