Combining Evolutionary Algorithms With Reaction Rules Towards Focused Molecular Design

João Correia
jfscorreia95@gmail.com
CEB - Centre of Biological Engineering, University of Minho
Braga, Portugal
LABBELS - Associate Laboratory
Braga/Guimarães, Portugal

Vítor Pereira
vpereira@ceb.uminho.pt
CEB - Centre of Biological Engineering, University of Minho
Braga, Portugal
LABBELS - Associate Laboratory
Braga/Guimarães, Portugal

Miguel Rocha
mrocha@di.uminho.pt
CEB - Centre of Biological Engineering, University of Minho
Braga, Portugal
LABBELS - Associate Laboratory
Braga/Guimarães, Portugal

ABSTRACT

Designing novel small molecules with desirable properties and feasible synthesis continues to pose a significant challenge in drug discovery, particularly in the realm of natural products. Reaction-based gradient-free methods are promising approaches for designing new molecules as they ensure synthetic feasibility and provide potential synthesis paths. However, it is important to note that the novelty and diversity of the generated molecules highly depend on the availability of comprehensive reaction templates. To address this challenge, we introduce ReactEA, a new open-source evolutionary framework for computer-aided drug discovery that solely utilizes biochemical reaction rules. ReactEA optimizes molecular properties using a comprehensive set of 22,949 reaction rules, ensuring chemical validity and synthetic feasibility. ReactEA is versatile, as it can virtually optimize any objective function and track potential synthetic routes during the optimization process. To demonstrate its effectiveness, we apply ReactEA to various case studies, including the design of novel drug-like molecules and the optimization of pre-existing ligands. The results show that ReactEA consistently generates novel molecules with improved properties and reasonable synthetic routes, even for complex tasks such as improving binding affinity against the PARP1 enzyme when compared to existing inhibitors.

CCS CONCEPTS

• **Computing methodologies → Genetic programming**.

KEYWORDS

evolutionary algorithms, drug discovery, reaction rules

ACM Reference Format:
João Correia, Vítor Pereira, and Miguel Rocha. 2023. Combining Evolutionary Algorithms With Reaction Rules Towards Focused Molecular Design. In *Genetic and Evolutionary Computation Conference (GECCO '23), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583131.3590413

GECCO '23, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0119-1/23/07…$15.00
https://doi.org/10.1145/3583131.3590413

1 INTRODUCTION

Throughout history, humans have been searching and cataloging compounds, studying their effects on biological systems in an effort to find products that can improve quality of life. Natural products (NPs) which are produced by living organisms, such as bacteria, fungi, and plants have long been a rich source of drug candidates, with many of the most essential drugs in use today, including antibiotics, anticancer agents, and anti-inflammatory drugs being NPs or based on NPs. The unique and vast chemical diversity of NPs, which have been optimized through the natural evolution process to serve specific biological functions, makes them an ideal source for drug discovery, as they are enriched with bioactive molecules that span a broader range of the chemical space when compared to synthetic small molecules [24].

Discovering a drug is a long and expensive process, typically taking over 8 years and costing between 314 million to 2.8 billion US dollars [43]. For decades, drugs have been discovered by searching through libraries of biologically active natural and synthetic chemical molecules. However, the recent adoption of computational methods, such as computer-aided drug design (CADD) techniques, into the drug discovery process has resulted in a reduction of both time and cost compared to traditional trial-and-error experimentation. CADD techniques are mainly used for the rapid assessment of chemical libraries to guide and accelerate the early stages of developing new active compounds. These techniques include virtual screening, designing virtual libraries, optimizing leads, and designing new compounds from scratch.

CADD techniques can generally be grouped into two main types: ligand-based drug design (LBDD) and structure-based drug design (SBDD) [46]. LBDD involves the identification of small molecules (ligands) that can bind and modulate the activity of a particular target protein, such as a receptor or enzyme. LBDD relies on the physicochemical properties of known ligands and does not consider the three-dimensional structure of the target protein. However, LBDD is limited by the availability of known ligands for a given target protein, which may not always exist. In contrast, SBDD involves using the three-dimensional structure of a protein or other biological target to design and develop small molecule drugs that can bind to and inhibit or activate the target.

In many CADD projects, virtual objective functions are used to predict properties of molecules, such as their biological activity and ADME (Absorption, Distribution, Metabolism, and Excretion) properties. In virtual screening (VS), these objective functions are evaluated for all molecules in a virtual library to identify the most

promising ones. Typically, the molecules being screened are either commercially available or have well-defined synthetic routes allowing for a quick transition from *in silico* to *in vitro* studies. However, these libraries only represent a very small and biased portion of the drug-like chemical space leading to a lack of chemical novelty.

De novo drug design partially solves the chemical novelty problem by designing novel structures that provide a desired biological response, while maintaining certain pharmacokinetic properties. However, navigating and sampling the vast possible chemical space of drug-like molecules in an efficient way is not a trivial task. CADD techniques face the challenge of finding optimal solutions, finding a trade-off between exploring global solutions and exploiting local optimum, as there may be many good regions of the chemical space.

Over the last decades, gradient-based and gradient-free molecular optimization methods have received attention for *de novo* drug design. In gradient-free approaches, the molecular generation is guided towards optimal molecules by the use of population-based stochastic optimization algorithms, such as evolutionary algorithms (EAs) and swarm intelligence [12, 20, 22, 36, 40, 42]. On the other hand, gradient-based molecular optimization approaches are based on deep learning (DL) architectures, such as variational autoencoders, generative adversarial networks, recurrent neural networks, and reinforcement learning [18, 23, 29, 47]. These models, once trained on large datasets of chemical structures, are capable of sampling the learned chemical space.

Regarding the level of specificity of molecular structure generation, gradient-free tools can be divided into three main groups: atom-based, fragment-based, and reaction-based. Atom-based approaches act by applying simple atom/bond level operations such as adding, removing, and replacing individual atoms or bonds from the molecules. In theory, these types of operations are able to generate every possible structure resulting in high novelty and diversity of the generated molecules. However, these approaches can produce invalid or less accessible structures, so it is essential to control these properties during the generation process.

Fragment-based approaches use fragmentation schemes and retrosynthetic disconnections to generate new molecules by adding, replacing, or removing groups of atoms. The novelty and diversity of the resulting molecules depend on the initial set of fragments, and their size and number can control the level of space exploration. However, like atom-based approaches, the validity and synthetic accessibility of the molecules can still be problematic.

Reaction-based approaches use libraries of *in silico* chemical reaction rules to generate new molecules by applying them to a set of reactants. This approach can produce highly novel and diverse molecules in fewer steps by making significant structural changes. However, the effectiveness of this method can be limited by the small number of available reaction templates, which may hinder its ability to explore unknown parts of chemical space and reduce the diversity of the generated molecules. Both atom, fragment, and reaction-based gradient-free approaches have been demonstrated to be effective in many studies (Table 1).

