frontiers
in Bioengineering and
Biotechnology

TYPE Original Research
PUBLISHED 26 June 2026
DOI 10.3389/fbioe.2026.1802948

OPEN ACCESS

EDITED BY
Fengjie Cui,
Jiangsu University, China

REVIEWED BY
Marina de Leeuw,
Agricultural Research Organization - Volcani Institute, Israel
Arunangshu Das,
Indian Institute of Technology Delhi, India

*CORRESPONDENCE
Thomas Abeel,
T.Abeel@tudelft.nl

RECEIVED 03 February 2026
REVISED 08 May 2026
ACCEPTED 28 May 2026
PUBLISHED 26 June 2026

CITATION
Van Lent P, Paz SM, Schmitz J and Abeel T (2026) Comparing metabolic engineering scenarios using simulated design-build-test-learn-cycles.
Front. Bioeng. Biotechnol. 14:1802948.
doi: 10.3389/fbioe.2026.1802948

Comparing metabolic engineering scenarios using simulated design-build-test-learn-cycles

Paul Van Lent[1], Sara Moreno Paz[2], Joep Schmitz[2] and Thomas Abeel[1,3]*

[1]Delft Bioinformatics Lab, Intelligent Systems, Delft University of Technology, Delft, Zuid-Holland, Netherlands, [2]Department of Science and Research, dsm-firmenich, Delft, Zuid-Holland, Netherlands, [3]Infectious Disease and Microbiome Program, Broad Institute of MIT and Harvard, Cambridge, MA, United States

Introduction: Design-Build-Test-Learn (DBTL) cycles are a widely employed engineering framework in metabolic engineering. Nonetheless, their performance depends on a wide range of experimental and algorithmic design choices, whose combined effects on the successful optimization of microbial strains remain an open question.
Methods: In this study, we performed in silico DBTL cycles based on metabolic kinetic models to quantitatively assess how key process parameters affect strain optimization outcomes across four distinct metabolic pathway models. This includes parameters governing DNA library design, experimental budget limitations, and machine learning configuration.
Results: The results show that screening capacity is a dominant driver of optimization success, whereas DNA sequencing capacity has surprisingly little impact, despite its importance for model training. Selecting top-producing strains for sequencing consistently outperforms stratified sampling, highlighting a trade-off between predictive accuracy and optimization efficiency. DNA library structure strongly affects performance: increasing the number of editable positions generally improves outcomes, while expanding the set of gene targets can hinder optimization due to increased dimensionality or sparse sampling.
Discussion: Together, these findings offer actionable guidance for designing more effective DBTL workflows and underscore the value of simulation frameworks for exploring metabolic engineering strategies prior to experimental implementation.

KEYWORDS
DBTL cycles, kinetic modeling, machine learning, metabolic engineering, systems biology

1 Introduction

Metabolic engineering aims to improve microorganisms by genetically modifying their metabolic pathways to boost the synthesis of desired products Cho et al. (2022). This can be used to develop microbial cell factories that facilitate the synthesis of bulk and specialty chemicals using potentially sustainable substrates Tickner et al. (2021). However, this development from a proof-of-principle strain to a microbial cell factory is a complicated process, often taking many years of development and significant financial investment Beall et al. (2019). Reducing strain development times is essential for the widespread adoption of bioprocesses for manufacturing.

Advances in high-throughput screening and DNA sequencing technologies have allowed for the targeting of multiple metabolic pathway elements simultaneously for optimization Dietrich et al. (2010). This approach is known as combinatorial pathway optimization Jeschek et al. (2017). By targeting many elements in parallel, the likelihood of finding an optimal configuration for the production of a desired compound increases compared to the sequential de-bottlenecking of metabolic pathways. However, as the number of pathway elements targeted increases, the number of theoretical designs combinatorially explodes. To navigate this large landscape of possible designs, the Design-Build-Test-Learn (DBTL) cycle is widely used in metabolic engineering. In the Design phase, a set of designs is chosen based on the selection of targets and experimental design. This is then followed by the Build and Test phase, during which the strains are constructed and screened for their performance. The results from the screening phase are used to guide new designs in the subsequent DBTL cycle (Learn phase). This iterative approach has been shown to be successful in many applications Opgenorth et al. (2019); Hägele et al. (2025). However, due to the flexibility of the DBTL cycle framework, many open questions remain regarding how DBTL cycle process parameters affect strain optimization performance.

Machine learning (ML) has been increasingly used to propose new strain designs for subsequent DBTL cycles Radivojević et al. (2020); Yang et al. (2025); Wei et al. (2025). These ML-assisted approaches are promising for automated strain engineering, as they effectively close the loop between DBTL cycles in a data-driven manner HamediRad et al. (2019). However, the success of the recommendation strategy is closely linked to the process parameters related to the DBTL cycle. This includes experimental design choices related to DNA sequencing and screening (phenotyping) budget, DNA library design choices Jeschek et al. (2016), and exploration-exploitation tradeoffs for optimization. Often, these design choices are a consequence of experimental budget constraints, lab capabilities, and experience. Consequently, elucidating how ML-assisted optimization performance is affected by these experimental design choices is of high importance to strain engineering.

In silico experiments serve as a versatile tool for evaluating metabolic engineering scenarios that would not be practically feasible in an experimental setting Moreno-Paz et al. (2024a); van Lent et al. (2023); Roy et al. (2021). Previously, we developed simulated DBTL cycles, a framework based on kinetic models to consistently evaluate ML methods across different metabolic engineering scenarios van Lent et al. (2023). This benchmarking framework provided insights into supervised ML methods, data requirements, and budgetary constraints. Whereas numerous studies have examined the machine-learning components of recommendation strategies Wei et al. (2025); Sabzevari et al. (2022); HamediRad et al. (2019), the current work instead focuses on how the process parameters of the DBTL cycle influence the optimization performance of an ML-assisted strategy. Four hypothetical metabolic pathways of diverse topologies and complexities are implemented in a batch bioprocess kinetic model of *Saccharomyces cerevisiae*. These different pathway models are used as a test suite for comparing various metabolic engineering scenarios. The aim is to quantitatively assess the role of process parameters and their influence on strain optimization, thereby informing experimental design of DBTL cycles in industrial metabolic engineering. We offer code and workflows for user-specific simulated DBTL cycle scenarios at https://github.com/AbeelLab/BenchmarkProjectDSMF.

2 Materials and methods

2.1 Kinetic models

2.1.1 Kinetic models: implementation properties

Four bioprocess models of varying complexity and scale were constructed using the *jaxkineticmodel* Python package van Lent et al. (2025a) and subsequently exported to SBML format Bornstein et al. (2008) for standardized representation and downstream analysis. Each model describes a hypothetical metabolic pathway, integrated in a bioprocess model of *Saccharomyces cerevisiae*, that is designed to satisfy carbon balance across all reactions. Biomass formation reactions were parameterized to yield physiologically reasonable biomass yields of approximately 0.5 g biomass per Gram glucoseVan Hoek et al. (1998).

