JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

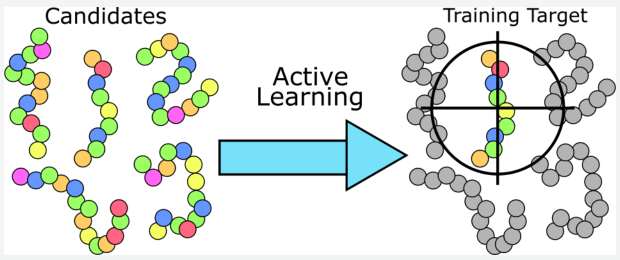
# Investigating Active Learning and Meta-Learning for Iterative Peptide Design

Rainier Barrett and Andrew D. White*

**Cite This:** *J. Chem. Inf. Model.* 2021, 61, 95−105 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

**ABSTRACT:** Often the development of novel functional peptides is not amenable to high throughput or purely computational screening methods. Peptides must be synthesized one at a time in a process that does not generate large amounts of data. One way this method can be improved is by ensuring that each experiment provides the best improvement in both peptide properties and predictive modeling accuracy. Here, we study the effectiveness of active learning, optimizing experiment order, and meta-learning, transferring knowledge between contexts, to reduce the number of experiments necessary to build a predictive model. We present a multitask benchmark database of peptides designed to advance these methods for experimental design. Each task is a binary classification of peptides represented as a sequence string. We find neither active learning method tested to be better than random choice. The meta-learning method Reptile was found to improve the average accuracy across data sets. Combining meta-learning with active learning offers inconsistent benefits.



## INTRODUCTION

Although great strides have been made in predictive computational modeling where large data sets exist, the use of computational modeling where data is scarce and/or expensive has been limited. Data expense and scarcity is the norm in materials design.[1] Computer-aided design often relies on physics-based predictive methods,[2,3] which require no data, but cannot predict complex properties like activity of a drug or refractive index of a thin film. Even if physics-based modeling is used, there is no clear mechanism to improve predictions as data is gathered in the course of testing. Thus, human intuition is often the state-of-the-art method for choosing which new peptides to test when there are small amounts of data.

Peptides are a popular target for biomaterials design.[4−9] Unlike most synthetic polymers, the amino acid sequence of a desired peptide can be controlled at the monomer level to form specific sequences.[10] Being biologically derived polymers, peptides are generally biocompatible. They are relatively easy to synthesize and some undergo self-assembly to form ordered structure spontaneously.[11,12] Creating or coating an object with functional peptides is an appealing way to create biocompatible materials with various applications in medicine,[13,14] drug delivery,[15,16] and more.[6,7,12,17,18]

Here, we apply active learning to binary classification of peptides across a variety of binary tasks like predicting solubility or activity against bacteria. We examine two standard active learning methods: query by committee (QBC)[19] and uncertainty minimization[20] with supervised learning of a deep convolutional neural network. We also examine meta-learning to see if there is a benefit in transferring knowledge from one task to another. We only allow our active learners to choose from the data set of labeled peptides (rather than generate new sequences), so that we can accurately assess the selection during training. However, these methods are ultimately evaluated not based on the peptide chosen but the accuracy of the resulting trained model. The rationale is that there are many competing design constraints in peptide design (e.g., synthetic feasibility, cost, bioavailability, etc.), and thus, it is better to have an accurate model than a finite set of examples proposed to be active.

Computer-aided design of bioactive peptides is an established field of research with a variety of approaches, spanning from quantitative structure−activity relationship (QSAR) modeling to machine learning. For example, Franco et al.[21] used a genetic algorithm to optimize an antimicrobial peptide derived from the guava plant, and Hancock et al.[22] used a QSAR model to find antibiofilm peptides. For further information on peptide design principles and modeling efforts, see Torres et al.,[23] Fjell et al.,[24] and Gromski et al.[25] The goal of this work is not to compete with these methods but to assess active learning and meta-learning as potential ways to improve iterative discovery of peptides in this setting.

**Received:** August 13, 2020
**Published:** December 22, 2020

Active learning has a history as an extension to design of experiments, which is about choosing the optimal experiments to do with limited resources. Our concern is a sequence of experiments where the results of the previous experiments influence our decision of the next, whereas optimal design of experiments is about choosing the best experiment prior to beginning and assumes a linear model. Active learning is this process of choosing the next experiment optimally.[26] It is sometimes called optimal experimental design,[27] targeted experiment design,[28] sequential design of hypotheses,[29] optimal learning,[30] and artificial intelligence scientific discovery[31] depending on the goal and problem context.

Within this framework, there are a variety of approaches depending on the form of the task model and utility function. If the task model is probabilistic, like above, utility functions which maximize information gain,[27,32] reduce model uncertainty,[33] maximize expected model change,[26] or reduce model variance[34] can be chosen. Within variance reduction methods, there are so-called A-optimal,[34] D-optimal,[34,35] and E-optimal[36] approaches which minimize the covariance matrix according to different assumptions. If the task model is nonparametric, Bayesian approaches are well suited.[28] One still has a choice of utility function (also known as acquisition function) and can, for example, maximize the expected model improvement at each experiment.[35,37] Bayesian approaches can work with recent deep learning methods through Bayesian convolutional neural networks[38] or through the connection between dropout and neural network uncertainty.[39]

If the task model is not stochastic but there is flexibility in parameter choice or multiple models to choose from, then there are a variety of pooled or consensus active learning approaches. Query By Committee (QBC) is an active learning approach that maintains a committee of models and chooses the next experiment based on where the committee disagrees most.[40] This "disagreement" is quantified using one of the various utility functions described above, for each committee member, and combining the results in some way, for example, using summation or taking the mean. There are also active learning pool methods for specific model types, for example, k-nearest neighbor,[41] logistic regression,[42,43] and linear regression with noise.[44] One can also treat the choice of model as a probability distribution, and then, the pool of models can be viewed as a stochastic or Bayesian model to use any of the previous utility functions.

There are active learning methods that are independent of the task model and instead find a characteristic set of data.[45] This can be done by clustering the data, optimizing a function which describes local variance,[46] finding regions of high uncertainty,[47] or by finding a characteristic subset of data called a "core set" via submodular function optimization.[41,48] These methods are closely related to semisupervised learning, which tries to use unlabeled data to improve a model when labeled data is sparse.[49,50]

Another closely related topic to active learning is Bayesian optimization or global optimization of black box functions. There the goal is to optimize a function while evaluating it a minimum number of times. To connect this to active learning described above, view the "expensive function" as the experiment. A surrogate stochastic model is constructed, often through bootstraping or nonparametric models, and that surrogate model is used within an active learning framework to reduce the number of the function evaluations.[51] Then, active learning is applied so that each function evaluation improves the maximum function value and/or understanding of the model. This method can be equivalent to the variance reduction techniques discussed above if the same utility functions are chosen.[52] More sophisticated active learning methods can be used with the surrogate model, including Gaussian process regression and complex portfolios of acquisition (utility) functions.[53,54] A common critique of Bayesian optimization is that it typically scales between $O(n^2)$ to $O(n^3)$, depending on approximations made, and struggles with high-dimensional surrogate models. However, this is not applicable here, since our goal is learning in systems with dozens of experiments not thousands. A recent overview on the application of Bayesian optimization to materials design can be found in Frazier and Wang.[55]

An early example of active learning in chemistry can be found with van de Walle and Ceder,[56] where a phase diagram was explored using variance minimization active learning on cluster expansions. Active learning works well, in general, with choosing cluster expansions via variance minimization.[57,58] More recently, Lookman et al.[59] explored elastic properties with ab initio calculations using a Bayesian optimization technique (Efficient Global Optimization) to optimize the ratio of three metals. The authors applied the same method to design new piezoelectrics[60] with a four-dimensional design space. Gopakumar et al.[30] also showed how active learning methods that balance exploration and exploitation can work well on two- and three-dimensional systems. This active learning approach to finding compositions with Bayesian optimization is quickly gaining popularity in the informatics community for low-dimensional systems.[61−63]

Kim et al.[64] explored active learning methods to find polymers with a specific glass transition temperature. This is a high-dimensional system because the polymers are represented with a variety of descriptors. It was made tractable by treating a list of 731 possible polymers as known a priori. At each step, the model evaluates all 731 possible polymers. An earlier example using a fixed molecule set can also be found in Warmuth et al.,[65] where candidate drug molecules were selected from a vendor catalogue with a variance minimization active learning algorithm.