In this study, we introduce ReactEA, a modular and problem-agnostic evolutionary CADD approach that utilizes biochemical reaction rules to manipulate molecules. To optimize user-specified objective functions, a suite of EAs from the jMetalPy framework [4] is employed. The initial population of seed molecules is used to generate a new population, and the fitness of these new molecules is calculated using the objective functions. The top-scoring molecules then proceed to the next generation, and this process is repeated until a stopping criterion is met. ReactEA was used to optimize existing inhibitors and design novel molecules with various objectives, such as drug-likeliness, synthetic accessibility, similarity to a target molecule, and docking affinity against the PARP1 enzyme.

2 MATERIALS AND METHODS

2.1 Biochemical Reaction Rules

In computational chemistry, Reaction Rules are templates that encode the conversion of reactants into products. They identify substructures in the reactants based on a given chemical reaction scheme, enabling researchers to evaluate potential reactions and generate new product molecules. This process can easily be automated using several chemoinformatics tools like RDKit [26]. Figure 1 shows an example of an aromatic decarboxylation reaction rule (Fig. 1 A) and two reactions used to create the rule (Fig. 1 B).

Although the ReactEA framework is compatible with any set of reaction rules, in this work we focus specifically on exploring rules encoding biochemical transformations to enable the exploitation of sustainable cell factories for the production of natural drugs. The used reaction rules were sourced from two primary databases: the RetroRules database [11] and the Metabolic In Silico Network Expansion (MINE) Databases [37]. RetroRules includes 350,224 reaction rules, covering over 15,000 biochemical transformations. These rules vary in enzyme specificity and consider the atomic environment around the reaction center at different diameters (2 to 16, increments of 2). By predicting *de novo* reactions of promiscuous enzymes, this approach expands natural chemical diversity. In this study, only 13,035 reaction rules that can be expressed in the forward direction and with a diameter of 2 (the most permissive) were used.

The MINE databases include known and predicted metabolites generated with a comprehensive set of 9,914 hand-curated reaction rules. The reaction rules from MINE were combined with those from RetroRules to create a comprehensive set of 22,949 biochemical reaction rules. These rules are represented in the SMARTS notation, which can be used by open-source chemoinformatics tools. The validity of each rule was confirmed with the RDKit chemoinformatics toolkit.

2.2 Framework Overview

The general workflow starts with the definition of the problem, which includes defining the objective functions that are being minimized/ maximized, the constraints that the solutions must satisfy (if any), and the representation of the solutions. This determines the search space that the algorithm will explore and the criteria by which the solutions will be evaluated.

ReactEA is designed to optimize molecular properties by exploring the chemical space of NPs using biochemical reaction rules. The framework starts by receiving the molecular properties to optimize and an initial population of molecules that will serve as the seed for the next generations. First, the initial population of molecules is initialized and evaluated. At every generation, a selection operator retrieves the mating pool from the solution list (the population)

Table 1: Examples of the different gradient-free atom, fragment, and reaction-based methods.

Method	Molecule construction method	Evolutionary technique
Kawai *et al.* [19]	Atom-based	Genetic Algorithm
iSyn [28]	Reaction-based	Genetic Algorithm
GB-GA [17]	Atom-based	Genetic Algorithm
MolFinder [22]	Atom-based	Conformational Space Annealing
AutoGrow4 [36]	Reaction-based	Genetic Algorithm
EvoMol [27]	Atom-based	Genetic Algorithm
LEADD [20]	Fragment-based	Genetic Algorithm
ChemGE [44]	Fragment-based	$(\mu + \lambda)$ Evolutionary Strategy
MSO [42]	Atom-based	Particle Swarm Optimization
MOARF [14]	Fragment-based	Multi-objective Evolutionary Algorithm
CReM [33]	Fragment-based	Stochastic exploration

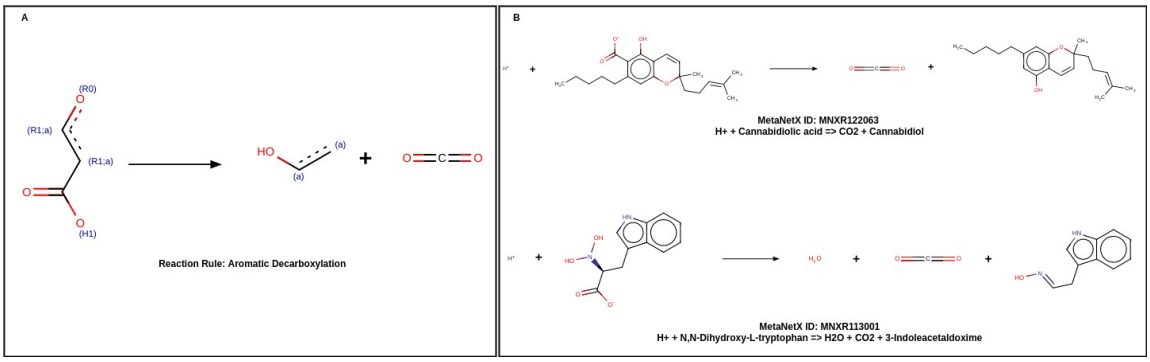

Figure 1: Reaction rule encoding an Aromatic Decarboxylation transformation (A) and original reactions (B), in which a carboxylic acid group (-COOH) is removed from the compound.

for reproduction. A mutation operator is then applied to yield a new list of solutions (the offspring). The solutions of this offspring population are evaluated using the specified objective functions, and a replacement strategy is applied to update the population for the next generation. This process is repeated until a certain stopping criterion is met (e.g. maximum number of generations or mean fitness of all individuals). A general overview of the ReactEA framework is presented in Figure 2.

ReactEA is highly modular and extensible, making it easy for users to customize and extend its functionalities. It provides an interface for molecular property optimization using single-objective (SO) and multi-objective (MO) EAs. This allows users to quickly and easily experiment with different configurations without having to write complex code. Additionally, ReactEA includes a wide range of pre-defined reaction rules that serve as mutation operators to generate diverse molecules with optimized properties.

2.3 Initial population

The initial population serves as the starting point for the search for optimal solutions. Its definition plays an important role in determining the diversity and novelty of the compounds generated. A diverse initial population allows for a wider exploration of potential solutions, increasing the chances of finding multiple optimal solutions. In ReactEA, the initial population is defined by the user. This can be, for instance, a set of available precursors in a particular organism, a diverse set of molecular fragments for a *de novo* design experiment, or known ligands for lead optimization. The framework accepts molecules represented as SMILES strings [41], which are then transformed into RDKit Mol objects. All molecular operations, including the application of the Mutation operator, are performed using RDKit.

2.4 Population Fitness

In an EA, the fitness of a solution is a measure of its ability to solve the problem at hand, being determined by evaluating its performance on pre-defined objective functions.

