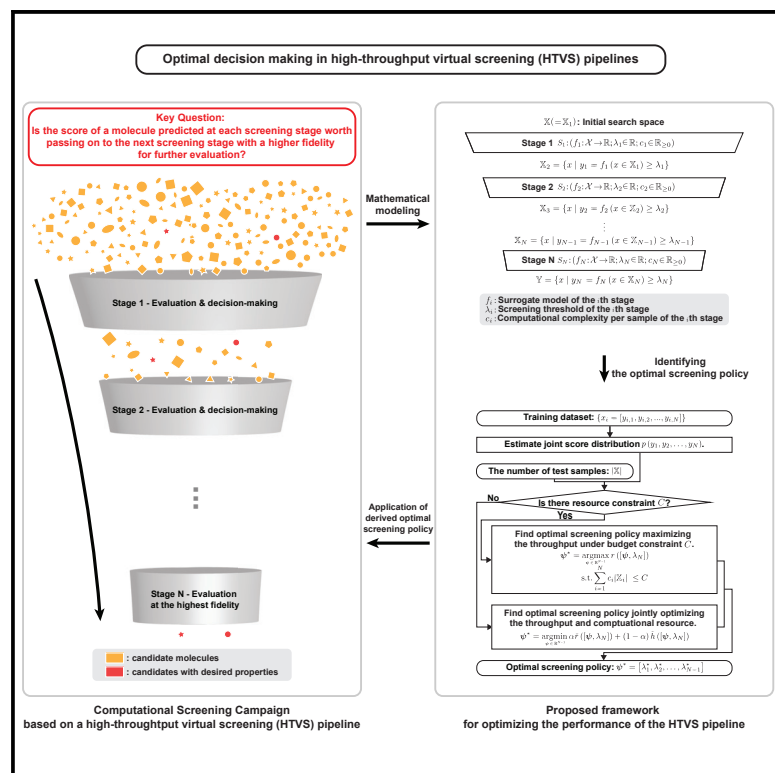


Patterns

Optimal decision-making in high-throughput virtual screening pipelines

Graphical abstract



Authors

Hyun-Myung Woo, Xiaoning Qian,
Li Tan, Shantenu Jha,
Francis J. Alexander,
Edward R. Dougherty,
Byung-Jun Yoon

Correspondence

bjyoon@ece.tamu.edu

In brief

The proposed framework can be used to design the optimal operational policy for a given high-throughput virtual screening campaign—e.g., for drug or material screening—to maximize the return on computational investment and accelerate scientific discoveries.

Highlights

- The problem of optimal virtual screening is mathematically formalized
- The proposed framework maximizes the ROI
- The proposed framework is generally applicable to any virtual screening task



Article

Optimal decision-making in high-throughput virtual screening pipelines

Hyun-Myung Woo,¹ Xiaoning Qian,^{2,3} Li Tan,³ Shantenu Jha,^{3,4} Francis J. Alexander,⁵ Edward R. Dougherty,² and Byung-Jun Yoon^{2,3,6,*}

¹Department of Biomedical & Robotics Engineering, Incheon National University, Incheon 22012, Republic of Korea

²Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA

³Computational Science Initiative, Brookhaven National Laboratory, Upton, NY 11973, USA

⁴Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854, USA

⁵Computing, Environment and Life Sciences, Argonne National Laboratory, Lemont, IL 60439, USA

⁶Lead contact

*Correspondence: bjyoon@ece.tamu.edu

<https://doi.org/10.1016/j.patter.2023.100875>

THE BIGGER PICTURE Screening large pools of molecular candidates to identify those with specific design criteria or targeted properties is demanding in various science and engineering domains. While a high-throughput virtual screening (HTVS) pipeline can provide efficient means to achieving this goal, its design and operation often rely on experts' intuition, potentially resulting in suboptimal performance. In this paper, we fill this critical gap by presenting a systematic framework that can maximize the return on computational investment (ROCI) of such HTVS campaigns. Based on various scenarios, we empirically validate the proposed framework and demonstrate its potential to accelerate scientific discoveries through optimal computational campaigns, especially in the context of virtual screening.



Development/Pre-production: Data science output has been rolled out/validated across multiple domains/problems

SUMMARY

The need for efficient computational screening of molecular candidates that possess desired properties frequently arises in various scientific and engineering problems, including drug discovery and materials design. However, the enormous search space containing the candidates and the substantial computational cost of high-fidelity property prediction models make screening practically challenging. In this work, we propose a general framework for constructing and optimizing a high-throughput virtual screening (HTVS) pipeline that consists of multi-fidelity models. The central idea is to optimally allocate the computational resources to models with varying costs and accuracy to optimize the return on computational investment. Based on both simulated and real-world data, we demonstrate that the proposed optimal HTVS framework can significantly accelerate virtual screening without any degradation in terms of accuracy. Furthermore, it enables an adaptive operational strategy for HTVS, where one can trade accuracy for efficiency.

INTRODUCTION

In various real-world scientific and engineering applications, the need for screening a large set of candidates to prioritize a small subset that satisfies certain criteria or possesses targeted properties arises fairly frequently. For example, since the coronavirus disease 2019 (COVID-19) outbreak, there have been significant concurrent efforts among various groups of scientists to identify or develop drugs that can provide a potential cure for this extremely infectious disease. One

such notable effort is IMPECCABLE (integrated modeling pipeline for COVID cure by assessing better leads)¹ whose operational objective is to optimize the number of promising ligands that potentially lead to the successful discovery of drug molecules. To this aim, IMPECCABLE utilized deep learning-based surrogates for predicting docking scores and multi-scale biophysics-based computational models for computing docking poses of compounds. Built on the strength of massive parallelism on exascale computing platforms combined with RADICAL-Cybertools (RCT) managing



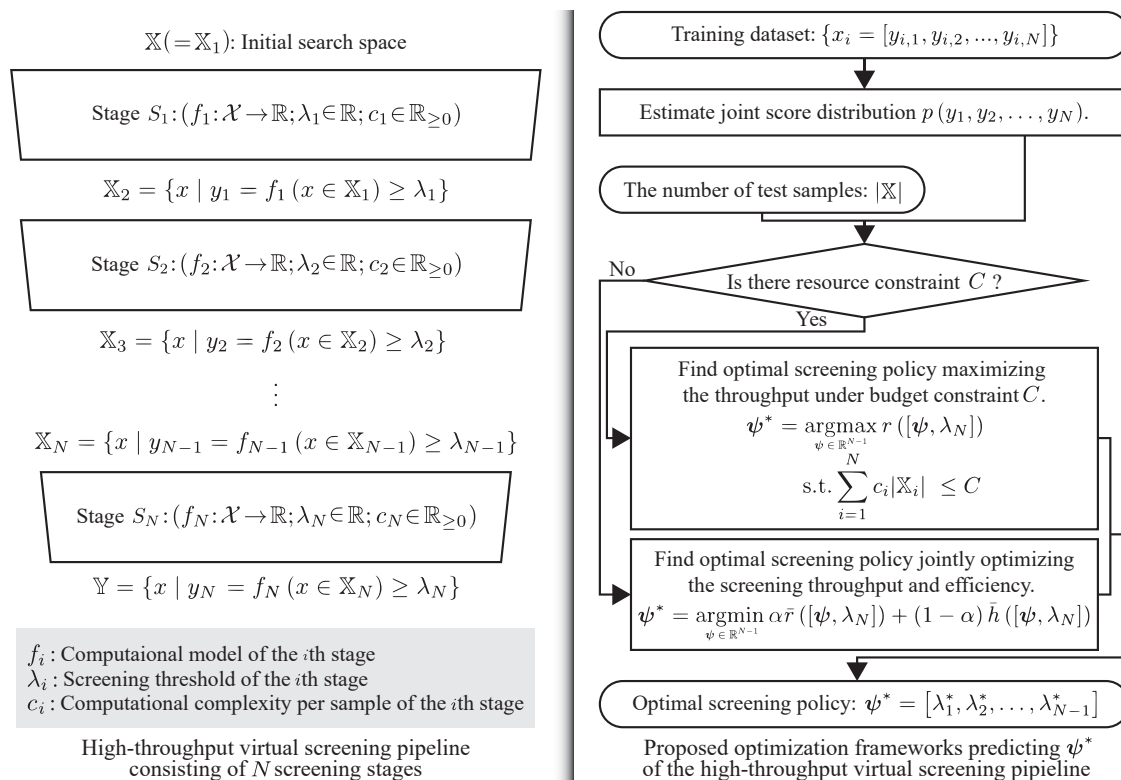


Figure 1. Illustration of a general HTVS pipeline (left) that consists of N screening stages for rapid and reliable identification of a set \mathbb{Y} of candidate molecules that likely possess the desired properties from a huge original set \mathbb{X}

Stage S_i evaluates all the molecules $x \in \mathbb{X}_i$, which passed the previous stage S_{i-1} , via computational model f_i . S_i passes the sample x to the next stage S_{i+1} if $f_i(x) \geq \lambda_i$. Otherwise, it discards the molecule. The proposed optimization framework shown on the right side predicts optimal screening policy $\psi^* = [\lambda_1^*, \lambda_2^*, \dots, \lambda_{N-1}^*]$ that yields either the maximal screening throughput under a resource budget constraint C or strikes the optimal balance between the screening throughput and efficiency according to the screening campaign scenario.

heterogeneous workflows, IMPECCABLE identified promising leads targeted at COVID-19.

Considering that severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) responsible for COVID-19 is known to rapidly mutate itself to create more infectious and deadlier variants,² such a drug screening process to identify effective antiviral drug candidates against a specific variant may have to be repeated as new variants emerge. However, considering that there are about 10^{68} compounds that can be considered for drug design in chemical space theoretically³ and the astronomical amount of computation that was devoted to the screening of drug candidates in IMPECCABLE¹ to screen 10^{11} candidates, this is without question a Herculean task that requires enormous resources and one that cannot be routinely repeated.

While different in scale and complexity, such high-throughput virtual screening (HTVS) pipelines have been widely utilized in various fields, including biology,^{4–8} chemistry,^{1,9–13} engineering,¹⁴ and materials science.^{15,16} However, the construction of such HTVS pipelines and the strategies for operating them heavily rely on expert intuition, often resulting in heuristic methods without systematic considerations for increasing the return on computational investment (ROI), a ratio of the number of desired candidates to computational resource investment in this context. It has remained a fundamental challenge to find a

general strategy to better construct or modify a screening pipeline and identify its operational policy resulting in a higher ROI despite an enormous search space.

In general, HTVS pipelines consist of multiple stages, where each stage consists of a scoring function that evaluates the property of the molecules with a different accuracy/fidelity and computational cost (e.g., one of the multi-fidelity computational models or surrogate models of a high-fidelity computational model). This is illustrated on the left side of Figure 1. At each stage in the pipeline, the molecular candidate is evaluated to determine whether the evaluation result appears promising enough to warrant passing it to the next—often more computationally expensive but more accurate—stage without unnecessarily wasting computational resources and time. In this way, the HTVS pipeline narrows down the number of candidate molecules, while sensibly allocating the available resources for investigating those that are promising and more likely to possess the desired property. The most promising candidates that remain at the end of screening may proceed to experimental validation, which is often more laborious, costly, and time-consuming. For example, an HTVS pipeline⁸ based on multi-fidelity surrogate models combined with an experimental platform successfully selected and reported a novel non-covalent inhibitor, MCULE-5948770040. The reported inhibitor has been

identified by screening over 6.5 million molecules, and it has been shown to inhibit the SARS-CoV-2 main protease. HTVS pipelines have been also widely used for materials screening. For example, a first-principles high-throughput screening pipeline for non-linear optical materials¹⁶ consisting of several computational predictors, based on density functional theory (DFT) calculations as well as data transformation and extraction methods, successfully identified deep-ultraviolet non-linear optical crystals that were reported in previous studies.^{17–23}

Although previous studies have demonstrated the advantages of constructing an HTVS pipeline for rapid screening of a huge set of molecules to narrow down the most promising molecular candidates that are likely to possess the desired properties, the problem of *optimal decision-making* for maximally increasing the ROI in such screening pipelines has not been extensively investigated to date. For example, how should one decide whether or not to pass a molecular candidate at hand to the next stage, given the score of the current stage? More specifically, in the HTVS example shown on the left side of Figure 1, how do we optimally determine the screening threshold λ_i of each stage S_i for a given HTVS structure to maximize the ROI? Furthermore, if we were to modify the HTVS structure or construct it from scratch by interconnecting multi-fidelity computational or surrogate models, what would be an advantageous structure of such an HTVS pipeline that is likely to lead to a higher ROI? This requires selecting the proper subset of the available multi-fidelity models, arranging them in the optimal order, and then exploring the interrelations among their predictive outcomes to make optimal operational decisions for the constructed HTVS pipeline.

To answer the aforementioned questions, we propose a computational framework for the optimization of HTVS pipelines that consist of multiple computational models with different costs and fidelity. The key idea is to estimate the joint probability distribution of predictive scores that result from the different stages constituting the HTVS pipeline, based on which we optimize the screening threshold values. We consider two optimization scenarios. First, we consider the case where the total computational budget is fixed, and the goal is to maximize the screening throughput defined as the number of promising candidates with the desired property within the given budget. Second, we assume that the computational budget is liquid and consider the case where we aim to balance the screening throughput and screening efficiency defined as a ratio of the number of promising candidates to the total computational consumption. We demonstrate the performance of the proposed HTVS pipeline optimization framework based on both *simulated data* as well as *real data*. In the simulated example, the joint distribution of the predictive scores from the multi-fidelity models at different stages is assumed to be known, based on which we extensively evaluate the performance of the proposed approach under various scenarios. As a second example, we consider the problem of screening for long non-coding RNAs (lncRNAs). In this example, we first construct an HTVS pipeline by interconnecting existing lncRNA prediction algorithms with varying costs and accuracy and apply our proposed framework for performance optimization. Both examples clearly demonstrate the advantages of our proposed scheme, which leads to a substantial reduction of the total computational cost at virtually no degradation in overall

screening performance. Furthermore, we show that the proposed framework enables one to make an informed decision to balance the trade-off between screening efficiency and screening throughput, where one could trade accuracy for higher efficiency, and vice versa.

RESULTS

In this section, we first formally describe the operational process of a general HTVS pipeline and provide a high-level overview of the proposed optimization framework. Two different scenarios will be considered. In the first scenario, the objective is to maximize the expected throughput of the HTVS pipeline under a fixed computational budget. In the second scenario, the objective is to jointly optimize the screening throughput and the computational efficiency of the pipeline. We validate the proposed optimization framework based on both synthetic and real data. First, we evaluate the performance of our optimization framework based on a four-stage HTVS pipeline, where the joint probability distribution of the predictive scores is assumed to be known. Next, we construct an HTVS pipeline for lncRNAs by interconnecting existing lncRNA prediction algorithms with different prediction accuracy and computational complexity. In this example, the joint distribution of the predictive scores from the different algorithms at different stages is learned from training data, based on which the proposed HTVS optimal framework is used to identify the optimal screening policy.

Overview of the proposed HTVS pipeline optimization framework

We assume that an HTVS pipeline consists of N screening stages $S_i : (f_i : \mathcal{X} \rightarrow \mathbb{R}; \lambda_i; c_i)$, $i = 1, 2, \dots, N$, connected in series as shown in Figure 1 (left), where $f_i : \mathcal{X} \rightarrow \mathbb{R}$ is a computational model for predicting the property of interest for a given molecule and λ_i is the screening threshold. The average computational cost per sample for f_i associated with the i th stage S_i is denoted by c_i . For simplicity, we will use S_i and f_i interchangeably. At each stage S_i , the corresponding surrogate model f_i is used to evaluate the property of all molecules $x \in \mathbb{X}_i$ that passed the previous screening stage S_{i-1} , where \mathbb{X}_i is given by:

$$\mathbb{X}_i = \{x | x \in \mathbb{X}_{i-1} \text{ and } f_{i-1}(x) \geq \lambda_{i-1}\}. \quad (\text{Equation 1})$$

By definition, we have $\mathbb{X}_1 \triangleq \mathbb{X}$, which contains the entire set of molecules to be screened. At stage S_i , every molecule $x \in \mathbb{X}_i$ whose property score $y_i = f_i(x)$ is below the threshold λ_i is discarded such that only the remaining molecules $x \in \mathbb{X}_{i+1}$ that meet or exceed this threshold are passed on to the next stage S_{i+1} . We assume that all molecules in \mathbb{X}_i at each stage S_i are batch processed to select the set of molecules \mathbb{X}_{i+1} that will be passed to the subsequent stage S_{i+1} , as it is often done in practice.^{24–26}

Although every stage S_i in the screening pipeline performs a down-selection of the molecules by assessing their molecular property based on the computational model $f_i(x)$ and comparing it against the threshold λ_i , we assume only the threshold values $\lambda_1, \lambda_2, \dots, \lambda_{i-1}$ of the first $N - 1$ stages will need to be determined while the threshold λ_N for the last screening stage S_N is predetermined. This reflects how such screening pipelines

are utilized in real-world scenarios. For example, in the IMPECCABLE pipeline,¹ as well as in many other computational drug discovery pipelines, potentially effective lead compounds that pass the earlier stages based on efficient but less accurate models will be assessed using computationally expensive yet highly accurate molecular dynamics (MD) simulations to evaluate the binding affinity against the target. Only the molecules whose binding affinity estimated by the MD simulations exceeds a reasonably high threshold set by domain experts may be further assessed experimentally as further steps are time consuming and labor intensive. Similarly, in a materials screening pipeline, the last screening stage may involve expensive calculations based on DFT, a quantum mechanical modeling scheme that is widely used for predicting material properties.^{27–35}

Based on this setting, our primary objective is to predict the optimal screening policy $\psi^* = [\lambda_1^*, \lambda_2^*, \dots, \lambda_{N-1}^*]$ that leads to the optimal operation of the HTVS pipeline. We consider two different scenarios. In the first scenario, we assume that the total computational budget for screening the candidate molecules is fixed, where the design goal would then be to identify the optimal screening policy that maximizes the screening throughput, namely, the percentage (or number) of potential molecules that meet or exceed the qualification in the last stage S_N (i.e., $f_N(x) \geq \lambda_N$). In the second scenario, we consider the case when the computational budget is not fixed and where the goal is to design the optimal policy that balances the screening throughput and screening efficiency (i.e., a ratio of the screening throughput to the required computational resource). This is done by defining and optimizing the objective function, which is a convex combination of the number of desired samples that we will miss and the overall computational cost.

Figure 1 (right) shows a flowchart summarizing the proposed approach for identifying the optimal screening policy $\psi^* = [\lambda_1^*, \lambda_2^*, \dots, \lambda_{N-1}^*]$ for the optimal operation of a given HTVS pipeline under the two screening scenarios described above. First, we estimate the joint distribution $p(y_1, y_2, \dots, y_N)$ of the predictive scores from the N stages based on the available training data. In the case where the probability density function (PDF) $p(y_1, y_2, \dots, y_N)$ is known *a priori*, this PDF estimation step will not be required. Given $p(y_1, y_2, \dots, y_N)$, we can predict the optimal screening policy $\psi^* = [\lambda_1^*, \lambda_2^*, \dots, \lambda_{N-1}^*]$ that leads to the optimal operational performance of the HTVS pipeline. Specifically, in the case where the total computational budget C is fixed, we find the optimal policy $\psi^* = [\lambda_1^*, \lambda_2^*, \dots, \lambda_{N-1}^*]$ that maximizes the screening throughput of the pipeline—i.e., the proportion of molecules that pass the last (and the most stringent/accurate) screening stage that meet the condition $f_N(x) \geq \lambda_N$ —under the budget constraint C . The formal definition of the given optimization problem is shown in Equation 4. Otherwise, we predict optimal screening policy $\psi^* = [\lambda_1^*, \lambda_2^*, \dots, \lambda_{N-1}^*]$ that jointly optimizes the screening throughput and screening efficiency based on a weighted objective function of the screening throughput and the computational efficiency. The formal definition of this joint optimization problem can be found in Equation 6. In this case, the balancing weight α can be used to trade throughput for computational efficiency, or vice versa. We note that the training dataset is only used for estimating the PDF $p(y_1, y_2, \dots, y_N)$ and not (directly) for finding the optimal screening

policy. In fact, the optimal policy $\psi^* = [\lambda_1^*, \lambda_2^*, \dots, \lambda_{N-1}^*]$ is determined by a function of up to three parameters: the joint score distribution $p(y_1, y_2, \dots, y_N)$, $|\mathbb{X}|$ (the number of potential molecules to be screened), and the total computational budget C (in the first screening scenario, where the computational budget is assumed to be limited).

Comprehensive performance analysis of the HTVS pipeline optimization framework

For comprehensive performance analysis of the proposed HTVS pipeline optimization framework, we consider a synthetic HTVS pipeline with four stages (i.e., $N = 4$), where the joint PDF of the predictive scores from all stages is assumed to be known. We vary the correlation levels between the scores from neighboring stages to investigate the overall impact on the performance of the optimized HTVS pipeline.

Specifically, we assume that the computational cost for screening a single molecule is 1 at stage S_1 , 10 at S_2 , 100 at S_3 , and 1,000 at S_N . As the per-molecule screening cost is fairly different across stages, the given setting for the synthetic HTVS pipeline allows us to clearly see the impact and significance of optimal decision-making on the overall screening throughput and efficiency of the screening pipeline.

Here, we consider the case when we have complete knowledge of the joint score distribution $p(y_1, y_2, y_3, y_4)$. The score distribution is assumed to be a multivariate uni-modal Gaussian distribution $\mathcal{G}(0, \Sigma(\rho))$, where the covariance matrix $\Sigma(\rho)$ is a Toeplitz matrix defined as follows:

$$\Sigma(\rho) = \begin{bmatrix} 1 & \rho & \rho - 0.1 & \rho - 0.2 \\ \rho & 1 & \rho & \rho - 0.1 \\ \rho - 0.1 & \rho & 1 & \rho \\ \rho - 0.2 & \rho - 0.1 & \rho & 1 \end{bmatrix}, \quad (\text{Equation 2})$$

where ρ is the correlation between neighboring stages S_i and S_{i+1} for $i = 1, 2, 3$. We assumed that the score correlation is lower between stages that are further apart, which is typically the case in real screening pipelines that consist of multi-fidelity models.

The primary objective of the HTVS pipeline is to maximize the number of potential candidates (i.e., screening throughput) that satisfy the final screening criterion (i.e., $f_4(x) \geq \lambda_4$) based on the highest-fidelity model at stage S_4 . The total number of all candidate molecules in the initial set \mathbb{X} is assumed to be 10^5 . We assume that we are given $\lambda_4 = 3.0902$ as prior information set by a domain expert, which results in 100 promising molecules (among 10^5 in \mathbb{X}) that satisfy the final screening criterion (i.e., $|\{x|f_4(x \in \mathbb{X}) \geq \lambda_4\}| = 100$). We validate the proposed HTVS optimization framework for two cases: first, for $\rho = 0.8$, where the neighboring stages yield scores that are highly correlated, and next, for $\rho = 0.3$ where the correlation is very low. Performance analysis results based on various other covariance matrices can be found in the supplemental information.

Performance of the optimized HTVS pipeline under computational budget constraint

Figure 2 shows the performance evaluation results for different HTVS pipeline structures optimized via the proposed framework

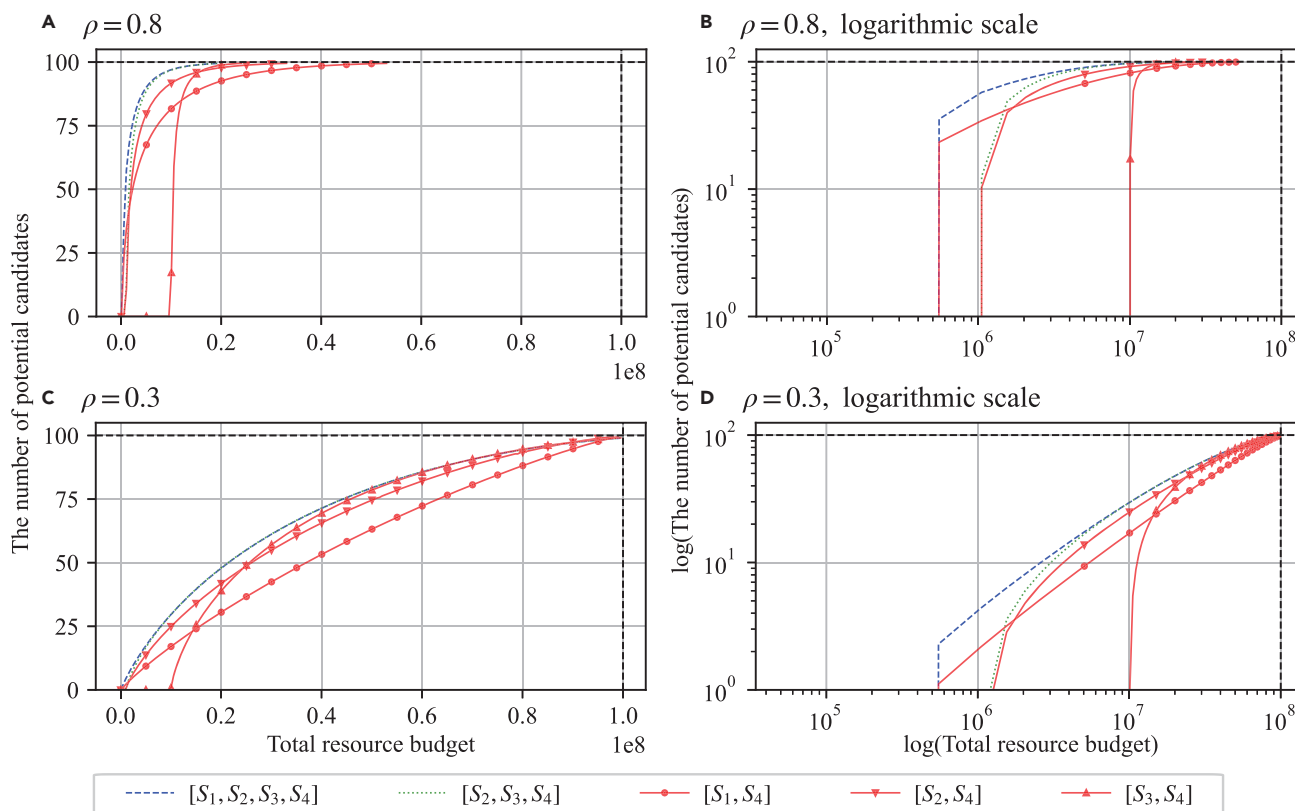


Figure 2. Performance assessment of the optimized HTVS pipelines

The number of candidate molecules that meet the desired screening criterion is shown as a function of the available computational budget. Results are shown for the case when the stages are highly correlated (A and B, $\rho = 0.8$) as well as when they have low correlation (C and D, $\rho = 0.3$). Note that (B and D), respectively, show identical contents of (A and C) in a logarithmic scale. The performance of the best-performing four-stage pipeline and the best-performing three-stage pipeline is shown. For comparison, we also show the performance of all two-stage pipelines. Note that only the best-performing configurations are shown for $N \geq 3$.

under a fixed computational resource budget. The total number of the desirable candidates detected by the pipeline is shown as a function of the available computational budget for two cases: (1) HTVS pipelines that consist of highly correlated stages ($\rho = 0.8$, Figures 2A and 2B) and (2) HTVS pipelines comprised of stages with low correlation ($\rho = 0.3$, Figures 2C and 2D). Note that Figures 2B and 2D, respectively, show identical contents of Figures 2A and 2C in a logarithmic scale for a better distinction between the curves with similar performance. The black horizontal and vertical dashed lines depict the total number of true candidates that meet the screening criterion (100 in this simulation) and the total computational budget required when screening all molecules in \mathbb{X} only based on the last stage S_4 (i.e., the highest-fidelity and most computationally expensive model), respectively. Figure 2 shows the performance of the best-performing $N = 4$ stage pipeline and that of the best-performing $N = 3$ pipeline. In addition, the performance of all $N = 2$ stage pipelines is shown for comparison.

First, as shown in Figures 2A and 2B, the performance curves of the pipelines consisting of only two stages (shown in red with different shapes of markers) demonstrate how each of the lower-fidelity stages S_1 – S_3 improves the screening performance when combined with the highest-fidelity stage S_4 and performance opti-

mized by our proposed framework. As shown in Figures 2A and 2B, the correlation between the lower-fidelity/lower-complexity stage S_i , $i = 1, 2, \dots, N - 1$, at the beginning of the HTVS pipeline and highest-fidelity/highest-complexity stage S_N at the end of the pipeline has a significant impact on the slope of the performance curve. For example, in the two-stage pipeline $[S_3, S_4]$, where the two stages are highly correlated to each other, we can observe the steepest performance improvement as the available computational budget increases. On the other hand, for the two-stage pipeline $[S_1, S_4]$, which consists of less correlated stages, the performance improvement is more moderate in comparison as the available computational budget increases. Note that the minimum required computational budget to screen all candidates is larger for the pipeline $[S_3, S_4]$ compared with that for $[S_1, S_4]$, which is due to the assumption that all candidates are batch processed at each stage. For example, with the minimum budget needed by pipeline $[S_3, S_4]$ to screen all candidates, the other pipelines $[S_1, S_4]$ and $[S_2, S_4]$ are capable of completing the screening and detecting more than 80% of the desirable candidates. Nevertheless, the detection performance improves with the increasing computational budget for all two-stage pipelines.

It is important to note that we can in fact simultaneously attain the advantage of using a lower-complexity stage (e.g., $[S_1, S_4]$)

that allows a “quick-start” with a small budget as well as the merit of using a higher-complexity stage (e.g., $[S_3, S_4]$) for rapid performance improvement with the budget increase by constructing a multi-stage HTVS pipeline and optimally allocating the computational resources according to our proposed optimization framework. This can be clearly seen in the performance curve for the four-stage pipeline $[S_1, S_2, S_3, S_4]$ (blue dashed line). The optimized four-stage pipeline consistently outperforms all other pipelines across all budget levels. Specifically, the optimized pipeline $[S_1, S_2, S_3, S_4]$ quickly evaluated all the molecular candidates in \mathbb{X} through the most efficient stage S_1 and sharply improved the screening performance through the utilization of more complex yet also more accurate subsequent stages in the HTVS pipeline in a resource-optimized manner. For example, the optimized four-stage pipeline detected 97% of the desirable candidates that meet the target criterion at only 10% of the total computational cost that would be required if one used only the last stage (which we refer to as the “original cost”). To detect 99% of the desired candidates, the optimized four-stage pipeline $[S_1, S_2, S_3, S_4]$ would need only about 14% of the original cost.

Among all three-stage pipelines (i.e., $N = 3$), pipeline $[S_2, S_3, S_4]$ yielded the best performance when performance-optimized using our proposed optimization framework (green dotted line in Figures 2A and 2B). As we can see in Figures 2A and 2B, the screening performance sharply increases as the available computational budget increases, thanks to the high correlation between S_4 and the prior stages S_2 and S_3 . However, due to the higher computational complexity of S_2 compared with that of S_1 , the optimized pipeline $[S_2, S_3, S_4]$ required a higher minimum computational budget for screening all candidate molecules compared with the minimum budget needed by a pipeline that begins with S_1 . Despite this fact, when the first stage S_2 in this three-stage HTVS pipeline is replaced by the more efficient S_1 , our simulation results (see Figure S41 in the supplemental information) show that the screening performance improves relatively moderately as the budget increases. Empirically, when all stages are relatively highly correlated to each other, the best strategy for constructing the HTVS pipeline appears to place the stages in increasing order of complexity and optimally allocate the computational resources to maximize the ROI. In fact, this observation is fairly intuitive and also in agreement with how screening pipelines are typically constructed in real-world applications.

Figures 2C and 2D show the performance evaluation results of the HTVS pipelines, where the screening stages are less correlated to each other ($\rho = 0.3$). Results are shown for different pipeline configurations, where the screening policy is optimized using the proposed framework to maximize the ROI. Overall, the performance trends were nearly identical to those shown in Figures 2A and 2B, although the overall performance is lower compared with the high correlation scenario ($\rho = 0.8$) as expected. While the screening performance of the optimized HTVS pipeline is not as good as the high-correlation scenario, the multi-stage HTVS pipeline with the optimized screening policy still provides a much better trade-off between the computational cost for screening and the detection performance. For example, if we were to use only the highest-fidelity model in S_4 for screening, the only way to trade accuracy for reduced

resource requirements would be to randomly sample the candidate molecules from \mathbb{X} and screen the selected candidates. The performance curve, in this case, would be a straight line connecting (0,0) and $(10^8, 100)$, below most of the performance curves for the optimized pipeline approach shown in Figures 2C and 2D. As in the previous case ($\rho = 0.8$), the best pipeline configuration was to interconnect all four stages, where the stages are connected to each other in increasing order of complexity.

Performance of the HTVS pipeline jointly optimized for screening throughput and efficiency

Table 1 shows the performance of the various HTVS pipeline configurations, where the screening policy was jointly optimized for both screening throughput and computational efficiency. The joint optimization problem is formally defined in Equation 6, and α was set to 0.5 in these simulations. As a reference, the first row (configuration $[S_4]$) shows the performance of solely relying on the last stage S_4 for screening the molecules without utilizing a multi-stage pipeline. The effective cost is defined as the total computational cost divided by the total number of molecules detected by the screening pipeline that satisfy the target criterion (i.e., average computational cost per detected candidate molecule or reciprocal of screening efficiency). The computational savings of a given pipeline configuration is calculated by comparing its effective cost to that of the reference configuration (i.e., $[S_4]$). As we can see in Table 1, our proposed HTVS pipeline optimization framework was able to significantly improve the overall screening performance across all pipeline configurations in a highly robust manner. For example, for $\rho = 0.8$, the optimized pipelines consistently led to computational savings ranging from 76.20% to 86.64% compared with the reference, while detecting 94%–99% of the desired candidates that meet the target criterion. Although the overall efficiency of the HTVS pipelines slightly decreases when the neighboring stages are less correlated ($\rho = 0.3$), the pipelines were nevertheless effective in saving computational resources. As shown in Table 1, the optimized HTVS pipelines detected 89%–96% of all desired candidate molecules with computational savings ranging between 36.46% and 54.87%.

