ACS Synthetic Biology
Research Article
Cite This: ACS Synth. Biol. 2020, 9, 157−168
pubs.acs.org/synthbio

Reinforcement Learning for Bioretrosynthesis

Mathilde Koch,[†] Thomas Duigou,[†] and Jean-Loup Faulon*[,†,‡,§]

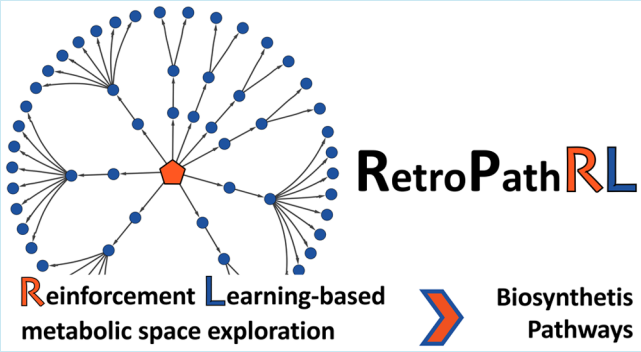
[†]Micalis Institute, INRA, AgroParisTech, Université Paris-Saclay, 78350 Jouy-en-Josas, France

[‡]iSSB Laboratory, Génomique Métabolique, Genoscope, Institut François Jacob, CEA, CNRS, Univ Evry, Université Paris-Saclay, 91057 Evry, France

[§]SYNBIOCHEM Center, School of Chemistry, University of Manchester, Manchester M13 9PL, U.K.

Supporting Information

ABSTRACT: Metabolic engineering aims to produce chemicals of interest from living organisms, to advance toward greener chemistry. Despite efforts, the research and development process is still long and costly, and efficient computational design tools are required to explore the chemical biosynthetic space. Here, we propose to explore the bioretrosynthesis space using an artificial intelligence based approach relying on the Monte Carlo Tree Search reinforcement learning method, guided by chemical similarity. We implement this method in RetroPath RL, an open-source and modular command line tool. We validate it on a golden data set of 20 manually curated experimental pathways as well as on a larger data set of 152 successful metabolic engineering projects. Moreover, we provide a novel feature that suggests potential media supplements to complement the enzymatic synthesis plan.



KEYWORDS: *retrosynthesis, reinforcement learning, Monte Carlo Tree Search, pathway design, metabolic engineering*

Efficient computational tools are required for metabolic engineering to achieve its true potential as a game-changer in the bioeconomy. Such tools include pathway design software to assist the metabolic engineer in finding new pathways for production of valuable chemicals. While some tools restrict themselves to reactions already present in databases,[1,2] others allow the generation of *de novo* reactions, using retrosynthesis algorithms.[3−11] At its core, a retrosynthesis algorithm is simple: break down a target molecule into simpler molecules that can be combined chemically or enzymatically to produce it and iterate recursively until all required compounds are either commercially available or present in the microbial strain of choice. Current bioretrosynthesis tools suffer from several limits. First of all, they are usually accessible through a Web server and not open-source, limiting an expert user's capacity to install them locally and tune them. Second, a number of parameters are often included within the pathway search, decided by the software designer and with limited capacity for a user to incorporate his own knowledge, solving for both retrosynthesis and parameters optimization. Some examples include enzyme performance,[11] predicted yield,[6−8,12,13] thermodynamics or cofactor usage.[10] Moreover, those tools do not include the latest advances in combinatorial search space exploration, pioneered in the field of artificial intelligence.

To address those limitations, we propose to make use of the Monte Carlo Tree Search (MCTS) reinforcement learning algorithm which has already revolutionized the field of artificial intelligence, as illustrated by the stunning victory of Google's AI (AlphaGo) against a Go master in 2016.[14−16] An interesting application used this algorithm combined with neural networks in chemical retrosynthesis, but acknowledging that natural compounds synthesis was beyond their scope.[17] To make available the algorithm to the wider audience, we implement it as RetroPath RL, an open source python package freely available on GitHub. RetroPath RL provides a programmatic access to discover biosynthesis pathways using MCTS, while allowing a number of augmentations and features to be used or developed by expert users to tune it to their needs. Lack of open-source computer-assisted pathway design tools is currently pointed out as one of the major limits faced by metabolic engineering projects.[18,19] Therefore, RetroPath RL is a timely software that will hopefully contribute to metabolic engineering realizing its true potential for green chemistry.

We propose first an explanation of the theoretical background giving context to the present work.

RESULTS AND DISCUSSION

Reaction Rules for Representing Enzymatic Reactions. Reaction rules describe the changes in bonding patterns when a set of substrates is transformed into a set of products in order to encode enzymatic reactions for retrosynthesis. An important

Received: November 4, 2019
Published: December 16, 2019

Downloaded via UNIV OF MANCHESTER on February 6, 2020 at 10:49:23 (UTC).
See https://pubs.acs.org/sharingguidelines for options on how to legitimately share published articles.

© 2019 American Chemical Society

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

feature of rules for retrosynthetic applications is that they need to be generalizable, so that they can be applied to a new substrate that was not from among the substrates the rules were learned on. Moreover, using generalized reaction rules is the first step toward predicting promiscuous reactions, as those reactions are often missing from metabolic databases. Modeling promiscuity is a key feature in metabolic engineering, as it has for example been estimated that 37% of *Escherichia coli* K12 enzymes have a promiscuous activity on structurally similar substrates.[20]

In our data-driven approach, we learn rules at various levels of specificity around the reaction present in the database, by keeping in the described pattern of the rule a varying number of atoms around the reaction center. We select those atoms using a number we call diameter that represents the distance in bonds around the reaction center: a rule at diameter 2 will include atoms at a distance of 1 around the reaction center, while a rule a diameter 10 will include atoms at a distance of 5 around the reaction center. Therefore, the rule at diameter 2 can apply to more diverse substrates and therefore encode more promiscuity than the rule at diameter 10. A more detailed description of reaction rules can be found in the literature[11,21] and in the Materials and Methods section.

Necessity of Ranking Reactions. The number of rules that can be applied greatly varies depending on the level of promiscuity considered and the chemical structure of the substrate. If this number is low enough, an exhaustive search can be considered, but otherwise, applying chemical reaction rules iteratively on substrates and their products leads to a combinatorial explosion. Encoding of reaction rules at various diameters allows selection of the degree of promiscuity, which has a high impact on those statistics. As an example, results for the rule sets used in this study are presented in Table 1, and for individual diameters in Supplementary Table S1. We can see from this table that the branching factor (the average number of rules that apply to a substrate) is around 100 when using rules at diameters 6, 10, and 16 (slightly promiscuous, medium, and very specific) and drastically increases with the level of promiscuity (adding rules at diameter 2 that are highly promiscuous lead to a branching factor of 900). Therefore, the more promiscuity we allow, the higher our branching factor becomes.

Table 1. Average Number of Applicable Rules on a Compound According to the Set of Diameters Used[a]

set of diameters used	6, 10, 16	2, 6, 10, 16	2, 4, 6, 8, 10, 12, 14, 16
average number of applicable rules	100	898	1183

[a]Information on average number of rules at all individual diameters are available in Supplementary Table S1.

Such a high branching factor is comparable to the Go game (branching factor of 250) and much higher than chess (35), and the reason why using algorithms that were successful in this domain could also be of interest for retrosynthesis.[14,17] We therefore use an algorithm (MCTS) that can effectively handle this combinatorial explosion, and a heuristic (chemical similarity) to guide the search.

Chemical Similarity and Sequence Availability for Reaction Ranking. Chemical similarity between query (applied on a new substrate) and the native chemical transformation has been used in various studies.[8,12,22,23] We adapted the strategy from Coley *et al.* that proceeds in a 2-step evaluation of the reaction. In a first step, before rule application, similarity between query and native substrates is calculated. After rule application, similarity between native and query products is also calculated. This allows accounting of similarity in a manner straightforward to use with monocomponent reaction rules (Figure 1A). Using this metric allows us to select chemical reactions similar to the ones present in metabolic databases, increasing our chances that this predicted reaction can be catalyzed.

However, in refs 12, 22, 23, the authors are interested in chemical retrosynthesis, whereas we are interested in enzyme-catalyzed reactions. Therefore, we combine (through multiplication) this chemical score with a scoring scheme that we developed previously.[11] Briefly, this biological score characterizes our confidence that a sequence exists to catalyze the desired enzymatic rule. We updated this scoring scheme to be a normalized score, between 0 and 1, to be in the same range as the chemical score (Figure 1B). This biological score has the useful property that rules at low diameter (*i.e.*, more promiscuous) are usually ranked lower than rules at high diameters (*i.e.*, more specific and trustworthy).

In RetroPath RL, this combined biochemical score is used for ranking and excluding reaction rules that are not considered trustworthy (similarity too low to the original reaction, or sequence availability too low).

Integrating Rule Ranking with the Monte Carlo Tree Search Reinforcement Learning Method. Reinforcement learning methods are a class of machine learning algorithms that learn how to take actions in an environment so as to maximize a cumulative reward. Their main strength is their capacity to balance exploration of unchartered space and exploitation of current knowledge.

A Monte Carlo Tree Search is a reinforcement learning algorithm that explores the unknown space with random sampling. It proceeds in 4 phases, repeated until a resource budget (time or number of iterations) has been exhausted (Figure 2):

- **Selection**. Starting from the root node (here, a chemical state containing the target compound), the best child nodes according to the selection policy are iteratively chosen until a leaf node is reached. In the proposed implementation we use a selection policy guided by the biochemical score of the reaction rule, unless otherwise stated.
- **Expansion**. Possible transformations are selected based on the ranking scheme presented above and the node is expanded (with a predefined maximum number of children).
- **Simulation or rollout**. This is an iterative procedure that starts by checking the status of the state. If it is terminal, a reward (or penalty) is returned according to the rewarding policy (detailed in the Materials and Methods section). If it is not terminal, a transformation is randomly sampled from available transformations and the process is repeated. This is performed until a maximum number of rollout steps or the maximal depth of the tree is reached.
- **Backpropagation or update**. The score obtained after exploring this node is returned to its parents to update their values and visit counts.

Evaluating RetroPath RL with a Golden Data Set. We first evaluate our implementation of MCTS for bioretrosynthesis on a manually curated data set of 20 compounds to identify the best settings for a retrosynthetic search on those compounds

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

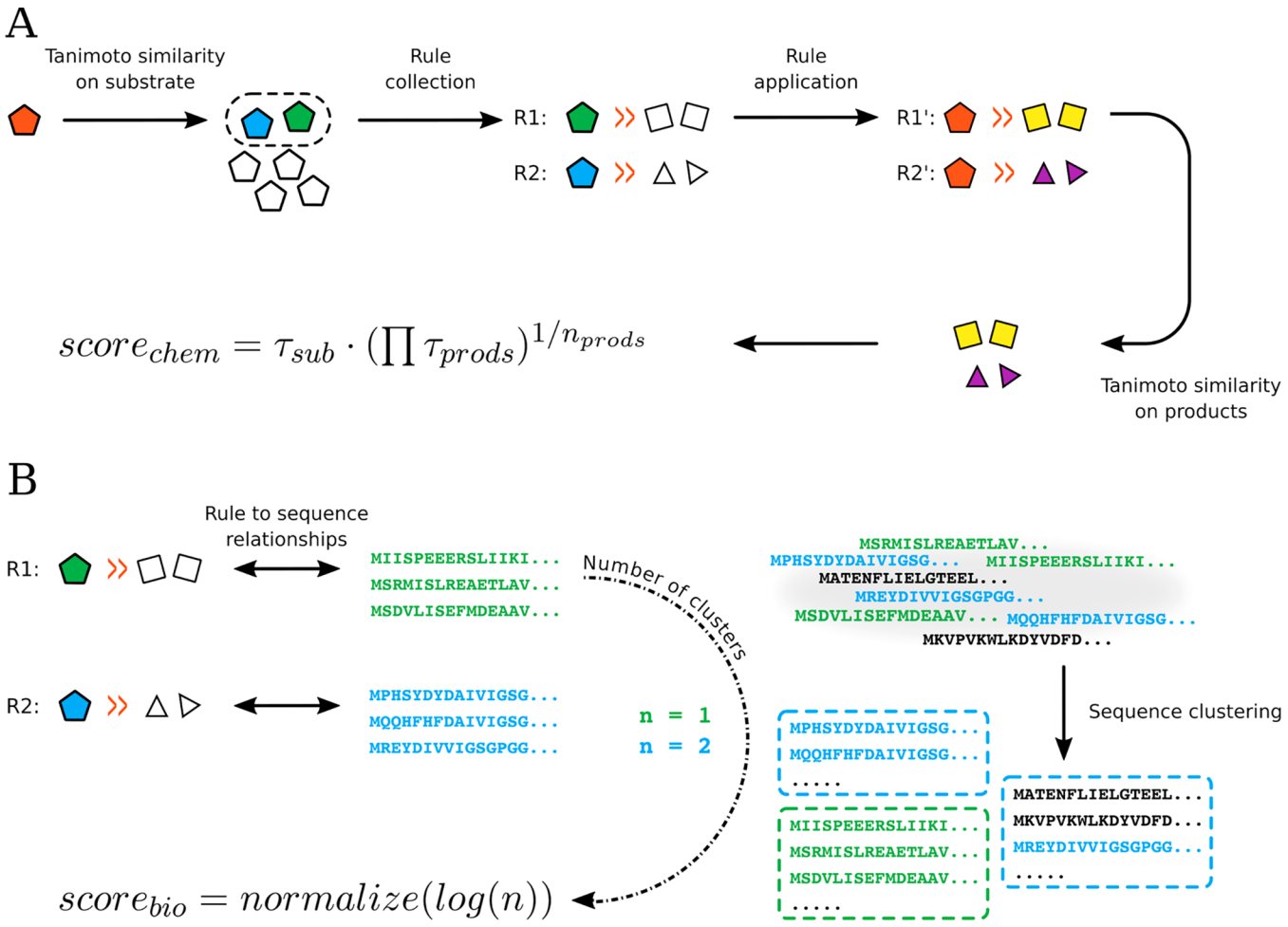


Figure 1. Chemical and biological score computations. For chemical score (A), we start by selecting substrates within the rule collection that are similar to the query substrate. We then apply those rule templates and check similarity of those products to the products the rules were learned on. For biological score (B), we establish rule-to-sequence relationships, and cluster all sequences based on sequence similarities. For each rule, we then count the number of cluster *n* spanning all related sequences. Score normalization is detailed in Materials and Methods.

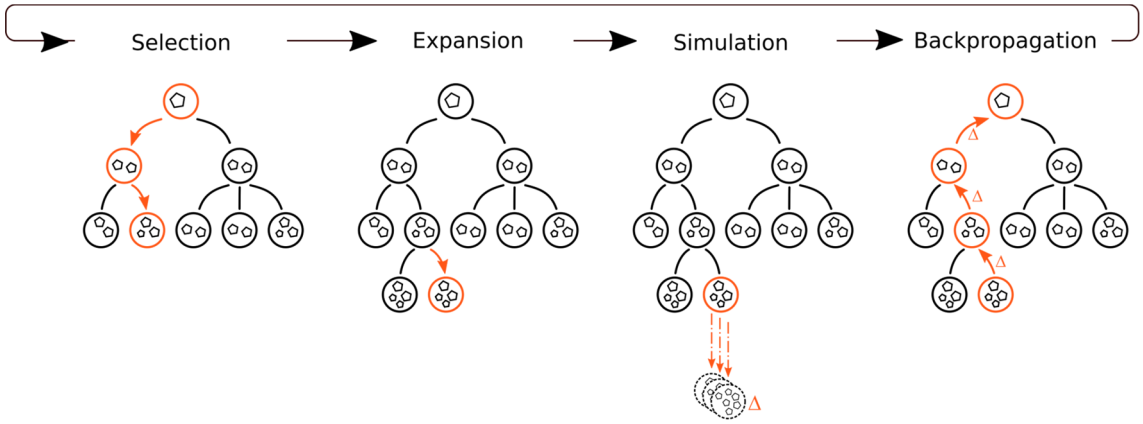


Figure 2. Monte Carlo Tree Search algorithm. Circles represent nodes, and pentagons molecules. Detailed explanations are in the main text and in the Materials and Methods section.

