Active Learning Approaches For Ligand Binding Affinity Prediction

02-750 - Final Project Report

Anushka Sinha(anushka3), Aman Virmani(avirmani), Alex Kullman(akullman), Shweta Jones(shwetasj)

GitHub Link to Code: https://github.com/sinhanushka12/02-750-Final-Project

Introduction

Background

Ligand binding results from the binding between a ligand and a target protein. Specific characteristics of these ligand binding sites on the target protein such as the shape or electrostatic property of the site facilitate binding. Ligand binding involves molecular recognition, where the ligand and protein undergo conformational changes to form stable complexes. Active learning plays a crucial role in computational drug discovery by optimizing the identification of top binders for a given target protein. Its iterative sampling process enables cost-effective screening of potential drug candidates compared to traditional exhaustive methods which can take up to multiple years.

Specific Aims

The project intends to use active learning to balance exploration, which seeks novel chemical spaces, and exploitation, which maximizes the identification of potent compounds, to accurately predict ligand binding activity with increased efficiency. The two primary learning models used for this project are a Gaussian process model and the Chemprop deep learning model. Benchmarking active learning approaches can help to identify the best labeling methods for ligand binding affinity. Query selection strategies for the Gaussian process include Upper Confidence Bound (UCB), biased-UCB, Mean, and Variance, whereas the Chemprop model employs uncertainty sampling using Monte Carlo dropout. Both the models used random sampling as a baseline for comparison. This project aims to understand the impact of batch size and uncertainty sampling on the accuracy of the models. Understanding the impact of batch size and sampling methods on the chosen models can help with future computational drug discovery studies.

Significance

Understanding ligand binding helps identify potential drug targets, such as enzymes, receptors, or ion channels involved in disease pathways. Knowledge of ligand-protein interactions guides rational drug design, facilitating the development of selective and potent therapeutics. Studying ligand binding elucidates the mode of action of drugs, providing insights into their efficacy, specificity, and potential side effects. With the increasing size of datasets, it’s becoming more important to find intelligent ways to identify potential drug candidates. Active learning has previously been demonstrated to find compounds with the most potential for further exploration while only exploring a fraction of the data space [1]. Defining performance for different active learning methods with various base learners can help inform future research and drug discovery efforts.

Data

Datasets

Two publicly available binding affinity datasets are used in this project. The dataset used for training and testing the Gaussian Process and Chemprop models is a binding affinity dataset for the target Tyrosine Kinase 2 (TYK2). It comprises the SMILES representation of 9,997 ligands and their associated binding affinity in terms of their relative binding free energy (RBFE) values which have been converted to pK_i values to quantify binding affinity. A higher value of pK_i indicates a higher binding affinity of the ligand towards the target. SMILES (Simplified Molecular Input Line Entry System) is a line notation system that represents chemical structures using concise ASCII (American Standard Code for Information Interchange) strings which are the basic character sets used to encode texts in computers. Figure 1A highlights the relatively normal distribution of the pK_i metric found in the TYK2 dataset.

The dataset used for pre-training the Chemprop model is a binding affinity dataset for a different target, Dopamine Receptor D2 (D2R). It contains SMILES of 2,502 different ligands and their associated binding affinity expressed in pK_i. Figure 1B shows the distribution of the pKi metric in the D2R dataset, which displays a positively skewed distribution. The negative pKi values found in the D2R dataset can be attributed to failed experiments whose results haven't been removed from the dataset (pKi values are not supposed to be negative).

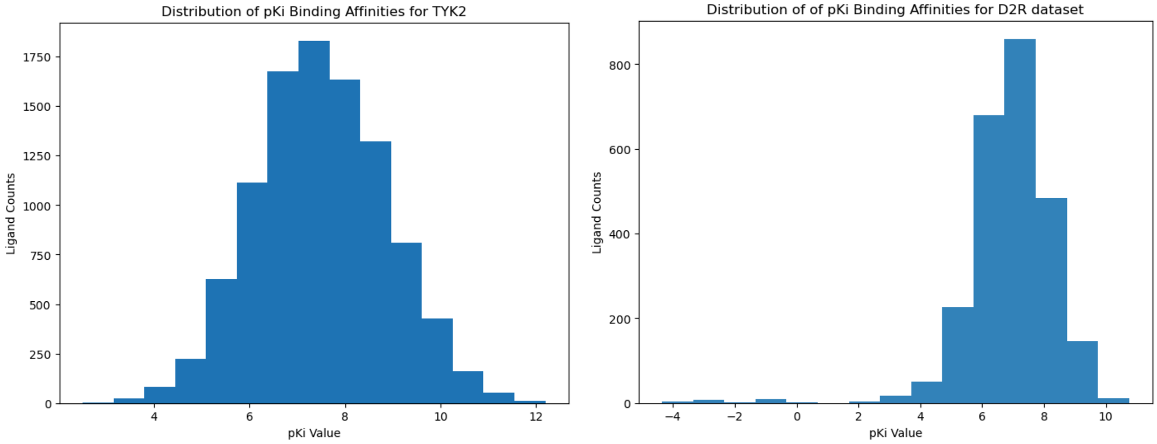


Figure 1: Histogram illustrating the distribution of pKi binding affinities. The x-axis represents the pKi values and the y-axis shows the counts of ligands at each binding affinity level. **(A)** Histogram illustrating the distribution of pKi binding affinities for the TYK2 dataset. The peak is near pKi = 7 with a bell curve shape showing a normal distribution. **(B)** Histogram illustrating the distribution of pKi binding affinities for the D2R dataset. The peak is near pKi = 7, with a noticeable positive skew and unusual negative values suggesting possible outliers.