For further evaluation of the proposed framework, we performed additional experiments based on the four-stage pipeline $[S_1, S_2, S_3, S_4]$. In this experiment, we first investigated the impact of α on the screening performance. Next, we compared the performance of the optimal screening policy with the performance of a baseline policy that mimics a typical screening scenario in real-world applications (e.g., see Saadi et al.¹). The baseline policy selects the top $R_s\%$ candidate molecules at each stage and passes them to the next stage while discarding the rest. This baseline screening policy is agnostic of the joint score distribution of the multiple stages in the HTVS and aims to reduce the overall computational cost by passing only the top candidates to subsequent stages that are more costly. Similar strategies are in fact often adopted in practice due to their simplicity. In our simulations, we assumed the proportion R_s is uniform across the screening stages. The performance evaluation results are summarized in Table 2. When the neighboring stages were highly correlated ($\rho = 0.8$), the optimized pipelines detected 100, 99, and 96 candidate molecules at a total cost of 19,727,704,

Table 1. Performance comparison of various HTVS pipeline structures jointly optimized via the proposed framework ($\alpha = 0.5$)

Configuration	High correlation ($\rho = 0.8$)				Low correlation ($\rho = 0.3$)			
	Potential candidates	Total cost	Effective cost	Computational savings (%)	Potential candidates	Total cost	Effective cost	Computational savings (%)
[S ₄]	100	100,000,000	1,000,000	0	100	100,000,000	1,000,000	0
[S ₁ , S ₄]	94	22,372,654	238,007	76.20	89	56,551,129	635,406	36.46
[S ₂ , S ₄]	96	15,511,702	161,580	83.84	90	43,620,751	484,675	51.53
[S ₃ , S ₄]	98	18,152,330	185,228	81.48	92	41,522,035	451,326	54.87
[S ₁ , S ₂ , S ₄]	97	17,890,176	184,435	81.56	94	53,340,817	567,456	43.25
[S ₁ , S ₃ , S ₄]	98	14,451,644	147,466	85.25	94	47,550,232	505,854	49.41
[S ₂ , S ₁ , S ₄]	97	18,291,054	188,568	81.14	94	53,513,582	569,293	43.07
[S ₂ , S ₃ , S ₄]	98	13,089,779	133,569	86.64	94	44,534,328	473,769	52.62
[S ₃ , S ₁ , S ₄]	99	19,505,326	197,023	80.30	94	48,708,112	518,171	48.18
[S ₃ , S ₂ , S ₄]	99	19,522,312	197,195	80.28	94	47,966,605	510,283	48.97
[S ₁ , S ₂ , S ₃ , S ₄]	99	14,147,264	142,902	85.71	96	50,336,621	524,340	47.57
[S ₁ , S ₃ , S ₂ , S ₄]	99	15,939,108	161,001	83.90	96	52,704,450	549,005	45.10
[S ₂ , S ₁ , S ₃ , S ₄]	99	14,348,794	144,937	85.51	96	50,366,503	524,651	47.53
[S ₂ , S ₃ , S ₁ , S ₄]	99	14,335,230	144,800	85.52	96	50,411,458	525,119	47.49
[S ₃ , S ₁ , S ₂ , S ₄]	99	20,560,571	207,682	79.23	96	53,249,970	554,687	44.53
[S ₃ , S ₂ , S ₁ , S ₄]	99	20,560,299	207,680	79.23	96	53,215,674	554,330	44.57

14, 147, 264, and 10, 926, 901, respectively. Interestingly, when α was reduced from 0.75 to 0.25 (i.e., trading accuracy for higher efficiency), the number of detected candidate molecules decreased only by 4 (i.e., from 100 to 96), while leading to an additional computational savings of 8 percentage points (i.e., from 80.27% to 88.62%). On the other hand, the performance of the baseline screening policy was highly unpredictable and very sensitive to the choice of R_s . For example, although the baseline with $R_s = 0.75$ found all the potential candidates, the effective cost of the baseline was significantly higher than that of the proposed optimized pipeline with $[\alpha = 0.75]$. For $R_s = 0.5$, the baseline detected 98 potential candidates (out of 100) with a total cost of 15,599,934, which was higher than the total cost of the optimized pipeline that detected 99 potential candidates. The baseline pipelines with $R_s = 0.1$ and 0.25 selected 26% and 78% of the potential candidates, respectively. Considering that the primary goal of such a pipeline is to detect the largest number of potential candidates, these results clearly show that this baseline screening scheme that mimics conventional screening pipelines resulted in unreliable screening performance even when the neighboring stages were highly correlated to each other. While the baseline may lead to reasonably good performance for certain R_s , it is important to note that we cannot determine the optimal R_s in advance as the approach is agnostic to the relationships between different stages. As a result, the application of this baseline screening pipeline may significantly degrade the screening performance in practice. When the correlation between the neighboring stages was low ($\rho = 0.3$), the overall performance of the proposed pipeline degraded as expected. In this case, the pipeline jointly optimized for screening throughput as well as screening efficiency with α set to 0.75, 0.5, and 0.25 detected 99, 96, and 86 potential candidates with the computational cost of 71,836,915, 50,336,621, and 28,563,886, respectively. As in the high correlation case, the

performance of the baseline scheme significantly varied and was sensitive to the choice of R_s .

Performance evaluation of the optimized HTVS pipeline for screening lncRNAs

In recent years, interest in lncRNAs have been constantly increasing in relevant research communities, as there is growing evidence that lncRNAs and their roles in various biological processes are closely associated with the development of complex and often hard-to-treat diseases including Alzheimer's disease,^{36–38} cardiovascular diseases,^{39,40} and several types of cancer.^{41–44} RNA sequencing techniques are nowadays routinely used to investigate the main functional molecules and their molecular interactions responsible for the initiation, progression, and manifestation of such complex diseases. Consequently, the accurate detection of lncRNA transcripts from a potentially huge number of sequenced RNA transcripts is a fundamental step in studying lncRNA-disease association. While several lncRNA prediction algorithms have been developed so far,^{45–48} each with its own pros and cons, no HTVS pipeline has been proposed to date for fast and reliable screening of lncRNAs.

To bridge the gap based on the proposed HTVS optimization framework, we first construct diverse HTVS pipeline structures by interconnecting four state-of-the-art lncRNA prediction algorithms—CPC2 (coding potential calculator 2),⁴⁷ CPAT (Coding Potential Assessment Tool),⁴⁵ PLEK (predictor of lncRNAs and messenger RNAs based on an improved k -mer scheme),⁴⁶ and lncFinder.⁴⁸ Then, we estimate the joint probability distribution of the scores from the given algorithms. The estimated score distribution is then used to derive the optimal screening policies in two different optimization scenarios. The performance of the optimized HTVS pipelines is comprehensively compared and analyzed from diverse perspectives.

Table 2. Performance comparison between the proposed pipeline $[S_1, S_2, S_3, S_4]$ jointly optimized for throughput and computational efficiency (with various α) and the baseline pipeline (with different screening ratio R_s) in terms of the total number of detected potential candidates after screening and the computational cost induced

approach	High correlation ($\rho = 0.8$)				Low correlation ($\rho = 0.3$)			
	Potential candidates	Total cost	Effective cost	Computational savings (%)	Potential candidates	Total cost	Effective cost	Computational savings (%)
Proposed ($\alpha = 0.75$)	100	19,727,704	197,277	80.27	99	71,836,915	725,625	27.44
Proposed ($\alpha = 0.5$)	99	14,147,264	142,902	85.71	96	50,336,621	524,340	47.57
Proposed ($\alpha = 0.25$)	96	10,926,901	113,822	88.62	86	28,563,886	332,138	66.79
Baseline ($R_s = 75\%$)	100	48,966,384	489,664	51.03	93	48,662,387	523,251	47.67
Baseline ($R_s = 50\%$)	98	15,599,934	159,183	84.08	69	15,600,165	226,089	77.39
Baseline ($R_s = 25\%$)	78	2,537,498	32,532	96.75	28	2,537,516	90,626	90.94
Baseline ($R_s = 10\%$)	26	400,000	15,385	98.46	6	400,008	66,668	93.33

In what follows, we describe the details of these steps and present the performance evaluation results of the optimal computational screening on real RNA transcripts in the GENCODE database.⁴⁹

Dataset and preprocessing

We collected the nucleotide sequences of *Homo sapiens* RNA transcripts from GENCODE version 38,⁴⁹ which consists of 48,752 lncRNA sequences and 106,143 protein-coding sequences. We filtered out sequences that contain any unknown nucleotides (other than A, U, C, or G) and sequences whose length exceeds 3,000 nt. This resulted in 45,216 lncRNA sequences and 79,030 protein-coding sequences. Next, we applied a clustering algorithm CD-hit⁵⁰ to lncRNAs and protein-coding RNAs, respectively, to remove redundant sequences. We finally obtained a set of 104,733 RNA transcripts consisting of 39,785 lncRNA sequences and 64,948 protein-coding sequences.

Construction of the lncRNA HTVS pipeline

For the construction of the lncRNA screening pipeline, we selected four state-of-the-art lncRNA prediction algorithms that have been shown to achieve good prediction performance: CPC2,⁴⁷ CPAT,⁴⁵ PLEK,⁴⁶ and LncFinder.⁴⁸

Table 3 summarizes the performance of the individual algorithm based on the GENCODE dataset, preprocessed as described previously. We assessed the accuracy, sensitivity, and specificity of the respective lncRNA prediction algorithms. For algorithm CPAT, which yields confidence scores between 0 and 1 rather than a binary output, we set the decision boundary to 0.5 for lncRNA classification. As shown in Table 3, LncFinder achieved the accuracy, sensitivity, and specificity of 0.8329, 0.7062, and 0.9678, respectively, outperforming all other algorithms in terms of accuracy and sensitivity. However, LncFinder also had the highest computational cost among the compared algorithms, where processing an RNA transcript required 2,495.6231 ms on average. CPAT was the second-best performer among the four in terms of accuracy and sensitivity. Furthermore, CPAT also achieved the highest specificity. CPC2 and PLEK were less accurate compared with LncFinder and CPAT in terms of accuracy, sensitivity, and specificity. Despite their high computational efficiency, both CPC2 and CPAT also outperformed PLEK based on overall accuracy.

As we previously observed from the performance assessment results based on the synthetic pipeline, the efficacy of the optimized HTVS pipeline is critically dependent on the interrelation between the stages constituting the pipeline. The proposed HTVS optimization framework aims to exploit the correlation structure across different screening stages to find the optimal screening policy either maximizing the screening throughput under a given budget constraint or striking the optimal balance between the screening throughput and screening efficiency depending on the optimization scenario under consideration. Here, we placed LncFinder—the most accurate and the most computationally costly algorithm among the four—in the final stage. In the first three stages in the HTVS pipeline, we placed CPC2, CPAT, and PLEK, in order of increasing computational complexity. After constructing the screening pipeline, we computed Pearson's correlation coefficient between the predictive output scores obtained from different algorithms. As shown in Figure 3, CPAT showed the highest correlation with LncFinder in the last stage (with a correlation coefficient of 0.93), the highest among the first three stages in the screening pipeline.

To apply our proposed HTVS pipeline optimization framework, we first estimated the joint probability distribution $p(y_1, y_2, y_3, y_4)$ of the predictive scores generated by the four different lncRNA prediction algorithms—CPC2 (y_1), CPAT (y_2), PLEK (y_3), and LncFinder (y_4)—via the expectation-maximization (EM) algorithm.⁵¹ For training, 4% of the preprocessed GENCODE data were used. Note that all the computational lncRNA identification algorithms considered in this study output protein-coding probabilities, hence a higher output value corresponds to a higher probability for a given transcript to be protein coding. Since our goal was to identify the lncRNAs, we multiplied the output scores generated by the algorithms by -1 such that higher values represent higher chances of being lncRNA transcripts. The screening threshold for the LncFinder in the last stage of the HTVS pipeline was set to $\lambda_4 = 0.2$, which leads to the optimal overall performance of LncFinder with a good balance between sensitivity and specificity.

Performance of the optimized lncRNA HTVS pipeline under computational budget constraint

Figure 4 shows the performance of the optimized lncRNA HTVS pipelines with various pipeline structures, where Figures 4A–4C illustrate the optimized pipelines with 2, 3, and 4 screening

Table 3. Performance of the four individual lncRNA prediction algorithms that constitute the lncRNA HTVS pipeline

Algorithm	Accuracy	Sensitivity	Specificity	Time per RNA (ms)
CPC2 ⁴⁷	0.7154	0.5760	0.9493	2.5265
CPAT ⁴⁵	0.8217	0.6861	0.9817	2.7336
PLEK ⁴⁶	0.7050	0.5666	0.9478	83.1765
LncFinder ⁴⁸	0.8329	0.7062	0.9678	2,495.6231

stages, respectively. The black horizontal dashed line indicates the total number of desired candidates (i.e., the total number of functional lncRNAs in the test set) and the black vertical dashed line shows the total computational cost (referred to as the original cost as before) that would be needed for screening all candidates based on the last stage LncFinder alone, without using the HTVS pipeline. Black vertical dotted lines are located at intervals of 1/10 of this original cost. Underneath each dotted line, the number of potential candidates (i.e., true functional lncRNAs) detected by each optimized HTVS pipeline is shown (see the columns in the table aligned with the dotted lines in the plots).

As before, we assumed that the candidates are batch processed at each stage. As a result, for a given pipeline structure, the computational cost of the first stage determines the minimum computational resources needed to start screening. The correlation between the neighboring stages was closely related to the slope of the corresponding performance curve, which is a phenomenon that we already noticed before based on synthetic pipelines. For example, at 10% of the original cost, the pipelines starting with PLEK (i.e., S_3) showed the worst performance among the tested pipelines in terms of the throughput. Specifically, $[S_3, S_4]$, $[S_3, S_1, S_4]$, $[S_3, S_2, S_4]$, $[S_3, S_1, S_2, S_4]$, and $[S_3, S_2, S_1, S_4]$ detected 6, 218, 6, 530, 6, 607, 6, 448, and 6, 601 lncRNAs, respectively. On the other hand, pipelines starting with either CPC2 or CPAT (i.e., S_1 or S_2) detected 9, 518 to 10, 033 lncRNAs at the same cost. In addition, pipelines $[S_2, S_4]$, $[S_1, S_2, S_4]$, $[S_2, S_1, S_4]$, $[S_2, S_3, S_4]$, $[S_3, S_2, S_4]$, $[S_1, S_2, S_3, S_4]$, $[S_1, S_3, S_2, S_4]$, $[S_2, S_1, S_3, S_4]$, $[S_2, S_3, S_1, S_4]$, $[S_3, S_1, S_2, S_4]$, and $[S_3, S_2, S_1, S_4]$ including the second stage associated with CPAT that is highly correlated to the last stage LncFinder showed the steepest performance improvement. As a result, all HTVS pipelines that include CPAT were able to identify nearly all true lncRNAs (i.e., 45, 697–47, 466) at only 50% of the original cost, regardless of which stage CPAT was placed in the pipeline.

While the structure of the HTVS pipeline impacts the overall screening performance, Figure 4 shows that our proposed optimization framework alleviated the performance dependency on the underlying structure by optimally exploiting the relationships across different stages. For example, even though the optimized pipeline $[S_1, S_2, S_4]$ outperformed the optimized pipeline $[S_1, S_2, S_3, S_4]$, which additionally included PLEK (i.e., S_3), the performance gap was not very significant. The maximum difference between the two pipeline structures in terms of the detected lncRNAs was 1, 202 when the computational budget was set at 40% of the original cost. However, when considering that PLEK (S_3) was computationally much more expensive compared with CPC2 (S_1) and CPAT (S_2) and also had a lower correlation

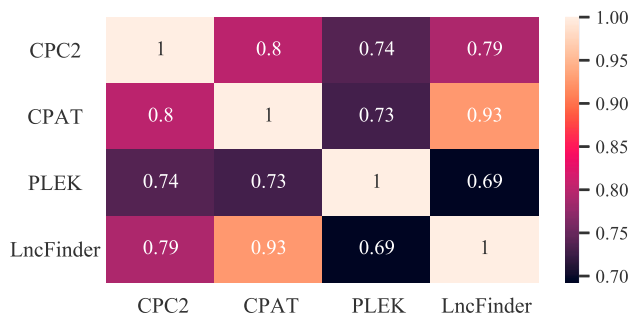


Figure 3. Heatmap showing Pearson's correlation coefficient between different stages

CPAT had the highest correlation to LncFinder. While PLEK was computationally more complex compared with CPAT, it showed a relatively lower correlation to LncFinder.

with LncFinder (S_4), the throughput difference of 1, 202 was only about 2.4% of the total lncRNAs in the test dataset, which is relatively small. Moreover, this throughput difference was drastically reduced as the available computational resources increased. For example, when the computational budget was set at 70% of the original cost the throughput difference between the two pipelines was only 50.

In practice, real-world HTVS pipelines may involve various types of screening stages using multi-fidelity surrogate models. The computational complexity and the fidelity of such surrogate models may differ significantly, and the structure of the pipeline may vastly vary depending on the domain experts designing the pipeline. Considering these factors, an important advantage of our proposed HTVS pipeline optimization framework is its capability to consistently attain efficient and accurate screening performance that may weather the effect of potentially suboptimal design choices in constructing real-world HTVS pipelines.

Performance of the lncRNA HTVS pipeline jointly optimized for screening throughput and efficiency

We evaluated the performance of the lncRNA HTVS pipelines, jointly optimized for both screening throughput and efficiency based on the proposed framework with $\alpha = 0.5$. The results for various pipeline configurations are shown in Table 4. On average, the optimized HTVS pipeline detected 48, 372 lncRNAs out of 50, 266 total lncRNAs in the test dataset. The average effective cost was 3, 067. Pipeline configurations that include CPAT (S_2) achieved relatively higher computational savings (ranging from 41.86% to 46.25%) compared with those without S_2 (ranging from 27.73% to 36.52%). As we have previously observed, our proposed optimization framework was effective in maintaining its screening throughput and efficiency even when the pipeline included a stage (e.g., PLEK) that is less correlated with the last and the highest-fidelity stage (i.e., LncFinder). In fact, the inclusion of a less-correlated stage in the HTVS pipeline does not significantly degrade the average screening performance. This is because the proposed optimization framework enables one to select the optimal threshold values that can sensibly combine the benefits of the most computationally efficient stages (such as CPC2 and CPAT in this case) as well as

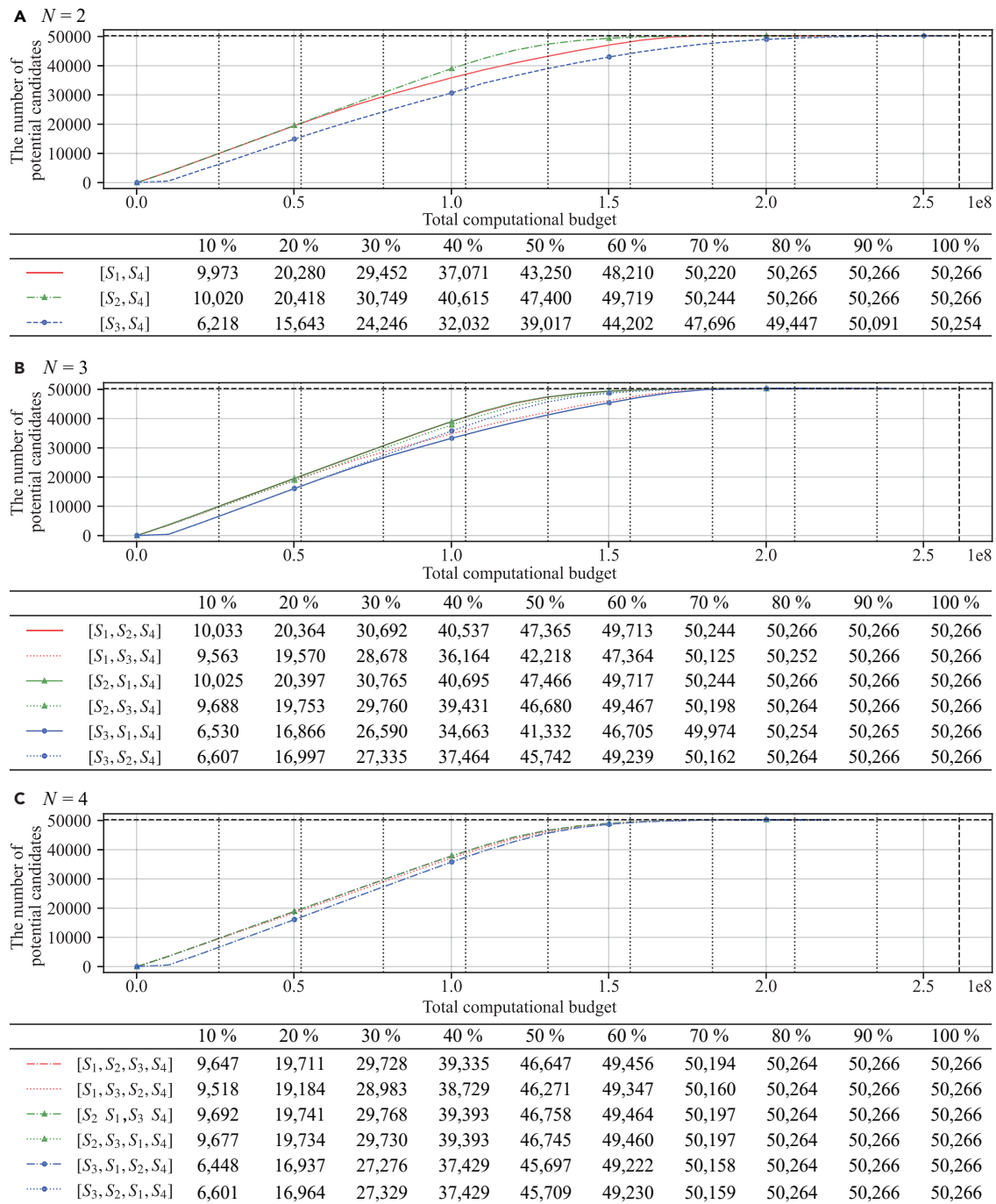


Figure 4. Performance evaluation of the optimized lncRNA HTVS pipeline

The number of potential candidates (i.e., lncRNAs) detected by the HTVS pipeline is shown under various computational budget constraints (x axis). Various different pipeline structures were tested, and the results show that the proposed optimization framework leads to reliable performance regardless of the structure used.

the highest-fidelity stage (LncFinder), thereby maximizing the expected ROCI. Similar observations can be made regarding the ordering of the multiple screening stages, as Table 4 shows that the average performance does not significantly depend on the actual ordering of the stages when the screening threshold values are optimized via our proposed framework. For example,

when using all four stages in the HTVS pipeline ($N = 4$), the optimized pipeline detected 48,402 lncRNAs on average and with consistent computational savings ranging between 42.27% and 44.00%. We also evaluated the accuracy of the potential candidates screened by the optimized HTVS pipeline based on four performance metrics: accuracy (ACC), sensitivity (SN),

Table 4. Performance evaluation of the lncRNA HTVS pipeline jointly optimized for throughput and efficiency ($\alpha = 0.5$)

Configuration	Potential candidates	Total cost (ms)	Effective cost	Computational savings (%)	Accuracy	Sensitivity	Specificity	F1
[S ₄]	50,266	261,374,090	5,200	0	0.8440	0.9264	0.7936	0.8186
[S ₁ , S ₄]	48,875	161,357,081	3,301	36.52	0.8429	0.9075	0.8034	0.8144
[S ₂ , S ₄]	47,950	134,366,143	2,802	46.12	0.8624	0.9215	0.8262	0.8357
[S ₃ , S ₄]	47,083	176,963,736	3,758	27.73	0.8450	0.8876	0.8188	0.8131
[S ₁ , S ₂ , S ₄]	48,210	134,748,992	2,795	46.25	0.8600	0.9216	0.8222	0.8333
[S ₁ , S ₃ , S ₄]	49,100	168,490,516	3,432	34.00	0.8442	0.9120	0.8026	0.8164
[S ₂ , S ₁ , S ₄]		134,812,024	2,796	46.23	0.8600	0.9216	0.8222	0.8334
[S ₂ , S ₃ , S ₄]	48,295	141,710,246	2,934	43.58	0.8602	0.9230	0.8218	0.8338
[S ₃ , S ₁ , S ₄]	49,119	171,803,403	3,498	32.73	0.8444	0.9124	0.8026	0.8166
[S ₃ , S ₂ , S ₄]	48,326	146,100,080	3,023	41.86	0.8600	0.9231	0.8214	0.8336
[S ₁ , S ₂ , S ₃ , S ₄]	48,402	140,954,256	2,912	44.00	0.8591	0.9228	0.8200	0.8326
[S ₁ , S ₃ , S ₂ , S ₄]	48,332	141,229,518	2,922	43.81	0.8587	0.9215	0.8203	0.8321
[S ₂ , S ₁ , S ₃ , S ₄]	48,409	141,022,859	2,913	43.98	0.8591	0.9229	0.8200	0.8326
[S ₂ , S ₃ , S ₁ , S ₄]	48,414	141,225,328	2,917	43.90	0.8591	0.9230	0.8200	0.8327
[S ₃ , S ₁ , S ₂ , S ₄]	48,424	145,321,388	3,001	42.29	0.8589	0.9228	0.8197	0.8324
[S ₃ , S ₂ , S ₁ , S ₄]	48,429	145,388,626	3,002	42.27	0.8589	0.9229	0.8197	0.8325

specificity (SP), and F1 score. Interestingly, all configurations except for [S₁, S₄] outperformed LncFinder in terms of ACC. In terms of SN, the optimized pipeline achieved an average sensitivity of 0.9177. All pipeline configurations resulted in higher specificity compared with LncFinder. Besides, pipeline configurations that include S₂ consistently outperformed LncFinder in terms of the F1 score.

Finally, we compared the performance of the optimized pipeline with that of the baseline approach that selects the top R_s % of the incoming candidates for the next stage, where R_s % is a parameter to be determined by a domain expert. For this comparison, we considered the four-stage pipeline [S₁, S₂, S₃, S₄]. The optimal screening policy was found based on our proposed framework using three different values of $\alpha \in \{0.25, 0.50, 0.75\}$. The baseline screening approach was evaluated based on four different levels of $R_s \in \{25\%, 50\%, 75\%\}$. The performance assessment results are summarized in Table 5. As shown in Table 5, the baseline approach detected fewer lncRNAs for all values of R_s compared with the optimized pipeline. Specifically, the jointly optimized pipeline detected 48,965, 48,402, and 47,106 lncRNAs at a cost of 148,155,016 ($\alpha = 0.75$), 140,954,256 ($\alpha = 0.50$), and 131,830,857 ($\alpha = 0.25$), respectively. On the other hand, the baseline approach with $R_s = 75\%$ detected only 39,079 lncRNAs at a total cost of 115,643,459. For $R_s = 50\%$ and $R_s = 25\%$, the baseline scheme detected only 12,653 and 1,402 lncRNAs, respectively. In terms of the four quality metrics (ACC, SN, SP, and F1), the optimized pipeline outperformed the baseline scheme in terms of ACC, SN, and F1. The optimized pipeline resulted in lower SP compared with the baseline. However, it should be noted that the potential candidates detected by the optimized HTVS pipeline are remarkably higher compared with the baseline approach. This is clearly reflected in the much lower SN of the baseline approach, as shown in Table 5. As a result, the baseline approach tends to achieve significantly lower ACC and F1 compared with the optimal screening scheme.

DISCUSSION

In this work, we proposed a general mathematical framework for identifying the optimal screening policy that can either maximize the ROI of an HTVS pipeline under a given computational resource constraint or strike the balance between the screening throughput and efficiency depending on the optimization scenarios. The need for screening a large set of molecules to detect potential candidates that possess the desired properties frequently arises in various science and engineering domains, although the design and operation of such screening pipelines strongly depend on expert intuitions and *ad hoc* approaches. We aimed to rectify this problem by taking a principled approach to HTVS, thereby maximizing the screening performance of a given HTVS pipeline, reducing the performance dependence on the pipeline configuration, and enabling quantitative comparison between different HTVS pipelines based on their optimal achievable performance.

We considered two scenarios for HTVS performance optimization in this study: first, maximizing the detection of potential candidate molecules that possess the desired property under a constrained computational budget and second, jointly optimizing the screening throughput and efficiency of the HTVS pipeline when there is no fixed computational budget for the screening operation. For both scenarios, we have thoroughly tested the performance of our proposed HTVS pipeline optimization framework. Comprehensive performance assessment based on synthetic HTVS pipelines as well as real lncRNA screening pipelines showed clear advantages of the proposed framework.

On the synthetic HTVS pipeline dataset, the optimized pipelines were able to remarkably maximize the screening throughput under a given resource constraint as shown in Figure 2. For example, one of the best-performing HTVS pipelines, [S₁, S₂, S₃, S₄], optimized via the proposed framework detected 99% of the promising candidates that retain the desired property

Table 5. Performance of the four-stage lncRNA HTVS pipeline [S_1, S_2, S_3, S_4] jointly optimized for throughput and efficiency is compared with that of the baseline screening approach

Approach	Potential candidates	Total cost (ms)	Effective cost	Computational savings (%)	Accuracy	Sensitivity	Specificity	F1
Proposed ($\alpha = 0.75$)	48,965	148,155,016	3,026	41.81	0.8553	0.9249	0.8126	0.8292
Proposed ($\alpha = 0.5$)	48,402	140,954,256	2,912	44.00	0.8591	0.9228	0.8200	0.8326
Proposed ($\alpha = 0.25$)	47,106	131,830,857	2,799	46.17	0.8650	0.9143	0.8348	0.8373
Baseline ($R_s = 0.75$)	39,079	115,643,459	2,959	43.10	0.8366	0.7761	0.8737	0.7801
Baseline ($R_s = 0.5$)	12,653	35,255,772	2,786	46.42	0.7170	0.2866	0.9807	0.4348
Baseline ($R_s = 0.25$)	1,402	4,963,415	3,540	31.92	0.6318	0.0330	0.9986	0.0638

at the cost of only 14% of the original cost that will be needed for screening all the initial candidates with the highest-fidelity model alone (i.e., S_4) when computational resources were constrained. When the HTVS pipelines were jointly optimized for throughput and computational efficiency, it discovered at least 94% of the true potential candidates while saving up to 86.64% of the original computational resource requirement (see Tables 1 and 2).

As shown in Figure 4, the optimized HTVS pipelines built for a real-world lncRNA screening campaign consistently demonstrated their practical efficacy in terms of maximizing the ROCI. All the optimized pipelines including CPAT, which is reasonably correlated to LncFinder, identified at least 91% of the true lncRNAs at 50% of the original computational cost. As illustrated in Table 4, the jointly optimized HTVS pipelines for throughput and computational efficiency selected 96.2% of the true lncRNAs and saved a computational resource of 41% compared with the original computational cost on average. Besides, the simulation results summarized in Table 5 demonstrate that the pipeline optimized via the proposed framework clearly outperformed the *agnostic* baseline approach.

Overall, not only does the HTVS optimization framework remove the guesswork in the operation of HTVS pipelines to maximize the throughput, enhance the screening accuracy, and minimize the computational cost, it also leads to reliable and consistent screening performance across a wide variety of HTVS pipeline structures. This is a significant benefit of the proposed framework that is of practical importance since it makes the overall screening performance robust to variations and potentially suboptimal design choices in constructing real-world HTVS pipelines. As there can be infinite different ways of building an HTVS pipeline in real scientific and engineering applications, it is important to note that our proposed optimization framework can guarantee near-optimal screening performance for any reasonable design choice regarding the HTVS pipeline configuration.

In addition, the comprehensive simulation results in diverse setups (see Figures S1–S48 in the supplemental information) have provided insights into how one can better design an HTVS pipeline when one should construct an HTVS pipeline from scratch. First, if one is allowed to analyze the interrelations of the screening stages with respect to the highest-fidelity model of interest *prior*, it might be generally effective not to include the screening stages that are less correlated to the highest-fidelity screening stage unless necessary. In other words, it might be computationally beneficial to exclude the screening stages with a lower ratio of correlation to their computational

complexity. Second, one of the effective strategies for ordering the screening stages of an HTVS pipeline is to place them in increasing order of computational complexity. Finally, care has to be taken when deciding the number of screening stages used for constructing an HTVS pipeline as its optimality is a complicated function of diverse factors including the computational complexity, order, and correlation of the screening stages in the HTVS pipeline and the positive sample ratio. However, it should be also noted that, as demonstrated, the proposed framework can alleviate the performance degradation from the suboptimal designs.

While the HTVS optimization framework presented in this paper is fairly general and may be applied to various molecular screening problems, there are interesting open research problems for extending its capabilities even further. For example, how can we design an optimal screening policy when the structure of the HTVS pipeline is not linear (i.e., sequential)? It would be interesting to investigate how one may design optimal screening policies for pipelines whose structure is a directed acyclic graph. Another interesting problem is how to construct an ideal screening pipeline if only a high-fidelity model is available and its lower-fidelity surrogates need to be designed/learned from this high-fidelity model from scratch. This problem has been recently investigated in a preliminary setting for the case of screening redox-active organic materials.⁵² Further research is needed to generalize this result and enable *joint* optimization of multiple surrogate models to optimize the performance of the overall HTVS pipeline. Finally, we can envision designing a dynamic screening policy—e.g., based on reinforcement learning—where there is no predetermined “pipeline structure” but a candidate molecule is dynamically steered to the next stage based on the screening outcome of the previous stage. In this case, instead of first learning the relations between different stages and then deriving the optimal screening policy—as we did in this study—we may learn, apply, and enhance the screening policy in a dynamic and adaptive manner, which may be better suited for specific applications.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Dr. Byung-Jun Yoon (bjyoon@ece.tamu.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- All raw RNA sequences that were used in this study are publicly available at GENCODE: https://www.genencodegenes.org/human/release_38.html.⁴⁹
- Python source code and the pre-processed RNA sequences are publicly available at Zenodo: <https://doi.org/10.5281/zenodo.8392685>.⁵³

Simulation environment

We optimized the screening policy—for both optimization problems defined in Equations 4 and 6—using the differential evolution optimizer⁵⁴ in the Scipy Python package (version 1.7.0). We performed all simulations on Ubuntu (version 20.04.2 LTS) installed on Oracle VM VirtualBox (version 6.1.22) that runs on a workstation equipped with an Intel i7-8809G CPU and 32 GB RAM.