The pathway topologies used in the kinetic models were derived from structures observed in real metabolic networks (see Table 1). Simplified representations of these network topologies are provided in Supplementary Figure SI1 Hari and Lobo (2020). Pathway A is based on the p-coumaric acid pathway Rodriguez et al. (2015), while Pathway B is loosely modeled after the glutamate pathway, incorporating a cyclic structure reminiscent of the Krebs cycle Vuoristo et al. (2015). Pathway C draws inspiration from the ornithine pathway and includes acetate formation as a byproduct Qin et al. (2015). Finally, Pathway D is loosely based on the cocaine biosynthetic pathway Wang et al. (2022). It should be noted that not all these pathways are native to *S. cerevisiae*.

To account for the metabolic burden associated with genetic modifications, a protein load term was incorporated Wu et al. (2016). Metabolic burden was represented by an increase in the maximum rate of a non-growth-associated maintenance reaction that depletes ATP required for biomass synthesis. Specifically, the parameter $v_{max}^{maintenance}$ in the maintenance ATP sink reaction (Equation 1) was scaled relative to a wild-type reference strain. In the reference condition, the maximum flux is given by v_{max}^{ref}, where all pathway enzyme concentrations $[E_i]$ are normalized to one. A constant C was introduced to ensure that the resulting fraction evaluates to unity under the reference enzyme concentration profile, thereby preserving $v_{max}^{maintenance} = v_{max}^{ref}$ for the wild-type model while enabling systematic variation of metabolic burden in perturbed strains.

$$v_{max}^{maintenance} = v_{max}^{ref} \cdot \frac{\left(C + \sum_{i=1}^{N} [E_i]\right)}{100} \tag{1}$$

2.1.2 Kinetic models: summary statistics

Table 1 summarizes the properties of all pathway models, including the numbers of species, reactions, parameters, and optimization targets. The topological complexity was quantified using the approach described in Ghavasieh and De Domenico (2024) (Supplementary Figure SI4).

TABLE 1 Key statistics of the four pathway models used in this study. The network complexity is a random walk measure for the stoichiometric matrix *S*. The topology characteristic gives a brief description of the features in the model.

Model	Species	Reactions	Parameters	Enzyme targets	Network complexity	Topology characteristic
Pathway A	23	23	86	20	0.30	Linear
Pathway B	18	16	59	12	0.32	Cycle
Pathway C	25	21	82	17	0.22	Cycle + linear
Pathway D	29	25	98	21	0.61	Cycle + linear

2.2 Metabolic engineering process

Kinetic models described in the previous section are used as a surrogate model for the simulation of metabolic engineering scenarios. Below, more details are provided on how these scenarios are simulated.

2.2.1 Simulated design-build-test-learn cycles

The simulation of DBTL cycles was introduced in van Lent et al. (2023) and subsequently re-implemented using JAX Bradbury et al. (2018); van Lent et al. (2025a). The number of enzyme targets (F) and promoters (X) for a set of library positions (P) is used to construct a DNA library. This DNA library is represented as a matrix with $F \times X$ rows by P columns, where each element corresponds to a promoter-gene pair. A probability is assigned to each element such that the positions sum to one. Strains are constructed based on random assembly, according to these probabilities, so that a single strain includes P promoter-gene pairs. This approach mirrors the method used in many combinatorial pathway optimization methods involving library transformation Moreno-Paz et al. (2024b); Jeschek et al. (2017). In the first DBTL cycle, all promoter-gene pairs are equiprobable. In subsequent cycles, probabilities are adjusted using an ML-assisted approach (see next section). For all strains that are constructed (S) during the build phase, the designs are numerically simulated using Diffrax with the Kvaerno5 numerical solver Kidger (2022); Kværnø (2004). In the test phase, the built strains are simulated as a batch bioprocess without additional glucose feeding. Then, 10% homoscedastic noise is introduced. Finally, these simulated strains serve as training data for a ML model, which will be used in the next DBTL cycle.

2.2.2 Recommending new strains: an ML-assisted approach

To automate the recommendation of new strains for comparing metabolic engineering scenarios across multiple DBTL cycles, we use an ML-assisted recommendation method that outputs a probability distribution for the DNA library that will be used in the next DBTL cycle.

2.2.2.1 Machine learning model

XGBoost is trained on the input strain design data and the simulated values of the product of interest Chen et al. (2015). Default hyperparameters were used, with the exception of number of boosting rounds and early stopping conditions (*num_boosting_rounds* = 20, *early_stopping_rounds* = 40). Ten-fold cross-validation was performed to estimate model performance using the Pearson correlation coefficient.

2.2.2.2 Recommending strains using the machine learning model

The DNA library was randomly sampled during each DBTL cycle to generate a large candidate set (600,000 variants). This down-sampling is necessary, as the combinatorial design space can be prohibitively large. Samples were then evaluated using the trained XGBoost model. The predicted strain performances were expressed relative to the parent strain and subsequently converted into a probability distribution using the softmax function (Equation 2) Sutton et al. (1998). In this formulation, the parameter β controls the trade-off between exploration of the DNA library design space and exploitation of the model predictions.

$$p\left(y_{pred}\right) = \frac{e^{\beta\left(y_{pred}-y_{parent}\right)}}{\sum_{i=1}^{N} e^{\beta\left(y_{pred}-y_{parent}\right)}} \tag{2}$$

Strains predicted to outperform the parent strain are assigned higher probabilities than those predicted to underperform. To obtain probabilities for each DNA element in the library, we summed the probabilities from Equation 2 over all strains containing that element. This procedure yields element-wise (column-wise) probability distributions across the DNA library, as determined by the ML model.

2.2.2.3 Dynamic balancing exploration-exploitation

To automatically balance exploration and exploitation, we selected the softmax temperature parameter β using an entropy-based heuristic. For a given β, we computed the entropy of the softmax distribution (Equation 3) according to

$$H\left(p(y)\right)_{\beta} = \sum_{i=1}^{N} p(y)_{\beta} \log\left(p(y)_{\beta}\right), \tag{3}$$

where $p(y)_{\beta}$ denotes the softmax probabilities over the N candidates at temperature β. We evaluated $H\left(p(y)\right)_{\beta}$ over a range of temperatures ($\beta \in [0, 100]$) and identified the value of β at which $\mathrm{d}H/\mathrm{d}\beta = 0$. This point corresponds to the steepest decrease in entropy, which can be interpreted as the onset of a transition from a high-entropy (more exploratory) to a low-entropy (more exploitative) regime. The β value at this point was then used to compute the updated DNA library sampling probabilities in each iteration. This procedure was inspired by the need to avoid premature convergence to local optima in Bayesian optimization and reinforcement learning settings Sutton et al. (1998); Berger-Tal et al. (2014).