To differentiate active compounds from inactive compounds, the top 5% of binders were identified as active, which corresponds to a pK_i threshold value of 9.8 for the TKY2 dataset. Figure 2 utilizes a UMAP to demonstrate the distribution of active and inactive compounds in the TYK2 dataset.

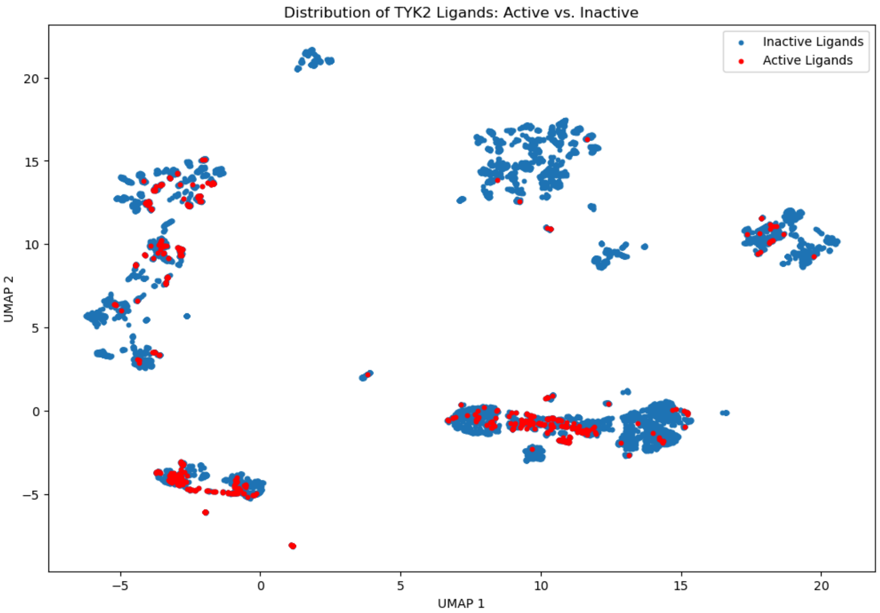


Figure 2: Distribution of the inactive and active compounds in the TYK2 dataset using UMAP where red represents the active compounds and blue represents the inactive compounds

Fingerprinting

To train a Gaussian Process Model, molecular structures of all the ligands present in the TYK2 dataset were converted to their respective Morgan fingerprints. This represents a chemical compound as a fixed-size binary vector where each bit represents the presence or absence of a feature, pattern, or substructure in the compound. Each molecular structure is broken down into constituent features or substructures, using a specific connectivity radius to determine which atoms and bonds are included. For this project, the size of the binary vector and the connectivity radius used to generate Morgan fingerprints are consistent with those employed in the referenced paper. Following their approach, the binary vector generated for each ligand has a size of 4,096 bits, with a connectivity radius of 4.

Methods

Gaussian Process Design

Gaussian Process (GP) is a supervised machine learning algorithm that can be used for non-parametric, non-linear regression, or classification tasks. Instead of modeling functions as a fixed set of parameters, GP models represent functions as distributions over functions. Typical GP models are based on two key parameters: a mean and a covariance function. The mean function returns the expected output of the model for the given input as the mean, and the covariance function (also called the kernel function) returns the similarity between any two pairs of points. The combination of these two functions provides a distance-weighted average of all trained data points, such that the weights are determined by the choice of the kernel function.

The GP design used in this study starts by utilizing the fingerprint-converted dataset to obtain an initial training set size of 1,000 randomly selected ligand fingerprints. From here, a GP is trained using this dataset, and the model performance is evaluated on all the remaining (8,997) ligands. The next data point, or batch of data points, is selected based on the query selection strategy and added to the training set. The GP model is retrained, and these steps are repeated until the training dataset contains 1,500 data points.

To perform query selection with GP, four different active learning strategies were implemented. These involved a mean acquisition function, a variance acquisition function, an upper confidence bound (UCB) acquisition function, and a biased UCB function. The mean acquisition function (only exploitation) simply returns the expected target value of each unlabeled point estimated by the GP, while the variance acquisition function (only exploration) returns the variance of each unlabeled point estimated by the GP. The UCB function sums the mean and variance, with an additional parameter (β) that tunes the amount of weight to put on the variance (**Eq. 1**). This provides a balance between exploration and exploitation of the model. For this implementation, β was tuned to place more weight on the variance than the mean. The biased UCB method starts with the standard UCB approach but multiplies the UCB value for each instance by the probability of the instance being active. This strategy models the "Expected UCB" strategy described in the paper by Rapp et al. [3], which is intended to bias the selection of queries based on a probabilistic classification. In this analysis, biasing the model increases the likelihood of selecting active ligands. This bias might make the model more useful in practical applications where active compounds are the most interesting. The probability of a compound being active was estimated using a random forest classifier. After each of the four aforementioned selection criteria were calculated, the largest n instances were chosen to be added to the training data, where n was the batch size. As a baseline comparison to these three active learning strategies, a random sampling method was also performed which randomly chooses data points until the maximum (1,500) training set size is achieved. An overview of this approach can be found in Figure 3.