Learning the output relations across multiple stages in an HTVS pipeline

As shown in Figure 1, the proposed optimization framework that identifies the optimal screening policy ψ^* takes a two-phase approach. In the first phase, we estimate the joint score distribution $p(y_1, y_2, \dots, y_N)$. Based on the estimated score distribution, we find the optimal screening policy ψ^* that maximizes the screening performance. To ensure good screening performance, accurate estimation of the joint score distribution $p(y_1, y_2, \dots, y_N)$ is crucial. In this study, we performed a spectral estimation under the assumption that the joint score distribution follows a multivariate Gaussian mixture model and estimated the parameters via the EM scheme.⁵¹

Finding the optimal screening policy under computational budget constraint

A formal objective is to find the optimal screening policy $\psi^* = [\lambda_1^*, \lambda_2^*, \dots, \lambda_{N-1}^*]$ that leads to the maximal number of desired candidates whose property at the highest fidelity satisfies a given condition (i.e., $f_N(x) \geq \lambda_N$) under a given computational resource constraint C . The interrelation between the property scores computed based on all stages S_1, S_2, \dots, S_N is captured by their joint score distribution $p(y_1, y_2, \dots, y_N)$. Based on the joint score distribution $p(y_1, y_2, \dots, y_N)$, we define reward function $r(\lambda)$, which is a function of N screening thresholds $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_N]$ of the stages $S_i, i = 1, 2, \dots, N$, as follows:

$$r(\lambda) = \int_{[\lambda_N, \lambda_{N-1}, \dots, \lambda_1]} \dots \int p(y_1, y_2, \dots, y_N) dy_1 dy_2 \dots dy_N. \quad (\text{Equation 3})$$

We can find the optimal screening policy $\psi^* = [\lambda_1^*, \lambda_2^*, \dots, \lambda_{N-1}^*]$ to be applied to the first $N - 1$ stages (S_i) that maximizes the reward under a given computational resource constraint C by solving the constrained optimization problem shown below:

$$\psi^* = \arg \max_{\psi \in \mathbb{R}^{N-1}} (r(\psi, \lambda_N)) \quad (\text{Equation 4})$$

$$\text{s.t. } \sum_{i=1}^N c_i |\mathbb{X}_i| \leq C,$$

where $|\mathbb{X}_i|$ is the number of molecules that passed the previous stages from S_1 to S_{i-1} . Formally, $|\mathbb{X}_i|$ is defined as:

$$|\mathbb{X}_i| = |\mathbb{X}| \int_{[\lambda_{i-1}, \lambda_{i-2}, \dots, \lambda_1]} \dots \int p_{1:i-1}(y_1, y_2, \dots, y_{i-1}) dy_1 dy_2 \dots dy_{i-1}, \quad (\text{Equation 5})$$

where $p_{1:i-1}$ denotes the marginal score distribution for y_1, \dots, y_{i-1} , which can be obtained by marginalizing $p(\cdot)$ over y_i to y_N .

Joint optimization of the screening policy for screening throughput and efficiency

In many real-world screening problems, including drug or material screening, the total computational budget for screening may not be fixed, and one may want to jointly optimize for both screening throughput as well as computational efficiency of screening. In such a scenario, we can formulate a joint optimization

problem to find the best screening policy that strikes the optimal balance between throughput and efficiency:

$$\psi^* = \arg \min_{\psi \in \mathbb{R}^{N-1}} \alpha \bar{r}(\psi, \lambda_N) + (1 - \alpha) \bar{h}(\psi, \lambda_N). \quad (\text{Equation 6})$$

The weight parameter $\alpha \in [0, 1]$ determines the relative importance between the relative reward function $\bar{r}(\psi, \lambda_N)$ and the normalized total cost function $\bar{h}(\psi, \lambda_N)$ defined as follows:

$$\bar{r}(\psi, \lambda_N) = \frac{r([-\infty, \lambda_N]) - r(\psi, \lambda_N)}{r([-\infty, \lambda_N])} \quad (\text{Equation 7})$$

$$= \frac{\int_{\lambda_N}^{\infty} p_N(y_N) dy_N - r(\psi, \lambda_N)}{\int_{\lambda_N}^{\infty} p_N(y_N) dy_N}, \quad (\text{Equation 8})$$

$$\bar{h}(\psi, \lambda_N) = \frac{1}{N|\mathbb{X}| \max_i c_i} \sum_{i=1}^N c_i |\mathbb{X}_i|. \quad (\text{Equation 9})$$

Note that p_N is the marginal score distribution for y_N , which is obtained by marginalizing $p(\cdot)$ over y_1 to y_{N-1} .

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at <https://doi.org/10.1016/j.patter.2023.100875>.

ACKNOWLEDGMENTS

This work was supported in part by the Department of Energy (DOE) award DE-SC0019303, the National Science Foundation (NSF) award 1835690, and the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (RS-2023-00279241). Portions of this research were conducted with the advanced computing resources provided by Texas A&M High-Performance Research Computing (HPRC). S.J. acknowledges ASCR Surrogates Benchmark Initiative project (DE-SC0021352), ECP CANDLE, and ExaWorks (Brookhaven National Laboratory under contract DE-SC0012704).

AUTHOR CONTRIBUTIONS

B.-J.Y., S.J., and F.J.A. initiated the project. H.-M.W. and B.-J.Y. proposed the method and designed the experiments. H.-M.W. implemented the algorithms, performed the simulations, analyzed the results, and wrote the initial manuscript. H.-M.W., X.Q., L.T., S.J., F.J.A., E.R.D., and B.-J.Y. discussed the results and edited the manuscript. H.-M.W. and B.-J.Y. revised the manuscript.

DECLARATION OF INTERESTS

B.-J.Y. is a member of the Advisory Board of *Patterns*.

Received: November 28, 2022

Revised: December 28, 2022

Accepted: October 13, 2023

Published: November 3, 2023

REFERENCES

1. Saadi, A.A., Alfe, D., Babuji, Y., Bhati, A., Blaiszik, B., Brace, A., Brettin, T., Chard, K., Chard, R., Clyde, A., et al. (2021). IMPECCABLE: integrated modeling pipeline for COVID cure by assessing better leads. In 50th International Conference on Parallel Processing, pp. 1–12.
2. Roy, B., Dhillon, J., Habib, N., and Pugazhandhi, B. (2021). Global variants of covid-19: Current understanding. *J. Biomed. Sci.* 8, 8–11.
3. Bohacek, R.S., McMartin, C., and Guida, W.C. (1996). The art and practice of structure-based drug design: a molecular modeling perspective. *Med. Res. Rev.* 16, 3–50.

4. Rieber, N., Knapp, B., Eils, R., and Kaderali, L. (2009). Rnaither, an automated pipeline for the statistical analysis of high-throughput rna screens. *Bioinformatics* 25, 678–679.
5. Studer, M.H., DeMartini, J.D., Brethauer, S., McKenzie, H.L., and Wyman, C.E. (2010). Engineering of a high-throughput screening system to identify cellulosic biomass, pretreatments, and enzyme formulations that enhance sugar release. *Biotechnol. Bioeng.* 105, 231–238.
6. Hartmann, A., Czuderna, T., Hoffmann, R., Stein, N., and Schreiber, F. (2011). Htpheo: an image analysis pipeline for high-throughput plant phenotyping. *BMC Bioinf.* 12, 148–149.
7. Sikorski, K., Mehta, A., Inngjerdigen, M., Thakor, F., Kling, S., Kalina, T., Nyman, T.A., Stensland, M.E., Zhou, W., de Souza, G.A., et al. (2018). A high-throughput pipeline for validation of antibodies. *Nat. Methods* 15, 909–912.
8. Clyde, A., Galanie, S., Kneller, D.W., Ma, H., Babuji, Y., Blaiszik, B., Brace, A., Brettin, T., Chard, K., Chard, R., et al. (2022). High-throughput virtual screening and validation of a sars-cov-2 main protease noncovalent inhibitor. *J. Chem. Inf. Model.* 62, 116–128.
9. Martin, R.L., Simon, C.M., Smit, B., and Haranczyk, M. (2014). In silico design of porous polymer networks: high-throughput screening for methane storage materials. *J. Am. Chem. Soc.* 136, 5006–5022.
10. Cheng, L., Assary, R.S., Qu, X., Jain, A., Ong, S.P., Rajput, N.N., Persson, K., and Curtiss, L.A. (2015). Accelerating electrolyte discovery for energy storage with high-throughput screening. *J. Phys. Chem. Lett.* 6, 283–291.
11. Chen, J.J.F., and Visco, D.P., Jr. (2017). Developing an in silico pipeline for faster drug candidate discovery: Virtual high throughput screening with the signature molecular descriptor using support vector machine models. *Eur. J. Med. Chem.* 140, 31–41.
12. Filer, D.L., Kothiya, P., Setzer, R.W., Judson, R.S., and Martin, M.T. (2017). tcpl: the toxcast pipeline for high-throughput screening data. *Bioinformatics* 33, 618–620.
13. Rebbeck, R.T., Singh, D.P., Janicek, K.A., Bers, D.M., Thomas, D.D., Launikonis, B.S., and Cornea, R.L. (2020). Ryr1-targeted drug discovery pipeline integrating fret-based high-throughput screening and human myofiber dynamic ca²⁺ assays. *Sci. Rep.* 10, 1791–1813.
14. Tran, A., Wildey, T., and McCann, S. (2020). smf-bo-2cogp: A sequential multi-fidelity constrained bayesian optimization framework for design applications. *J. Comput. Inf. Sci. Eng.* 20.
15. Yan, Q., Yu, J., Suram, S.K., Zhou, L., Shinde, A., Newhouse, P.F., Chen, W., Li, G., Persson, K.A., Gregoire, J.M., and Neaton, J.B. (2017). Solar fuels photoanode materials discovery by integrating high-throughput theory and experiment. *Proc. Natl. Acad. Sci. USA* 114, 3040–3043.
16. Zhang, B., Zhang, X., Yu, J., Wang, Y., Wu, K., and Lee, M.-H. (2020). First-principles high-throughput screening pipeline for nonlinear optical materials: Application to borates. *Chem. Mater.* 32, 6772–6779.
17. Chen, C., Wang, Y., Xia, Y., Wu, B., Tang, D., Wu, K., Wenrong, Z., Yu, L., and Mei, L. (1995). New development of nonlinear optical crystals for the ultraviolet region with molecular engineering approach. *J. Appl. Phys.* 77, 2268–2272.
18. Shi, G., Wang, Y., Zhang, F., Zhang, B., Yang, Z., Hou, X., Pan, S., and Poepelmeier, K.R. (2017). Finding the next deep-ultraviolet nonlinear optical material: Nh4b4o6f. *J. Am. Chem. Soc.* 139, 10645–10648.
19. Zhang, B., Shi, G., Yang, Z., Zhang, F., and Pan, S. (2017). Fluorooxoborates: beryllium-free deep-ultraviolet nonlinear optical materials without layered growth. *Angew. Chem., Int. Ed. Engl.* 56, 3916–3919.
20. Luo, M., Liang, F., Song, Y., Zhao, D., Xu, F., Ye, N., and Lin, Z. (2018). M2b10o14f6 (m = ca, sr): Two noncentrosymmetric alkaline earth fluorooxoborates as promising next-generation deep-ultraviolet nonlinear optical materials. *J. Am. Chem. Soc.* 140, 3884–3887.
21. Mutailipu, M., Zhang, M., Zhang, B., Wang, L., Yang, Z., Zhou, X., and Pan, S. (2018). Sr5o7f3 functionalized with [b5o9f3] 6- chromophores: Accelerating the rational design of deep-ultraviolet nonlinear optical materials. *Angew. Chem.* 57, 6095–6099.
22. Wang, Y., Zhang, B., Yang, Z., and Pan, S. (2018). Cation-tuned synthesis of fluorooxoborates: Towards optimal deep-ultraviolet nonlinear optical materials. *Angew. Chem.* 57, 2150–2154.
23. Zhang, Z., Wang, Y., Zhang, B., Yang, Z., and Pan, S. (2018). Cab5o7f3: A beryllium-free alkaline-earth fluorooxoborate exhibiting excellent nonlinear optical performances. *Inorg. Chem.* 57, 4820–4823.
24. Wilmer, C.E., Leaf, M., Lee, C.Y., Farha, O.K., Hauser, B.G., Hupp, J.T., and Snurr, R.Q. (2011). Large-scale screening of hypothetical metal-organic frameworks. *Nat. Chem.* 4, 83–89.
25. Gupta, S., Parihar, D., Shah, M., Yadav, S., Managori, H., Bhowmick, S., Patil, P.C., Alissa, S.A., Wabaidur, S.M., and Islam, M.A. (2020). Computational screening of promising beta-secretase 1 inhibitors through multi-step molecular docking and molecular dynamics simulations-pharmacoinformatics approach. *J. Mol. Struct.* 1205, 127660.
26. Kim, D.Y., Ha, M., and Kim, K.S. (2021). A universal screening strategy for the accelerated design of superior oxygen evolution/reduction electrocatalysts. *J. Mater. Chem. A Mater.* 9, 3511–3519.
27. Liu, T., Kim, K.C., Kaviani, R., Jang, S.S., and Lee, S.W. (2015). High-density lithium-ion energy storage utilizing the surface redox reactions in folded graphene films. *Chem. Mater.* 27, 3291–3298.
28. Kim, K.C., Liu, T., Lee, S.W., and Jang, S.S. (2016). First-principles density functional theory modeling of li binding: thermodynamics and redox properties of quinone derivatives for lithium-ion batteries. *J. Am. Chem. Soc.* 138, 2374–2382.
29. Kim, S., Kim, K.C., Lee, S.W., and Jang, S.S. (2016). Thermodynamic and redox properties of graphene oxides for lithium-ion battery applications: a first principles density functional theory modeling approach. *Phys. Chem. Chem. Phys.* 18, 20600–20606.
30. Liu, T., Kim, K.C., Lee, B., Chen, Z., Noda, S., Jang, S.S., and Lee, S.W. (2017). Self-polymerized dopamine as an organic cathode for li-and n-ion batteries. *Energy Environ. Sci.* 10, 205–215.
31. Park, J.H., Liu, T., Kim, K.C., Lee, S.W., and Jang, S.S. (2017). Systematic molecular design of ketone derivatives of aromatic molecules for lithium-ion batteries: First-principles dft modeling. *ChemSusChem* 10, 1584–1591.
32. Kang, J., Kim, K.C., and Jang, S.S. (2018). Density functional theory modeling-assisted investigation of thermodynamics and redox properties of boron-doped corannulenes for cathodes in lithium-ion batteries. *J. Phys. Chem. C* 122, 10675–10681.
33. Sood, P., Kim, K.C., and Jang, S.S. (2018). Electrochemical and electronic properties of nitrogen doped fullerene and its derivatives for lithium-ion battery applications. *J. Energy Chem.* 27, 528–534.
34. Sood, P., Kim, K.C., and Jang, S.S. (2018). Electrochemical properties of boron-doped fullerene derivatives for lithium-ion battery applications. *ChemPhysChem* 19, 753–758.
35. Zhu, Y., Kim, K.C., and Jang, S.S. (2018). Boron-doped coronenes with high redox potential for organic positive electrodes in lithium-ion batteries: a first-principles density functional theory modeling study. *J. Mater. Chem. A Mater.* 6, 10111–10120.
36. Ng, S.-Y., Lin, L., Soh, B.S., and Stanton, L.W. (2013). Long noncoding rnas in development and disease of the central nervous system. *Trends Genet.* 29, 461–468.
37. Tan, L., Yu, J.-T., Hu, N., and Tan, L. (2013). Non-coding rnas in alzheimer’s disease. *Mol. Neurobiol.* 47, 382–393.
38. Luo, Q., and Chen, Y. (2016). Long noncoding rnas and alzheimer’s disease. *Clin. Interv. Aging* 11, 867–872.
39. Congrains, A., Kamide, K., Oguro, R., Yasuda, O., Miyata, K., Yamamoto, E., Kawai, T., Kusunoki, H., Yamamoto, H., Takeya, Y., et al. (2012). Genetic variants at the 9p21 locus contribute to atherosclerosis through modulation of anril and cdkn2a/b. *Atherosclerosis* 220, 449–455.
40. Xue, Z., Hennelly, S., Doyle, B., Gulati, A.A., Novikova, I.V., Sanbonmatsu, K.Y., and Boyer, L.A. (2016). A g-rich motif in the Incrna braveheart interacts with a zinc-finger transcription factor to specify the cardiovascular lineage. *Mol. Cell* 64, 37–50.

41. Yang, G., Lu, X., and Yuan, L. (2014). Lncrna: a link between rna and cancer. *Biochim. Biophys. Acta* 1839, 1097–1109.
42. Shi, X., Sun, M., Liu, H., Yao, Y., Kong, R., Chen, F., and Song, Y. (2015). A critical role for the long non-coding rna gas5 in proliferation and apoptosis in non-small-cell lung cancer. *Mol. Carcinog.* 54, E1–E12.
43. Peng, W.-X., Koirala, P., and Mo, Y.-Y. (2017). Lncrna-mediated regulation of cell signaling in cancer. *Oncogene* 36, 5661–5667.
44. Carlevaro-Fita, J., Lanzós, A., Feuerbach, L., Hong, C., Mas-Ponte, D., Pedersen, J.S., and Johnson, R. (2020). Cancer Lncrna census reveals evidence for deep functional conservation of long noncoding rnas in tumorigenesis. *Commun. Biol.* 3, 1–16.
45. Wang, L., Park, H.J., Dasari, S., Wang, S., Kocher, J.-P., and Li, W. (2013). Cpat: Coding-potential assessment tool using an alignment-free logistic regression model. *Nucleic Acids Res.* 41, e74.
46. Li, A., Zhang, J., and Zhou, Z. (2014). Plek: a tool for predicting long non-coding rnas and messenger rnas based on an improved k-mer scheme. *BMC Bioinf.* 15, 311–410.
47. Kang, Y.-J., Yang, D.-C., Kong, L., Hou, M., Meng, Y.-Q., Wei, L., and Gao, G. (2017). Cpc2: a fast and accurate coding potential calculator based on sequence intrinsic features. *Nucleic Acids Res.* 45, W12–W16.
48. Han, S., Liang, Y., Ma, Q., Xu, Y., Zhang, Y., Du, W., Wang, C., and Li, Y. (2019). Lncfinder: an integrated platform for long non-coding rna identification utilizing sequence intrinsic composition, structural information and physicochemical property. *Briefings Bioinf.* 20, 2009–2027.
49. Frankish, A., Diekhans, M., Jungreis, I., Lagarde, J., Loveland, J.E., Mudge, J.M., Sisu, C., Wright, J.C., Armstrong, J., Barnes, I., et al. (2021). Gencode 2021. *Nucleic Acids Res.* 49, D916–D923.
50. Li, W., and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* 22, 1658–1659.
51. Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Stat. Soc. B* 39, 1–22.
52. Woo, H.-M., Allam, O., Chen, J., Jang, S.S., and Yoon, B.-J. (2022). Optimal high-throughput virtual screening pipeline for efficient selection of redox-active organic materials. *iScience* 26, 105735.
53. Woo, H.-M., Qian, X., Jha, S., Alexander, F., Dougherty, E., and Yoon, B.-J. (2023). Alexpecial/OCC: Patterns (Zenodo). <https://doi.org/10.5281/zenodo.8392685>.
54. Storn, R., and Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* 11, 341–359.

Patterns, Volume 4

Supplemental information

**Optimal decision-making in high-throughput
virtual screening pipelines**

Hyun-Myung Woo, Xiaoning Qian, Li Tan, Shantenu Jha, Francis J. Alexander, Edward R. Dougherty, and Byung-Jun Yoon

Performance evaluations of the high-throughput virtual screening pipelines with various covariance matrices

Case 1: Uniform correlation between the surrogate models

$$p(y_1, y_2, y_3, y_4) \sim \mathcal{N} \left(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.2 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.2 & 0.2 & 1 & 0.2 \\ 0.2 & 0.2 & 0.2 & 1 \end{bmatrix} \right) \quad (1)$$

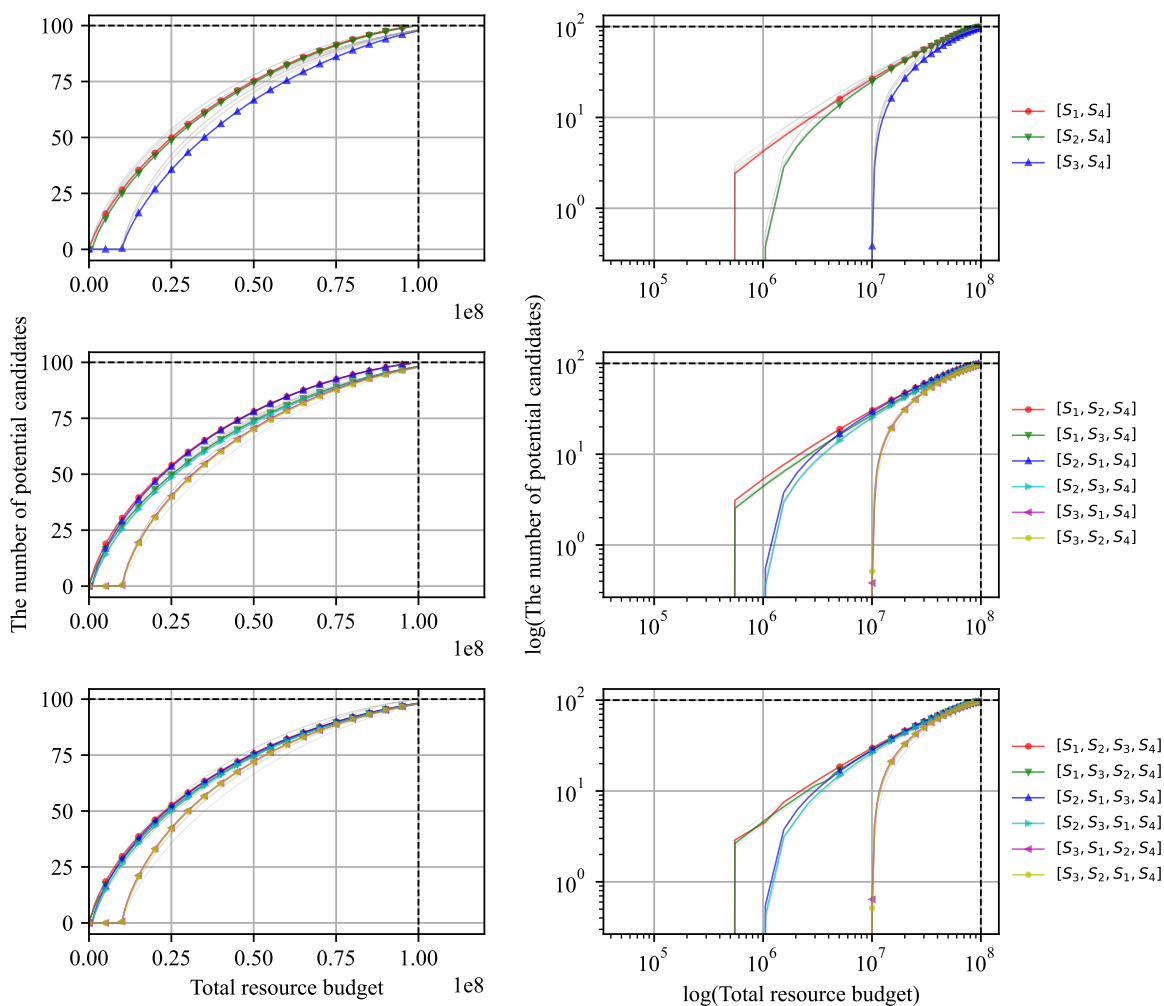


Figure S1: Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.

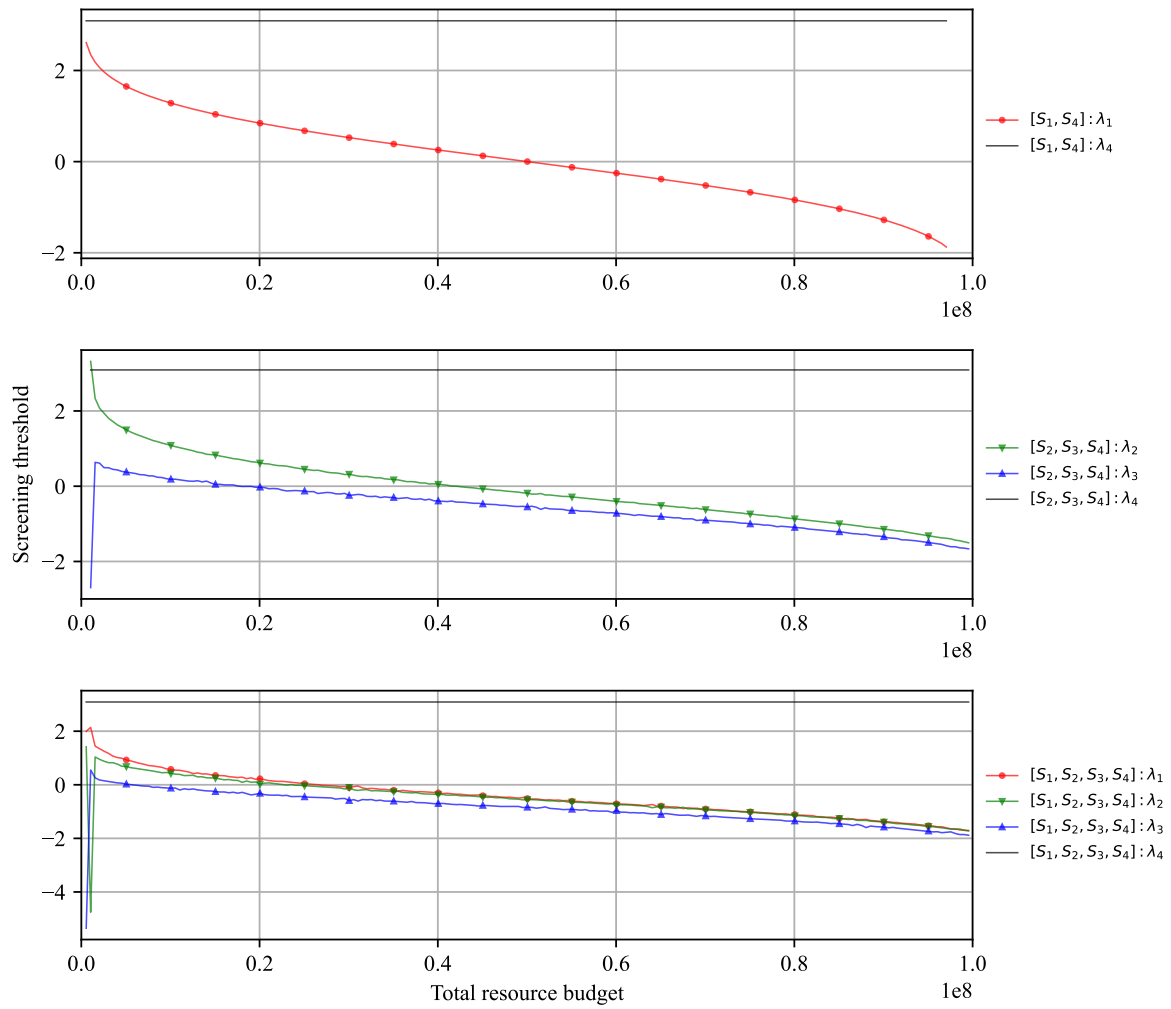


Figure S2: Screening thresholds of the optimized pipelines.

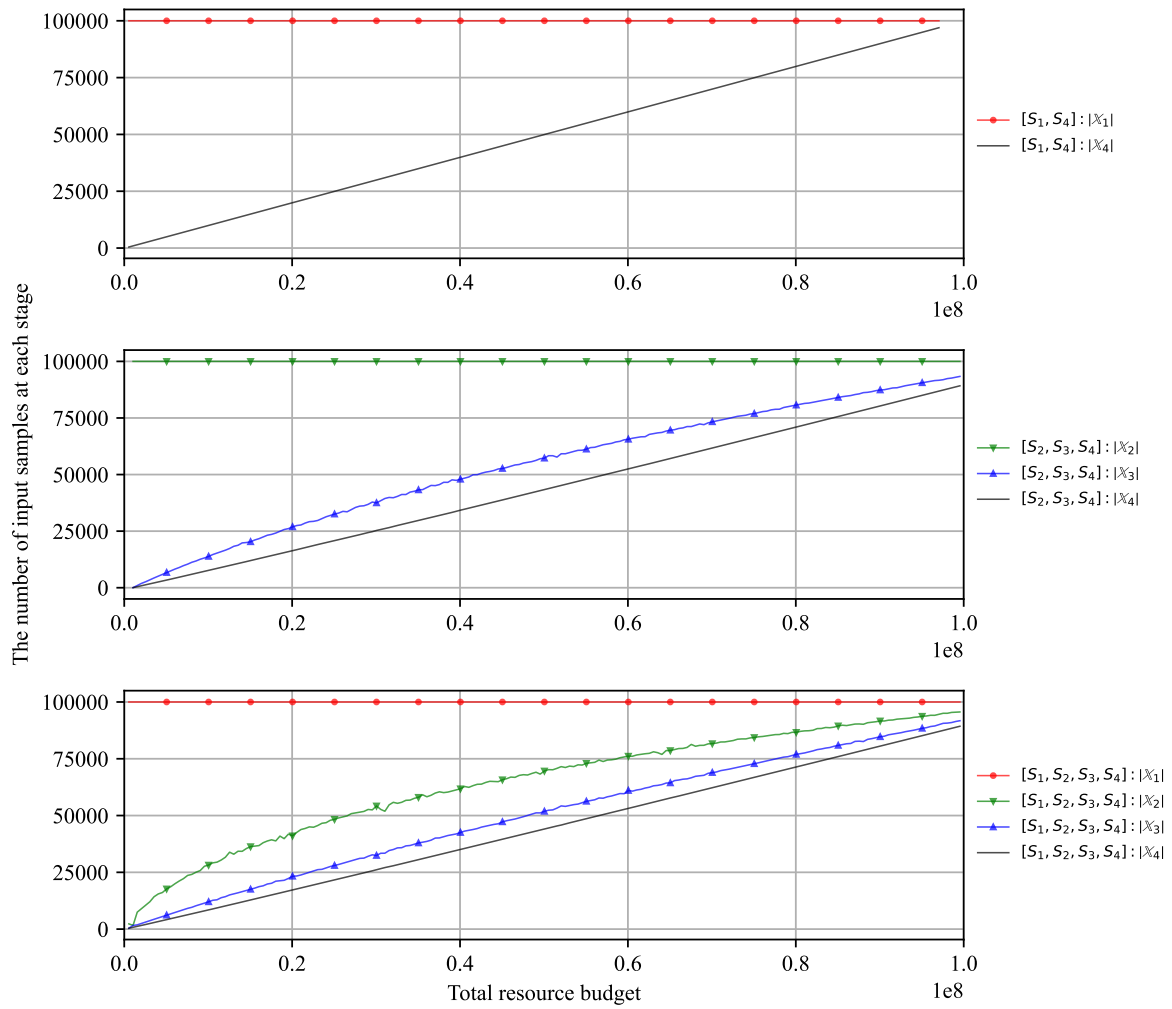


Figure S3: The number of input samples at each stage.