ReactEA can be used in both SO and MO problems. For that, a set of predefined objective functions, such as octanol–water partition coefficient (logP), drug-likeliness, molecular weight range, number of large rings, and stereoisomer count are provided. However, its modular design allows for the optimization of virtually any objective function, allowing those with computational background to easily implement their own objective functions of interest.

To implement a custom objective function, the user needs to implement a function that calculates some property of interest, specify if it is a maximization or minimization problem, and provide the worst possible fitness for invalid molecules. Additionally, MO functions can be combined into a single one through the use of a

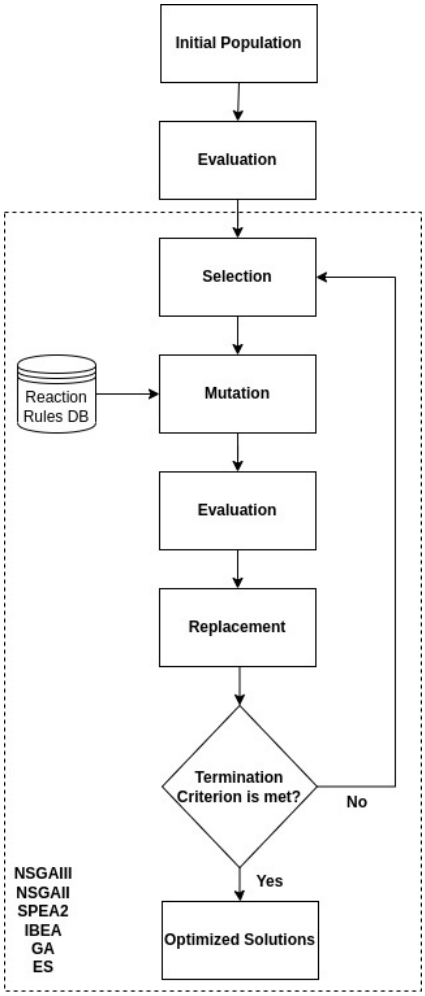

Figure 2: General outline of the ReactEA workflow.

weighted aggregated sum. In this case, a list of weights for each objective function must also be provided.

2.5 Selection Operator

The selection operator is used to choose which solutions will be used to create the next generation of solutions. There are different methods that can be used for selecting the mating population in EAs. ReactEA supports a wide range of selection operators as provided by the jMetalPy framework including a variety of Random, Rank, Roulette, and Tournament selection strategies. The best selection method to use depends on the specific problem being solved, while the different algorithms also work better with different selection schemes.

2.6 Mutation Operator

The Mutation Operator in ReactEA introduces variations into the population by creating novel compounds from the existing ones, by utilizing *in silico* biochemical reactions, executed using RDKit. Given a parent molecule, the process starts by randomly selecting a reaction rule from a predefined set of rules and attempting to apply it to the parent. This step is repeated until a successful reaction occurs or until a maximum number of attempts is reached. The resulting product molecules (offspring) are evaluated, and one of the most similar ones, within a specified range, is selected. This strategy avoids selecting byproducts or undesirable reaction outcomes, such as water or carbon dioxide molecules.

An illustration of the mutation operation is presented in Figure 3, where three different reaction rules are applied to a glycerol molecule, resulting in different product molecules. This approach ensures the validity and higher synthetic feasibility of the generated molecules, while also providing the ability to track the lineage of any mutant molecule to identify its origin and propose potential synthetic routes.

2.7 Replacement of solutions

In an EA, the replacement of solutions refers to the process of updating the current population with new solutions. The replacement strategy used can vary based on the specific problem being solved and the desired outcome of the algorithm. ReactEA offers a range of replacement strategies for its various EAs. Some algorithms replace the entire population with a new set of solutions generated from the current population, which helps to maintain diversity and avoid being trapped in a local optimum. On the other hand, some strategies only replace a portion of the population, striking a balance between exploring new solutions and exploiting the best solutions found so far. More complex algorithms may involve complex variations of these strategies. In addition to these strategies, some EAs utilize the concept of elitism, preserving the best-performing solutions from one generation to the next. This helps to ensure that the best solutions are not lost.

2.8 Termination Criterion

The termination criterion determines when the algorithm comes to a halt. The criterion ensures that the EA produces results within a reasonable time frame. The specific criterion used will vary depending on the optimization problem and the requirements of the application. When choosing a termination criterion, it is crucial to balance exploration of the search space and finding good solutions within a reasonable time. The criterion should be well-defined and easily evaluated during the course of the algorithm. ReactEA implements several criteria, including a maximum number of generations, a maximum run time, reaching a predetermined quality indicator, and a mean value of fitness of all objectives. The termination module can be easily extended to implement other termination criteria as needed.

2.9 Evolutionary Algorithms

Choosing the right EA to tackle a specific problem is not straightforward and requires careful consideration. The algorithm's effectiveness depends on various factors such as the problem's complexity, the size of the search space, and the design of the objective functions. Whether the problem requires SO or MO optimization or if the objective functions are continuous or discrete, different algorithms may perform better. Therefore, it is advisable to try out multiple

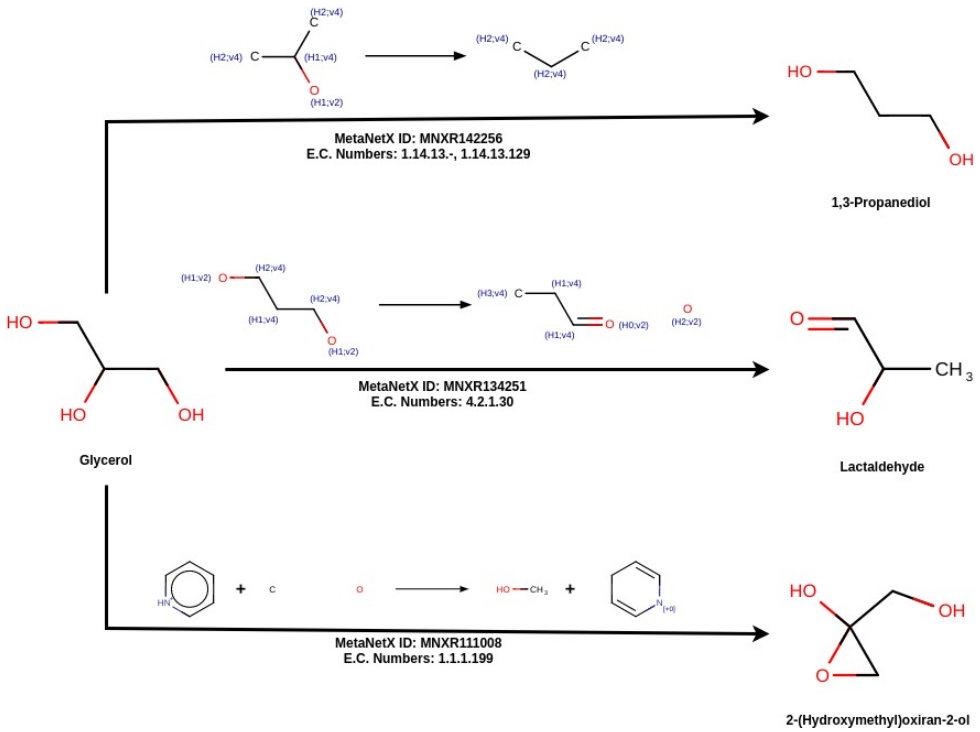

Figure 3: This is an example of how the mutation operator works: three different reaction rules are applied to create three distinct products (mutants) from a single reactant (parent). It is possible that different rules apply to the same compound, and one rule can also generate multiple different products.

algorithms and determine the most suitable one for a particular case.

