BayesCurveFit: Enhancing Curve Fitting in Drug Discovery with Bayesian Inference

Niu Du[1], Yang Chen[2], and Yu Qian[1,*]

[1]J. Craig Venter Institute, La Jolla, CA, USA
[2]Department of Statistics, University of Michigan, Ann Arbor, MI, USA

Abstract

Curve fitting is a key statistical method in the pharmaceutical industry for modeling the relationship between drug effects and dose levels. Traditional regression-based curve fitting is computationally efficient but sensitive to technical errors, particularly with small sample sizes. In practice, budget constraints often limit the number of dose levels that can be tested. We hypothesize that Bayesian inference offers a more robust alternative in resource-constrained settings. We developed BayesCurveFit, a pipeline that consists of three Bayesian techniques for dose-response curve fitting: Simulated Annealing (SA) for initial parameter estimation, Markov Chain Monte Carlo (MCMC) for posterior sampling, and Bayesian Model Averaging (BMA) to integrate predictions from multiple models. Our simulations demonstrated that BayesCurveFit outperforms regression-based methods on under-sampled data, with performance converging as sample size increases. Benchmarking experiments showed BayesCurveFit achieves higher accuracy than Ordinary Least Squares (OLS) and soft L1 regression when working with 10 or fewer observations. A real-world case study using the Kinobeads dataset further validated BayesCurveFit, revealing multiple novel drug-gene interactions overlooked by the popular OLS-based CurveCurator method. These interactions were confirmed in ChEMBL, a manually curated drug discovery database. These findings underscore BayesCurveFit's advantages in data-limited scenarios. BayesCurveFit is available as an open-source Python package, with the reprocessed Kinobeads dataset accessible via a web-based interface for public testing and validation.

1 Introduction

Computer-aided drug discovery (CADD) is producing an increasingly large number of candidate molecules requiring biological validation through treatment-effect and dose-response experiments, followed by regression-based modeling (*a.k.a.*, curve fitting) to identify the relationship between dose levels and observed responses [Sadybekov and Katritch, 2023]. Despite the high throughput of a CADD pipeline, less than 2.5% of the identified candidates can advance to later stages of development after screening [Sun et al., 2022]. Most of the candidates are eliminated through expensive and slow experimental validation. To prescreen all CADD-identified candidate molecules, it is not cost-effective to experimentally test all candidates at all different dose levels. Due to the

* Correspondence should be sent to: Yu "Max" Qian, Ph.D., mqian@jcvi.org.

limited budget, pharmaceutical companies that use CADD often measure only a small number of dose levels in the prescreening stage. However, the regression-based modeling process generally relies on a large number of tests to achieve reliable estimates [Roozbeh et al., 2021, Hodson, 2022, Motulsky and Brown, 2006].

The labor-intensive sample preparation process in dose-response experiments introduces both biological and technical variability, particularly during the culturing stage, which can differ across tests for different molecules. While technical variations are well known, few effective methods exist to mitigate their impact on downstream analysis without replicates. Quality control (QC) primarily identifies and removes extreme outliers, but minor to moderate technical variations often persist in the dataset. These variations, though informative, can introduce skewness and violate key assumptions of regression-based curve fitting, such as homoscedasticity and normally distributed errors. With large sample sizes, occasional technical variability has minimal impact on curve fitting. However, in small-sample settings, assessing model performance under these conditions becomes crucial. Previous studies largely overlooked this issue, partly because computational approaches in drug discovery (CADD) were not widely adopted at the time. Today, improving model sensitivity is a priority for pharmaceutical companies to prevent premature rejection of promising compounds, which could disrupt the entire CADD pipeline. Addressing small yet impactful technical variability to enhance model sensitivity in dose-response analysis is an emerging and important research challenge.

Bayesian statistics provide a robust framework for addressing these challenges by incorporating prior knowledge and providing a probabilistic interpretation of model parameters. Rather than relying on minimizing the mean squared error in regression analysis, the Bayesian approach maximizes the posterior probability of the parameters based on the given data (likelihood), a process known as maximum a posteriori (MAP). This approach excels in situations with limited data or outliers, as it supports more flexible and reliable model selection and parameter optimization[van de Schoot et al., 2021, Kruschke and Liddell, 2018]. However, in real-world applications, general-purpose Bayesian tools like STAN [Carpenter et al., 2017] and PyMC3 [Salvatier et al., 2016], while offering robust building blocks for Bayesian inference, do not provide a specific and complete pipeline with refinement steps for tasks such as curve fitting in dose-response data. Other domain-specific Bayesian statistics tools [Kurchin et al., 2019, Slugocki et al., 2019, Thijssen et al., 2016, Vyshemirsky and Girolami, 2008], though powerful, demand substantial expertise in statistics and programming to ensure proper handling of the workflow, making them less accessible to many researchers in biomedical and pharmacological sciences.

To address these challenges, we developed BayesCurveFit, a Bayesian inference workflow equipped with modular components for configuring and performing Bayesian estimation on undersampled dose-response data and other similar functional data types. Existing Bayesian tools based on a single method often fail to tackle all aspects of these problems. The core innovation of BayesCurveFit lies in its modular design, which enables a seamless, synchronized process for robust curve fitting that is both outlier-resistant and small-sample-friendly. Rather than relying on a single approach, BayesCurveFit integrates multiple carefully selected Bayesian techniques to enhance performance and usability.

The pipeline includes multiple model fitting and validation steps, each designed to complement the others. Algorithmically, BayesCurveFit operates in three key stages: Simulated Annealing (SA) for rapid initial parameter estimation, Markov Chain Monte Carlo (MCMC) for mode verification and potential correction, and Bayesian Model Averaging (BMA) to integrate predictions and optimize results across multiple models. Simulations demonstrate that BayesCurveFit delivers more robust

parameter estimates than regression-based methods when handling data with technical variations. In a real-world case study, BayesCurveFit identified novel molecule candidates missed by regression-based approaches using a limited number of tests. To showcase its versatility beyond dose-response modeling, we applied BayesCurveFit to tumor growth modeling for model selection, with results provided in the supplementary materials. From a usability standpoint, BayesCurveFit features an intuitive visualization interface, allowing users to explore optimal configurations before scaling up to batch-process large datasets. To facilitate real-time Bayesian analysis, we provide tutorial examples demonstrating how to implement parallelization and pipeline integration, supporting its adoption across diverse research environments and industry applications.

2 Methods and Workflow Design

BayesCurveFit is an open-source Python package designed for flexible curve fitting using any user-defined function involving an independent variable x, a dependent variable y, and fitting parameters θ. It streamlines the typically complex Bayesian workflow (Figure 1a) with an intuitive interface, making it accessible to users with minimal programming experience. By leveraging Bayesian inference, BayesCurveFit serves as a robust alternative to traditional regression analysis, particularly for modeling undersampled data, where uncertainty quantification is crucial (Figure 1b). The package employs a default Gaussian-Laplace probability density function (PDF), which effectively handles outliers by blending properties of Gaussian and Laplace distributions. This design mitigates the undue influence ("punishment") of extreme outliers on likelihood estimation, a common issue in regression-based methods. For advanced users and developers, BayesCurveFit is highly modular and customizable, allowing modifications such as replacing the default PDF with a skewed PDF or configuring alternative null models for posterior error probability (PEP) detection.

2.1 Initial Approximation and Posterior Sampling

Upon loading the data, BayesCurveFit employs Simulated Annealing (SA) to estimate the point-wise global optimum $\hat{\theta}$ and the nuisance parameters of the PDF. In its default configuration, the algorithm jointly estimates the parameters θ and the standard deviation σ of the Gaussian component by maximizing the conditional posterior probability:

$$(\hat{\theta}, \hat{\sigma}) = \arg\max_{\theta,\sigma} \left\{ P(\theta, \sigma \mid y) \propto P(y \mid \theta, \sigma)\, P(\theta)\, P(\sigma) \right\}. \tag{1}$$

Here, $\hat{\theta}$ and $\hat{\sigma}$ represent the MAP estimates of the parameters. The posterior probability $P(\theta, \sigma \mid y)$ is proportional to the product of the likelihood $P(y \mid \theta, \sigma)$ and the prior distributions $P(\theta)$ and $P(\sigma)$. While no specific prior distributions are imposed on θ and σ, users can define practical bounds for these parameters. For $P(\theta)$, users specify the parameter boundaries, restricting sampling to those regions. For $P(\sigma)$, the range is automatically set between 1×10^{-6} and the standard deviation (SD) of all observations, ensuring that the fitted SD remains lower than the observed noise. Within these boundaries, $P(\theta)$ and $P(\sigma)$ are effectively treated as uniform distributions, allowing the likelihood function to drive the approximation.

Additionally, a null model, assuming y is independent of x, is fitted using SA with a fixed σ equal to the standard deviation of all observations. This null model will later be compared to the alternative model in false positive estimations.

Following the SA-based initial approximation, BayesCurveFit employs Markov Chain Monte Carlo (MCMC) to sample the posterior distribution. Each MCMC chain is initialized from the global optima determined by SA and extended by 1,000 steps per iteration. The chains continue sampling until all Gelman-Rubin and Effective Sample Size diagnostics [Gelman et al., 2013] indicate convergence or until the maximum iteration limit is reached. For runs that fail to meet the convergence criteria, a warning is issued in the final output for the corresponding curve fitting.

