Contents lists available at ScienceDirect

Artificial Intelligence in the Life Sciences

journal homepage: www.elsevier.com/locate/ailsci

Research Article

Optimizing active learning for free energy calculations

James Thompson[a,*], W Patrick Walters[b], Jianwen A Feng[a], Nicolas A Pabon[b], Hongcheng Xu[a], Michael Maser[a], Brian B Goldman[b], Demetri Moustakas[b], Molly Schmidt[b], Forrest York[b]

[a] *Google Research, Mountain View, CA, United States*
[b] *Computation, Relay Therapeutics, Cambridge, MA, United States*

ARTICLE INFO

Keywords:
Artificial intelligence
Active learning
Machine learning
FEP
Free energy calculations

ABSTRACT

While Relative Binding Free Energy (RBFE) calculations have become a mainstay in lead optimization programs, the computational expense of performing these calculations has limited their broader application. Active learning (AL), a machine learning method used to direct a search iteratively, has explored larger chemical libraries using RBFE calculations. While AL has been successfully applied, there has not been a systematic study of the impact of parameter settings on the performance of AL. To address this gap, we have generated an exhaustive dataset of RBFE calculations on 10,000 congeneric molecules. We used this dataset to explore the impact of several AL design choices, including the number of molecules sampled at each iteration, the method used to select an initial sample, the method used to build a machine learning model, and the acquisition function that defines the balance between exploration and exploitation in the search. Our studies demonstrated that the performance of AL is largely insensitive to the specific machine learning method and acquisition functions used. In our studies, the most significant factor impacting performance was the number of molecules sampled at each iteration where selecting too few molecules hurts performance. Under the best conditions, we were able to identify 75% of the 100 top scoring molecules by sampling only 6% of the dataset. We hope that the dataset of 10K molecules will provide the basis for future studies exploring additional AL strategies. The source code and supporting data for the work are available at https://github.com/google-research/google-research/tree/master/al_for_fep.

1. Introduction

Over the last few years, several studies have been published showing the ability of relative binding free energy (RBFE) calculations to rank order the binding affinities of molecules in a congeneric series [1,2]. While this capability provides a powerful means of prioritizing synthesis in a lead optimization program, the method's utility is somewhat constrained by its computational cost. Even with speedups provided by graphics processing units (GPUs), a single calculation typically takes several hours to complete. Given these constraints and the requirement to run the calculations on GPUs, drug discovery teams are typically limited to running, at most, tens of free energy calculations in a design cycle.

To appreciate the time and cost of RBFE calculations, consider a chemical library of 10,000 molecules. Assuming that each RBFE calculation requires eight GPU hours, we would consume 80,000 GPU hours or 9.1 years of GPU calculation. While these calculations can be run in parallel across multiple GPUs, the cost of cloud computing resources could be prohibitive. For example, at $2 per GPU hour, it would cost $160K to run a set of RBFE calculations for 10,000 molecules.

One means of overcoming these time and cost limitations is a machine learning (ML) method known as active learning (AL) [3–5]. In AL for lead optimization, an ML model is used to direct an iterative search aimed at identifying the top ranked molecules while drastically reducing the number of RBFE calculations performed. The AL process begins by selecting a subset of examples from a larger dataset and using this subset to build a model to direct subsequent selections. At each iteration, the model is updated with additional examples. Search strategies generally try to achieve two main goals: “exploration” and “exploitation”. The goal of exploration is to broadly sample the data to attain a more global understanding of the search landscape and potentially identify new promising areas. The goal of exploitation strategies is to focus on promising local regions to identify the highest scoring examples. Most strategies will attempt to achieve both goals to some degree but will prioritize one over the other.

When applying AL with RBFE calculations, we typically begin with a virtual library consisting of several thousand congeneric molecules, which were enumerated based on a few chemical reactions. The application of AL to identify the most promising molecules in the library typically follows the following steps (illustrated in Fig. 1).

[*] Corresponding author.
E-mail address: thompj@google.com (J. Thompson).

https://doi.org/10.1016/j.ailsci.2022.100050
Received 26 October 2022; Received in revised form 1 December 2022; Accepted 2 December 2022
Available online 5 December 2022

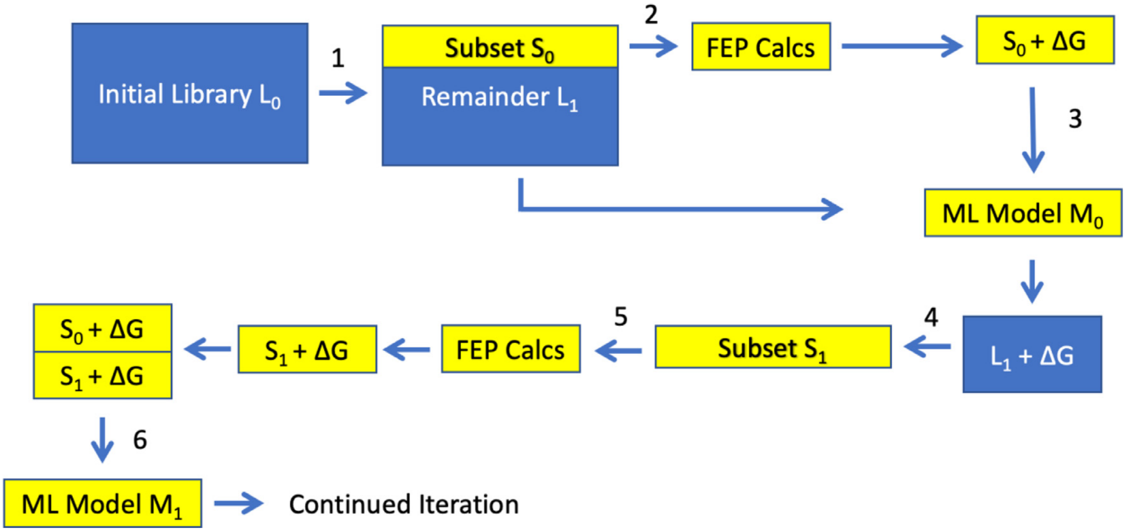


Fig. 1. A schematic view of the AL process

(1) A library of molecules, L_0, is sampled to produce an initial set of molecules S_0.
(2) The molecules in S_0 are removed from L_0 to produce L_1, and RBFE calculations are carried out for each molecule in S_0. At this point, we have a set of chemical structures for the molecules in S_0 and their associated ΔG values.
(3) A machine learning model M_0 is trained to predict ΔG for molecules in S_0 from their chemical structures.
(4) The model predicts ΔG for the molecules in L_1. These predictions guide the selection of a new subset, $S_{1,}$ from the remaining molecules. The molecules in S_1 are removed from L_1 to produce L_2.
(5) RBFE calculations are performed for the molecules in S_1.
(6) The molecules and ΔG from S_1 are added to S_0 to produce a new training set $S_{0,1}$. A model, M_1, is trained to predict ΔG for molecules in $S_{0,1}$.
(7) Steps 4–6 are repeated for a predetermined number of iterations.

In an RBFE calculation, a thermodynamic cycle is used to estimate the binding free energy of a protein-ligand complex. The result of this calculation is the relative difference in binding free energy between two protein-ligand complexes. This difference in binding free energy is typically referred to as ΔΔG. By comparing ΔΔG to an experimentally measured binding free energy, one can arrive at an estimate of the binding free energy, ΔG. In the interest of simplicity, we will refer to ΔG throughout this manuscript.

While there have been a few papers published showing the application of AL to RBFE calculations [6–8], there has been limited investigation of the impact of design choices on the ability of AL methods to identify the most promising molecules. To establish a baseline for comparing AL approaches, we performed free energy calculations on every molecule in a library of 10,000 congeneric molecules. The dataset generated from these exhaustive calculations enabled us to examine the impact of several design choices on our ability to identify the 100 molecules with the lowest ΔG. In this study, we evaluated the impact of the following factors.

- The number of molecules sampled at each iteration.
- The strategy used to select the initial sample.
- The method used to build the machine learning model.
- The acquisition function, which balances exploration and exploitation in the search.