Recently, Tallorin et al.[66] explored the use of Bayesian optimization for de novo design of peptide substrates. Their work is similar in that both this work and theirs used a sequence model with the goal of minimizing the number of experiments required to train a model. The differences are that they were modeling with regression, allowed for complete choice in sequence space, did not have a goal of creating accurate models, and did not use a deep learning model but a Naïve Bayes classifier. Their work is an experimentally validated demonstration that intelligently designing experiments with predictive models can reduce the required number peptides that need to be synthesized and tested.

Another topic explored in this work is meta-learning. Meta-learning is a technique for improving few-shot learning across multiple tasks. The goal, in our nomenclature above, is to make the task model depend on hyperparameters $\xi$ that are trained to work well across multiple tasks. Then, on a new task, using $\hat{\xi}$, few new examples are required to improve performance. Active learning and meta-learning are connected because both are concerned with maximizing the value of data. As the goal is to minimize the number of experiments required for new systems, we find it natural to consider this method on our data set. Examples of meta-learning for accelerating task models can be found in transfer learning,[67] few-shot learning,[68−70] and automated machine learning.[71] One application that has recently connected active learning and meta-learning is in model-free reinforcement learning.[72,73] Pang et al.[74] and Fang et al.[75] have also explored the connection between active learning and meta-reinforcement learning.

## ■ METHODS

Functional peptide data sets were prepared by mapping each amino acid to a one-hot encoded vector of dimension $[N \times 20]$, where $N$ is the length of the peptide. Five past published databases were used to generate training data sets in this work: the Antimicrobial Peptide Database (APD),[76] the collection of protein surface fragments from White et al.,[77] the PROSO II database,[78] library hits with activity against TULA-2 protein,[79] and library hits with activity against SHP-2.[80] The APD contains peptides flagged with a variety of activities, such as antibacterial, antifungal, antiviral, anticancer, and hemolytic activities. The PROSO II databse contains peptides and proteins categorized as soluble or insoluble. The TULA-2 and SHP-2 libraries contain fixed-width peptides optimized for binding to a specific target. Eight data sets were chosen from the APD: (1) "antibacterial", (2) "anticancer", (3) "antifungal", (4) "anti-HIV", (5) "anti-MRSA", (6) "antiparasital", (7) "antiviral", and (8) "hemolytic". All sequences above length 200 were excluded. This wide variety of tasks represents a range of modeling goals in peptide research. Table 1 shows the negative data sets used for training each classifier. Here, "human" refers to the aforementioned collection of human protein surface fragments.

Active learning is typically formulated as an optimization problem.[26] Consider $N$ observation pairs $x_i$, $y_i$ of features and labels, respectively, with $i \in [0, 1, ..., N]$ indicating the order of observation. Assume that $y_i$ is a class label, and $x_i$ is a feature vector of real numbers. We have a *task model*, $P_{\theta_i}(y|x)$, that assigns a probability to each class label for a feature $x$ and is defined by parameters $\theta_i$, which are updated after each new observation. In this work, $P_\theta$ is a deep-learning convolutional neural network. $\theta_i$ is updated according to some training procedure after a new $x_i$, $y_i$ pair is observed. In active learning, we choose $x_{i+1}$ from our fixed data set of $x$, $y$ pairs according to

$$x_{i+1} = \underset{x_j}{\operatorname{argmax}} A_\psi[x_j, P_{\theta_i}(y_j|x_j)] \tag{1}$$

where $A_\psi(\cdot)$ is a functional of the task model and possibly $x$. The $j$ subscript only indicates that for a given $x_j$ we must use the corresponding $y_j$ to evaluate the functional. $A_\psi(\cdot)$ can be defined by parameters $\psi$, although it is normally fixed. For example, $A_\psi(\cdot)$ could be the most uncertain point, which gives

$$x_{i+1} = \underset{x}{\operatorname{argmax}}[1 - P_{\theta_i}(\hat{y}|x)] \tag{2}$$

where $\hat{y}$ is the most likely class label for $x$ under $P_{\theta_i}$.[20] $A_\psi(\cdot)$ is called the acquisition function or utility function depending on the problem setting.[26]

Here, the task model is a deep convolutional neural network classifier, necessitating negative training examples as well as positive. One corresponding negative training data set was generated for each positive data set. Two types of negative data were generated: scrambled data with amino acid distributions identical to the "soluble" data set but length distributions identical to the corresponding positive data set, and samples

**Table 1. Positive and Negative Examples Chosen for Training the Classifiers**[a]

| Positive data set | Size | Negative data sets |
|---|---|---|
| antibacterial | 2079 | shp2[b], tula2, insoluble[c], antifungal[d], anti-HIV, anticancer[e], scrambled |
| anticancer | 183 | shp2, tula2, insoluble, antifungal, anti-HIV, antiparasital, antibacterial[e], scrambled |
| antifungal | 891 | shp2, tula2, insoluble, anti-HIV, anticancer, antibacterial, scrambled |
| anti-HIV | 87 | shp2, tula2, insoluble, antifungal, anticancer, antiparasital, antibacterial, scrambled |
| anti-MRSA | 119 | shp2, tula2, insoluble, anti-HIV, anticancer, antiparasital, scrambled |
| antiparasital | 90 | shp2, tula2, insoluble, anti-HIV, anticancer, scrambled |
| antiviral | 150 | shp2, tula2, insoluble, antifungal, anticancer, antiparasital, antibacterial, scrambled |
| hemolytic | 253 | shp2, tula2, insoluble, human[f], scrambled |
| soluble | 7028 | insoluble[g] |
| shp2 | 120 | scrambled[h] |
| tula2 | 65 | scrambled[h] |
| human | 2880 | insoluble, hemolytic, scrambled |

[a]Negative data sets were sampled to be the same size as the positive data sets. [b]It is exceedingly rare to find antibacterial peptides, so it is assumed that the SHP-2/Tula-2 binding peptides are not good examples of antibacterial peptides. [c]Insoluble peptides cannot be successful antibacterial peptides. [d]Antifungal and antibacterial activity are different mechanisms, so it is unlikely that a given peptide is both antifungal and antibacterial. [e]It is known that antimicrobial and anticancer peptides often have a similar method of action, and there is significant overlap between the two data sets. Including these data sets in one another's negative example sets might be expected to reduce overall model accuracy. This conservative choice of data set still resulted in accuracy near baseline on these two tasks in the context of meta-learning and active learning. [f]It is assumed that fragments of proteins found on surfaces of proteins found in humans are not hemolytic (kill red blood cells). [g]No scrambled data set is necessary because there are known negative examples. [h]Classifying SHP-2 and Tula-2 is in the context of fixed-length peptides so only scrambled peptides with the same length are used for classification.

from data sets which are expected to have no intersection with positive examples (Table 1). These are expected to be rather challenging negative examples, since scrambled peptides likely have many physical properties of the positive examples. Generally, the nonintersecting data sets are also naturally occurring peptides that likely are biologically relevant. To generate the scrambled negative data set, a number of peptides were generated randomly, with lengths sampled from the same length range as their respective positive set, and residues sampled from the frequency distribution of the soluble data set. The nonintersecting examples are shown in Table 1 along with rationale in the caption. To balance classes, the negative data sets were sampled down to be the same size as the corresponding set of positive examples.

Peptide sequences are encoded as one-hot vectors of dimension $N \times 20$ (with $N$ the length of the peptide), where each index in the second dimension corresponds to the index of the amino acid in the $n$th position in the alphabet of amino acids: [A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V]. Activity was encoded as a one-hot label vector of length 2, where [1, 0] indicates a positive label and [0, 1] indicates a negative label.

