Molecular Matchmaker: selecting binding peptide-aptamer pairs using machine learning

Aishwarya Mandyam* Yuhao Wan* Luis Ceze Jeff Nivala Kevin Jamieson

Paul G. Allen School of Computer Science & Engineering, University of Washington

Abstract

The ability to predict the behavior of specific biomolecular interactions can have a meaningful impact in a variety of domains, including drug design and medical diagnostics. Current methods to determine binding affinity between a pair of molecules can be time-consuming, redundant, and require extensive wet lab experimentation. Our work introduces new datasets of aptamer-peptide pairs collected through a multiplexed mRNA display-based aptamer binding assay that can be used to train binding affinity prediction models. We also present models that make nontrivial prediction of aptamer-peptide binding affinities and discuss our preliminary work to computationally generate an aptamer that binds to a given target peptide.

1 Introduction

Many therapeutics and diagnostics rely on the ability to recognize specific peptide motifs, which are short sequences of 5-10 amino acids. One way to recognize the presence of a particular target peptide is to use an *aptamer* designed to bind to the peptide with high specificity, minimizing interaction with off-target peptides. An aptamer is most commonly a single-stranded piece of DNA or RNA of around 40 nucleotides that effectively serves as an affinity reagent for other molecules such as peptides, whole proteins, cells or small molecules. Once the aptamer binds to its target, it can trigger a reporter, like fluorescence, for readout or even modify the activity of the target molecule to achieve some task. For example, cancer cells display peptides called neoepitopes. [1] shows that these neoepitopes can be immunotherapy targets. If researchers know an aptamer sequence that will bind to a target neoepitope, the aptamer can be used as a vehicle for drug delivery to cancerous cells to aid in immunotherapy.

While many applications require identifying a novel peptide-binding aptamer, the process is labor intensive and slow. One way to do this is Systematic Evolution of Ligands by Exponential Enrichment (SELEX) [2, 3]. Starting from a large pool of random aptamer sequences, the SELEX process enriches for aptamers that bind to the target molecule with high affinity. This process continues over many rounds of selection and enrichment for aptamers with the highest probability of binding to the target molecule [4]; the process ends once the initial pool of random aptamers is filtered down to one or a few sequences with high binding affinity. While effective, the SELEX method's multiple rounds of selection and enrichment can consume many days to weeks of labor in the lab. The method must also be conducted individually for each target molecule. While different computational methods have been explored to accelerate the SELEX process, such as replacing initial random aptamer pools with pre-structured aptamer libraries, they only marginally improve the SELEX pipeline [5].

Our goal is to use machine learning techniques to construct an in-silico model that can computationally construct, de novo, an aptamer that binds to any given target peptide sequence, eliminating the need for laborious lab experiments. The challenge in building this model is the lack of quality training data. Though SELEX data can generate a number of one vs. many training data examples (i.e., a few aptamers in a large pool that bind to a single, specific target molecule), machine learning models need training data that consists of abundant target examples and their binding aptamers (many vs. many) in order to generalize and predict an aptamer sequence for any given target.

Our primary contribution is a new dataset for training models that predict peptide-aptamer binding affinity. We collected these data using a novel experimental method that generates binding data for many different target peptides and aptamer pairs in multiplex. Our second contribution is a preliminary investigation of deep learning models trained on this dataset. We demonstrate that these models can make non-trivial prediction of binding affinities of held-out peptide-aptamer pairs, validating that the dataset contains useful information. Third, given a target peptide sequence, we propose a simple mixed integer optimization procedure to maximize our estimator's predicted binding affinity in order to output promising aptamer-peptide pairs.