To ensure reproducibility and provide the ability to extend this work, we have made the source code and data available at https://github.com/google-research/google-research/tree/master/al_for_fep.

This work builds on prior publications demonstrating the application of AL and RBFE for selecting molecules from large congeneric datasets. Table 1 provides some context for this work by comparing key design choices for this study and previous work. In addition, the next section briefly reviews the relevant literature.

One of the first reports of the application of AL to RBFE energy calculations was a 2019 publication by Konze and coworkers at Schrödinger [6]. In this paper, the authors used several computational techniques to select a small set of molecules from a library of 300K potential CDK2 inhibitors. A set of property filters was used to narrow the initial 300K library to 110K. Docking calculations were then used to reduce the library to 70K molecules. The AL process began by randomly selecting 1K molecules from the 70K pool. The molecules in the 1K sample were subjected to short (1ns) free energy calculations using Schrödinger's FEP+. These RBFE calculations were used to train a machine learning model using Schrödinger's AutoQSAR, which generates models using several different descriptor sets and ML algorithms and selects the most predictive model. Molecules with the 1K top predictions from the ML model were subjected to 1ns FEP+ calculations and chemical structures, and ΔG values for these molecules were added to the training set. This process was repeated for five cycles. During the AL process, 5255 molecules were profiled in 1ns FEP+ calculations. Ligands predicted to bind with an affinity <10 nM were then profiled using a more extensive (20 ns) FEP protocol that included cycle closure. Sampling only 5.8% of the 70K library, the ML model recovered 67% of the 197 single-digit nM binders expected to exist based on statistical analysis.

In a 2022 preprint [7], Gusev and coworkers describe the application of AL and RBFE to the design of inhibitors of the SARS-CoV-2 papain-like protease (PLpro). The authors began with a large library of commercially available molecules, which were reduced to a set of 8175 docked structures. A diverse subset of 45 molecules was initially selected and subjected to RBFE calculations using the thermodynamic integration (TI) method implemented in AMBER 18. An AutoQSAR similar to the approach used by Konze above was used to train an ML model to predict ΔG from chemical structures. Following the initial selection, five cycles of balanced selection were employed to identify molecules to add to the training set. In these balanced selection cycles, the 200 molecules with the most negative predicted ΔG values were clustered to 30. The best scoring molecule from each cluster was added to the training set. Following the initial six AL cycles described above, a cycle of random selection was performed to enhance the diversity of the training set. In the final cycle, a "greedy" selection was performed to select the 30 molecules with the most negative ΔG. In this approach, which balanced exploration and exploitation, 253 RBFE calculations were per-

Table 1
A comparison of key design choices for this study and previous literature. EN = Elastic Net, RF = Random Forest, GBR = Gradient Boosted Regression, MLP = Multi-layer Perceptron, GPR = Gaussian Process Regression, HS = Half Sampling, EI = Expected Improvement, PI = Probability of Improvement, UCB = Upper Confidence Bound

Paper	Number of Molecules	Molecules per Iteration	Number of Iterations	Molecular Descriptors	Machine Learning Models	Acquisition Function(s)	Exhaustive Comparison
Konze [6]	70,000	1,000	5	AutoQSAR	AutoQSAR	Greedy	No
Gusev [7]	8,175	30-45	8	AutoQSAR	AutoQSAR	Mixed	No
Khalak [8]	2,351	100	7	RDKit 2D and 3D	Ensemble MLP	Narrowing, Greedy, Random, Mixed, Uncertainty	No
This work	10,000	20-100	10	RDKit Morgan FP	EN, RF, GBR, MLP, GPR	Greedy, HS, PI, EI, UCB	Yes

formed. From these calculations, 24.5% of the ligands had a predicted 10x improvement in binding affinity, and 6% had a predicted 100x improvement over a starting reference molecule with a 2.3 uM IC50.

In another 2022 preprint, Khalak and coworkers describe the application of AL and RBFE to the selection of a set of PDE2 inhibitors [8]. In this paper, the authors initially selected molecular descriptors and an AL method by examining model performance using experimental binding affinity data from 2,351 molecules synthesized for a PDE2 drug discovery project. This validation led them to use a set of 2D and 3D descriptors from the RDKit for model building. Having selected the most appropriate descriptors, they evaluated several acquisition functions similar to those described above.

- A "narrowing" strategy was used to balance exploration and exploitation. After an initial random selection, multiple ML models were generated, and 20 molecules each were selected from the best-performing models. After three cycles of narrowing, a greedy strategy, where the molecules with the best predicted binding affinity were selected. The authors compared this narrowing/greedy approach with three other selection methods.
- An uncertainty-based approach where molecules were selected based on the standard deviation of the predictions across five models.
- A simple random selection of 100 molecules per iteration.
- A mixed approach, which used a 3:1 mixture of the molecules predicted to be the most potent to the molecules chosen with the uncertainty approach described above.

When evaluating the ability of the models to recover the 50 most potent molecules, the authors found that the greedy and mixed approaches yielded the best performance. These parameters were subsequently used to direct an AL approach designed to identify PDE2 inhibitors prospectively. It should be noted that the validation studies here were performed with biological data rather than ΔG from RBFE. One can argue that ΔG should predict binding affinity and that this is the most appropriate way to tune an AL approach. However, it may also be possible that a model tuned on biological data may not necessarily produce molecules with the best distribution of ΔG from RBFE calculations.

2. Methods

2.1. Dataset generation

We chose to target the TYK2 kinase in our AL study because previous work has demonstrated the successful application of RBFE for predicting the binding affinity of TYK2 inhibitors [9,10]. This success has been due, in part, to the availability of high-resolution inhibitor-bound TYK2 crystal structures (used as input to RBFE simulations) in the Protein Data Bank (PDB) and of high-quality TYK2 inhibitor binding affinity benchmark datasets (used to validate RBFE predictions). To generate the dataset used in this work, we enumerated a congeneric series of 10,000 TYK2 inhibitors based on the same amino-pyrimidine scaffold explored by Rufa and coworkers [10] and exemplified in Patent No. US 8,486,950 B2. 573 TYK2 inhibitors were extracted from Patent No. US 8,486,950 B2 and decomposed into unique R-groups along the pattern shown in Fig. 2. Distinct R-groups were then combinatorially recombined to generate a preliminary inhibitor library in a process described below.

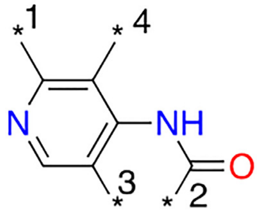


Fig. 2. Schematic depicting the R-group decomposition used to generate our virtual library of 10,000 TYK2 inhibitors combinatorially.

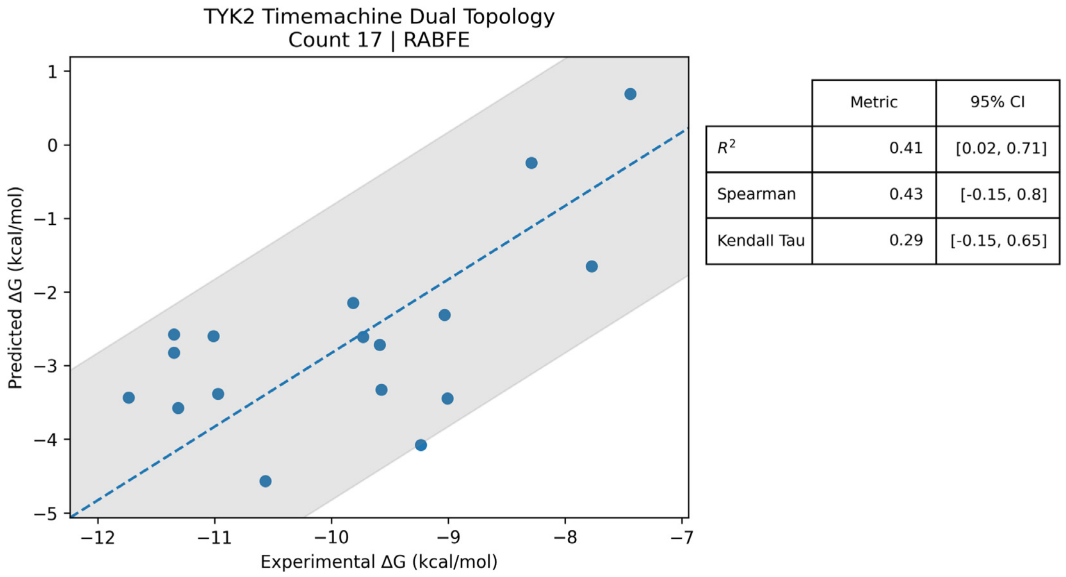
Many of the 399 unique R1 groups and 62 unique R2 groups resulting from decomposition were fairly elaborate. To better represent the types of preliminary transformations typically explored in medicinal chemistry campaigns, we supplemented these with an additional set of "simple" R1 groups ($N = 30$) and R2 groups ($N = 139$). The resulting 429 R1 groups and 201 R2 groups were combinatorially combined with 3 different R3 groups (at R4 only H was permitted) to produce 203,406 unique products, which were filtered for "drug-like" properties (MW <= 500, -2 <= XLogP <= 4, HBA < 10, HBD < 5, TPSA_NO < 120, rotatable bonds <= 8, halogens <= 4, formal charge = 0, HAC <= 32), resulting in 92,644 candidate inhibitors. Ligands were prepared using Maestro's LigPrep with default settings, and compounds with non-zero net charge were filtered out, leaving 77,365 neutral candidates for docking.

