Assessing Methods and Obstacles in Chemical Space Exploration

Authors:

Shawn Reeves[1], Benjamin DiFrancesco[1], Vijay Shahani[1], Stephen MacKinnon[1], Andreas Windemuth[1], Andrew E. Brereton[1]

Affiliations:

1. Cyclica Inc. 207 Queens Quay W Suite 420, Toronto, ON M5J 1A7, Canada

Abstract

Benchmarking the performance of generative methods for drug design is complex and multifaceted. In this report, we propose a separation of concerns for *de novo* drug design, categorizing the task into three main categories: **generation**, **discrimination**, and **exploration**. We demonstrate that changes to any of these three concerns impacts benchmark performance for drug design tasks. In this report we present Deriver, an open-source Python package that acts as a modular framework for molecule generation, with a focus on integrating multiple generative methods. Using Deriver, we demonstrate that changing parameters related to each of these three concerns impacts chemical space traversal significantly, and that the freedom to independently adjust each is critical to real-world applications having conflicting priorities. We find that combining multiple generative methods can improve optimization of molecular properties and lower the chance of becoming trapped in local minima. Additionally, filtering molecules for drug-likeness (based on physicochemical properties and SMARTS pattern matching) before they are scored may hinder exploration, but can also improve the quality of the final molecules. Finally, we demonstrate that any given task has an exploration algorithm best suited to it, though in practice linear probabilistic sampling generally results in the best outcomes, when compared to Monte Carlo sampling or greedy sampling. Deriver is being made freely available, to help others interested in collaboratively improving existing methods in de novo drug design centered around inheritance of molecular structure, modularity, extensibility, and separation of concerns.

Introduction

Algorithmic generation of Novel Chemical Entities (NCEs) is a challenging computational problem with practical applications in drug design, materials design, agriculture, and other chemical-based industries. Several recently described approaches to this problem integrate deep-learning to rapidly generate compounds with more desirable attributes (De Cao & Kipf, 2018; Grisoni et al., 2020; Jin et al., 2019; Kadurin et al., 2017; Kusner et al., 2017; Neil et al., 2018). In particular, generative deep learning approaches tend to excel when coupled with very specific tasks, such as producing pools of molecules matching property distributions of a training set (Polykovskiy et al., 2019) or achieving highly competitive performance on defined benchmark tasks (Brown et al., 2019)). The highly specialized nature of generative deep learning methods can, however, pose significant drawbacks, such as an inability to generalize outside of the domain of the training data (Zhavoronkov, 2019; Lowe, 2019), or the need for expensive metaheuristics to find promising regions in a latent space (Winter et al., 2019; Gómez-Bombarelli, 2018; Jin, 2019). It may also require training new generators for each discriminative task (Altae-Tran et al., 2017), and suffers from the black box nature of neural network predictions. To maximize utility and the ease in which a human expert can usefully participate in an AI guided design-make-test cycle (Green et al., 2019), it is critical that design pipelines be flexible, tunable, modular, and interpretable.

In computer science, separation of concerns (SoC) (Dijkstra, 1982) is a software design principle, which describes the paradigm of modular software to facilitate reuse, upgrading, and testing of individual components. For any drug design algorithm, there are three main concerns: (1) the method(s) for generating chemically valid candidate molecule structures (**generation**); (2) the assessment of candidate molecules, including determining to what degree (if any) a candidate satisfies the constraints of the objective(s) (**discrimination**); and (3) the search strategy, which integrates generation and discrimination to guide chemical space traversal (**exploration**). While several studies (Polishchuk, 2020; Hartenfeller & Schneider, 2011; Green et al., 2019; Segler et al., 2017) have made similar distinctions between components of a *de novo* drug design algorithm, no system adequately leverages the separation of these concerns in their implementation, specifically with respect to decoupling the chemical space search (exploration).