Related Work Recent work shows that ML can be used to predict aptamer-protein interactions using a database of published aptamer-protein pairs as training data (positive data) and computationally generating non-binding pairs (negative data) [6, 7, 8]. While all these results used the same aptamer database (Aptamer Base [9]) that includes data on around 1000 aptamers and around 150 proteins, their approaches differ mainly in the ways they extract sequence features and the method they use to generate negative (non-binding) aptamer-peptide examples. While the model of [8] appears to achieve state-of-art for predicting aptamer-protein binding pairs, several aspects of their methodology have led us to pursue alternative strategies. First, the methodology uses a relatively small database that consists of only a few hundred aptamer-peptide pairs, therefore, the authors resort to computational methods (i.e., SMOTE) to generate more examples. Given the large potential sequence space of all possible aptamer and protein sequences, we consider it likely that such a model will overfit. Second, the model uses semi-random DNA sequences as negative example aptamers by filtering against sequences with secondary structures that are characteristic of known aptamers. The procedure of generating negative data likely trains a model that learns the filter characteristics, which does not necessarily contribute to better characterizing the aptamer space de novo.

Our work defines binding affinity as an approximation to the probability of binding: an affinity measurement of 0.99 indicates that the pair of molecules is 99% likely to bind given that they were exposed to each other in solution. Note that unlike some of the preceding works, we do not measure binding affinity directly but instead measure whether a binding event occurred, which we model as a probabilistic process.

2 Methods

Dataset We produced two datasets (one positive, one negative) from four independent replicates of an experiment. These four replciates are IID, so a model trained on one replicate can be feasiably evaluated on the others.

Each replicate consists of two sets of aptamer-peptide pairs, positive and negative pairs. Each sequencing pair results from a "molecular snapshot" taken during each experiment and readout with high-throughput DNA sequencing, where a large pool of aptamers (strands of uniformly random DNA sequences) were mixed with a large library of nearly random, mRNA-labeled peptides (peptide sequences closely follow the "NNK" distribution. [10]). When this snapshot is taken on the positives, an aptamer that is physically close to a peptide has a higher chance of being paired with that peptide's mRNA sequence in the dataset, than to any other random peptide present in the pool. From the peptide's mRNA open reading frame, we can infer its peptide sequence. The outcome of this process is that aptamer-peptide pairs that bind with higher affinities will more likely be close to each other when the snapshot is taken. However, there is a non-zero possibility that a given aptamer-peptide pair is near to one another by chance (i.e., random diffusion) or for other reasons specific to the experimental setup that are not due to aptamer-peptide binding. Thus, the result of the positive experiments is a set of aptamer-peptide pairs that may or may not have interacted due to high affinity aptamer-peptide interactions.

The negative dataset was generated when peptides were removed from the experiment and only the mRNA was included along with the aptamers. In this case, the experiment still generates a molecular snapshot of aptamer-peptide pairs, but we know this co-occurrence and association are completely independent of an aptamer-peptide binding event. The negatives can be used as a negative control.

Regions of the aptamer-peptide space where positives demonstrate increased density relative to controls are regions where peptide-dependent binding is likely. We can identify these regions in the datasets using standard binary classification. The next section presents a generative model for the data that justifies the use of binary classification.

Probabilistic Generative Model The preceding datasets are more complex than a simple, long list of pairs with a binary label describing whether or not they bound. Here, we propose a probabilistic generative model that can interpret the datasets' outcomes. This model motivates a natural loss function to train ML models that predict binding affinity. Let $\mathcal{X}$ be the complete set of peptides of length 8, and let $\mathcal{Y}$ be the complete set of aptamers of length 40. A peptide distribtion P_X is induced by a random library and follows the so-called NNK distribution: each amino acid in the sequence is drawn IID from a known multinomial distribution over the 20 amino acids [10]. An aptamer distribution P_Y is induced by choosing each nucleotide uniformly at random so that all 4^{40} aptamers are equally likely.