The task model is a convolutional neural network whose structure was partially based on that of a model from our past work in peptide modeling.[81] The model structure is shown schematically in Figure 1. The first layer of the neural network

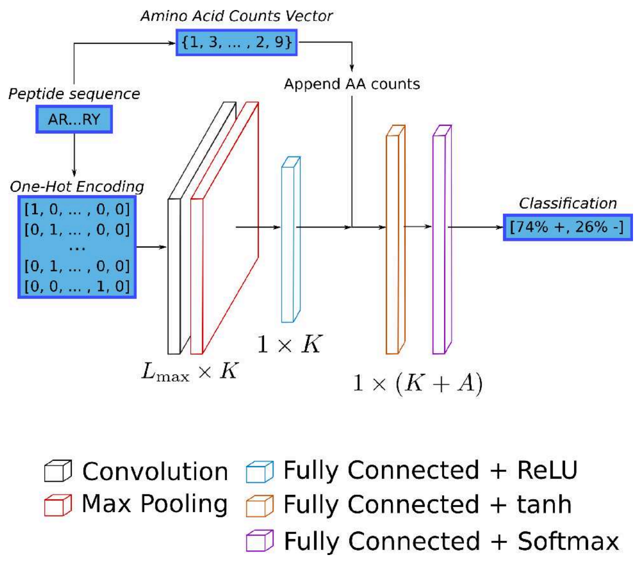


**Figure 1.** Neural network structure for active learning. Here, $L_{\max}$ is the maximum width of a peptide in the data set (although a convolution can use any length), $K$ is the number of motif classets, and $A$ is the length of the amino acid alphabet. Peptides are first translated to a one-hot encoding ($L_{\max} \times A$) and a vector of normalized amino acid counts ($1 \times N$). The output of the max pool layer is passed through one fully connected layer with ReLU activation; then, amino acid counts are appended to the output. This is then passed into two more fully connected layers with a final output dimension of 2 for positive and negative class labels. Labels below neural network layers indicate the dimensionality of the data as it passes through the layer.

is a convolutional layer with a weight matrix of dimension $[W \times A \times K]$, where $W$ and $K$ conceptually represent peptide "motif widths" and number of "motif classes," respectively, and $A$ is the length of the amino acid alphabet considered (20 for the naturally occurring peptides used here). The next layer is a mean pool layer, which captures which "motif class" is most likely in the input peptide by pooling across the peptide's length dimension, leaving a $1 \times K$ vector.

The output of the max pool layer is concatenated with the relative frequency of each amino acid in the input peptide ($1 \times 20$ vector) and standardized (0−1) sequence length. This is then fed to three fully connected (FC) layers with the ReLU activation function and then to one FC layer with softmax activation for classification, ensuring that the outgoing vector of size 2 adds up to 1, since this vector is meant to represent the likelihood of assignment to the positive or negative classes. The final output is compared with the true label vector (classification) or the true activity (regression), and loss is calculated as the cross-entropy between the two. The minimization algorithm used during training was TensorFlow's[82] Adam optimizer with parameters recommended in Kingma and Ba[83] and a learning rate of 0.001. This architecture is relatively simple compared to state-of-the art deep leanring for sequence prediction tasks of peptides. Recently, Veltri et al.[84] developed a deep learning model for predicting antimicrobial activity. As our goal is to explore active and meta-learning, we use a simpler architecture. Namely, we do not use embeddings, and we use smaller and fewer filters and fully connected layers after features instead of a recurrent neural network.

The method of uncertainty minimization is designed to favor exploration to maximize information gained per training iteration of a model. This is achieved by choosing a new training point based on some measure of the uncertainty of the machine learning model used. In this sense, it is well aligned with an eventual goal of automated experiment selection because it can minimize the number of necessary experiments to characterize a property space well. This, in turn, would lead to a reduction in operating costs and time spent performing experiments.

For uncertainty minimization training, the model described previously was used, with $W$ and $K$ both chosen to be 6. Before training, model weights are randomly initialized. The model uncertainty is calculated as the variance of the output vector of the neural network for each peptide. One peptide is then sampled, with probabilities of selection for each peptide weighted by their respective variances under the current model parameters, i.e., $p(1 - p)$. This is different than eq 1, where argmax is used instead of sampling. The chosen peptide is then used to train the neural network for one training iteration; then, an additional 16 training iterations are performed with batch size 16, sampling uniformly from all previously observed peptides for training input. Thus, in the first iteration, the first point is used for 50 training steps; then, in the second, the first and second points used an expected value of 12.5 times each, etc. This process is repeated for 50 training epochs for one training run. To gather statistics, 30 training runs of 50 epochs each were performed for each data set. We also investigated model performance after 100 training runs of 10 epochs to evaluate the few-shot learning capability of the various combinations of active and meta-learning methods used here.

The QBC method employed in this work uses the same approach as uncertainty minimization, but instead of a single task model that is trained and used for selection, a committee of 10 models is used. The nine additional models are all structured in the same way as the model used in uncertainty minimization but used different hyperparameters. They differ in the dimensions of the weights matrix used in the convolution layer, having all combinations of $3 \leq w \leq 5$ and $3 \leq k \leq 5$ along with the $W = 6$, $K = 6$ model used in uncertainty minimization. In QBC, input data are passed to each committee member (task model), and each one produces a prediction. The average variance among all models is used as the sampling weight for selecting a new training point, and training is performed in the same way as for uncertainty minimization, with the same number of epochs and training runs.

To compare these active learning methods against a base case, we use two control training methods with the same model as used for the uncertainty minimization method. The first control is a "baseline" Adam training, where the model trains on all peptides for 5000 steps with a batch size of 32. The second is a more direct comparison to the active learning methods called "random", where peptides are chosen randomly, and the model is trained in the same way as in the active learning methods (batch size 16, 16 iterations, 50 epochs).

The meta-learning method used in this work is Reptile from Nichol et al.[85] It is related to the work by Finn et al.,[86] called model agnostic meta-learning. In our notation, the goal of meta-learning is to optimize the initial parameters of the task

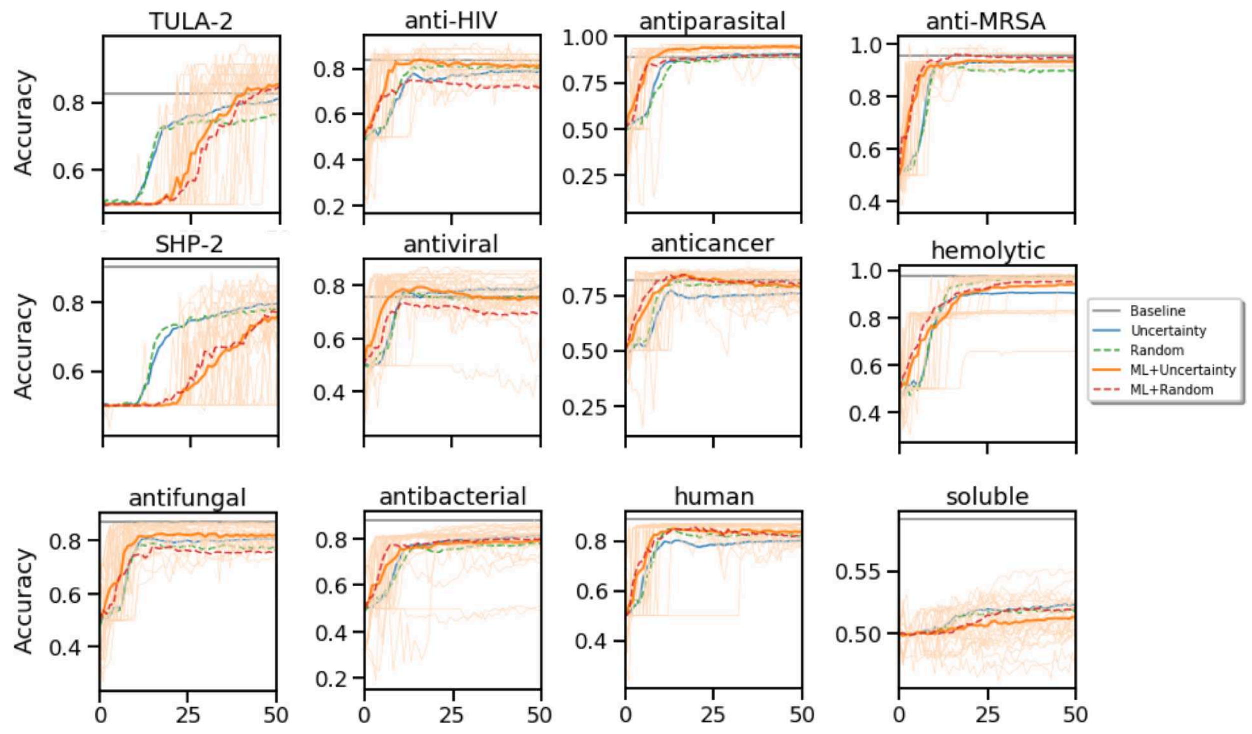


**Figure 2.** Training curves of uncertainty minimization active learning compared with baseline (gray) trained across all data points and randomly choosing examples. The $y$-axis is accuracy on withheld data. Light traces are individual 30 runs, and the dark trace is median (only one set of traces is shown). Each run has a different train/withheld split and random number generator seeds. Each subplot is a different task, arranged in increasing order of number of training points from left to right, top to bottom.