TABLE 2 Description of process parameters.

Symbol	Description	Model A	Model B	Model C	Model D
F	Number of enzyme targets	6–19	6–12	6–17	5–21
X	Number of distinct promoters	2–6	4	4	4
P	Number of positions in library	4–10	4–10	4–10	4–10
S	Number of screened strains	50–1200	50–1200	50–1200	50–1200
N	Number of DNA sequenced strains	50–200	50–200	50–200	50–200
Sampling approach	DNA sequencing selection method	Stratified or best sampling	Stratified or best sampling	Stratified or best sampling	Stratified or best sampling
β	Exploration-exploitation parameter	0–100	0–100	0–100	0–100

2.3 Metabolic engineering scenarios

To understand how the success of the DBTL cycle is impacted by design choices, the key DBTL cycle process parameters that were identified are reported in Table 2. Typical ranges for these parameters are shown, but they may vary depending on the specific pathway model (see code availability). For each model, we rank the enzyme importance using sensitivity analysis on the kinetic model Zi (2011). This serves as the ground truth ranking. Ten percent homoscedastic noise is added to the simulated production values.

Two rounds of computational experiments were conducted. In the first round, each individual process variable listed in Table 2 was evaluated for pathway model A to identify those that significantly influenced the optimization process. This is performed on one pathway model to reduce the computational load for more elaborate follow-up experiments. In the second round, informed by these results, a large combinatorial experiment was designed using the subset of variables deemed most likely to affect metabolic flux optimization performance. The variables included the number of enzyme targets (F), the number of screened strains (S), the number of sequenced strains (N), and the number of engineerable positions in the DNA library (P).

For each pathway model, all feasible combinations of these variables were generated, resulting in between 144 and 320 distinct process configurations, depending on the model. For each configuration, five simulated DBTL cycles were executed. Performance was quantified by computing the area under the curve (AUC) of the production of the best built strain over five DBTL cycles and (ii) the median production across all built strains within each cycle. Each configuration was repeated for 10 independent simulation runs to account for stochastic variability. For certain pathway models, specific variable combinations could not be simulated (e.g., due to numerical stiffness or runtime errors); these configurations were excluded from subsequent analysis.

2.4 Code availability

Code and results are available on GitHub at https://github.com/AbeelLab/BenchmarkProjectDSMF.

3 Results

3.1 Quantitative comparison of optimization processes reveals key DBTL cycle process parameters

Design-Build-Test-Learn (DBTL) cycles are widely used in metabolic engineering, helping to manage the extensive combinatorial design options of potential strain designs Jeschek et al. (2017). Despite its utility, the DBTL cycle involves many parameters that could affect the optimization of microbial strains. These include factors related to the experimental budget, experimental design, and ML-related factors. Given the substantial time and resources necessary for metabolic engineering experiments, employing simulated DBTL cycles offers a cost-efficient alternative to real-world experimentation for assessing process parameters in various scenarios Roy et al. (2021). In this research, we employ a previously developed kinetic model-based framework to assess the impact of process parameters on four distinct metabolic pathway optimization tasks van Lent et al. (2023).

As an example of how simulated DBTL cycles are used to evaluate metabolic engineering strategies, we compared an ML-assisted recommendation approach with a random DNA library transformation baseline Wei et al. (2025). Five rounds of optimization were chosen, as this is a number of DBTL cycles that could be encountered in real-world settings. In the baseline approach, each DBTL cycle involves constructing a DNA library through random integration of genetic elements into the host strain, after which the best-performing strain is selected and used as the parent for the next iteration. In contrast, the ML-assisted strategy employs a gradient bandit algorithm to iteratively update strain-selection preferences based on predicted productivity van Lent et al. (2025b); Sutton et al. (1998) (see 2), thereby increasing the likelihood of identifying higher-performing parent strains for subsequent DBTL cycles. As expected, the ML-assisted method outperforms the baseline in terms of the average production across the set of constructed strains (Figure 1A). A similar, albeit less pronounced, improvement is observed when considering the maximum-producing strain (Figure 1B). Together, these results show that ML-assisted recommendations can enhance strain performance relative to random DNA library transformation.