The experiment is modeled as follows. A set $X \subset \mathcal{X}$ of peptides is drawn IID from P_X, and $Y \subset \mathcal{Y}$ is drawn IID from P_Y. Among all pairs in $X \times Y$, we observe a random subset, which we denote as our positives dataset $S = \{(x_i, y_i)_{i=1}^n\}$. There are two pathways for a pair $(x, y) \in X \times Y$ to be included in S: the first through the pair binding, and the second through experimental error. We assume that given $(x, y) \in X \times Y$, the pair has some probability of the aptamer binding to the peptide and appearing in S which we denote as $\mu(x, y)$. That is, if an individual x and

y are put in solution over a period of time, it is the likelihood that they will be bound to one another at the instant of our snapshot and contained in our subsample. The second pathway is due to random molecular collisions driven by diffusion (non-specific interactions), and we assume that there is a non-zero probability $\omega(x, y) \in [0, 1]$ of any $(x, y) \in X \times Y$ appearing in S independent of whether or not they bind. Thus, the probability that a specific pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is observed in S, $\mathbb{P}((x, y) \in S)$ is

$$\mathbb{P}((x, y) \in S | x \in X, y \in Y)\mathbb{P}(x \in X)\mathbb{P}(y \in Y) = [\mu(x, y) + (1 - \mu(x, y))\omega(x, y)]\, P_X(x)P_Y(y) =: p_S(x, y).$$

We make the approximation for simplicity that each $(x, y) \in S$ is an IID draw from $p_S(x, y)/C_S$, where C_S is a normalization constant to make $\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_S(x, y)/C_S = 1$. We denote the negatives set $T = \{(x_i, y_i)_{i=1}^m\}$ identically to S except the experiment was designed so that all aptamer-peptide-specific binding affinity is nullified, essentially forcing $\mu(x, y) = 0$. Thus,

$$\mathbb{P}((x, y) \in T) = \mathbb{P}((x, y) \in T | x \in X, y \in Y)\mathbb{P}(x \in X)\mathbb{P}(y \in Y) = \omega(x, y)P_X(x)P_Y(y) =: p_T(x, y)$$

Similarly, we approximate each sample in T as an IID draw from $P_T(x, y)/C_T$, where C_T is a normalization constant.

Our goal is to learn a surrogate function $f_\theta(x, y)$ parameterized by θ (e.g., a neural network with weights θ) that accurately estimates the true binding probability $\mu(x, y)$. For some $\alpha > 0$ and $\beta > 0$, we propose the following optimization problem inspired by regularized maximum likelihood estimation

$$\widehat{\theta} = \arg\max_\theta \tfrac{1}{|S|} \sum_{(x,y)\in S} \log(f_\theta(x, y)) + \tfrac{\alpha}{|T|} \sum_{(x,y)\in T} \log(1 - f_\theta(x, y)) + \beta \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} P_X(x)P_Y(y) \log(1 - f_\theta(x, y)).$$

A larger α penalizes predicting a pair from the negative set T as positive. Because we hypothesize that binding is a very rare event relative to the vast $\mathcal{X} \times \mathcal{Y}$ space, a larger β penalizes predicting a *random* pair (which presumably does not bind) as positive. We solve this optimizion problem in PyTorch using stochastic gradient descent.

Assuming our predictors f_θ have sufficiently large statistical capacity, as $|S|, |T| \to \infty$, the empirical averages concentrate around their expectations, and we have that

$$\begin{aligned} f_{\widehat{\theta}}(x, y) &\to \arg \max_{\eta\in[0,1]} \frac{p_S(x, y)}{C_S} \log(\eta) + \alpha \frac{p_T(x, y)}{C_T} \log(1 - \eta) + \beta P_X(x)P_Y(y) \log(1 - \eta) \\ &= \frac{p_S(x, y)/C_S}{p_S(x, y)/C_S + \alpha p_T(x, y)/C_T + \beta P_X(x)P_Y(y)} \\ &= \frac{(\mu(x, y) + (1 - \mu(x, y))\omega(x, y)}{(\mu(x, y) + (1 - \mu(x, y))\omega(x, y) + \alpha\omega(x, y)C_S/C_T + C_S\beta}. \end{aligned}$$