(chosen compounds and rationale for selection are available in the Materials and Methods section). This data set allows us to verify that bioretrosynthesis tools suggest pathways that have been experimentally described in the literature, ensuring the biological relevance of the predicted pathways. Comparing RetroPath RL's features for expert users (see Supplementary Note S1) on this golden data set allows us to select the best parameters possible for biological relevance. While various metrics could be available to describe what a good bioretrosynthesis algorithm should do, there is no obvious consensus. Should such an algorithm be fast? Return a lot of pathways? Return fewer but more reliable pathways? We use three criteria for comparing algorithms. First, it should return pathways for as many compounds as possible. Second, results should include the

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

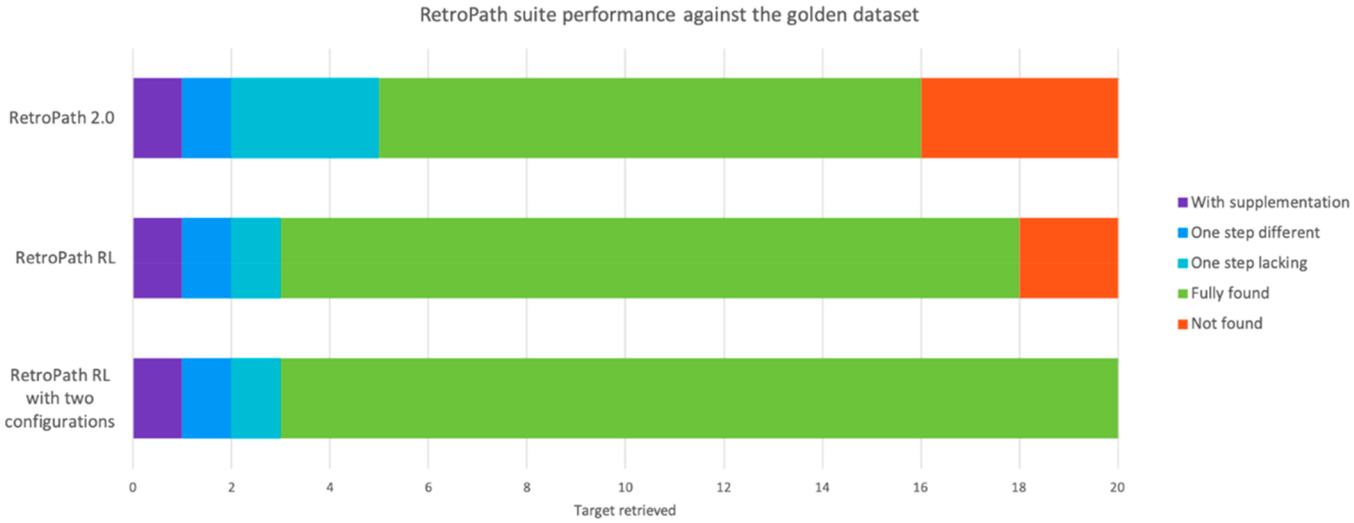


Figure 3. Results of the RetroPath suite against the golden data set to identify the experimental pathway. We compared results of RetroPath 2.0, the default configuration of RetroPath RL, and a combination of results between the default configuration and a more tolerant one on the used scores with a timeout of 3 h. With supplementation (purple) means a supplement has to be provided in the media to identify the correct experimental pathway. One step different (dark blue) means only one step differs from the described pathway, for example, by using a different cosubstrate. One step lacking (light blue) means the search algorithm found a pathway identical to the experimental one, except one step which was short-cut. Fully found (green) means the experimental pathway was found without restriction. Not found (orange) means the experimental pathway was not found.

chosen literature-described experimental pathway (exact intermediates are found). For parameter sets with identical results on those two criteria, the third criterion is that a better parameter combination should return the experimental pathway in fewer iterations.

The best parameters set we found used chemical and biological thresholds of 0.3, and a maximum of 10 allowed children per node (detailed configuration is available as Supplementary Table S2 and effects of various parameters were investigated and presented in Supplementary Note S1). To generate the results presented in Figure 3, we ran RetroPath 2.0, and RetroPath RL with its default configuration and a configuration with more tolerant chemical scores for rule application. Detailed analysis of results on this data set is available in Supplementary Note S2. We provide an example result in Figure 4 for mesaconic acid, interesting as it could be a renewable precursor to the commodity chemical methacrylic acid.[24]

We can see from these results that the algorithm encoded within RetroPath RL suggests experimentally relevant pathways for biochemists, as it finds the exact pathway described in the literature 75% of the time with strict settings, and 95% of the time when trying more tolerant settings on failed compounds, media supplementation or using another cofactor (lacking one step is considered a failure). We can see this is better than our previous algorithm, proving that RetroPath RL suggests more experimentally relevant pathways for metabolic engineering.

Importance of Our Scoring Schemes. Although all parameters of interest are evaluated in Supplementary Figures S1−S12 in Supplementary Note S1, we detail here the impact of the scoring cut-offs and scheme. As mentioned in the theoretical background section, we use a biochemical score, based on both chemical similarity and estimation of enzyme sequence availability. We analyzed algorithm behavior for various biological, chemical and biochemical scores, as well as when guided only by similarity, biological score or no scoring scheme (classical algorithm). The results are presented in Figure 5 and validate our approach. We can see the score contributing most to the importance of the cutoff parameter values is the biological score that ensures experimental availability of the enzymatic sequence, while the score that contributes most to the guidance of the Monte Carlo Tree Search is the chemical similarity component, showing both scores are necessary for ensuring efficient search and experimental relevance of predictions.

Evaluating RetroPath RL on Successful Metabolic Engineering Projects. After validating biological relevance of predictions for metabolic engineering, we tested RetroPath RL on a larger data set. Our previous tool[10] was tested on the LASER database[25,26] that compiles successful metabolic engineering projects, completed with compounds taken from the Metabolic Engineering journal (see Materials and Methods, available as Supplementary Data S1). Given the curation level of this data set, we checked the number of compounds for which we could find a pathway, and not the exact experimental pathway.

We ran our RetroPath 2.0 software using the same RetroRules rules (diameters 6, 10 and 16) and solved 77.6% compounds (118 out of 152), finding a median number of pathways of 4.5. With RetroPath RL and a score cutoff of 0.3, we solved 121 compounds (79.6%) with a median number of pathways of 11.5. Without score cutoff, we obtain 6 more compounds, yielding a success rate of 83.6% (results are available as Supplementary Data S2). The main advantage of the Monte Carlo Tree Search algorithm implemented in RetroPath RL over the brute force algorithm of RetroPath 2.0 is its capacity to find longer pathways. When giving RetroPath RL a maximum allowed depth of 10 steps throughout this study, it found pathways of 10 steps. Indeed, the memory requirements of the exhaustive search performed by RetroPath 2.0 essentially limit it to 5 step pathways, while RetroPath RL can explore longer pathways and therefore find more solutions given the same allowed time. Its focus on promising areas of the search space also lead RetroPath RL to propose more solutions for the same compound.

Supplement Finder for Media Supplementation. The literature pathway that we identified for TPA used xylene for media supplementation, *i.e.*, xylene is not a metabolite from the

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

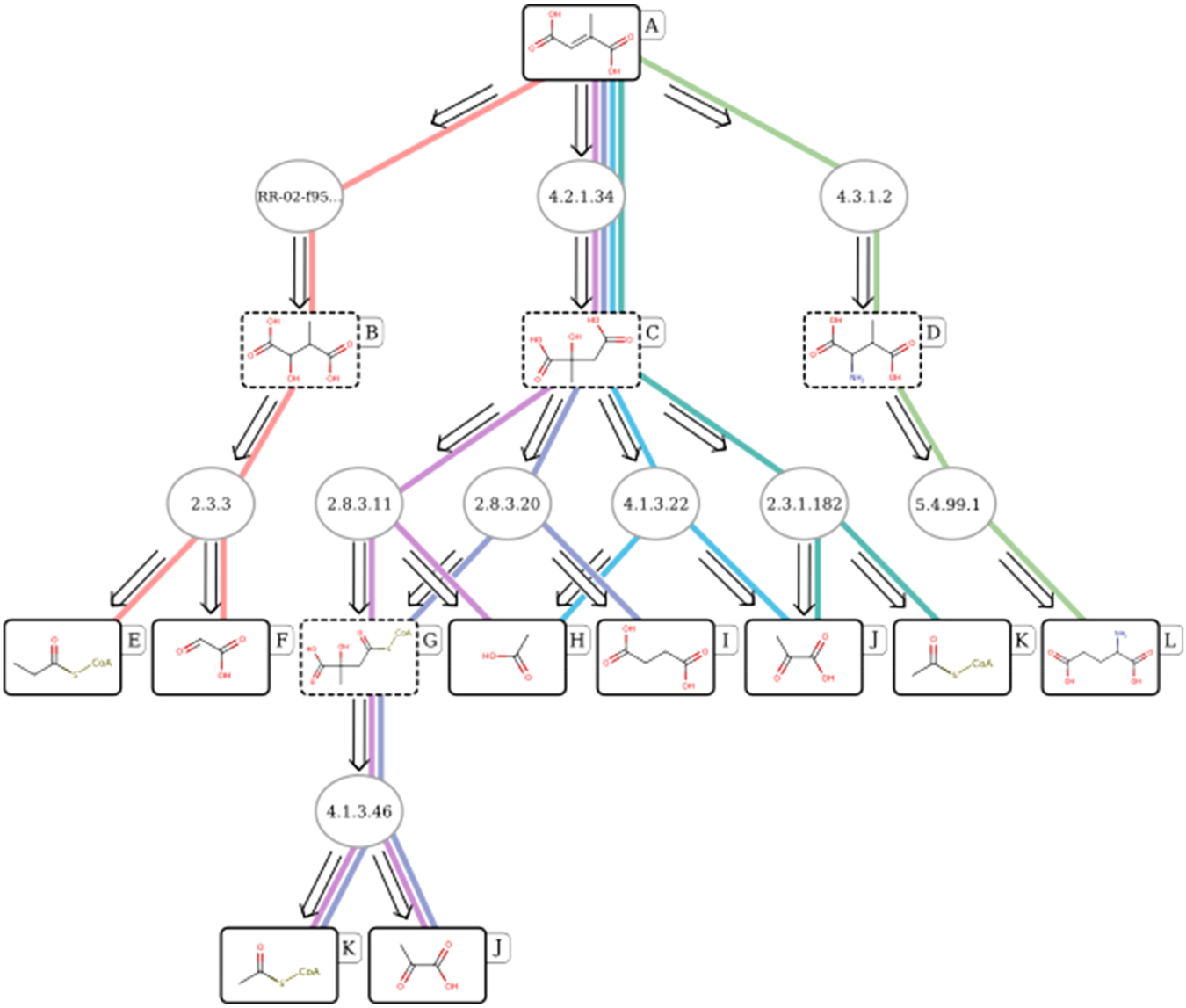


Figure 4. Example of bioretrosynthetic scope obtained from RetroPath RL for mesaconic acid. Dead end compounds have been removed, pathways lying in the scope are depicted with distinct colors. Compounds are represented by their chemical structures and reactions by their EC numbers or their rule ID of no EC is known. Mesaconic acid (at the top) and sink compounds (precursor metabolites) are surrounded by a solid line, while intermediates are surrounded by a dashed line. Only the pathways predicted up to 3 steps are shown. Compound names: mesaconic acid (A), 3-methylmalic acid (B), citramalic acid (C), 3-methylaspartic acid (D), propanoyl-CoA (E), glyoxylate (F), citramalyl-CoA (G), acetate (H), succinate (I), pyruvate (J), acetyl-CoA (K), glutamate (L).

microbial strain but was necessary for producing TPA and experimentally added to the strain's growth media.[27] Knowing this allowed us to add xylene to the list of starting compounds for retrosynthetic search. Media supplementation is commonly done in metabolic engineering but rarely integrated into retrosynthesis tools. We therefore developed a RetroPath RL feature that analyses search trees and suggests potential media supplements, available from the chemical provider Sigma. We first extracted supplement information from LASER, as well as two TPA experimental pathways known from the literature. Filtering out gene inducers, antibiotics or early precursors, we obtain a list of 8 curated pathways with supplementation (available as Supplementary Data S3).[27−34] In 5 cases out of 8, we retrieve the compound that was used for supplementation in the described experimental pathway. For 2 cases, we did not find the experimentally described supplement but suggest other possibilities. In the last case, no relevant chemical was suggested as a potential supplement. Here, we used availability in the Sigma-Aldrich catalogue as a criterion on whether a compound could be an interesting supplement. However, this feature can be used with any database of interest, for example, with in-house compounds of a laboratory. One could include criteria such as capacity to cross membranes, solubility, toxicity, cost or any other feature of interest and select compounds that are biologically relevant for their application of interest.

Custom Use of RetroPath RL: Avoid Toxic Intermediates. We sought to make our tool as modular and flexible as possible, therefore allowing expert users to input their knowledge. We showcase this by implementing a toxicity score to bias the search away from toxic compounds.[6,35] This toxicity score is negative (between 0 and −10 in our training data set) for toxic compounds and set to 0 otherwise. The strength of using bias in MCTS is that it favors preferred routes (here, avoiding toxic intermediates) but can still find results if

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

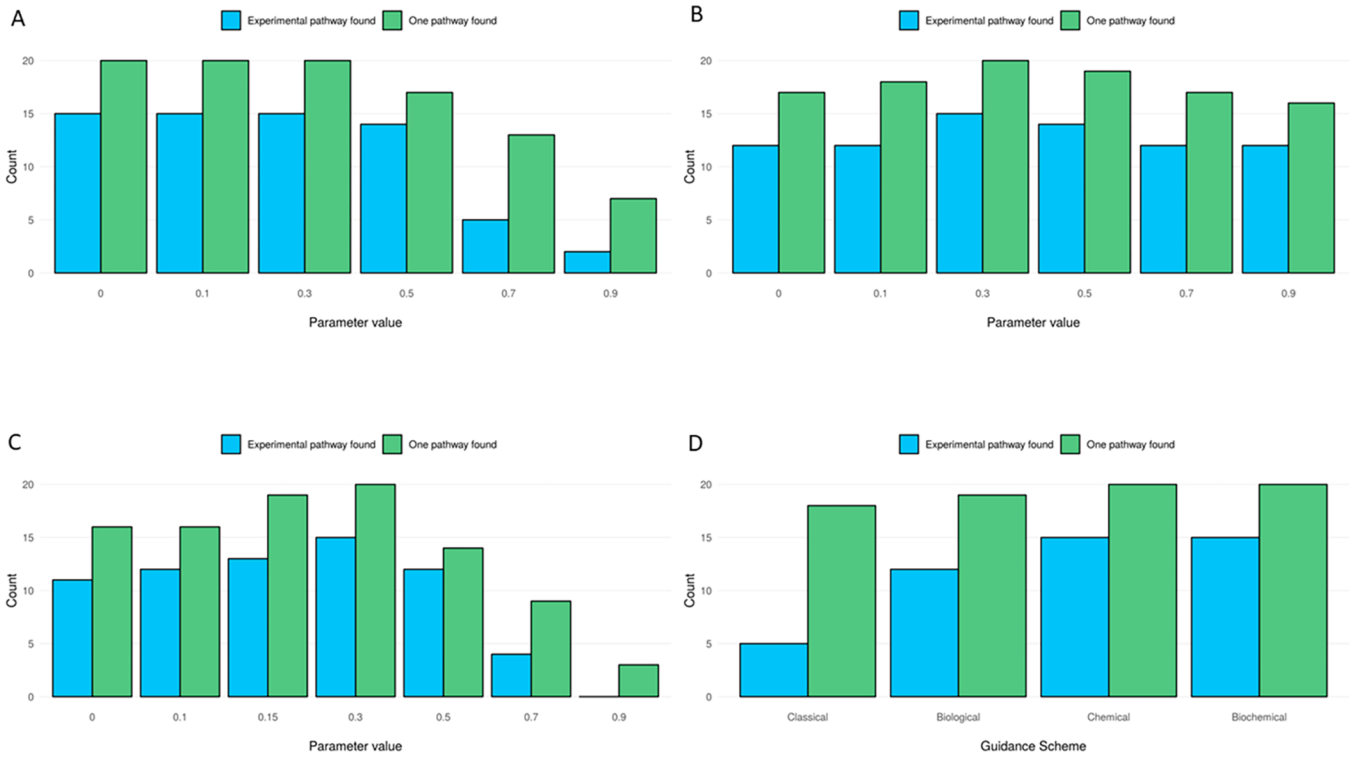


Figure 5. Impact of scoring scheme on retrieval performance of RetroPath RL. We compared results between using a biological score cutoff (A), a chemical score cutoff (B), and a biochemical score cutoff (C) varying between 0 and 0.9. In (D) we compared results between guiding the search based on the Classical UCT formula, a formula guided by biological scoring, chemical scoring, or biochemical scoring. One pathway found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

those preferred routes are not successful, in contrary to other algorithms that would exclude those intermediates altogether. While implementing this feature did not change our results on the golden data set (*i.e.*, the experimental pathway was identified for the same compounds), the order as well as the total number of returned pathways was impacted (Supplementary Table S3).