The execution of MCMC was achieved using the emcee package for its efficiency and light weight Foreman-Mackey et al. [2019]. In the automated mode we defaulted DESnookerMove (dimensions > 3, Braak and Vrugt [2008]) and StretchMove (dimensions ≤ 3, Goodman and Weare [2010]) for exploration of the parameter's posterior space. Additionally a penalization mechanism was implemented as follows:

$$\text{penalties} = -0.5 \left(\frac{\|\theta_{\text{proposed}} - \hat{\theta}\|_2}{\sigma_{\text{penalty}}} \right)^2 \tag{2}$$

Here, distances represents the Euclidean distance between the proposed parameters (θ_{proposed}) and the estimated optimal parameters ($\hat{\theta}$). The penalties are calculated using a penalty function, where σ_{penalty} is a scaling parameter that controls the width of the penalty, defaulted to 3.0 to represent the 99.7% confidence interval in a standard normal distribution. This penalized log-likelihood function effectively keeps the sampler near the global optima, preventing it from drifting unless there is substantial evidence of a different peak.

2.2 Bayesian Model Averaging

After the posterior space has been sufficiently sampled, Gaussian Mixture Models (GMMs) were adopted to fit each parameter's distribution sampled by MCMC. We evaluated GMMs with varying numbers of components (from 1 to K) and selected the model with the highest sum log-likelihood for each parameter. This approach ensures that the chosen GMM provides the best fit to the posterior distribution. The probability density of each parameter θ_i, representing the optimum for each Gaussian component in the GMM, can be expressed as:

$$P(\theta_i \mid y) \approx \sum_{k=1}^{K_i} \pi_{ik} \mathcal{N}(\theta_i \mid \mu_{ik}, \sigma_{ik}^2) \tag{3}$$

where K_i is the number of components for the chosen GMM for parameter θ_i, π_{ik} are the mixture weights for each parameter/component combination, μ_{ik} are the means, and σ_{ik}^2 are the variances of the Gaussian components. The weighted mean parameter $\bar{\theta}_i$ is calculated as:

$$\bar{\theta}_i = \sum_{k=1}^{K_i} \pi_{ik} \mu_{ik} \tag{4}$$

The standard deviation of the parameter estimates $\bar{\sigma}_i$ is then:

$$\bar{\sigma}_i = \sqrt{\sum_{k=1}^{K_i} \pi_{ik}(\sigma_{ik}^2 + \mu_{ik}^2) - \left(\sum_{k=1}^{K_i} \pi_{ik}\mu_{ik}\right)^2} \tag{5}$$

To determine the log probability of the averaged model $\bar{\theta}$, we average over the discrete model space defined by the GMM fitting. The log probability is computed by combining the log-likelihoods of all candidate models, weighted by their respective GMM weights $w(\theta_i)$:

$$w(\theta_i) = \prod_{k=1}^{K_i} \pi_{ik} \tag{6}$$

$$\log P(\bar{\theta} \mid y) = \sum_{\theta_i \in \Theta} \log\left(w(\theta_i) P(\theta_i \mid y)\right) \tag{7}$$

Here, Θ represents the set of all possible parameter combinations $(\theta_1, \theta_2, \ldots, \theta_n)$ based on the GMM components.

2.3 Calculating Posterior Error Probability (PEP)

Once the best-estimated parameters $\bar{\theta}$ are determined, the Bayesian Information Criterion (BIC, [Kass and Raftery, 1995]) is calculated as:

$$\text{BIC} = p\log(N) - 2\log P(\bar{\theta} \mid y) \tag{8}$$

where p is the number of estimated parameters and N is the number of observations.

To compute the PEP, we first determine the differences between each model's BIC and the minimum BIC value:

$$\Delta_0 = \text{BIC}_0 - \min(\text{BIC}_0, \text{BIC}_1) \tag{9}$$

$$\Delta_1 = \text{BIC}_1 - \min(\text{BIC}_0, \text{BIC}_1) \tag{10}$$

Here, Δ_0 and Δ_1 represent the information loss for the null and alternative models, respectively. A smaller Δ indicates stronger support for a model, with $\Delta = 0$ identifying the best-fitting model.

The PEP is then computed using:

$$PEP = \frac{\exp\left(-0.5\Delta_0\right)}{\exp\left(-0.5\Delta_0\right) + \exp\left(-0.5\Delta_1\right)} \tag{11}$$

This formula estimates the probability that the null model is the true model relative to the alternative model based on their BIC values. Specifically, we compare two models: the null model where y is independent of x, and the alternative model with the fitting function. The PEP quantifies

the likelihood that the observations are incorrectly attributed to the alternative model, providing a direct measure of model probability derived from BIC without the need for p-values or additional adjustments.

3 Experiments

We implemented a simulation module in BayesCurveFit to generate datasets based on user-specified fit functions, error PDFs, and parameter bounds. A simulation study is used to evaluate the performance of BayesCurveFit by comparing it with regression-based curve fitting methods, using a 4-parameter log-logistic equation as the model and a null model where y is independent of x. Accuracy of the fitting is quantitatively assessed by calculating the root mean square error (RMSE) between the fitted curves and the ground truth derived from the parameters used in the simulation.

In a real case study, we applied BayesCurveFit to identify drug-gene interactions in the classic Kinobeads dataset [Klaeger et al., 2017] and compared the results with those obtained from regression-based analyses in the existing methods. A second real case study exploring BayesCurveFit's application to tumor growth data, with model selection to facilitate flexible yet fair model comparison, is included in the supplementary information (SI).

3.1 Datasets

For a fair comparison, public datasets in the well-known method CurveCurator [Bayer et al., 2023] are used. The 40 μM drugs subset from the Cancer Therapeutic Response Portal (CTRP) [Seashore-Ludlow et al., 2015], previously fitted using CurveCurator with OLS regression was used to generate the simulation data. This dataset consists of 16 independent observations across a dose range from 1×10^{-9} to 1×10^{-4} M. To model the error distribution, we calculated the residual errors for each observation based on the OLS-fitted parameters, which are the ground truth in our simulation experiment. We then explored fitting the error distribution using two models: a single Gaussian and a two-Gaussian mixture model. The two-Gaussian mixture model yielded better log-likelihood values in model selection (Figure S1), and was therefore chosen for the error distribution fitting (Figure 2a). Using the two-Gaussian mixture model to describe the error distribution, 1,000 random curves were simulated based on the 4-parameter log-logistic equation, with varying sample sizes from 5 to 50. Finally, random errors sampled from the fitted error distribution were added to the simulated curves (Figure 2b).

In the real case study, the Kinobeads dataset [Klaeger et al., 2017] was obtained from CurveCurator's data repository [Bayer et al., 2023], which contains 54,223 dose-response curves from testing 277 drugs. The values from 8 columns containing "Ratio" (labeled Ratio 1 through Ratio 8) are used as the dependent variables. The drug concentrations, ranged between 1×10^{-9} M and 1×10^{-4} M, are independent variables according to CurveCurator's Kinobeads TOML file. To independently assess the identified primary targets, reference data were sourced from the EMBL-EBI Open Targets database Ochoa et al. [2022] and the GtoPdb database [Harding et al., 2023]. Detailed specifications of the utilized datasets are available in the SI.

3.2 Benchmarking Experiment

To compare the performance of BayesCurveFit with OLS regression, we applied BayesCurveFit and OLS fitting (scipy.optimize.curve_fit with "trf" method) to all simulation data across varying sample sizes: 5, 8, 10, 15, 25, 30, and 50. For each fitted curve, we generated 50 data points from both the fitted model and the simulated ground truth, regardless of the sample size used for fitting. We then calculated the RMSE between these data points to assess the precision of the fitting. To provide a comprehensive comparison, we also included a soft L1 regression method (scipy.optimize.least_squares), which is known for its robustness to outliers. All of the fittings were implemented with the same bound settings.

Based on the main component of the two-Gaussian error distribution, which has a standard deviation $\sigma = 0.023$, we set an RMSE cutoff of 0.05 (approximately twice the standard deviation) to define the fits as accurate (RMSE ≤ 0.05) or not accurate (RMSE > 0.05) to distinguish the accurate fits with balanced sensitivity and specificity. Figure 2c shows the comparison results, where the points within the highlighted regions represent the number of fitted curves considered to accurately reflect the true dose-response trend. The results indicate that BayesCurveFit outperformed both OLS and soft L1 regression when the sample size was less than 30, particularly in the range of 8–10 observations. For larger sample sizes, all methods exhibited similar performance, approaching nearly 100% accurate fittings (Figure 2d).

3.3 Kinobeads Case Study

To compare BayesCurveFit and CurveCurator head-to-head, we re-analyzed the Kinobeads dataset originally used by CurveCurator, which contains 54,223 dose-response curves from 277 drugs, each with 8 observations plus one baseline value. We compared BayesCurveFit with CurveCurator by evaluating significant drug-gene interactions classified using a hyperbolic boundary (Figure 3a-b). While CurveCurator uses p-values and $\log_2$ fold change (Log2FC) of relative responses for classification, BayesCurveFit employs PEP alongside Log2FC. This results in slight differences in the boundary criteria due to the nature of the statistical measures used. Both methods adopt a Log2FC threshold of -0.5; however, BayesCurveFit uses a PEP threshold of 0.2 with a boundary function defined as $10^{-0.2}/(\text{abs}(\text{Log2FC}) - 0.5)$. In contrast, CurveCurator applies a p-value cutoff of 0.1 and defines a more stringent threshold at higher absolute Log2FC values. This adjustment is justified because PEP, representing the local false discovery rate, is generally more conservative than the p-value Käll et al. [2008]. In our case study (Figure 3c), we observed that a p-value of 0.1 is approximately equivalent to a PEP of 0.2 for comparable fits.