Thus, we observe that $f_{\widehat{\theta}}(x, y)$ is monotonic in $\mu(x, y)$, indicating that this optimization procedure will yield a prediction that correlates with binding. The quantities α, β in the optimization problem are treated as hyperparameters to control the false alarm rate (FAR). For instance, if experimental error is large relative to binding probability, i.e., $\omega(x, y) > \mu(x, y)$, the FAR can be controlled by increasing α. Where $\omega(x, y) = 0$, increasing β controls the FAR since otherwise, if $\beta = 0$, then $f_{\widehat{\theta}}(x, y)$ would tend towards 1 for any positive $\mu(x, y)$. We use cross validation to choose α and β. Specifically, for a desired FAR of $\delta > 0$, we could choose $\alpha \geq 0, \beta \geq 0$ and a threshold $\gamma \in (0, 1)$ such that, on a test set $\mathbb{E}_{(x,y)\sim P_X P_Y}[\mathbf{1}\{f_{\widehat{\theta}}(x, y) > \gamma\}] \leq \delta$ and $\mathbb{E}_{(x,y)\sim T}[\mathbf{1}\{f_{\widehat{\theta}}(x, y) > \gamma\}] \leq \delta$, but making $\mathbb{E}_{(x,y)\in S}[\mathbf{1}\{\{f_{\widehat{\theta}}(x, y) > \gamma\}]$ as large as possible. Alternatively, we could select $\alpha > 0$ and $\beta > 0$ based on test AUC with respect to S versus the worst-case of $P_X P_Y$ and T.

In-silico Aptamer Optimization Given a surrogate model $f_{\widehat{\theta}}(x, y)$ for the true binding probability $\mu(x, y)$, we can use the model to identify promising aptamers that bind to a given target peptide. That is, given a target peptide $x_0 \in \mathcal{X}$, we want to identify $\arg\max_{y\in\mathcal{Y}} f_{\widehat{\theta}}(x, y)$. Unfortunately, $\mathcal{Y}$ is a discrete set and therefore inefficient to optimize over. Thus, we perform a convex relaxation and solve $\arg\max_{\widetilde{y}\in\text{convhull}(\mathcal{Y})} f_{\widehat{\theta}}(x, \widetilde{y})$, where $\text{convhull}(\mathcal{Y}) = \{z \in [0, 1]^{40\times4} : \sum_{j=1}^4 z_{i,j} = 1\ \forall i\}$. While optimizable using multiplicative weights, the solution to this relaxed problem may not result in an integral solution (i.e., not in $\mathcal{Y}$). To effectively round a $\widetilde{y} \in \text{convhull}(\mathcal{Y})$ to a solution in $\mathcal{Y}$, we begin by naively rounding $\widetilde{y}_i \to \arg\max_{j=1,2,3,4} \widetilde{y}_{i,j}$. Then, starting with this $y_0 \in \mathcal{Y}$, we attempt all single nucleotide changes and greedily choose the swap that increases $f_{\widehat{\theta}}$ the most. We then repeat until convergence.

Input and Encoding We encode string representations of the peptide and aptamer molecules into numerical or matrix-based representations as input to our models. We use one-hot encoding for aptamers as a 40×4 matrix in all experiments. The one-hot encoding for peptides is an 8x20 matrix. We additionally encode the peptides using the BLOSUM62 original features matrix, which is a pairwise substitution matrix. We encode an amino acid by its particular row in the BLOSUM62 matrix, which is a vector of length 20. The resulting BLOSUM encoding for a peptide is an 8x20 matrix.

Network Architecture We experiment with four architectures that use a combination of linear and convolutional layers. (1) LinearBaseline: this network includes one linear layer with input of an encoded aptamer and peptide that are flattened and concatenated. (2) ConvBaseline: this network is similar to (1) except that it includes one hidden convolutional layer and one linear layer, separated by a maxpool layer. (3) LinearTwoHead: this network includes three linear layers that process an encoded aptamer and two linear layers that process an encoded peptide. The result is then concatenated and processed by one linear layer. (4) ConvTwoHead: this network includes two convolutional layers that process the peptide and two convolutional layers that process the aptamer. The result is then concatenated and processed by a linear layer. All layers are separated by a maxpool layer.