While we tested it with toxicity, biasing the search could also be used to encourage pathways to be found from a set of privileged metabolites (core metabolism), by cost, availability in the cell or any other metric the advanced user wishes to use, making MCTS ideal to incorporate biological knowledge into retrosynthetic search.

■ CONCLUSION AND REMARKS

We showcase here the use of Monte Carlo Tree Search for bioretrosynthesis, and provide an implementation as RetroPath RL, an open-source Python package available on GitHub. While MCTS has already been implemented for chemical synthesis,[17] RetroPath RL is to our knowledge the first application of this algorithm to biochemistry and metabolic engineering. It is a versatile, modular command line tool, build for metabolic engineers, that takes as input a compound of interest, a microbial strain as sink (*i.e.*, the list of available precursor metabolites) and a set of reaction rules.

While RetroPath RL can take as input any set of chemical rules in encoded into the SMARTS linear notation, giving great freedom to users to use reactions from their pathways or microbial strain of expertise, the tests presented in this article use rules from our RetroRules database.[21] RetroRules was built using a data-driven approach showcased in earlier version of RetroPath,[11] allowing promiscuity encoding. Using both reaction rules at various diameters and chemical similarity scoring, we can tune the allowed promiscuity within our search, which is an advantage over building reaction rules from EC numbers as is commonly done in metabolic engineering. Another advantage of a data-driven approach is that it is bound to get more precise and information-rich as metabolic databases expand.

We validated RetroPath RL by verifying on a manually curated data set that a literature-described pathway could be found for 20 different compounds. The fact that the experimental pathways were found for 75% of compounds using default settings and 95% of the time using more tolerant ones, with much better results when using biochemically guided search, confirms the ability of RetroPath RL to suggest experimentally viable pathways for metabolic engineers on numerous compounds.

Moreover, following the standards we set in our previous paper,[11] we tested RetroPath RL on a larger data set and found a pathway for 83.6% of compounds that were results of successful metabolic engineering projects. These results confirm that RetroPath RL generalizes well to metabolic engineering compounds outside the manually curated golden data set. Moreover, RetroPath RL suggests more pathways (median number of 11.5 *versus* 4.5), giving metabolic engineers more suggestions with which to exercise their expertise.

While a restricted number of microbial strains was provided with RetroPath RL, the user can provide his own strain or supplement the media with a given compound of interest. For example, when supplementing xylene to the microbial strain an experimentally described pathway to TPA[27] is found, and when adding thebaine to the media, a pathway for morphine production is also found.[31] A novel feature that should be welcomed by the metabolic engineering community is the ability

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

of RetroPath RL to suggest media supplements to complement the enzymatic synthesis plan. This feature, tested on 8 pathways, found the experimentally described supplement for 5 of them, and also suggests other supplements to test.

We showcase RetroPath RL's modularity by biasing the search toward less toxic intermediates, and also provide expert users 2 strategies to speed-up the retrosynthetic search, either by storing results in a database (Supplementary Note S3) or extending a search from previously run search trees (Supplementary Note S4). This possibility can be particularly interesting for users that wish to find alternative pathways after the results from a first search.

Another feature of interest is the ability to use RetroPath RL for biosensor design, as we also demonstrated with our previously developed tools.[36,37] This can be used in conjunction with a data set of detectable compounds[38] to allow for design of Sensing Enabling Metabolic Pathways.

Despite the advantages presented above, RetroPath RL still presents some limits, mostly related to pathway ranking. The authors of this article believe modular design, with a downstream analysis of selected pathways, to be a more appropriate course of action than ranking within the tool: so many genome or growth conditions modifications can increase a pathway yield that including all ranking schemes into one to come up with a "best" pathway neglects both metabolic engineers' expertise and years of lessons from the industry. However, most developers present all integrated tools that perform pathway search and ranking together. Ranking schemes in such tools can involve accounting for enzyme integration into the microbial strain, kinetics, toxicity, carried flux, thermodynamics or preferred cofactor usage.[10,13] Our approach provides two advantages over integrated tools. First, ranking the pathways separately allows for a much more modular integration, and the ability to integrate new ideas into ranking much more easily. Moreover, as shown in the custom use section, for advanced users attached to using these schemes, it is possible to integrate them easily given RetroPath RL's modular code and the ability of the Monte Carlo Tree Search algorithm to bias the search toward or away from properties of interest.

While stereochemistry has been described as one of nature's most interesting advantages compared to the traditional chemical industry,[39] it is not used in results presented here, mainly due to current technical limitations to the way stereochemistry is handled by the cheminformatics packages we used at the moment. However, since our formalism uses rules SMARTS which are stereoaware,[40] and our standardization schemes can leave or remove stereochemistry, users wishing to use stereochemistry in RetroPath RL can do so.

Another major advance that could be included in RetroPath RL would be to guide reaction selection steps through learned values instead of similarity. For example, this was implemented in ref 17 or ref 41. However, the authors learned values from Reaxys,[42] which contains 12.4 million single-step reactions (compared to around 80 000 in MetaNetX, including reactions without chemical structures[43]). Therefore, using learned values in bioretrosynthesis seems out of reach for the moment, but could become available in the coming years due to the intense curation efforts under way in the community. Moreover, those learned values in bioretrosynthesis would have to be strain-dependent, as an intermediate compound's value for bioretrosynthesis depends highly on the strain of interest.

In conclusion, we present here a highly modular tool using one of the latest tree search algorithms for bioretrosynthesis. This tool is modular enough for expert users to input their expert knowledge and has been thoroughly tested on data sets of interest to the community.

■ MATERIALS AND METHODS

All chemical operations were performed using RDKit release 2019.03.1.0 and Python 3.6.

Compound Standardization. All compounds were standardized using the following steps:

- Sanitizing chemical depictions using RDKit's SanitizeMol method.
- Removing isotope.
- Neutralizing charges.
- Removing stereo.
- Converting back and forth to InChI[44] to ensure tautomerism consistency.

Reaction Rule Encoding. Our reaction rules are generated as presented in our RetroRules database.[21] Briefly, we extracted known biochemical reactions from the MetaNetX[43] database version 3.1 and filtered out incomplete reactions. We identified the reaction centers based on atom−atom mappings we performed using the Reaction Decoder software[45] (version 2.1.0). We decomposed multisubstrates reactions into monosubstrate components by considering one substrate per component and only the subset of products that share at least one atom with the substrate. Monocomponents involving a typical cofactor (such as CO_2, ATP, NADH, ...) as substrate were excluded. Finally, each monocomponent reaction is encoded into a collection of reaction rules using the SMARTS[40] formalism for a diameter ranging from 2 to 16 around the reaction center.

The main difference with the RetroRules procedure aforementioned is that we use implicit hydrogen notation in the reaction rules instead of explicit, which allows a much faster computation of standardization of compounds and reaction rule application through RDKit.

To validate the reaction rules, we checked that applying the reaction rules on the substrates used as templates produce the products described in the template reactions. The success rates range from 99.4% for reaction rules at diameter 2 to 99.8% for those at diameter 16. We also performed a cross-radius consistency check by ensuring that sets of products produced by large diameter reaction rules (*e.g.*, reaction rules generated with a diameter of 10) are always subsets of products produced by reaction rules at any smaller diameter (*e.g.*, reaction rules for diameter 8), which show a success rate of 99.3%. Only reaction rules passing successfully both procedures have been retained and used, which represent almost 146 000 distinct reaction rules (available for download on RetroRules Web site, download page, release rr02) and model more than 18 000 biochemical reactions in both directions.

Branching Factor Calculation. We generated what we called the extended metabolic space (EMS) at 1 step, *i.e.*, the metabolic space that can be reached by applying our reaction rules once on all compounds of MetaNetX[43] being used as a template for generating at least one reaction rule. We filtered out substrates having a molecular weight greater than 1 kDa. We set a timeout cutoff of 1 s on rule application. We then analyzed the results presented in Table 1 and Supplementary Table S1 using the NetworkX python library.[46]

Chemical Score Calculation. After compound standardization, we calculate a 1024 binary Morgan fingerprints

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

vector using the RDKit method GetMorganFingerprintAsBitVect at diameter 4.

For substrate score, the query substrate is compared to all native substrates a rule was learned on using Tanimoto score,[47,48] and the maximum score is kept.

For products and substrate scoring, the procedure is as follows.

For each native (n_sub, n_products) couple:

- Calculate Tanimoto of query and native substrate.
- Generate all combinations of native and query products (below 1000 to avoid a combinatorial explosion. The number of combinations is $n!$ where n is the number of products generated by the rule application).
- For each combination, calculate the geometric mean of the Tanimoto scores of products.
- Keep the highest combination score.
- The score of this native combination is the product of the substrate score and the highest combination score.

The score of the rule is the highest score of all native substrates and products the rule was generated from.

Rules generating a different number of products from the template receive a score of −1.

Biological Score Calculation. A penalty is calculated as presented in our RetroPath 2.0 paper.[11] Briefly, we clustered reaction rules according to the EC number annotations inherited from the template reactions. Independently, we clustered enzyme sequences collected from UniProt[49] (release 2019_04) according to sequence similarity using the cd-hit software[50] (version 4.6.8). We establish the sequences to reaction rules associations based on Rhea[51] (v98), MetaCyc[52] (v21.5), and Reactome[53] (v66) cross-links. The penalty score was then computed as

$$\text{penalty}_{\text{score}}(\text{rule}) = \log_{10}(n_{\text{rule}})$$

with n_{rule} the number of distinct clusters that contains sequences associated with the rule.

The final biological score was obtained after normalization of this penalty so that it becomes a score comprised between 0 and 1 using the functions below. We chose the regularization parameter so that half of the rules would have a score above 0.5 and the other half below.

$$\text{score}_{\text{bio}}(\text{rule}) = \frac{1}{1 + \alpha \cdot \text{weighted}_{\text{penalty}}(\text{rule})}$$

with

$$\text{weighted}_{\text{penalty}}(\text{rule}) = \frac{\text{penalty}_{\text{score}}(\text{rule})}{\sqrt{\text{radius}(\text{rule})}}$$

and

$$\alpha = \frac{1}{\text{median}(\text{weighted}_{\text{penalty}})}$$

Biochemical Score Calculation. The biochemical score was computed as the product of the chemical and biological scores:

$$\text{score}_{\text{biochem}}(\text{rule}) = \text{score}_{\text{bio}}(\text{rule}) \times \text{score}_{\text{chem}}(\text{rule})$$

Sinks Construction. Except stated, sink compounds—*i.e.*, the list of available precursor metabolites—have been extracted from genome-scale metabolic models by only collecting the chemicals that lie in the cytosol compartment. In addition, we filtered out "dead-end" compounds, *i.e.*, compounds that cannot be produced by any reactions in a steady-state metabolic model, that we detected by performing a Flux Variability Analysis using the COBRApy package[54] (v0.15.3).

Chemical structures have been obtained using cross-links from the models to metabolic databases. In case no cross-link or not any valid structure was found, the PubChem[55] database was examined using compound names as query. Finally, all chemical structures were standardized as described in the Compound Standardization section.

Available Sinks. The available sinks provided with our software are iML1515,[56] iJO1366,[57] and core *E. coli* metabolism,[58] as well as *Bacillus subtilis* iYO844 model[59] and our set of detectable compounds for biosensor design.[38] The genome scale models were obtained from the BiGG Models database.[60] By default, we used sinks from the iML1515 model.

RetroPath 2.0 Configuration. For all tests made with RetroPath 2.0, we perform all executions on a recent workstation. We use the sink extracted from the iML1515 model, and we set the maximum pathway length to 5, the maximum number of structures to keep for next iteration to 1000, and a 3 h per execution time budget. We then use the rp2paths software available on GitHub at https://github.com/brsynth/rp2paths to extract the pathways from RetroPath 2.0 output.[11]

Monte Carlo Tree Search Implementation. The aim of Markov decision processes is to model sequential decision processes of an agent in an environment.[61] Its most notable components are states (representing positions in a game) and actions (allowed transformations from the state).

In RetroPath RL, following the method developed by ref 17, we consider states to be a set of molecules. The initial state contains only the target compound one desires to produce. Actions are transformations of any molecule of the state that do not transform the compound into itself nor produce a nonsink compound that has already been produced before in the synthesis plan so as to avoid loop searches.

A state is considered terminal if all compounds are in the sink, if no move can be applied to this state or if the maximum allowed depth has been reached.

Monte Carlo Tree Search is a reinforcement learning approach that builds a search tree and stochastically explores search space to bias search toward most promising regions of combinatorial space, following steps presented in Figure 2B and detailed below.

Selection. Starting from the root node, the best child nodes according to the selection policy are iteratively chosen until a leaf node is reached. The formula we used is

$$\text{Value} = \frac{\text{Node}_{\text{score}}}{\text{Node}_{\text{visits}}} + \text{UCTK} \cdot \text{score}_{\text{chem}} \cdot \text{score}_{\text{bio}} \cdot \sqrt{\frac{\text{Parent}_{\text{visits}}}{1 + \text{Node}_{\text{visits}}}}$$

where $\text{Node}_{\text{score}}$ is the cumulative score from rollouts, $\text{Node}_{\text{visits}}$ is the number of visits to this node, UCTK is the UCT (Upper Confidence Trees) constant used (balances between exploitation and exploration), chemical and biological scores are the scores of the move leading to this node, and $\text{Parent}_{\text{visits}}$ is the number of visits of this Node's parents.

Other policies have been developed, notably with only chemical, biological score, or no scoring. In our implementation,

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

grandchildren of a node can only be explored if all his children have had at least *minimal_visits* number of visits. This allows mandatory rollout on different branches at least once to favor exploration.

The Classical UCT formula (without guidance) is the following:

$$\text{Value} = \frac{\text{Node}_{\text{score}}}{\text{Node}_{\text{visits}}} + \text{UCTK}\cdot\sqrt{\frac{\text{Parent}_{\text{visits}}}{1 + \text{Node}_{\text{visits}}}}$$

Expansion. For each compound that is not in the sink, its *n* best moves are identified and stored (with *n* the maximal number of children allowed for the node). Then, the *n* best moves overall on the state are selected and children created iteratively (one at the first visit of the node, the next at the next visit and so on) for each of these moves and a rollout is performed.

Rollout. Rollout is an iterative procedure that starts by checking the status of the state. If it is terminal, a reward (or penalty) is returned according to the rewarding policy. If it is not terminal, a transformation is randomly sampled from available transformations and the process is repeated. This is performed until a maximum number of rollout steps or the maximal depth of the tree is reached. The function for random sampling used throughout this study gives weight chemical × biological score to moves, therefore giving more probability of being sampled to higher scoring moves according to our scoring scheme.

Rewarding Policy. A state is rewarded as follows:

- Receives a penalty of −1 when no compound is solved.
- Receives a bonus of 5 when the state is fully solved.
- Receives a score of number_found/total_number when only a fraction of molecules in the state are solved.