10.3389/fbioe.2026.1802948

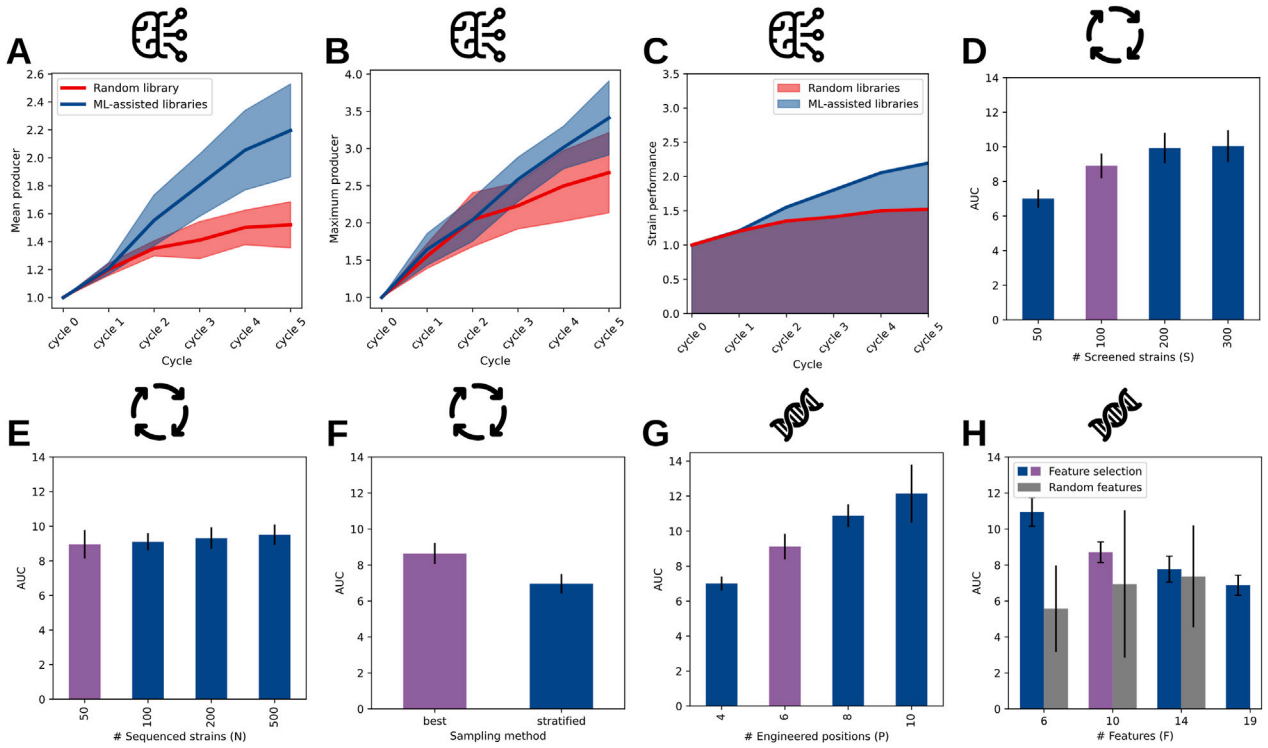

FIGURE 1
Evaluating process variables for pathway model **(A)**. In plots D-H, the purple bar represents the baseline scenario ($S = 100, N = 50, F = 10, P = 6, X = 4, \beta = 10$). Three categories of process parameters are illustrated with pictograms: parameters related to ML, those related to the DBTL cycle, and parameters concerning library design. **(A)** A comparison between a random library optimization approach and a strategy utilizing ML-assisted recommendations. The y-axis indicates the average production of the recommended strain set relative to the parent strain's reference production. **(B)** A similar graph, but depicting the top producer among the recommended strains. **(C)** To assess scenarios, optimization performance is measured using the area under the curve (AUC). **(D)** A comparison of approaches for selecting strains from screening results for DNA sequencing. The purple bar denotes the baseline scenario maintained across all plots. **(E)** Optimization performance with varying DNA sequencing capacities. **(F)** Sample selection strategies for screening: choosing the top producers (best sampling) versus a representative set (stratified sampling). **(G)** The number of DNA library positions that can be integrated into the host strain's genome. **(H)** The impact of feature selection on optimization performance, with grey bars showing random feature selection.

Relating to DBTL cycle parameters, the impacts of screening capacity (S), DNA sequencing capacity (N), and screening-to-sequencing strain selection are evaluated in terms of the area under the curve (Figure 1C). Increasing screening capacity consistently improved optimization performance (Figure 1D). This improvement likely occurs because larger screens increase the probability of including top-producing strains in the training data. However, the improvement between screening 200 strains and 300 strains was relatively minor, suggesting that for this pathway, the sampling density with 200 samples was enough to find high-performance strains. Interestingly, increasing DNA sequencing capacity had little effect on optimization performance (Figure 1E), despite the expectation that providing more data to the ML model would enhance predictive accuracy and consequentially a better optimization performance.

We further compared two strain-selection strategies for sequencing: (i) selecting the top producers identified in screening (best sampling) and (ii) selecting a stratified set that spans the observed production range (stratified sampling). These strategies reflect a trade-off when choosing strains for further DNA sequencing: choosing the best screened strains is desirable from a metabolic flux optimization perspective, whereas a stratified approach is more desirable when aiming to reduce training data bias (and thereby predictive performance). From an optimization perspective, it is clear that the best sampling approach led to a notably better optimization performance compared to stratified sampling (Figure 1F). In the case of stratified sampling, there is a chance that the best strain is not used as the new parent strain, which may result in worse optimization performance.

Finally, we considered parameters associated with the DNA library, including the number of positions (P) and the number of gene targets (F). Findings suggest a significant effect of the DNA library positions on optimization performance by enhancing the exploration of the combinatorial design space (Figure 1G). Although increasing the number of library positions might currently be technically challenging for strain engineering, these results underscore the potential benefits of developing synthetic biology tools that enable the integration of more genetic elements. Regarding the choice of the number of gene targets considered in the DNA library, reducing the number of targets significantly improves the optimization efficiency of ML-assisted recommendation (1H). However, this is only the case when the gene targets are chosen carefully, as selecting the wrong gene targets can result in worse performance than if no selection is performed

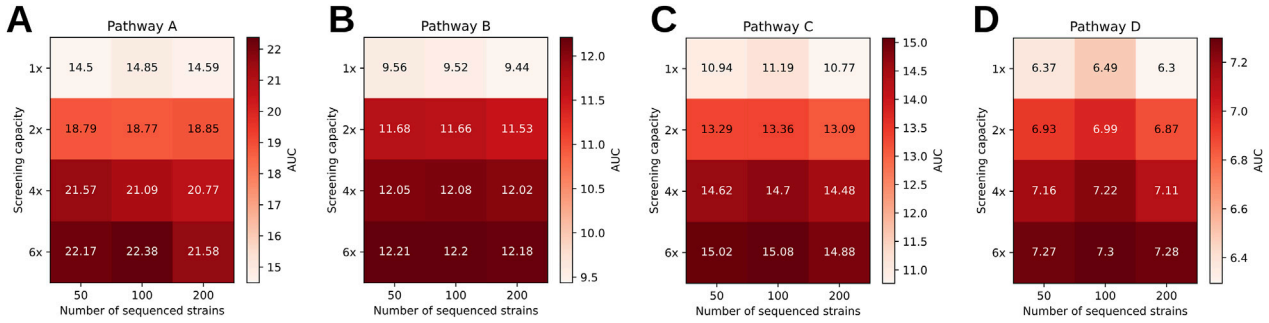

FIGURE 2
Screening capacity has a strong impact on ML-assisted optimization performance. **(A)** The optimization problem of pathway model A-C had 6 gene targets ($F = 6$), while pathway model D consisted of five targeted genes ($F = 5$). Six library positions were considered for all models ($P = 6$). Screening capacity has a strong impact on the area under the curve performance, while DNA sequencing has showed little effect on the optimization performance. **(B–D)** Similar heatmap plots for pathway model B-D.

(shown in gray). Consequently, feature selection methods, whether data-driven or grounded in mechanistic understanding, are crucial for improving metabolic engineering Van Rosmalen et al. (2024); Alonso-Gutierrez et al. (2015).

In summary, the results on one metabolic pathway indicate that optimization performance is impacted by screening capacity (S), the strain selection strategy for further DNA sequencing selection, the number of library positions (P), and the number of gene targets (F), whereas the number of sequenced strains (N) does not appear to impact performance. Effects of other process parameters, including promoter strengths, exploration-exploitation trade-offs in ML-assisted recommendation, and the number of promoters included per gene, are reported in SI 3.

3.2 Screening capacity, not sequencing capacity, is a key driver of optimization performance

Although the first section identified some important process parameters, this was only performed on one metabolic pathway optimization problem. We therefore further substantiate our analysis of these identified parameters to include three additional metabolic pathways with varying complexities (see Materials and methods).