Exploration is the bridge between generation and discrimination. It allows an NCE generating system to subsample the otherwise cost-prohibitive chemical space using optimization techniques such as: evolutionary approaches (Ertl & Lewis, 2012; Jensen, 2019), reinforcement-learning (Guimaraes et al., 2018; Popova et al., 2018; Olivecrona et al., 2017; Neil et al., 2018), Monte-Carlo Tree Search (Jensen, 2019), and other metaheuristics such as Bayesian- (Gómez-Bombarelli, 2018; Jin et al., 2019) or particle-swarm-based (Winter et al., 2019) optimizations of the latent space. Effectively traversing chemical space with traditional

methods depends on the “step-size” of the methods being used to generate candidate molecules. The CReM framework for structure generation (Polishchuk, 2020) describes these step-sizes with a distinction between atom-, reaction-, and fragment-based generators. An atom-based approach uses simple rules like addition, substitution, or deletion of bonds and atoms. Atom-based approaches can traverse the broadest range of chemical space, at the cost of potentially creating chemically non-viable or synthetically un-feasible molecules. In contrast, a reaction-based approach builds new molecules by simulating standard chemical reactions starting with commercially available organic starting material. The reaction-based strategy generates valid and synthetically accessible molecules by design, at the cost of restricting chemical space exploration, sometimes significantly (Lessel & Lemmen, 2019). Fragment-based approaches represent a middle ground between reaction-based and atom-based methods, where molecular fragments representing one or more atoms are collectively added, substituted or removed in single steps in accordance with a chemistry-based ruleset (eg. Degen et al., 2008). The choice of fragments to recombine directly influences the tradeoff between synthetic feasibility and breadth of chemical space traversal, with lower-complexity fragments progressively sharing more of the functional properties of atom-based generators.

With molecules produced algorithmically, there is an acknowledged trade-off between their validity and synthetic accessibility, and the breadth of “acceptable” chemical space that can be explored (Polishchuk, 2020). This balance can be further influenced by intrinsic factors (in the rules or structure of the methods themselves), and extrinsic factors (ie. a discrete discriminative mechanism like a filter). In the case of some deep-learning methods, such as: autoencoders (Kusner et al., 2017; Gomez-Bombarelli et al., 2018), generative adversarial networks (Guimaraes et al., 2018; De Cao and Kipf, 2018), and recursive neural networks (Arús-Pous et al., 2019; Olivecrona et al., 2017; Segler et al., 2018), intrinsic restrictions on chemical space are typically incorporated through learning from the training data (Arús-Pous et al., 2019). After training, this imposes an immutable constraint on chemical space exploration, though its impact can only be inferred indirectly (Lessel & Lemmen, 2019). In contrast to these methods, heuristic approaches enable more methods to control the restrictions on chemical space exploration (eg. Verhellen & Van den Abeele, 2020). For example, extrinsic filtering can be applied (or not) at any point in the search process, modified as needed, and facilitates the reuse of a single workflow after adjusting the definition of an acceptable molecule. Additionally, the modularity of generative methods permits more or less restrictive rule sets to be used to generate candidates. This equips a human expert to better assess the trade-off between molecule acceptability and chemical space exploration for any given task.

The regions of chemical space explored by an iterative algorithm are directed both by the generative method(s) used, and by how the search is updated with the information obtained from the discrimination of previously generated molecules. In the case of an evolutionary algorithm, the search strategy involves a scheme for selecting certain candidates to derive children from, and a greedy selection mechanism may traverse chemical space more or less effectively than a probabilistic one. For many deep learning approaches, guided exploration of chemical space may not be a design objective, and no search strategy is implemented. These

approaches instead seek to accurately represent the distribution of the training data, ideally incorporating the notions of synthetic accessibility, validity, and quality (Segler et al., 2017; Ertl et al., 2018).