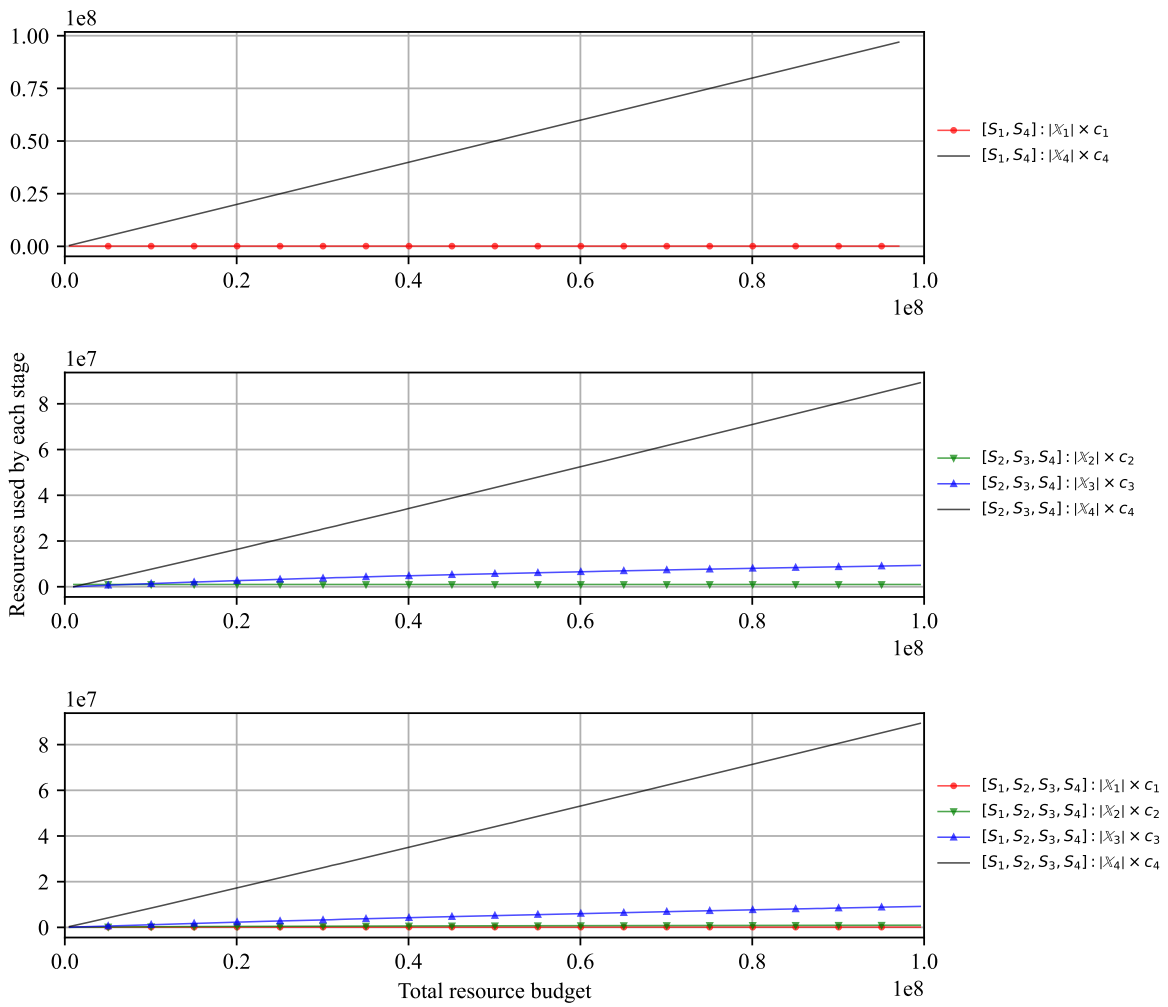


Figure S4: Resources used by each stage.

$$p(y_1, y_2, y_3, y_4) \sim \mathcal{N} \left(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1 \end{bmatrix} \right) \quad (2)$$

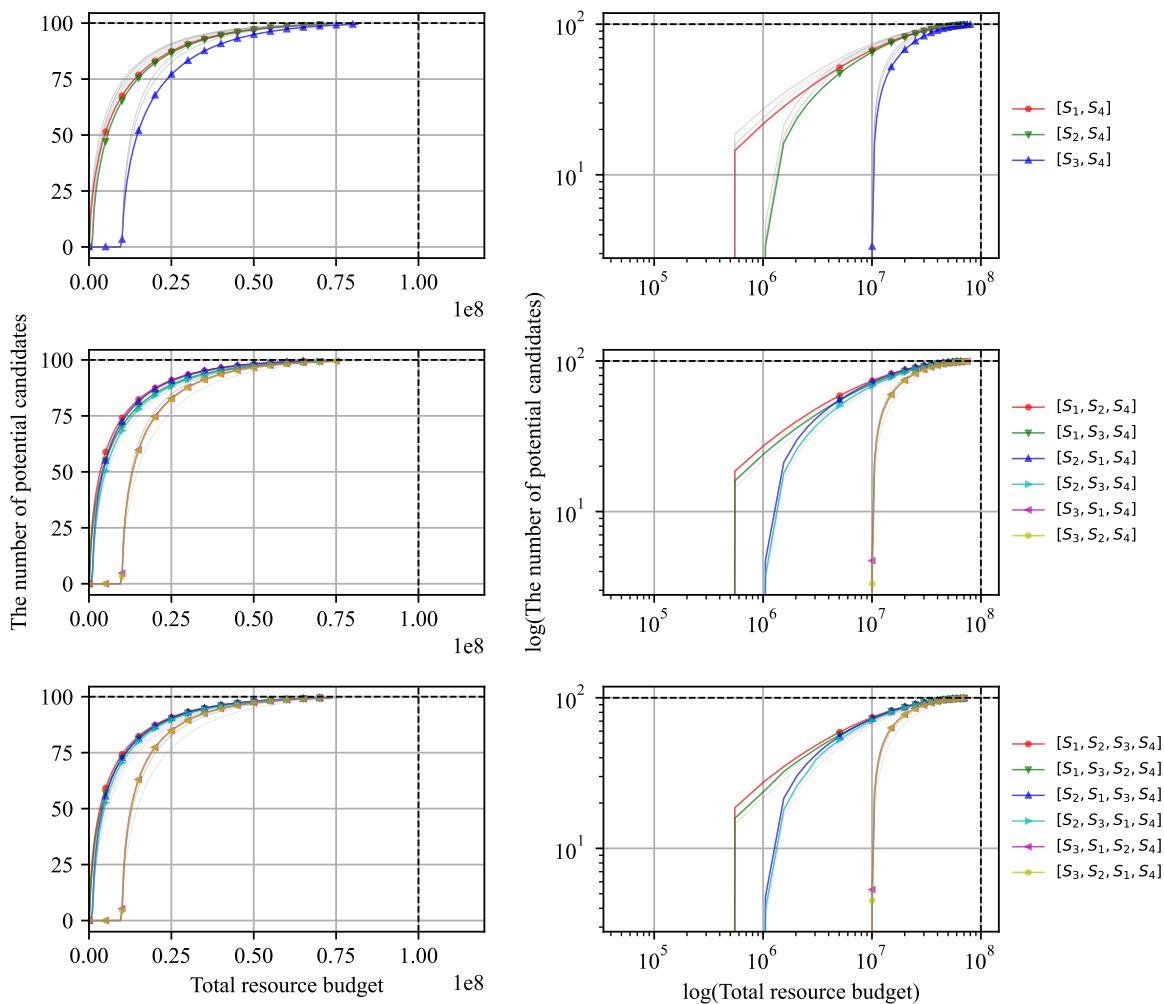


Figure S5: **Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.**

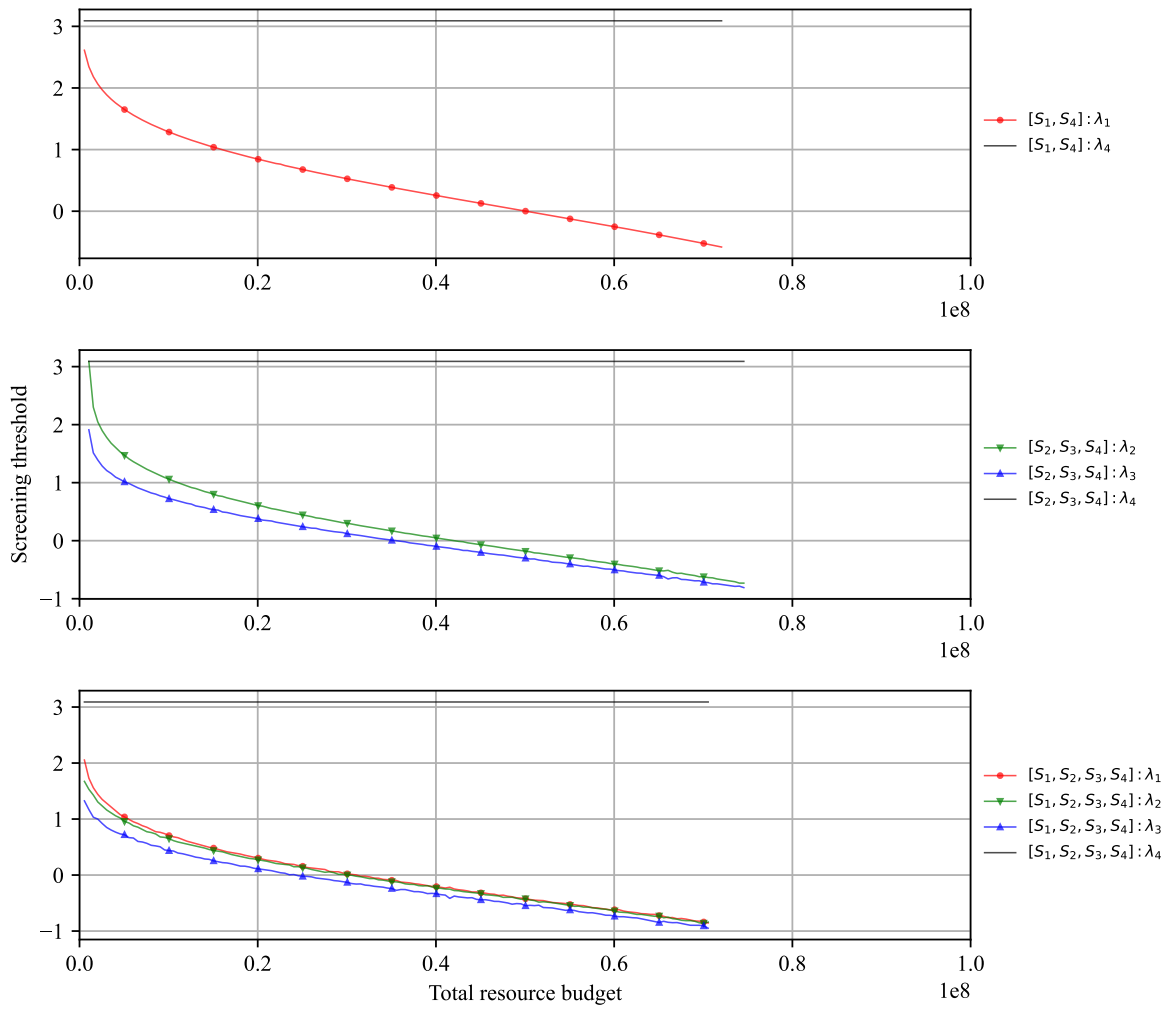


Figure S6: Screening thresholds of the optimized pipelines.

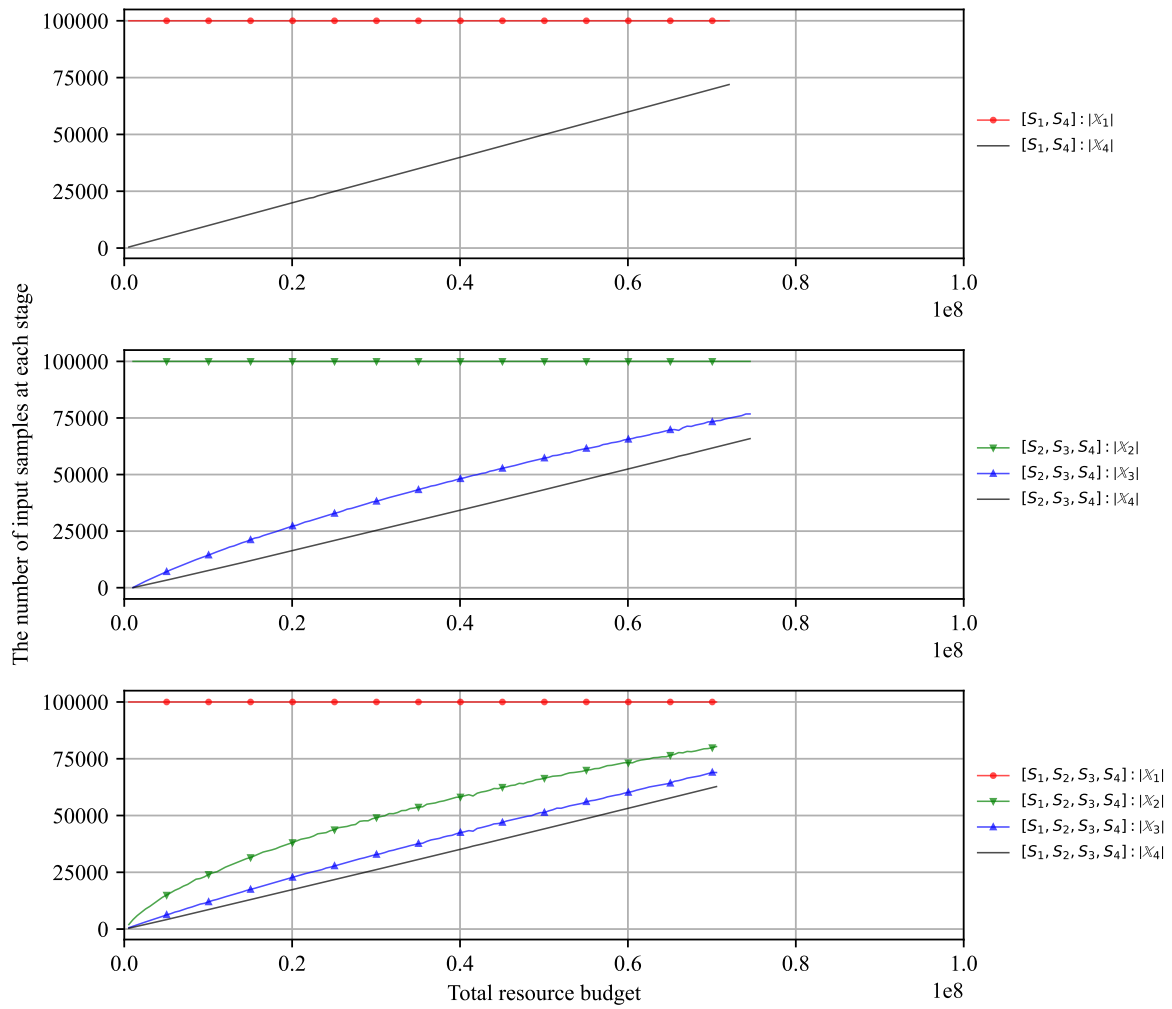


Figure S7: The number of input samples at each stage.

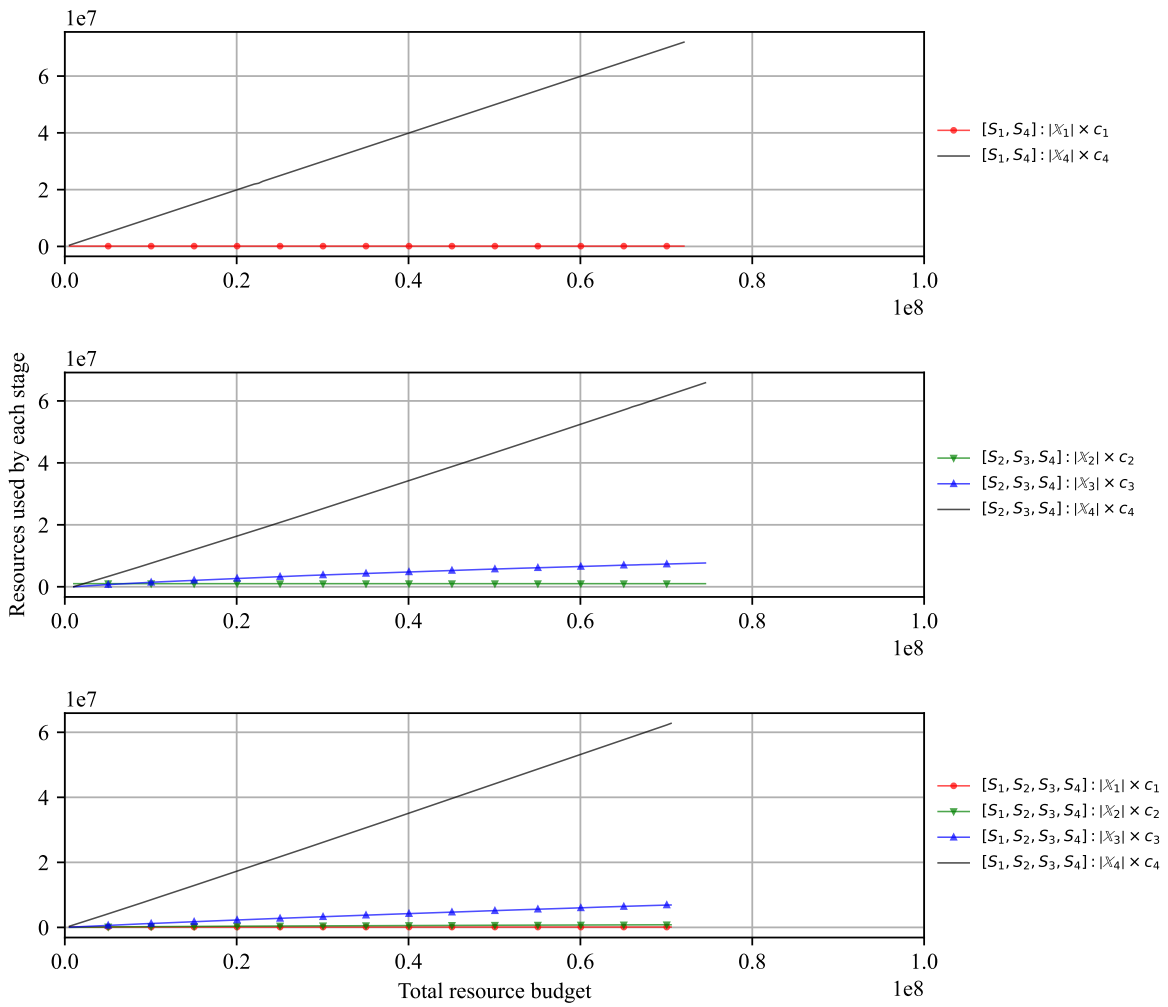


Figure S8: Resources used by each stage.

$$p(y_1, y_2, y_3, y_4) \sim \mathcal{N} \left(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.8 & 0.8 & 0.8 \\ 0.8 & 1 & 0.8 & 0.8 \\ 0.8 & 0.8 & 1 & 0.8 \\ 0.8 & 0.8 & 0.8 & 1 \end{bmatrix} \right) \quad (3)$$

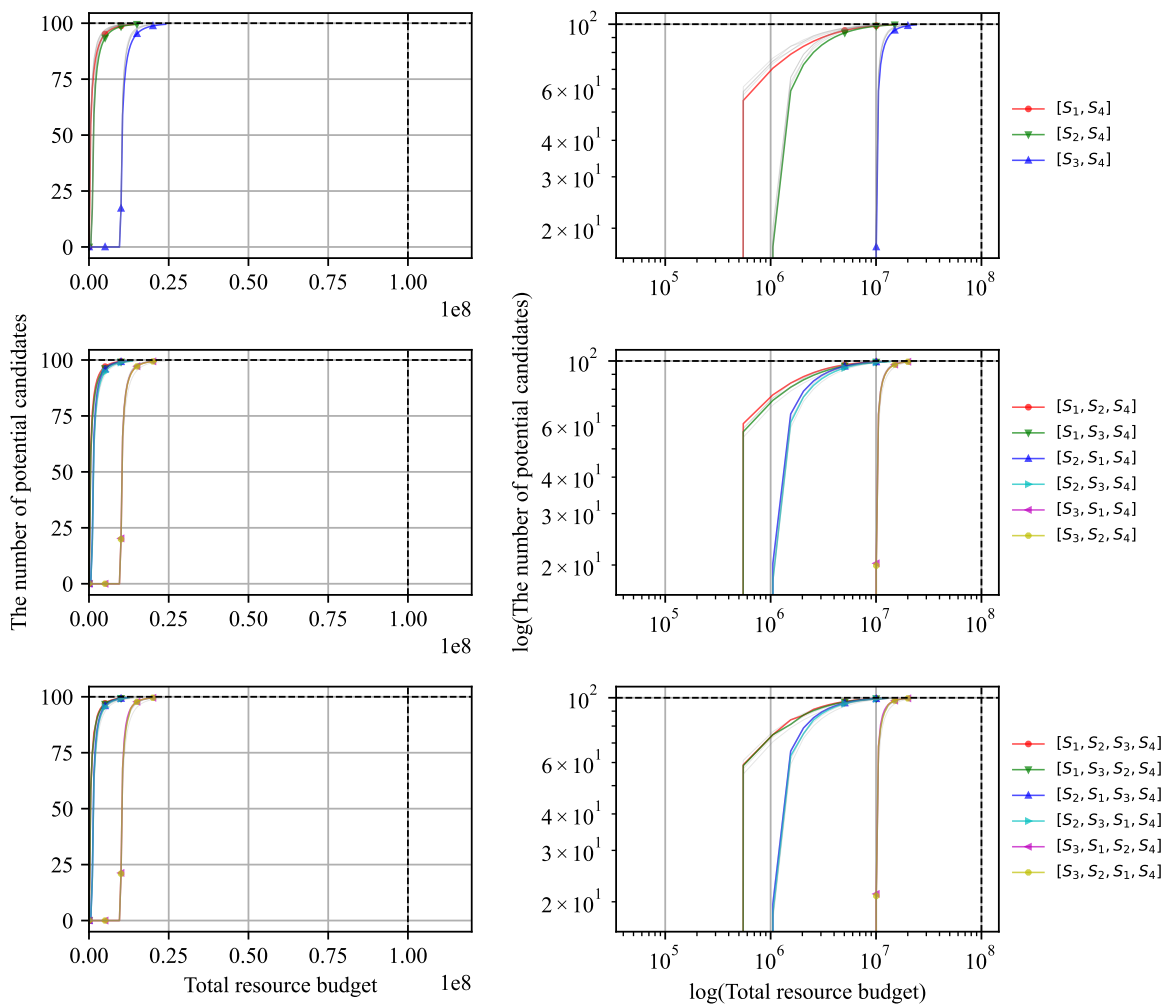


Figure S9: **Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.**

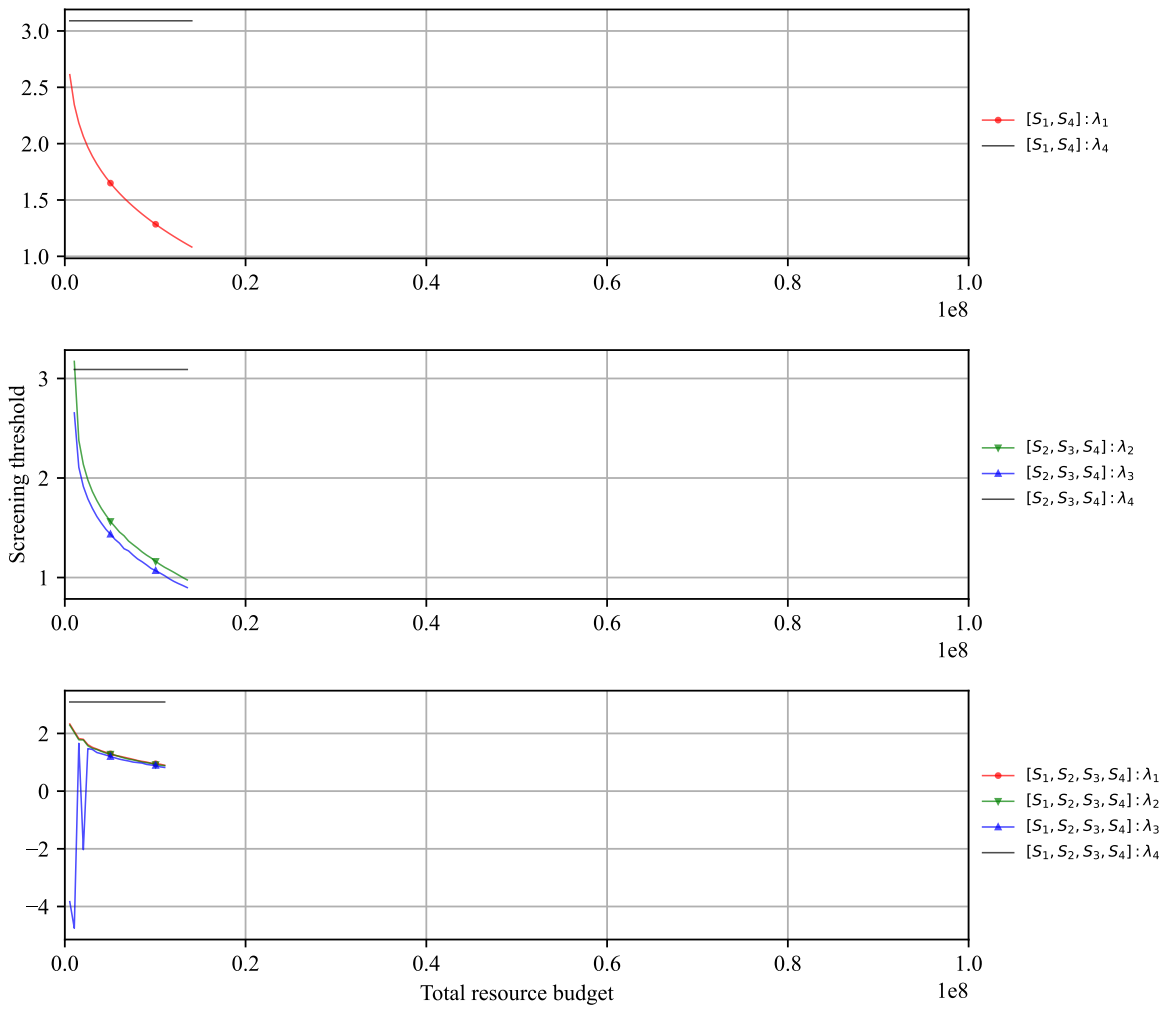


Figure S10: Screening thresholds of the optimized pipelines.

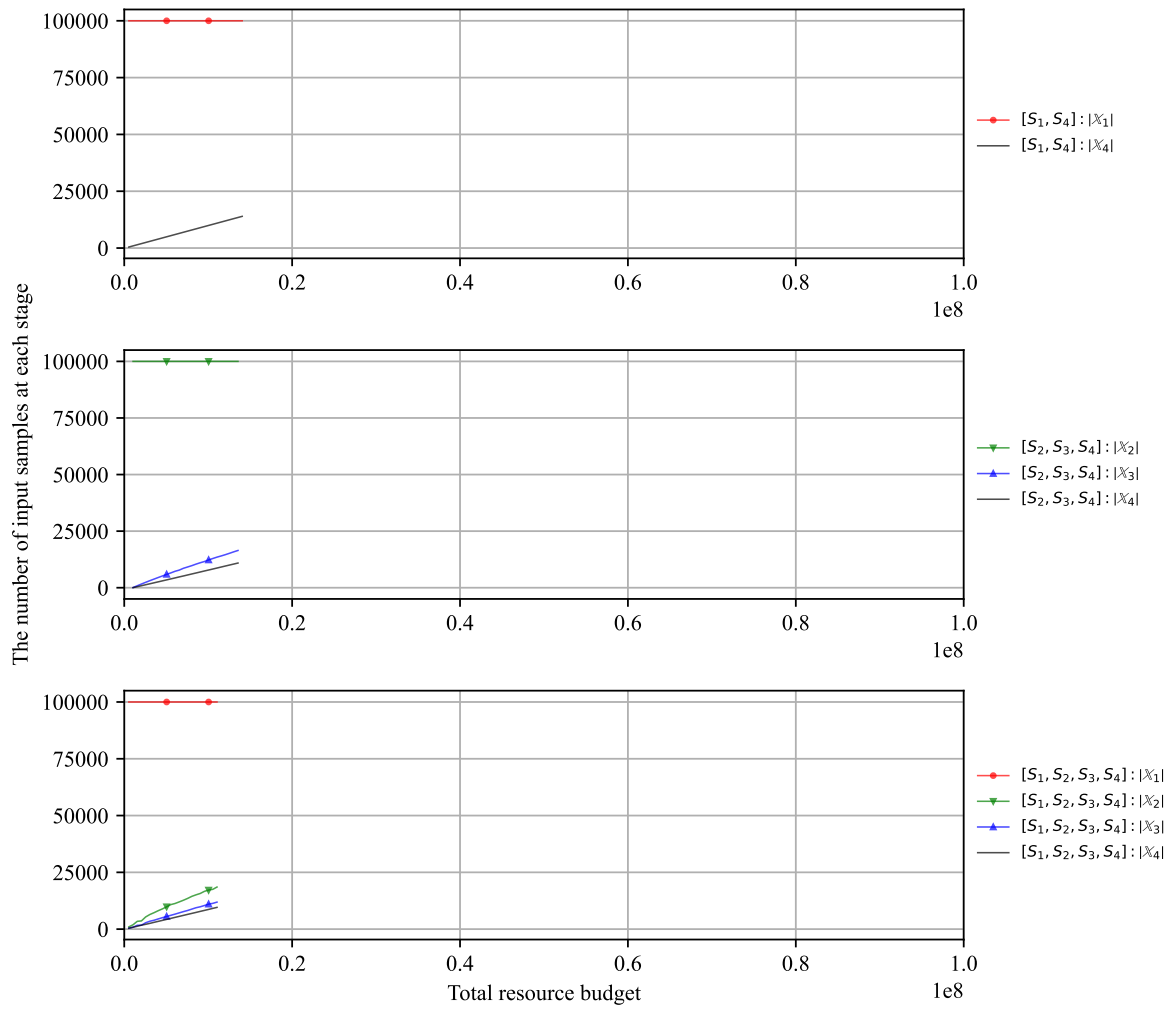


Figure S11: The number of input samples at each stage.

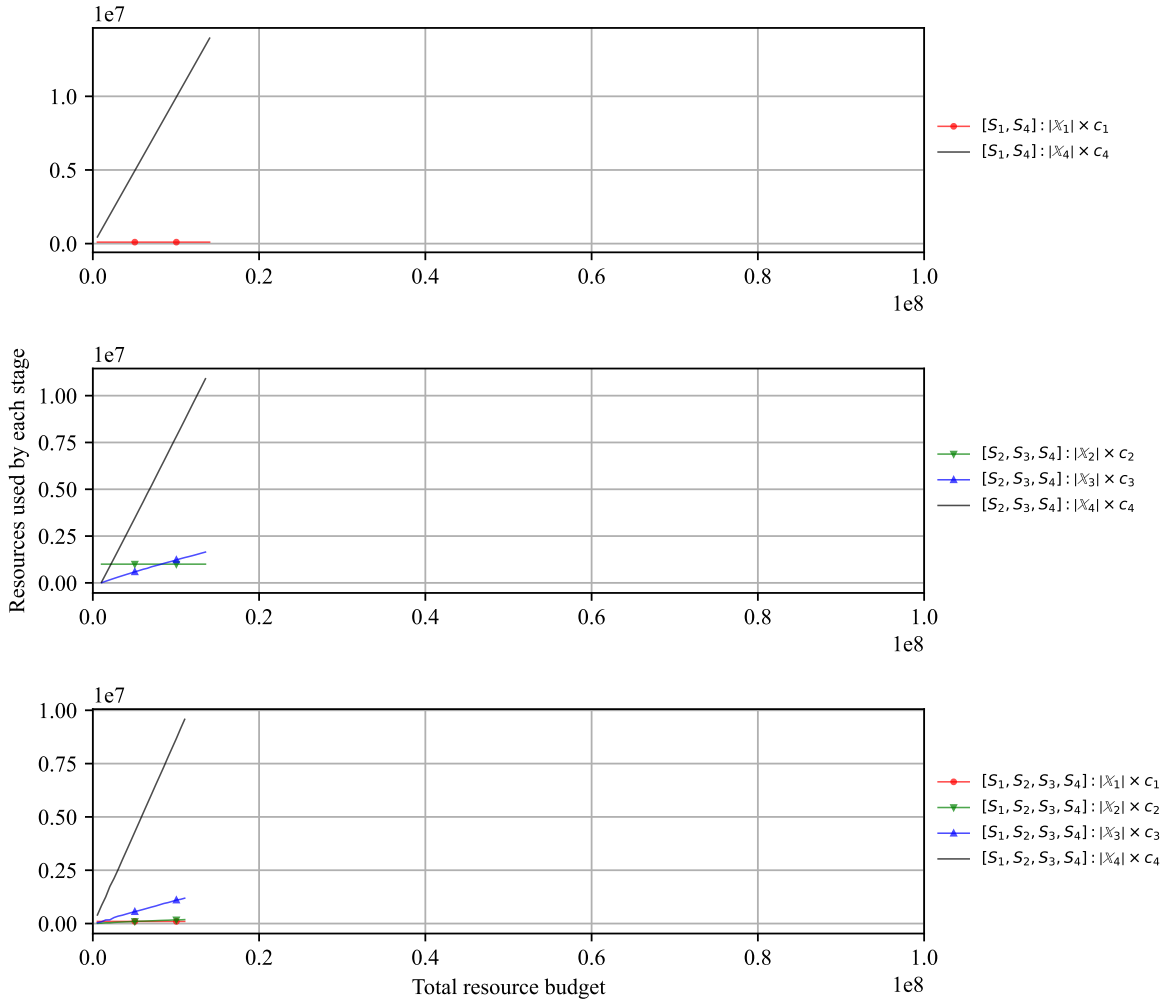


Figure S12: Resources used by each stage.