Update. The node and its parents update their value and node counts according to the results obtained from the above rewarding scheme after rollout.

Returning Complete Pathways. Each time a full pathway is found during the tree expansion, the pathway is returned, and an additional bonus of 10 is received by the node, to allow for biasing toward similar successful pathways. At the end of the search, the most visited pathway is returned ("best"), and all pathways are returned ranked in order of decreasing biochemical score.

Golden Data Set Construction. In order to perform golden data set curation, we focused on articles where pathways were explicitly described (*i.e.*, no missing steps, and available intermediate compounds). We retrieved compound structures in PubChem[55] and EC numbers from Brenda[62] based on enzyme name as given in the article. We selected pathways of strictly more than 1 step. We then verified for each step that we had chemical rules available in our RetroRules[21] database to encode the described transformations. This ensures a fair comparison between tools using our publicly available reaction rules, in order to evaluate separately retrosynthesis tools and the underlying chemical rules. The detailed list with references is available as Supplementary Table S4, and the pathways as Supplementary Data S1.

Experimental Pathway Comparison. In order to compare the pathways found by RetroPath 2.0 or RL and the experimental pathways described in the literature, we have two types of information: compound identity and EC number identity. We consider the reaction EC number to be equal if it is identical up to 3 digits (1.1.1.x is identical to 1.1.1.y). Since spontaneous reactions do not have an EC number, we used compound identity for comparison, and EC number as additional information.

LASER Retrieval and Metabolic Engineering Completion. We build the LASER data set by parsing target molecule and microbial strain information from the LASER database published by Winkler *et al.* which provides a curated list of more than 600 successfully implemented metabolic designs.[25] When available, we store the chemical structure provided by MetaCyc (v2RL), otherwise we query the PubChem database based on target compound names. We augmented this list with target compounds reported in the Metabolic Engineering journal in 2016 (volumes 33−38) and published in RetroPath 2.0.[11] All chemical structures have been standardized using the procedure described in the Compound Standardization section. The final data set used in the present paper contains 211 unique structures that are provided as Supplementary Data S2.

Extracting Supplements from LASER. LASER provides a "Media" line that contains addition to the media, extracted using Natural Language Processing. This can include antibiotics, promoter inducers or supplements of interest required to build the pathway. We removed all compounds that did not concern pathway supplements and removed early precursor supplementation. Structures were obtained from PubChem. We obtain a list of only 6 pathways satisfying these requirements, and 8 when we also add two pathways for TPA from literature.[27] This list[27−34] is available as Supplementary Data S4.

Supplement Finder Feature. The Supplement finder functions as follows:

- Load search tree in memory.
- Explore all nodes and keep compound structures that complete of a chemical state (*i.e.*, all other compounds of the state are solved).
- Compounds that allow for completion of more than *N* states (which would complement *N* pathways) are kept. Here *N* = 1: any compound that can complement a pathway is kept.
- Compounds are filtered according to presence in a Database of interest. Here, we filtered according to presence in the Sigma catalogue.
- We keep the *N* best suggestions (according to number of pathways that are completed). Here, we returned up to 20 potential supplements.
- All completed pathways are extracted for future analysis.

Toxicity Implementation. We used data from EcoliTox[35] and XTMS[6] to build a QSAR model (using as input features 1024 binary Morgan fingerprints vector calculated with the RDKit method GetMorganFingerprintAsBitVect at diameter 4), predicting log(IC50) of the compounds. We train our model using scikit-learn[63] (version 0.19.1). The model is a multilayer perceptron trained with the default parameters from scikit-learn except the following parameters: maximum iteration of 20000, adaptive learning rate, adam solver, early stopping and the following layers: (10, 100,100, 20). This model has a Leave-One-Out (LOO) score of 0.81. In prediction mode, toxicity was used only if the predicted log(IC50) ≤ 0. The modified UCT function that was used is

$$\text{Value} = \frac{\text{Node}_{\text{score}}}{\text{Node}_{\text{visits}}} + \text{bias}_k\frac{\text{toxicity}}{1 + \text{Node}_{\text{visits}}} + \text{UCTK}\cdot\text{score}_{\text{chem}}\cdot\text{score}_{\text{bio}}\cdot\sqrt{\frac{\text{Parent}_{\text{visits}}}{1 + \text{Node}_{\text{visits}}}}$$

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

Hardware. We ran our tests on 2 calculation clusters and 1 personal computer. Tests involving using the NoSQL Database and media supplementation were run on a desktop computer with the following characteristics: CPU is Intel Xeon W-2145 CPU @ 3.7 GHz and 32 G of RAM.

Tests for the golden data set evaluation were performed on the Migale (https://migale.inra.fr) cluster (2.0 to 2.4 GHz CPUs). Allocated resources per job were set to 1 vCPU and 20 GB. Tests for the LASER database evaluation were run on the IFB cluster (http://taskforce-nncr.gitlab.cluster.france-bioinformatique.fr/doc/cluster-desc/) (2.3 GHz CPUs). Allocated resources per job were set to 1 vCPU and 40 GB. Cluster tests were run using snakemake[64] 5.4.0.

Code and Data Availability. RetroPath RL is available at https://github.com/brsynth/RetroPathRL under a MIT license.

Reaction rules are available at https://retrorules.org/dl.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acssynbio.9b00447.

Supplementary Data S1: Golden data set pathways under json and SBML format (ZIP)

Supplementary Data S2: LASER database and Metabolic Engineering compounds; Supplementary Data S3: LASER database and Metabolic Engineering results; Supplementary Data S4: Targets and supplements for experimentally described pathways (ZIP)

Supplementary Table S1: Mean number of rules applying to a compound at various diameters; Supplementary Table S2: Detailed configuration data for RetroPath RL runs on validation data sets; Supplementary Table S3: Toxicity biased results; Supplementary Table S4: Golden data set structures and references; Supplementary Note S1 (with Supplementary Figures S1−S12): Detailed parameters explanation and analysis; Supplementary Note S2: Detailed golden data set analysis; Supplementary Note S3 (with Supplementary Figure S13): Database sped-up calculations; Supplementary Note S4 (with Supplementary Figure S14): Extending a previous search (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: jean-loup.faulon@inra.fr.

ORCID

Jean-Loup Faulon: 0000-0003-4274-2953

Author Contributions

M.K., T.D., and J.-L.F. designed the study. M.K. developed the MCTS algorithm. T.D. developed rule extraction and application code, strain analysis, and LASER extraction. Both M.K. and T.D. tested and validated the software. M.K., T.D., and J.-L.F. wrote the paper.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We are grateful to the INRA MIGALE bioinformatics platform (https://migale.inra.fr) for providing computational resources. We would like to thank the French Institute of Bioinformatics (IFB, ANR-11-INBS-0013, https://www.france-bioinformatique.fr) for providing storage and computing resources on its Core Cluster. M.K is supported by DGA (French Ministry of Defense) and Ecole Polytechnique. J.-L.F. acknowledges support from BBSRC/EPSRC (grant number BB/M017702/1) and from the ANR (grant numbers ANR-15-CE21-0008 and ANR-17-CE07-0046). We thank Pablo Carbonell for providing the toxicity data.

■ REFERENCES

(1) Rodrigo, G., Carrera, J., Prather, K. J., and Jaramillo, A. (2008) Desharky: Automatic Design of Metabolic Pathways for Optimal Cell Growth. *Bioinformatics 24* (21), 2554−2556.

(2) Moriya, Y., Shigemizu, D., Hattori, M., Tokimatsu, T., Kotera, M., Goto, S., and Kanehisa, M. (2010) PathPred: An Enzyme-Catalyzed Metabolic Pathway Prediction Server. *Nucleic Acids Res. 38*, W138−W143.

(3) Henry, C. S., Broadbelt, L. J., and Hatzimanikatis, V. (2010) Discovery and Analysis of Novel Metabolic Pathways for the Biosynthesis of Industrial Chemicals: 3-Hydroxypropanoate. *Biotechnol. Bioeng. 106* (3), 462−473.

(4) Carbonell, P., Fichera, D., Pandit, S. B., and Faulon, J. L. (2012) Enumerating Metabolic Pathways for the Production of Heterologous Target Chemicals in Chassis Organisms. *BMC Syst. Biol. 6*, 10.

(5) Yim, H., Haselbeck, R., Niu, W., Pujol-Baxley, C., Burgard, A., Boldt, J., Khandurina, J., et al. (2011) Metabolic Engineering of Escherichia Coli for Direct Production of 1,4-Butanediol. *Nat. Chem. Biol. 7* (7), 445−452.

(6) Carbonell, P., Parutto, P., Herisson, J., Pandit, S. B., and Faulon, J. L. (2014) XTMS: Pathway Design in an eXTended Metabolic Space. *Nucleic Acids Res. 42*, W389−94.

(7) Liu, M., Bienfait, B., Sacher, O., Gasteiger, J., Siezen, R. J., Nauta, A., and Geurts, J. M. W. (2014) Combining Chemoinformatics with Bioinformatics: In Silico Prediction of Bacterial Flavor-Forming Pathways by a Chemical Systems Biology Approach 'Reverse Pathway Engineering'. *PLoS One 9* (1), No. e84769.

(8) Campodonico, M. A., Andrews, B. A., Asenjo, J. A., Palsson, B. O., and Feist, A. M. (2014) Generation of an Atlas for Commodity Chemical Production in Escherichia coli and a Novel Pathway Prediction Algorithm, GEM-Path. *Metab. Eng. 25*, 140−58.

(9) Hadadi, N., Hafner, J., Shajkofci, A., Zisaki, A., and Hatzimanikatis, V. (2016) ATLAS of Biochemistry: A Repository of All Possible Biochemical Reactions for Synthetic Biology and Metabolic Engineering Studies. *ACS Synth. Biol. 5* (10), 1155−66.

(10) Kumar, A., Wang, L., Ng, C. Y., and Maranas, C. D. (2018) Pathway Design Using de Novo Steps through Uncharted Biochemical Spaces. *Nat. Commun. 9* (1), 184.

(11) Delépine, B., Duigou, T., Carbonell, P., and Faulon, J. L. (2018) Retro Path2.0: A Retrosynthesis Workflow for Metabolic Engineers. *Metab. Eng. 45*, 158−70.

(12) Cho, A., Yun, H., Park, J. H., Lee, S. Y., and Park, S. (2010) Prediction of Novel Synthetic Pathways for the Production of Desired Chemicals. *BMC Syst. Biol. 4*, 35.

(13) Tokic, M., Hadadi, N., Ataman, M., Neves, D., Ebert, B. E., Blank, L. M., Miskovic, L., and Hatzimanikatis, V. (2018) Discovery and Evaluation of Biosynthetic Pathways for the Production of Five Methyl Ethyl Ketone Precursors. *ACS Synth. Biol. 7* (8), 1858−73.

(14) Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., et al. (2016) Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature 529* (7587), 484−89.

(15) Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., et al. (2018) A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go through Self-Play. *Science 362* (6419), 1140−44.

(16) Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., et al. (2017) Mastering the Game of Go without Human Knowledge. *Nature 550* (7676), 354−59.

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

(17) Segler, M. H. S., Preuss, M., and Waller, M. P. (2018) Planning Chemical Syntheses with Deep Neural Networks and Symbolic AI. *Nature 555* (7698), 604−10.
(18) Casini, A., Chang, F. Y., Eluere, R., King, A. M., Young, E. M., Dudley, Q. M., Karim, A., et al. (2018) A Pressure Test to Make 10 Molecules in 90 Days: External Evaluation of Methods to Engineer Biology. *J. Am. Chem. Soc. 140* (12), 4302−16.
(19) Lee, S. Y., Kim, H. U., Chae, T. U., Cho, J. S., Kim, J. W., Shin, J. H., Kim, D. I., Ko, Y. S., Jang, W. D., and Jang, Y. S. (2019) A Comprehensive Metabolic Map for Production of Bio-Based Chemicals. *Nat. Catal. 2* (1), 18−33.
(20) Nam, H., Lewis, N. E., Lerman, J. A., Lee, D. H., Chang, R. L., Kim, D., and Palsson, B. O. (2012) Network Context and Selection in the Evolution to Enzyme Specificity. *Science 337* (6098), 1101−4.
(21) Duigou, T., du Lac, M., Carbonell, P., and Faulon, J. L. (2019) Retro Rules: A Database of Reaction Rules for Engineering Biology. *Nucleic Acids Res. 47* (D1), D1229−35.
(22) Coley, C. W., Rogers, L., Green, W. H., and Jensen, K. F. (2017) Computer-Assisted Retrosynthesis Based on Molecular Similarity. *ACS Cent. Sci. 3* (12), 1237−45.
(23) Hadadi, N., Peyhani, H. M., Miskovic, L., Seijo, M., and Hatzimanikatis, V. (2019) Enzyme Annotation for Orphan and Novel Reactions Using Knowledge of Substrate Reactive Sites. *Proc. Natl. Acad. Sci. U. S. A. 116* (15), 7298−7307.
(24) Yadav, S. K., Schmalbach, K. M., Kinaci, E., Stanzione, J. F., III, and Palmese, G. R. (2018) Recent Advances in Plant-Based Vinyl Ester Resins and Reactive Diluents. *Eur. Polym. J. 98*, 199−215.
(25) Winkler, J. D., Halweg-Edwards, A. L., and Gill, R. T. (2015) The LASER Database: Formalizing Design Rules for Metabolic Engineering. *Metab. Eng. Commun. 2*, 30−38.
(26) Winkler, J. D., Halweg-Edwards, A. L., and Gill, R. T. (2016) Quantifying Complexity in Metabolic Engineering Using the LASER Database. *Metab. Eng. Commun. 3*, 227−33.
(27) Bramucci, M. G., McCutchen, C. M., Nagarajan, V., and Thomas, S. M. (2001) Microbial production of terephthalic acid and isophthalic acid. US6187569 B1.
(28) Wang, J., Tian, J., and Xu, J. H. (2006) A method for producing terephthalic acid by Comamonas testosteroni DSM6577. *Chin. J. Cata. 27*, 297−298.
(29) Saini, M., Hong Chen, M., Chiang, C. J., and Chao, Y. P. (2015) Potential production platform of n-butanol in Escherichia coli. *Metab. Eng. 27*, 76−82.
(30) Zhao, S., Jones, J. A., Lachance, D. M., Bhan, N., Khalidi, O., Venkataraman, S., Wang, Z., and Koffas, M. A. G. (2015) Improvement of catechin production in Escherichia coli through combinatorial metabolic engineering. *Metab. Eng. 28*, 43−53.
(31) Thodey, K., Galanie, S., and Smolke, C. D. (2014) A microbial biomanufacturing platform for natural and semisynthetic opioids. *Nat. Chem. Biol. 10*, 837−844.
(32) Han, S. H., Kim, B. G., Yoon, J. A., Chong, Y., and Ahn, J.-H. (2014) Synthesis of Flavonoid O-Pentosides by Escherichia coli through Engineering of Nucleotide Sugar Pathways and Glycosyltransferase. *Appl. Environ. Microbiol. 80* (9), 2754−2762.
(33) Kimura, K., Tomita, N., Uemura, H., Aki, T., Ono, K., and Kamisaka, Y. (2009) Improvement of Stearidonic Acid Production in Oleaginous Saccharomyces cerevisiae. *Biosci., Biotechnol., Biochem. 73*, 1447−1449.
(34) Becker, J. V., Armstrong, G. O., van der Merwe, M. J., Lambrechts, M. G., Vivier, M. A., and Pretorius, I. S. (2003) Metabolic engineering of Saccharomyces cerevisiae for the synthesis of the wine-related antioxidant resveratrol. *FEMS Yeast Res. 4* (1), 79−85.
(35) Planson, A. G., Carbonell, P., Paillard, E., Pollet, N., and Faulon, J. L. (2012) Compound Toxicity Screening and Structure-Activity Relationship Modeling in Escherichia Coli. *Biotechnol. Bioeng. 109* (3), 846−50.
(36) Delépine, B., Libis, V., Carbonell, P., and Faulon, J. L. (2016) Sensi Path: Computer-Aided Design of Sensing-Enabling Metabolic Pathways. *Nucleic Acids Res. 44* (W1), W226−31.
(37) Libis, V., Delépine, B., and Faulon, J. L. (2016) Expanding Biosensing Abilities through Computer-Aided Design of Metabolic Pathways. *ACS Synth. Biol. 5* (10), 1076−85.
(38) Koch, M., Pandi, A., Delépine, B., and Faulon, J. L. (2018) A Dataset of Small Molecules Triggering Transcriptional and Translational Cellular Responses. *Data in Brief 17*, 1374−78.
(39) Lin, G. M., Warden-Rothman, R., and Voigt, C. A. (2019) Retrosynthetic Design of Metabolic Pathways to Chemicals Not Found in Nature. *Curr. Opin. Syst. Biol. 14*, 82−107.
(40) Daylight Chemical Information Systems, Inc. (2008) *Daylight Theory Manual*; https://www.daylight.com/dayhtml/doc/theory/.
(41) Schreck, J. S., Coley, C. W., and Bishop, K. J. M. (2019) Learning Retrosynthetic Planning through Simulated Experience. *ACS Cent. Sci. 5* (6), 970−81.
(42) *Reaxys*, Elsevier Life Sciences, http://www.reaxys.com.
(43) Moretti, S., Martin, O., Van Du Tran, T., Bridge, A., Morgat, A., and Pagni, M. (2016) Meta Net X/MNXref–Reconciliation of Metabolites and Biochemical Reactions to Bring Together Genome-Scale Metabolic Networks. *Nucleic Acids Res. 44* (D1), D523−26.
(44) Heller, S., McNaught, A., Stein, S., Tchekhovskoi, D., and Pletnev, I. (2013) InChI - the Worldwide Chemical Structure Identifier Standard. *J. Cheminf. 5* (1), 7.
(45) Rahman, S. A., Torrance, G., Baldacci, L., Cuesta, S. M., Fenninger, F., Gopal, N., Choudhary, S., et al. (2016) Reaction Decoder Tool (RDT): Extracting Features from Chemical Reactions. *Bioinformatics 32* (13), 2065−2066.
(46) Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008) Exploring Network Structure, Dynamics, and Function Using Network. In *Proceedings of the 7th Python in Science Conference (SciPy2008)* (Varoquaux, G., Vaught, T., and Millman, J., Eds.) pp 11−15.
(47) Maggiora, G. M., and Shanmugasundaram, V. (2004) Molecular Similarity Measures. *Methods Mol. Biol. 275*, 1−50.
(48) Bajusz, D., Rácz, A., and Héberger, K. (2015) Why Is Tanimoto Index an Appropriate Choice for Fingerprint-Based Similarity Calculations? *J. Cheminf. 7*, 20.
(49) The UniProt Consortium (2017) UniProt: The Universal Protein Knowledgebase. *Nucleic Acids Res. 45* (D1), D158−D169.
(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* (13), 1658−59.
(51) Morgat, A., Lombardot, T., Axelsen, K. B., Aimo, L., Niknejad, A., Hyka-Nouspikel, N., Coudert, E., et al. (2017) Updates in Rhea - an Expert Curated Resource of Biochemical Reactions. *Nucleic Acids Res. 45* (D1), D415−18.
(52) Caspi, R., Billington, R., Fulcher, C. A., Keseler, I. M., Kothari, A., Krummenacker, M., Latendresse, M., et al. (2018) The MetaCyc Database of Metabolic Pathways and Enzymes. *Nucleic Acids Res. 46* (D1), D633−39.
(53) Fabregat, A., Jupe, S., Matthews, L., Sidiropoulos, K., Gillespie, M., Garapati, P., Haw, R., et al. (2018) The Reactome Pathway Knowledgebase. *Nucleic Acids Res. 46* (D1), D649−55.
(54) Ebrahim, A., Lerman, J. A., Palsson, B. O., and Hyduke, D. R. (2013) COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC Syst. Biol. 7*, 74.
(55) Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., et al. (2019) PubChem 2019 Update: Improved Access to Chemical Data. *Nucleic Acids Res. 47* (D1), D1102−9.
(56) Monk, J. M., Lloyd, C. J., Brunk, E., Mih, N., Sastry, A., King, Z., Takeuchi, R., et al. (2017) iML1515, a Knowledgebase That Computes Escherichia Coli Traits. *Nat. Biotechnol. 35* (10), 904−8.
(57) Orth, J. D., Fleming, R. M. T., and Palsson, B. O. (2010) Reconstruction and Use of Microbial Metabolic Networks: The Core Escherichia coli Metabolic Model as an Educational Guide. *EcoSal Plus*, DOI: 10.1128/ecosalplus.10.2.1.
(58) Orth, J. D., Conrad, T. M., Na, J., Lerman, J. A., Nam, H., Feist, A. M., and Palsson, B. O. (2011) A Comprehensive Genome-Scale Reconstruction of Escherichia Coli Metabolism–2011. *Mol. Syst. Biol. 7*, 535.