While it can be difficult to measure the effectiveness of chemical space exploration, community benchmarks provide a useful way to assess the performance of different approaches to the same task. For drug design, the recently presented Guacamol framework (Brown et al., 2019) has become a useful standard to compare to, although discussion is ongoing on how to improve these benchmarks (Renz et al., 2020). The Guacamol benchmarks are broadly divided into two categories: distribution-learning and goal-directed. Distribution-learning benchmarks focus on benchmarking a model's ability to generate valid (corresponding to a molecular graph), unique (non-duplicated) and novel (not seen in training data) candidates with the physical property distributions of a training dataset. The goal-directed benchmarks are designed to assess ability to find regions of chemical space, which optimize a specific objective function. Guacamol's twenty goal-directed benchmarks assess disparate goal-directed behaviours, where a model is expected to start from a library of known bioactive molecules and explore chemical space guided by feedback from an external scoring function. These benchmarks range from scoring molecules based on fingerprint similarity to a target molecule, to combined objectives where physical properties, substructure patterns, and chemical similarity are simultaneously assessed. For a more detailed description of each benchmark, please refer to the original publication by Brown et al. (2018).

In this study, we present Deriver: a single framework which facilitates the integration of a number of tunable generative methods, permits discrimination of molecules based on arbitrary or shifting objectives, and combines these two processes in a separable manner. To demonstrate the usage of Deriver we employ Guacamol's goal-directed benchmarks (Brown et al. 2019) to explore some of the described trade-offs associated with effective chemical space traversal under various constraints. Deriver produces 100% unique, valid, and novel molecules (based on the provided seeds) by design; as such, the focus of this paper are the goal-directed benchmarks.

Strategy

Generators

Deriver facilitates the combination of multiple methods for generating molecules, including the canonicalization and sanitization of generated molecules. Since version 2.3.10, there are five primary generators implemented in Deriver: a fragment-based method using the RDKit

(Landrum, 2006) implementation of BRICS (Degen et al., 2008); a naive SELFIES (Krenn et al., 2020) mutator; an exhaustive single-atom replacement method based on SELFIES (called Scanner); and two graph-based methods from Jensen et al. (Jensen, 2019).

BRICS stands for “Breaking of Retrosynthetically Interesting Chemical Substructures” and is a fragment-based approach, producing fragments by decomposing molecules according to predetermined rules (Degen et al., 2008). In Deriver, seed molecules are exhaustively broken into BRICS fragments up to a certain complexity (no more than 7 intact BRICS bonds) and recombined with fragments from a library database (pre-generated by the same method). For this report, a fragment library was generated by fragmenting all SMILES from the `guacamol_v1_all.smiles` file, available through the guacamol_baselines package (Brown et al., 2019). In Deriver, the `fragment.libgen` method is used to generate fragment libraries, and the `derive_brics` method is used to generate new molecules using BRICS.

SELFIES (SELF-referencIng Embedded Strings), is a molecular string representation with the useful property that, unlike the ubiquitous SMILES (Simplified Molecular Line Entry System) string representation (Weininger, 1988), every SELFIES corresponds to a valid molecule and every molecule has a unique SELFIES (Krenn et al., 2020). Deriver makes use of SELFIES in more than one way: in the original implementation of the `derive_selfies` function, it applies a number of string additions, substitutions, and deletions to derive child SELFIES from parent SELFIES. This is an atom-based approach and will be referred to as ‘naive’ SELFIES, since it does not incorporate crossover between candidate molecules.

In addition to the multi-mutation naive SELFIES, Deriver implements a method called “Scanner” (`scan_selfies`), based on the concept of ‘positional analogue scanning’ (Verhellen & Van den Abeele, 2020). Scanner exhaustively applies every possible single atom substitution from a predefined set, to every seed molecule, enumerating all single steps in chemical space from the entire seed population. An important consideration when using Scanner is that unlike most other methods it is impossible to request a specific number of child molecules; it will always fully enumerate the local chemical space.

Deriver also implements two methods previously described in Jensen (2019) with notable performance on the Guacamol benchmarks (Brown et al., 2019). The Jensen methods apply handmade rules with concern for validity to molecular graphs, expressed as either SMILES (as in Brown et al., 2004) or SELFIES. In Deriver, these methods perform mutation and crossover operations as in the work of (Virshup et. al, 2013) and (Brown et al., 2004), respectively. The assumption made in Jensen’s SMILES-based implementation, regarding the allowed size of candidate molecules, was removed.