A docking receptor was prepared from TYK2 PDB structure 4GIH [11] (chosen because its cocrystal ligand shares the same chemical scaffold as the compounds in our virtual library) using the Schrodinger Protein Preparation and Receptor Grid Generation workflows within Maestro [12] using default settings. Candidates were docked with standard precision Glide into the 4GIH receptor using maximum common scaffold (MCS) core constraints (4GIH ligand as reference) and enhanced ligand sampling, resulting in 66,250 hits that docked successfully. Stereo-expansion of compounds with ambiguous stereochemistry produced some duplicates. After docking hits were deduplicated and zwitterionic compounds (zero net charge, nonzero net absolute charge) were removed, 39,527 filtered hits remained.

At this point, we implemented additional filters to cull the hitlist down to 10,000 compounds that we would explore through AL. Compounds with HAC < 20 or HAC > 30 were filtered out (the 4GIH reference ligand has 23 heavy atoms), as were compounds in the bottom 20% of the Glide docking score distribution. From these remaining 22,649 hits, we selected the 10,000 compounds most similar to the 4GIH ligand as measured by Tanimoto similarity with 1024-bit Morgan fingerprints with a radius of two for free energy calculations.

2.2. Binding free energy calculations

The free energy method chosen for the 10,000 compounds was a Dual Topology [13] relative binding free energy protocol available through



	Metric	95% CI
R^2	0.41	[0.02, 0.71]
Spearman	0.43	[-0.15, 0.8]
Kendall Tau	0.29	[-0.15, 0.65]

Fig. 3. Reference Tyk2 compounds from JACS set [1]. The performance of the RABFE protocol on the Tyk2 set is between FEP+ (0.79) and Amber TI (0.24) [2] in R^2 at 0.41. There is an offset between the predicted ΔG and the experimental ΔG due to the bias introduced in the RABFE method; this bias has no impact on the correlation.

the Timemachine MD engine [14] called Relatively Absolute Binding Free Energy (RABFE). Fig. 3 shows a comparison between the values calculated by this protocol and experimentally determined values for 17 compounds. The protocol achieved an R^2 of 0.41, which is competitive with other existing approaches.

This protocol uses the co-crystal ligand from PDB 4GIH as the reference compound in a star map to provide a validated pose that can restrain the 10,000 compounds within the binding pocket. The RABFE protocol comprises two legs: the solvent leg and the complex leg.

The solvent leg predicts the dehydration free energy of the target ligand, not evaluating the reference compound. The solvent leg is made up of two stages: conversion and decoupling. The conversion stage decharges the target ligand and converts the Lennard-Jones terms to a simplified set of atom-type based parameters. The decoupling stage removes the discharged and standardized ligand from the solvent.

The complex leg predicts the energy of replacing the reference ligand from the complex with the target ligand. It uses the reference compound to restrain the target ligand and ensure engagement with the binding site. The complex leg comprises three stages: a conversion stage and two decoupling stages. The complex conversion stage is the same as the solvent conversion stage in that it decharges and standardizes the target ligand in the binding pocket. There are two decoupling stages which both include the target compound and the reference compound restrained to each other; neither compound interacts with the other. The decoupling stage begins by harmonically restraining the two compounds to each other. The restraints do not require a bijective mapping between atoms in the two compounds, which allows for mapping any atom in one compound to the atoms in the other compound as long as they are within 0.5 Å. Once restrained, one compound begins fully interacting while the other is non-interacting, the two compounds are then swapped by coupling and decoupling the nonbonded parameters. For these two stages, one additional window is run without restraints, whose trajectory is post-processed into samples as if factorizable restraints were present - allowing for separability of the partition function, resulting in end-states whose standard state correction can be computed (though assumed and shown to be constant for the entire dataset).

It is important to note that the conversion stages of both legs as well as the decoupling stage of the solvent leg do not include the co-crystal ligand. The RABFE protocol intended to provide accurate rank ordering of compounds rather than the mean unsigned error (MUE) of binding predictions, and without computing the solvent leg for the co-crystal ligand there is a constant shift in the ΔG prediction which is the dehydration free energy of the reference compound. By introducing this constant shift, it reduces the amount of simulations required to run per compound as well as reducing noise in the final ΔG estimate.

The following is the equation used to get the final ΔG value of an RABFE run:

$$\begin{aligned}\Delta G_{bind} = &\left(\Delta G_{solv-conv} + \Delta G_{solv-decoupling}\right)\\ &- \left(\Delta G_{complex-conv} - \Delta G_{complex-decouple-a->b} + \Delta G_{complex-decouple-b->a}\right)\end{aligned}$$

See Appendix A for details regarding our TimeMachine calculations.

2.3. Experiment setup

We use an AL framework with pool-based sampling. In this setup, we aim to find the ligands with the lowest calculated ΔG values from the set of ligands generated as described above while sampling as few molecules as possible. The procedure consists of the following major steps:

(1) Select a subset of candidates from the pool as the initial training set. Remove these from the candidate pool.
(2) Recover ground truth values for candidates in the current training set. In prospective experiments, this would require running the Free Energy Method. Since we have precomputed these values, we simply look them up.
(3) Use the training set to fit a regression model.
(4) Use the regression model to score candidates from the selection pool and select a new subset of candidates to add to the training set.
(5) Repeat steps 2–4.

Fig. 4 shows a diagram of the full approach. Molecules are fetched from the candidate pool for an initial round of metadata analysis. The metadata analysis includes selected candidate clustering, and the results are deposited into the database for later analysis. See Section 2.4 for a list of the approaches used to select the initial training batches. The pipeline then repeatedly passes through cycles of FEP simulation, regression model training, and next batch selection. The batch selection criteria are based on the acquisition functions described in the methods section. The pipeline in our survey was time-bound to perform