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

(59) Oh, Y. K., Palsson, B. O., Park, S. M., Schilling, C. H., and Mahadevan, R. (2007) Genome-Scale Reconstruction of Metabolic Network in Bacillus Subtilis Based on High-Throughput Phenotyping and Gene Essentiality Data. *J. Biol. Chem. 282* (39), 28791−99.
(60) King, Z. A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J. A., Ebrahim, A., Palsson, B. O., and Lewis, N. E. (2016) BiGG Models: A Platform for Integrating, Standardizing and Sharing Genome-Scale Models. *Nucleic Acids Res. 44* (D1), D515−22.
(61) Sutton, R. S., and Barto, A. G. (1998) *Reinforcement Learning: An Introduction*, MIT Press.
(62) Jeske, L., Placzek, S., Schomburg, I., Chang, A., and Schomburg, D. (2019) BRENDA in 2019: A European ELIXIR Core Data Resource. *Nucleic Acids Res. 47* (D1), D542−49.
(63) Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011) Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res. 12*, 2825−2830.
(64) Köster, J., and Rahmann, S. (2012) Snakemake–a Scalable Bioinformatics Workflow Engine. *Bioinformatics 28* (19), 2520−22.

DOI: 10.1021/acssynbio.9b00447
ACS Synth. Biol. 2020, 9, 157−168

Reinforcement Learning for Bio-Retrosynthesis
- Supplementary Information

Mathilde Koch[1], Thomas Duigou[1], and Jean-Loup Faulon[1,2,3]†

[1] Micalis Institute, INRA, AgroParisTech, Université Paris-Saclay, Jouy-en-Josas, France
[2] iSSB Laboratory, Génomique Métabolique, Genoscope, Institut François Jacob, CEA, CNRS, Univ Evry, Université Paris-Saclay, 91057 Evry, France
[3] SYNBIOCHEM Center, School of Chemistry, University of Manchester, Manchester, UK

†To whom correspondence should be addressed: jean-loup.faulon@inra.fr

The Supplementary Data for this article contains:

- **Supplementary Table 1**: Mean number of rules applying to a compound at various diameters.
- **Supplementary Table 2**: Detailed configuration data for RetroPath RL runs on validation datasets.
- **Supplementary Table 3**: Toxicity biased results.
- **Supplementary Table 4**: Golden dataset structures and references.
- **Supplementary Note 1** (with **Supplementary Figures 1 to 12)**: Detailed parameters explanation and analysis.
- **Supplementary Note 2**: Detailed golden dataset analysis
- **Supplementary Note 3** (with **Supplementary Figure 13**): Database sped-up calculations
-
- **Supplementary Note 4** (with **Supplementary Figure 14**): Extending a previous search
- **Supplementary Data 1**: Golden dataset pathways under json and SBML format.
- **Supplementary Data 2**: Laser database and Metabolic Engineering compounds.
- **Supplementary Data 3**: Laser database and Metabolic Engineering results.
- **Supplementary Data 4**: Targets and supplements for experimentally described pathways.

Supplementary Table 1: Average number of rules that apply to a compound according to the diameter at which the rule is used. This shows that more promiscuous rules (at lower diameters) generate a bigger combinatorial space.

Diameter used	2	4	6	8	10	12	14	16
Average number of rules that apply	797	227	73	35	20	14	11	8

Supplementary Table 2: Detailed configuration data for RetroPath RL runs on validation datasets.

Parameter name	**Golden Default**	**Golden Rescue**	**Laser Default**	**Laser Rescue**
Itermax	10000	10000	50000	10000
Expansion width	10	15	10	10
Time budget (s)	10800 when comparing to RP2, 14400 otherwise	10800 when comparing to RP2, 14400 otherwise	10800	10800
Max depth	7	7	10	7
UCT Policy	Biochemical	Biochemical	Biochemical	Biochemical
UCTK	2	2	2	2
Rollout Policy	Uniform on biochemical score	Uniform on biochemical score	Uniform on biochemical score	Uniform on biochemical score
Max rollout	2	2	2	2
Chemical scoring	Substrate and product	Substrate and product	Substrate and product	Substrate and product
Virtual visits	0	0	0	0
Rule diameters	6, 10, 16	6, 10, 16	6, 10, 16	2, 6, 10, 16
Biological score cut-off	0.3	0.15	0.3	0
Substrate score cut-off	0.3	0.15	0.3	0
Chemical score cut-off	0.3	0.15	0.3	0
Minimal visits count	1	1	1	1
Fire timeout (s)	1	1	1	1
Penalty	-1	-1	-1	-1
Reward	5	5	5	5
Full pathway reward	10	10	10	10
Seed	42	42	42	42

Supplementary Table 3: Toxicity biased results. Comparing results between the default configuration and the one using the toxicity score to bias the search. Total pathway number is the total number of pathways found with the given time and iteration budget. The iteration and rank for the experimental pathway refer to the pathway described in the golden dataset.

Compound	**Total pathway number.** default - *toxicity*	**Iteration for finding the experimental pathway.** default - *toxicity*	**Rank of experimental pathway.** default - *toxicity*
3-methyl-1-butanol	1 - *1*	4 - *4*	1 - *1*
1,4-Butanediol	NA - *NA*	NA - *NA*	NA - *NA*
2-amino-1,3-propanediol	1 - *1*	1 - *1*	1 - *1*
2,5-DHBA	34 - *31*	88 - *97*	4 - *4*
benzyl alcohol	13 - *22*	1815 - *2213*	4 - *8*
caroten	4 - *4*	6508 - *6443*	1 - *1*
cis,cis muconate	11 - *11*	159 - *176*	6 - *6*
glutaric acid	61 - *74*	1114 - *571*	13 - *7*
lycopene	106 - *103*	101 - *96*	1 - *1*
mesaconic acid	14 - *14*	9 - *9*	1 - *1*
naringenin	38 - *40*	408 - *1011*	1 - *9*
N-methylpyrrolinium	9 - *10*	NA - *NA*	NA - *NA*
p-hydroxystyrene	59 - *61*	34 - *33*	1 - *1*
piceatannol	34 - *38*	8581 - *8996*	32 - *36*
pinocembrin	25 - *25*	66 - *55*	2 - *2*
protopanaxadiol	1 - *2*	NA - *NA*	NA - *NA*
styrene	43 - *45*	14 - *13*	2 - *2*
TPA	9 - *10*	NA - *NA*	NA - *NA*
vanillin	34 - *36*	169 - *171*	4 - *4*
violacein	3 - *3*	158 - *158*	3 - *3*

Supplementary Table 4: Golden dataset. This dataset contains the compounds, structures and references used for experimental pathway analysis presented in Results - golden set.

Name	InChI	Reference
3-methyl-1-butanol	InChI=1S/C5H12O/c1-5(2)3-4-6/h5-6H,3-4H2,1-2H3	[1]
1,4-Butanediol	InChI=1S/C4H10O2/c5-3-1-2-4-6/h5-6H,1-4H2	[2, 3]
2-amino-1,3-propanediol	InChI=1S/C3H9NO2/c4-3(1-5)2-6/h3,5-6H,1-2,4H2	[4]
2,5-DHBA	InChI=1S/C7H6O4/c8-4-1-2-6(9)5(3-4)7(10)11/h1-3,8-9H,(H,10,11)	[5]
benzyl alcohol	InChI=1S/C7H8O/c8-6-7-4-2-1-3-5-7/h1-5,8H,6H2	[6]
caroten	InChI=1S/C40H56/c1-31(19-13-21-33(3)25-27-37-35(5)23-15-29-39(37,7)8)17-11-12-18-32(2)20-14-22-34(4)26-28-38-36(6)24-16-30-40(38,9)10/h11-14,17-22,25-28H,15-16,23-24,29-30H2,1-10H3/b12-11+,19-13+,20-14+,27-25+,28-26+,31-17+,32-18+,33-21+,34-22+	[7]
cis,cis muconate	InChI=1S/C6H6O4/c7-5(8)3-1-2-4-6(9)10/h1-4H,(H,7,8)(H,9,10)/p-2/b3-1-,4-2-	[8]
glutaric acid	InChI=1S/C5H8O4/c6-4(7)2-1-3-5(8)9/h1-3H2,(H,6,7)(H,8,9)	[9]
lycopene	InChI=1S/C40H56/c1-33(2)19-13-23-37(7)27-17-31-39(9)29-15-25-35(5)21-11-12-22-36(6)26-16-30-40(10)32-18-28-38(8)24-14-20-34(3)4/h11-12,15-22,25-32H,13-14,23-24H2,1-10H3/b12-11+,25-15+,26-16+,31-17+,32-18+,35-21+,36-22+,37-27+,38-28+,39-29+,40-30+	[10]
mesaconic acid	InChI=1S/C5H6O4/c1-3(5(8)9)2-4(6)7/h2H,1H3,(H,6,7)(H,8,9)/b3-2+	[11]
naringenin	InChI=1S/C15H12O5/c16-9-3-1-8(2-4-9)13-7-12(19)15-11(18)5-10(17)6-14(15)20-13/h1-6,13,16-18H,7H2	[12]
N-methylpyrrolinium	InChI=1S/C5H10N/c1-6-4-2-3-5-6/h4H,2-3,5H2,1H3/q+1	[13]
p-hydroxystyrene	InChI=1S/C8H8O/c1-2-7-3-5-8(9)6-4-7/h2-6,9H,1H2	[14]

piceatannol	InChI=1S/C14H12O4/c15-11-5-10(6-12(16)8-11)2-1-9-3-4-13(17)14(18)7-9/h1-8,15-18H/b2-1+	[15]
pinocembrin	InChI=1S/C15H12O4/c16-10-6-11(17)15-12(18)8-13(19-14(15)7-10)9-4-2-1-3-5-9/h1-7,13,16-17H,8H2/t13-/m0/s1	[16]
protopanaxadiol	InChI=1S/C30H52O3/c1-19(2)10-9-14-30(8,33)20-11-16-29(7)25(20)21(31)18-23-27(5)15-13-24(32)26(3,4)22(27)12-17-28(23,29)6/h10,20-25,31-33H,9,11-18H2,1-8H3/t20-,21+,22-,23+,24-,25-,27-,28+,29+,30+/m0/s1	[17]
styrene	InChI=1S/C8H8/c1-2-8-6-4-3-5-7-8/h2-7H,1H2	[18]
TPA	InChI=1S/C8H6O4/c9-7(10)5-1-2-6(4-3-5)8(11)12/h1-4H,(H,9,10)(H,11,12)	[19]
vanillin	InChI=1S/C8H8O3/c1-11-8-4-6(5-9)2-3-7(8)10/h2-5,10H,1H3	[20]
violacein	InChI=1S/C20H13N3O3/c24-10-5-6-15-12(7-10)14(9-21-15)17-8-13(19(25)23-17)18-11-3-1-2-4-16(11)22-20(18)26/h1-9,21,24H,(H,22,26)(H,23,25)/b18-13+	[21, 22]

Supplementary Note 1 : Parameter Role and Effects

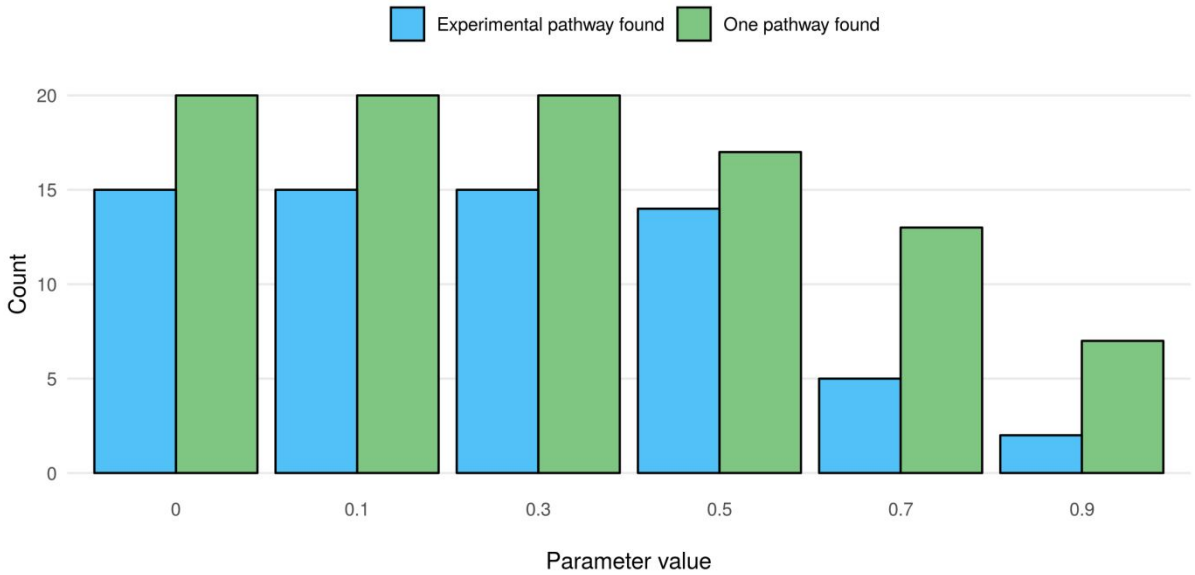
The aim of this supplementary note is to detail the different parameters available in RP3 and their roles and effects.