Discriminators

In addition to the external scoring functions implemented by the Guacamol benchmarks, Deriver implements a functionality for optionally filtering molecules. There are three categories of filtering by which a molecule can be rejected: physicochemical property ranges, the presence of specified SMARTS pattern, or absence thereof. All of these filter components are optional and tunable. Following filtering, two objects are returned by the Deriver API: the list of candidates which passed all filters, and a dictionary containing information about every derived molecule, whether it passed all filters, on which criteria it was rejected, and its physicochemical properties. The filters in this study applied both, the drug-likeness based physicochemical property restrictions listed in Table 1 (the default in Deriver), and the unwanted SMARTS filters described in the Guacamol benchmark study (Brown et al., 2019). Each of the four SMARTS pattern sets (PAINS, Glaxo, SureCHEMBL, BAI) are available in Deriver, and are implemented via the `rd_filters` python package. Enabling filtering in Deriver activates PAINS and Glaxo filtering rules by default.

Exploration

Evolutionary algorithms for chemical space traversal typically involve an iterative process where, starting from some seed population, (1) new candidate molecules are generated, (2) some fraction of previous candidates are selected as seeds on the basis of their objective score, and (3) the process is repeated (Hartenfeller & Schneider, 2011). The method for selecting seeds from the population, the size of population, the number of seeds, and the number of new candidates to generate are all potentially confounding when comparing generative methods. It is important to properly control these, especially when comparing against other baseline models via benchmarks. While the use of Deriver doesn't prohibit the selection of any particular exploration algorithms, in this report we limit our choice of exploration algorithm to three basic types: (1) greedy sampling, (2) linear probabilistic sampling (as used in the Guacamol implementation of Jensen's graph-based genetic algorithm (Brown et al., 2019)), and (3) an adapted form of the Metropolis-Hastings algorithm (Hastings, 1970).

The greedy sampling method simply selects the top *n* highest scoring molecules from the combined population of new candidate molecules and previous top molecules. The linear probabilistic sampling method normalizes the scores within the population (dividing each score by the population sum) before sampling the population (with replacement) using the normalized scores as probabilities.

The Metropolis sampling method makes use of two additional parameters: the highest score *h* from the previous generation, and a temperature *T* which decays each generation. For each molecule in the population, if its score exceeds or matches *h*, it is chosen as a seed for the next generation. For the remaining population of molecules, scores are converted to weights using

the following formula:

LATEX: e^{-(h-score)/T}

$$e^{-(h-score)/T}$$

The weights are normalized by division by the sum of the weights, and these remaining molecules are sampled (without replacement) according to the normalized weights (as probabilities). The intent of Metropolis sampling is to gradually decrease the temperature and shift the selection paradigm from exploration toward purely greedy sampling, over multiple generations. Notably, this method is always semi-greedy, since any new top scoring candidates in each generation are always selected as seeds.

Results

The effect of generator step-size on chemical space exploration

To assess the changes to chemical space exploration, and to benchmark performance, Deriver was applied for each of the twenty Guacamol goal-directed benchmarks while changing only the generators. When applied to a given benchmark, Deriver generates approximately 10,000 candidate molecules to score and assess for each generation. Greedy sampling is used, such that the top 100 molecules seen so far are used to reseed the generator in the next iteration.

Multiple methods, each having different degrees of granularity, are used as generators: BRICS (coarse granularity), naive SELFIES (high granularity), Scanner (highest granularity), and specified combinations of these methods. When using BRICS and naive SELFIES in tandem, the 10,000 requested molecules were divided into 7000 BRICS-based molecules and 3000 naive SELFIES-based molecules. When the Scanner method is also enabled, it may supply on the order of 10,000 additional candidate molecules, but this is highly variable. The 70:30 ratio between BRICS- and naive SELFIES-based molecules was chosen following casual observation of effectiveness made during prior experiments and represents a tunable parameter that is likely to affect the outcome of a goal-directed benchmark. The optimal number of candidate molecules selected per generation is dependent on the computational costs associated with the descrimination step. Descrimination based on inexpensive ligand-based strategies such as QSAR models may benefit from higher compound counts per-generation, while discrimination based on molecular dynamics or docking simulations would warrant more selective thresholds.