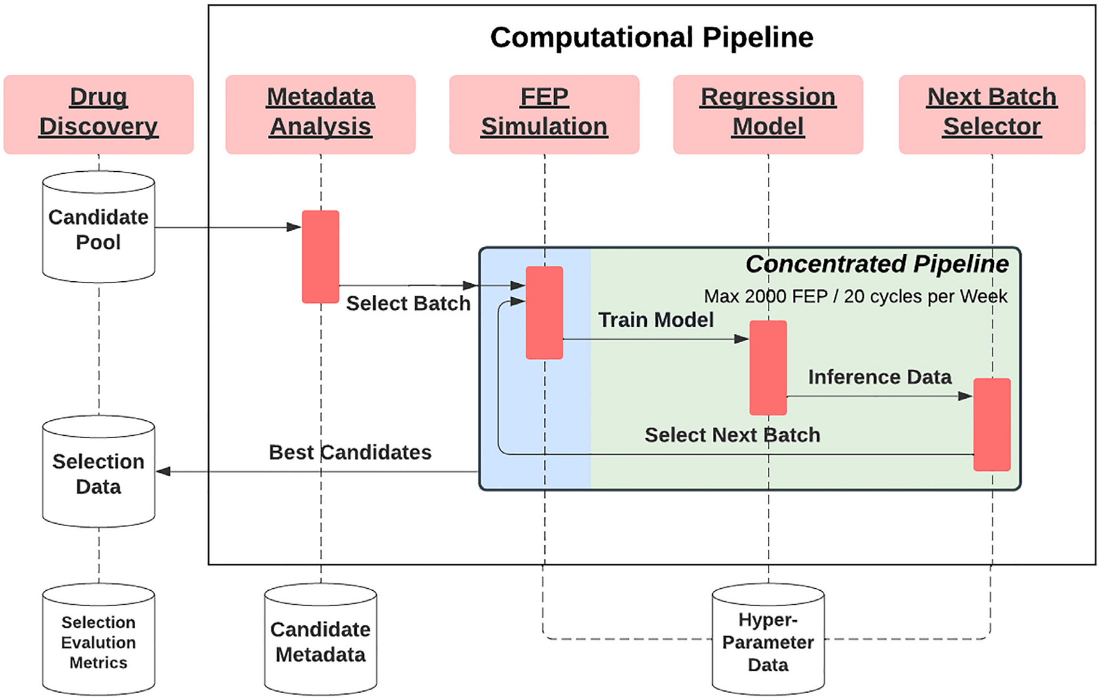


Fig. 4. Active learning pipeline diagram for free energy calculations.

a maximum of 2000 FEP simulations or 20 cycles of batch selection within a week. Once the batch selection iterations are complete, selected molecules are evaluated to assess the selection quality of the pipeline.

2.4. Survey design

We evaluated the different combinations of batch size, acquisition method, machine learning model, and initial training set selection strategy. Table 2 lists the values and approaches we tested for each component.

To provide an initial performance baseline for AL on the TYK2 dataset, we focus on a few elementary model types. We use the implementations available in the scikit-learn python library [15] with default values, unless otherwise stated (see Appendix B). Similarly, although there exist sophisticated batch selection policies, we use sequential selection where we select the top molecules as indicated by the selection policy at each cycle.

In our retrospective setting, there is a synergistic relationship between the number of cycles run and batch size. For example, running 10 cycles with a batch size of 5 requires the same amount of compute as running 5 cycles with a batch size of 10. These two configurations could be considered equivalent if we were only constrained by the number of selections we could make. However, in a prospective environment, these dimensions are independently limited by access to compute and time, respectively. Batch size is directly limited by access to GPUs or other hardware. An active learning campaign also has to complete quickly in order to be useful in a lead optimization program. A reasonable timeline might be 2-3 days, which would only allow for about 5 cycles. We aren't quite as constrained in the retrospective setting and choose to relax these constraints to maximums of 100 for the batch size and 20 for the number of cycles.

The main goal of our AL task is to quickly identify molecules that have minimal ΔG values. Intuitively, this requires an accurate rank ordering of the molecules. When training models, we formulate the estimation task as a regression problem. These two tasks are related, but it is possible to train a model that results in higher recall of target molecules, but has a lower accuracy in ΔG predictions, or vice versa. In our experiments, we track three metrics to monitor performance on both the rank ordering task and the regression task: recall, clustered recall and root mean squared error (see Appendix B.4).

We use RDKit Morgan fingerprints with a radius of 3 and size of 2048 as input features for all experiments.

The KMeans and Random initial training sets use randomization when choosing molecules. We generate five independent instances for each of these approaches. We also run each configuration in triplicate to account for any randomness in the machine learning method or the acquisition function.

Table 2
Values and approaches tested in our survey of active learning on the TYK2 dataset. See Appendix B for short descriptions of each approach and the specific settings used.

Parameter	Choices Explored
Batch Size	20, 40, 60, 80, 100
Machine Learning Method	Elastic Net (ENET),Random Forest (RF), Gradient Boosting Machine (GBM), Multi-layer Perceptron (MLP), Gaussian Process Regressor (GPR)
Acquisition Function	Greedy, Half Sampling (HS), Probability of Improvement (PI)*, Expected Improvement (EI)*, Upper Confidence Bound (UCB)*
Initial Training Set	Random, Worst by ΔG, Farthest by ΔG, KMeans
Metrics	Recall, Root Mean Squared Error (RMSE), Cluster Recall

* These acquisition methods use an uncertainty estimation for batch selection. We only apply these to active learning campaigns using the GPR model. We avoid constructing uncertainty estimates (e.g. with ensembling) with other machine learning methods.

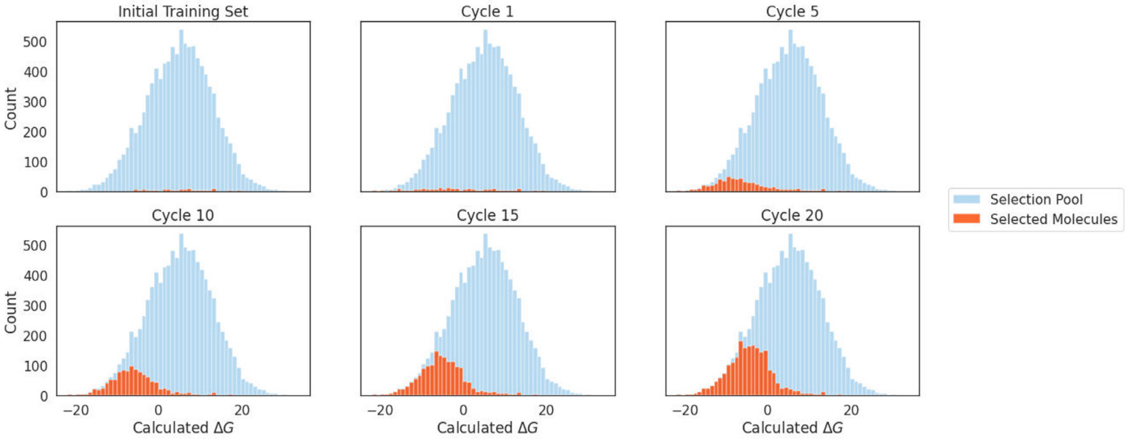


Fig. 5. The distribution of ΔG values of selected molecules across a single active learning campaign using greedy selection and MLP with a batch size of 100.