A number of methods and ideas were taken or inspired from the following master thesis, which uses Monte Carlo Tree Search against a computer game[23].

Rule selection parameters

Biological score

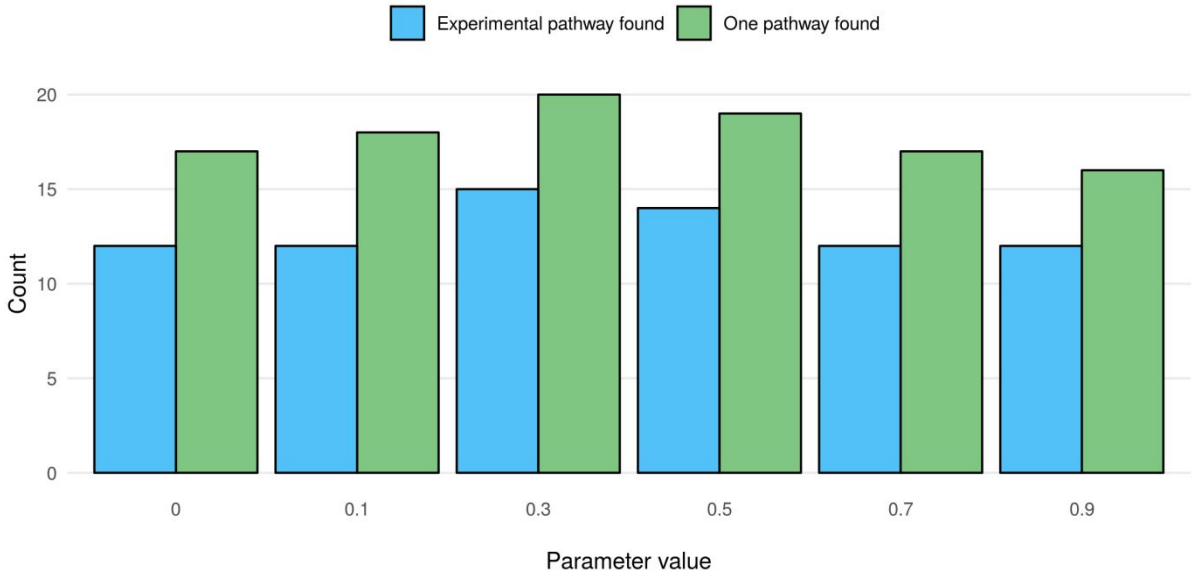
As described in the main text, this score characterises our confidence that a sequence exists to catalyse the reaction of interest. It is normalised between 0 and 1. Using a cut-off on this score removes less trustworthy reaction rules. We see in **Supplementary Figure 1** that results do not vary between using a cut-off from 0 to 0.3, and we start losing pathways of interest when the cut-off is superior or equal to 0.5. A cut-off of 0.3 therefore seems to be a good trade-off between confidence in the existence of a sequence and obtaining enough retrosynthesis results.



Supplementary Figure 1: Impact of biological score cut-off on retrieval performance of RetroPath RL. We compared results between using a biological score cut-off varying between 0 and 0.9. One pathway found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways. Also presented as **Figure 5A**.

Chemical score

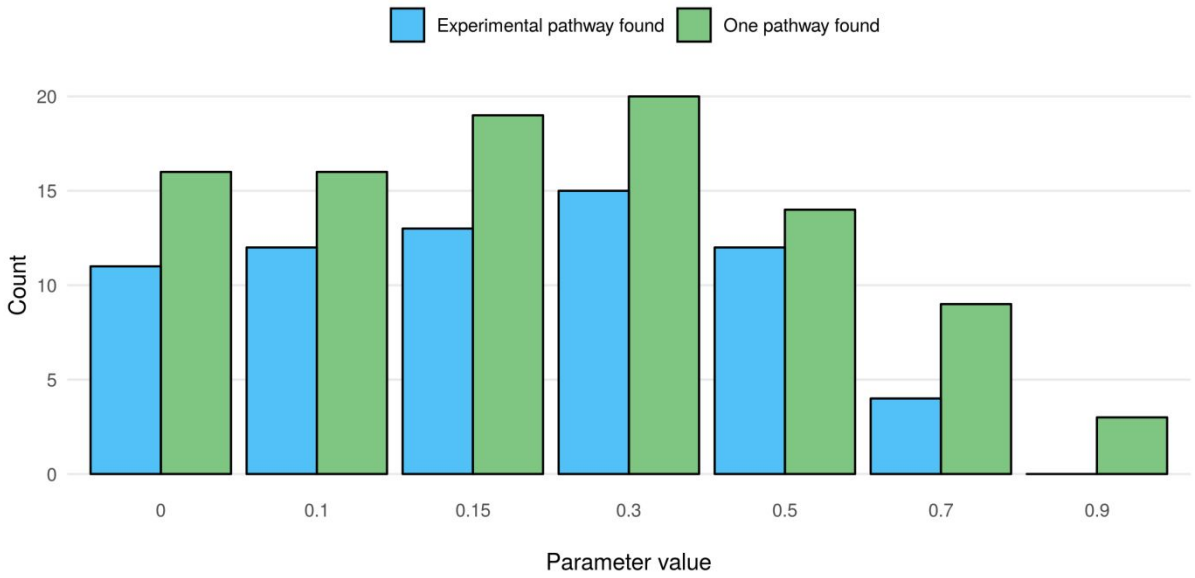
As described in the main text, this score characterizes our confidence that the reaction rule learned on a substrate and product from a database of interest can truly be applied to a new substrate, based on similarity between substrates and products of the native reaction versus the query reaction. We can see in **Supplementary Figure 2** that allowing reactions that are too different leads the tree to explore too diverse pathways, while being too conservative leads to loss of useful reactions. 0.3 therefore seems to be a good cut-off between confidence that the reaction rule does apply to the compound and allowing exploration of the metabolic space.



Supplementary Figure 2: Impact of chemical score cut-off on retrieval performance of RetroPath RL. We compared results between using a chemical score cut-off varying between 0 and 0.9. One pathway found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways. Also presented as **Figure 5B**.

Biological and chemical score

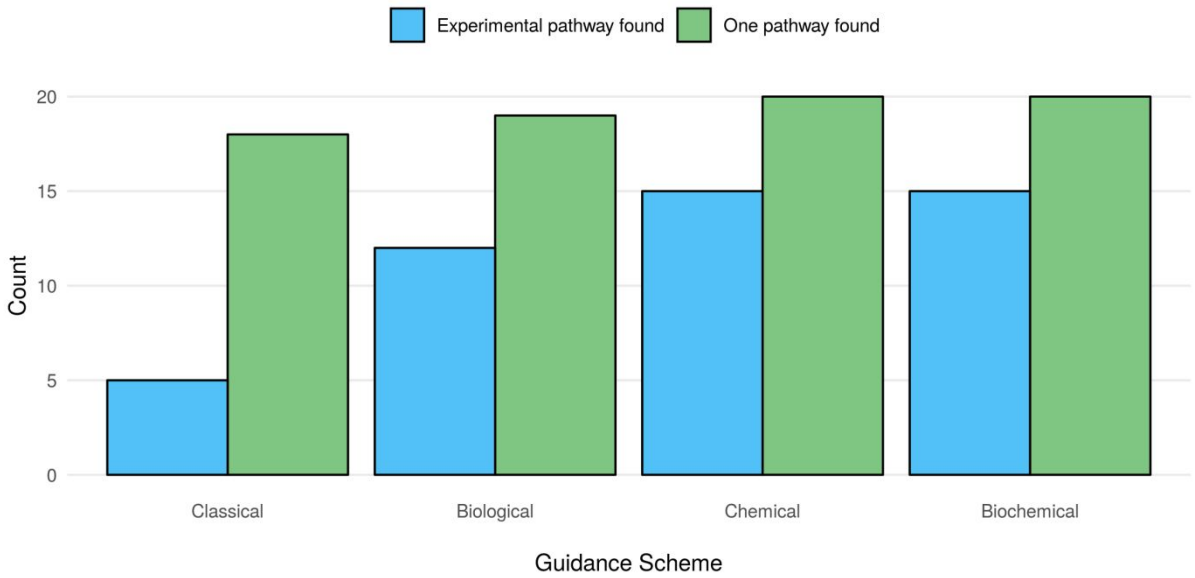
Here we varied both the chemical and biological scores, set at the same value. We can see in **Supplementary Figure 3**, as in Supplementary Figures 1 and 2, that cut-offs of 0.3 provide the best trade-off between exploration and confidence.



Supplementary Figure 3: Impact of biochemical score cut-off on retrieval performance of RetroPath RL. We compared results between using chemical score cut-off and biological score cut-off (set at the same value) varying between 0 and 0.9. One pathway found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways. Also presented as **Figure 5C**.

UCT policy

While these policies are used to tune the exploration/exploitation balance, we modified it to guide the search and see the importance of that guidance on finding results. We can see in **Figure 5D** and **Supplementary Figure 4** the best UCT policy to guide our exploration of the metabolic space is our formula including the biochemical score.

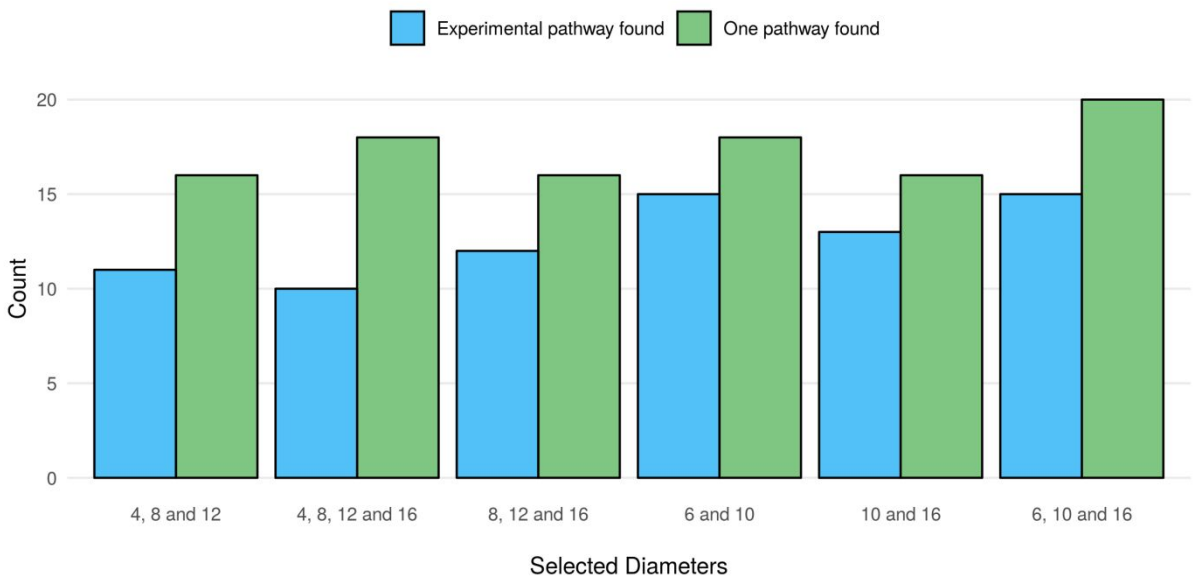


Supplementary Figure 4: Impact of guiding scheme on retrieval performance of RetroPath RL. We compared results between guiding the search based on the Classical UCT formula, a formula guided by Biological scoring, Chemical scoring or Biochemical scoring. One pathway

found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways. Also presented as **Figure 5D**.

Diameters

Diameters characterize the degree of promiscuity we allow in a reaction rule: higher diameters are more specific, while lower diameters are more promiscuous. We found a good trade-off was to allow rules at different levels of promiscuity, using diameters of 6, 10 and 16 (low, medium and high specificity), as shown in **Supplementary Figure 5**.

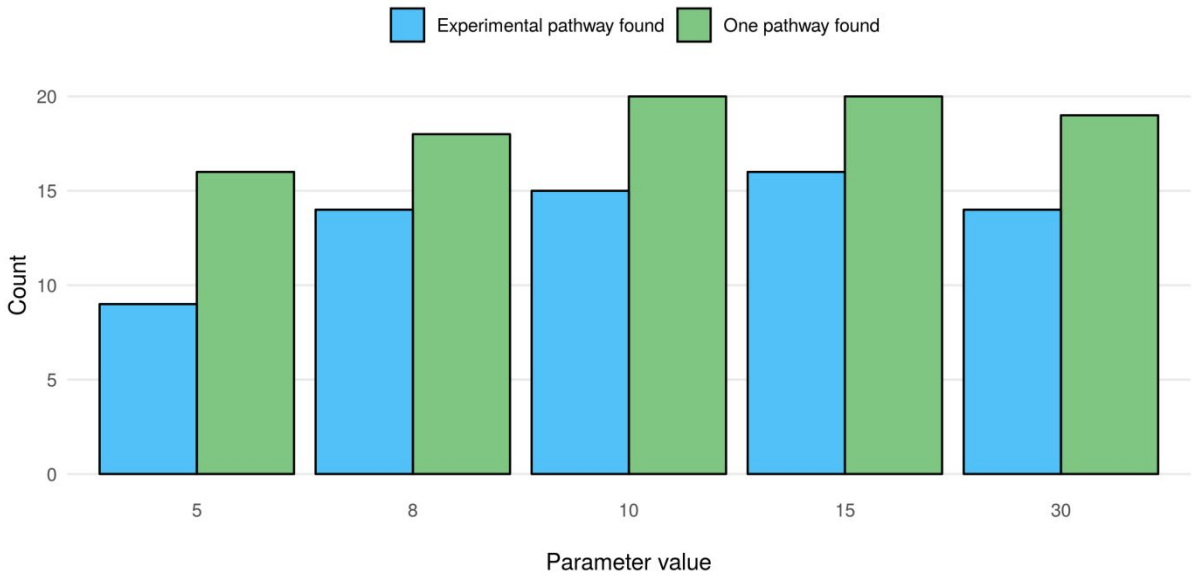


Supplementary Figure 5: Impact of allowed rule diameters on retrieval performance of RetroPath RL. We compared results between using different diameter sets. One pathway found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways.