Upon comparing the classification results of the two methods, 4,795 interactions were identified by both methods as significant. BayesCurveFit detected an additional 976 interactions, while CurveCurator identified 14 additional interactions. Notably, BayesCurveFit captured several well-established interactions that were missed by CurveCurator (Figure 3a,b). For example, the *KIT* gene is a known target of Imatinib, which is used to treat gastrointestinal stromal tumors through KIT inhibitionHeinrich et al. [2003]. We further validated the significant interactions detected by both methods using manual annotations from the original study Klaeger et al. [2017] and cross-referenced drug-target information from the EMBL-EBI Open Targets database and the GtoPdb database. These databases include 337 drug-target interactions that overlap with the Kinobeads data. BayesCurveFit identified 285 interactions supported by manual labeling, including 7 primary targets of the corresponding drugs. Additionally, BayesCurveFit detected 7 interactions found in the

cross-referenced databases that were not identified by either the manual labeling or CurveCurator (Table 1).

We examined the interactions uniquely identified by each method. The drug-primary target interactions detected by BayesCurveFit but not CurveCurator showed clear dose-response trends that exceeded the -0.5 Log2FC threshold (Figure 4). In these instances, both methods produced similar fittings; however, CurveCurator's results seemed affected by one or two outliers, leading to deviations in the fitted curves and the omission of significant interactions. This meets the expectation that Bayesian methods offer increased robustness to outliers, improving the model sensitivity to detect meaningful interactions. In contrast, For the 14 interactions detected by CurveCurator but not by BayesCurveFit, we observed that discrepancies were often due to the presence of a final observation with an anomalously low value, which influenced the regression to deviate from the overall trend (Figure S2). Only a few cases showed similar fittings between both methods (e.g., Crenolanib–EPHA5, Pacritinib–CAMK4, and SAR-407899–PRKACB). To assess the validity of these interactions, we conducted a literature search but did not find strong evidence supporting significant interactions for these cases (details in Table S1).

Due to limited space in the main text, application of BayesCurveFit on another real use case for modeling tumor growth can be found in SI. To enable users to better visualize the differences in fitting, we have developed a web application for easy viewing of the fitted curves, available at `https://huggingface.co/spaces/ndu-bioinfo/bayescurvefit_viewer`.

4 Discussions

Curve fitting in scenarios with low sample sizes and outlier-prone data poses significant challenges for traditional regression methods like OLS and even robust regression techniques. These methods rely on standard assumptions about the error distribution and can be disproportionately influenced by outliers, leading to biased or unreliable results. In fields like drug discovery, environmental science, and other areas where samples are costly and scarce, it is crucial to have fitting methods that can effectively handle such data limitations. BayesCurveFit addresses these challenges by leveraging Bayesian inference to provide a more robust and reliable curve-fitting approach for under-sampled and outlier-contaminated data. By using a customized PDF that better handles outliers and properly estimating the PDF parameters during the optimization workflow, BayesCurveFit demonstrates superior performance in fitting data with low sample sizes compared to OLS and other regression-based methods. In Bayesian fitting, each additional observation that falls into the high-likelihood region of the PDF can rapidly increase the probability of the alternative hypothesis, allowing for a more responsive adjustment to the data. This contrasts with regression-based methods, which adjust the curve fitting at a slower pace as they gradually counterbalance the influence of outliers.

One of the key innovations in BayesCurveFit's workflow is performing MCMC sampling after SA optimization. By starting the MCMC chains from the global optimum determined by SA, we eliminate the need for manual decisions on chain burn-in that would be necessary if sampling began from random initial positions. This automation reduces human intervention and ensures that the sampling process is more efficient and reliable. Additionally, we employ GMMs to fit the posterior distributions before applying BMA. This approach allows us to average over the discrete model space defined by the GMM fits, simplifying the integration process by avoiding the need to compute complex marginal probabilities required when integrating over continuous parameters.

Despite this simplification, BMA has demonstrated excellent performance equivalent to methods involving integration over continuous parameters, both theoretically and empirically [Raftery and Zheng, 2003]. To ensure that the MCMC sampler remains efficient and effective in capturing the relevant posterior distributions without being misled by outliers or local maxima, we automated the selection of different move strategies within the MCMC sampling mechanism. Specifically, we utilized the `DESnookerMove` and `StretchMove` methods. The `DESnookerMove` efficiently explores complex, multimodal distributions, making it ideal for cases requiring thorough exploration across multiple modes Braak and Vrugt [2008]. In contrast, the `StretchMove` was chosen for its simplicity and efficiency when the parameter space is well covered by the ensemble Goodman and Weare [2010]. By combining these strategies, we mitigate the risk of non-converged sampling outcomes that could compromise the final optimal estimation.

Furthermore, we chose not to extensively explore the full posterior space for parameter uncertainty estimation because, by the nature of the experimental design, each curve fitting is expected to have one optimal solution. The presence of technical variations and outliers can introduce undesired local maxima far from the global optimum, leading to biased parameter estimates in the downstream BMA step. Re-sampling the posterior space using top likelihood quantiles might remove these small peaks, but it requires heuristic operations to determine appropriate thresholds and may distort the distribution of the optimal peak. Instead of removing the small peaks, we added a penalty term to the log-likelihood function used in the MCMC sampling, ensuring that the sampler remains close to the global optimum unless strong evidence justifies exploring further.

An additional advantage of BayesCurveFit lies in its use of PEP instead of traditional p-values for false detection. PEP is a form of local false discovery rate (local FDR), which provides a more powerful measure than the p-values by focusing on the probability of false positives for individual observations rather than controlling the error globally across all tests Efron et al. [2001]. This local nature allows BayesCurveFit to be sensitive in detecting true effects, especially in noisy or undersampled data. Furthermore, PEP allows for less stringent thresholds in cases of large effect sizes because large effect sizes inherently carry strong evidence of true positives. PEP adjusts for this by naturally giving less weight to outliers or sparse observations that fall outside the trend. As a result, BayesCurveFit reduces the influence of these anomalies, leading to more reliable detection of significant trends even in the presence of noise or variability. This contrasts with methods like OLS, where a small number of outliers can disproportionately affect the fitting and lead to false detections.

Building easy-to-use software tools is also our goal. In developing BayesCurveFit, we focused on integrating and optimizing existing well-established Bayesian principles and tools, such as the core Bayesian conditional probability theorem, MCMC convergence diagnostics, and PEP definitions. By leveraging existing software packages such as the dual annealing optimizer from scikit-learn for SA optimization and the `emcee` package for MCMC sampling, we concentrated on streamlining the workflow and enhancing usability and computational efficiency without reinventing entirely new methods. This integration allowed us to build a robust and efficient tool that is both user-friendly and theoretically sound.

5 Conclusions

Aiming at improving the regression-based curve fitting analysis, BayesCurveFit offers a robust and efficient solution for identifying drug-response relationship in undersampled and outlier-prone

datasets. Through workflow optimization, we successfully integrated multiple well-established Bayesian principles and computational methods into ByesCurveFit for performance enhancement. Our experiments demonstrated the effectiveness of BayesCurveFit to handle small sample sizes, mitigate impact of outliers, and provide reliable parameter estimates for curve fitting. Using both simulation and two real datasets, we showed that BayesCurveFit is not only useful for modeling drug-response relationship but also generally applicable to address other curve-fitting problems when experiment data are scarce or noisy. Focusing on pipeline extensibility and software usability, we implemented BayesCurveFit in a modular way and provided tutorials for pipeline runs, with real data examples publicly accessible online.

6 Code and Data Availability

The version of BayesCurveFit used in this manuscript is v0.5.1. The software and documentation are freely available on GitHub under the open-source Apache 2.0 license: `https://github.com/ndu-bioinfo/BayesCurveFit`.

All computational analyses were conducted on a machine equipped with an Apple M2 chip running macOS Sequoia 15.1.1. Supplementary data can be accessed at `https://github.com/ndu-bioinfo/BayesCurveFit_SI`.

Table 1: BayesCurveFit Identified Primary-Target Inhibitions Not Detected by CurveCurator

Drug	**Gene**	**Original Label**	**Bayes-CurveFit**	**Curve-Curator**	**CHEMBL Id**	**Primary targeted genes**
At-7519	CDK4	Yes	Yes	No	CHEMBL445813	CDK1, CDK4, CDK2, CDK5, CDK6, CDK7, CDK3, CDK16, CDK18, CDK8, CDK17, CDK11B, CDK15, CDK19, CDK14, CDK9, CDK10, CDK13, CDK20, CDK12
Danusertib	AURKA	Yes	Yes	No	CHEMBL402548	AURKB, AURKC, AURKA
Foretinib	PDGFRB	No	Yes	No	CHEMBL1230609	FLT1, FLT4, KDR, TEK, FLT3, MET, KIT, PDGFRB, PDGFRA
Icotinib	EGFR	Yes	Yes	No	CHEMBL2087361	EGFR
Imatinib	PDGFRB, KIT	No	Yes	No	CHEMBL941	ABL1, PDGFRB, KIT, BCR
Lucitanib	FGFR1	No	Yes	No	CHEMBL2220486	FLT1, FLT4, KDR, FGFR2, FGFR1
Ly-2584702	RPS6KB1	Yes	Yes	No	CHEMBL3545076	RPS6KB1
Midostaurin	PRKCD	No	Yes	No	CHEMBL608533	PDGFRB, PDGFRA, KIT, PRKD3, PRKCI, PRKCA, PRKCG, PRKCD, PRKCB, PRKCZ, PRKCE, PRKCH, PRKD1, PRKCQ, FLT3, KDR
Pha-793887	CDK6	No	Yes	No	CHEMBL1230607	CDK1, CDK4, CDK2, CDK5, CDK6, CDK7, CDK3, CDK16, CDK18, CDK8, CDK17, CDK11B, CDK15, CDK19, CDK14, CDK9, CDK10, CDK13, CDK20, CDK12
Regorafenib	BRAF, EPHA2, MAPK11	Yes	Yes	No	CHEMBL1946170	RET, FLT1, FLT4, KDR, TEK, FRK, KIT, DDR2, FGFR2, EPHA2, MAPK11, RAF1, PDGFRB, PDGFRA, ABL1, FGFR1, NTRK1, BRAF
Ucn-01	CDK2	No	Yes	No	CHEMBL574737	CDK4, PRKD3, PRKCI, PRKCA, PRKCG, PRKCD, PRKCB, PRKCZ, PRKCE, PRKCH, PRKD1, PRKCQ, CDK6, PDPK1, CHEK1, CDK2

References

F. P. Bayer, M. Gander, B. Kuster, and M. The. Curvecurator: a recalibrated f-statistic to assess, classify, and explore significance of dose–response curves. *Nature Communications*, 14(1):7902, 2023.