$$UCB(x, \beta) = \mu(x) + \beta\sigma(x)$$
$$\beta = 2 * log(\frac{D*i^2*\pi^2}{6\lambda})$$

Equation 1: Calculating the UCB relies on adjusting β to favor either the mean or the variance of the predictions. *D* represents the dataset size, *i* represents the number of iterations, and λ controls the weight placed on the variance (smaller λ yields a larger □ and thus more importance on the variance).

Three batch sizes (1, 25, and 50) were tested on all four query selection methods to understand the tradeoff between efficiency and accuracy for more realistic modeling. Additionally, each combination of batch size and query selection method was run three times for three different starting seeds.

Due to extreme runtime constraints with performing this many trials on a large dataset, some trade-offs were imposed to ensure all results could be produced and the differences in strategies were still recognizable. This included using a smaller training set size (starting at 1,000 ligands and stopping at 1,500 ligands) as well as using a linear kernel (which likely does not represent the true distribution of the data) to eliminate the need for any optimization steps.

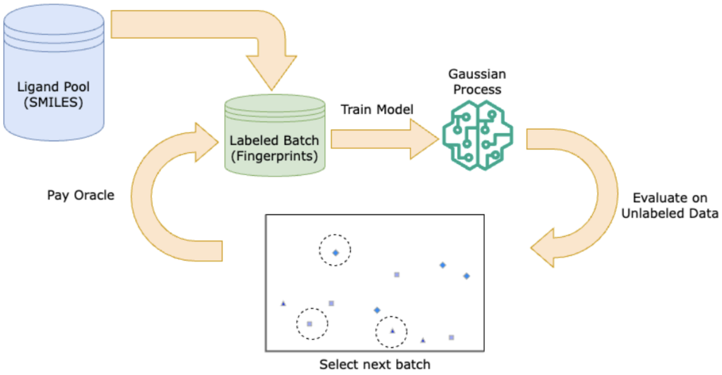


Figure 3: Schematic of the Gaussian Process Pipeline

Chemprop Design

Chemprop is a directed message-passing neural network (D-MPNN) that falls under graph-convolutional neural networks (GCNN). GCNNs learn atomic embeddings by using the local features, updating hidden representations of atoms and bonds in a chemical structure by treating them as vertices and edges, respectively. It combines the atomic embeddings into a single molecular embedding which gets processed by a feed-forward neural network. The output is the predicted property of the molecule, in this case, the pKi value for the ligands.

The Chemprop model was first pre-trained on the D2R dataset. Pretraining a neural network on a similar dataset allows for quicker optimization of the model when training the model on the dataset of interest. The model was first trained on the full D2R dataset, saved, and then loaded and used with the TYK2 dataset. This step improved the model performance on the TYK2 dataset since it was otherwise trained on a small set of instances for relatively few epochs due to runtime constraints.

The Chemprop model takes as input the SMILES form of the ligands directly and was initialized with a random training set of 1,000 instances. The general training, evaluation, and instance selection process was the same as with GP and ran until it reached 1,500 selected instances (**Fig. 4**). In each iteration, the model was trained for 30 epochs using the mean-variance estimation as the loss function.

Two different query selection methods were tested with Chemprop. Uncertainty sampling was implemented using Monte Carlo (MC) dropout as the measure of uncertainty. As a baseline for comparison, random sampling was also implemented. MC dropout is a typical method in neural networks where variational inference is approximated by taking a “sample from the posterior and predictive distributions of a given model” [2]. This is done by inactivating random nodes in the neural network and predicting the unknown target values multiple times. This means that for each instance, there are multiple forward passes with different dropout configurations, and the variance of these predictions allows for the estimation of uncertainty.

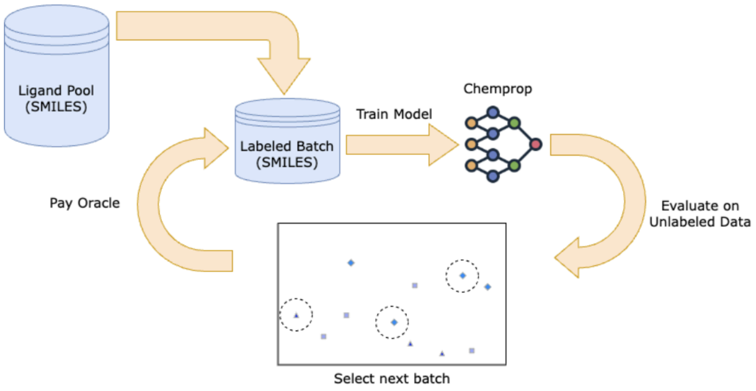


Figure 4: Schematic of the Chemprop Pipeline

Similar to the tradeoffs made for the GP model, tradeoffs were made to run the Chemprop model on the TYK2 dataset. Given the extremely high runtimes and computational power required to run the model, higher batch sizes of 50 and 100 were used. Each combination of batch size and query selection method was run three times for the same three starting seeds as in GP to ensure any randomness was consistent across both methods. Training epochs were limited to 30, even though higher epochs did result in higher accuracy. Similar to what was done in the GP, a smaller training set size of 1,000 was used and the model was trained until 1,500 instances made up the training set.