ReactEA offers a comprehensive collection of EAs and some stochastic optimization algorithms for molecular property optimization: simulated annealing (SA) [1], genetic algorithm (GA) [35], evolution strategy (ES) [3], and local search (LS) [38] for SO optimization, and Non-dominated Sorting Genetic Algorithm III (NSGAIII)) [9], Non-dominated Sorting Genetic Algorithm II (NSGAII) [8], Indicator-based Evolutionary Algorithm (IBEA) [49], and Strength Pareto Evolutionary Algorithm 2 (SPEA2) [2] for MO optimization. Although designed for many objectives, MO EAs can also benefit SO optimization by improving convergence and exploration of the search space [31]. This approach is especially advantageous for complex problems that require a diverse set of solutions to cover different regions of the search space.

2.10 Development Environment

ReactEA was developed using Python version 3.8. Molecular operations like SMILES validity, standardization, reaction SMARTS validity, and reaction product generation were done using RDKit version 2022.03.1. EAs were implemented with jMetalPy version 1.5.5. Source code, small data files, and usage examples are available at https://github.com/BioSystemsUM/ReactEA. For reproducibility, all data and code used to generate the presented results are also available at https://zenodo.org/record/7630352.

3 RESULTS AND DISCUSSION

3.1 Optimization of Drug-Likeness, Solubility, and Synthetic Accessibility

Evaluating the performance of *de novo* design methods often involves testing them on simplified tasks, such as maximizing quantitative estimate of drug-likeness (QED) or logP. These objectives are straightforward to calculate and demonstrate the ability to generate molecules that meet specific goals. While they do not reflect the complexity of real-world drug discovery experiments, they can still serve as good indicators to identify compounds that are more likely to be successful drugs.

In this study, we compared the performance of ReactEA in optimizing simple tasks such as the QED, the penalized octanol–water partition coefficient (pLogP), the synthetic accessibility score (SAS), and the ChEMBL-Likeness Score (CLScore) with some state-of-the-art methods. The QED score, calculated using RDKit and ranging between 0 and 1, indicates how closely a compound resembles known drugs in terms of physical and chemical properties. A higher score suggests better drug-like properties, including oral bioavailability, low toxicity, and the ability to pass through the blood-brain barrier.

The logP value measures the solubility of a molecule in water versus lipids, with positive values indicating lipophilicity and negative values indicating hydrophilicity. In drug discovery, logP is crucial for determining the absorption, transportation, and distribution of drugs. Lipinski's Rule of 5 states that for optimal oral and

intestinal absorption, drugs must have a logP value below 5, with a preferable range of 1.35 to 1.8. In drug design studies, pLogP is often optimized instead, which is a penalized version of logP that accounts for large rings and synthetic accessibility (Equation 1). The pLogP score was calculated using RDKit and normalized across the ZINC dataset.

$$pLogP(m) = logP(m) - SA(m) - RingPenalty(m) \quad (1)$$

Synthetic complexity is an estimate of how difficult it would be to synthesize a molecule. We used the SAS metric proposed by Ertl and Schuffenhauer [13] to predict synthetic accessibility. SAS takes into account the molecule's similarity to reference drug-like compounds and the presence of synthetically complex features such as unusual rings and many stereo centers. SAS ranges from 1 to 10, with higher values indicating synthetic complexity. We also calculated a normalized SAS score, ranging from 0 to 1. RDKit was used to calculate the SAS scores.

The CLScore [5] is another metric that predicts synthetic accessibility. It compares a molecule to a subset of biologically active compounds in ChEMBL [30] using circular substructures called molecular shingles. Each shingle is assigned a weight based on its frequency of occurrence in the ChEMBL subset, and the sum of these weights is divided by the total number of shingles in the molecule to calculate a score. Higher scores indicate greater similarity to ChEMBL molecules. RDKit was used to calculate the CLScore scores.

The initial population consisted of a set of precursors from the *Escherichia coli* iJO1366 metabolic model [32], retrieved from the RetroPathRL GitHub repository [21]. Only molecules with valid InChI [16] were considered, resulting in a set of 648 precursors. The validity and conversion to SMILES notation were performed using RDKit.

We run GA, ES, NSGAIII, NSGAII, SPEA2, and IBEA with default parameters for 100 generations or until no improvements were observed for 5 generations, using the Ecoli Sink Set as the initial population. The results showed similar performance to state-of-the-art methods under comparable conditions. Direct comparisons are difficult due to differences in the conditions used by different methods (e.g. number of generations, restrictions on molecular size), thus results were compared to a limited set of studies reported in [27]. Fair comparisons were made by limiting the maximum number of heavy atoms to 38 in the pLogP optimization.

In the QED optimization, the ReactEA achieved a score of 0.948, comparable to other best-performing methods [27, 42, 45, 48], except for the method proposed by Zhang *et al.* [47] with a score of 0.954. The regular and normalized pLogP optimization resulted in maximum scores of 13.88 and 11.19, respectively, which were outperformed only by EvolMol [27]. The SAS optimization resulted in the best possible score of 1.0, outperforming scores of 0.95 reported in [27] and [7]. In the CLScore optimization, a top score of 6.78 was achieved, outperforming the score of 6.552 reported in [27]. The complete set of results is available in Supplementary Table 1.

It is important to note that optimizing these metrics does not reflect the complexity of real-world drug discovery experiments. For instance, the best molecule obtained in the pLogP optimization was a non-drug-like compound consisting of 38 carbon atoms. While the best-scoring molecules in the QED optimization display desirable drug-like properties, optimizing this metric alone may result in unrealistic molecules as it does not consider synthetic accessibility, pharmacokinetics, and target activity. The generated molecules from the SAS and CLScore optimizations are quite simple, mostly consisting of small alkanes and cycloalkanes. In the case of CLScore, it may be more meaningful to optimize values between 3 and 5, as they correspond to the peak in the ChEMBL distribution. These limitations suggest that objective functions must be carefully designed and that MO optimization should be integrated into drug discovery studies, along with additional targeted objectives.