C. J. F. Braak and J. A. Vrugt. Differential evolution markov chain with snooker updater and fewer chains. *Statistics and Computing*, 18:435–446, 2008. URL `https://api.semanticscholar.org/CorpusID:14669369`.

B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76, 2017.

B. Efron, R. Tibshirani, J. D. Storey, and V. Tusher. Empirical bayes analysis of a microarray experiment. *Journal of the American statistical association*, 96(456):1151–1160, 2001.

D. Foreman-Mackey, W. Farr, M. Sinha, A. Archibald, D. Hogg, J. Sanders, J. Zuntz, P. Williams, A. Nelson, M. de Val-Borro, et al. emcee v3: A python ensemble sampling toolkit for affine-invariant mcmc. *The Journal of Open Source Software*, 4(43):1864, 2019.

A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin. *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013. ISBN 9781439840955.

J. Goodman and J. Weare. Ensemble samplers with affine invariance. *Communications in applied mathematics and computational science*, 5(1):65–80, 2010.

S. D. Harding, J. F. Armstrong, E. Faccenda, C. Southan, S. H. Alexander, A. P. Davenport, M. Spedding, and J. A. Davies. The IUPHAR/BPS Guide to PHARMACOLOGY in 2024. *Nucleic Acids Research*, 52(D1):D1438–D1449, 10 2023. ISSN 0305-1048. doi: 10.1093/nar/gkad944. URL https://doi.org/10.1093/nar/gkad944.

M. C. Heinrich, C. L. Corless, G. D. Demetri, C. D. Blanke, M. Von Mehren, H. Joensuu, L. S. McGreevey, C.-J. Chen, A. D. Van den Abbeele, B. J. Druker, et al. Kinase mutations and imatinib response in patients with metastatic gastrointestinal stromal tumor. *Journal of clinical oncology*, 21(23):4342–4349, 2003.

T. O. Hodson. Root mean square error (rmse) or mean absolute error (mae): When to use them or not. *Geoscientific Model Development Discussions*, 2022:1–10, 2022.

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90 (430):773–795, 1995.

S. Klaeger, S. Heinzlmeir, M. Wilhelm, H. Polzer, B. Vick, P.-A. Koenig, M. Reinecke, B. Ruprecht, S. Petzoldt, C. Meng, J. Zecha, K. Reiter, H. Qiao, D. Helm, H. Koch, M. Schoof, G. Canevari, E. Casale, S. R. Depaolini, A. Feuchtinger, Z. Wu, T. Schmidt, L. Rueckert, W. Becker, J. Huenges, A.-K. Garz, B.-O. Gohlke, D. P. Zolg, G. Kayser, T. Vooder, R. Preissner, H. Hahne, N. Tõnisson, K. Kramer, K. Götze, F. Bassermann, J. Schlegl, H.-C. Ehrlich, S. Aiche, A. Walch, P. A. Greif, S. Schneider, E. R. Felder, J. Ruland, G. Médard, I. Jeremias, K. Spiekermann, and B. Kuster. The target landscape of clinical kinase drugs. *Science*, 358(6367):eaan4368, 2017. doi: 10.1126/science.aan4368. URL https://www.science.org/doi/abs/10.1126/science.aan4368.

J. K. Kruschke and T. M. Liddell. The bayesian new statistics: Hypothesis testing, estimation, meta-analysis, and power analysis from a bayesian perspective. *Psychonomic bulletin & review*, 25:178–206, 2018.

R. Kurchin, G. Romano, and T. Buonassisi. Bayesim: A tool for adaptive grid model fitting with bayesian inference. *Computer Physics Communications*, 239:161–165, 2019.

L. Käll, J. D. Storey, M. J. MacCoss, and W. S. Noble. Posterior error probabilities and false discovery rates: Two sides of the same coin. *Journal of Proteome Research*, 7(1):40–44, 2008. ISSN 1535-3893. doi: 10.1021/pr700739d. URL https://doi.org/10.1021/pr700739d.

H. J. Motulsky and R. E. Brown. Detecting outliers when fitting data with nonlinear regression—a new method based on robust nonlinear regression and the false discovery rate. *BMC Bioinformatics*, 7(1):123, 2006. doi: 10.1186/1471-2105-7-123. URL https://doi.org/10.1186/1471-2105-7-123.

D. Ochoa, A. Hercules, M. Carmona, D. Suveges, J. Baker, C. Malangone, I. Lopez, A. Miranda, C. Cruz-Castillo, L. Fumis, M. Bernal-Llinares, K. Tsukanov, H. Cornu, K. Tsirigos, O. Razuvayevskaya, A. Buniello, J. Schwartzentruber, M. Karim, B. Ariano, R. Martinez Osorio, J. Ferrer, X. Ge, S. Machlitt-Northen, A. Gonzalez-Uriarte, S. Saha, S. Tirunagari, C. Mehta, J. Roldán-Romero, S. Horswell, S. Young, M. Ghoussaini, D. Hulcoop, I. Dunham, and E. McDonagh. The next-generation Open Targets Platform: reimagined, redesigned, rebuilt. *Nucleic Acids Research*, 51(D1):D1353–D1359, 11 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac1046. URL https://doi.org/10.1093/nar/gkac1046.

A. E. Raftery and Y. Zheng. Discussion: Performance of bayesian model averaging. *Journal of the American Statistical Association*, 98(464):931–938, 2003.

M. Roozbeh, S. Babaie-Kafaki, and Z. Aminifard. Two penalized mixed–integer nonlinear programming approaches to tackle multicollinearity and outliers effects in linear regression models. *Journal of Industrial & Management Optimization*, 17(6), 2021.

A. V. Sadybekov and V. Katritch. Computational approaches streamlining drug discovery. *Nature*, 616(7958):673–685, 2023.

J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, 2016.

B. Seashore-Ludlow, M. G. Rees, J. H. Cheah, M. Cokol, E. V. Price, M. E. Coletti, V. Jones, N. E. Bodycombe, C. K. Soule, J. Gould, B. Alexander, A. Li, P. Montgomery, M. J. Wawer, N. Kuru, J. D. Kotz, C. S.-Y. Hon, B. Munoz, T. Liefeld, V. Dančík, J. A. Bittker, M. Palmer, J. E. Bradner, A. F. Shamji, P. A. Clemons, and S. L. Schreiber. Harnessing Connectivity in a Large-Scale Small-Molecule Sensitivity Dataset. *Cancer Discovery*, 5(11):1210–1223, 11 2015. ISSN 2159-8274. doi: 10.1158/2159-8290.CD-15-0235. URL `https://doi.org/10.1158/2159-8290.CD-15-0235`.

M. Slugocki, A. B. Sekuler, and P. Bennett. Bayesfit: A tool for modeling psychophysical data using bayesian inference. *Journal of Open Research Software*, 7(1), 2019.

D. Sun, W. Gao, H. Hu, and S. Zhou. Why 90% of clinical drug development fails and how to improve it? *Acta Pharmaceutica Sinica B*, 12(7):3049–3062, 2022.

B. Thijssen, T. M. Dijkstra, T. Heskes, and L. F. Wessels. Bcm: toolkit for bayesian analysis of computational models using samplers. *BMC Systems Biology*, 10:1–8, 2016.

R. van de Schoot, S. Depaoli, R. King, B. Kramer, K. Märtens, M. G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen, et al. Bayesian statistics and modelling. *Nature Reviews Methods Primers*, 1(1):1, 2021.

V. Vyshemirsky and M. Girolami. Biobayes: a software package for bayesian inference in systems biology. *Bioinformatics*, 24(17):1933–1934, 2008.