Case 2: Non-uniform correlation between the surrogate models

$$p(y_1, y_2, y_3, y_4) \sim \mathcal{N} \left(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.2 & 0.2 & 0.8 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.2 & 0.2 & 1 & 0.2 \\ 0.8 & 0.2 & 0.2 & 1 \end{bmatrix} \right) \quad (4)$$

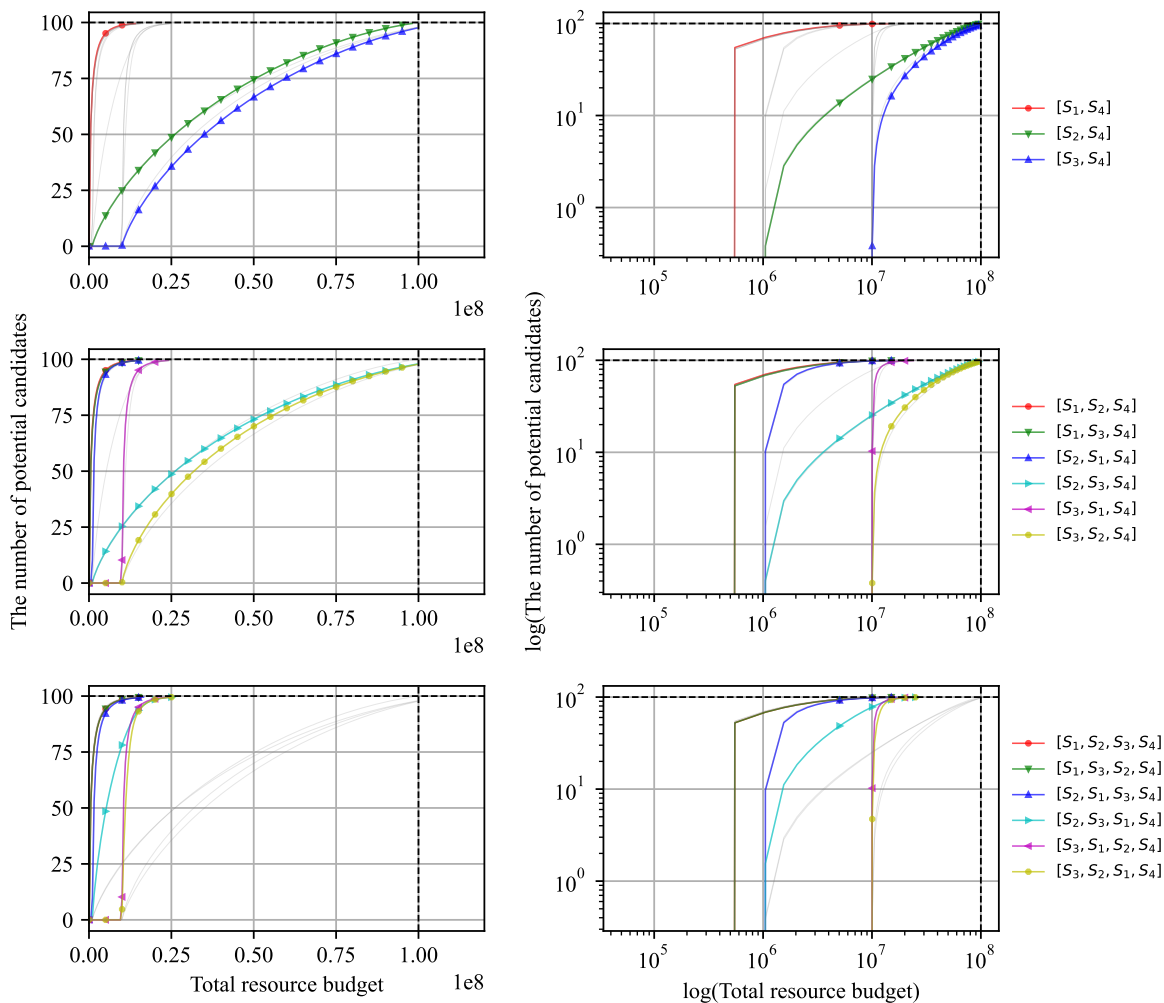


Figure S13: Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.

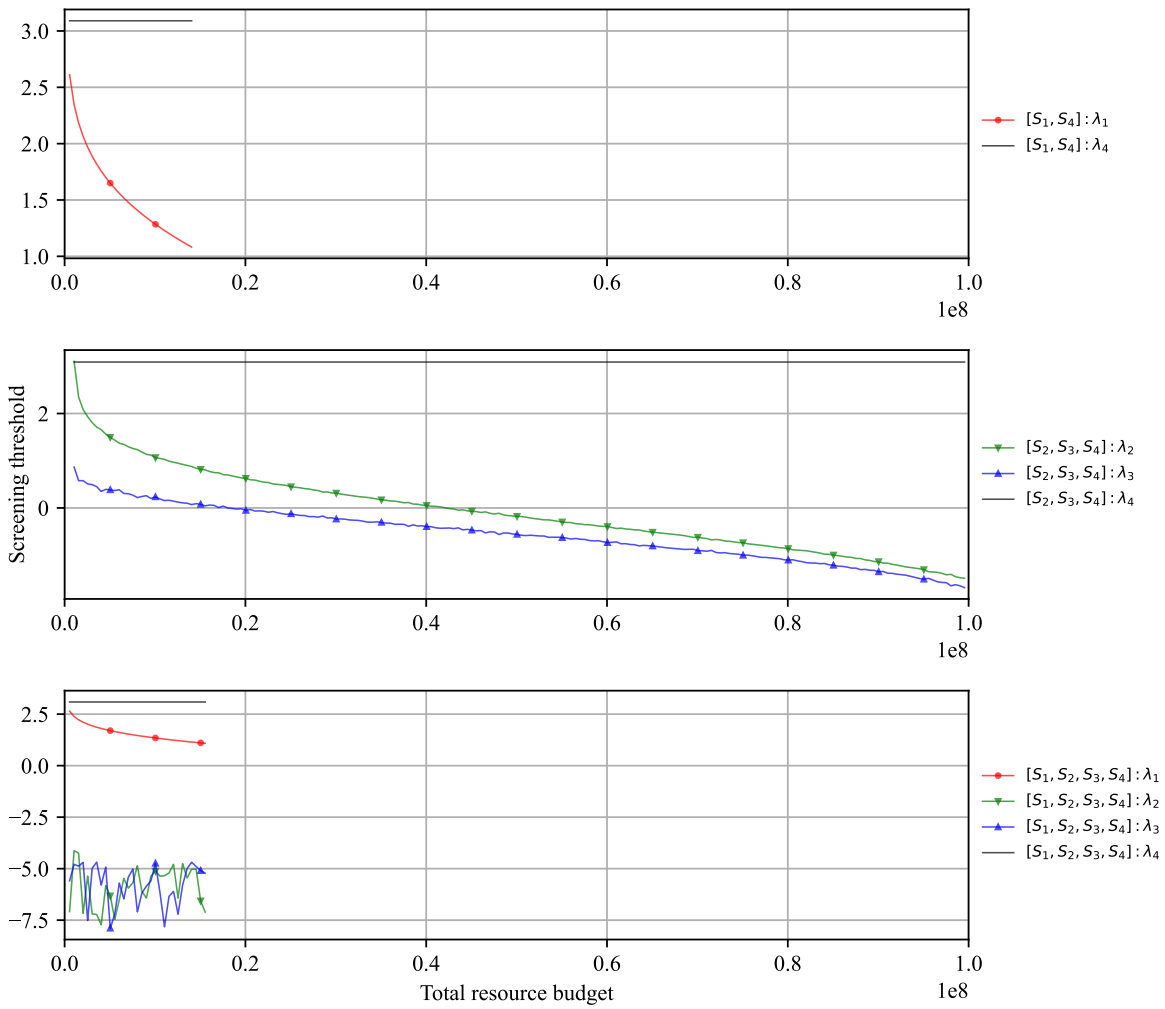


Figure S14: Screening thresholds of the optimized pipelines.

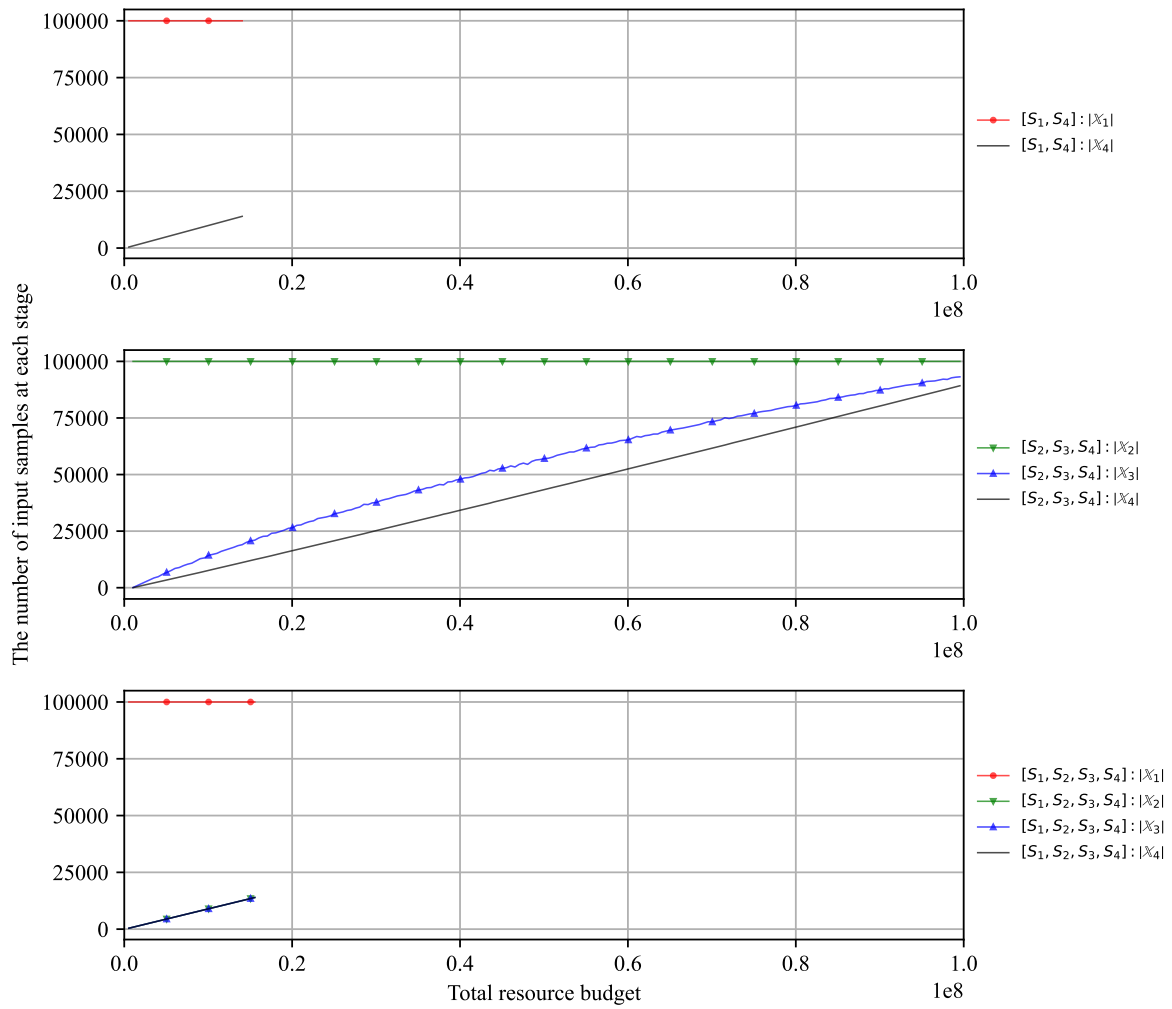


Figure S15: The number of input samples at each stage.

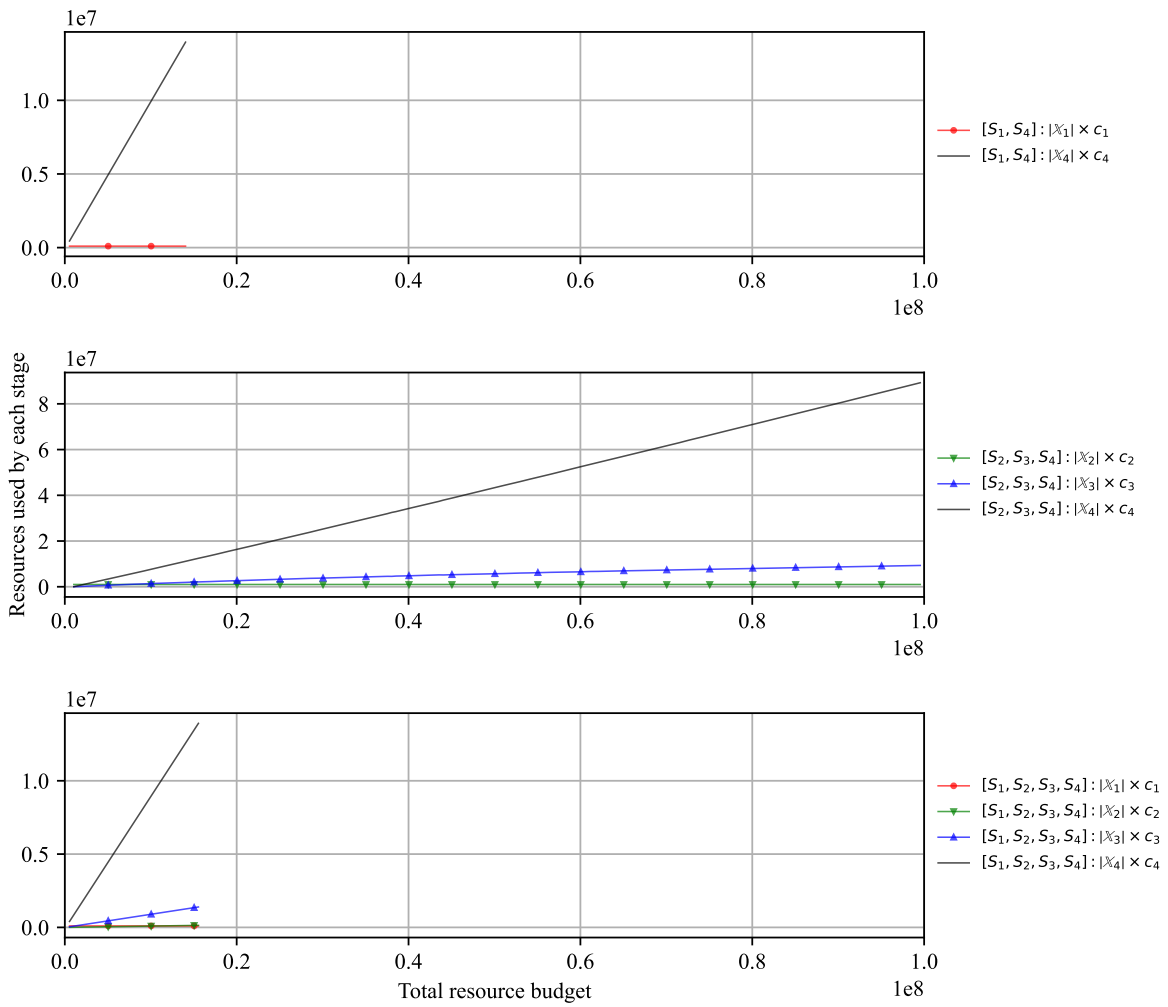


Figure S16: Resources used by each stage.

$$p(y_1, y_2, y_3, y_4) \sim \mathcal{N} \left(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.2 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 & 0.8 \\ 0.2 & 0.2 & 1 & 0.2 \\ 0.2 & 0.8 & 0.2 & 1 \end{bmatrix} \right) \quad (5)$$

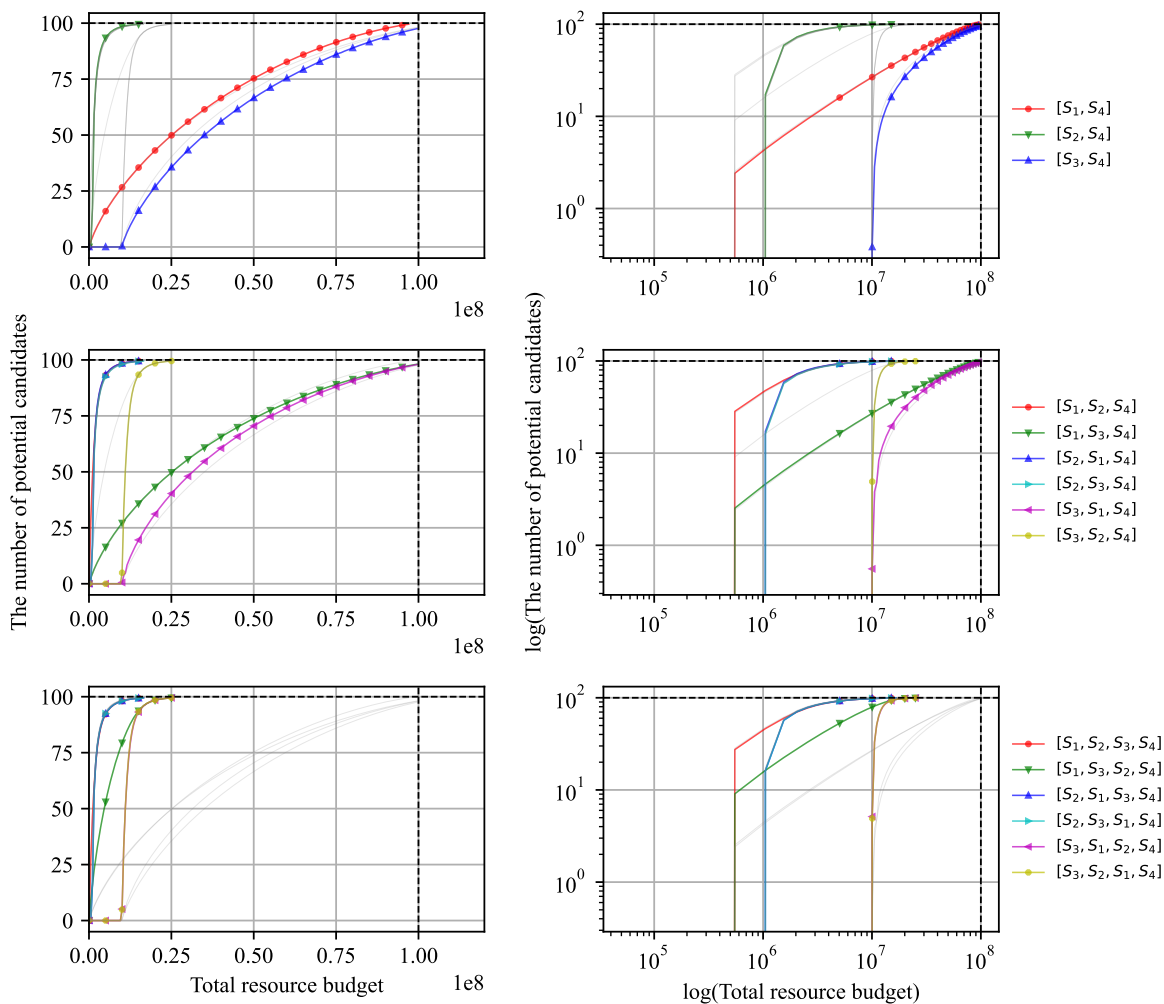


Figure S17: Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.

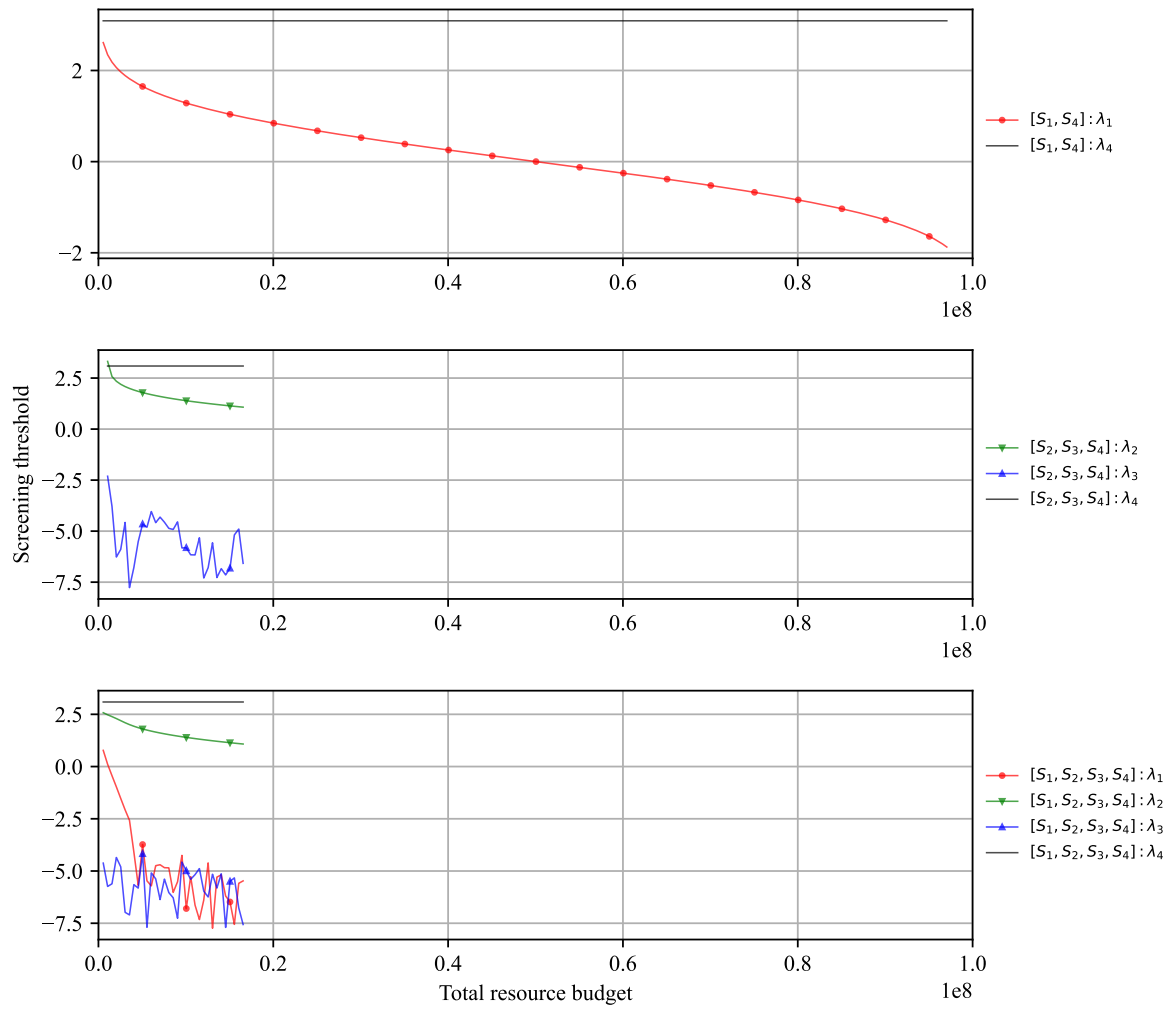


Figure S18: Screening thresholds of the optimized pipelines.

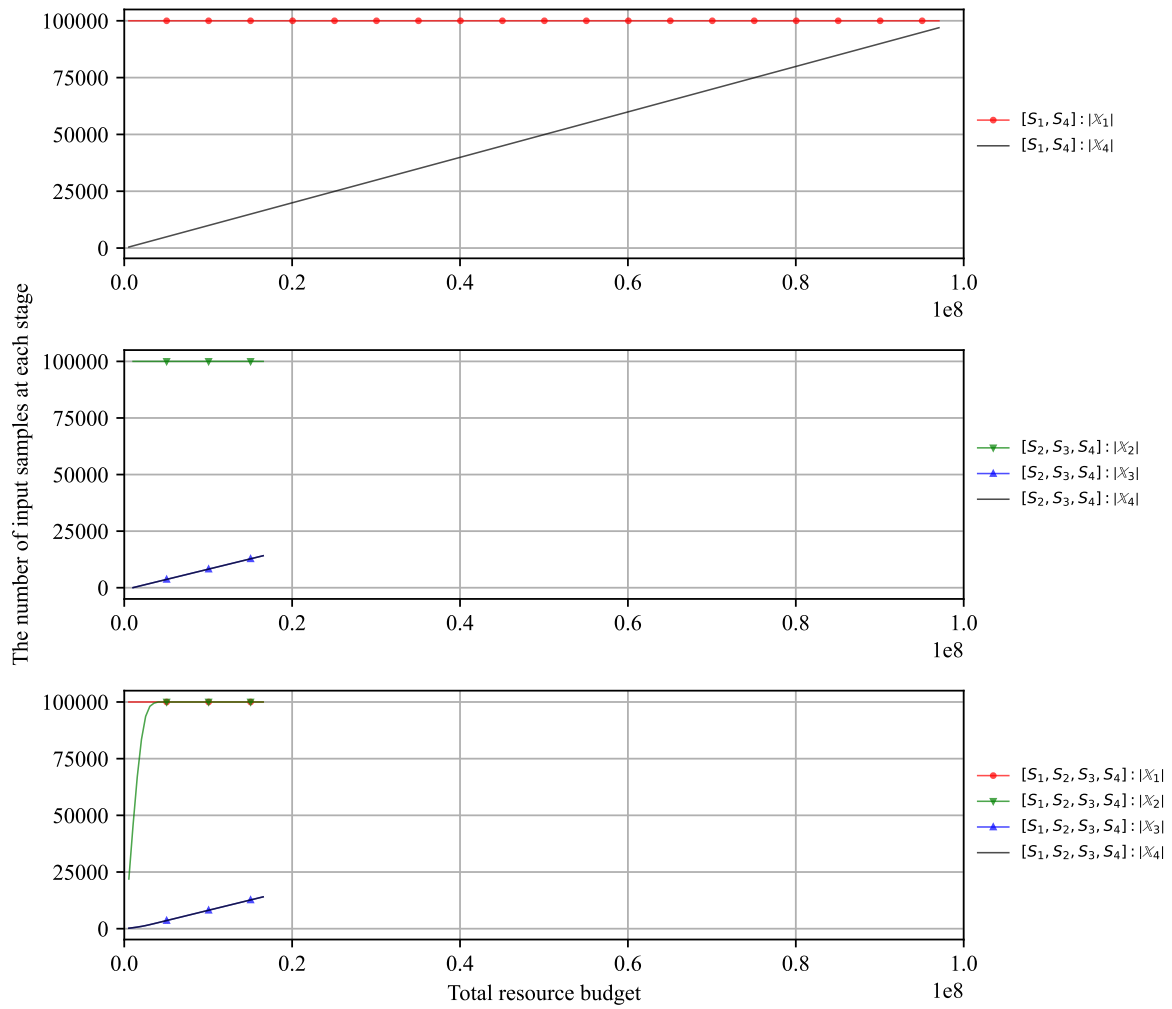


Figure S19: The number of input samples at each stage.

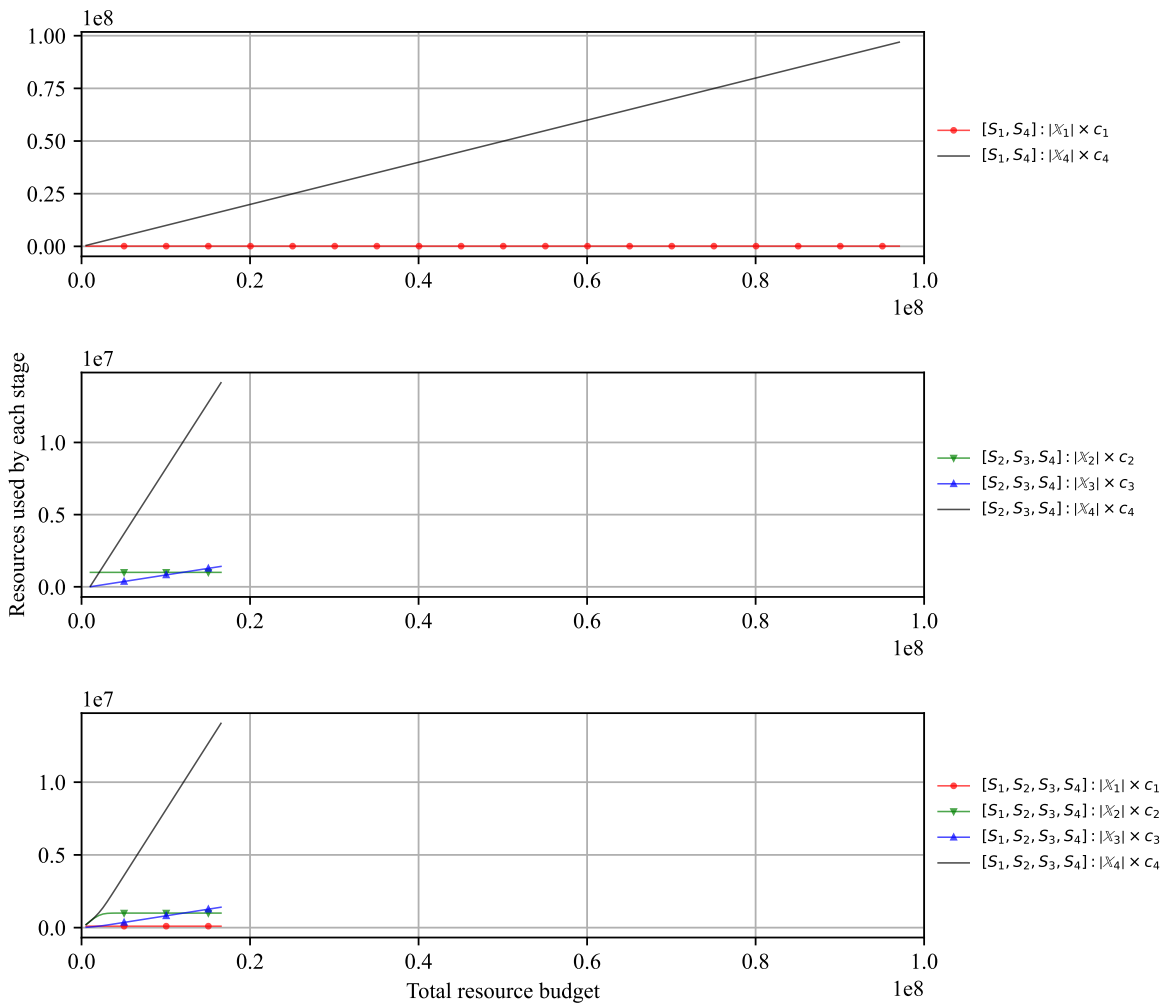


Figure S20: Resources used by each stage.

$$p(y_1, y_2, y_3, y_4) \sim \mathcal{N} \left(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.2 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.2 & 0.2 & 1 & 0.8 \\ 0.2 & 0.2 & 0.8 & 1 \end{bmatrix} \right) \quad (6)$$

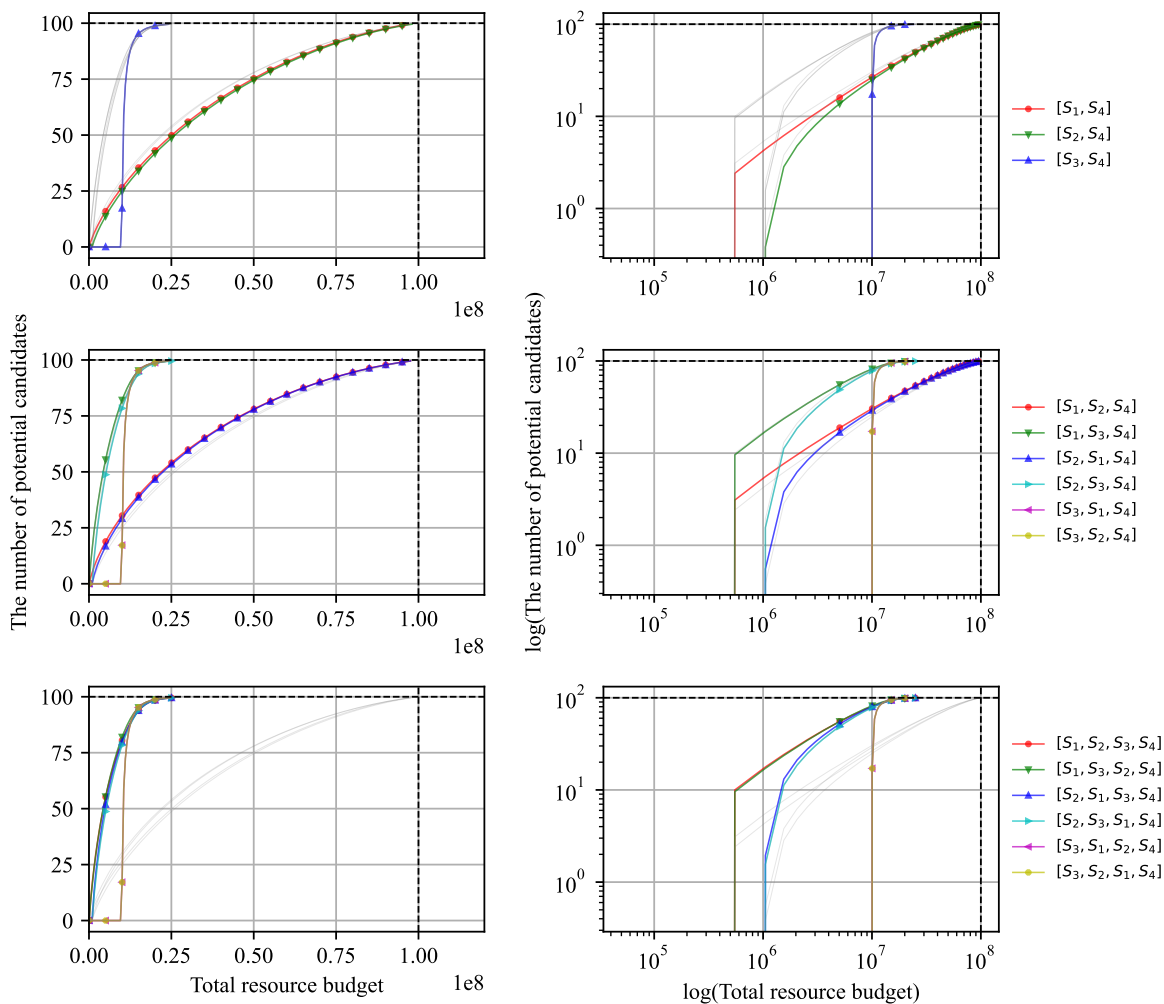


Figure S21: Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.