Figure 1 demonstrates that combining more than one generative method may result in improved solutions to chemical space navigation problems, when compared to using a single method alone. While the naive SELFIES method is seen to generally perform well alone, its performance may be altered by mixing its derived candidates with those from the BRICS and Scanner methods. For instance, in the Median Molecules 2 benchmark, the combination of all three methods results in the highest score of 0.4397. Another example is seen in the Osimertinib MPO benchmark, where the BRICS + naive SELFIES combination outperforms all other methods at a score of 0.9779, and the further addition of Scanner reduces performance to 0.9404. The improvement in performance observed when combining methods, which is most clearly seen in the multi-parameter-optimization objectives, is likely a result of combining coarse and fine-grained methods for chemical space navigation; in this instance, BRICS is a coarse-grained fragmentation method, while the SELFIES and Scanner methods are increasingly fine-grained. Furthermore, the two methods incorporating Scanner frequently converged in many fewer generations than those which did not, while only using BRICS frequently resulted in no convergence within 200 generations (Figure S1).

(Figure 1: All Benchmarks)

The effect of generator granularity can be clearly seen when the starting population consists of only a single candidate, as in the Ranolazine MPO Benchmark shown in Figure 2. The scoring function for this benchmark integrates the following objectives: similarity to Ranolazine, maximization of logP and of TPSA, and the presence of exactly 1 fluorine atom (Brown et al., 2019). For this benchmark, any Deriver which used BRICS had all top-100 molecules with scores exceeding 0.5 (scores range from 0 to 1 with closer to 1 being better) after just one generation. In contrast, the worst-of-top-100 and mean-of-top-100 for naive SELFIES are slightly below and above 0.2, respectively, after one generation. For the Scanner-only Deriver, the worst-of-top-100 score is approximately 0 at the same point. In addition, the Scanner-only Deriver reached convergence quickly, producing a less-than-optimal candidate set with a final score of 0.8977. Interestingly, while the BRICS and naïve SELFIES combination was similarly high-performing (score of 0.9623), further combination with scanner converged to a higher score (0.9935), and in 135 fewer generations. The BRICS only Deriver reached the maximum permitted number of generations (200), suggesting that it may be possible to achieve higher scores given more time.

(Figure 2: Ranolazine MPO Benchmark)

Another particularly informative benchmark is the Perindopril MPO benchmark, seen in Figure 3. The per-generation performance illuminates some of the emergent behaviours arising from combining generative methods, as well as the vulnerability of some approaches to the stopping criteria. Convergence is said to be reached if the mean score of the top-*n* molecules does not increase for 5 consecutive generations, where *n* is the expected number of candidates to return

to Guacamol (in this benchmark, *n*=100). For each single-method Deriver, the mean score plateaus at ~0.7, whereas for all three methods combined it plateaus near 0.725. Removing Scanner increases the score to 0.75 for [BRICS + naive SELFIES]. The second plateau near 0.82, for [BRICS + naive SELFIES], highlights that omitting the Scanner method may avoid deleterious local-optimum trapping. In other words, it is not always advantageous to blindly combine every available method.

(Figure 3: Perindopril MPO Benchmark)

The impact of filtering approaches on chemical space exploration

The two most common methods for filtering molecules in de novo design are to apply a filter persistently through each iteration of an experiment, such that every scored molecule has necessarily passed all filters (Yuan et al., 2011; Green et al., 2019), or, to counter-screen at the end of the design process by filtering the final scored molecules (especially to augment linked generator-discriminators, as in Zhavoronkov et al., 2019). Interestingly, the modularity built into Deriver permits a third option: a delayed filtering mechanism, in which the algorithm is allowed to explore chemical space without filtering, until it is turned on by reaching some important threshold (typically a first convergence). It is important to consider how filtering is applied, as it not only impacts the quality of the produced molecules, but also the trajectory through chemical space.