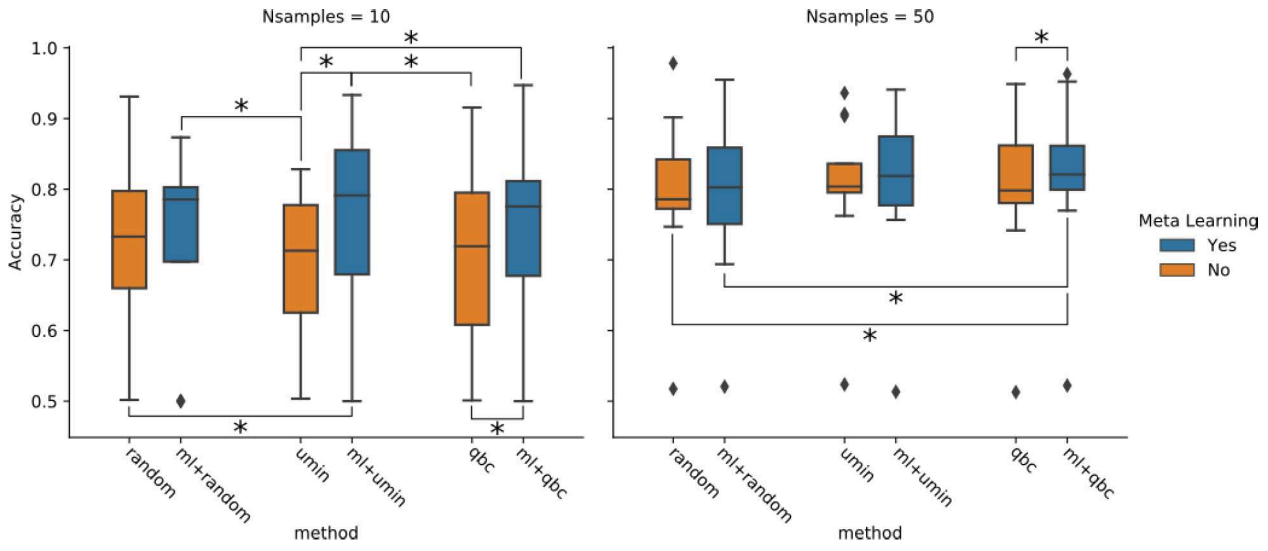


**Figure 3.** Box-and-whisker plot comparing average accuracy values after 10 and 50 training examples across all 12 data sets for five different methods explored in this work. Asterisks indicate statistically significant difference ($p \leq 0.05$) in mean accuracy, calculated using Wilcoxon's signed ranks test among the 12 average accuracy values compared between two methods. Meta-learning significantly improves few-shot performance over uncertainty minimization or QBC alone when combined with these active learning methods, but only ML+QBC shows significant performance improvement after 50 training examples. This indicates that meta-learning can be a good tool for increasing few-shot learning in settings where data is scarce. See Tables S7 and S8 for all possible $p$-value pairs.

model, $\theta_0$, to work well given a sampled task $\tau$. $P(\tau)$ is taken to be uniform across our data sets. $\theta_0$ is optimized by Adam optimization of a meta-objective function

$$E_{P(\tau)}[\mathcal{L}[U(\psi, \theta_0, \tau)]] \qquad (3)$$

where $\tau$ is the data set corresponding to a set $(x, y)$, $\psi$ are the parameters defining the active learning method $A_\psi(\cdot)$, and $\theta_0$ is the given initial task model parameters. $\mathcal{L}$ is the usual loss function, and $U$ is a stand-in for doing $J$ steps of active learning training with $A_\psi(\cdot)$. The gradient of this meta-objective requires a Hessian, but Reptile approximates this with a Taylor expansion. In this work, $J = 16$, meaning we train with active learning on 16 peptides each time with a batch size of 16. Here, 2500 meta-learning iterations were done, and then, early stopping was used to prevent overfitting. This was done

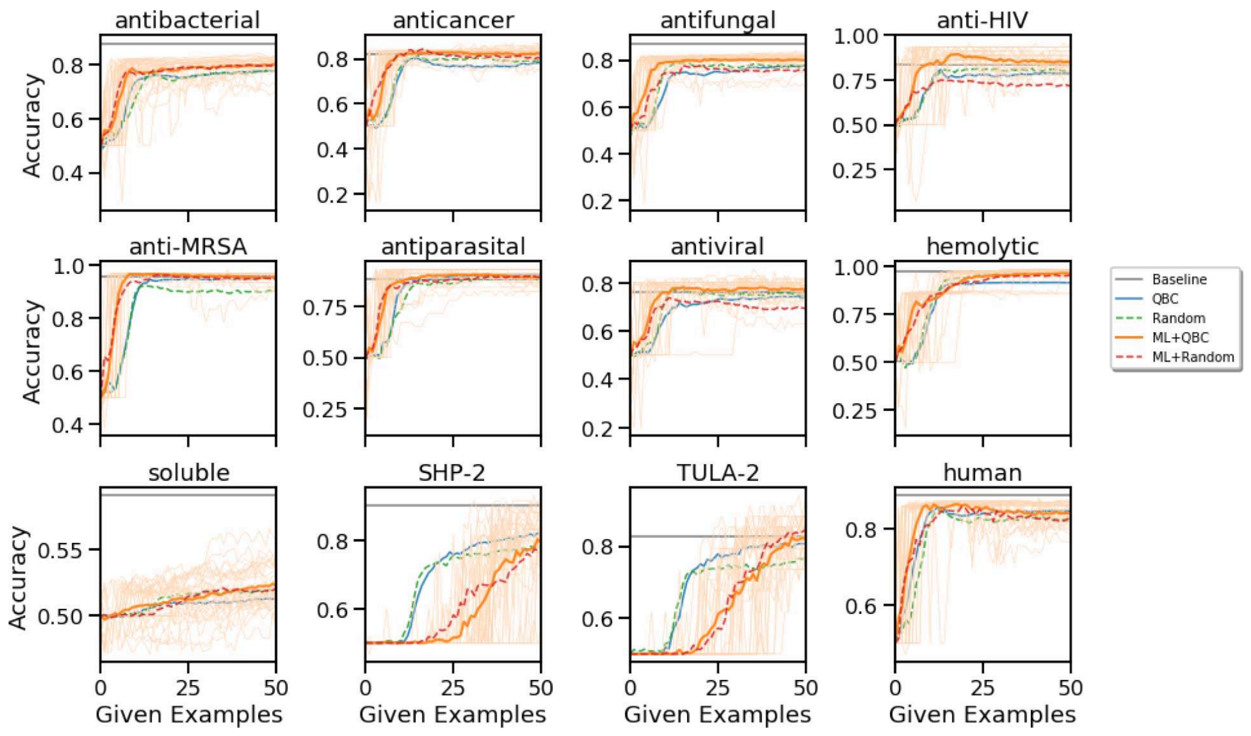


**Figure 4.** Training curves of uncertainty minimization active learning with and without Reptile meta-learning compared with baseline (gray) trained across all data points and training with randomly chosen examples with and without meta-learning. The *y*-axis is accuracy on withheld data. Light traces are the 50 individual runs, and the dark trace is the median. Each run has a different train/withheld split and random number generator seeds. Each subplot is a different task which was withheld during meta-learning. Meta-learning offers inconsistent improvements, while active learning consistently offers no improvements over random, unless paired with meta-learning (although still not consistently).

on an 80% split of the left-out data set, and then, results were reported on the 20% remaining sequences of the left-out data set.