Results found on pathway model A showed that increasing screening capacity greatly improves optimization, whereas increasing DNA sequencing capacity does not. To validate this finding, a grid search is performed for these two parameters. Figure 2 presents four heatmaps for the different metabolic pathway models. The screening ratio indicates the down-sampling factor after screening. For instance, in a scenario involving 200 DNA-sequenced strains, a screening ratio of six leads to 1200 simulated strains. Across all models, an increase in the screening ratio correlates with enhanced AUC performance. This reinforces the result that the number of DNA-sequenced strains appears to have minimal effect on performance (Figure 1E). While the overall performance differs between metabolic pathways, the independence of performance from DNA sequencing capabilities is generally observed. This is somewhat unexpected since the ML model is exclusively trained on this selection. It is often assumed that a larger training dataset would yield a superior model. In terms of optimization, the emphasis lies in identifying high-producing strains, which becomes more feasible with enhanced screening capabilities. In industrial metabolic engineering, the screening throughput of workflows may therefore represent a more significant bottleneck than DNA sequencing capacity.

3.3 Sampling top producers for screenings outperforms stratified approaches for metabolic flux optimization

Regression models in ML are susceptible to data imbalances stemming from selection biases, as noted in studies Yang et al. (2021); Tepeli et al. (2024); Steininger et al. (2021). In the field of metabolic engineering, these biases are often introduced when selecting which strains to sequence for further study. Ideally, from a metabolic flux optimization standpoint, one would select the highest-performing strains, even though they may not represent the underlying genotypic distribution of the DNA library well. To assess the implications of this selection, we compared the best sampling strategy (selecting top producers) with a stratified sampling method (see Materials and methods).

Across all pathway models, the selection of top producers consistently yields a higher AUC compared to using the stratified method (Figures 3A–D). Notably, in metabolic pathway model D, after completing two cycles, the performance of the strain actually diminishes (Figure 3D). This decline is attributable to the protein load effect accounted for in the kinetic models; as additional enzyme copies are included, biomass flux decreases due to increased maintenance (Supplementary Figure SI2). Nevertheless, selecting top producers in all DBTL cycles proves more effective than the stratified approach. Conversely, the performance of the ML model shows a slightly different trend. In pathway models A and B, the Pearson correlation coefficient on cross-validation sets is actually lower when using the best sampling method compared to the stratified method (Figures 3E,F), which aligns with expectations for a biased distribution Steininger et al. (2021). In contrast, for pathway models C and D, no differences are observed between the two selection methods (Figures 3G,H), potentially because the chosen strains more accurately reflect the underlying distribution.

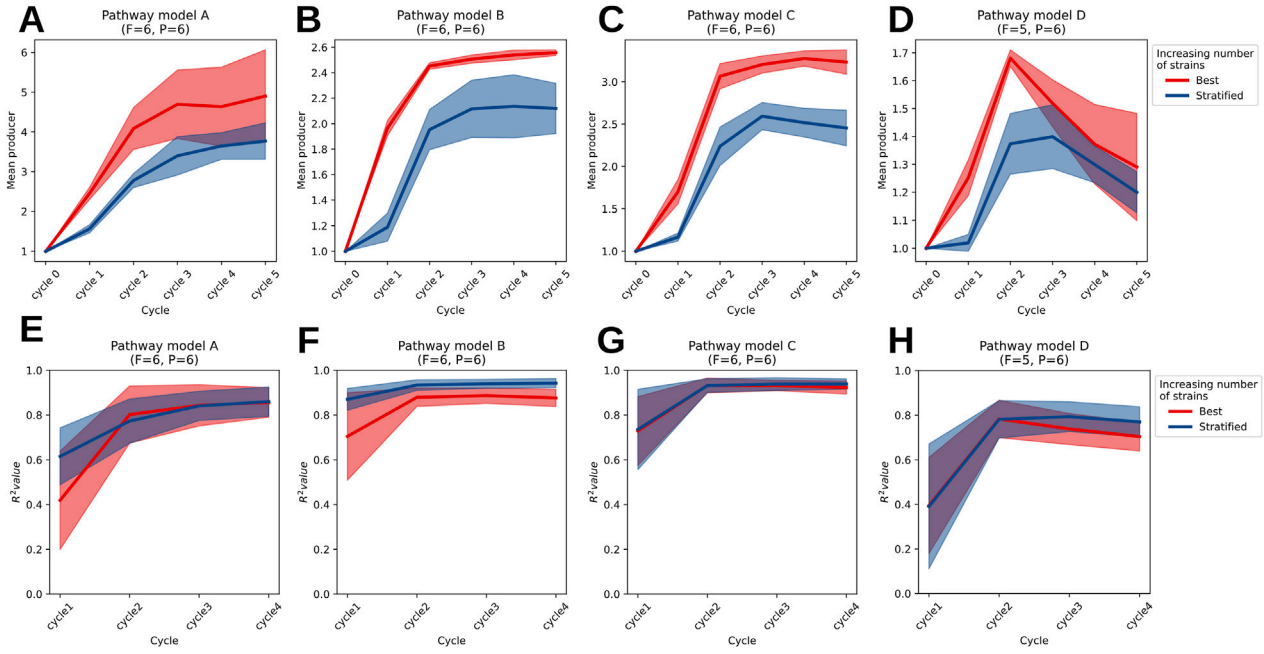

FIGURE 3
Selection method of strains for DNA sequencing **(A–D)** The optimization result for the pathway models. The stratified approach selects uniformly across the range of production values, while the best sampling approach selects the top producers (see Materials and methods). In general, product optimization favors a best sampling approach. **(E–H)** The Pearson correlation coefficient R^2 of the XGBoost model. A higher performance is typically observed for the stratified sampling approach.

To summarize, these results highlight a common trade-off in metabolic engineering. While a stratified selection approach improves the predictive accuracy of the XGBoost model, a bias towards selecting high-producing strains can result in better overall performance in strain optimization. In terms of metabolic flux optimization, sampling the best strains outperforms the stratified approach and should be preferred. However, alternative selection strategies that consider both aspects may help to mitigate this trade-off.

3.4 DNA library parameters have mixed impact on strain optimization performance

In Figures 1G,H, it was noted that parameters concerning the DNA library design have a considerable impact on performance. For a DNA library composed of a set number of positions (P), each capable of accommodating different promoter-gene-terminator combinations ($X \times F$), the theoretical space of combinatorial designs is represented by $(F \times X)^P$ strains. Although enlarging the DNA library size can enhance genetic variation among strains, it also results in a combinatorial explosion of the design space Jeschek et al. (2017). In this context, we aim to understand the effect of the number of included gene targets (F) and the number of positions (P) for the four pathway models.

We first evaluated how the number of gene targets in the library influenced performance across the four pathway models (Figure 4A). Increasing the number of gene targets negatively affected performance for models A and C, whereas models B and D were mostly unaffected. This mixed effect might be explained by two reasons. First, as the dimensionality of the design space increases, the associated optimization problem becomes more challenging. Problems defined over larger gene sets are inherently more difficult to solve, as can be observed in models A and C Frazier (2018); Del Giudice (2021). Second, because each strain contains only six genes ($P = 6$), adding more candidate gene targets beyond this limit leads to sparser coverage of the combinatorial design space. This results in a less informative training dataset.