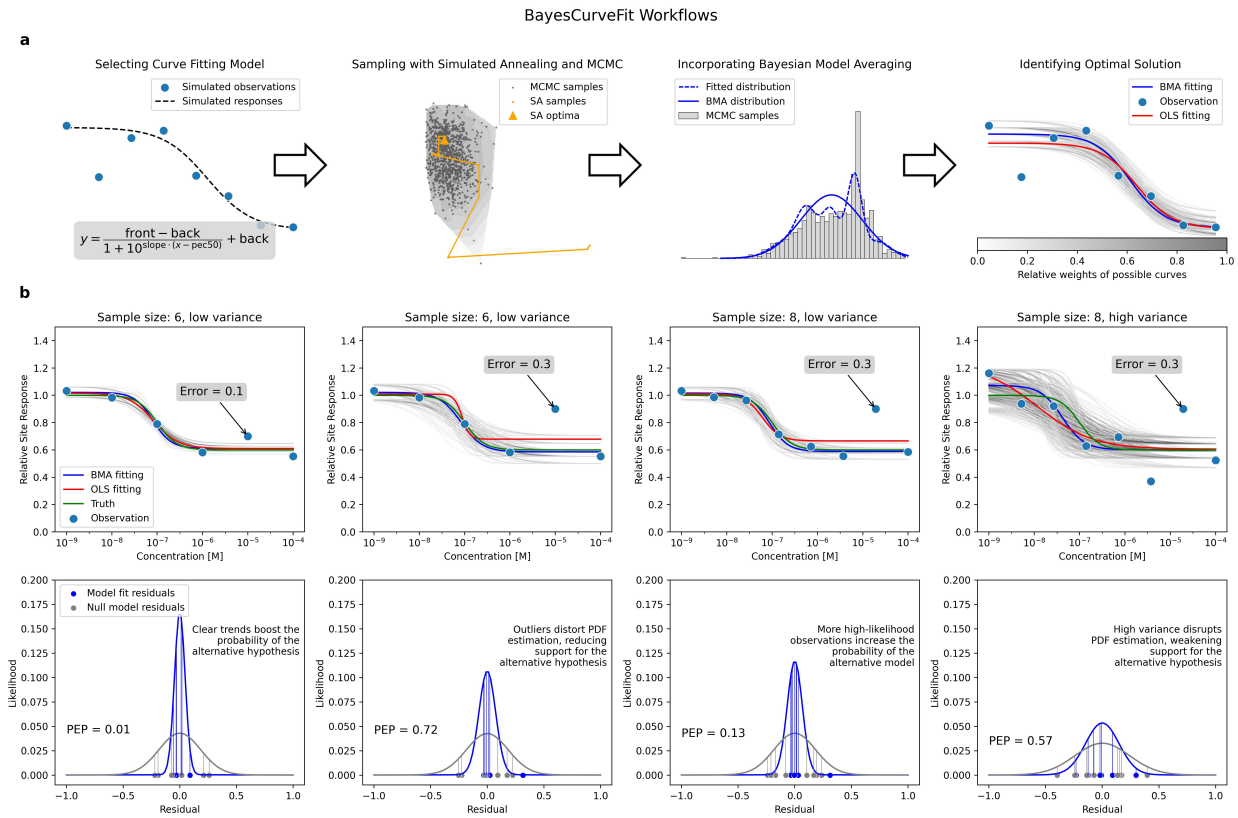


Figure 1: **BayesCurveFit Workflow and Demonstration of Fitting Undersampled Dose-Response Observations. a.** BayesCurveFit offers flexible, user-defined model selection for fitting observations. For example, we use a 4-parameter log-logistic function to fit simulated data (see SI for detailed parameter explanation). Once users select the appropriate model, the workflow begins by approximating a global optimum through simulated annealing (illustrated by the orange lines), where fitting parameters are estimated point-by-point. Next, Markov Chain Monte Carlo (MCMC) sampling is employed to further explore the posterior distribution. In this posterior space, potential solutions (peaks from the fitted distribution) are identified across varying probabilities based on MCMC samples. Finally, Bayesian Model Averaging (BMA) integrates these solutions, weighting them by their likelihoods to determine the best-fitting parameters. This approach leads to more accurate estimation of the true dose-response trend compared to regression, particularly under the scenarios described. **b.** The BMA fitting result is based on evidence supported by observations, which are influenced by three factors: 1) the level of variance in the observations, 2) the presence of higher-than-expected-variant errors which may or may not be considered outliers, and 3) the sample sizes. This demonstration uses simulated data where these factors are carefully controlled to ensure consistency across comparisons when they are not the focus. Each column represents a unique case with varying factors mentioned above. The top figure shows the BayesCurveFit (blue) and OLS (red) fitted results compared to the simulation's true source (green). The grey-scaled curves represent other possible solutions with varying weights, similar to the final plot in **a**. The bottom figure displays the approximated error distributions for both the null and the alternative hypothesis, with vertical lines indicating the likelihood of observation residuals from the fitted curves: blue for the fitted curve (alternative hypothesis) and grey for the null hypothesis. Additionally, the posterior error probability (PEP), which is computed as the probability of the null hypothesis in the total probability, is shown for each demonstration.

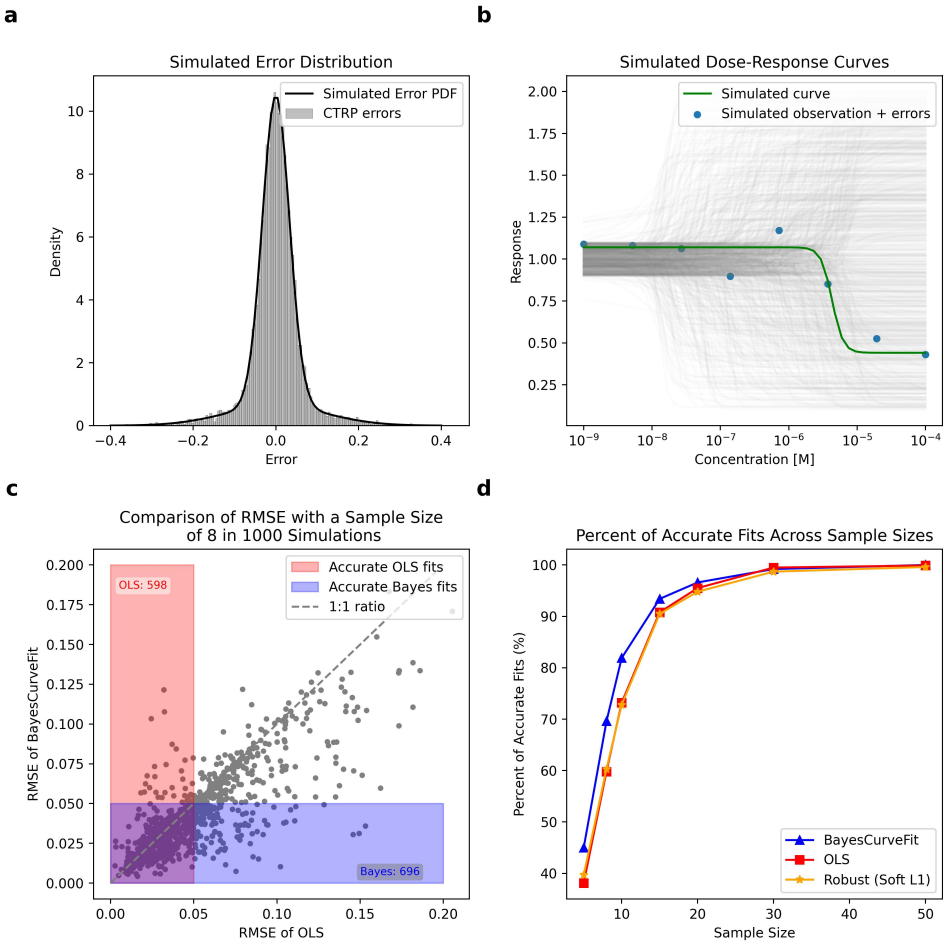


Figure 2: **Benchmark Comparison of BayesCurveFit and Regression-Fitted Dose-Response Models. a.** Simulated error distribution from the CTRP viability dataset, which includes 1211 OLS-fitted dose-response curves across 16 dose levels. The errors were calculated by assuming the fitted curve as the source of truth while excluding extreme residuals (>10 or ≤-10). The error distribution is then fitted with a two-Gaussian mixture model centered at 0, with corresponding σ values of 0.023 and 0.122, and weight factors of 0.808 and 0.192, respectively. **b.** Simulated dose-response curves with randomly sampled parameters: $5 < \text{pec50} < 8$, $0.1 < \text{slope} < 10$, $0.9 < \text{front} < 1.1$, and $0.1 < \text{back} < 2.0$. The curve highlighted in green shows one example simulation, and the blue dots represent the simulated observations based on this curve. The light grey curves show all of the simulated curves used for testing. **c.** Comparison between BayesCurveFit and OLS. Each grey dot represents a simulated curve. RMSE values produced by the OLS and the BayesCurveFit methods are plotted on X-axis and Y-axis, respectively. The colored rectangular regions highlight the number of fits below the 0.05 RMSE threshold, which were considered accurate fits in the test. **d.** Comparison of SoftL1, OLS and BayesCurveFit across varying sample sizes, showing the percentage of accurate fits at different sample sizes.