AUC values and accuracies are reported on withheld data which was 20% of the data set size. Training curves are shown in Figures 2 and 4, and exact final values for withheld set accuracy and ROC AUC values are reported in the Supporting Information in Tables S1−S4. The accuracies and AUCs reported here are on the withheld data set from training. During meta-learning, the location of active/inactive labels (first or second index) was swapped between tasks to prevent overfitting to active being in one position or another. This gives minimal zero-shot accuracy but helps illustrate the rate of training, as shown in Figures 2 and 4. The minimized loss function was cross entropy between label probabilities and true labels. Error bars and individual traces are different due to split differences of data and random number initial parameter seeds.

## ■ RESULTS AND DISCUSSION

Figure 2 shows the active learning, meta-learning, and baseline results across the 12 data sets for the uncertainty minimization active learning strategy. The baseline results (gray) show that the convolutional neural network provides reasonable results across the range of modeling tasks despite its simple architecture. The subpanels are arranged in increasing order of training set size, which were 65, 87, 90, 119, 120, 150, 183, 253, 891, 2079, 2880, 7028, respectively. Overall, training set size does not correlate to learning performance. The solubility task showed the worst overall performance despite being the largest data set. Solubility classification is challenging because many of the training examples are folded proteins, requiring long-range sequence correlations to model properly. The length range of the solubility set is large, ranging from 19 to 198 amino acids. However, the ranges of the antifungal (2−143) and antibacterial (2−183) sets are similar, so this is not likely the source of difficulty. Indeed, solubility prediction is known to be a difficult task, with state-of-the-art models achieving reported accuracies between 0.52 and 0.77.[87,88] Our simple convolutional neural network has an accuracy of 0.59, which is within this range.

Figure 2 also shows the comparison of choosing peptides randomly and choosing peptides for which the model has maximum uncertainty (uncertainty minimization). Uncertainty minimization (blue line) is not better than choosing randomly (dashed green), in general. It is sometimes worse and sometimes better in the long term, but for few-shot learning of up to 10 training examples, it is not significantly different from random choice (Figure 3).

To assess the effect of meta-learning on reducing the experiment number, it was evaluated both alone and in combination with active learning. Results from combining meta-learning with uncertainty minimization are shown in Figure 2. Meta-learning consistently improves initial gains in accuracy, except in the soluble, SHP-2, and TULA-2 tasks. This is likely because these tasks are quite different from the other nine, so feature reuse is less important. Note that meta-learning results were controlled for label correlation due to data set overlap by randomly swapping labels (negatives become positive and vice versa) between meta-training trials. Thus, although many negative data sets overlap—as do the positive sets in the cases of anticancer and antibacterial—meta-learning is unable to overfit to the labels of any one set.

Combining uncertainty minimization with meta-learning only sometimes increases accuracy over random choice but

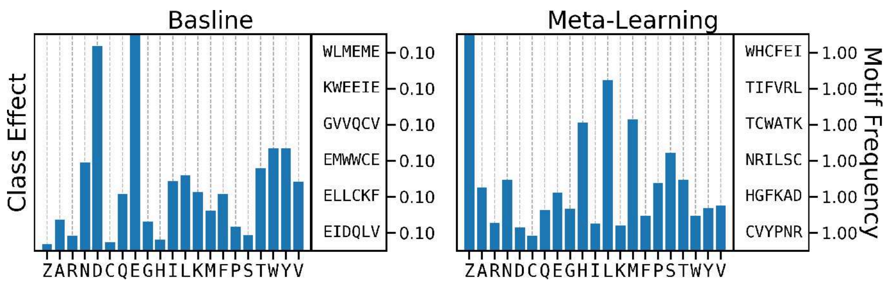


**Figure 5.** Features across tasks for baseline model. The barplot shows the partial derivative of probability of activity with respect to amino acid count averaged across training data. This gives importance for assigning label. The right side of the plots shows the maximum magnitude weight in the convolution, which roughly corresponds to the most attended motif in the sequence. The *y*-axis label on the right side shows the frequency of this motif across 50 training iterations. "Z" is the normalized length of the peptide.

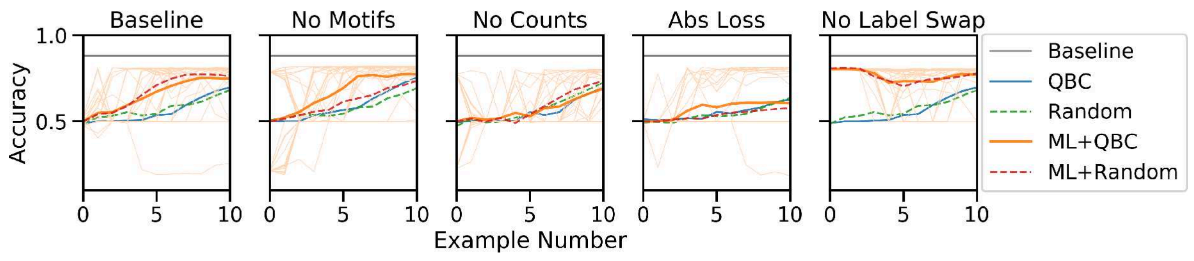


**Figure 6.** Training curves for different model choices, using the antibacterial data set. Baseline is the same in each panel, and the baseline subplot is the model as presented in text. "No Motifs" has the convolution layers removed. "No Counts" has amino acid counts removed. "Abs Loss" uses an absolute difference loss instead of cross-entropy. "No Label Swap" means that labels were not swapped during meta-learning so that zero-shot accuracy is maximized. These results show that meta-learning is not always better but is consistently a good choice for few-shot learning.

significantly improves few-shot accuracy on average after 10 training examples (Figure 3). In some data sets, final accuracy approaches or even exceeds baseline levels with less than 25 examples, whereas the baseline is trained on all data. This demonstrates the advantage of using meta-learning.

Receiver operator characteristic (ROC) curves provide an accounting of the balance between type I and type II errors. This is important for peptide activity because, due to the large design space, false positives are more detrimental to a model's usefulness. The area under curve (AUC) of a ROC curve gives a scalar representing the quality of the ROCs. Note that here we enforced balanced classes (same number of positive/negative examples in the training set). This ensures that accuracy and ROC are good measures of performance, while other metrics become necessary in cases with unbalanced training sets. The ROC AUCs are reported in the Supporting Information in Tables S1 and S2 and Figure S1. As observed in Figures 2 and 3, there is no significant gain to be had in using uncertainty minimization active learning. Also, 50 examples are not enough to match the baseline models without meta-learning. The source of variance in this work is because there are many ways to choose 50 peptides from the data sets, and the tasks of the data sets are disparate. Also, some data sets are small. Significant exceptions are the SHP-2 and soluble data sets, which show poor performance with limited examples relative to baseline models. In particular, SHP-2 seems to require the full data set to achieve good accuracy. This may be due to the importance of motifs in this data set[89] and lack of feature reuse between data sets due to its unique task (binding affinity with specific enzyme).

The QBC training results are reported in Figure 4. QBC does not consistently provide better performance than random choice or uncertainty minimization. QBC significantly improves with meta-learning. As shown in Figure 3, QBC seems to have the best few-shot and final performance when combined with meta-learning, although uncertainty minimization is as good for few-shot only. The Supporting Information contains tabulated AUCs and accuracies in Tables S1−S8.

Overall, meta-learning usually improves accuracy when combined with active learning methods across these data sets, as shown in Figure 3. Recent analysis of meta-learning has shown mixed results across tasks. Raghu et al.[90] showed that model agnostic meta-learning methods like Reptile only learn to reuse features across tasks. To assess how feature reuse can be applicable in these data sets, we performed a basic sensitivity analysis in Figure 5 which gives insight into the various features used in the modeling. Figure 5 shows the features for the baseline model on the antibacterial data set and the zero-shot features for meta-learning. Note that all motif frequencies are 1.0 since this is the meta-learned parameters, not a realization of training. The results show that meta-learning does not have the same features found in the baseline model, although some of the important amino acids are shared (N, C, H). Some of the amino acids within the motifs are shared, but they are not identical.