For all architectures, we use a batch size of 64 samples, a stochastic gradient descent optimizer, and binary cross entropy loss, and we decay the learning rate by a factor of 0.9 every 10 epochs. Our models were implemented in PyTorch. For each of the networks, a preliminary hyperparameter search of α and β values showed that $\alpha = 1$ for linear networks, and $\alpha = 7$ for convolutional networks seem to perform better than values of $(0.1, 1, 10, 100, 1000)$.

3 Results

We measure our model performance using receiver operating characteristic (ROC) curves and AUC numbers, based on two of our experimental replicate positive datasets in combination with both respective negative replicate data and randomly generated NNK data, across 2 encoding styles and 4 network architectures. The plots for replicate 4 are shown in Figure 1, and the plots from replicate 3 are shown in Figure 2 in the appendix. For the AUC numbers, we report a 95% confidence interval of AUC ±0.005 using 25k samples.

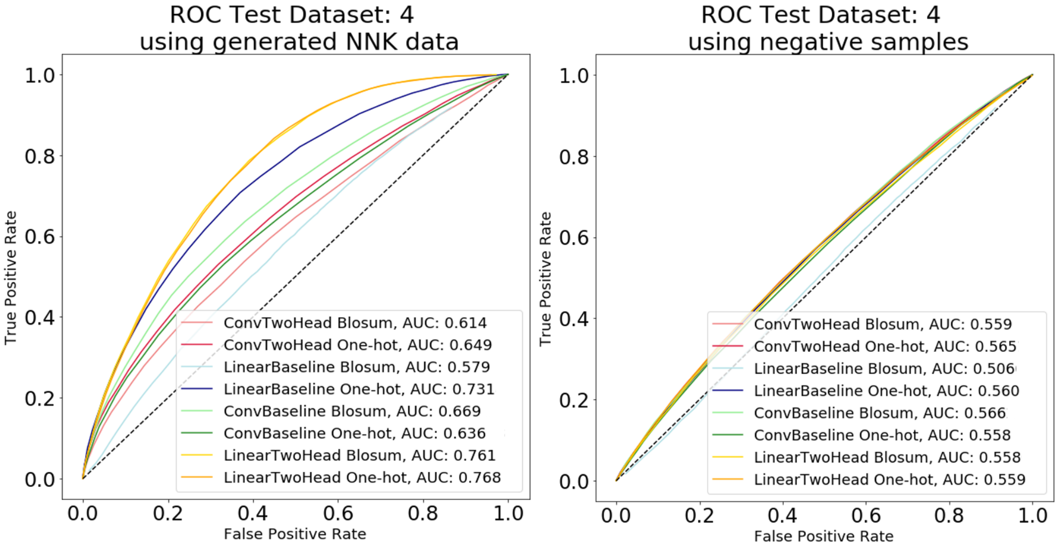

Figure 1: Our results demonstrate that LinearTwoHead network performs the best across all tasks. The choice of encoding does not seem to affect performance significantly, though BLOSUM encoding results in better performance on more complicated networks.

4 Discussion

This work introduces new datasets that help us predict binding affinity and demonstrate that our models can make non-trivial binding affinity predictions. We also explore convex optimization techniques that computationally generate aptamers for target peptides. To date, we have implemented two such techniques that use an approach based on stochastic gradient descent to search for aptamers. One technique finds aptamers that, according to our networks, bind with higher affinity than those associated with a target peptide in our original dataset. We plan to explore more robust optimization techniques. However, our research demonstrates that even when relaxing search problem constraints to those addressable using convex optimization, we achieve promising results. Finally, to understand our networks' capacity to automate the SELEX pipeline, we must perform additional experiments to validate whether computationally generated pairs can be biologically validated. This paper presents a baseline in predicting binding affinity and introduces required resources to ensure that we can reliably proceed in that direction.

References

[1] Cory A Brennick, Mariam M George, William L Corwin, Pramod K Srivastava, and Hakimeh Ebrahimi-Nik. 2017. Neoepitopes as cancer immunotherapy targets: key challenges and opportunities. Immunotherapy, 9(4):361–371. PMID: 28303769.1.0.2

[2] Tuerk C, Gold L. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. Science (80-). 1990.