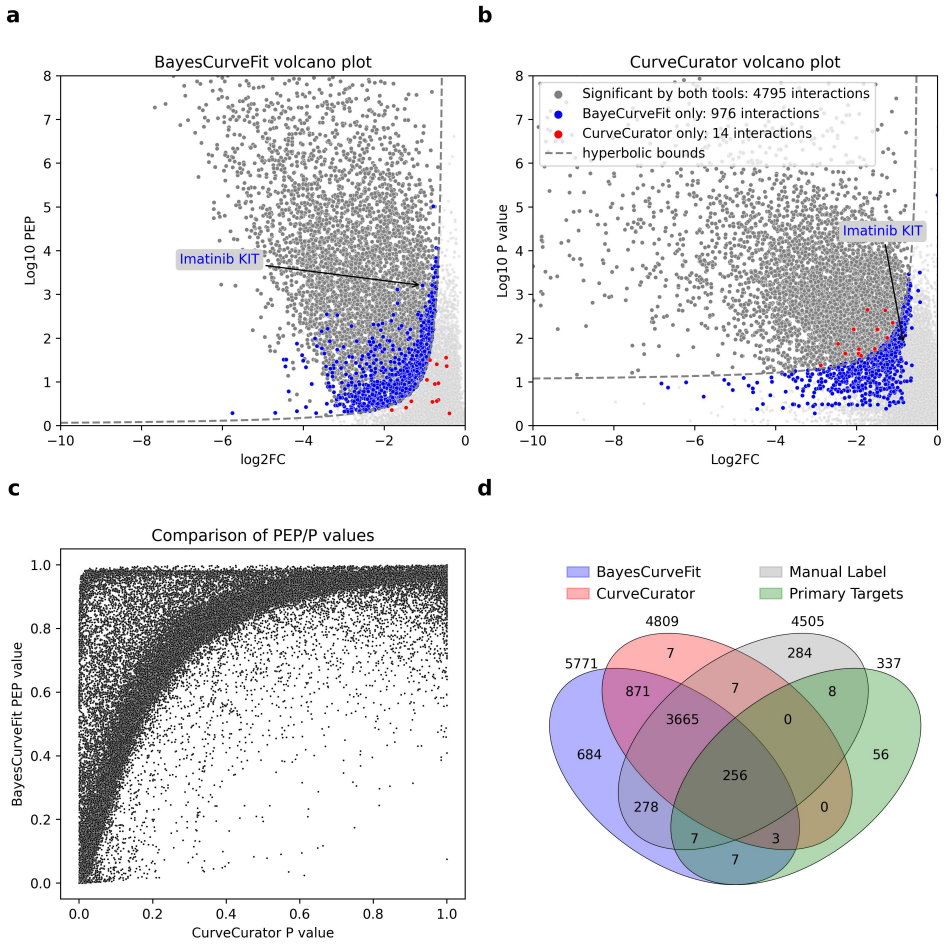


Figure 3: **Comparison of significant interactions predicted by BayesCurveFit and CurveCurator.** Volcano plots displaying significant interactions detected by BayesCurveFit **(a)** and CurveCurator **(b)**. Color highlighted points indicate interactions where the methods differ in their predictions. **(c)** Comparison of posterior error probabilities (PEP) from BayesCurveFit with p-values from CurveCurator for all interactions (each dot a interaction). **(d)** Venn diagram illustrating the overlap and differences in significant interactions identified by both methods, including manual annotations from the original study and primary targets cross-referenced from public databases.

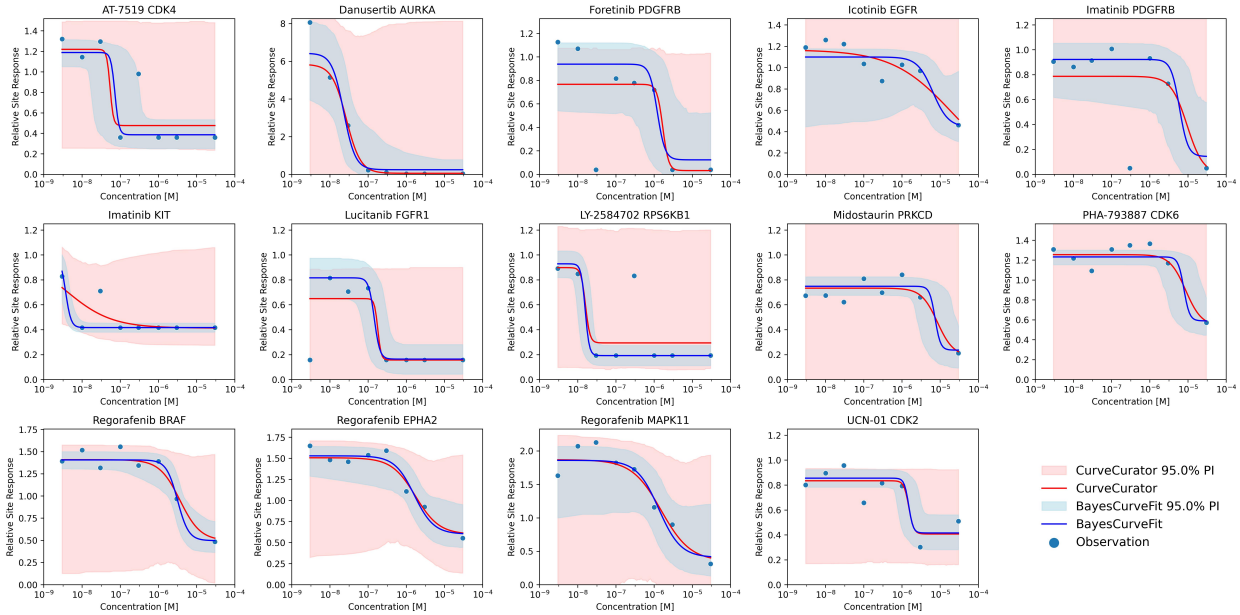


Figure 4: **The 14 primary drug-target gene inhibitions identified by BayesCurveFit but not detected by CurveCurator** The solid lines represent the fitted curves (blue: BayesCurveFit; red: CurveCurator). The shaded regions (blue: BayesCurveFit; red: CurveCurator) are the 95% prediction intervals (PIs). These PIs were determined by generating 1,000 simulated dose-response curves using parameter sets randomly sampled from the estimated error distributions of the parameters (based on their means and uncertainties). At 50 evenly spaced dose points covering the tested dose intervals, the 2.5th and 97.5th percentiles of the simulated responses define the bounds of the PIs, illustrating the uncertainty in model predictions due to variability in the parameter estimates. The robustness of BayesCurveFit, with a much smaller PI than CurveCurator, can be clearly seen.

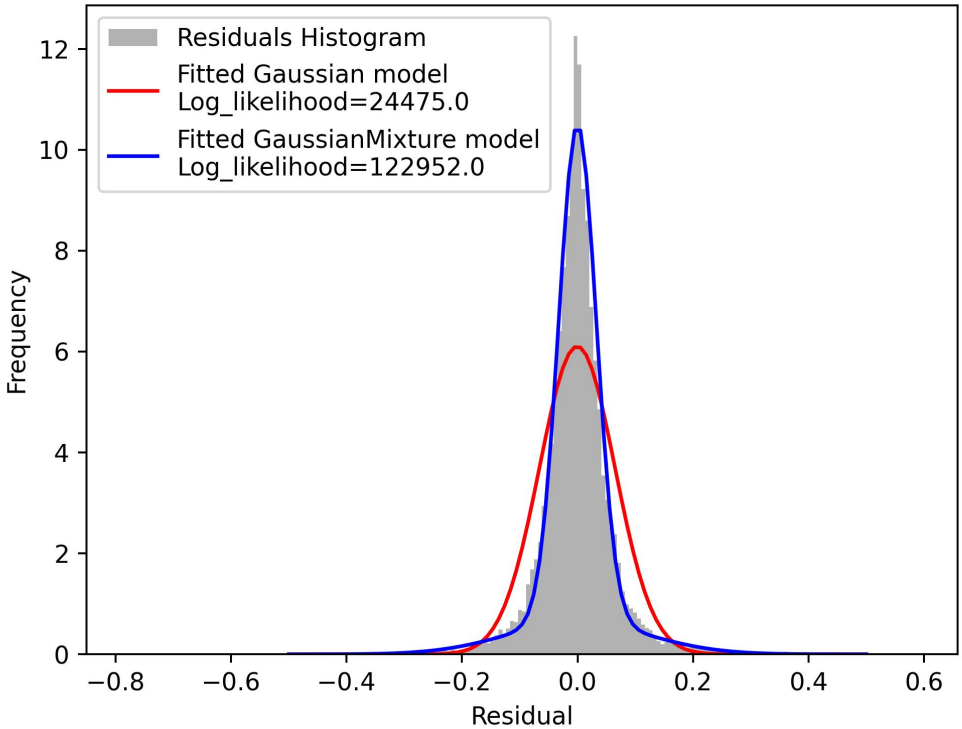


Figure S1: **Residual distributions from the CTRP dataset were analyzed and fitted using both a single Gaussian model and a two-component Gaussian Mixture Model. The latter provided a significantly better fit to the residual distribution compared to the single Gaussian model, as evidenced by the higher sum of log-likelihood values calculated from the respective fitted curves**