To ensure our that conclusions about meta-learning and QBC being preferred for peptide design are robust, we explored four alternative model choices. These are shown in Figure 6 for only the antibacterial task (although meta-learning traces were trained on all but antibacterial). The second subplot is ablation of the motif convolutions, i.e., using only amino acid count vectors for training. Without the motifs, accuracy is roughly the same, and only the combination of QBC with meta-learning is advantageous over random choice. Training with only motifs (convolutions) and no amino acid counts reduces performance, in general, and entirely removes the benefits of meta-learning, active learning, and the combination of the two. Using the original model structure but switching from a cross-entropy loss to absolute error loss reduces accuracy for all methods. The last plot shows what happens if we remove the label swapping during meta-learning, which is used to reduce overfitting to "active" peptides. Zero-shot performance is improved, due to good correlation of activity in peptides across tasks. Meta-learning is preferred in this setting, when it is known that the class labels of active vs inactive can be reused across tasks. These results indicate that the benefits of meta-learning paired with active learning can be sensitive to the model structure used. In particular, if the chosen loss function is poor, few-shot accuracy improvement may be impossible. The benefits of meta-learning combined with active learning seem to depend on the neural network features, as evidenced by the difference between the "No Motifs" and "No Counts" subplots of Figure 6.

Finally, to evaluate whether beta calibration[91] interacted favorably with these methods, the uncertainty minimization trials were repeated with beta calibration of the network output. In this case, beta calibration fitting is done separately after each training iteration, i.e., once per choice of new training point(s). The intent of beta calibration is to ensure that the scores output by the neural network reflect the actual probabilities that a given point belongs to the positive class. However, beta calibration did not have an appreciable effect on the accuracy or training efficiency (Figure S2).

We assess statistical significance in Figure 3. The box-and-whisker plots show average accuracy values for the 10-example and 50-example training runs described in the Methods section. We performed Wilcoxon's signed ranks test on the 12 training accuracy values for a given method, comparing between all possible pairs of the six methods explored here, obtaining $p$-values for each pair. Asterisks are shown in Figure 3 for pairs with $p \leq 0.05$. Methods with no asterisk were not significantly different. Exact values for all $p$-values for accuracy can be found in the Supporting Information in Tables S7 and S8. We performed a similar analysis for the average AUC values for each method, also found in the Supporting Information in Figure S1 with $p$-values in Tables S5 and S6. We can see from Figure 3 that after 50 training examples nearly all the methods explored here are not significantly different on average from random choice (with or without meta-learning). In the few-shot case, with only 10 training examples, the differences between methods are more pronounced. In this case, combining meta-learning with either uncertainty minimization or QBC yields significantly better average accuracy than either active learning method alone. This supports the hypothesis that meta-learning can enhance the accuracy of few-shot learning across multiple tasks, even for a relatively simple convolutional neural network.

## ■ CONCLUSIONS

This work has presented a data set of 12 different peptide classification tasks and explored active learning and meta-learning strategies for predicting peptide activity on them. The simple deep convolutional neural network used here offers greater than 85% accuracy across all tasks except soluble tasks, which is known to be a difficult task and was within reported ranges for other state-of-the-art models. We expect that more complex models with attention, more layers, and long-range interactions in sequence space could improve the accuracy. The two active learning strategies explored here were not found to provide significant improvements over sampling peptides randomly. Meta-learning was found to improve few-shot accuracy with 10 training examples only when combined with active learning. Here, every meta-learning method performed significantly better on average than uncertainty minimization with no meta-learning. Both meta-learning with uncertainty minimization and meta-learning with QBC were significantly better than either active learning srategy alone. In contrast, after 50 training examples, only meta-learning with QBC was found to significantly increase average accuracy over random choice, indicating that the benefits of meta-learning are short-lived but well-suited to a data-scarce context. These conclusions also hold in the zero-shot learning setting, but model ablation shows that they are dependent on loss choice and model features. This work provides a new peptide multitask data set and benchmark results for standard active learning and meta-learning methods and shows that meta-learning could have significant benefits for experimental settings where data is scarce.

**Data Availability.** The data sets used in this work are available in the following GitHub repository: https://github.com/ur-whitelab/peptide-ai.

**Code Availability.** The code used to produce the results shown here is available in the following GitHub repository: https://github.com/ur-whitelab/peptide-ai.

## ■ ASSOCIATED CONTENT

### Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c00946.

Tables with exact values for accuracy and AUC results for each data set and method. Statistical analysis box-and-whisker plot for AUC. Training accuracy curves using beta calibration and uncertainty minimization. (PDF)

## ■ AUTHOR INFORMATION

### Corresponding Author

**Andrew D. White** − *Department of Chemical Engineering, University of Rochester, Rochester, New York 14627, United States;* orcid.org/0000-0002-6647-3965; Email: andrew.white@rochester.edu

### Author

**Rainier Barrett** − *Department of Chemical Engineering, University of Rochester, Rochester, New York 14627, United States;* orcid.org/0000-0002-5728-9074

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c00946

### Notes

The authors declare no competing financial interest.

## ■ ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grants 1764415 and 1751471 and the Maximizing Investigators' Research Award Grant #R35 GM137966 by the National Institute of General Medical Sciences under the National Institutes of Health.

## ■ REFERENCES

(1) Hutchinson, M. L.; Antono, E.; Gibbons, B. M.; Paradiso, S.; Ling, J.; Meredig, B. Overcoming Data Scarcity with Transfer