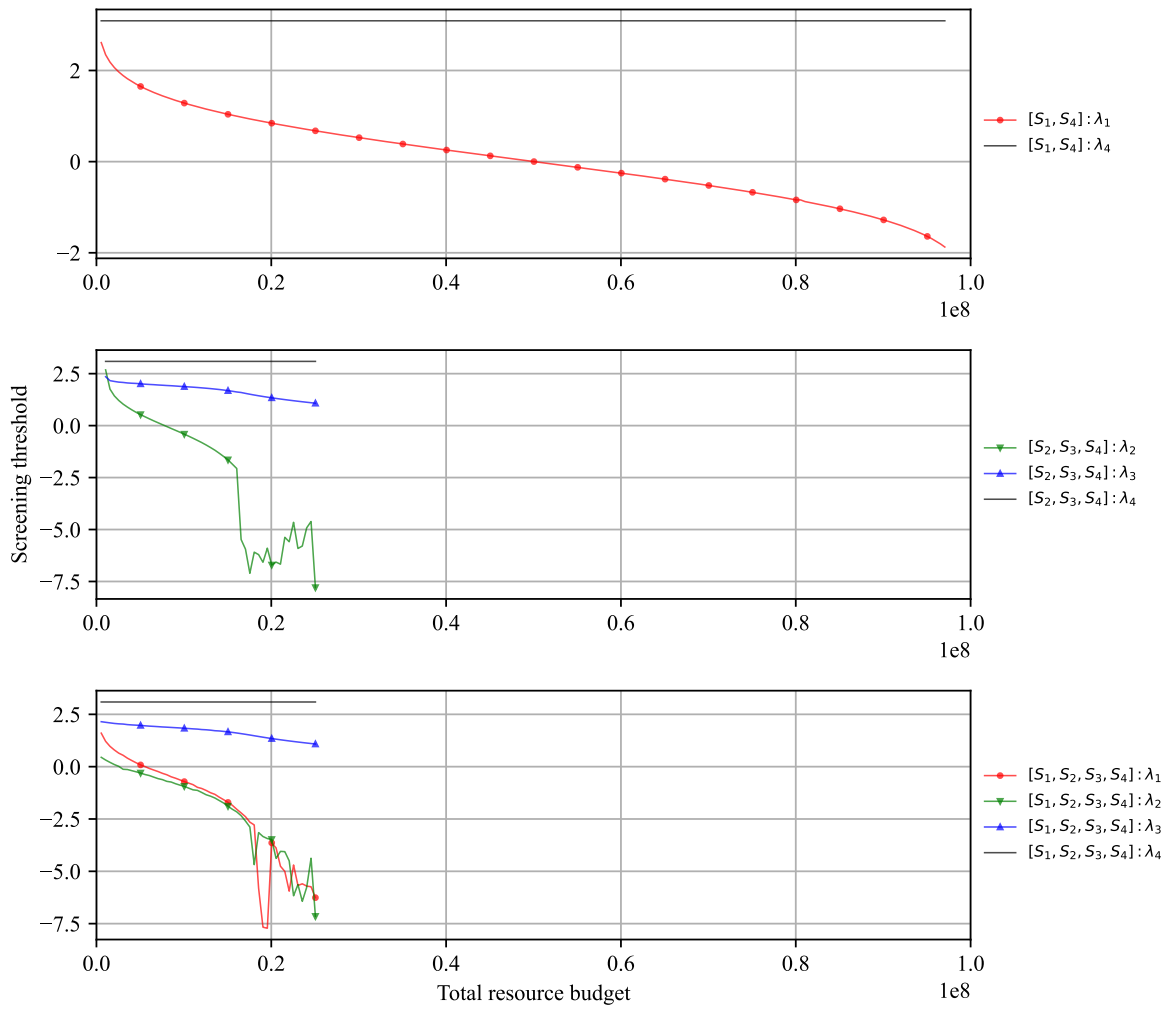


Figure S22: Screening thresholds of the optimized pipelines.

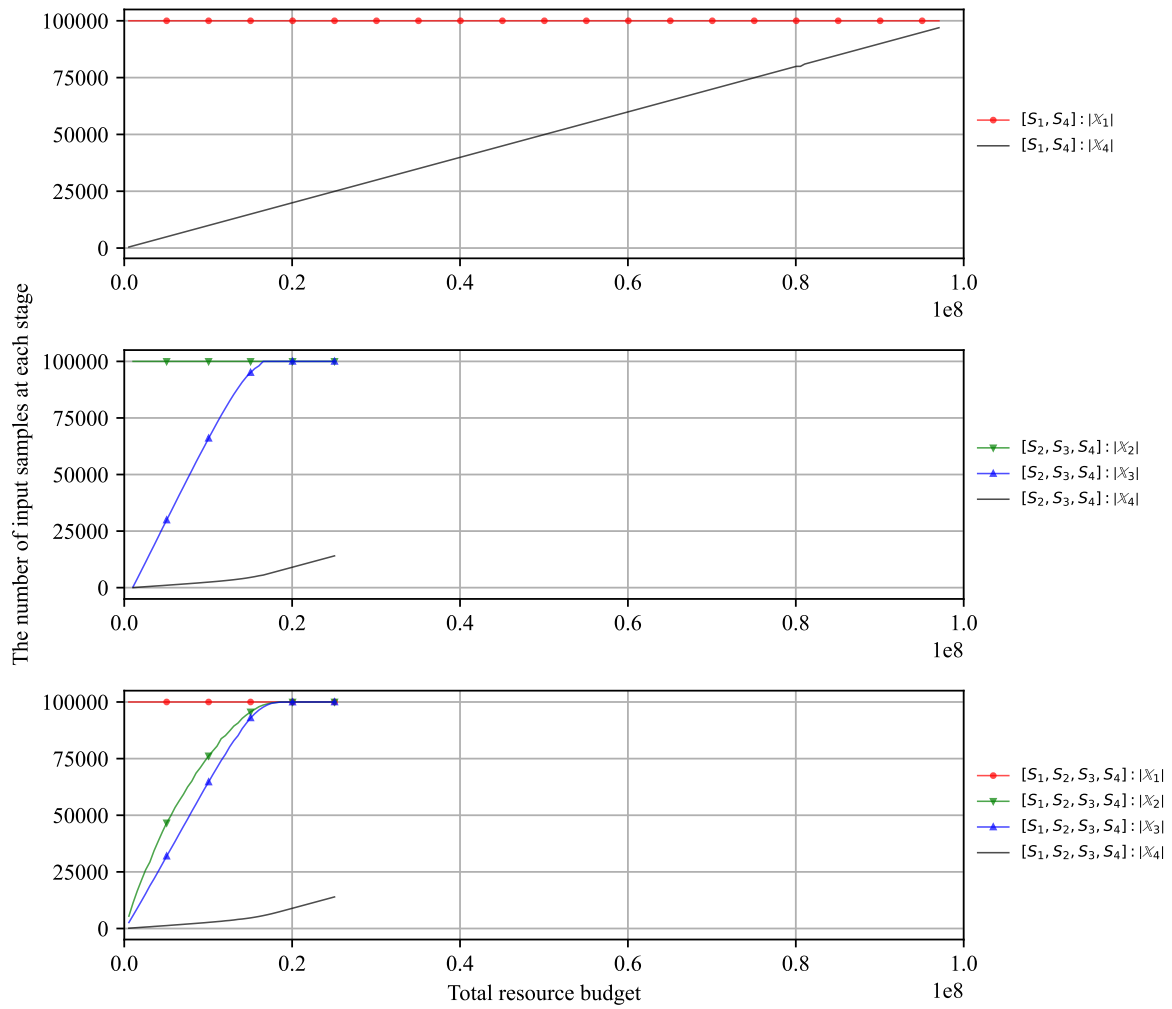


Figure S23: The number of input samples at each stage.

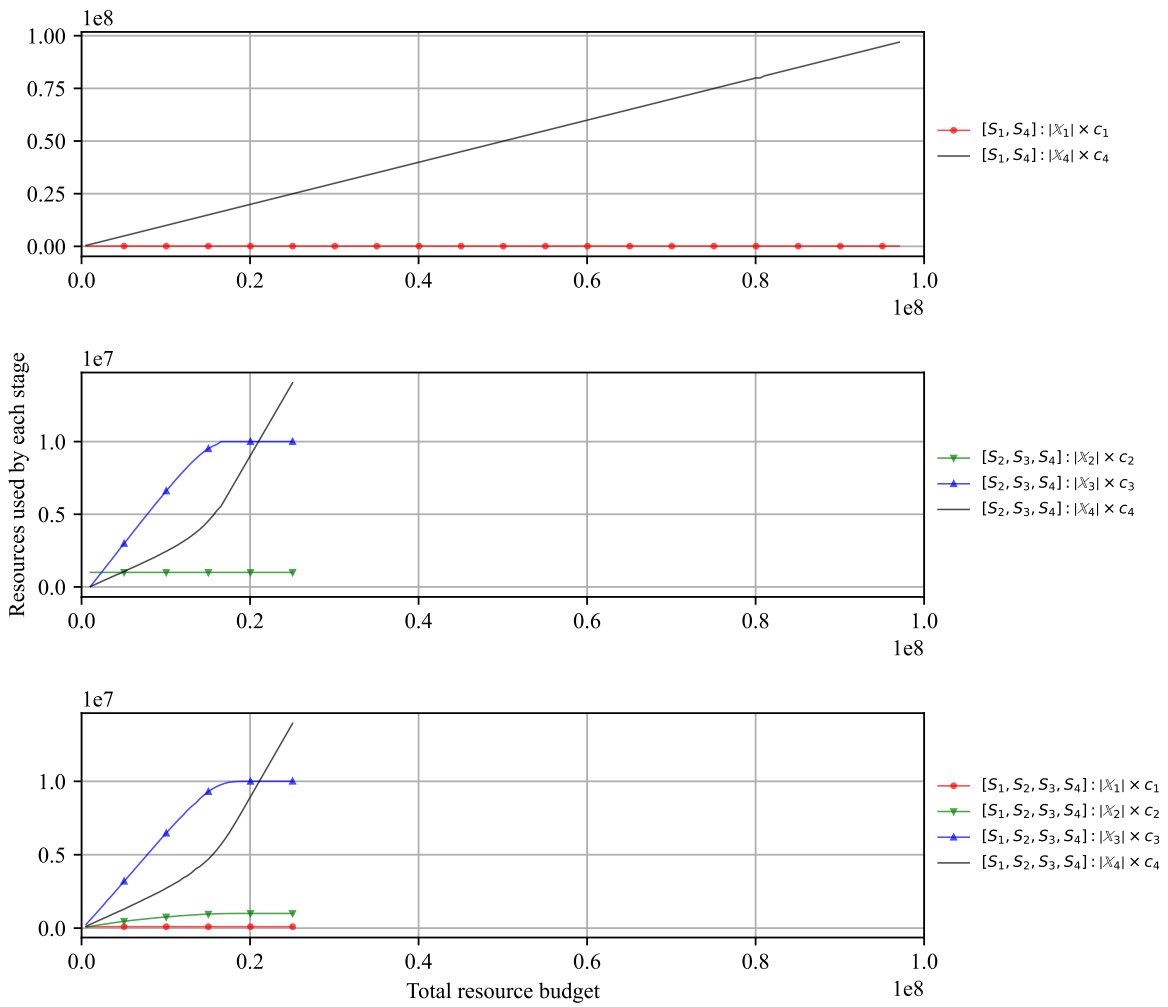


Figure S24: Resources used by each stage.

$$p(y_1, y_2, y_3, y_4) \sim \mathcal{N} \left(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.2 & 0.2 & 0.2 \\ 0.2 & 1 & 0.8 & 0.8 \\ 0.2 & 0.8 & 1 & 0.8 \\ 0.2 & 0.8 & 0.8 & 1 \end{bmatrix} \right) \quad (7)$$

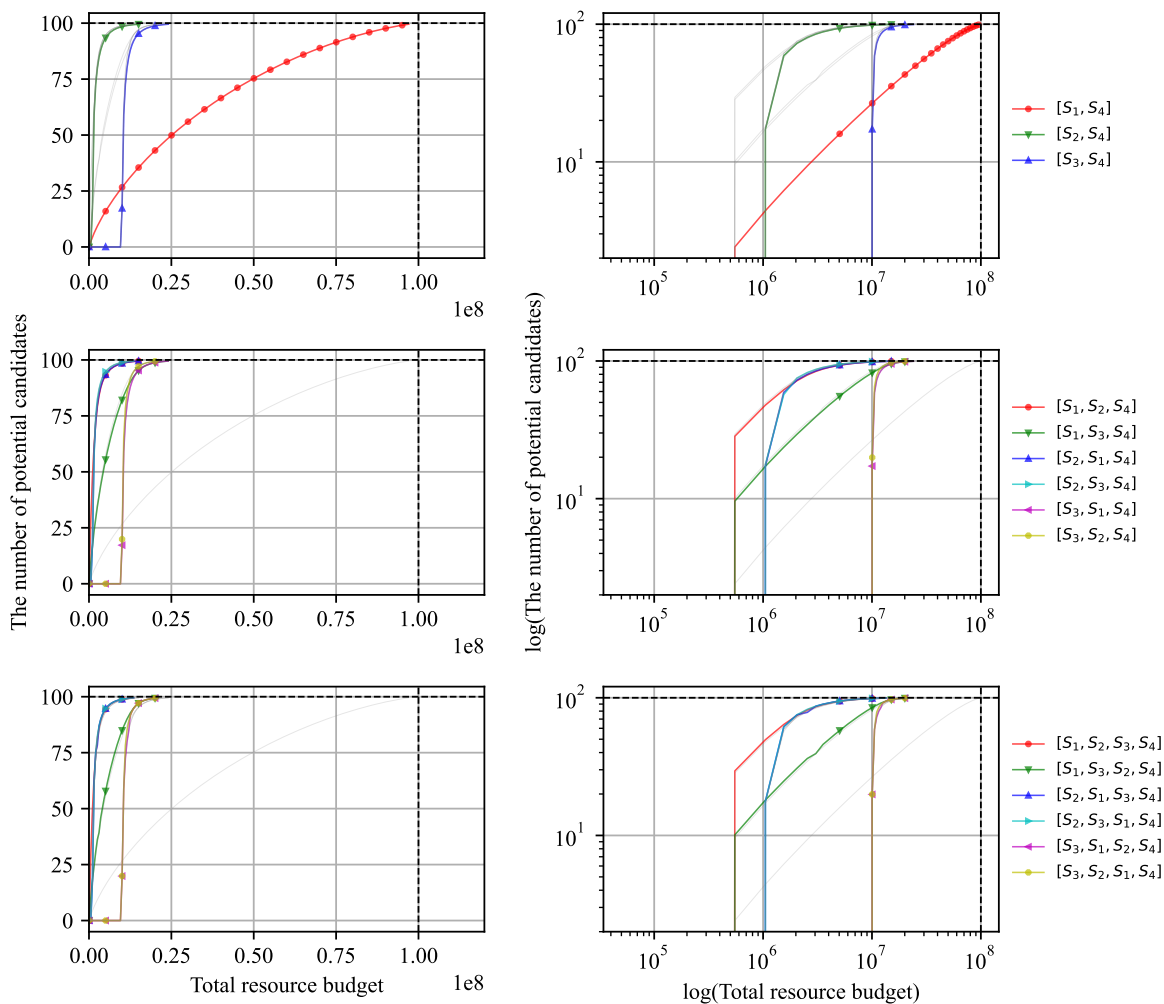


Figure S25: Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.

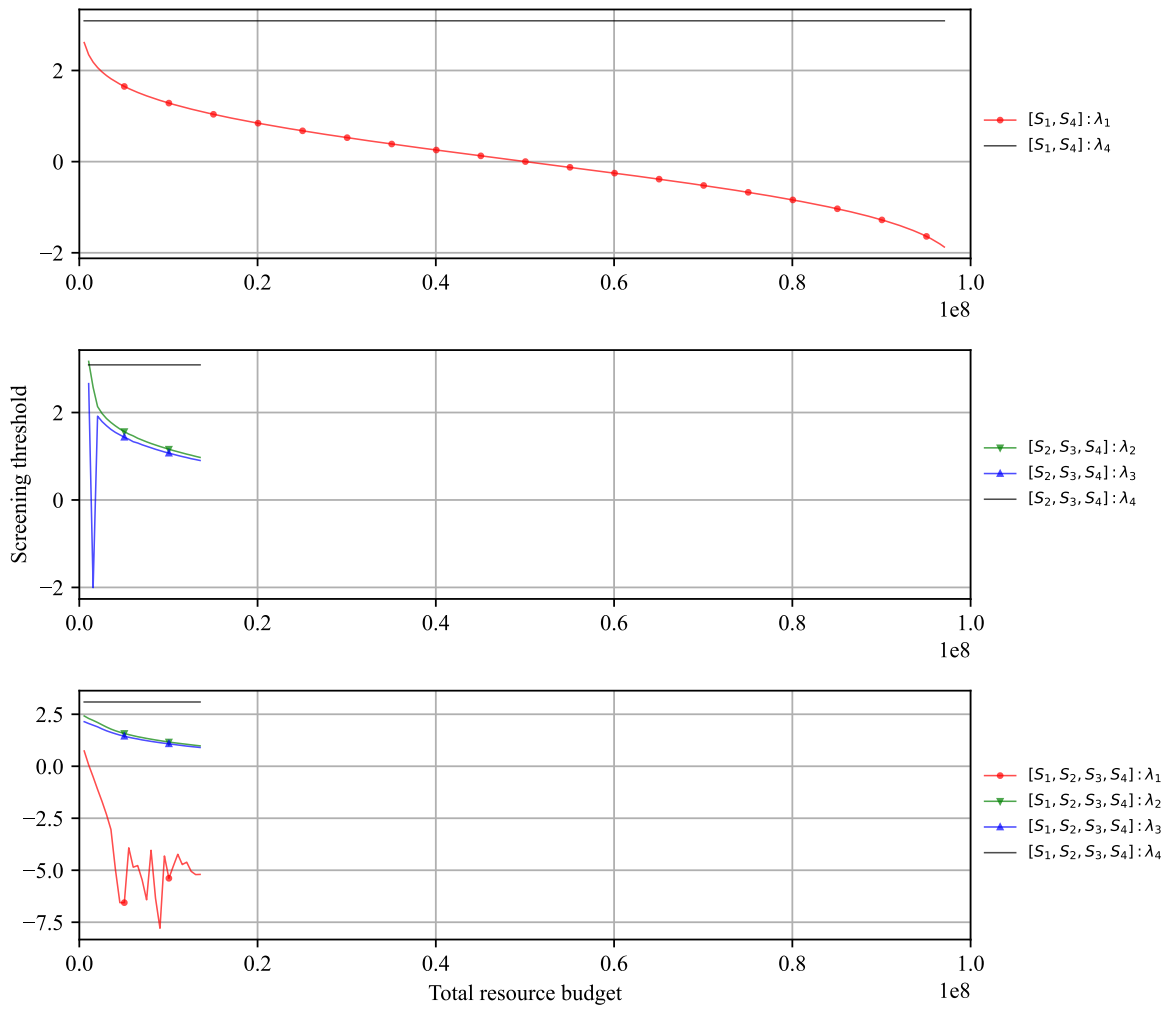


Figure S26: Screening thresholds of the optimized pipelines.

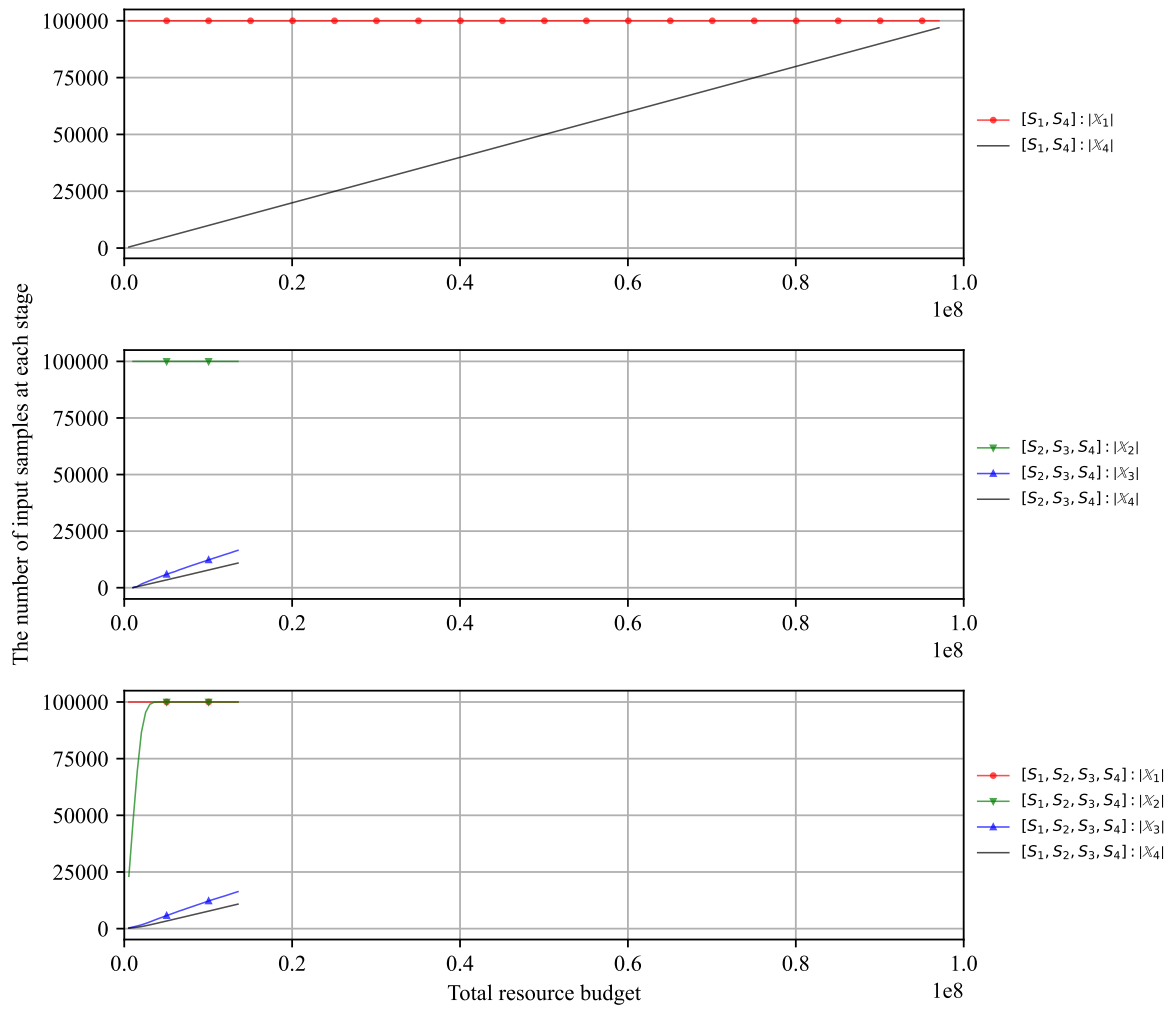


Figure S27: The number of input samples at each stage.

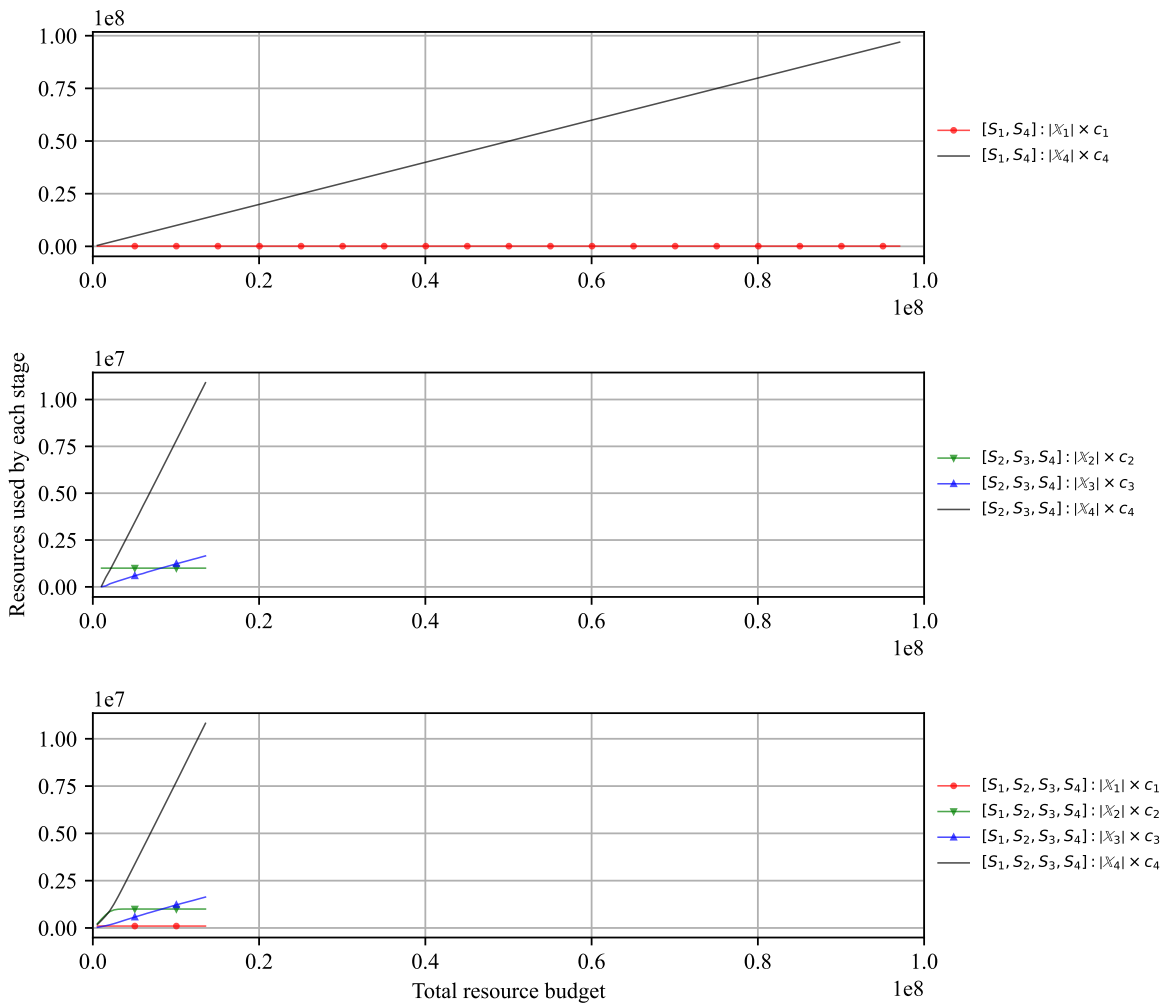


Figure S28: Resources used by each stage.

$$p(y_1, y_2, y_3, y_4) \sim \mathcal{N} \left(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.2 & 0.8 & 0.8 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.8 & 0.2 & 1 & 0.8 \\ 0.8 & 0.2 & 0.8 & 1 \end{bmatrix} \right) \quad (8)$$

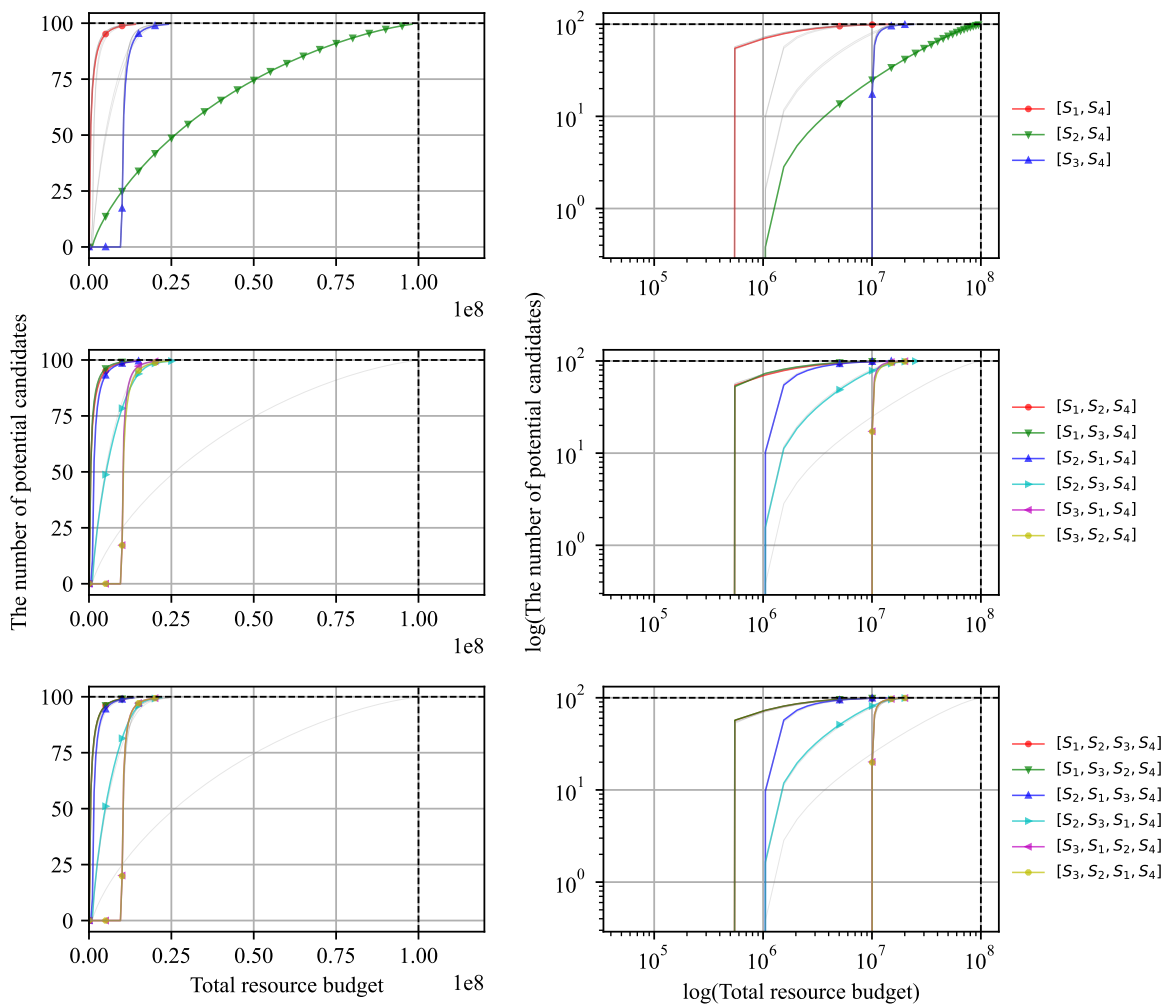


Figure S29: Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.

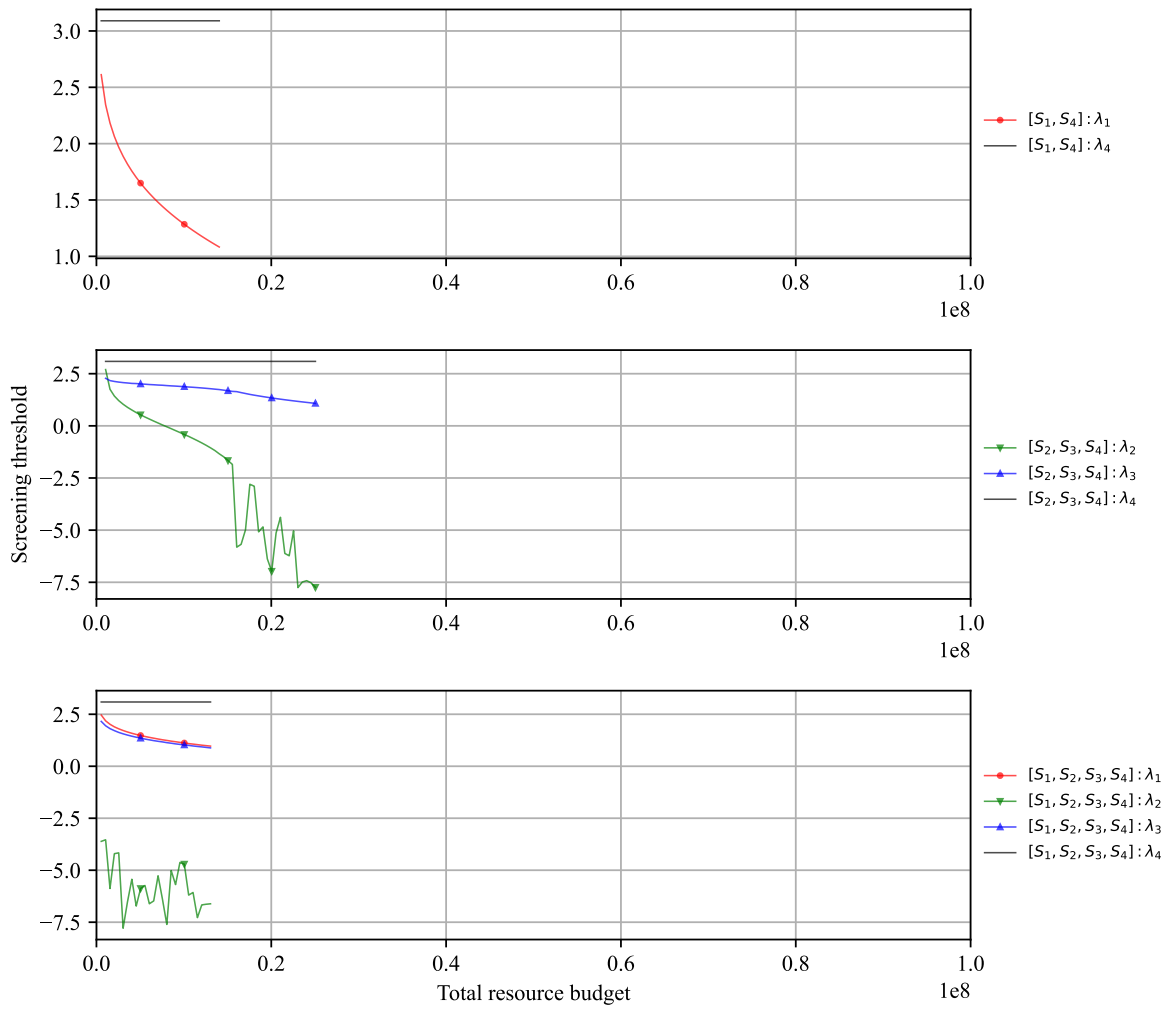


Figure S30: Screening thresholds of the optimized pipelines.