3.2 Similarity to Aspartame

Designing novel molecules based on existing bioactive ones, or incorporating specific core structures and scaffolds is a common strategy in molecular design. This approach allows for the preservation of desirable properties while exploring new areas of chemical space, simplifies synthesis by using the core structure or scaffold as a template, and increases the likelihood of obtaining bioactive molecules due to their pre-determined structural features.

In order to assess the ability of ReactEA in finding paths to target molecules, we evaluated its performance in optimizing for the similarity to the Aspartame molecule. The similarity was defined as the Tanimoto Similarity [34] between Morgan fingerprints with radius 2 and 1024 bits. Additionally, to understand the impact of the initial population on the performance of ReactEA, we conducted experiments using the GA. We tested 8 different initial populations, each consisting of 100 molecules, including NP-based approved drugs from ChEMBL and scaffolds from known NPs.

The molecules from ChEMBL were selected by choosing NP-based approved small molecule drugs with molecular weights between 100 and 399, resulting in 184 valid molecules. We created 4 different initial populations from this set. For the first set, called "Representative ChEMBL", we selected the 100 most representative molecules by computing the Morgan fingerprints of each molecule, then reducing the dimensionality to two components using t-SNE. Next, using KMeans clustering with 100 clusters, we selected the molecule closest to the centroid of each cluster. For the second set, called "Similar ChEMBL", we selected 100 molecules from the same cluster by following a similar approach as for the Representative ChEMBL set, except that only 2 clusters were computed. The remaining two sets, "ChEMBL Top100" and "ChEMBL Worst100", consist of the 100 molecules with the best and worst similarity to Aspartame, respectively.

The set of 368 NP scaffolds was retrieved from the study by Lai et al. [25]. We also created 4 different initial populations of 100 molecules each using the same approach as for the ChEMBL data, and these sets are referred to as "Scaffolds Representative", "Scaffolds Similar", "Scaffolds Top100", and "Scaffolds Worst100".

For each initial population, the algorithm was run over 100 generations and repeated 10 times. The selection operator was Binary Tournament Selection, and the replacement of solutions was done by selecting the 100 fittest unique solutions among the previous solutions and offspring.

It is usually advantageous to start with a diverse initial population as this can increase the chances of exploring a wider search space. Despite this, ReactEA has demonstrated the ability to quickly

improve solutions even with a non-diverse initial population. In our experiments, we found no significant differences in the average score of generated molecules when using different initial populations (Table 2, Supplementary Figure 1). This can be associated with the fact that applying reaction rules to molecules can drastically change their score. The similarity to Aspartame scores ranged between 0.598 and 0.653. Interestingly, the initial population with the lowest average score was the Representative ChEMBL, which suggests that the diversity of the initial population does not significantly impact the final solutions. With the exception of the Representative ChEMBL, Representative Scaffolds, and Scaffolds Top100 sets, ReactEA was able to reach the Aspartame molecule or one of its stereoisomers (Supplementary Figure 2).

In addition to evaluating the similarity of the generated molecules to Aspartame, we also assessed their novelty, diversity, QED, and SAS. We found that all generated molecules were unique and distinct from the starting population. As expected, the internal similarity was high, with the generated molecules sharing many substructures. The average QED and SAS values of the generated molecules were comparable to those of Aspartame.

3.3 Docking to PARP1

The Poly [ADP-ribose] polymerase 1 (PARP1) enzyme plays a crucial role in DNA repair, cell signaling, and gene regulation. It was the first member of the PARP family of enzymes, which add ADP-ribose groups to target proteins, to be identified. PARP1 primarily mediates multiple DNA damage repair pathways, which repair DNA damage caused by various internal and external factors. If left unrepaired, this damage could lead to the development of several diseases, including cancer. Therefore, the inhibition of PARP1 is being explored as a treatment option for various cancers, such as ovarian, breast, and prostate cancer.

To show how ReactEA can be used for lead optimization, we used a library of 94 seed molecules, including 11 known PARP inhibitors (retrieved from http://www.clinicaltrials.gov) and 83 molecular fragments derived from them using Breaking of Retrosynthetically Interesting Chemical Substructures (BRICS) decomposition [10] by the authors of AutoGrow4.

Our objective function uses the DOCKSTRING package [15] for optimizing the binding affinity against the PARP1 enzyme. DOCKSTRING is a user-friendly Python wrapper of the AutoDock Vina package [39]. AutoDock Vina produces high-quality docking poses and reasonable binding free energy predictions while having a relatively low computational cost.

To ensure the generated molecules had desirable physical and chemical properties, we applied a set of molecular filters before docking. If a compound did not pass all the filters, it was assigned the worst fitness. These filters ensured that the molecules had a logP value between -0.4 and 5.6, a molecular weight between 160 and 500, a molar refractivity between 40 and 130, and between 20 and 70 heavy atoms.

The best-scoring molecule, along with the reaction rules used to generate it, is displayed in Figure 4. It has a predicted binding affinity of -14.5 kcal/mol, considerably better than the best-known inhibitor in the initial population (Olaparib with -12.3 kcal/mol). The molecule was generated with the GA. Other EAs also had comparable performance with the best binding affinities ranging from -12.5 kcal/mol to -14.5 kcal/mol (Table 3). The binding affinity of the best molecule of each EA at each generation and the best molecules generated by each EA can be consulted in Supplementary Figures 3 and 4.

The predicted binding affinities could easily be improved by simply raising the number of generations. However, doing so would come with a number of drawbacks. For one, AutoDock Vina and other docking tools tend to favor larger molecules [6]. Additionally, longer runs may lead to the emergence of unwanted functional groups that have the potential to be mutagenic or have unfavorable pharmacokinetic properties. Similarly, the synthesizability of compounds tends to decline in later generations as the accumulation of mutations causes the population to significantly deviate from the original molecules. Therefore, in general, a lower number of generations is preferred when optimizing molecular docking. However, these challenges can be addressed by using a MO optimization approach, such as the one offered by ReactEA, that can consider multiple objectives, such as binding affinity to the target, molecular weight, synthetic accessibility, and other relevant factors.

4 CONCLUSIONS

We have introduced ReactEA, a new open-source reaction-based EA framework for molecule generation. ReactEA can be used in SO and MO problems, optimizing any (set of) measurable objective functions with its large set of biochemical reaction rules. This allows for the generation of valid structures and assures chemical feasibility, with potential synthesis paths provided for the generated molecules.

Table 2: Impact of different initial populations on the performance of ReactEA. The percentage of unique and novel molecules, the similarity between the molecules, the mean and best similarity to the aspartame, and the QED and SAS are showcased.