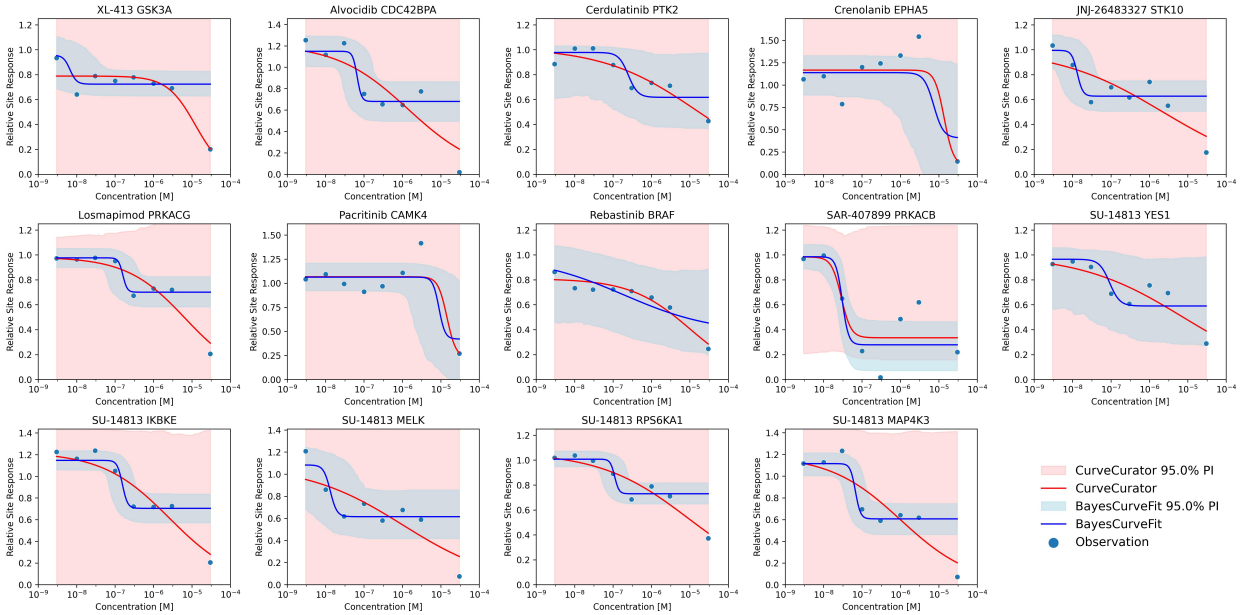
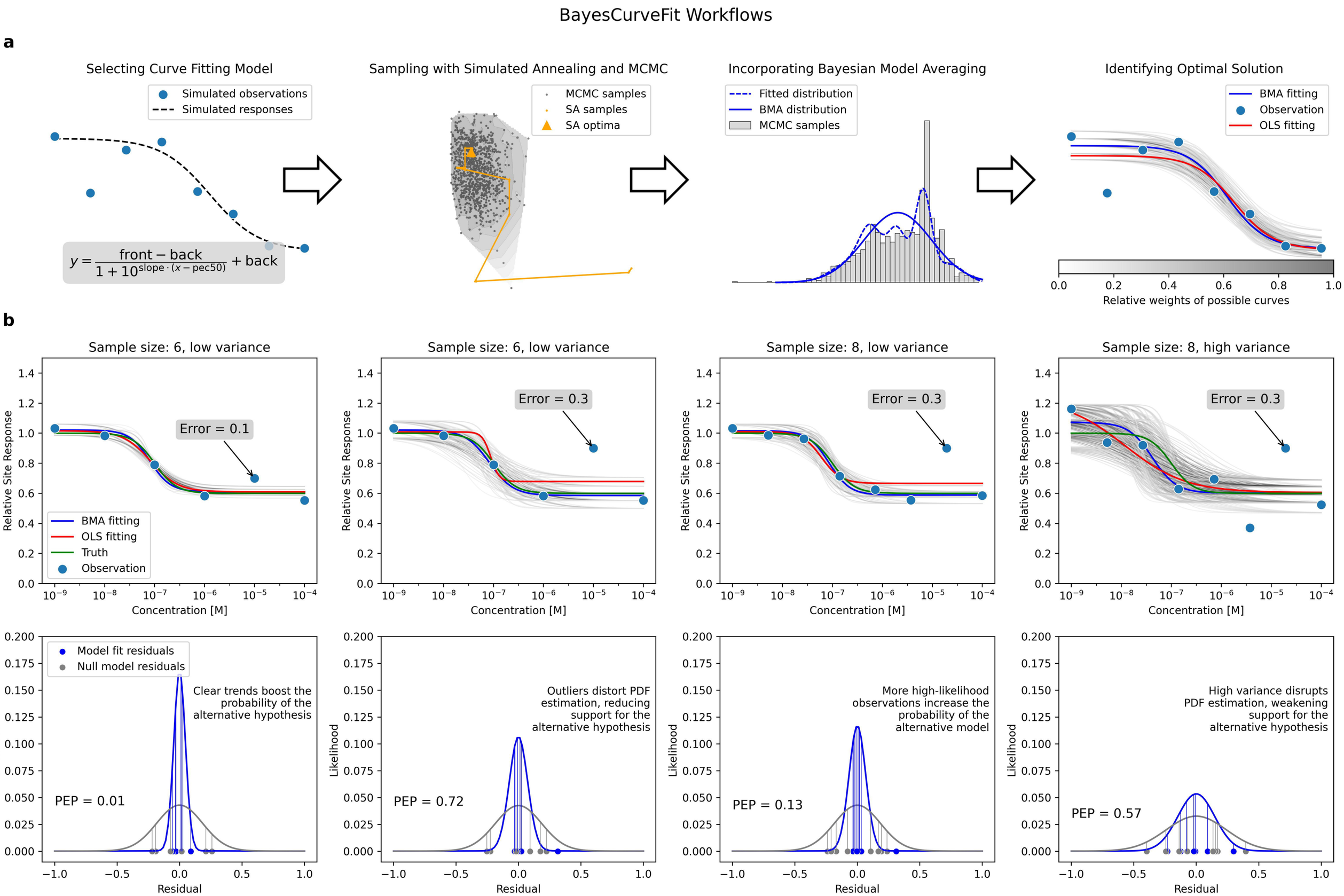
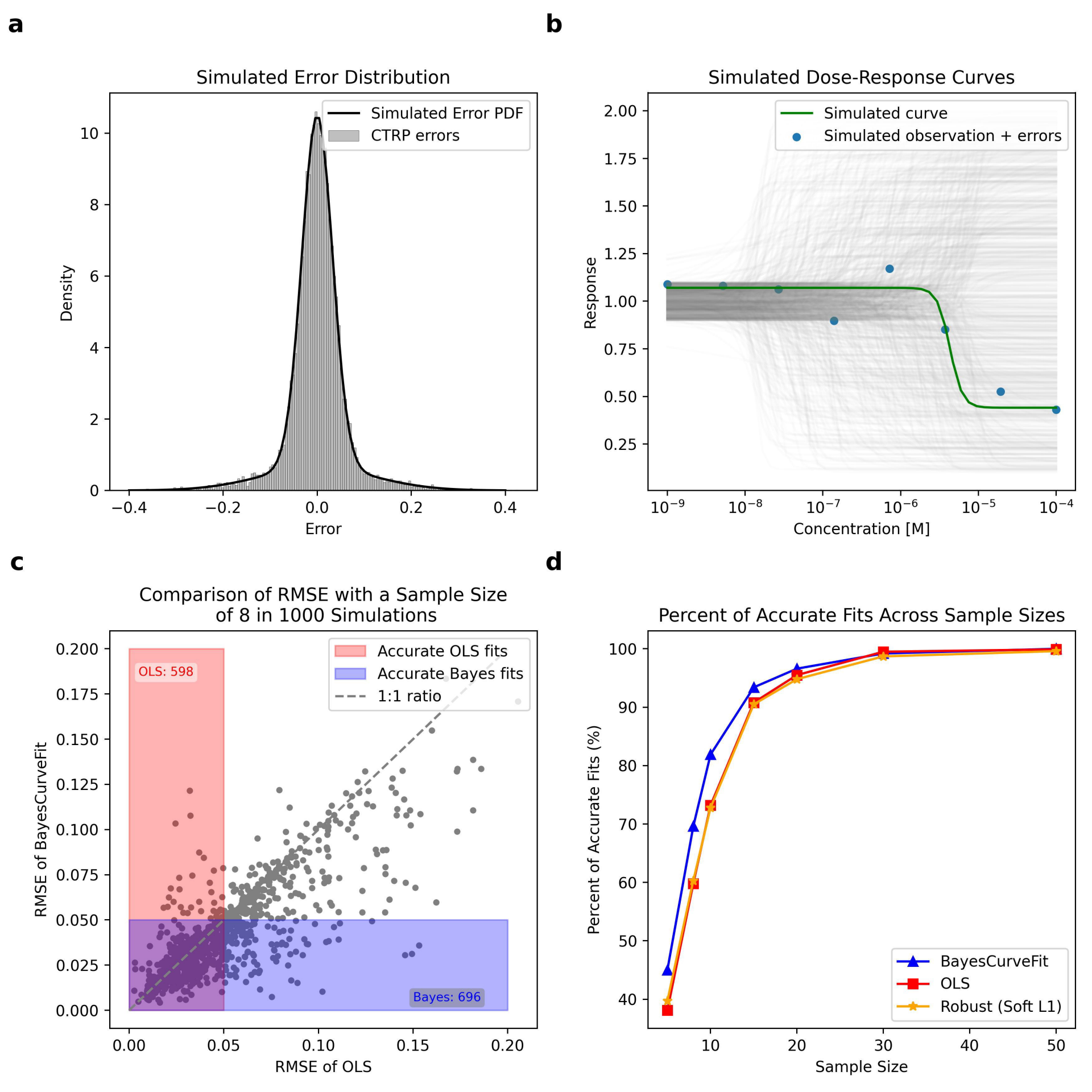
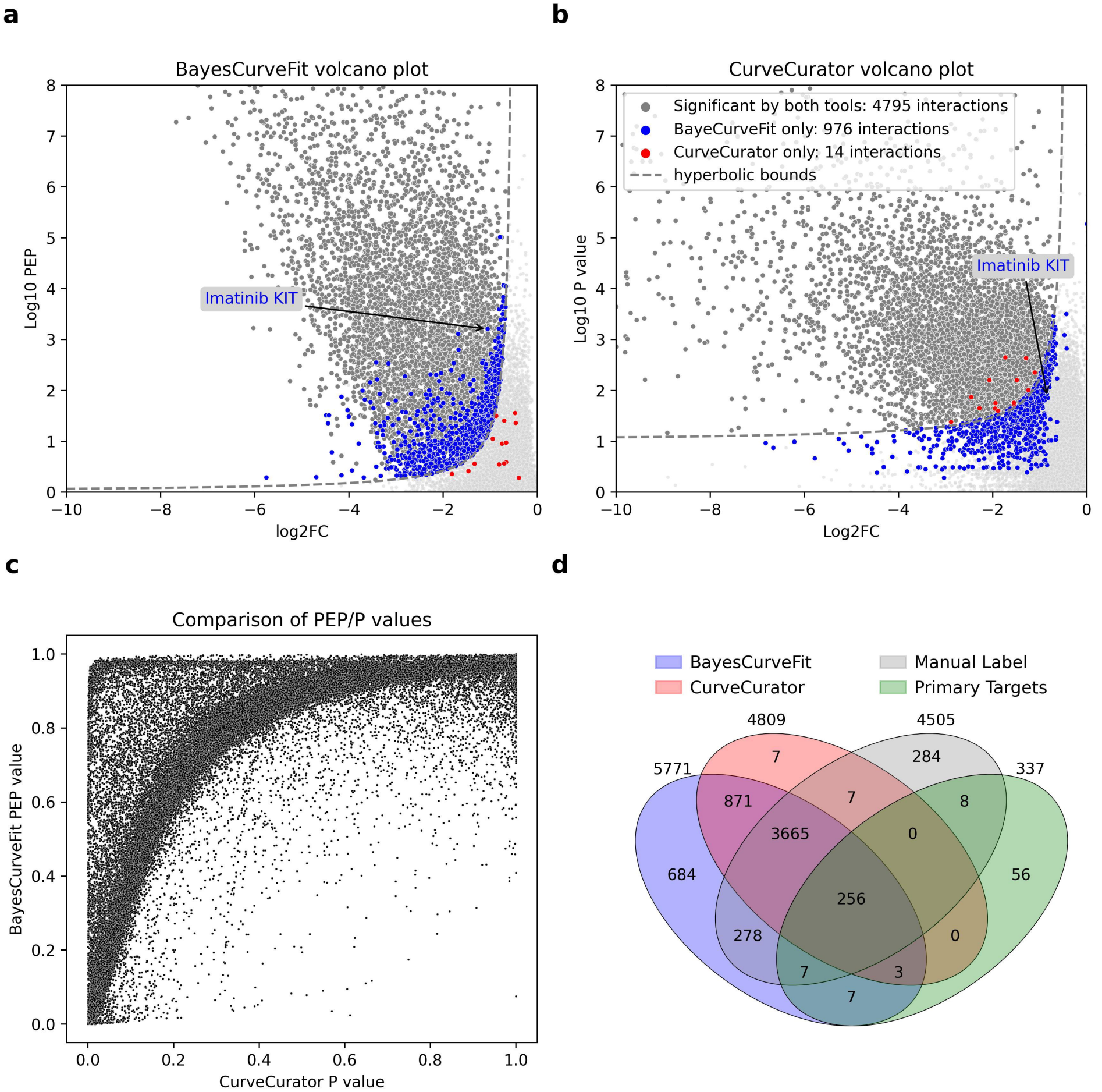
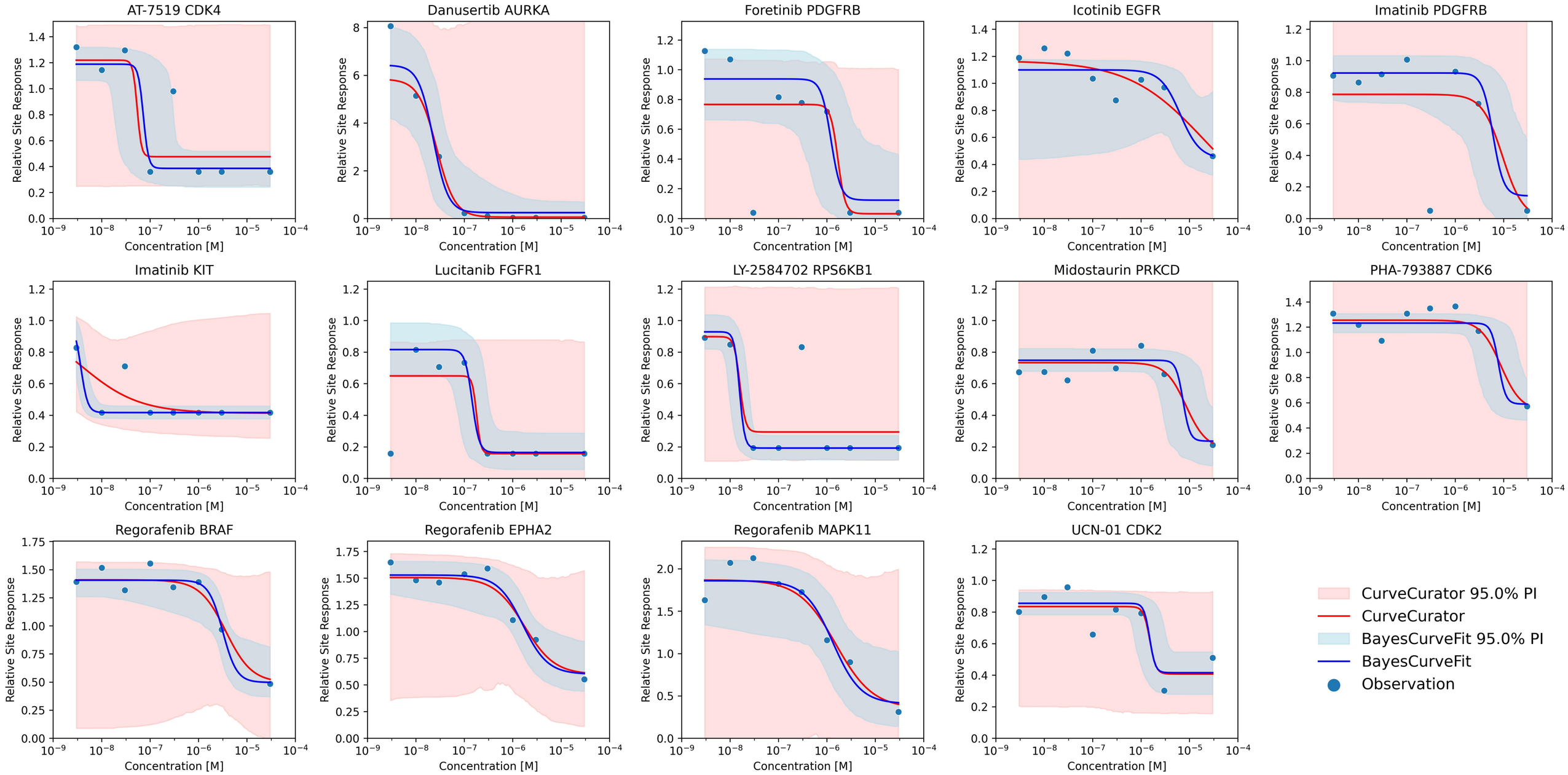