Results

GP Results

Plotting the iterative changes to the model as data was added, it is apparent that the mean acquisition function performed the worst as shown in Figure 5 by the lowest R^2 and the highest RMSE values across the three batch sizes (1, 25, and 50). Meanwhile, the best performers were the variance acquisition function with a batch size of 1, followed by the UCB acquisition function with a batch of size 1. This conclusion is highlighted in Figure 6, which compares the performance of the final model using R^2 and RMSE scoring metrics after the training set has obtained a total of 1,500 data points.

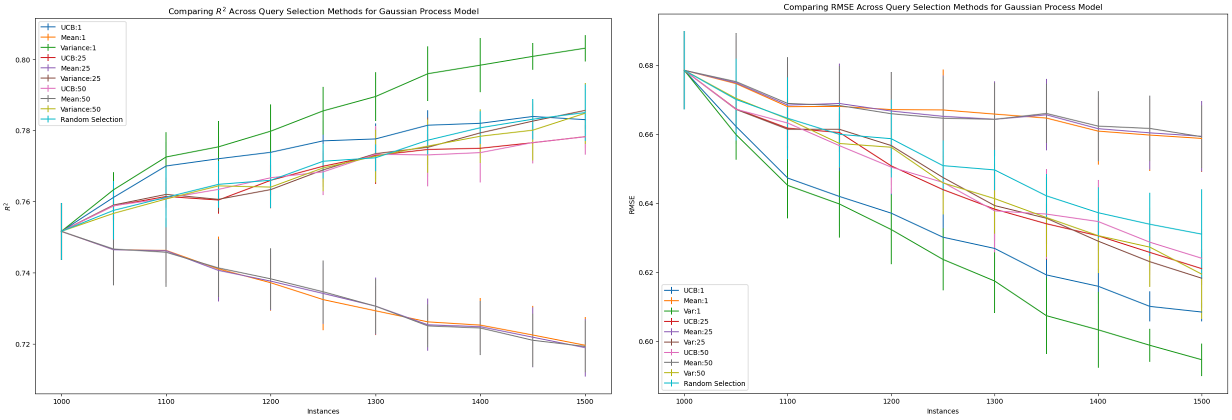


Figure 5: Comparison plots showing **(A)** R-squared (R^2) values and **(B)** Root Mean Squared Error (RMSE) values for different batch sizes and training set instances. The graphs indicate the performance of random sampling, UCB, Mean, and Variance for batch sizes 1, 25, and 50.

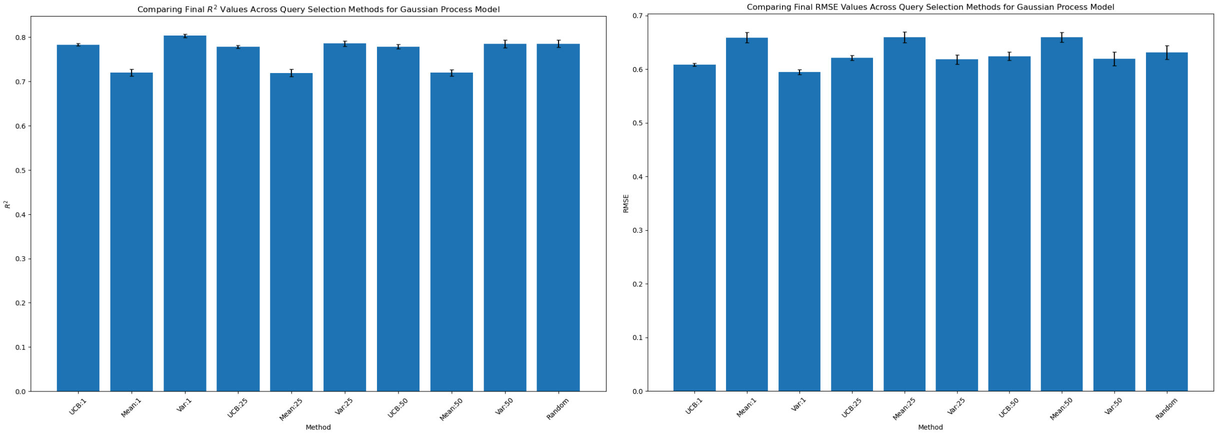
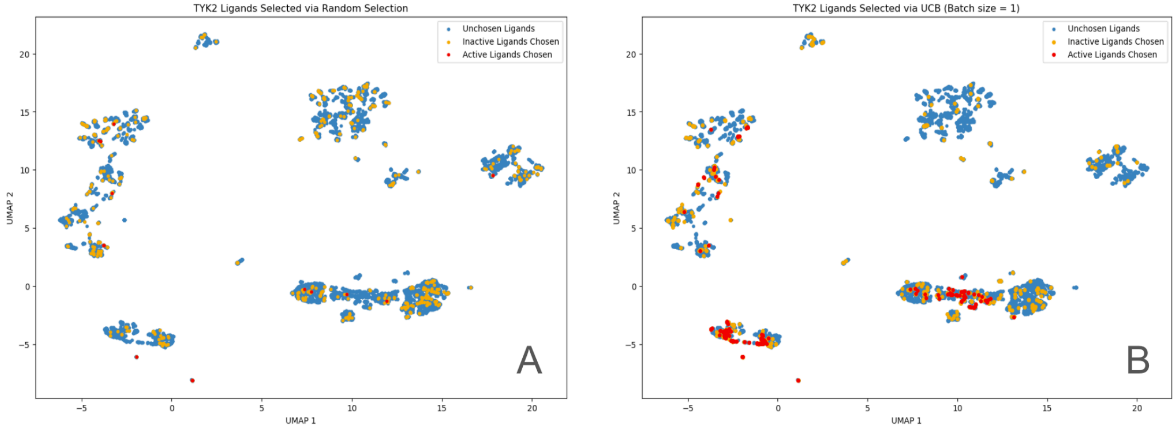