Learning. In *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017.
(2) Shi, X.; Siahrostami, S.; Li, G.-L.; Zhang, Y.; Chakthranont, P.; Studt, F.; Jaramillo, T. F.; Zheng, X.; Nørskov, J. K. Understanding activity trends in electrochemical water oxidation to form hydrogen peroxide. *Nat. Commun.* **2017**, *8*, 701.
(3) Afzal, M. A. F.; Haghighatlari, M.; Ganesh, S. P.; Cheng, C.; Hachmann, J. Accelerated Discovery of High-Refractive-Index Polyimides via First-Principles Molecular Modeling, Virtual High-Throughput Screening, and Data Mining. *J. Phys. Chem. C* **2019**, *123*, 14610−14618.
(4) Krishna, O. D.; Kiick, K. L. Protein- and peptide-modified synthetic polymeric biomaterials. *Biopolymers* **2010**, *94*, 32−48.
(5) Mart, R. J.; Osborne, R. D.; Stevens, M. M.; Ulijn, R. V. Peptide-based stimuli-responsive biomaterials. *Soft Matter* **2006**, *2*, 822−835.
(6) Zaccaria, S.; van Gaal, R. C.; Riool, M.; Zaat, S. A. J.; Dankers, P. Y. W. Antimicrobial peptide modification of biomaterials using supramolecular additives. *J. Polym. Sci., Part A: Polym. Chem.* **2018**, *56*, 1926−1934.
(7) Wang, L.; Zhang, Y.; Wu, A.; Wei, G. Designed graphene-peptide nanocomposites for biosensor applications: A review. *Anal. Chim. Acta* **2017**, *985*, 24−40.
(8) de Mel, A.; Jell, G.; Stevens, M. M.; Seifalian, A. M. Biofunctionalization of biomaterials for accelerated in situ endothelialization: A review. *Biomacromolecules* **2008**, *9*, 2969−2979.
(9) Hamley, I. W. Small Bioactive Peptides for Biomaterials Design and Therapeutics. *Chem. Rev.* **2017**, *117*, 14015−14041.
(10) Chan, W.; White, P. *Fmoc Solid Phase Peptide Synthesis: A Practical Approach*; Oxford University Press, 1999; Vol. *222*.
(11) Cui, H.; Webber, M. J.; Stupp, S. I. Self-assembly of peptide amphiphiles: From molecules to nanostructures to biomaterials. *Biopolymers* **2010**, *94*, 1−18.
(12) Chetia, M.; Debnath, S.; Chowdhury, S.; Chatterjee, S. Self-assembly and multifunctionality of peptide organogels: Oil spill recovery, dye absorption and synthesis of conducting biomaterials. *RSC Adv.* **2020**, *10*, 5220−5233.
(13) Hosoyama, K.; Lazurko, C.; Muñoz, M.; McTiernan, C. D.; Alarcon, E. I. Peptide-Based Functional Biomaterials for Soft-Tissue Repair. *Front. Bioeng. Biotechnol.* **2019**, *7*, 205.
(14) Tabatabaei Mirakabad, F. S.; Khoramgah, M. S.; Keshavarz, K. F.; Tabarzad, M.; Ranjbari, J. Peptide dendrimers as valuable biomaterials in medical sciences. *Life Sci.* **2019**, *233*, 116754.
(15) Shah, A.; Malik, M. S.; Khan, G. S.; Nosheen, E.; Iftikhar, F. J.; Khan, F. A.; Shukla, S. S.; Akhter, M. S.; Kraatz, H.-B.; Aminabhavi, T. M. Stimuli-responsive peptide-based biomaterials as drug delivery systems. *Chem. Eng. J.* **2018**, *353*, 559−583.
(16) Varanko, A.; Saha, S.; Chilkoti, A. Recent trends in protein and peptide-based biomaterials for advanced drug delivery. *Adv. Drug Delivery Rev.* **2020**, *156*, 133−187.
(17) Sanghvi, A. B.; Miller, K. P.; Belcher, A. M.; Schmidt, C. E. Biomaterials functionalization using a novel peptide that selectively binds to a conducting polymer. *Nat. Mater.* **2005**, *4*, 496−502.
(18) Zong, J.; Cobb, S. L.; Cameron, N. R. Peptide-functionalized gold nanoparticles: Versatile biomaterials for diagnostic and therapeutic applications. *Biomater. Sci.* **2017**, *5*, 872−886.
(19) Freund, Y.; Seung, H. S.; Shamir, E.; Tishby, N. Selective Sampling Using the Query by Committee Algorithm. *Mach. Learn.* **1997**, *28*, 133−168.
(20) Lewis, D.; Gale, W. A Sequential Algorithm for Training Text Classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.
(21) Porto, W. F.; Irazazabal, L.; Alves, E. S. F.; Ribeiro, S. M.; Matos, C. O.; Pires, A. S.; Fensterseifer, I. C. M.; Miranda, V. J.; Haney, E. F.; Humblot, V.; Torres, M. D. T.; Hancock, R. E. W.; Liao, L. M.; Ladram, A.; Lu, T. K.; de la Fuente-Nunez, C.; Franco, O. L.; et al. In silico optimization of a guava antimicrobial peptide enables combinatorial exploration for peptide design. *Nat. Commun.* **2018**, *9*, 1490.
(22) Haney, E. F.; Brito-Sánchez, Y.; Trimble, M. J.; Mansour, S. C.; Cherkasov, A.; Hancock, R. E. Computer-aided Discovery of Peptides that Specifically Attack Bacterial Biofilms. *Sci. Rep.* **2018**, *8*, 1871.
(23) Torres, M. D.; Sothiselvam, S.; Lu, T. K.; de la Fuente-Nunez, C. Peptide Design Principles for Antimicrobial Applications. *J. Mol. Biol.* **2019**, *431*, 3547−3567.
(24) Fjell, C. D.; Hiss, J. A.; Hancock, R. E.; Schneider, G. Designing antimicrobial peptides: Form follows function. *Nat. Rev. Drug Discovery***2012**, *11*, 37−51.
(25) Gromski, P. S.; Henson, A. B.; Granda, J. M.; Cronin, L. How to explore chemical space using algorithms and automation. *Nat. Rev. Chem.* **2019**, *3*, 119−128.
(26) Settles, B. From Theories to Queries: Active Learning in Practice. In *Active Learning and Experimental Design Workshop in Conjunction with AISTATS*, 2011.
(27) Liepe, J.; Filippi, S.; Komorowski, M.; Stumpf, M. P. H. Maximizing the Information Content of Experiments in Systems Biology. *PLoS Comput. Biol.* **2013**, *9*, e1002888.
(28) Vanlier, J.; Tiemann, C. A.; Hilbers, P. J.; Van Riel, N. A. W. A Bayesian approach to targeted experiment design. *Bioinformatics* **2012**, *28*, 1136−1142.
(29) Zacks, S. 5 Adaptive Designs for Parametric Models. In *Handbook of Statistics*; Ghosh, S., Rao, C. R., Eds.; Elsevier, 1996; Vol. *13*, pp 151−180. DOI: 10.1016/S0169-7161(96)13007-8.
(30) Gopakumar, A. M.; Balachandran, P. V.; Xue, D.; Gubernatis, J. E.; Lookman, T. Multi-objective Optimization for Materials Discovery via Adaptive Design. *Sci. Rep.* **2018**, *8*, 1−12.
(31) Buchanan, B.; Feigenbaum, E. A.; Lederberg, J. Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry. *Mach. Intell.* **1968**, *4*, 209−261.
(32) Li, X.; Guo, Y. Adaptive Active Learning for Image Classification. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, 2013.
(33) Joshiy, A. J.; Porikli, F.; Papanikolopoulos, N. Multi-Class Batch-Mode Active Learning for Image Classification. In *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AL, 2010.
(34) Mackay, D. J. C. Information-Based Objective Functions for Active Data Selection. *Neural Comput* **1992**, *4*, 590−604.
(35) Chaloner, K.; Verdinelli, I. Bayesian experimental design: A review. *Stat. Sci.* **1995**, *10*, 273−304.
(36) Flaherty, P.; Arkin, A. P.; Jordan, M. Robust Design of Biological Experiments. In *Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2005.
(37) Kapoor, A.; Grauman, K.; Urtasun, R.; Darrell, T. Active Learning with Gaussian Processes for Object Categorization. In *2007 IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007; pp 1−8.
(38) Gal, Y.; Islam, R.; Ghahramani, Z. Deep Bayesian Active Learning with Image Data. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, NSW, Australia, 2017; pp 1183−1192.
(39) Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on Machine Learning*, New York, USA, 2016; pp 1050−1059.
(40) Seung, H. S.; Opper, M.; Sompolinsky, H. Query by Committee. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. New York, USA, 1992; pp 287−294.
(41) Wei, K.; Iyer, R.; Bilmes, J. Submodularity in Data Subset Selection and Active Learning. In Proceedings of the 32nd International Conference on Machine Learning, 2015; pp 1954−1963.
(42) Hoi, S. C.; Jin, R.; Zhu, J.; Lyu, M. R. Batch mode active learning and its application to medical image classification. *Proceedings of the 23rd international conference on Machine learning* **2006**, 417−424.