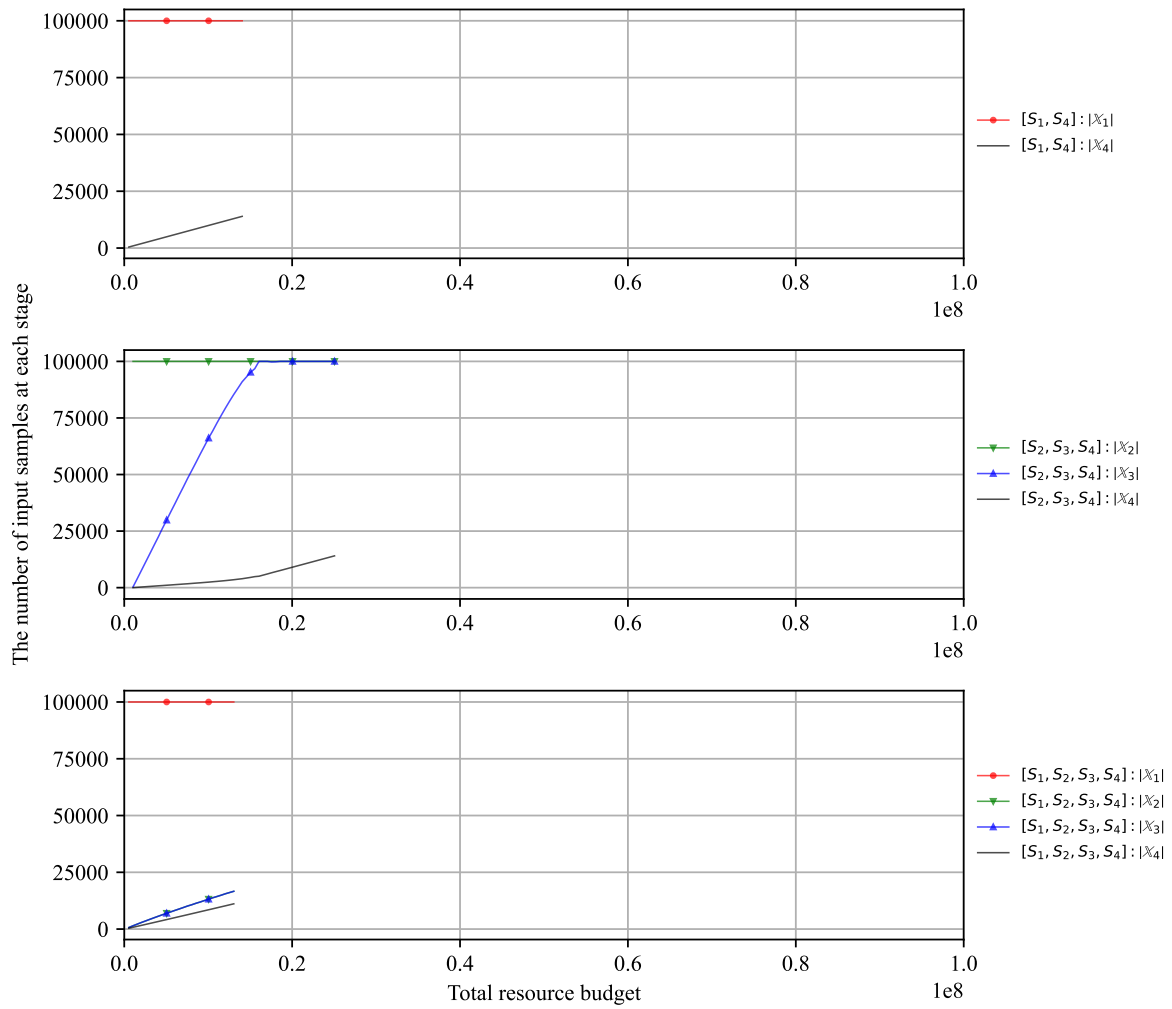


Figure S31: The number of input samples at each stage.

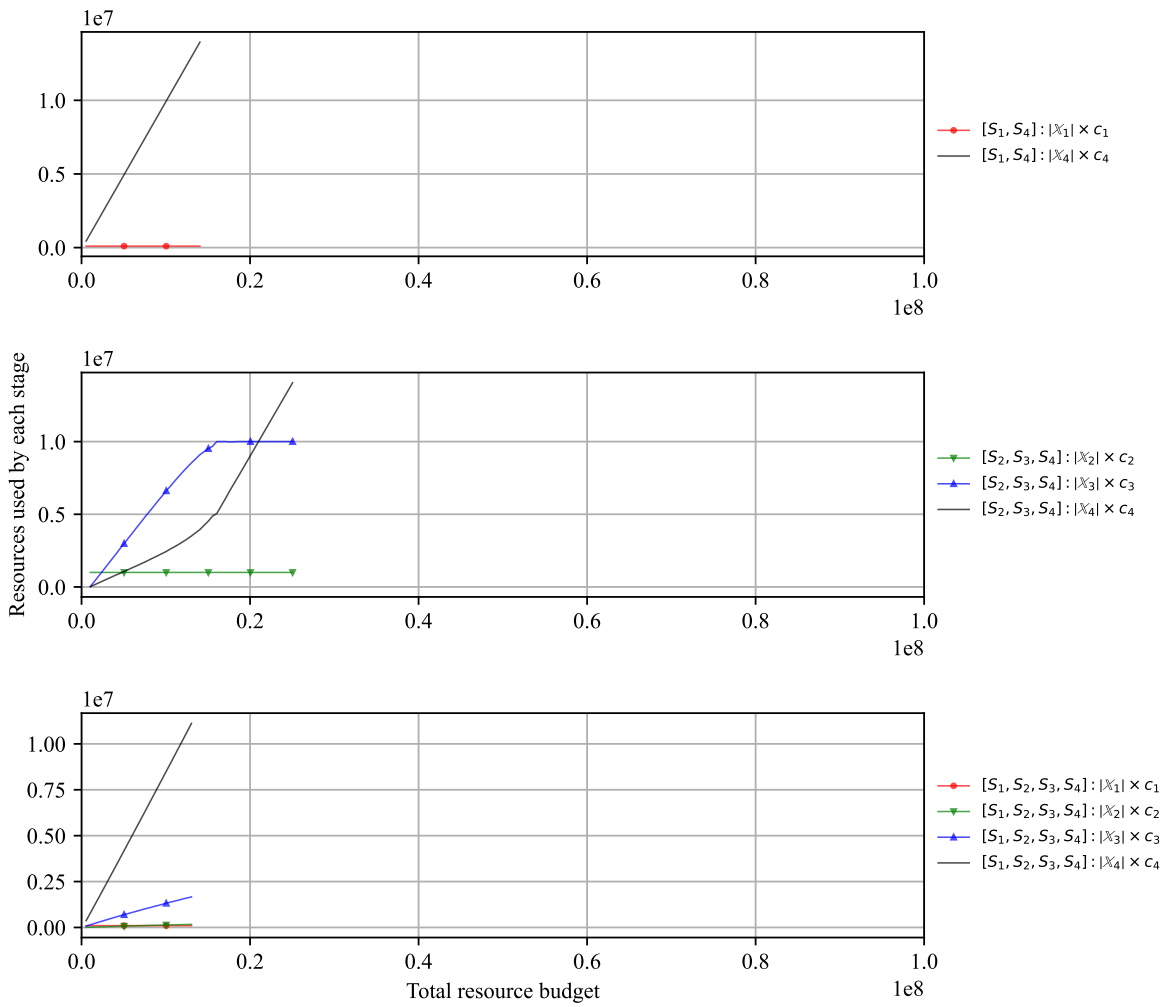


Figure S32: Resources used by each stage.

$$p(y_1, y_2, y_3, y_4) \sim \mathcal{N} \left(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.8 & 0.2 & 0.8 \\ 0.8 & 1 & 0.2 & 0.8 \\ 0.2 & 0.2 & 1 & 0.2 \\ 0.8 & 0.8 & 0.2 & 1 \end{bmatrix} \right) \quad (9)$$

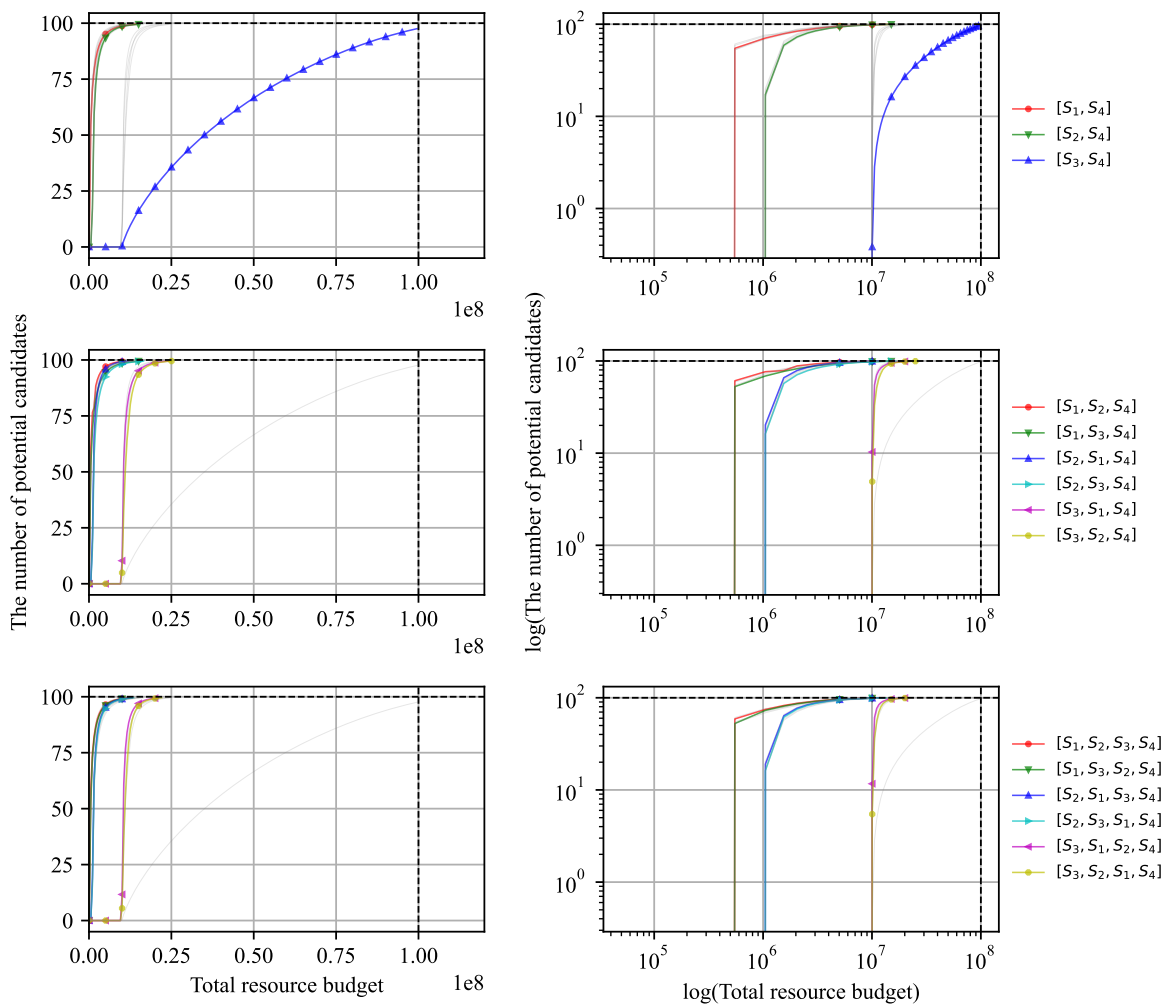


Figure S33: Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.

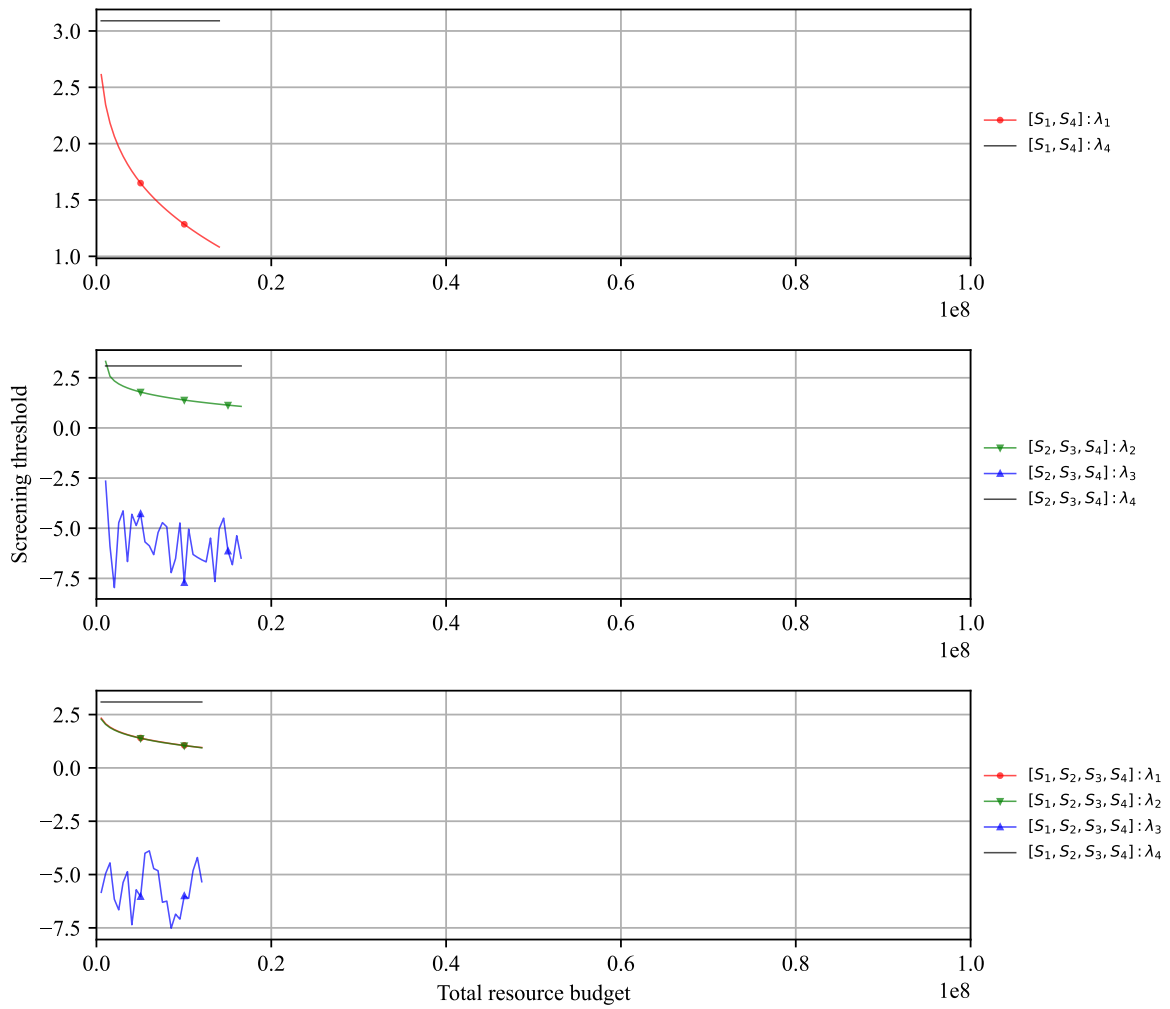


Figure S34: Screening thresholds of the optimized pipelines.

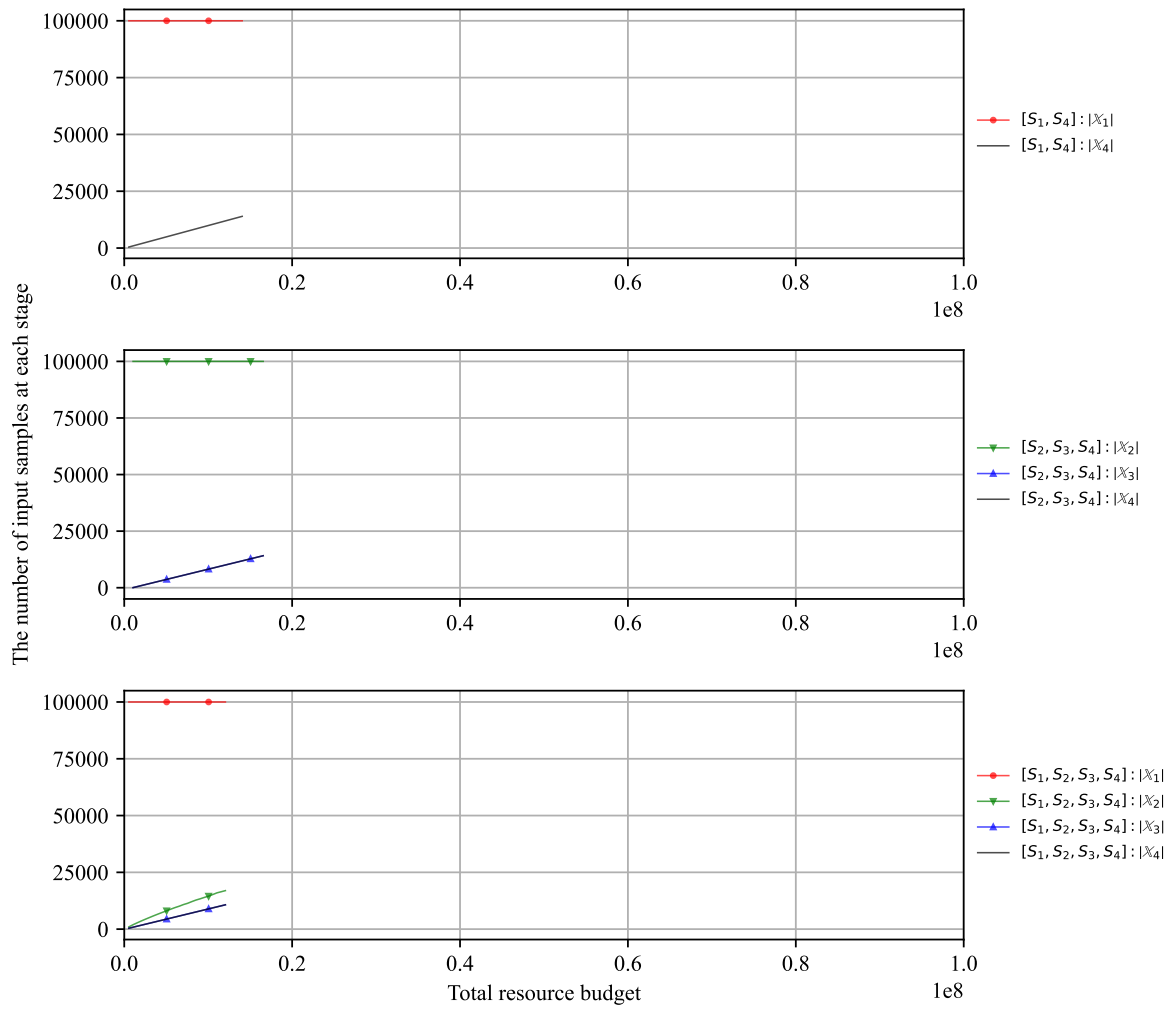


Figure S35: The number of input samples at each stage.

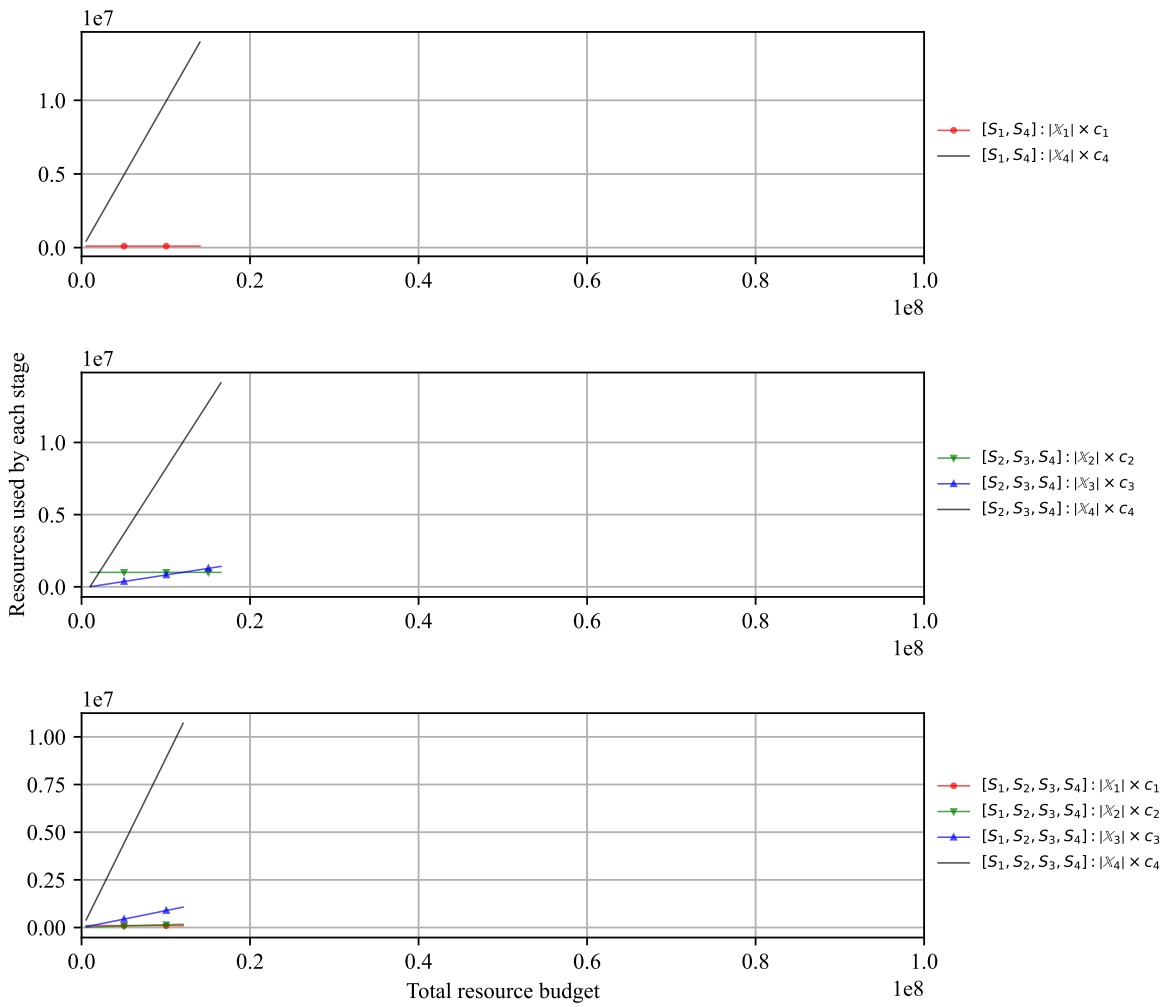


Figure S36: Resources used by each stage.

$$p(y_1, y_2, y_3, y_4) \sim \mathcal{N} \left(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.3 & 0.2 & 0.1 \\ 0.3 & 1 & 0.3 & 0.2 \\ 0.2 & 0.3 & 1 & 0.3 \\ 0.1 & 0.2 & 0.3 & 1 \end{bmatrix} \right) \quad (10)$$

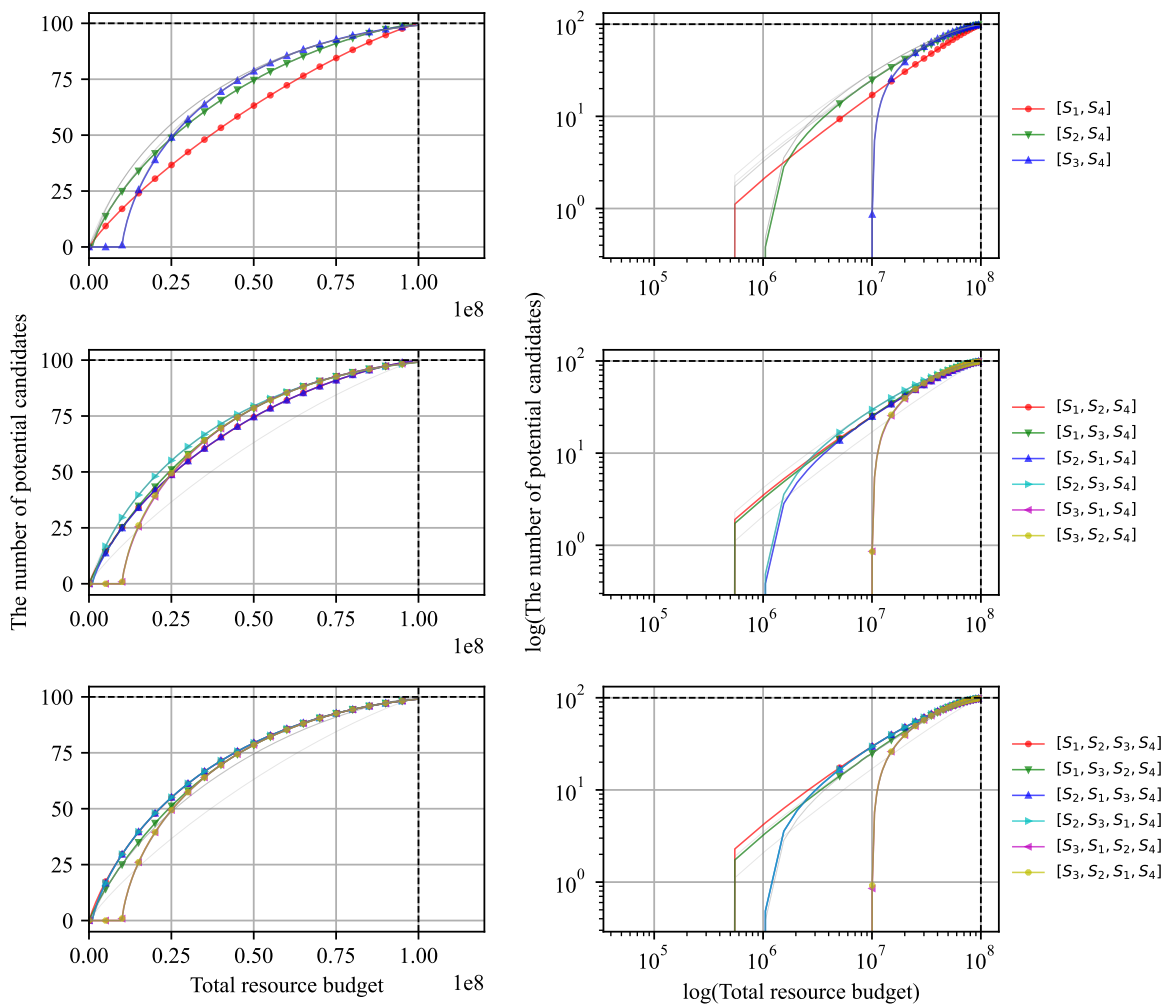


Figure S37: Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.

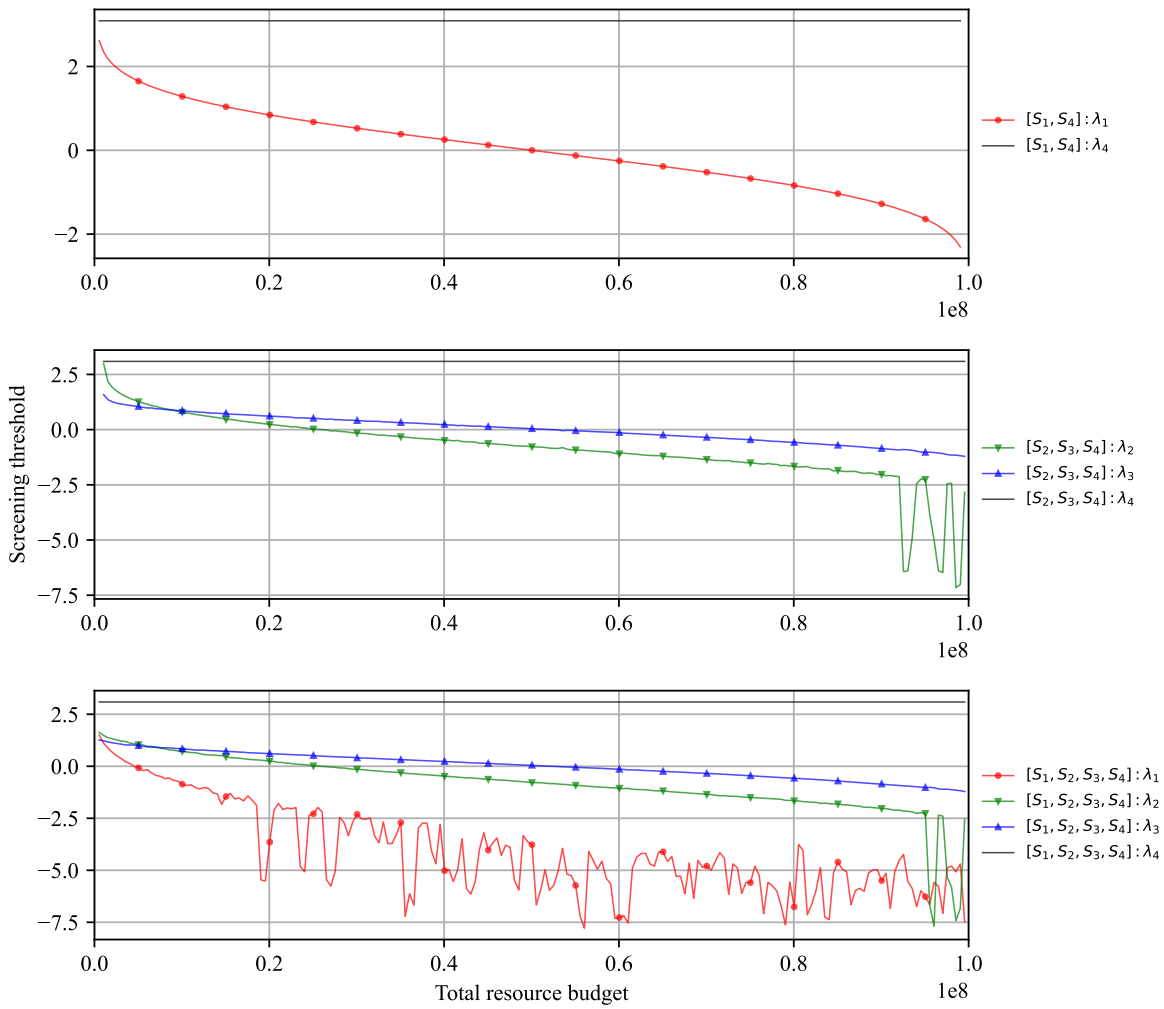


Figure S38: Screening thresholds of the optimized pipelines.

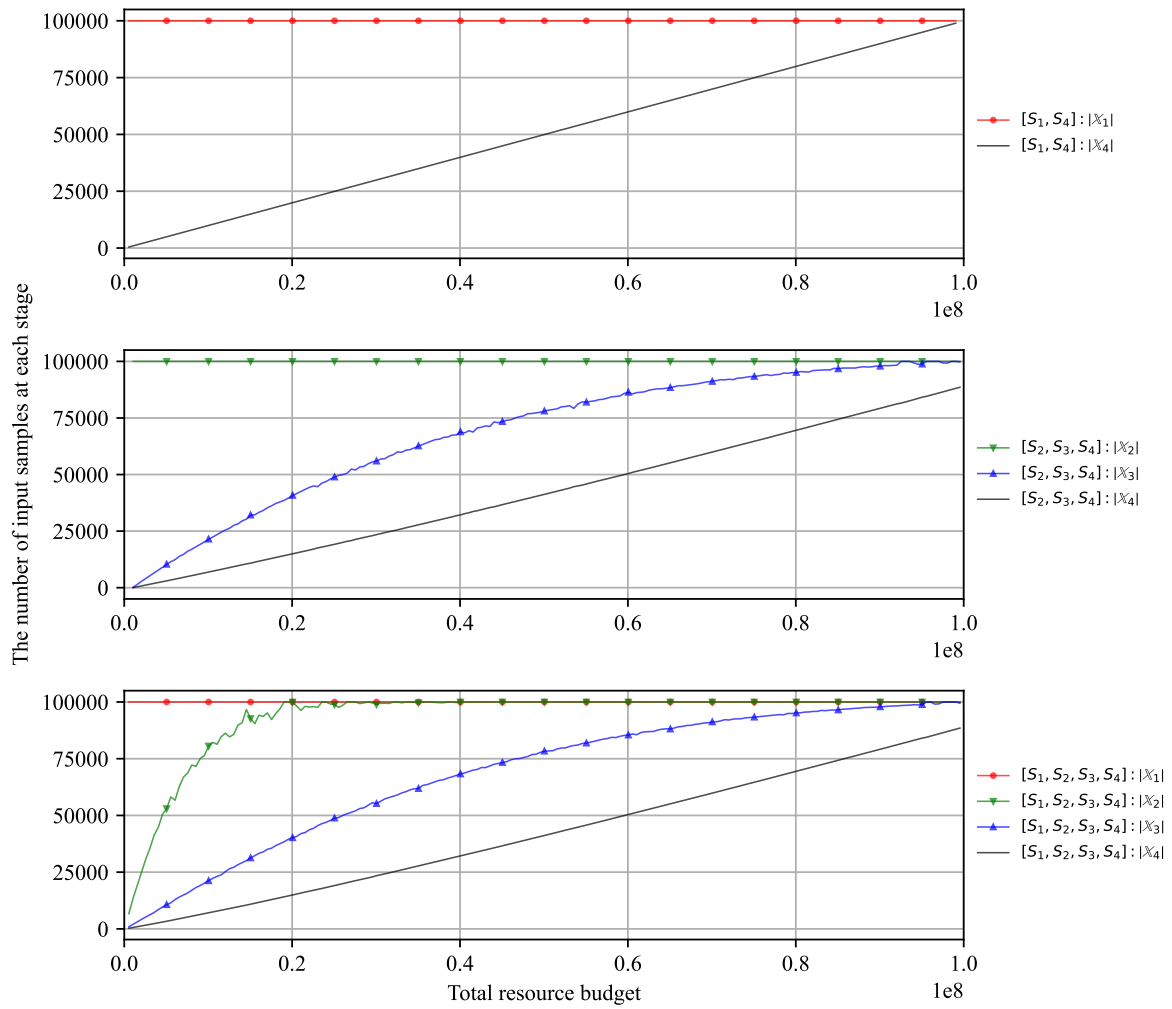


Figure S39: The number of input samples at each stage.