Figure 6: Comparison bar plots showing final **(A)** R-squared (R^2) values and **(B)** Root Mean Squared Error (RMSE) values for different batch sizes and training set instances. The graphs indicate the performance of random sampling, UCB, Mean, and Variance for batch sizes 1, 25, and 50.

To provide more context to the types of ligands selected by the models for each of the query selection strategies, the ligands in the dataset were denoted as either "active" or "inactive" based on the criteria described earlier. Looking at Figure 7, which highlights the active and inactive ligands chosen by the various strategies (for a batch size of 1 as it performed best overall), it's evident that the UCB and mean selection functions outperform the variance and random selection functions. The best-performing strategy in this context is the mean selection function which had an active ligand selection rate of 61.8%. The UCB, variance, and random selection functions had active ligand selection rates of 29.8%, 12.2%, and 2.8% respectively. Overall, these results highlight the benefits of query selection strategies (mainly those that choose ligands based on the highest average predicted binding affinity) to select active ligands for training a model, compared to randomly selected points across a dataset.



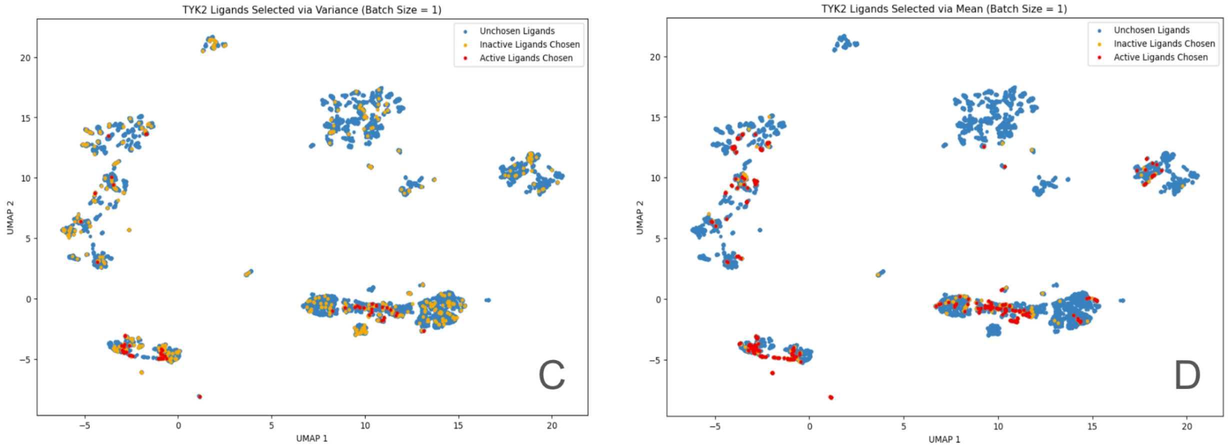


Figure 7: UMAPs displaying the number of captured active and inactive ligands by **(A)** Random, **(B)** UCB, **(C)** Mean, and **(D)** Variance for the batch size of 1. The orange dots represent the inactive ligands chosen, the red dots represent the active ligands chosen, and the blue dots represent the ligands not chosen by the sampling method for the GP Model.

With the goal of improving the model performance to select more active ligands for training (exploitation), while also selecting a wide enough distribution of points to maintain model accuracy, a biased UCB query selection strategy was implemented. The results of this strategy show that there was a slight decrease in performance (**Fig. 8**), but the selection of active ligands increased substantially compared to the unbiased UCB method (**Fig. 9**). Specifically, the biased UCB method had an active ligand selection rate of 50.8%, which was far more than the rate of the unbiased UCB method (29.8%).