(43) Guo, Y.; Schuurmans, D. Discriminative Batch Mode Active Learning. In *Neural Information Processing Systems*, 2007; Vol. *20*, pp 593−600.
(44) Yu, K.; Bi, J.; Tresp, V. Active learning via transductive experimental design. *Proceedings of the 23rd International Conference on Machine Learning*. 2006; pp 1081−1088.
(45) Cohn, D. A.; Ghahramani, Z.; Jordan, M. I. Active learning with statistical models. *J. Artif. Intell. Res.* **1996**, *4*, 129−145.
(46) Yang, Y.; Ma, Z.; Nie, F.; Chang, X.; Hauptmann, A. G. Multi-Class Active Learning by Uncertainty Sampling with Diversity Maximization. *Int. J. Comput. Vis.* **2015**, *113*, 113−127.
(47) Cohn, D.; Atlas, L.; Ladner, R. Improving generalization with active learning. *Mach. Learn.* **1994**, *15*, 201−221.
(48) Sener, O.; Savarese, S. Active Learning for Convolutional Neural Networks: A Core-Set Approach. *arXiv*, arXiv:1708.00489, 2018.
(49) Corduneanu, A.; Jaakkola, T. S. On information regularization. *arXiv*, arXiv:1212.2466, 2012.
(50) Szummer, M.; Jaakkola, T. S. Information regularization with partially labeled data. In *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, 2003; pp 1049−1056.
(51) Jones, D. R.; Schonlau, M.; Welch, W. J. Efficient global optimization of expensive black-box functions. *J. Glob. Optim.* **1998**, *13*, 455−492.
(52) Chaloner, K.; Verdinelli, I. Bayesian experimental design: A review. *Stat. Sci.* **1995**, *10*, 273−304.
(53) Hoffman, M. D.; Brochu, E.; de Freitas, N. Portfolio Allocation for Bayesian Optimization. *arXiv*, arXiv:1009.541, 2011.
(54) Shahriari, B.; Wang, Z.; Hoffman, M. W.; Bouchard-Côté, A.; de Freitas, N. An Entropy Search Portfolio for Bayesian Optimization. *arXiv*, arXiv:1406.4625, 2014.
(55) Frazier, P. I.; Wang, J. Bayesian Optimization for Materials Design. *Springer Ser. Mater. Sci.* **2016**, *225*, 45−75.
(56) van de Walle, A.; Ceder, G. Automating first-principles phase diagram calculations. *J. Phase Equilib.* **2002**, *23*, 348.
(57) Mueller, T.; Ceder, G. Exact expressions for structure selection in cluster expansions. *Phys. Rev. B* **2010**, *82*, 184107.
(58) Seko, A.; Tanaka, I. Grouping of structures for cluster expansion of multicomponent systems with controlled accuracy. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2011**, *83*, 224111.
(59) Lookman, T.; Balachandran, P. V.; Xue, D.; Hogden, J.; Theiler, J. Statistical inference and adaptive design for materials discovery. *Curr. Opin. Solid State Mater. Sci.* **2017**, *21*, 121−128.
(60) Yuan, R.; Liu, Z.; Balachandran, P. V.; Xue, D.; Zhou, Y.; Ding, X.; Sun, J.; Xue, D.; Lookman, T. Accelerated Discovery of Large Electrostrains in BaTiO3-Based Piezoelectrics Using Active Learning. *Adv. Mater.* **2018**, *30*, 1702884.
(61) Fukazawa, T.; Harashima, Y.; Hou, Z.; Miyake, T. Bayesian optimization of chemical composition: A comprehensive framework and its application to RFe12 -type magnet compounds. *Phys. Rev. Mater.* **2019**, *3*, 053807.
(62) Wen, C.; Zhang, Y.; Wang, C.; Xue, D.; Bai, Y.; Antonov, S.; Dai, L.; Lookman, T.; Su, Y. Machine learning assisted design of high entropy alloys with desired property. *Acta Mater.* **2019**, *170*, 109−117.
(63) Rickman, J. M.; Lookman, T.; Kalinin, S. V. Materials informatics: From the atomic-level to the continuum. *Acta Mater.* **2019**, *168*, 473−510.
(64) Kim, C.; Chandrasekaran, A.; Jha, A.; Ramprasad, R. Active-learning and materials design: The example of high glass transition temperature polymers. *MRS Commun.* **2019**, *9*, 860−866.
(65) Warmuth, M. K.; Rätsch, G.; Mathieson, M.; Liao, J.; Lemmen, C.Active Learning in the Drug Discovery Process. In *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, 2001; pp 1449−1456.
(66) Tallorin, L.; Wang, J.; Kim, W. E.; Sahu, S.; Kosa, N. M.; Yang, P.; Thompson, M.; Gilson, M. K.; Frazier, P. I.; Burkart, M. D.; Gianneschi, N. C. Discovering de novo peptide substrates for enzymes using machine learning. *Nat. Commun.* **2018**, *9*, 5253.
(67) Pan, S. J.; Yang, Q. A survey on transfer learning. *IEEE T. Knowl. Data Eng.* **2010**, *22*, 1345−1359.
(68) Altae-tran, H.; Ramsundar, B.; Pappu, A. S.; Pande, V. Low Data Drug Discovery with One-Shot Learning. *ACS Cent. Sci.* **2017**, *3*, 283−293.
(69) Snell, J.; Swersky, K.; Zemel, R. S. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Longbeach, CA, USA, 2017; pp 4077−4087.
(70) Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, Barcelona, Spain, 2016; pp 3630−3638.
(71) Quanming, Y.; Mengshuo, W.; Hugo, J. E.; Isabelle, G.; Yi-Qi, H.; Yu-Feng, L.; Wei-Wei, T.; Qiang, Y.; Yang, Y. Taking human out of learning applications: A survey on automated machine learning. *arXiv*, arXiv:1810.13306v4, 2018.
(72) Duan, Y.; Schulman, J.; Chen, X.; Bartlett, P. L.; Sutskever, I.; Abbeel, P.RL2: Fast Reinforcement Learning via Slow Reinforcement Learning. *arXiv*, arXiv:1611.0277, 2016.
(73) Gupta, A.; Mendonca, R.; Liu, Y.; Abbeel, P.; Levine, S. Meta-Reinforcement Learning of Structured Exploration Strategies. In *32nd Conference on Neural Information Processing Systems*, MontrealCanada, 2018.
(74) Pang, K.; Dong, M.; Wu, Y.; Hospedales, T. Meta-Learning Transferable Active Learning Policies by Deep Reinforcement Learning. *arXiv*, arXiv:1806.04798, 2018.
(75) Fang, M.; Li, Y.; Cohn, T. Learning how to Active Learn: A Deep Reinforcement Learning Approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, 2017; p 595−605.
(76) Wang, G.; Li, X.; Wang, Z. APD3: the antimicrobial peptide database as a tool for research and education. *Nucleic Acids Res.* **2016**, *44*, D1087−93.
(77) White, A. D.; Nowinski, A. K.; Huang, W.; Keefe, A. J.; Sun, F.; Jiang, S. Decoding nonspecific interactions from nature. *Chem. Sci.* **2012**, *3*, 3488−3494.
(78) Smialowski, P.; Doose, G.; Torkler, P.; Kaufmann, S.; Frishman, D. PROSO II—a new method for protein solubility prediction. *FEBS J.* **2012**, *279*, 2192−2200.
(79) Cheng, G.; Xue, H.; Li, G.; Jiang, S. Integrated antimicrobial and nonfouling hydrogels to inhibit the growth of planktonic bacterial cells and keep the surface clean. *Langmuir* **2010**, *26*, 10425−8.
(80) Sweeney, M. C.; Wavreille, A.-S. S.; Park, J.; Butchar, J. P.; Tridandapani, S.; Pei, D. Decoding protein-protein interactions through combinatorial chemistry: sequence specificity of SHP-1, SHP-2, and SHIP SH2 domains. *Biochemistry* **2005**, *44*, 14932−14947.
(81) Barrett, R.; Jiang, S.; White, A. D. Classifying antimicrobial and multifunctional peptides with Bayesian network models. *Peptide Sci.* **2018**, *110*, No. e24079.
(82) Abadi, M.; Agarwal, A.; Barham, P. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. arXiv*, arXiv:1603.04467, 2015.
(83) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. *Comput. Res. Repository* **2015**, *6980*, 1−10.
(84) Veltri, D.; Kamath, U.; Shehu, A. Deep learning improves antimicrobial peptide recognition. *Bioinformatics* **2018**, *34*, 2740−2747.
(85) Nichol, A.; Achiam, J.; Schulman, J. On First-Order Meta-Learning Algorithms. *arXiv*, arXiv:1803.02999, 2018.
(86) Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017.
(87) Khurana, S.; Rawi, R.; Kunji, K.; Chuang, G.-Y.; Bensmail, H.; Mall, R. DeepSol: a deep learning framework for sequence-based protein solubility prediction. *Bioinformatics* **2018**, *34*, 2605−2613.
(88) Chang, C. C. H.; Song, J.; Tey, B. T.; Ramanan, R. N. Bioinformatics approaches for improved recombinant protein

production in Escherichia coli: Protein solubility prediction. *Briefings Bioinf.* **2014**, *15*, 953−962.

(89) White, A. D.; Keefe, A. J.; Nowinski, A. K.; Shao, Q.; Caldwell, K.; Jiang, S. Standardizing and simplifying analysis of peptide library data. *J. Chem. Inf. Model.* **2013**, *53*, 493−499.

(90) gRaghu, A.; Raghu, M.; Bengio, S.; Vinyals, O. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. *arXiv*, arXiv:1909.09157, 2019.

(91) Kull, M.; De Menezes, T.; Filho, S.; Flach, P. Beta Calibration: A Well-Founded and Easily Implemented Improvement on Logistic Calibration for Binary Classifiers. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, 2017; pp 623−631.