The effect of increasing the number of library positions is shown in Figure 4B. For pathway models A and C, enhancing the number of library positions leads to a notable improvement in the AUC. Conversely, for pathway model B, such an increase does not result in any noticeable effect, indicating that only a limited number of genes may be crucial for performance enhancement. Interestingly, for pathway model D, the increased number of positions has a negative effect. This is likely due to the more prominent effect of protein load in pathway model D compared to the other pathway models (see Supplementary Figure SI2). Introducing more gene copies reduces biomass production, thus hindering performance. This shows that while an expansion of engineering capabilities is potentially beneficial, it may depend on the specific pathway to be optimized.

The design of libraries is crucial in combinatorial pathway optimization Jeschek et al. (2017). Proposed strategies for DNA library design aim to balance genetic variation while preventing combinatorial explosion Jeschek et al. (2016). Here, we demonstrate that choosing appropriate genes and limiting the number of genes considered can enhance performance. Furthermore, the expansion of the number of DNA positions affects the area under the specific problem being addressed.

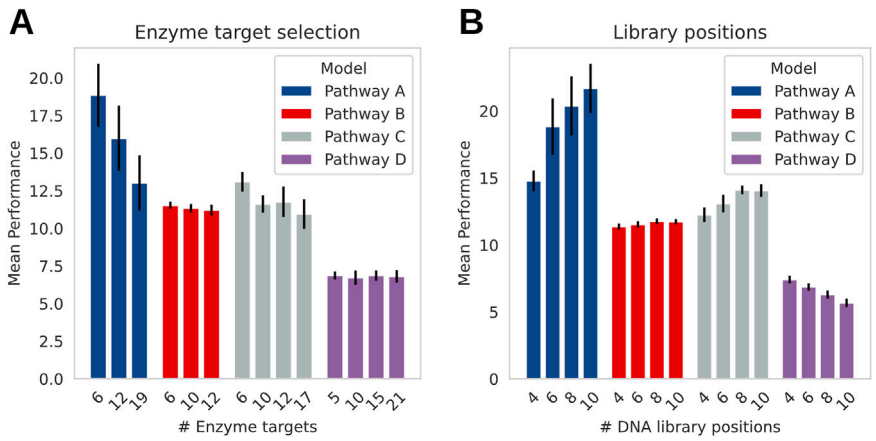

FIGURE 4
Library design: effect of feature selection and number of positions on optimization performance. **(A)** Number of enzyme targets utilized for optimization in each pathway model. Enzyme targets were selected based on sensitivity analysis of the kinetic model Zi (2011). In general, fewer enzyme targets are associated with enhanced optimization performance. **(B)** The number of DNA library positions that can be incorporated into a single strain. Typically, increasing the number of positions leads to improved performance, although it depends on the specific optimization issue (see Discussion).

4 Discussion

The DBTL cycle is a flexible and widely adopted framework for navigating the landscape of strain design options in metabolic flux optimization. In this study, we investigate how optimization performance depends on parameters associated with DNA library design and the DBTL cycle when using ML-assisted recommendation methods Moreno-Paz et al. (2024b); Pandi et al. (2022); Wei et al. (2025). To accomplish this, we employed simulated DBTL cycles, which offer a cost-effective alternative to real-world experimentation van Lent et al. (2023).

Our analysis began by identifying seven key parameters related to the DBTL cycle (see 2, specifically in the Design phase (number of features and positions), the Build and Test phase (screening capacity, sequencing capacity, and sampling approach), and the Learn phase (the exploration-exploitation trade-off). These were tested on one pathway model to determine which parameters had a significant effect on optimization performance and should be further investigated (see Figure 1; Supplementary Figure SI3). This showed that screening capacity, the sampling approach for DNA sequencing, the number of positions in the library, and the number of features had the most prominent effect on performance.

An interesting finding when expanding the analysis to multiple pathways is that screening capacity, but not DNA sequencing capacity, was a major driver of optimization performance (Figure 2). Although increasing the number of samples would be expected to improve predictive accuracy, this does not necessarily translate into improved recommendations. Here, we distinguish between predictive performance, defined as the model's ability to accurately estimate outcomes across the design space, and recommendation performance, which reflects the model's ability to identify and prioritize high-performing candidates. We hypothesize that this discrepancy arises because increased screening capacity raises the likelihood of observing high-performing strains. As a result, the training data become enriched in regions of the design space most relevant for optimization, enabling the model to better distinguish and prioritize top-performing designs, even if overall predictive accuracy improves only modestly. This may also explain why sampling the best strains for DNA sequencing is a better strategy than the stratified approach, as the ML model observes more high-performing strains in this scenario. In this regard, expanding screening capacity can enhance the predictive performance of ML models, as it enlarges the domain and range of data on which the model is trained.

For gene target selection, enzyme sensitivities with respect to the product were employed to rank and select the most important enzymes (see Materials and methods). In reality, performing gene target selection can be difficult, as it is not *a priori* known which genes affect production. This can have substantial consequences on metabolic engineering success (Figure 1H). To address this, genome-scale modeling approaches could be used to choose gene targets Van Rosmalen et al. (2024); Domenzain et al. (2025). Expanding this analysis to other pathways indeed verifies that performance is positively impacted by feature selection, although the impact may differ between products (Figure 4A).

For the number of library positions, we found that for pathway model D, the optimization performance decreases, in contrast to the other models (Figure 4B). This is likely an attribute of the network topology of pathway model D, which had a byproduct route that might be favored over the product route (see SI 2). Furthermore, it could also be a consequence of performing library transformation. As more gene copies are added, the protein load increasingly affects biomass formation, resulting in decreased production. This can also be observed in Figure 3D, where production drops after three DBTL cycles.

In the present study, our optimization process focused on the addition of individual gene copies to a host strain. While this approach is straightforward and experimentally tractable, it does not capture the full range of possible metabolic engineering interventions. In particular, strategies involving gene downregulation or knockouts could provide valuable alternatives,

as demonstrated by approaches such as OptKnock Burgard et al. (2003), especially in systems where metabolic burden or metabolic regulation principles are a limiting factor. Incorporating such interventions would allow for a more comprehensive exploration of the design space. However, implementing downregulation typically requires the replacement or modification of native promoters, which introduces additional experimental complexity compared to gene over-expression. For this reason, these strategies were not included in the present study. Furthermore, although our work focused on yeast, other host organisms may exhibit polycistronic gene organization, which could influence optimization outcomes. Finally, although the implemented pathways were topologically diverse, metabolic regulation principles such as product inhibition were not included. While these scenarios fall outside the scope of this study, they could be addressed in future work using similar modeling approaches.