Set	Unique/Novel	Internal Sim.	Mean Sim. to Aspartame	Best Sim. to Aspartame	QED	SAS
ChEMBL Representative	100%	0.457	0.598	0.803	0.436	3.030
ChEMBL Similar	100%	0.500	0.642	1.000	0.452	2.953
ChEMBL Top100	100%	0.510	0.653	1.000	0.447	2.971
ChEMBL Worst100	100%	0.506	0.639	1.000	0.447	2.971
Scaffolds Representative	100%	0.487	0.623	0.828	0.445	3.004
Scaffolds Similar	100%	0.508	0.645	1.000	0.428	2.973
Scaffolds Top100	100%	0.476	0.630	0.869	0.450	2.991
Scaffolds Worst100	100%	0.519	0.649	1.000	0.447	3.015

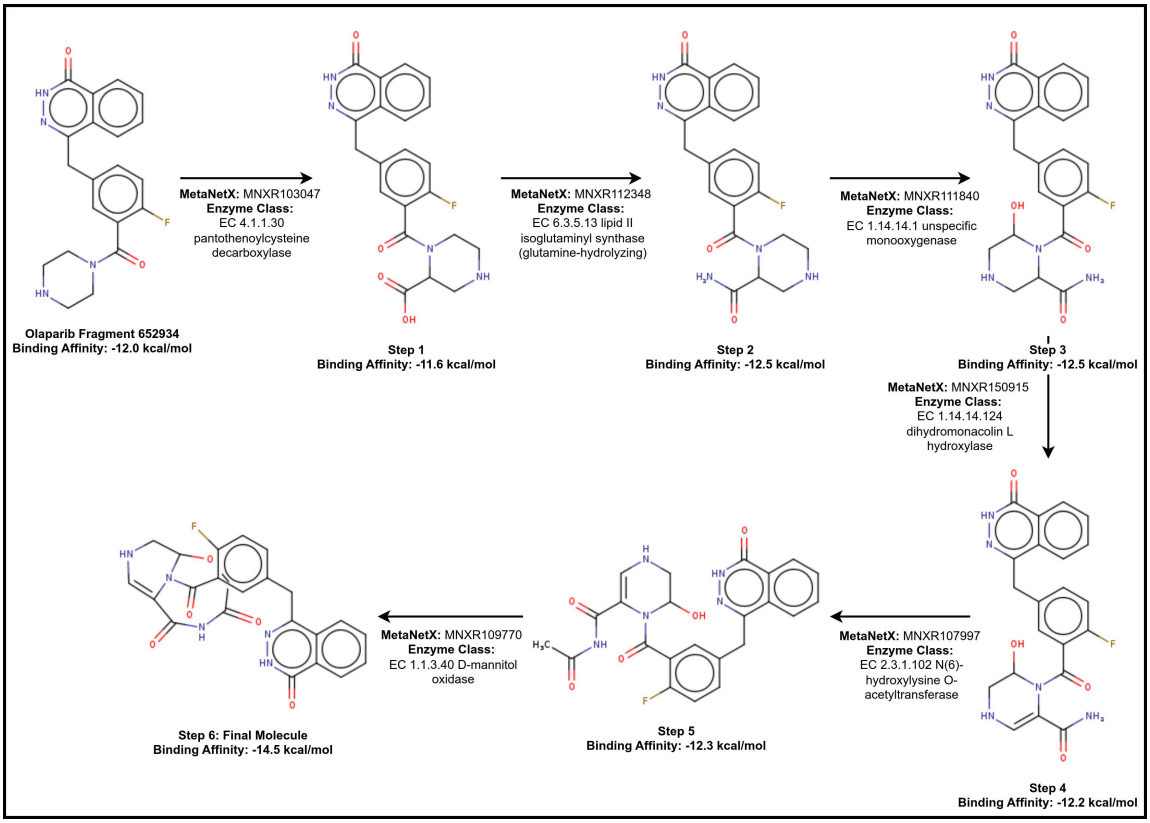

Figure 4: The molecule with the highest binding affinity and the reaction templates that were applied to transform the starting compound (Olaparib Fragment 652934) into the final molecule are presented, along with the intermediate molecules and the corresponding EC numbers for the original reactions.

Table 3: Results of the Docking to PARP1 optimization. The binding affinities of the initial population and the performance of the different EAs are shown.

EA	Worst	Best	Mean	Std. Dev.
Init. Pop.	50.0000	-12.1000	35.277660	25.8735
NSGAIII	-11.1000	-14.4000	-11.769149	0.5998
NSGAII	-10.6000	-12.5000	-11.198936	0.4759
SPEA2	-10.7000	-12.7000	-11.427660	0.5284
IBEA	-10.9000	-13.5000	-11.547872	0.5303
GA	-10.7000	-14.5000	-11.378723	0.6341
ES	-11.5000	-13.1000	-11.954255	0.3935

Compared to previous reaction-based EAs, such as Autogrow4 and iSyn, which use a limited set of click chemistry rules, ReactEA utilizes a more comprehensive set of reaction rules to enumerate possible biosynthetic routes connecting target molecules to precursors. This represents a step forward towards a more sustainable, economically viable, and environmentally responsible approach to chemical production. Our results demonstrate ReactEA's high configurability and versatility, achieving excellent results in optimizing simple objectives like QED or SAS, as well as more complex tasks like similarity to a target molecule and docking to proteins.

One of the limitations of ReactEA and other reaction-based approaches is their dependence on the available reaction rules and their coverage of the chemical space. While ReactEA uses a comprehensive set of biochemical reaction rules, their number still limits the chemical space that can be explored, potentially reducing the novelty and diversity of the generated molecules. Additionally, the reaction rules are limited to human knowledge, which may limit the exploration of new areas of the chemical space. Despite its relatively low computational demands, evaluating the results provided by ReactEA requires chemical expertise as the quality of generated molecules is currently either subjective or measured imperfectly.

ACKNOWLEDGMENTS

Centre of Biological Engineering (CEB, University of Minho) for financial and equipment support. Portuguese Foundation for Science and Technology (FCT) under the scope of the strategic funding of UIDB/04469/2020 unit and through a Ph.D. scholarship awarded to João Correia (SFRH/BD/144314/2019). European Commission through the project *SHIKIFACTORY100 - Modular cell factories for the production of 100 compounds from the shikimate pathway* (Reference 814408).

REFERENCES

[1] E.H.L. Aarts and J.H.M. Korst. 1989. *Simulated annealing and Boltzmann machines : a stochastic approach to combinatorial optimization and neural computing*. Wiley.