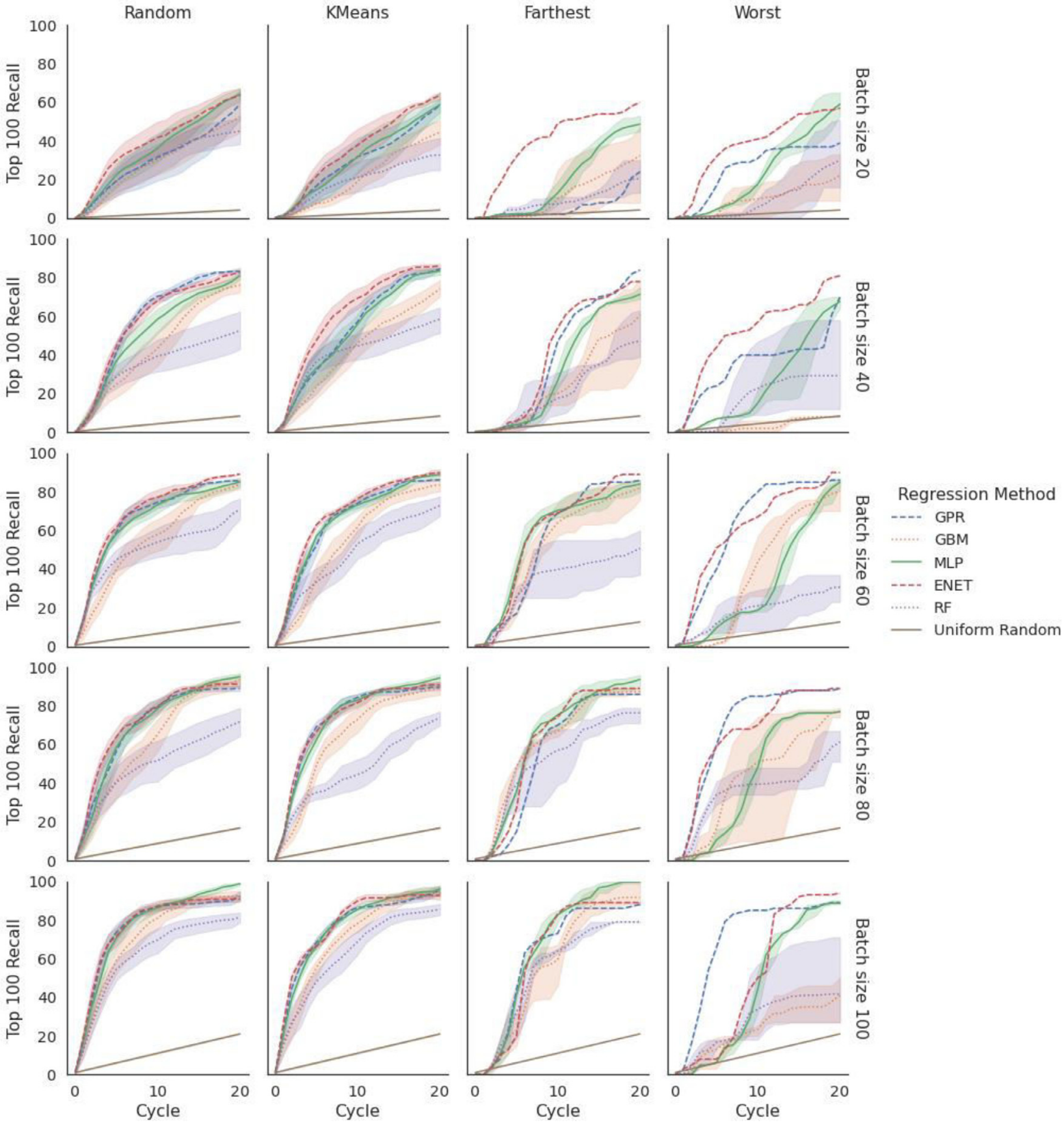


Fig. 6. Recall comparison of regression methods and batch sizes using a greedy acquisition function. The choice of batch size has a larger effect on the number of molecules recalled than the choice of regression algorithm. Shaded areas represent 95% confidence intervals.

Table 3
Recall values across different initial training sets using a batch size of 60 and 10 active learning cycles. Values in parentheses are the 95% confidence interval. The choice of initial training set had little effect on the top molecule recovery.

Name	Random	KMeans	Farthest	Worst
Gaussian Process	74.6 (71.19, 78.01)	74.00 (71.17, 76.83)	69	80
Gradient Boosted Machines	57.87 (48.49, 67.24)	61.80 (55.61, 67.99)	64.00 (41.92, 86.08)	43.67 (4.05, 83.28)
ElasticNet	77.20 (72.65, 81.75)	76.54 (74.18, 78.90)	69	67
Multilayer Perceptron	73.27 (68.47, 78.06)	73.20 (69.89, 76.51)	70.33 (59.99, 80.68)	18.67 (15.80, 21.54)
Random Forest	53.93 (45.25, 62.62)	53.13 (44.68, 61.59)	39.67 (2.38, 76.96)	21.00 (-7.21, 49.21)

3. Results and discussion

With our AL approach, we were able to recover the vast majority of the best 100 molecules by ΔG while sampling a small subset of the data. Fig. 6 shows the cumulative recall of the top 100 molecules at every cycle using the greedy acquisition strategy. Our largest experiment used a batch size of 100 for 20 AL cycles, which ends up selecting 20% of the full dataset. Starting from random batches of molecules, we recovered an average of 98.7 of the best 100 molecules by ΔG with the MLP model. Fig. 5 shows the progression of the ΔG values recovered during a single campaign. Using GPR, ENET, or GBM, the recovery drops to 91-93. RF models recovered the least, with an average of 81.3.

As a comparative baseline, Fig. 6 includes the expected recall when samples are uniformly sampled from the candidate pool at each cycle with no consideration for any machine learning model. Since this approach requires no training or computation, any proposed AL strategy should outperform these expected results. Except for some RF and GBM campaigns starting from the Worst initial set, all of the active learning approaches we experiment with greatly outperform this baseline.

We were able to reduce the total number of molecules sampled in a campaign and still maintain a relatively high recall. With 10 cycles and a batch size of 60, the AL campaigns end up selecting only 6% of the dataset. The ENET, GPR and MLP algorithms were still able to recover between 73 and 77 of the top 100 molecules on average. We do see performance start to degrade more quickly if we continue to lower the batch size, but even with a batch size of 20, which would result in a campaign selecting 1% of the dataset, the best models were able to recover around 40 of the top 100 molecules.

Unless stated otherwise, we focus on the experiment setup with a batch size of 60 and 10 AL cycles as a configuration where we see good but not saturated recall values while only sampling a relatively small portion of the selection pool.

The choice of the initial training set had little impact on the recall of the best-performing models (Table 3). Using initial training sets constructed by selecting molecules from different *k*-means clusters aimed to provide a more diverse sample of the molecule space for the models to learn from. We see almost identical recall performance as starting with random molecules, especially among the top three models. GPR and ENET models could even maintain reasonable recall values when starting from the worst molecules by ΔG. RF and GBM were the only models that were unable to consistently escape from the poor starting point. MLP also had difficulty through 10 cycles, but was usually able to catch up to GPR and ENET in later cycles.

Greedy acquisition is an extreme exploitation strategy. We also ran experiments using the half sampling acquisition function, a highly explorative strategy [16]. Using half sampling acquisition at every cycle significantly decreases the average recall to 10-20 for all methods when starting from random molecules. We see the expected general improvement in the accuracy of models trained throughout an AL campaign, though. The RF and GBM models saw an improvement of 1-2 in RMSE, while there were more modest improvements (< 0.5) for GP, MLP, and ElasticNet models.

To see if a combination of exploration and exploitation strategies provide an increase in recall over purely one or the other, we experiment with two approaches for alternating between the greedy and half sampling acquisition methods. In one, we perform a certain number of

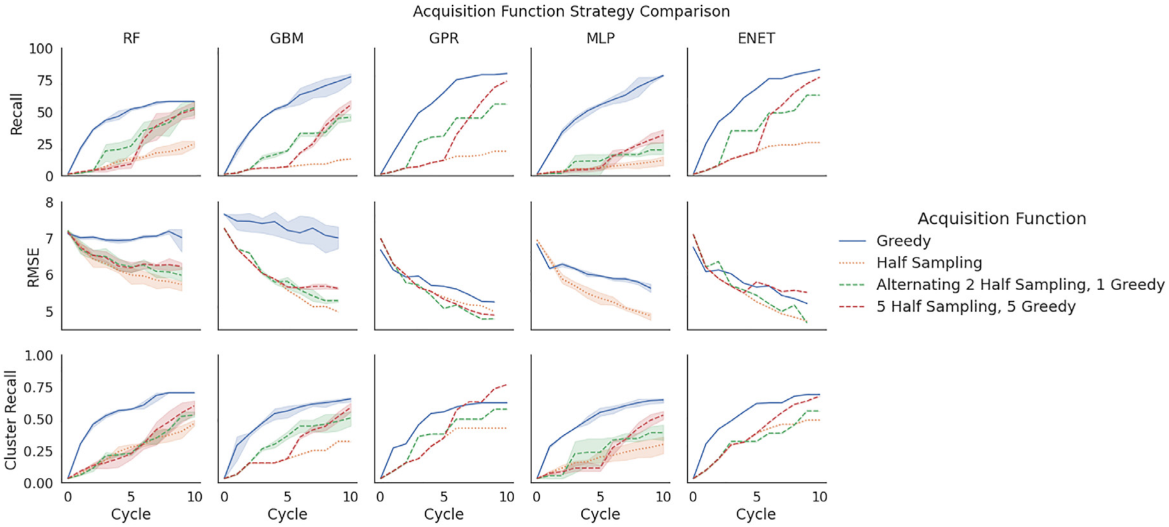


Fig. 7. Recall, RMSE and Cluster Recall for different machine learning methods using alternating acquisition functions. Experiments were run with a batch size of 60. The alternating acquisition strategy provides an improvement in RMSE, but is unable to match the recall rates of a purely greedy strategy. There is also no consistent benefit in increasing the diversity of the target molecules recovered. Shaded areas represent 95% confidence intervals. RMSE for the alternating strategies with the MLP method are not shown due to spikes caused by overfitting.