Balancing exploration and exploitation in machine-learning-driven recommendations within the DBTL cycle is known to be important Zhang et al. (2020); van Lent et al. (2025b); McInerney et al. (2018). In Figure 1, we fixed β to a single value to enable consistent comparisons between scenarios. In practice, the preferred exploration–exploitation balance may shift over the course of a combinatorial pathway optimization experiment. For instance, one might prioritize exploration in the early DBTL cycles and then gradually increase exploitation in later cycles. To avoid searching for this exploration-exploitation schedule, we applied an entropy-based approach to adaptively manage this trade-off (see Materials and methods). We observed that this strategy resulted only in a minor improvement over a static approach (Fig. 4.0.3B). Nonetheless, alternative strategies for controlling exploration versus exploitation are conceivable and may be worth exploring in future work.

Overall, our results highlight practical considerations that can be used to improve the DBTL cycle for real-world metabolic engineering. As the variety of metabolic engineering scenarios that can be considered is abundant, workable templates for running alternative scenarios of the DBTL cycle are provided.

Data availability statement

The datasets generated can be found in the 4tu repository (doi: 10.4121/f2833f74-2586-4632-8c44-7d98ead11a34). Scripts and workflows can be found on the github page https://github.com/AbeelLab/BenchmarkProjectDSMF.

Author contributions

PV: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review and editing. SP: Conceptualization, Investigation, Methodology, Supervision, Writing – review and editing. JS: Conceptualization, Investigation, Methodology, Project administration, Supervision, Validation, Writing – review and editing. TA: Conceptualization, Investigation, Methodology, Project administration, Supervision, Validation, Writing – review and editing.

Funding

The author(s) declared that financial support was received for this work and/or its publication. This work is supported by the AI4b.io program (to P.H.van Lent) (https://www.ai4b.io/), a collaboration between TU Delft and dsm-firmenich, and is fully funded by dsm-firmenich (https://www.dsm-firmenich.com/en/home.html) and the RVO (Rijksdienst voor Ondernemend Nederland) (https://www.rvo.nl/). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Conflict of interest

Authors SP and JS were employed by dsm-firmenich.

The remaining author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declared that generative AI was used in the creation of this manuscript. To improve manuscript readability.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fbioe.2026.1802948/full#supplementary-material

References

Alonso-Gutierrez, J., Kim, E.-M., Batth, T. S., Cho, N., Hu, Q., Chan, L. J. G., et al. (2015). Principal component analysis of proteomics (pcap) as a tool to direct metabolic engineering. *Metab. Engineering* 28, 123–133. doi:10.1016/j.ymben.2014.11.011

Beall, R. F., Hwang, T. J., and Kesselheim, A. S. (2019). Pre-market development times for biologic versus small-molecule drugs. *Nat. Biotechnol.* 37, 708–711. doi:10.1038/s41587-019-0175-2

Berger-Tal, O., Nathan, J., Meron, E., and Saltz, D. (2014). The exploration-exploitation dilemma: a multidisciplinary framework. *PloS One* 9, e95693. doi:10.1371/journal.pone.0095693

Bornstein, B. J., Keating, S. M., Jouraku, A., and Hucka, M. (2008). Libsbml: an api library for sbml. *Bioinformatics* 24, 880–881. doi:10.1093/bioinformatics/btn051

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., et al. (2018). *Jax: Composable Transformations of Python+ Numpy Programs*.

Burgard, A. P., Pharkya, P., and Maranas, C. D. (2003). Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol. Bioengineering* 84, 647–657. doi:10.1002/bit.10803

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., et al. (2015). Xgboost: extreme gradient boosting. *R. Package Version 0* (1), 1–4.

Cho, J. S., Kim, G. B., Eun, H., Moon, C. W., and Lee, S. Y. (2022). Designing microbial cell factories for the production of chemicals. *Jacs Au* 2, 1781–1799. doi:10.1021/jacsau.2c00344

Del Giudice, M. (2021). Effective dimensionality: a tutorial. *Multivar. Behavioral Research* 56, 527–542. doi:10.1080/00273171.2020.1743631

Dietrich, J. A., McKee, A. E., and Keasling, J. D. (2010). High-throughput metabolic engineering: advances in small-molecule screening and selection. *Annu. Review Biochemistry* 79, 563–590. doi:10.1146/annurev-biochem-062608-095938

Domenzain, I., Lu, Y., Wang, H., Shi, J., Lu, H., and Nielsen, J. (2025). Computational biology predicts metabolic engineering targets for increased production of 103 valuable chemicals in yeast. *Proc. Natl. Acad. Sci.* 122, e2417322122. doi:10.1073/pnas.2417322122

Frazier, P. I. (2018). A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*. doi:10.48550/arXiv.1807.02811

Ghavasieh, A., and De Domenico, M. (2024). Diversity of information pathways drives sparsity in real-world networks. *Nat. Phys.* 20, 512–519. doi:10.1038/s41567-023-02330-x

Hägele, L., Trachtmann, N., and Takors, R. (2025). The knowledge driven dbtl cycle provides mechanistic insights while optimising dopamine production in escherichia coli. *Microb. Cell Factories* 24, 111. doi:10.1186/s12934-025-02729-6

HamediRad, M., Chao, R., Weisberg, S., Lian, J., Sinha, S., and Zhao, H. (2019). Towards a fully automated algorithm driven platform for biosystems design. *Nat. Communications* 10, 5150. doi:10.1038/s41467-019-13189-z

Hari, A., and Lobo, D. (2020). Fluxer: a web application to compute, analyze and visualize genome-scale metabolic flux networks. *Nucleic Acids Research* 48, W427–W435. doi:10.1093/nar/gkaa409

Jeschek, M., Gerngross, D., and Panke, S. (2016). Rationally reduced libraries for combinatorial pathway optimization minimizing experimental effort. *Nat. Communications* 7, 11163. doi:10.1038/ncomms11163

Jeschek, M., Gerngross, D., and Panke, S. (2017). Combinatorial pathway optimization for streamlined metabolic engineering. *Curr. Opinion Biotechnology* 47, 142–151. doi:10.1016/j.copbio.2017.06.014

Kidger, P. (2022). On neural differential equations. *arXiv Preprint arXiv:2202.02435*. doi:10.48550/arXiv.2202.02435

Kværnø, A. (2004). Singly diagonally implicit runge–kutta methods with an explicit first stage. *BIT Numer. Math.* 44, 489–502. doi:10.1023/b:bitn.0000046811.70614.38

McInerney, J., Lacker, B., Hansen, S., Higley, K., Bouchard, H., Gruson, A., et al. (2018). "Explore, exploit, and explain: personalizing explainable recommendations with bandits," in Proceedings of the 12th ACM conference on recommender systems, 31–39.