Exploration parameter

Expansion width

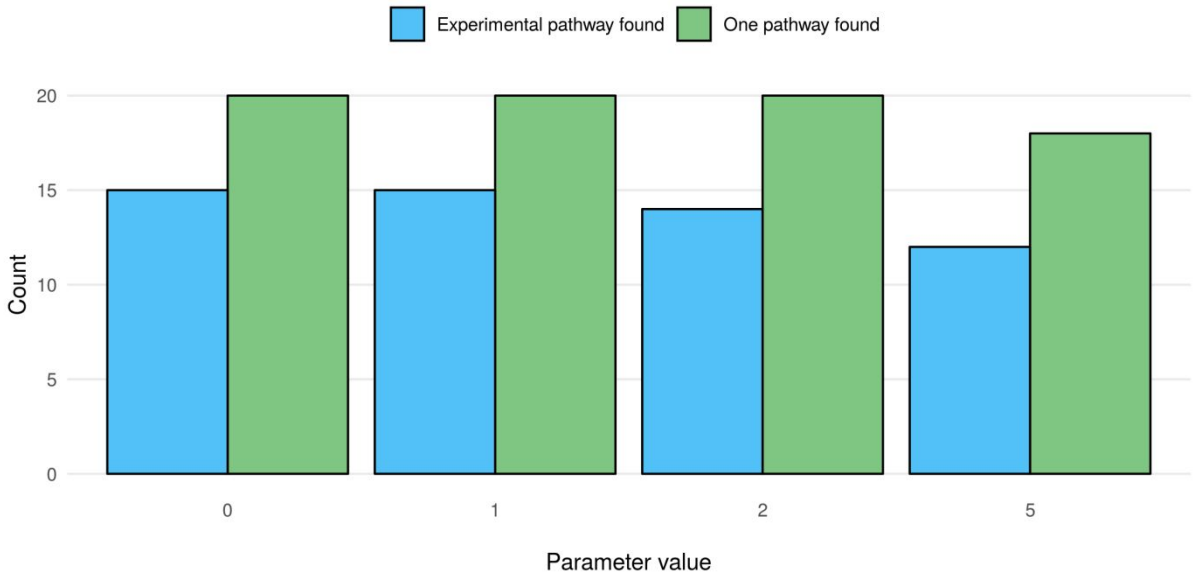
It is the number of children a node is allowed to have. We found 10 and 15 provided a good trade-off between exploration and exploitation, as shown in **Supplementary Figure 6**. We usually tested with 10 children and expanded to 15 for failed compounds.



Supplementary Figure 6: Impact of expansion width on retrieval performance of RetroPath RL. We compared results between using different expansion width (number of allowed children per node). One pathway found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways.

Minimal visit counts

In our implementation, grand-children of a node can only be explored if all his children have had at least minimal_visits visits, where this value was set at 1 in the default settings. This allows mandatory rollout on different branches at least once to favour exploration. We can see from **Supplementary Figure 7** that results are similar when not forcing this exploration with a parameter set to 0.

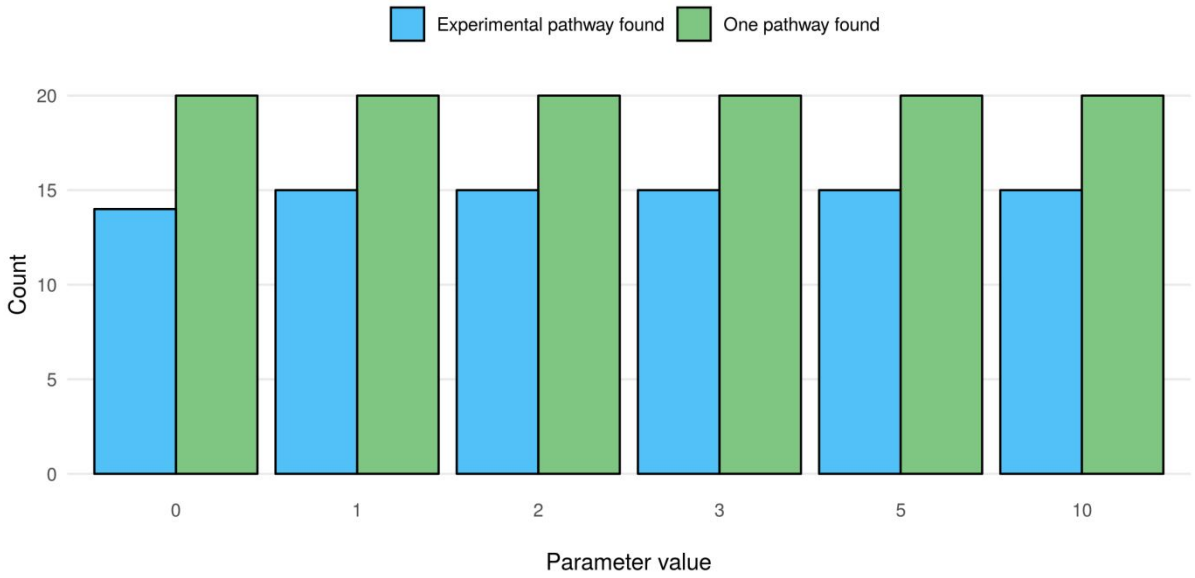


Supplementary Figure 7: Impact of minimal visit counts on retrieval performance of RetroPath RL. We compared results between using different minimal visit count values. One pathway

found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways.

Rollout

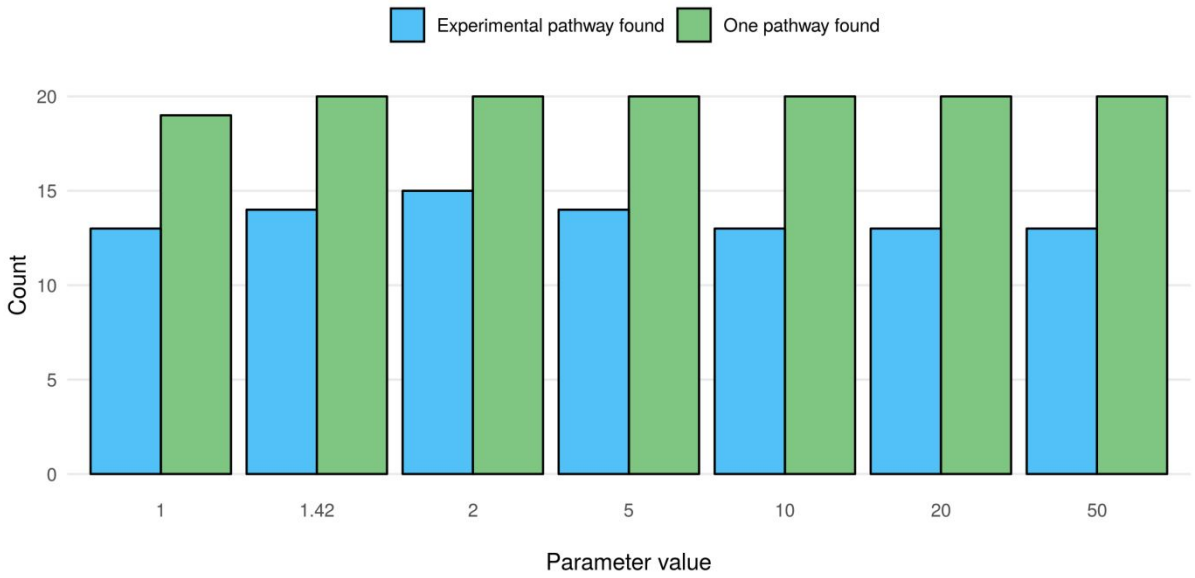
This is the rollout depth: the number of reactions performed before analysing the state and returning the state's reward or penalty. **Supplementary Figure 8** shows that rollout depth does not impact our capacity of finding experimental results on the golden dataset. However, unshown results (considering the iteration at which those results are found) suggest best rollout values are either 2 or 3.



Supplementary Figure 8: Impact of rollout depth on retrieval performance of RetroPath RL. We compared results between using different rollout depths. One pathway found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways.

UCTK

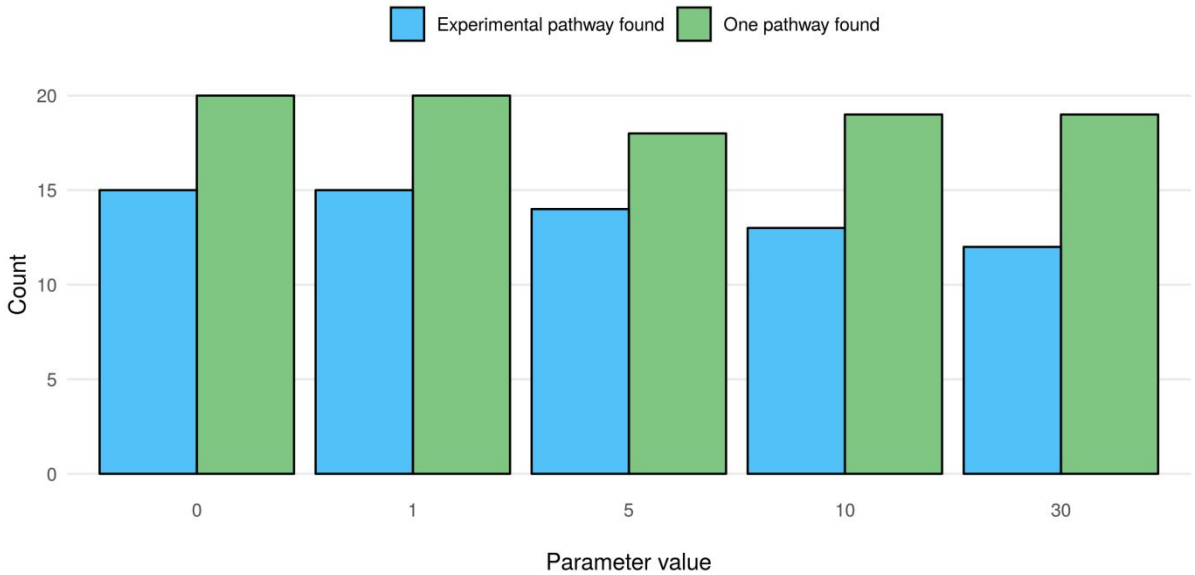
This constant balances the trade-off between exploration and exploitation in the UCT formula. We can see in **Supplementary Figure 9** that the value allowing the best retrieval rate from the golden dataset is a constant of 2.



Supplementary Figure 9: Impact of exploration constant value on retrieval performance of RetroPath RL. We compared results between using different exploration constant values (UCTK). One pathway found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways.

<u>Virtual visits</u>

This is the number of visits a new node starts with. The concept of virtual visits is that giving an initial value to a node will return more stable rollout results as they will be smoothed by a number less close to 0 rather than being very stochastic at low values. We can see this strategy did not give better results in our MCTS for bio-retrosynthesis in **Supplementary Figure 10**.

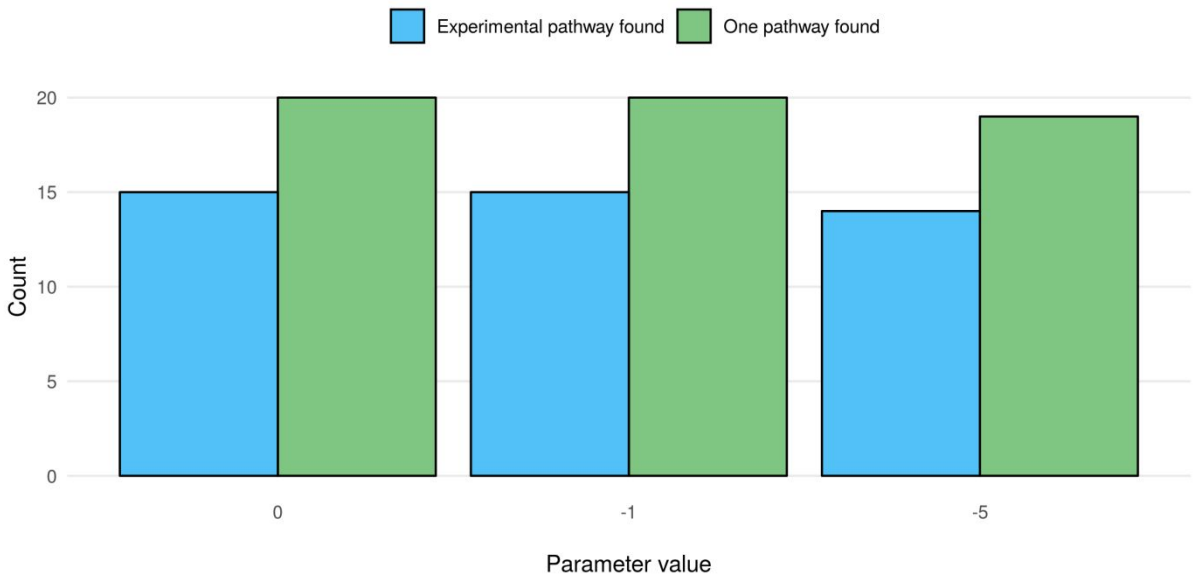


Supplementary Figure 10: Impact of virtual visits on retrieval performance of RetroPath RL. We compared results between using different virtual visits numbers. One pathway found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways.

Solution rewarding

Penalty

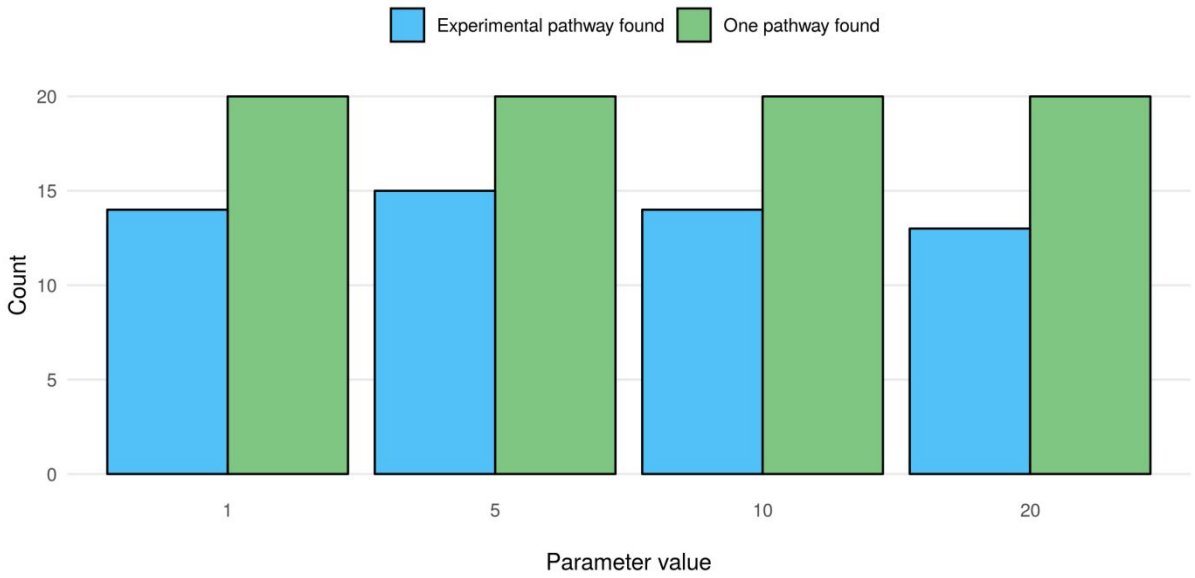
This is the value returned when no compound of the state is within the chassis (including at the end of rollout). We can see in **Supplementary Figure 11** that increasing penalty does not yield better results in our case, and a value of -1 penalises enough the unsuccessful rollout results.



Supplementary Figure 11: Impact of penalty on retrieval performance of RetroPath RL. We compared results between using different penalties. One pathway found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways.

Reward

Reward is the value returned when all compounds of the state are solved, to encourage exploration of the same area of the Tree. We can see from **Supplementary Figure 12** that a value of 5 provides a good trade-off between exploration of other areas of the tree and exploitation of promising regions.



Supplementary Figure 12: Impact of reward on retrieval performance of RetroPath RL. We compared results between using different rewards. One pathway found means that at least one pathway has been predicted. Experimental pathway found means that the experimental pathway is from amongst the predicted pathways.