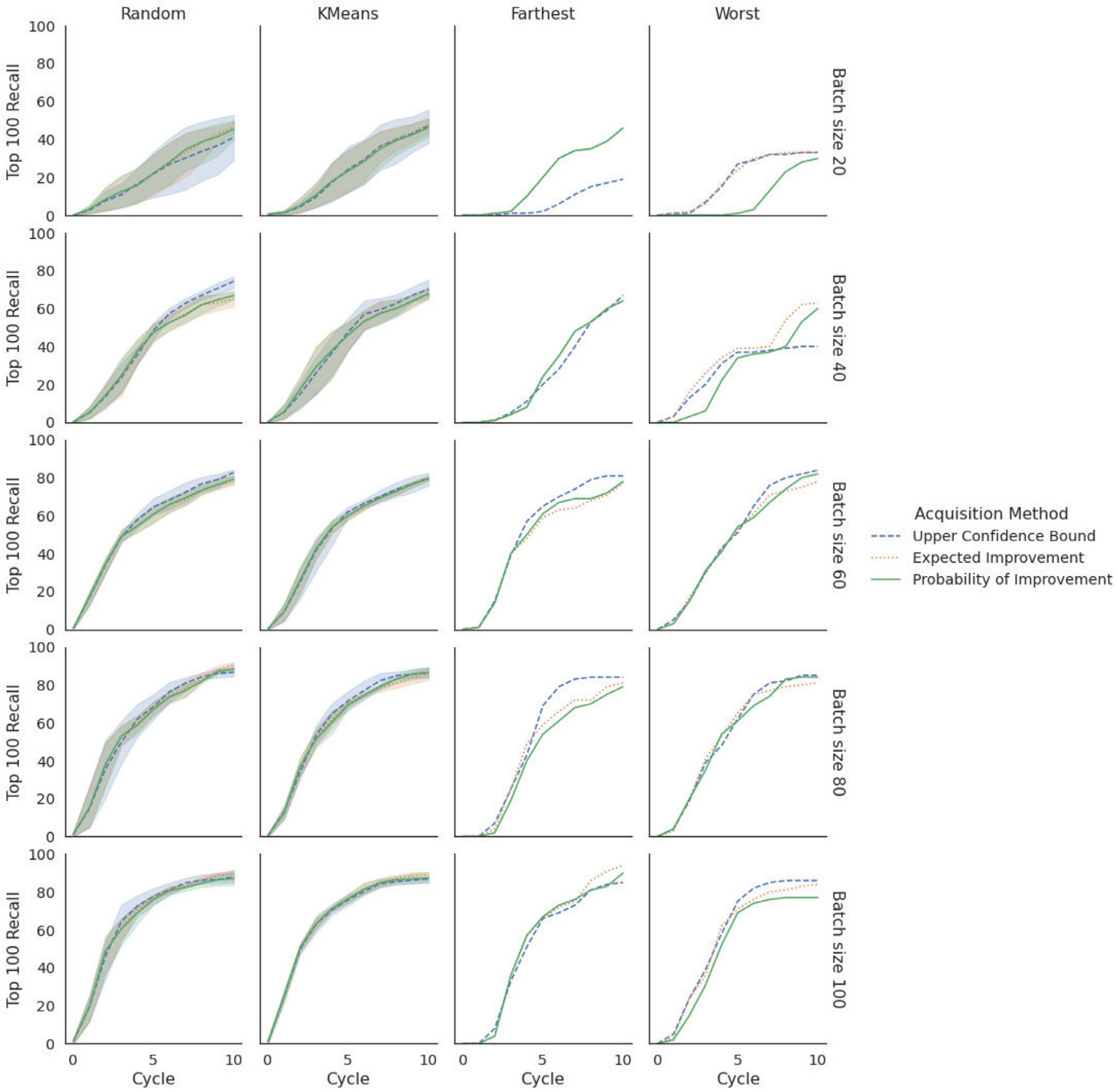


Fig. 8. Comparison of recall across acquisition methods when using a Gaussian process regressor. UCB values are shown with a beta value of 10. A tradeoff value of 0.0 was used for both probability of improvement and expected improvement algorithms. Shaded areas are 95% confidence intervals over five independent runs. For the Random and KMeans initial training sets, this also includes an aggregation over five independently generated initial training sets.

half-sampling acquisitions before switching to greedy acquisition. In the other, we alternate between a small number of cycles with half sampling acquisitions and a small number of cycles with a greedy acquisition function. Results are shown in Fig. 7.

Alternating these two acquisition functions provided improvements in RMSE over using greedy alone, but it did not lead to a consistent increase in the overall recall achieved or the diversity of the molecules recovered. When utilizing half sampling cycles before permanently switching to greedy cycles, the recall measures can almost catch up to those seen in greedy-only campaigns by the tenth cycle. This approach also did not extend the recall to new clusters of target molecules, however.

3.1. Uncertainty based selections

Using acquisition functions that incorporate an uncertainty measure, we recover a similar number of target molecules than with acquisition functions that do not consider uncertainty. Fig. 8 shows results for uncertainty based selections with GPR models. All acquisition methods perform comparatively with each other as well, recovering between 75 and 85 of the top 100 molecules after 10 cycles with a batch size of 60, starting from a random initial set.

There was a small dip in recall when starting from the Farthest or Worst initial training sets, but all of the acquisition methods also proved fairly robust with respect to the choice of the initial training set, overall. The EI and PI acquisition methods also achieved the same performance across multiple values of the tradeoff parameter.

4. Conclusions

In this study, we examined the impact of several key design choices on the performance of AL methods as applied to RBFE calculations. To perform an objective comparison, we generated a chemical library of 10,000 molecules and carried out RBFE calculations on all of the molecules in the library. With this set of RBFE results in hand, we could systematically compare the impact of several design choices that direct the search. As a realistic performance metric, we examined the ability of different AL variants to recover the 100 molecules with the most negative ΔG. Our experiments showed that AL was surprisingly insensitive to design choices such as the choice of ML method or the acquisition function. The most important factor in our study was the number of molecules sampled at each iteration. While we achieved reasonable performance with samples as small as 20 molecules, the results were highly variable. However, when the sample size was 60 or larger, we were able to consistently recover half of the top 100 molecules within the first five iterations.

We believe the methods employed in this study are consistent with the way in which AL and RBFE calculations would be used to drive design in a drug discovery program. When applying any computational method in a drug discovery project, we must be conscious of two factors, time and cost. Drug discovery projects typically run on design, make, and test cycles that typically encompass two to three weeks. First, ideas for compounds are generated either by scientists, algorithms, or a combination of the two. These ideas are then evaluated based on some combination of intuition and computation. Finally, compounds are synthesized and tested, and new hypotheses are formulated based on the resulting data. For a computational method to be impactful, reliable results should be generated in a few days. If it takes more than a couple of days to generate a computational result, the drug discovery team may have moved on to other ideas.

In practice, running five cycles of AL with 60–100 molecules per iteration should be consistent with the goals of minimizing time and cost. At eight hours and 60 molecules per iteration, five iterations could be performed in less than two days. Five cycles of AL with sixty molecules per cycle would require 300 RBFE calculations. Assuming eight hours per calculation, we would consume 2400 h of GPU time. At $2 per GPU hour, the total cost would be $4,800, a cost equivalent to synthesizing a few molecules, given typical pharmaceutical costs.

Some previously published AL studies have employed rather exotic combinations of acquisition functions that attempted to increase exploration by using techniques such as clustering to select diverse subsets from molecules with the highest predicted scores. These sampling schedules, which focus on exploration in early iterations and shift to exploitation toward the end of the search, seem to be based on intuition rather than systematic study. For this dataset, our work shows that simple random initialization coupled with a greedy acquisition function performs as well as other methods that attempt to balance exploration and exploitation. The number of molecules chosen as each iteration was the only prominent factor affecting performance.