Figure S2: **The 14 possible false positive drug-target gene inhibitions identified by CurveCurator but not detected by BayesCurveFit** The solid lines represent the fitted curves (blue: BayesCurveFit; red: CurveCurator). The shaded regions (blue: BayesCurveFit; red: CurveCurator) are the 95% prediction intervals (PIs). These PIs were determined by generating 1,000 simulated dose-response curves using parameter sets randomly sampled from the estimated error distributions of the parameters (based on their means and uncertainties). At 50 evenly spaced dose points covering the tested dose intervals, the 2.5th and 97.5th percentiles of the simulated responses define the bounds of the PIs, illustrating the uncertainty in model predictions due to variability in the parameter estimates. The robustness of BayesCurveFit, with a much smaller PI than CurveCurator, can be clearly seen.


BayesCurveFit Workflows
a
Selecting Curve Fitting Model
Simulated observations
Simulated responses
$y = \frac{\text{front} - \text{back}}{1 + 10^{\text{slope} \cdot (x - \text{pec50})}} + \text{back}$
Sampling with Simulated Annealing and MCMC
MCMC samples
SA samples
SA optima
Incorporating Bayesian Model Averaging
Fitted distribution
BMA distribution
MCMC samples
Identifying Optimal Solution
BMA fitting
Observation
OLS fitting
0.0
0.2
0.4
0.6
0.8
1.0
Relative weights of possible curves
b
Sample size: 6, low variance
Sample size: 6, low variance
Sample size: 8, low variance
Sample size: 8, high variance
Relative Site Response
Concentration [M]
Error = 0.1
Error = 0.3
Error = 0.3
Error = 0.3
BMA fitting
OLS fitting
Truth
Observation
Model fit residuals
Null model residuals
Likelihood
Residual
Clear trends boost the probability of the alternative hypothesis
PEP = 0.01
Outliers distort PDF estimation, reducing support for the alternative hypothesis
PEP = 0.72
More high-likelihood observations increase the probability of the alternative model
PEP = 0.13
High variance disrupts PDF estimation, weakening support for the alternative hypothesis
PEP = 0.57


a
Simulated Error Distribution
Simulated Error PDF
CTRP errors
Density
Error
b
Simulated Dose-Response Curves
Simulated curve
Simulated observation + errors
Response
Concentration [M]
c
Comparison of RMSE with a Sample Size of 8 in 1000 Simulations
Accurate OLS fits
Accurate Bayes fits
1:1 ratio
OLS: 598
Bayes: 696
RMSE of BayesCurveFit
RMSE of OLS
d
Percent of Accurate Fits Across Sample Sizes
BayesCurveFit
OLS
Robust (Soft L1)
Percent of Accurate Fits (%)
Sample Size


a
BayesCurveFit volcano plot
Log10 PEP
log2FC
Imatinib KIT
b
CurveCurator volcano plot
Log10 P value
Log2FC
Imatinib KIT
Significant by both tools: 4795 interactions
BayeCurveFit only: 976 interactions
CurveCurator only: 14 interactions
hyperbolic bounds
c
Comparison of PEP/P values
BayesCurveFit PEP value
CurveCurator P value
d
BayesCurveFit
CurveCurator
Manual Label
Primary Targets
5771
4809
4505
337
7
284
871
7
8
3665
0
684
256
56
278
0
7
3
7


AT-7519 CDK4
Danusertib AURKA
Foretinib PDGFRB
Icotinib EGFR
Imatinib PDGFRB
Imatinib KIT
Lucitanib FGFR1
LY-2584702 RPS6KB1
Midostaurin PRKCD
PHA-793887 CDK6
Regorafenib BRAF
Regorafenib EPHA2
Regorafenib MAPK11
UCN-01 CDK2
Relative Site Response
Concentration [M]
CurveCurator 95.0% PI
CurveCurator
BayesCurveFit 95.0% PI
BayesCurveFit
Observation