Other parameters - for other applications

Heavy saving

Saves search Tree state during the search instead of only at the end. Used to analyse Search evolution.

Stop at first result

The search stops once a single result is found.

Fire time-out and standardisation time-out

Timeouts on rule application on a compound.

Organism name

Choose another organism from our predefined list of sinks.

Complementary sink

Adding compounds to the sink as supplements. Can also be used to provide an entirely new sink following the required format.

Other parameters - exploratory

Remark: no detailed comparison was performed in this article on those parameters, contrary to the parameters presented above.

k_RAVE

For balancing of Rapid Action Value Estimation. The idea is to provide moves with results from rollouts elsewhere in the Tree, to give them an initial value. This will decrease in importance as the node itself is visited but provides a fast initial value.

Bias_k

When using bias (for example towards toxicity), how to weight this value.

Progressive bias

When used in conjunction with bias_k, can give an initial value to a node based on various policies: high reward, current state reward, no reward… This also helps initial estimation of the node value rather than rely only on costly rollouts.

Progressive widening

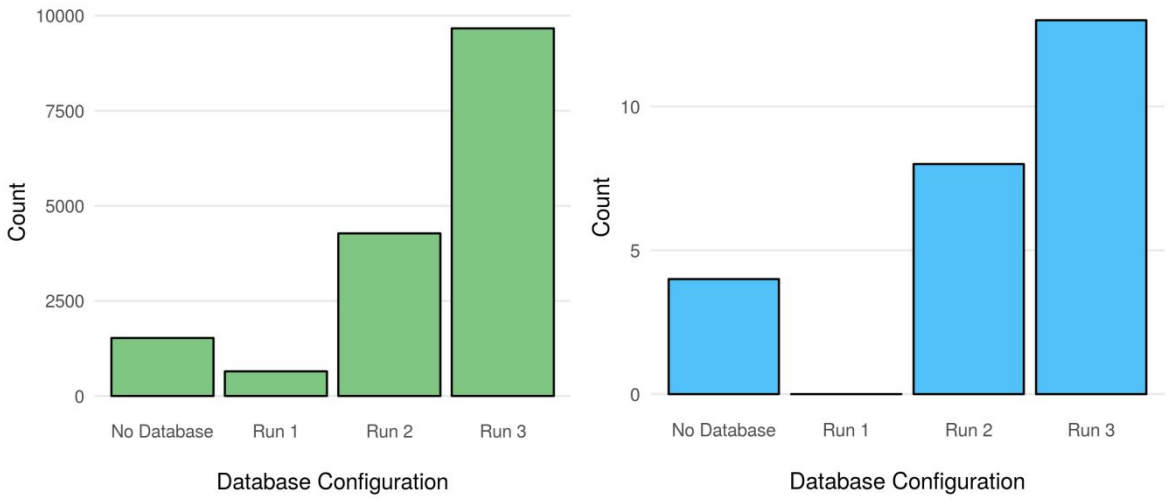
Allow a number of children different according to the number of visits of the node. This is to avoid expanding too much in spaces of the tree search that are actually not interesting.

Supplementary Note 2 : Detailed golden dataset analysis

Results comparing RetroPath RL and RetroPath 2.0 (run with the same set of rules at diameters 6, 10 and 16 and a timeout of 3 hours) on the golden dataset are presented in **Figure 2**. For all compounds, at least one pathway was found with those settings with RetroPath RL, while one compound had no pathway with RetroPath 2.0. For 2-amino-1,3-propanediol, the same core pathway was found, but the identified co-substrate in the first step was different (D-alanine predicted instead of D-glutamate for the experimental pathway, the main substrate being dihydroxyacetone phosphate). (one step different in **Figure 2** - dark blue). For four compounds (TPA, N-methylpyrrolinium, 1,4 BDO and protopanaxadiol), the experimental pathway was not found for different reasons. For TPA, the described experimental pathway in our golden dataset starts from a compound added to the mix, xylene. Running our workflow adding this compound to the sink allows us to find the experimental pathway (**Figure 2** - purple). For protopanaxadiol and N-methylpyrrolinium, we ran the MCTS using a different set of parameters, allowing both to explore more reactions (15 instead of 10) and more tolerance on the scores (cut-offs of 0.15 instead of 0.3). With those new settings we found the experimental pathway for both compounds. For 1,4 BDO, the experimental pathway was not found with these new settings either, but a similar pathway (lacking only one enzymatic step) was found with the default configuration. It transforms 4-hydroxybutyryl Coa into 1,4 BDO without using 4-Hydroxybutyraldehyde, supposedly catalysed by EC number (1.2.1.84: alcohol-forming fatty acyl-CoA reductase, without the alcohol dehydrogenase step from the literature example). The rest of this pathway is identical to the experimental pathway (one step lacking in **Figure 2** - light blue).

Supplementary Note 3 : Database speed-up calculations

When running a rule-based retrosynthetic algorithm, the most time and power consuming steps are rule application steps which require subgraph matching, as this is an NP hard subgraph isomorphism problem. We implemented a NoSQL database that allows for a frequent user to store rule application calculations. To allow for a fair comparison between runs and algorithms, it was not activated in the results presented previously. However, when this feature is active, the results of the first rule application on a compound is stored in the database. When the same calculation is encountered in a later run of RetroPath RL, results are retrieved from the database. This allows for faster runs of the algorithm and therefore larger and deeper exploration within the same time budget. For example, we ran the TPA retrosynthetic search 4 times: without the database, and with the database for the first, second and third time, allowing 1 hour and 100 000 iterations at each step. We present in **Supplementary Figure 13** the number of iterations performed in 1 hour, as well as the number of pathways found. While we can see the first run with the database is not as efficient as the run without it (reaches less iterations and does not find a pathway), we can see that having filled the database allows for more exploration of the tree in runs 2 and 3, where in the same allocated time more iterations are performed and more pathways found.



Supplementary Figure 13: Database sped-up retrosynthetic search. We compared results between a classical run where computation is performed on the fly versus storing results in a Database. **A)** Reached iteration is the number of iterations performed by the algorithm in the given timeframe (1hour). **B)** Found pathways is the number of found pathways per run.

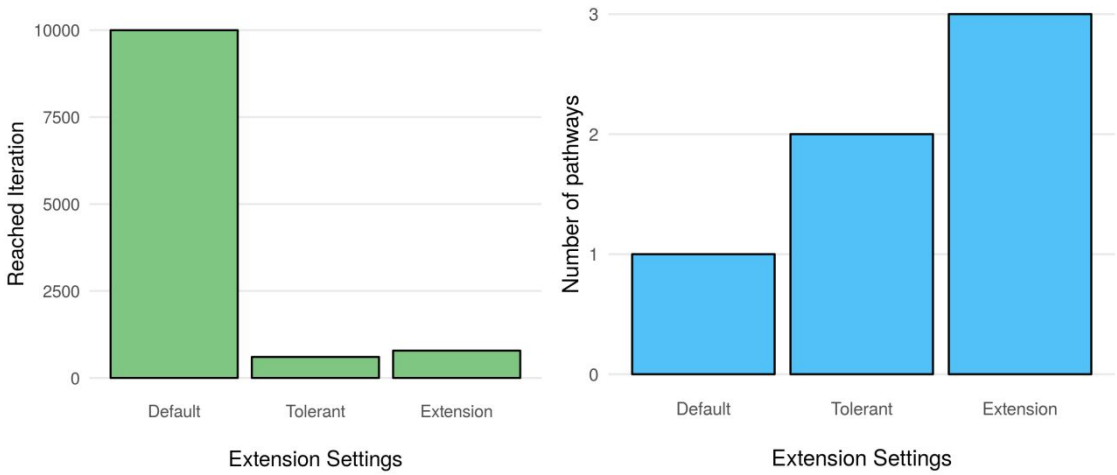
Method: Rule calculation cache using a NoSQL database

Rule calculation can be optionally cached into a NoSQL database in order to optimise the running time of RetroPath RL. We released this cache system as an optional python package named “rp3_cache” that is available on GitHub at https://github.com/brsynth/rp3_cache. Technically, the cache system relies on the Mongo DB database (“Mongo DB” n.d.) that is embedded into a Docker (“Docker” n.d.) container to make the implementation agnostic of the operating system.

Supplementary Note 4 : Extending a previous search

Another feature of interest for expert users is the possibility to extend a previously run tree. For example, considering the example of protopanaxadiol, the experimental pathway was not found with our default settings, but was with more tolerant settings (15 children instead of 10 and a score cut-off of 0.15 instead of 0.3). However, instead of starting from scratch and losing the previously made calculations, it is possible to restart the search from a saved tree, with more tolerant settings. The results after running for 4h and 10 000 iterations are presented in **Supplementary Figure 14**. With the default settings, the iteration budget is spent finding only 1 pathway (not the experimental one). Using more tolerant settings is slower (604 iterations are performed in the allotted time) and 2 pathways are found. Extending the original tree allows for performing more iterations (786) and finding one more pathway, when compared to starting from scratch.

Therefore, we advise a user not satisfied with the results returned by RetroPath RL – be it that they are unreasonable given his expert knowledge or that he wants additional alternative pathways – to restart the search with smaller scoring cut-offs and more children per node (expansion_width parameter). We also advise him to use the database presented in **Supplementary Note 3** so that he saves chemical results even if he wants to restart the run from scratch.



Supplementary Figure 14: Extending a previous search. We compared results between default settings, more tolerant settings and extending the saved searched. **A)** Reached iteration is the number of iterations performed by the algorithm in the given timeframe (1hour). **B)** Found pathways is the number of found pathways per run.

Method: Tree extension

The tree extension procedure starts with a tree containing search results. If n children were allowed in the first run and the extension allows m more children, up to n + m children can be found for a given node. Node scores and visit counts are re-initialised, and nodes are first flagged for extension then extended when they are first visited in the new search.

Supplementary references

(1) Connor, M.R., Liao, J.C. (2008) Engineering of an Escherichia Coli Strain for the Production of 3-Methyl-1-Butanol. *Appl. Environ. Microbiol*. 74 (18): 5769–75.

(2) Hwang, H.J., Park, J.H., Kim, J.H., Kong, M.K., Kim, J.W., Park, J.W., Cho, K.M., Lee, P.C. (2014) Engineering of a Butyraldehyde Dehydrogenase of Clostridium Saccharoperbutylacetonicum to Fit an Engineered 1,4-Butanediol Pathway in Escherichia Coli. *Biotechnol. Bioeng*. 111 (7): 1374–84.

(3) Yim, H., Haselbeck, R., Niu, W., Pujol-Baxley, C., Burgard, A., Boldt, J., Khandurina, J., *et al*. (2011) Metabolic Engineering of Escherichia Coli for Direct Production of 1,4-Butanediol. *Nat. Chem. Biol.* 7 (7): 445–52.

(4) Luo, Y., Zhao, Q., Liu, Q., Feng, Y. (2019) An Artificial Biosynthetic Pathway for 2-Amino-1,3-Propanediol Production Using Metabolically Engineered Escherichia Coli. *ACS Synth. Biol.* 8 (3): 548–56.

(5) Shen, X., Wang, J., Gall, B.K., Ferreira, E.M., Yuan, Q., Yan, Y. (2018) Establishment of Novel Biosynthetic Pathways for the Production of Salicyl Alcohol and Gentisyl Alcohol in Engineered Escherichia Coli. *ACS Synth. Biol.* 7 (4): 1012–17.

(6) Pugh, S., McKenna, R., Halloum, I., Nielsen, D.R. (2015) Engineering Escherichia Coli for Renewable Benzyl Alcohol Production. *Metab. Eng. Commun.* 2: 39–45.

(7) Yang, J., Guo, L. (2014) Biosynthesis of β-Carotene in Engineered E. Coli Using the MEP and MVA Pathways. *Microb. Cell Fact*. 13: 160.

(8) Lin, Y., Sun, X., Yuan, Q., Yan, Y. (2014) Extending Shikimate Pathway for the Production of Muconic Acid and Its Precursor Salicylic Acid in Escherichia Coli. *Metab. Eng.* 23: 62–69.

(9) Park, S.J., Kim, E.Y., Noh, W., Park, H.M., Oh, Y.H., Lee, S.H., Song, B.K., Jegal, J., Lee, S.Y. (2013) Metabolic Engineering of Escherichia Coli for the Production of 5-Aminovalerate and Glutarate as C5 Platform Chemicals. *Metab. Eng.* 16: 42–47.

(10) Ma, T., Shi, B., Ye, Z., Li, X., Liu, M., Chen, Y., Xia, J., Nielsen, J., Deng, Z., Liu, T. (2019) Lipid Engineering Combined with Systematic Metabolic Engineering of Saccharomyces Cerevisiae for High-Yield Production of Lycopene. *Metab. Eng.* 52: 134–42.

(11) Wang, J., Zhang, K. (2015) Production of Mesaconate in Escherichia Coli by Engineered Glutamate Mutase Pathway. *Metab. Eng.* 30: 190–96.

(12) Santos, C.N.S., Koffas, M., Stephanopoulos, G. (2011) Optimization of a Heterologous Pathway for the Production of Flavonoids from Glucose. *Metab. Eng.* 13 (4): 392–400.

(13) Ping, Y., Li, X., Xu, B., Wei, W., Wei, W., Kai, G., Zhou, Z., Xiao, Y.. (2019) Building Microbial Hosts for Heterologous Production of N -Methylpyrrolinium. *ACS Synth. Biol.* 8 (2): 257–63.

(14) Qi, W.W., Vannelli, T., Breinig, S., Ben-Bassat, A., Gatenby, A.A., Haynie, S.L., Sariaslani, F.S. (2007) Functional Expression of Prokaryotic and Eukaryotic Genes in Escherichia Coli for Conversion of Glucose to P-Hydroxystyrene. *Metab. Eng.* 9 (3): 268–76.

(15) Shrestha, A., Pandey, R.P., Sohng, J.K. (2019) Biosynthesis of Resveratrol and Piceatannol in Engineered Microbial Strains: Achievements and Perspectives. *Appl. Microbiol. Biotechnol*. 103 (7): 2959–72.

(16) Kim, B.G., Lee, H., Ahn, J.H. (2014) Biosynthesis of Pinocembrin from Glucose Using Engineered Escherichia Coli. *J. Microbiol. Biotechnol.* 24 (11): 1536–41.

(17) Wu, Y., Xu, S., Gao, X., Li, M., Li, D., Lu, W. (2019) Enhanced Protopanaxadiol Production from Xylose by Engineered Yarrowia Lipolytica. *Microb. Cell Fact*. 18 (1): 83.
(18) McKenna, R., Nielsen, D.R. (2011) Styrene Biosynthesis from Glucose by Engineered E. Coli. *Metab. Eng.* 13 (5): 544–54.
(19) Bramucci, M.G., McCutchen, C.M., Nagarajan, V., Thomas, S.M. (2001) Microbial production of terephthalic acid and isophthalic acid. USPTO 6187569 B1. *US Patent*, issued 2001.
(20) Kunjapur, A.M., Tarasova, Y., Prather, K.L.J. (2014) Synthesis and Accumulation of Aromatic Aldehydes in an Engineered Strain of Escherichia Coli. *J. Am. Chem. Soc*. 136 (33): 11644–54.
(21) Hoshino, T. (2011) Violacein and Related Tryptophan Metabolites Produced by Chromobacterium Violaceum: Biosynthetic Mechanism and Pathway for Construction of Violacein Core. *Appl. Microbiol. Biotechnol*. 91 (6): 1463–75.
(22) Jones, J.A., Vernacchio, V.R, Lachance, D.M., Lebovich, M., Fu, L., Shirke, A.N., Schultz, V.L., Cress, B., Linhardt, R.J., Koffas, M.A.G. (2015) ePathOptimize: A Combinatorial Approach for Transcriptional Balancing of Metabolic Pathways. *Sci. Rep.* 5: 11301.
(23) Kozelek, T. 2009. Methods of MCTS and the Game Arimaa. Edited by Jan Hric. Master, Charles University in Prague Faculty of Mathematics and Physics.