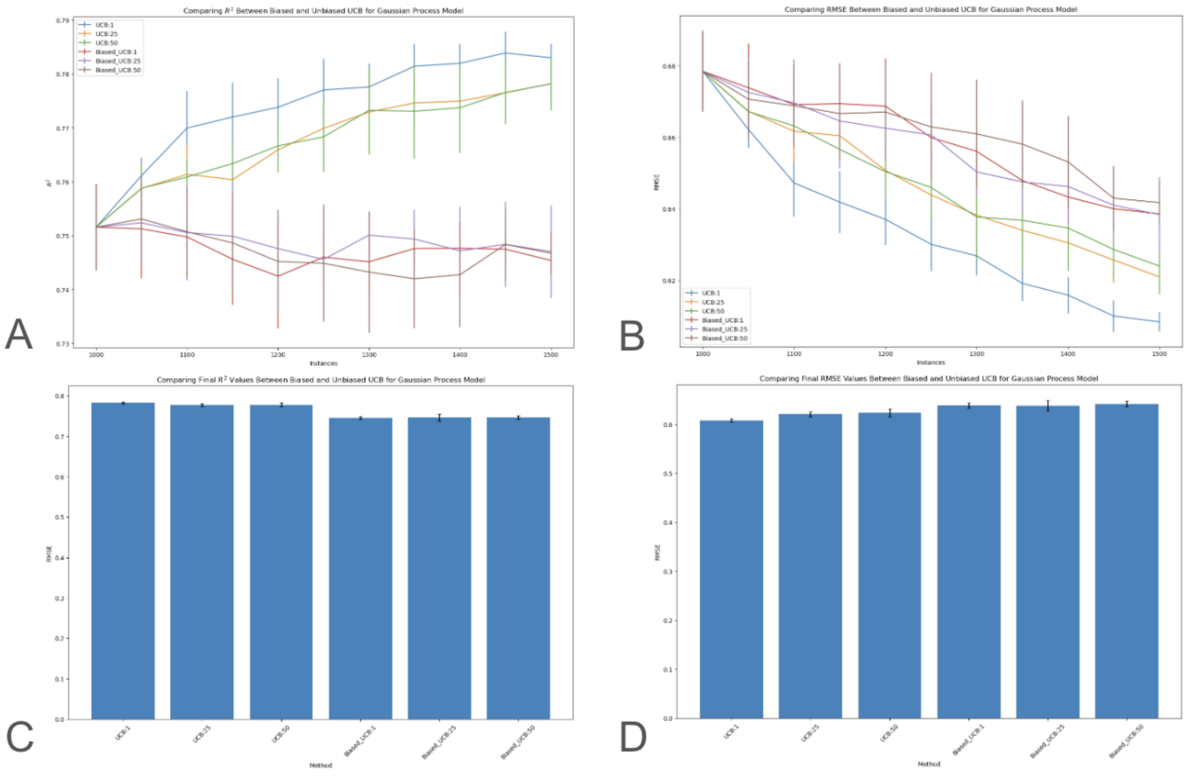


Figure 8: Comparing the performance of the unbiased and biased UCB query selection methods across different batch sizes. **A** and **B** display the RMSE and R^2 values of the GP model iteratively as data points are added. **C** and **D** highlight the final RMSE and R^2 values of the model on the unseen data after selecting a total of 1,500 ligands for training.

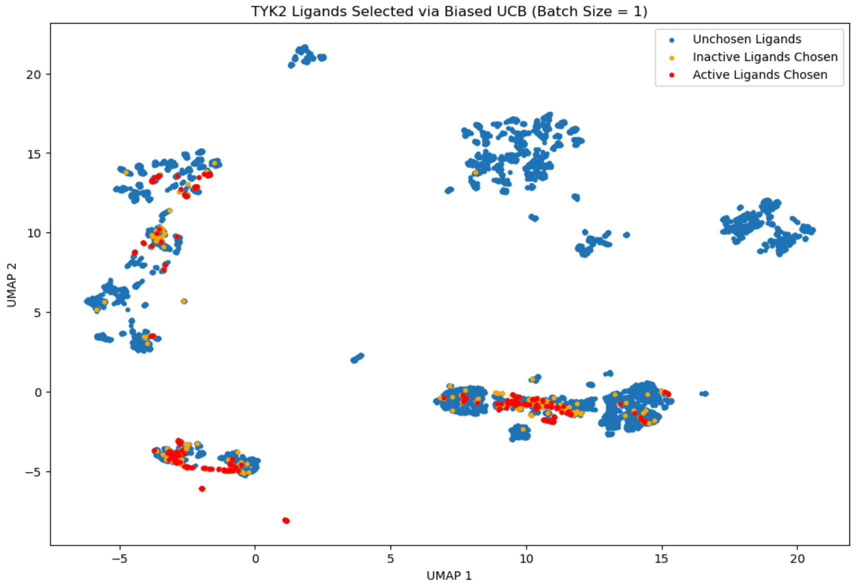


Figure 9: UMAP displaying the number of captured active and inactive ligands by the biased UCB query selection function for a batch size of 1. The orange dots represent the inactive ligands chosen, the red dots represent the active ligands chosen, and the blue dots represent the unchosen ligands for the GP model.