Of course, it must be noted that the results presented here may be specific to our dataset and survey design. For instance, it is possible that the dataset we generated is easily optimized, reducing the benefit of explorative selections. It is also possible that a design choice we did not explore (e.g. molecular representation) has a larger impact on performance than those we did. It is encouraging to note that greedy selection was also a top performer in the recent preprint from Khalak, where the authors performed a limited comparison of AL approaches. We hope that this work will prompt others to perform similar comparisons and release their datasets.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All code and data is available as open source

Acknowledgments

The authors would like to thank our colleagues at Google and Relay Therapeutics for their comments and suggestions on this manuscript. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix

A. TimeMachine Details

All of the simulations were run under the following conditions. The ligand parameterization was done with the OpenFF 1.1.0 force field [22] with charges from AM1-BCC [23]. The protein force field used to parameterize PDB 4GIH was Amber ff99SB-ILDN [24], having been prepared using the default configuration of Maestro's Protein Prep workflow with the addition of capping the termini. The protein was then solvated with a 10 Å buffer of tip3p [25] water, resulting in a complete box size of (59Å, 73Å, 61 Å).

The solvent simulations were all 40A water boxes. The initial complex and solvent boxes were minimized and equilibrated for 200000 steps using the reference compound. The timestep of the atomistic simulations is 2.5fs with HMR enabled. We used a BAOAB [26] langevin integrator thermostat at 300 K with a Monte Carlo Barostat running every 25 steps. Each window was run for 2 nanoseconds. Each stage was run with 64 windows, for a total of 320 windows per compound. Energies and errors were computed using pair BAR [27]. Experiments were run with the Timemachine Github Sha d7ad70929271960279dfb5a08c2beac77423745a [28].

Three simulations failed to complete, leaving the final dataset with a size of 9,997.

B. Survey Details

B.1 Machine learning methods

Elastic Net (ENET). In linear regression models, the target scalar value is modeled as a linear combination of parameters. With a naive approach, linear models can be prone to overfitting the training data, especially if the size of the training dataset is small. Different regularization techniques help mitigate overfitting. Elastic Net is a linear regression model that utilizes L_1 and L_2 regularization which influence the regression coefficients to be small (L_2 regularization) or even zero (L_1 regularization).

We use the cross validation estimator available in sklearn with l1_ratio values of 0.1, 0.5, 0.75, 0.95, and 0.99.

Random forest (RF). Random Forest regression [17] is an ensemble technique utilizing a collection of decision trees. Each individual tree is fit to both a random set of training examples and a random subset of training features. During prediction, expected value and uncertainty estimates can be computed from the collection of individual trees in the random forest.

We set the max_depth to 25, min_samples_split to 0.02 and min_samples_leaf to 0.01.

Gradient boosting machine (GBM). The Gradient Boosting Machine [18] is an improved method compared to the Random Forest. The GBM method aims to minimize prediction error via a gradient-based method. As opposed to fitting training data in each decision tree independently in the RF model, the GBM method progressively builds new Gradient Boosting Regression Trees (GBRT) upon the prediction error of the previous ensemble of GBRTs. By repeatedly adding new GBRTs into the collection of existing predictors, the error or the loss of the prediction can be minimized.

We use a max_depth of 10, min_samples_split of 0.006 and min_samples_leaf of 0.002.

Multi-layer perceptron (MLP). The multi-layer perceptron [19] is a fully connected, feed-forward neural network optimized by stochastic gradient descent. For this work, we use a network consisting of a single hidden layer of 100 neurons.

Gaussian process regression (GPR). Gaussian Process Regression [20,21] is a Bayesian technique that computes probability distributions over a specified set of functions (the prior) that fit the training data. This prior information is encoded by a similarity matrix which is utilized by the technique during training. One of the major benefits of GPR is that expected values and uncertainty estimates for the prediction on new instances are directly calculated by the method. For the work presented in this paper, the similarity matrix for GPR is computed using the Tanimoto similarity between training instances.

B.2 Acquisition methods

Greedy. Greedy is a naive method in which top molecules ranked by the model score are always selected for the next batch to perform FEP simulations.

Half sampling (HS). Half sampling uses a data subsampling technique similar to jackknifing to derive an uncertainty value for each molecule [16]. The molecule with the highest uncertainty is selected first. The half sampling approach can then update the uncertainty measures of other molecules before knowing the ground truth of the initially selected molecule. Batches of molecules can be selected by iteratively selecting the molecule with the highest uncertainty and updating the uncertainty values.

Probability of improvement (PI). The probability of improvement acquisition policy aims to select the molecule that has the highest probability of improving upon the currently identified best score. The PI score for a molecule is computed utilizing the standard deviation associated with that molecule to compute the amount of probability mass that molecule has above the current best solution. It is important to note that PI does not consider the magnitude of the improvement.

Expected improvement (EI). The expected improvement policy is analogous to the PI policy but considers the magnitude of the improvement. The value is computed for a molecule by calculating the expected value of the probability density that molecule has above the current best solution. The EI method can be augmented with a parameter that can encourage more exploration.

For both PI and EI methods, we explore tradeoff values of 0, 0.25, 0.5, 0.75 and 1.

Upper confidence bound (UCB). The upper confidence bound acquisition policy is an ‘optimistic’ method that selects molecules based on their potential to yield optimal values. The score for a particular molecule is computed by simply summing the predicted mean value of that molecule with its predicted standard deviation. This policy enables molecules with a moderate mean value but large standard deviations to potentially be selected as candidate points. The UCB method also contains an adjustable parameter that can optionally be used to weight the standard deviation. The larger this weight the more the policy favors exploration of the input space.

We experiment with beta values of 1, 10 and 100.

B.3 Initial training sets

Random. As a baseline training set construction strategy, we use a naive random selection. Ligands are sampled uniformly from the full dataset (without replacement). This approach may generate extremely favorable or unfavorable starting sets due to chance. To get a better idea of a typical random starting point, experiments were performed across five independently generated variations for each batch size.

Worst By ΔG. The ligands with the highest ΔG values are chosen. In a prospective experiment, it would be impossible to know if this was the starting set but it would also be highly unlikely. We include this starting set as a worst case scenario to see if active learning approaches would be able to climb the full ΔG optimization hill without getting stuck.

Farthest from best. For each ligand in the dataset, we compute the minimum distance between the ligand and any of the 100 ligands with the lowest ΔG value. The ligands with the maximum distances are chosen. The distance between two ligands is taken as one minus the Tanimoto similarity between the two ligands.

The Worst By ΔG and Farthest From Best initial sets are two approaches that attempt to challenge AL strategies by forcing them to begin the optimization process as far away from the optimal molecules as possible, one using ΔG values to define distance and the other using (1 - Tanimoto similarity).

KMeans. The initial training sets generated by our Random strategy may over or under sample certain areas of the ligand space defined by the candidate pool. The KMeans starting set aims to sample ligands spread across this ligand space to see if an initial training set of diverse structures provides any benefit over uniform random selection. First, we cluster the ligands using K-Means clustering over the ligands’ fingerprints. A single ligand is then randomly chosen from each cluster. We can control the total number of ligands chosen by setting the parameter K. Since this is another random initialization strategy, we again independently generate five variations for each batch size.

For each AL campaign, the size of the initial training data set is the same as the acquisition batch size.

B.4 Evaluation metrics

Recall. The 100 molecules with the lowest calculated ΔG values are considered the best candidates. The lowest 100 calculated ΔG values range from -13.35 to -21.8. Recall is calculated as the number of best candidates in selected molecules from any cycle of an AL campaign.

We only consider ΔG in our experiments. The molecules with the lowest ΔG scores therefore represent the best candidates for moving forward in the drug development process in our experiments. We consider this the main objective of the AL approaches in our experimental set up.

Root mean squared error (RMSE). The square root of the difference between model predicted ΔG and calculated ΔG values across the full data set. We use RMSE as a measure of the machine learning method’s overall accuracy.