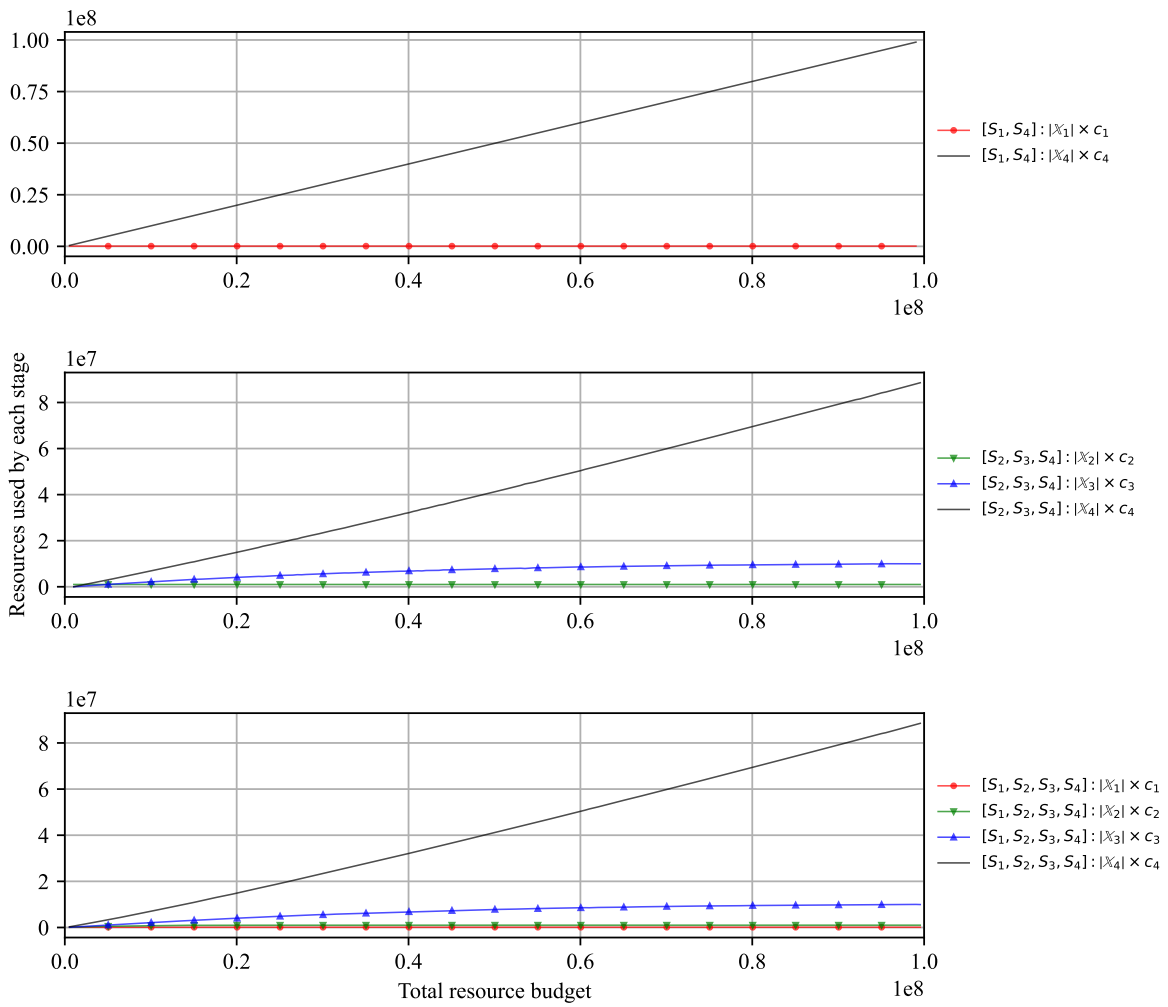


Figure S40: Resources used by each stage.

$$p(y_1, y_2, y_3, y_4) \sim \mathcal{N} \left(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.8 & 0.7 & 0.6 \\ 0.8 & 1 & 0.8 & 0.7 \\ 0.7 & 0.8 & 1 & 0.8 \\ 0.6 & 0.7 & 0.8 & 1 \end{bmatrix} \right) \quad (11)$$

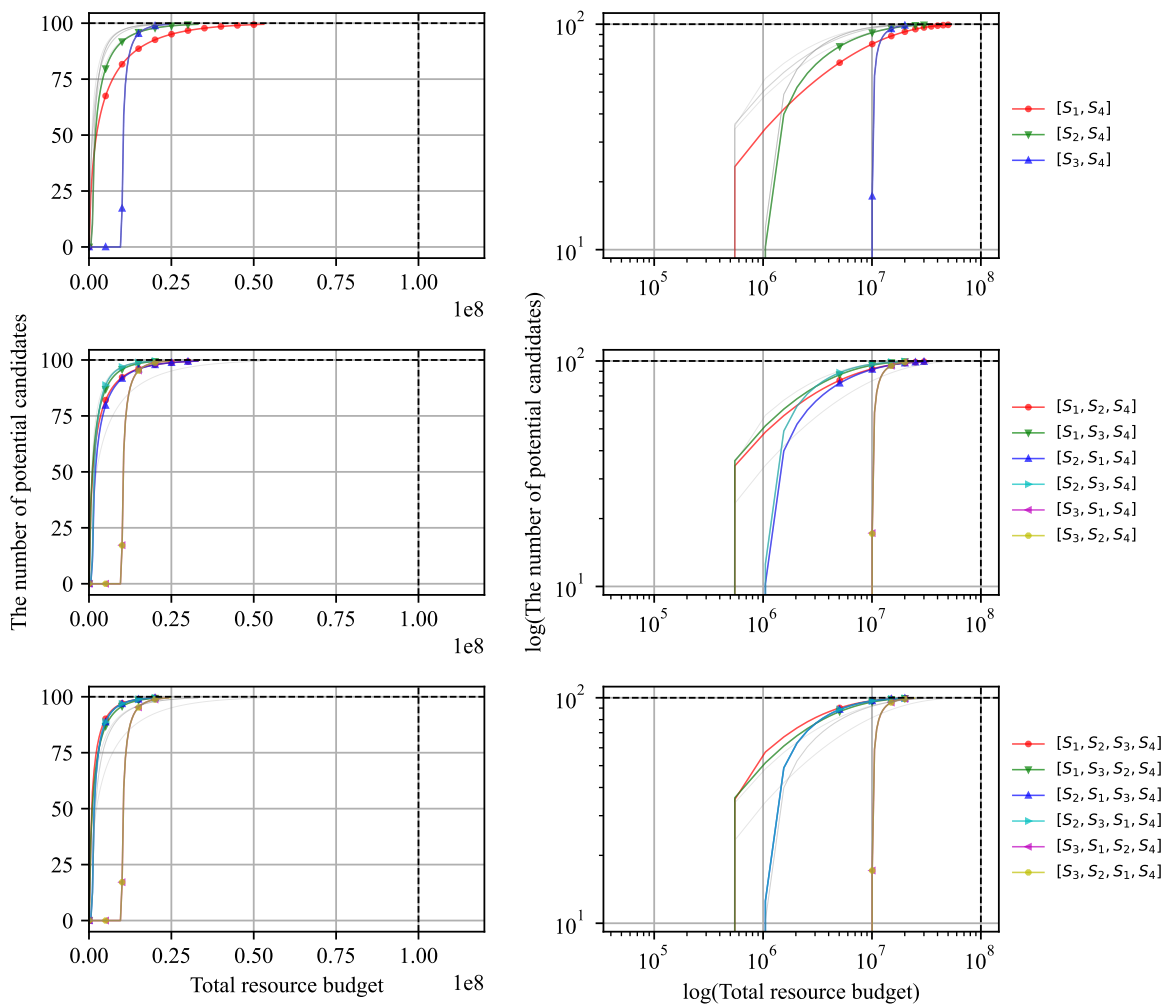


Figure S41: Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.

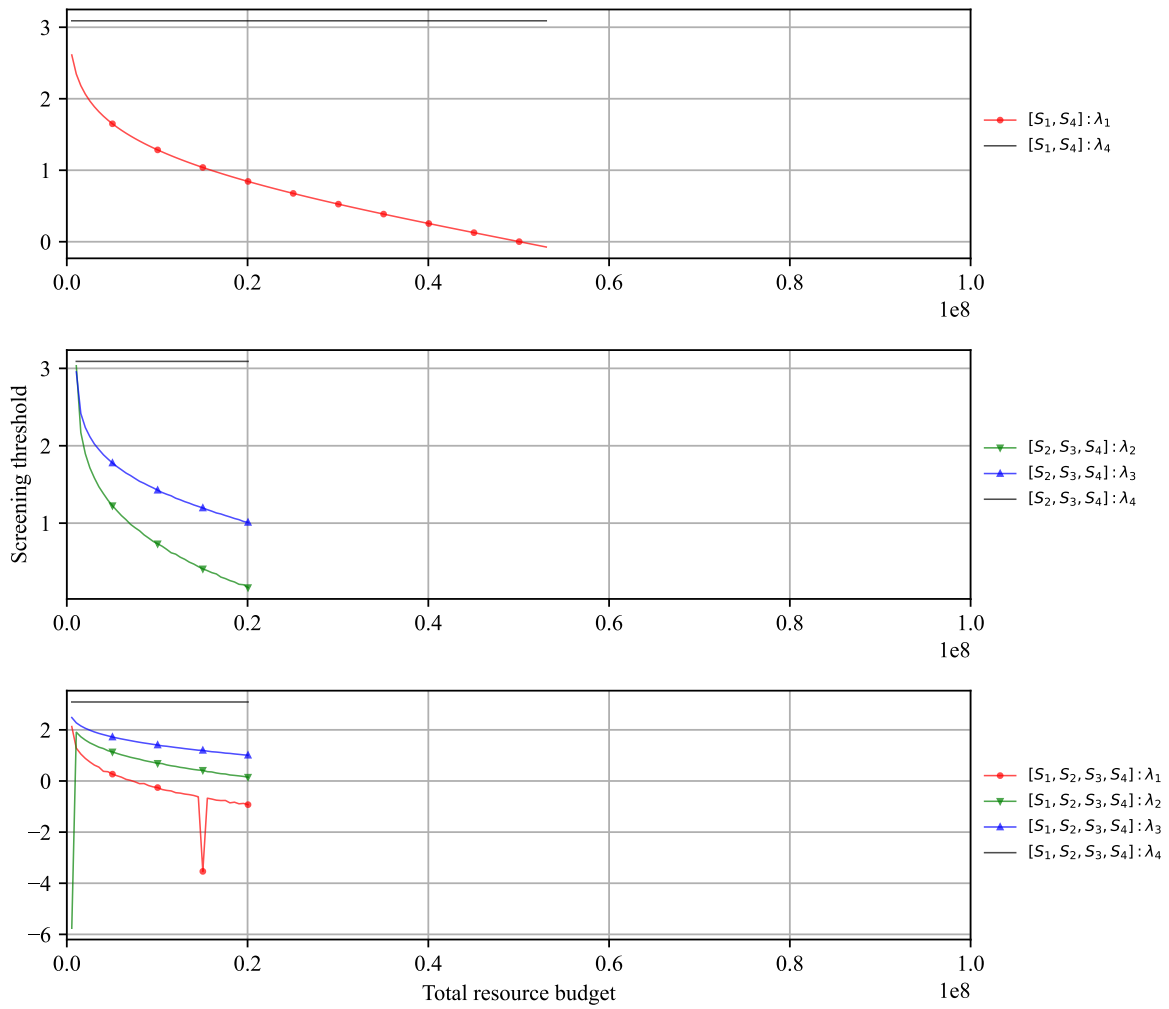


Figure S42: Screening thresholds of the optimized pipelines.

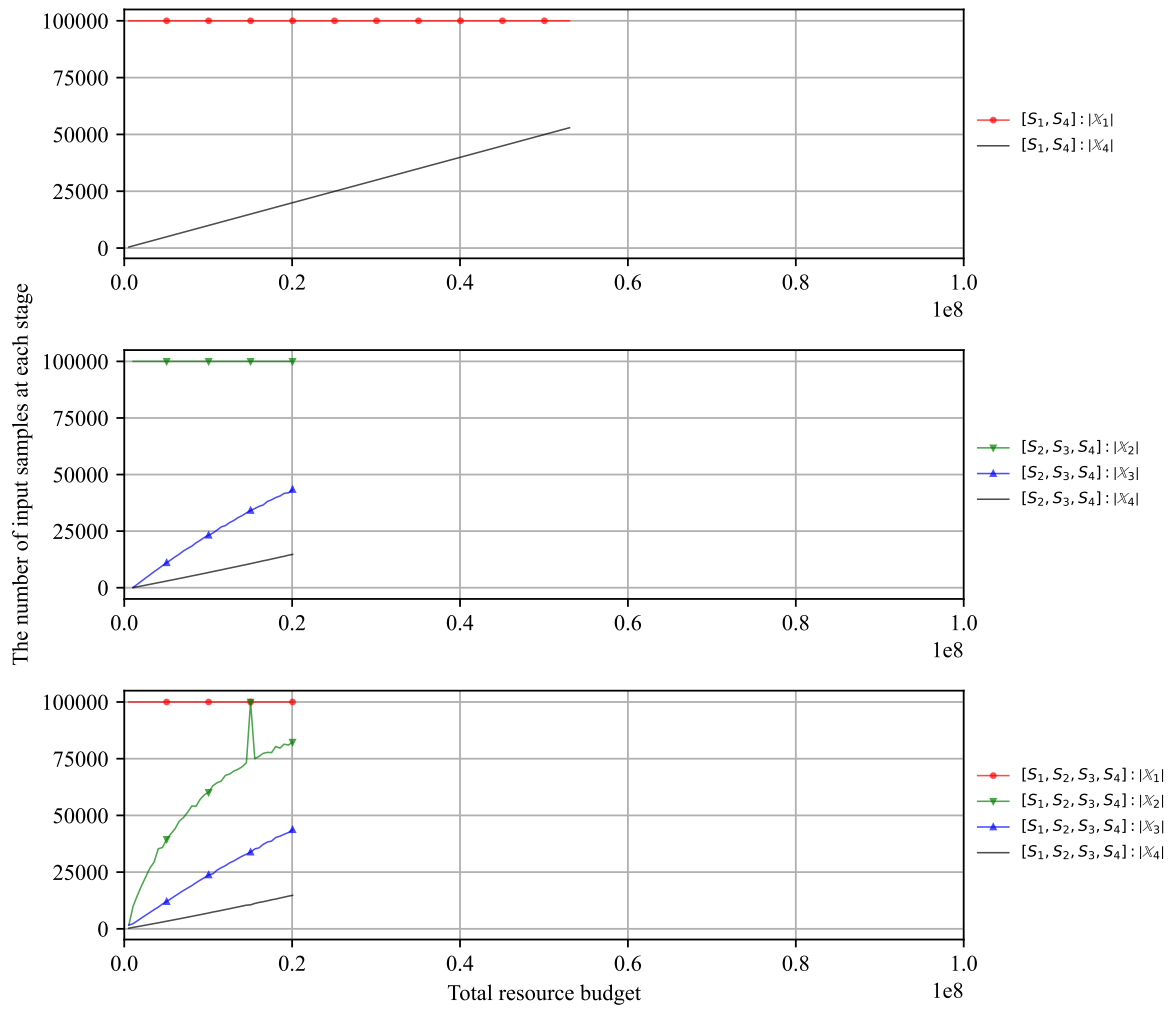


Figure S43: The number of input samples at each stage.

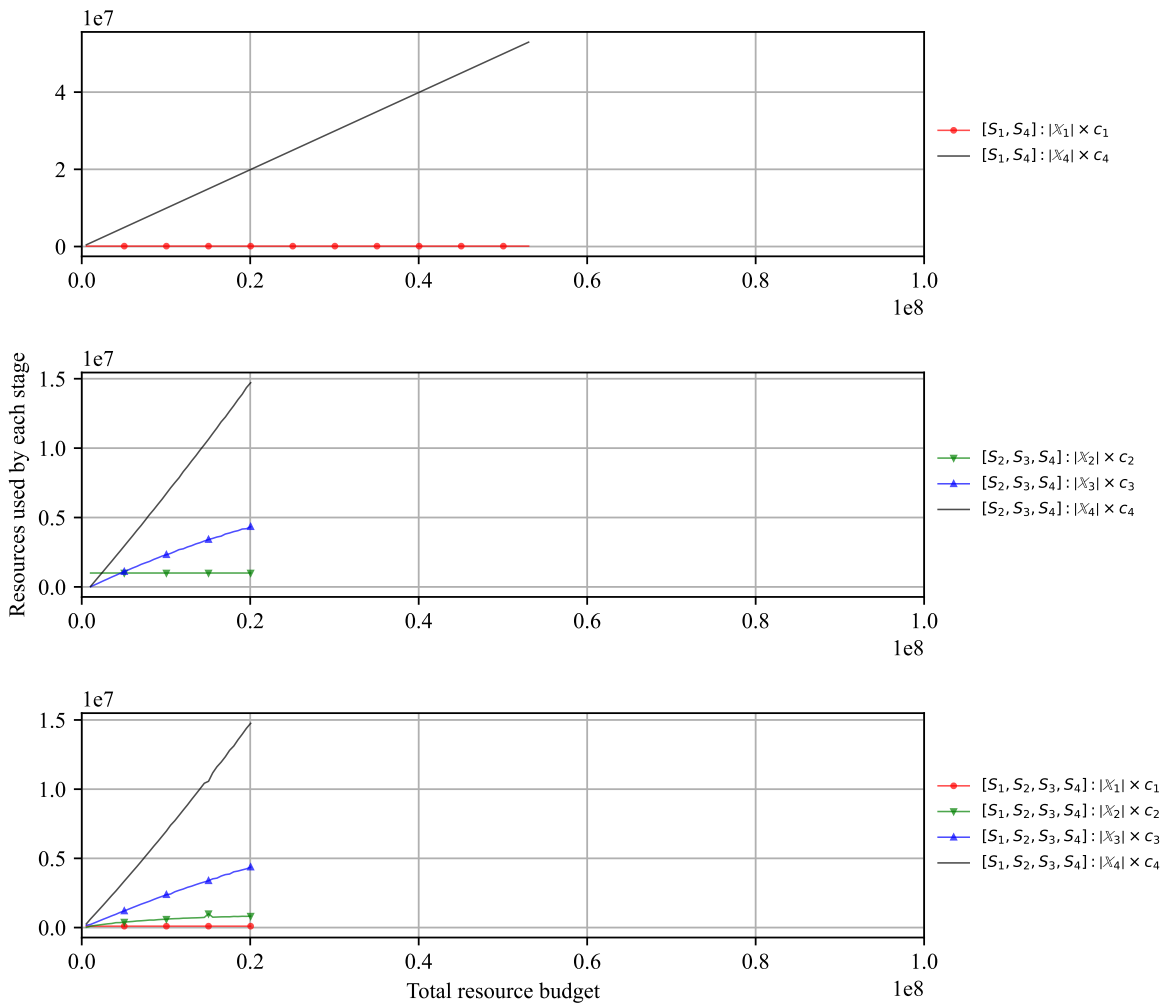


Figure S44: Resources used by each stage.

Performance evaluations of high-throughput virtual screening pipelines for identifying long non-coding RNAs

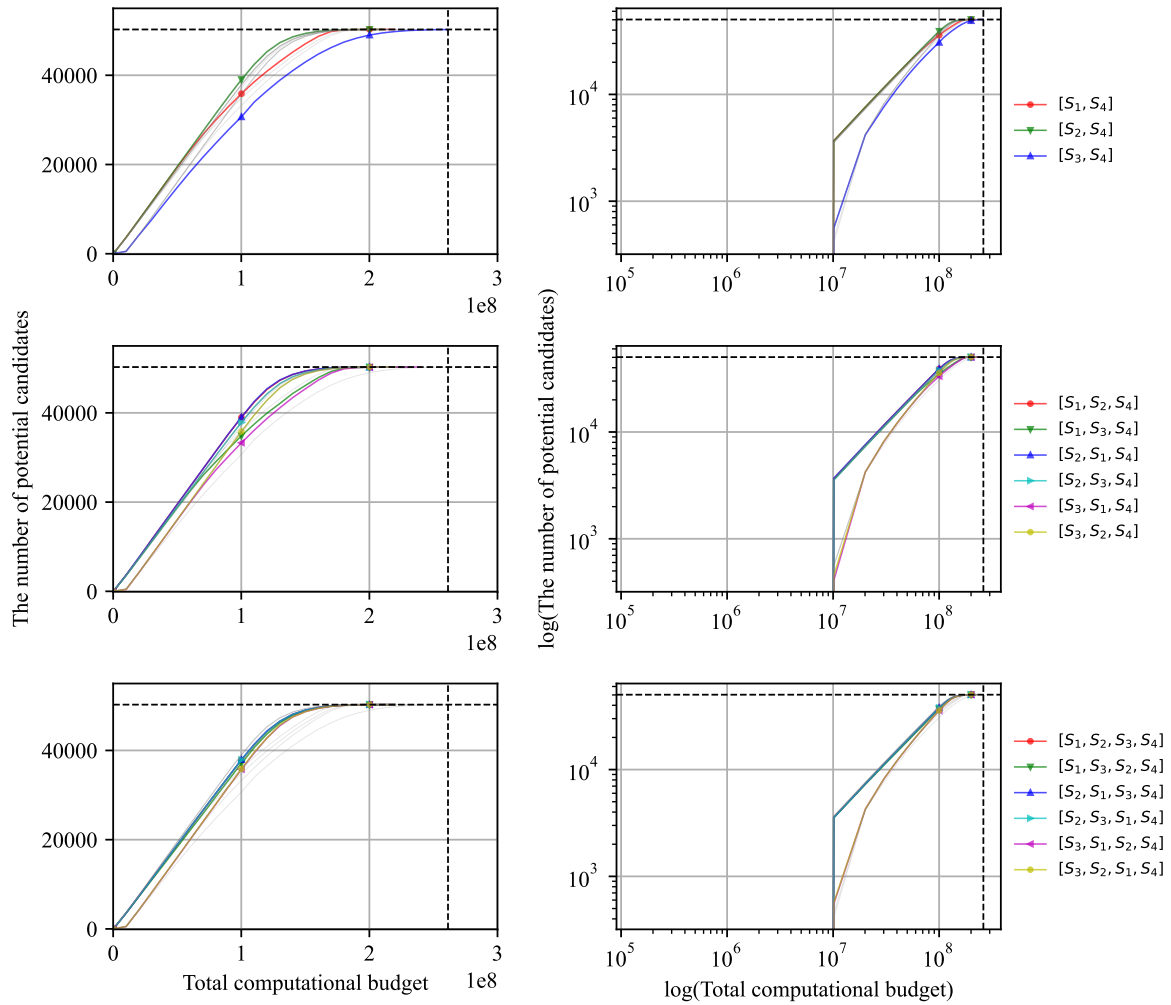


Figure S45: Performance comparison of the optimized high-throughput virtual screening pipelines in terms of discovery capability.

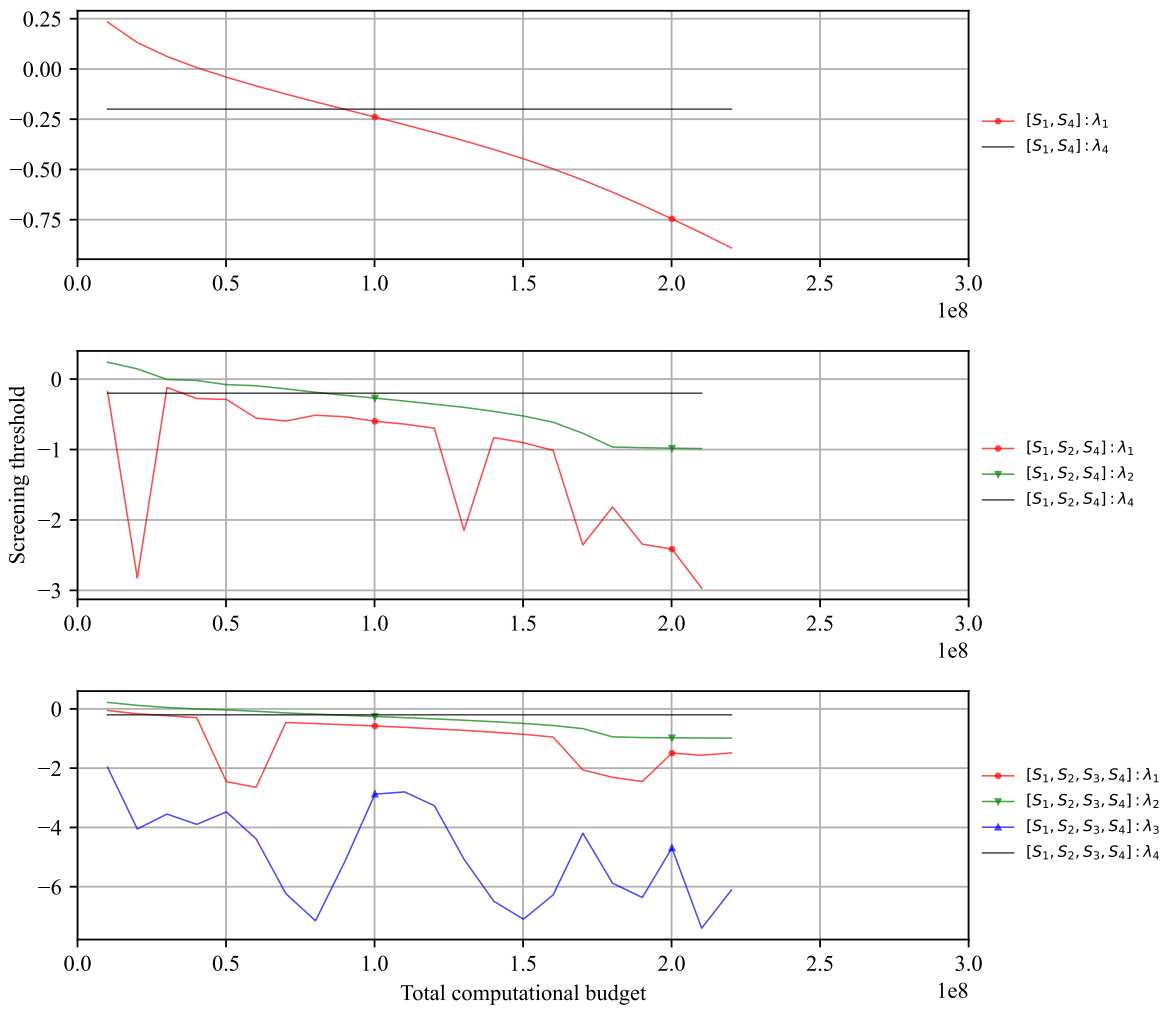


Figure S46: Screening thresholds of the optimized pipelines.

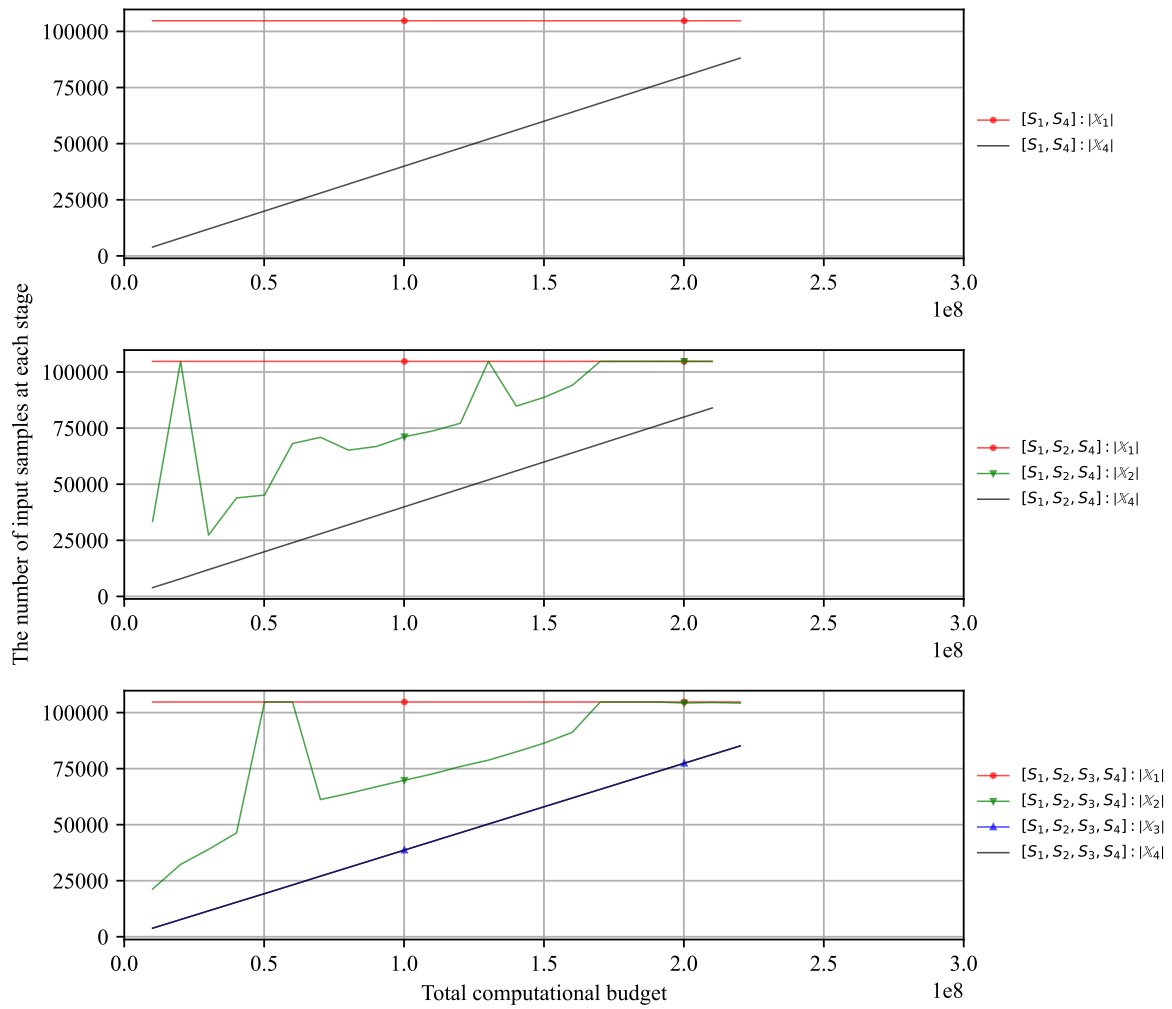


Figure S47: The number of input samples at each stage.

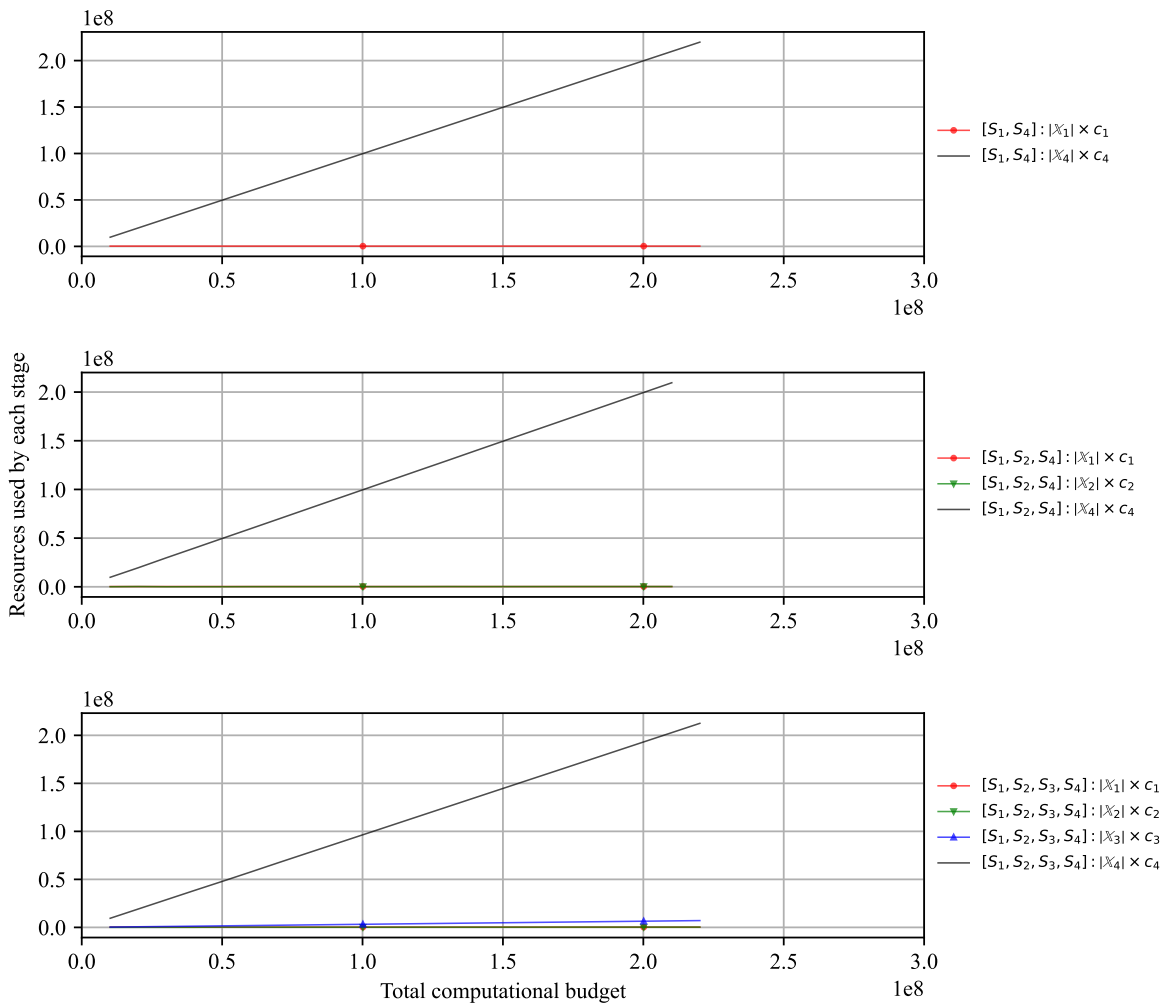


Figure S48: Resources used by each stage.