[3] Ellington AD, Szostak JW. In vitro selection of RNA molecules that bind specific ligands. Nature. 1990.

[4] Darmostuk M, Rimpelova S, Gbelcova H, Ruml T. Current approaches in SELEX: An update to aptamer selection technology. Biotechnology Advances. 2014.

[5] Davis JH, Szostak JW. Isolation of high-affinity GTP aptamers from partially structured RNA libraries. Proc Natl Acad Sci USA. 2002.

[6] Li BQ, Zhang YC, Huang GH, Cui WR, Zhang N, Cai YD. Prediction of aptamer-target interacting pairs with pseudo-amino acid composition. PLoS One. 2014;9:e86729.

[7] Zhang L, Zhang C, Gao R, et al. Prediction of aptamer-protein interacting pairs using an ensemble classifier in combination with various protein sequence attributes. BMC Bioinformatics. 2016; 17(1):225.

[8] Jianwei Li, Xiaoyu Ma, Xichuan Li et al. PPAI: a web server for predicting protein-aptamer interactions, BMC Bioinformatics volume 21, Article number: 236 (2020).

[9] Cruz-Toledo, J., McKeague, M., Zhang, X., Giamberardino, A., McConnell, E., Francis, T., DeRosa, M. C., & Dumontier, M. (2012). Aptamer Base: a collaborative knowledge base to describe aptamers and SELEX experiments. Database: the journal of biological databases and curation, 2012, bas006. https://doi.org/10.1093/database/bas006.

[10] Nov Y. When second best is good enough: Another probabilistic look at saturation mutagenesis. Appl Environ Microbiol. 2012.

5 Appendix

Additional Results Figure 2 reports the ROC curves and AUC numbers on experimental replicate 3, in combination with both negative replicate data and randomly generated NNK data, across 2 encoding styles and 4 network architectures. For the AUC numbers, we report a 95% confidence interval of AUC ± 0.005 using 25k samples.

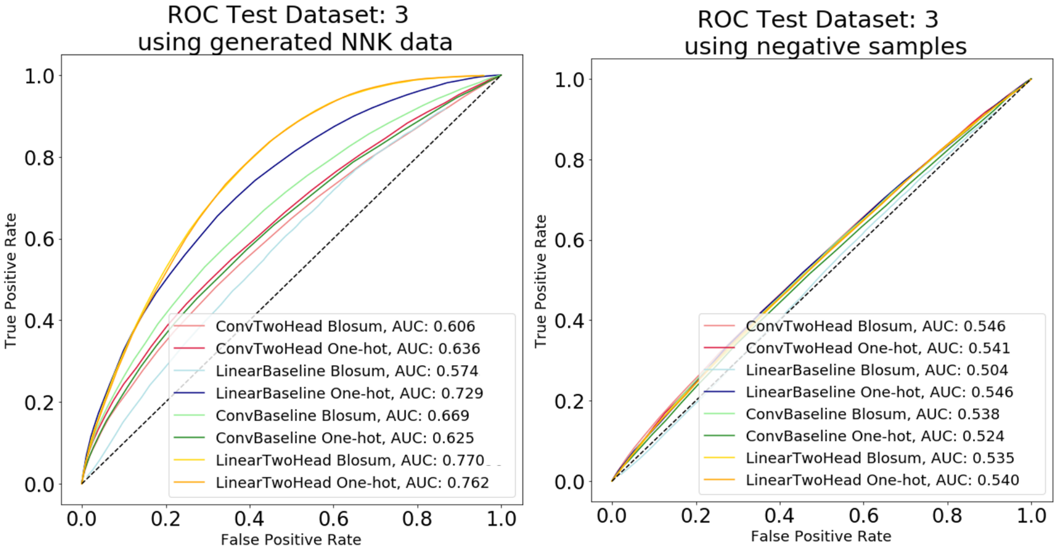

Figure 2: Like with the experimental replicate dataset 4, it seems like encoding does not play a large role in performance across networks. It seems like our LinearTwoHead architecture performs best across both tasks here as well.