Chemprop Results

Figure 10 compares R-squared (R^2) and RMSE values obtained from iteratively training the Chemprop model using random sampling and uncertainty sampling (Monte Carlo dropout) as query selection strategies. It is evident from Figure 10A that random sampling maintains a lower R^2 value throughout the simulation compared to uncertainty sampling for both batch sizes of 50 and 100. It is also evident from Figure 10B that random sampling produces a higher RMSE value throughout the simulation compared to uncertainty sampling for both batch sizes. Uncertainty sampling not only starts with a lower RMSE score but also continues to decrease as more training instances are added, suggesting that it effectively leverages uncertainty to improve prediction accuracy. Moreover, Figures 11A and 11B indicate that the final R^2 value obtained using random sampling is lower than that achieved with Monte Carlo dropout. Similarly, the final RMSE value using random sampling is higher than the final RMSE with Monte Carlo dropout, demonstrating that uncertainty sampling through Monte Carlo dropout achieves better model performance in terms of both metrics.

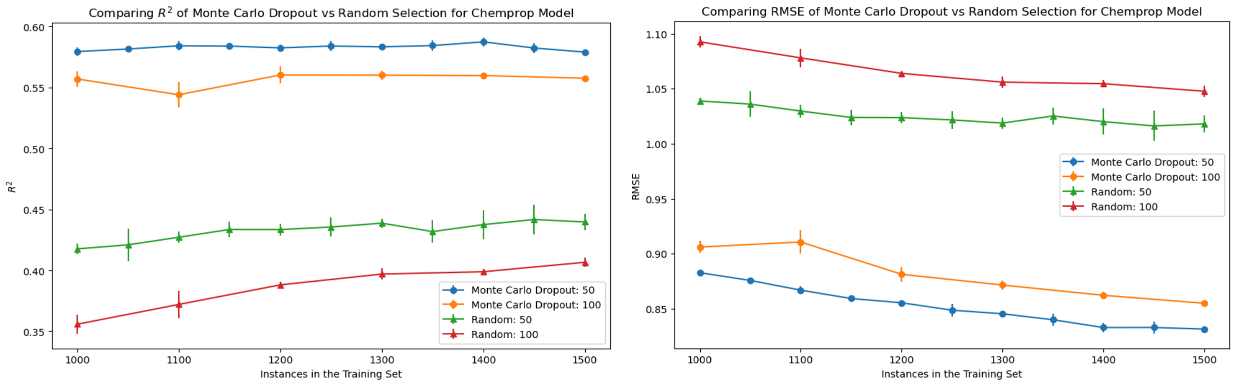


Figure 10: Comparison plots showing **(A)** R-squared (R^2) values and **(B)** Root Mean Squared Error (RMSE) values for different batch sizes and training set instances. The blue and orange lines represent uncertainty sampling using Monte Carlo dropout with batch sizes of 50 and 100, respectively, while the green and red lines represent random sampling.

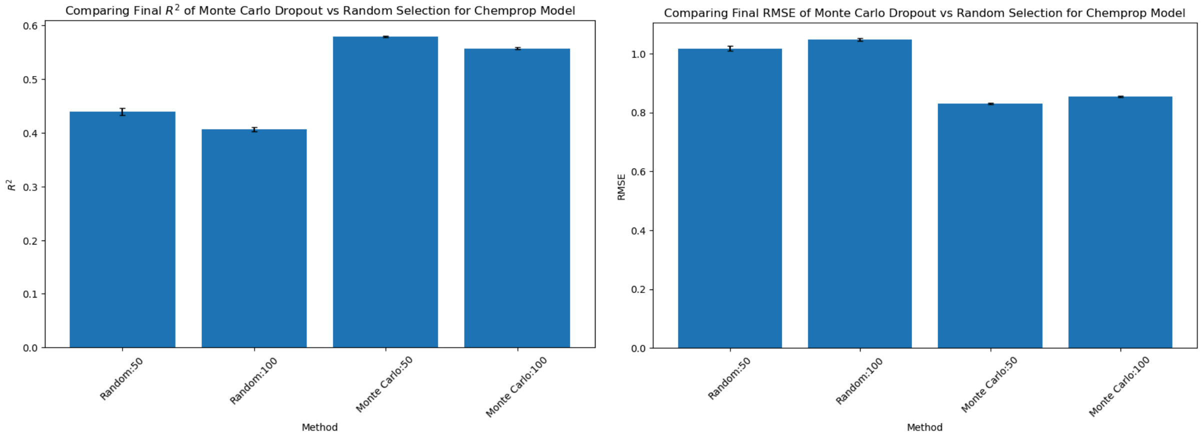


Figure 11: Bar plots comparing the final **(A)** R-squared (R^2) values and **(B)** Root Mean Square Error (RMSE) values for different batch sizes and sampling methods in a Chemprop model. The first two bars represent the results for random sampling with batch sizes of 50 and 100, while the last two bars represent the results for Monte Carlo Dropout.

Conclusion

Table 1 provides the runtime, R-squared (R^2) values, and Root-Mean Square Error (RMSE) values of all the different models with their respective query selection strategies and batch sizes. The direct comparison between these methods highlights the trade-off between runtime and higher accuracies to make better conclusions about which method is best for ligand binding prediction.