[2] Vincent J. Amuso and Jason Enslin. 2007. The Strength Pareto Evolutionary Algorithm 2 (SPEA2) applied to simultaneous multi- mission waveform design. In *2007 International Waveform Diversity and Design Conference*. IEEE. https://doi.org/10.1109/wddc.2007.4339452

[3] Thomas Bäck. 1996. Evolution strategies: An alternative evolutionary algorithm. In *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1–20. https://doi.org/10.1007/3-540-61108-8_27

[4] Antonio Benítez-Hidalgo, Antonio J. Nebro, José García-Nieto, Izaskun Oregi, and Javier Del Ser. 2019. jMetalPy: A Python framework for multi-objective optimization with metaheuristics. *Swarm and Evolutionary Computation* 51 (Dec. 2019), 100598. https://doi.org/10.1016/j.swevo.2019.100598

[5] Sven Bühlmann and Jean-Louis Reymond. 2020. ChEMBL-Likeness Score and Database GDBChEMBL. *Frontiers in Chemistry* 8 (Feb. 2020). https://doi.org/10.3389/fchem.2020.00046

[6] Max W. Chang, Christian Ayeni, Sebastian Breuer, and Bruce E. Torbett. 2010. Virtual Screening for HIV Protease Inhibitors: A Comparison of AutoDock 4 and Vina. *PLoS ONE* 5, 8 (Aug. 2010), e11955. https://doi.org/10.1371/journal.pone.0011955

[7] Nicola De Cao and Thomas Kipf. 2018. MolGAN: An implicit generative model for small molecular graphs. (2018). https://doi.org/10.48550/ARXIV.1805.11973

[8] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T Meyarivan. 2000. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In *Parallel Problem Solving from Nature PPSN VI*. Springer Berlin Heidelberg, 849–858. https://doi.org/10.1007/3-540-45356-3_83

[9] Kalyanmoy Deb and Himanshu Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (Aug. 2014), 577–601. https://doi.org/10.1109/tevc.2013.2281535

[10] Jörg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. 2008. On the Art of Compiling and Using 'Drug-Like' Chemical Fragment Spaces. *ChemMedChem* 3, 10 (Oct. 2008), 1503–1507. https://doi.org/10.1002/cmdc.200800178

[11] Thomas Duigou, Melchior du Lac, Pablo Carbonell, and Jean-Loup Faulon. 2018. RetroRules: a database of reaction rules for engineering biology. *Nucleic Acids Research* 47, D1 (Oct. 2018), D1229–D1235. https://doi.org/10.1093/nar/gky940

[12] Lars Elend, Luise Jacobsen, Tim Cofala, Jonas Prellberg, Thomas Teusch, Oliver Kramer, and Ilia A. Solov'yov. 2022. Design of SARS-CoV-2 Main Protease Inhibitors Using Artificial Intelligence and Molecular Dynamic Simulations. *Molecules* 27, 13 (June 2022), 4020. https://doi.org/10.3390/molecules27134020

[13] Peter Ertl and Ansgar Schuffenhauer. 2009. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics* 1, 1 (June 2009). https://doi.org/10.1186/1758-2946-1-8

[14] Nicholas C. Firth, Butrus Atrash, Nathan Brown, and Julian Blagg. 2015. MOARF, an Integrated Workflow for Multiobjective Optimization: Implementation, Synthesis, and Biological Evaluation. *Journal of Chemical Information and Modeling* 55, 6 (June 2015), 1169–1180. https://doi.org/10.1021/acs.jcim.5b00073

[15] Miguel García-Ortegón, Gregor N. C. Simm, Austin J. Tripp, José Miguel Hernández-Lobato, Andreas Bender, and Sergio Bacallado. 2022. DOCKSTRING: Easy Molecular Docking Yields Better Benchmarks for Ligand Design. *Journal of Chemical Information and Modeling* 62, 15 (2022), 3486–3502. https://doi.org/10.1021/acs.jcim.1c01334

[16] Stephen R Heller, Alan McNaught, Igor Pletnev, Stephen Stein, and Dmitrii Tchekhovskoi. 2015. InChI, the IUPAC International Chemical Identifier. *Journal of Cheminformatics* 7, 1 (May 2015). https://doi.org/10.1186/s13321-015-0068-4

[17] Jan H. Jensen. 2019. A graph-based genetic algorithm and generative model/-Monte Carlo tree search for the exploration of chemical space. *Chemical Science* 10, 12 (2019), 3567–3572. https://doi.org/10.1039/c8sc05372c

[18] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018. Junction Tree Variational Autoencoder for Molecular Graph Generation. https://doi.org/10.48550/ARXIV.1802.04364

[19] Kentaro Kawai, Naoya Nagata, and Yoshimasa Takahashi. 2014. De Novo Design of Drug-Like Molecules by a Fragment-Based Molecular Evolutionary Approach. *Journal of Chemical Information and Modeling* 54, 1 (Jan. 2014), 49–56. https://doi.org/10.1021/ci400418c

[20] Alan Kerstjens and Hans De Winter. 2022. LEADD: Lamarckian evolutionary algorithm for de novo drug design. *Journal of Cheminformatics* 14, 1 (Jan. 2022). https://doi.org/10.1186/s13321-022-00582-y

[21] Mathilde Koch, Thomas Duigou, and Jean-Loup Faulon. 2019. Reinforcement Learning for Bioretrosynthesis. *ACS Synthetic Biology* 9, 1 (Dec. 2019), 157–168. https://doi.org/10.1021/acssynbio.9b00447

[22] Yongbeom Kwon and Juyong Lee. 2021. MolFinder: an evolutionary algorithm for the global optimization of molecular properties and the extensive exploration of chemical space using SMILES. *Journal of Cheminformatics* 13, 1 (March 2021). https://doi.org/10.1186/s13321-021-00501-7

[23] Youngchun Kwon, Jiho Yoo, Youn-Suk Choi, Won-Joon Son, Dongseon Lee, and Seokho Kang. 2019. Efficient learning of non-autoregressive graph variational autoencoders for molecular graph generation. *Journal of Cheminformatics* 11, 1 (Nov. 2019). https://doi.org/10.1186/s13321-019-0396-x

[24] Hugo Lachance, Stefan Wetzel, Kamal Kumar, and Herbert Waldmann. 2012. Charting, Navigating, and Populating Natural Product Chemical Space for Drug Discovery. *Journal of Medicinal Chemistry* 55, 13 (May 2012), 5989–6001. https://doi.org/10.1021/jm300288g

[25] Junyong Lai, Jianxing Hu, Yanxing Wang, Xin Zhou, Yibo Li, Liangren Zhang, and Zhenming Liu. 2020. Privileged Scaffold Analysis of Natural Products with Deep Learning-based Indication Prediction Model. *Molecular Informatics* 39, 11 (May 2020), 2000057. https://doi.org/10.1002/minf.202000057

[26] Greg Landrum, Paolo Tosco, Brian Kelley, Ric, Sriniker, Gedeck, Riccardo Vianello, David Cosgrove, NadineSchneider, Eisuke Kawashima, Dan N, Andrew Dalke, Gareth Jones, Brian Cole, Matt Swain, Samo Turk, AlexanderSavelyev, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Daniel Probst, Kazuya Ujihara, Vincent F. Scalfani, Guillaume Godin, Axel Pahl, Francois Berenger, JLVarjo, Strets123, JP, and DoliathGavid. 2022. rdkit/rdkit: 2022_09_3 (Q3 2022) Release. https://doi.org/10.5281/ZENODO.7415128