Moreno-Paz, S., Schmitz, J., and Suarez-Diez, M. (2024a). *In silico* analysis of design of experiment methods for metabolic pathway optimization. *Comput. Struct. Biotechnol. J.* 23, 1959–1967. doi:10.1016/j.csbj.2024.04.062

Moreno-Paz, S., Van der Hoek, R., Eliana, E., Zwartjens, P., Gosiewska, S., Martins dos Santos, V. A., et al. (2024b). Machine learning-guided optimization of p-coumaric acid production in yeast. *ACS Synthetic Biology* 13, 1312–1322. doi:10.1021/acssynbio.4c00035

Opgenorth, P., Costello, Z., Okada, T., Goyal, G., Chen, Y., Gin, J., et al. (2019). Lessons from two design–build–test–learn cycles of dodecanol production in escherichia coli aided by machine learning. *ACS Synthetic Biology* 8, 1337–1351. doi:10.1021/acssynbio.9b00020

Pandi, A., Diehl, C., Yazdizadeh Kharrazi, A., Scholz, S. A., Bobkova, E., Faure, L., et al. (2022). A versatile active learning workflow for optimization of genetic and metabolic networks. *Nat. Commun.* 13, 3876. doi:10.1038/s41467-022-31245-z

Qin, J., Zhou, Y. J., Krivoruchko, A., Huang, M., Liu, L., Khoomrung, S., et al. (2015). Modular pathway rewiring of saccharomyces cerevisiae enables high-level production of l-ornithine. *Nat. Commun.* 6, 8224. doi:10.1038/ncomms9224

Radivojević, T., Costello, Z., Workman, K., and Garcia Martin, H. (2020). A machine learning automated recommendation tool for synthetic biology. *Nat. Communications* 11, 4879. doi:10.1038/s41467-020-18008-4

Rodriguez, A., Kildegaard, K. R., Li, M., Borodina, I., and Nielsen, J. (2015). Establishment of a yeast platform strain for production of p-coumaric acid through metabolic engineering of aromatic amino acid biosynthesis. *Metab. Engineering* 31, 181–188. doi:10.1016/j.ymben.2015.08.003

Roy, S., Radivojevic, T., Forrer, M., Marti, J. M., Jonnalagadda, V., Backman, T., et al. (2021). Multiomics data collection, visualization, and utilization for guiding metabolic engineering. *Front. Bioengineering Biotechnology* 9, 612893. doi:10.3389/fbioe.2021.612893

Sabzevari, M., Szedmak, S., Penttilä, M., Jouhten, P., and Rousu, J. (2022). Strain design optimization using reinforcement learning. *PLoS Computational Biology* 18, e1010177. doi:10.1371/journal.pcbi.1010177

Steininger, M., Kobs, K., Davidson, P., Krause, A., and Hotho, A. (2021). Density-based weighting for imbalanced regression. *Mach. Learn.* 110, 2187–2211. doi:10.1007/s10994-021-06023-5

Sutton, R. S., and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*, Cambridge: MIT press 1, 9–11.

Tepeli, Y. I., de Wolf, M., and Gonçalves, J. P. (2024). Metric-dst: mitigating selection bias through diversity-guided semi-supervised metric learning. *arXiv Preprint arXiv:2411.18442*. doi:10.48550/arXiv.2411.18442

Tickner, J., Geiser, K., and Baima, S. (2021). Transitioning the chemical industry: the case for addressing the climate, toxics, and plastics crises. *Environ. Sci. Policy Sustain. Dev.* 63, 4–15. doi:10.1080/00139157.2021.1979857

Van Hoek, P., Van Dijken, J. P., and Pronk, J. T. (1998). Effect of specific growth rate on fermentative capacity of baker's yeast. *Appl. Environmental Microbiology* 64, 4226–4233. doi:10.1128/AEM.64.11.4226-4233.1998

van Lent, P., Schmitz, J., and Abeel, T. (2023). Simulated design–build–test–learn cycles for consistent comparison of machine learning methods in metabolic engineering. *ACS Synth. Biol.* 12, 2588–2599. doi:10.1021/acssynbio.3c00186

van Lent, P., Bunkova, O., Magyar, B., Planken, L., Schmitz, J., and Abeel, T. (2025a). Jaxkineticmodel: neural ordinary differential equations inspired parameterization of kinetic models. *PLOS Comput. Biol.* 21, e1012733. doi:10.1371/journal.pcbi.1012733

van Lent, P., van der Hoek, R., Paz, S. M., Kooi, I., Jonkers, M., Zwartjens, P., et al. (2025b). Machine learning-assisted pathway optimization in large combinatorial design spaces: a p-coumaric acid case study. *bioRxiv*, doi:10.1101/2025.06.13.659482

Van Rosmalen, R., Moreno-Paz, S., Duman-Özdamar, Z., and Suarez-Diez, M. (2024). Cfsa: comparative flux sampling analysis as a guide for strain design. *Metab. Eng. Commun.* 19, e00244. doi:10.1016/j.mec.2024.e00244

Vuoristo, K. S., Mars, A. E., Sangra, J. V., Springer, J., Eggink, G., Sanders, J. P., et al. (2015). Metabolic engineering of the mixed-acid fermentation pathway of escherichia coli for anaerobic production of glutamate and itaconate. *Amb. Express* 5, 61. doi:10.1186/s13568-015-0147-y

Wang, Y.-J., Huang, J.-P., Tian, T., Yan, Y., Chen, Y., Yang, J., et al. (2022). Discovery and engineering of the cocaine biosynthetic pathway. *J. Am. Chem. Soc.* 144, 22000–22007. doi:10.1021/jacs.2c09091

Wei, Y., Peng, B., Xie, R., Chen, Y., Qin, Y., Wen, P., et al. (2025). Deep active optimization for complex systems. *Nat. Comput. Sci.* 5, 1–12. doi:10.1038/s43588-025-00858-x

Wu, G., Yan, Q., Jones, J. A., Tang, Y. J., Fong, S. S., and Koffas, M. A. (2016). Metabolic burden: cornerstones in synthetic biology and metabolic engineering applications. *Trends Biotechnology* 34, 652–664. doi:10.1016/j.tibtech.2016.02.010

Yang, Y., Zha, K., Chen, Y., Wang, H., and Katabi, D. (2021). "Delving into deep imbalanced regression," in *International conference on machine learning* (PMLR), 11842–11851.

Yang, J., Lal, R. G., Bowden, J. C., Astudillo, R., Hameedi, M. A., Kaur, S., et al. (2025). Active learning-assisted directed evolution. *Nat. Commun.* 16, 714. doi:10.1038/s41467-025-55987-8

Zhang, J., Petersen, S. D., Radivojevic, T., Ramirez, A., Pérez-Manríquez, A., Abeliuk, E., et al. (2020). Combining mechanistic and machine learning models for predictive engineering and optimization of tryptophan metabolism. *Nat. Commun.* 11, 4880. doi:10.1038/s41467-020-17910-1

Zi, Z. (2011). Sensitivity analysis approaches applied to systems biology models. *IET Systems Biology* 5, 336–346. doi:10.1049/iet-syb.2011.0015