Model	Query Selection	Batch Size	Runtime (s)	RMSE	R²
Gaussian Process	Random	1	1,704.04	0.63	0.78
		25	35.74	0.63	0.79
		50	24.80	0.62	0.79
	UCB	1	3,888.62	0.61	0.78
		25	200.78	0.62	0.78
		50	124.50	0.62	0.78
	UCB- Biased	1	3,847.85	0.64	0.75
		25	206.87	0.64	0.75
		50	125.52	0.64	0.75
	Mean	1	3,863.30	0.66	0.72
		25	180.26	0.66	0.72
		50	106.58	0.66	0.72
	Variance	1	3,384.94	0.59	0.80
		25	176.77	0.62	0.79
		50	108.21	0.62	0.79
Chemprop	Random	50	11,388.56	1.02	0.44
		100	5,854.94	1.05	0.41
	Uncertainty	50	10,646.36	0.83	0.58
		100	5,970.17	0.85	0.56

Table 1: Table depicting the runtime, R-squared (R^2) value, and Root Mean Square Error (RMSE) values of all the methods chosen for this project for different batches over the same set of random seeds.

Comparison of Models

Given the results, the GP model may be a better fit for ligand binding affinity prediction. However, further analysis and testing needs to be done to make a definitive conclusion as the project scope did not make it feasible. For example, smaller batch sizes tended to perform better for Chemprop, however, it was not possible to test this as it was too computationally intensive as seen by its runtime. Other factors such as the size of the dataset or larger numbers of epochs could lead to Chemprop outperforming GP. Despite these factors, the results show that the GP model outperforms the Chemprop model in this study, both in model performance and runtime. This was expected as this was also what Gorlanta et al. observed in their comparisons [1].

Comparison of Active Learning Strategies

Based on Table 1, the variance acquisition method performs the best in producing the lowest RMSE and highest R^2 value. According to Table 1, it is clear that uncertainty sampling leads to a higher R^2 value and a lower RMSE value compared to random sampling regardless of the batch size for the Chemprop model. Consistent across both models, these results indicate that choosing the most uncertain instance results in the most accurate model when using the same amount of training data. Uncertainty sampling would be expected to outperform random sampling, which was observed with both models. However, with the GP model, the UCB selection function was expected to perform the best but the variance method had the best performance. UCB balances selecting instances to improve the model with selecting instances that the model should be confident about and is intended to take advantage of the strengths of each of those individual methods. The UCB method did perform well overall, with only the variance method performing slightly better. One explanation for this could be that the GP was slightly worse at predicting instances with the highest predicted values. This would also explain why the biased UCB, which picks more active compounds, performed slightly worse than the standard UCB.

When looking at the number of active ligands selected for GP, the mean selection function performed the best, followed by the biased UCB. This would be expected because the mean selection function selects the highest predicted values (binding affinity) and the biased-UCB selection function is intended to bias selection towards active compounds. Considering model accuracy using R^2 and RMSE values, it is evident that there is a slight tradeoff between selecting active ligands and maintaining strong model performance. However, in a real-world application, this tradeoff is likely worthwhile as the goal of drug discovery pipelines is to uncover active binding compounds. As a result, providing a model that selects training instances biased towards active ligands would likely serve well to decrease the amount of time it would take to uncover a novel active ligand.

Comparison of Batch Sizes

For the GP model, smaller batch sizes performed better for both scoring metrics. For UCB, UCB-biased, Mean, and Variance, the batch size of 1 outperformed the other batch sizes, albeit marginally. For the Chemprop model, a smaller batch size outperforms in terms of R^2 and RMSE values for random and uncertainty sampling using Monte Carlo dropout. As shown in Table 1, a batch size of 50 has a higher R^2 value than a batch size of 100 for both random and uncertainty sampling which indicates that a smaller batch size is more effective regardless of the query selection strategy. Additionally, Table 1 shows that a batch size of 50 produces a lower RMSE value than a batch size of 100 for both random and uncertainty sampling, which reinforces the idea that smaller batch sizes contribute to more accurate and consistent performance in the Chemprop model, regardless of the sampling strategy. These results across both models would be expected because smaller batch sizes generally lead to better model performance. The trade-off is that the computational resources needed for small batch sizes can be prohibitively high as seen with the Chemprop model. Choosing the optimal batch size will depend on the specific needs of the study.

Final Thoughts

The project performed a comprehensive analysis of different methods and strategies for predicting ligand binding affinity. It explored the relationship between query selection strategies and batch sizes and their impact on runtimes and accuracies. The study shows the need for strategically designed model training for identifying top binders, which potentially could save a significant amount of time and money in drug discovery pipelines.

References:

[1] Gorantla, Rohan et al. “Benchmarking Active Learning Protocols for Ligand-Binding Affinity Prediction.” https://pubs.acs.org/doi/10.1021/acs.jcim.4c00220

[2] Bench, Ciaran. "Monte Carlo Dropout: A Practical Guide." Medium, 2022, https://medium.com/@ciaranbench/monte-carlo-dropout-a-practical-guide-4b4dc18014b5#:~:text=MC%20Dropout%20provides%20a%20means,distributions%20of%20a%20given%20model.

[3]Rapp, J.T., Bremer, B.J. & Romero, P.A. Self-driving laboratories to autonomously navigate the protein fitness landscape. *Nat Chem Eng* 1, 97–107 (2024). https://doi.org/10.1038/s44286-023-00002-4