[27] Jules Leguy, Thomas Cauchy, Marta Glavatskikh, Béatrice Duval, and Benoit Da Mota. 2020. EvoMol: a flexible and interpretable evolutionary algorithm for unbiased de novo molecular generation. *Journal of Cheminformatics* 12, 1 (Sept. 2020). https://doi.org/10.1186/s13321-020-00458-z

[28] Hongjian Li, Kwong-Sak Leung, Chun Ho Chan, Hei Lun Cheung, and Man-Hon Wong. 2014. iSyn: WebGL-Based Interactive De Novo Drug Design. In *2014 18th International Conference on Information Visualisation*. IEEE. https://doi.org/10.1109/iv.2014.10

[29] Xuhan Liu, Kai Ye, Herman W. T. van Vlijmen, Michael T. M. Emmerich, Adriaan P. IJzerman, and Gerard J. P. van Westen. 2021. DrugEx v2: de novo design of drug molecules by Pareto-based multi-objective reinforcement learning in polypharmacology. *Journal of Cheminformatics* 13, 1 (Nov. 2021). https://doi.org/10.1186/s13321-021-00561-9

[30] David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, María Gordillo-Marañón, Fiona Hunter, Laura Junco, Grace Mugumbate, Milagros Rodriguez-Lopez, Francis Atkinson, Nicolas Bosc, Chris J Radoux, Aldo Segura-Cabrera, Anne Hersey, and Andrew R Leach. 2018. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research* 47, D1 (Nov. 2018), D930–D940. https://doi.org/10.1093/nar/gky1075

[31] P. Murugan, S. Kannan, and S. Baskar. 2009. Application of NSGA-II Algorithm to Single-Objective Transmission Constrained Generation Expansion Planning. *IEEE Transactions on Power Systems* 24, 4 (Nov. 2009), 1790–1797. https://doi.org/10.1109/tpwrs.2009.2030428

[32] Jeffrey D Orth, Tom M Conrad, Jessica Na, Joshua A Lerman, Hojung Nam, Adam M Feist, and Bernhard Ø Palsson. 2011. A comprehensive genome-scale reconstruction of Escherichia coli metabolism–2011. *Molecular Systems Biology* 7, 1 (Jan. 2011), 535. https://doi.org/10.1038/msb.2011.65

[33] Pavel Polishchuk. 2020. CReM: chemically reasonable mutations framework for structure generation. *Journal of Cheminformatics* 12, 1 (April 2020). https://doi.org/10.1186/s13321-020-00431-w

[34] David Rogers and Mathew Hahn. 2010. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* 50, 5 (April 2010), 742–754. https://doi.org/10.1021/ci100050t

[35] Kumara Sastry, David Goldberg, and Graham Kendall. [n. d.]. Genetic Algorithms. In *Search Methodologies*. Springer US, 97–125. https://doi.org/10.1007/0-387-28356-0_4

[36] Jacob O. Spiegel and Jacob D. Durrant. 2020. AutoGrow4: an open-source genetic algorithm for de novo drug design and lead optimization. *Journal of Cheminformatics* 12, 1 (April 2020). https://doi.org/10.1186/s13321-020-00429-4

[37] Jonathan Strutz, Kevin M Shebek, Linda J Broadbelt, and Keith E J Tyo. 2022. MINE 2.0: enhanced biochemical coverage for peak identification in untargeted metabolomics. *Bioinformatics* 38, 13 (May 2022), 3484–3487. https://doi.org/10.1093/bioinformatics/btac331

[38] Dirk Sudholt. 2006. Local Search in Evolutionary Algorithms: The Impact of the Local Search Frequency. In *Algorithms and Computation*. Springer Berlin Heidelberg, 359–368. https://doi.org/10.1007/11940128_37

[39] Oleg Trott and Arthur J. Olson. 2009. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry* (2009), NA–NA. https://doi.org/10.1002/jcc.21334

[40] Eelke van der Horst, Patricia Marqués-Gallego, Thea Mulder-Krieger, Jacobus van Veldhoven, Johannes Kruisselbrink, Alexander Aleman, Michael T. M. Emmerich, Johannes Brussee, Andreas Bender, and Adriaan P. IJzerman. 2012. Multi-Objective Evolutionary Design of Adenosine Receptor Ligands. *Journal of Chemical Information and Modeling* 52, 7 (June 2012), 1713–1721. https://doi.org/10.1021/ci2005115

[41] David Weininger. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Modeling* 28, 1 (Feb. 1988), 31–36. https://doi.org/10.1021/ci00057a005

[42] Robin Winter, Floriane Montanari, Andreas Steffen, Hans Briem, Frank Noé, and Djork-Arné Clevert. 2019. Efficient multi-objective molecular optimization in a continuous latent space. *Chemical Science* 10, 34 (2019), 8016–8024. https://doi.org/10.1039/c9sc01928f

[43] Olivier J. Wouters, Martin McKee, and Jeroen Luyten. 2020. Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018. *JAMA* 323, 9 (March 2020), 844. https://doi.org/10.1001/jama.2020.1166

[44] Naruki Yoshikawa, Kei Terayama, Masato Sumita, Teruki Homma, Kenta Oono, and Koji Tsuda. 2018. Population-based De Novo Molecule Generation, Using Grammatical Evolution. *Chemistry Letters* 47, 11 (Nov. 2018), 1431–1434. https://doi.org/10.1246/cl.180665

[45] Jiaxuan You, Bowen Liu, Rex Ying, Vijay Pande, and Jure Leskovec. 2018. Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation. https://doi.org/10.48550/ARXIV.1806.02473

[46] Wenbo Yu and Alexander D. MacKerell. 2016. Computer-Aided Drug Design Methods. In *Methods in Molecular Biology*. Springer New York, 85–106. https://doi.org/10.1007/978-1-4939-6634-9_5

[47] Chenrui Zhang, Xiaoqing Lyu, Yifeng Huang, Zhi Tang, and Zhenming Liu. 2019. Molecular Graph Generation with Deep Reinforced Multitask Network and Adversarial Imitation Learning. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE. https://doi.org/10.1109/bibm47256.2019.8983277

[48] Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N. Zare, and Patrick Riley. 2019. Optimization of Molecules via Deep Reinforcement Learning. *Scientific Reports* 9, 1 (July 2019). https://doi.org/10.1038/s41598-019-47148-x

[49] Eckart Zitzler and Simon Künzli. 2004. Indicator-Based Selection in Multiobjective Search. In *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 832–842. https://doi.org/10.1007/978-3-540-30217-9_84