Cluster recall. We first cluster the full dataset using K-means clustering using a K of 100. We consider a cluster a target cluster if it contains at least one of the top 100 molecules by ΔG. A recalled cluster is a cluster that contains at least one molecule that was selected by an AL campaign at any cycle. The cluster recall is the percentage of target clusters that are also recovered clusters. We use this metric to probe if active learning strategies are identifying a single motif that is common in the best scoring ligands or if they can identify diverse structures with nearly optimal scores.

For our metrics, we compute and average the cluster recall across five independent K-means clustering instances.

References

[1] Wang L, et al. Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field. J Am Chem Soc 2015;137(7):2695–703. Feb.[Online]. Available http://pubs.acs.org/doi/abs/10.1021/ja512751q .

[2] Song LF, Lee T-S, Zhu C, York DM, Merz KM Jr. Using AMBER18 for relative free energy calculations. J Chem Inf Model 2019;59(7):3128–35 Jul.. doi:10.1021/acs.jcim.9b00105.

[3] Reker D, Schneider G. Active-learning strategies in computer-assisted drug discovery. Drug Discov Today 2015;20(4):458–65 Apr.. doi:10.1016/j.drudis.2014.12.004.

[4] Reker D, Schneider P, Schneider G, Brown JB. Active learning for computational chemogenomics. Future Med Chem 2017;9(4):381–402 Mar.. doi:10.4155/fmc-2016-0197.

[5] Warmuth MK, Liao J, Rätsch G, Mathieson M, Putta S, Lemmen C. Active learning with support vector machines in the drug discovery process. J Chem Inf Comput Sci 2003;43(2):667–73 Mar.. doi:10.1021/ci025620t.

[6] Konze KD, et al. Reaction-based enumeration, active learning, and free energy calculations to rapidly explore synthetically tractable chemical space and optimize potency of cyclin-dependent kinase 2 inhibitors. J Chem Inf Model 2019;59(9):3782–93 Sep.. doi:10.1021/acs.jcim.9b00367.

[7] F. Gusev, E. Gutkin, M.G. Kurnikova, and O. Isayev, “Active learning guided drug design lead optimization based on relative binding free energy modeling,” ChemRxiv, Jul. 2022, doi: 10.26434/chemrxiv-2022-krs1t.

[8] Y. Khalak, G. Tresadern, D.F. Hahn, B.L. de Groot, and V. Gapsys, “Chemical space exploration with active learning and alchemical free energies,” 2022. https://s3.eu-west-1.amazonaws.com/assets.prod.orp.cambridge.org/0b/7ad8a59dac44ca8ce3f80befa86951_no_meta.pdf?AWSAccessKeyId=ASIA5XANBN3JD4H36OMV&Expires=1658241896&Signature=Q05vbZ5pCSoi7EJBfjhqWhD%2B4zI%3D&response-cache-control=no-store&response-content-disposition=inline%3B%20filename%20%3D%22chemical-space-exploration-with-active-learning-and-alchemical-free-energies.pdf%22&response-content-type=application%2Fpdf&x-amz-security-token=FwoGZXIvYXdzEOD%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaDL0qO4sv29dhs%2Fb%2BWiKtAXEIGIe5D6rw1p21eJKm%2FlZvpXaIBZRMX%2FpshDEwWJNLq91D%2Bs0ZaoPh8r5MRL%2FQ7YzIylBWrKraILixEKJvQuAvql7VXUolxnwwv52pZKPo%2Ffqem2WDq2X8ce0HxWEgBLao9jiBqdJgAHPbZ0IsEsq%2BMnSm0FTRjBcuE2%2FNzrXk%2BAlCGSCTZ84fIT8sP%2FmGP7PO3tmU9p62umHb79a8BgQBNT5F6WOSplkWDqWnKMCE25YGMi1JTgrokcgrZoB5csHeXo245xrSiZrZ%2F2bHHfodQNNnqjEFoethEVSu5qaIt7w%3D (accessed Jul. 19, 2022).

[9] Hahn D, et al. Best practices for constructing, preparing, and evaluating protein-ligand binding affinity benchmarks [Article v1.0]. Living J Comp Mol Sci 2022;4(1):1497 1497Aug.. doi:10.33011/livecoms.4.1.1497.

[10] D.A. Rufa et al., “Towards chemical accuracy for alchemical free energy calculations with hybrid physics-based machine learning /molecular mechanics potentials,” bioRxiv, p. 2020.07.29.227959, Jul. 30, 2020. doi: 10.1101/2020.07.29.227959.

[11] Liang J, et al. Lead identification of novel and selective TYK2 inhibitors. Eur J Med Chem 2013;67:175–87 Sep.. doi:10.1016/j.ejmech.2013.03.070.

[12] Schrödinger, Inc., Schrödinger software suite. [Online]. Available: https://www.schrodiger.com

[13] Rocklin GJ, Mobley DL, Dill KA. Separated topologies–a method for relative binding free energy calculations using orientational restraints. J Chem Phys 2013;138(8):085104 Feb.. doi:10.1063/1.4792251.

[14] J. Fass, J. Kaus, M. Wittmann, F. York, Y. Zhao, TimeMachine. [Online]. Available: https://github.com/proteneer/timemachine

[15] Pedregosa, et al. Scikit-learn: machine learning in Python. J Mach Learn Res 2011;12:2825–30.

[16] Heavlin WD. On ensembles, i-optimality, and active learning. J Stat Theory Pract 2021;15(3):66 May. doi:10.1007/s42519-021-00200-4.

[17] Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP. Random forest: a classification and regression tool for compound classification and QSAR modeling. J Chem Inf Comput Sci 2003;43(6):1947–58 Nov.. doi:10.1021/ci034160g.

[18] Sheridan RP, Wang WM, Liaw A, Ma J, Gifford EM. Extreme gradient boosting as a method for quantitative structure–activity relationships. J Chem Inf Model 2016;56(12):2353–60. Dec.[Online]. Available http://pubs.acs.org/doi/abs/10.1021/acs.jcim.6b00591 .

[19] González-Arjona D, López-Pérez G, Gustavo González A. Non-linear QSAR modeling by using multilayer perceptron feed-forward neural networks trained by back-propagation. Talanta 2002;56(1):79–90. Jan.[Online]. Available https://www.ncbi.nlm.nih.gov/pubmed/18968482 .

[20] P. Renz and S. Hochreiter, “Uncertainty estimation methods to support decision-making in early phases of drug discovery.”

[21] DiFranzo A, Sheridan RP, Liaw A, Tudor M. Nearest neighbor gaussian process for quantitative structure-activity relationships. J Chem Inf Model 2020 Oct.. doi:10.1021/acs.jcim.0c00678.

[22] Qiu Y, et al. Development and benchmarking of open force field v1.0.0-the parsley small-molecule force field. J Chem Theory Comput 2021;17(10):6262–80 Oct.. doi:10.1021/acs.jctc.1c00571.

[23] Jakalian A, Jack DB, Bayly CI. Fast, efficient generation of high-quality atomic charges. AM1-BCC model: II. Parameterization and validation. J Comput Chem 2002;23(16):1623–41 Dec.. doi:10.1002/jcc.10128.

[24] Lindorff-Larsen K, et al. Improved side-chain torsion potentials for the Amber ff99SB protein force field. Proteins 2010;78(8):1950–8 Jun.. doi:10.1002/prot.22711.

[25] Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, Klein ML. Comparison of simple potential functions for simulating liquid water. J Chem Phys 1983;79(2):926–35 Jul.. doi:10.1063/1.445869.

[26] Leimkuhler B, Matthews C. Rational construction of stochastic numerical methods for molecular sampling. Appl Math Res Express 2012;2013(1):34–56 Jun.. doi:10.1093/amrx/abs010.

[27] Bennett CH. Efficient estimation of free energy differences from Monte Carlo data. J Comput Phys 1976;22(2):245–68 Oct.. doi:10.1016/0021-9991(76)90078-4.

[28] TimeMachine Commit. [Online]. Available: https://github.com/proteneer/timemachine/commit/d7ad70929271960279dfb5a08c2beac77423745a