To assess how different approaches for filtering impact chemical space exploration, Deriver is again assessed on the twenty goal-directed benchmarks from Guacamol, while varying the strategy for applying the algorithmic filters (as described in the Strategy section, under Discriminators). In this experiment, the Deriver implementation of Jensen's graph based genetic algorithm (Jensen, 2019) is used with the same parameters as originally used in the Guacamol benchmarks (Brown et al., 2019). Four versions are compared, which only differ in how filtering is applied: (1) unfiltered; (2) filtered persistently; (3) delayed filtering; and (4) a counter-screen that applies the filters only at the end of each benchmark. For each of these cases it is important to consider both the benchmark performance as well as the "quality" of the final molecules as drug candidates, of which Brown et al. (2019) evaluated by detecting undesirable substructures.

In most cases, delayed filtering worsens performance to a lesser extent than does persistent filtering (Figure 4). While the unfiltered version had a higher total score (the summed score across all benchmarks) than the delayed filter (17.85 unfiltered compared to 17.60 delayed), the Troglitazone rediscovery benchmark result is worth highlighting. Filtered Deriver methods are unable to succeed because Troglitazone itself does not pass one of the SMARTS filters (SureCHEMBL). The counter-screened molecules almost always performed worse than all other approaches (Figure 4), suggesting a need to integrate drug-likeness objectives into the selective pressures being applied throughout the process. Supplementary Table 1 shows the fraction of the top-100 and fraction of requested molecules which passed all filters, alongside the scores,

for each benchmark. The delayed filtering method was able to produce the required number of acceptable candidates for each benchmark except the Sitagliptin MPO. The counter-screen was performed on the unfiltered Deriver results and occasionally resulted in the entire top population being eliminated.

(Figure 4: Comparison of Filtering Methods on all Benchmarks)

Filtering molecules before they are scored hinders chemical space traversal. Not only are fewer candidate molecules available from one generation to the next, but any filtered molecule will not be scored and will never seed new exploration. This helps to explain the behavior of the persistent filter Deriver over time (Figure 5); initial progress toward the solution is slower, and the highest scoring molecules are not optimized as well as in the delayed filter or the unfiltered Deriver. In contrast, the behavior of both the unfiltered and the delayed filter Deriver are very similar to each other until convergence, with variation attributable to chance. After convergence, the filters are enabled in the delayed filter Deriver, and the mean score of the whole population decreases sharply as new sub-optimal but filter-passing candidates extend the original population, followed by a second round of selection-based improvement convergence to an even higher overall score (Figure 5). It should be noted that because convergence occurs twice, the convergence criterion may be considered less strict, and so amenable to greater scores at the cost of additional computation.

(Figure 5: Comparison of Filtering Methods on Perindopril MPO)

To complement the algorithmic quality checks, a separate assessment was conducted based on the blind opinion of two medicinal chemists. The molecules generated by each filtering method on the Zaleplon MPO benchmark were combined into an unlabelled set of 264 unique molecules (Supplementary Figure 3), and the chemists were asked to label each “acceptable” molecule, individual chemists were allowed to impose their own definition of acceptability. Table 2 indicates the consensus of acceptability between both chemists and algorithmic filters, and serves as a proxy for the number of potential candidates of interest each method might produce. Here the delayed filter and persistent filter Deriver were comparable, with 40% compared to 37% consensus acceptance respectively. Medicinal chemist 1 was far more strict than medicinal chemist 2, who rejected about half as many molecules and a similar number to the algorithmic filters. Interestingly, upon inspection of the disagreement between medicinal chemists, ~59% (63/107) of the rejections could be attributed to the presence of a Michael acceptor, which has potential for covalent modification. Accepting or rejecting this group may depend heavily on the project criteria and the hit discovery philosophy of the medicinal chemist. Table 3 indicates that not only is there a large degree of discord between chemists and the substructure filters, but also between individual chemists as seen in (Kutchukian et al., 2012).

It is of particular interest to see cases in which the medicinal chemists accepted a molecule which was rejected algorithmically (of which there are 14), and the reverse situation, where the filters appear to have missed some undesirable characteristic apparent to chemists (occurred

24 times). Supplementary Figures 4 and 5 illustrate these cases alongside the reasons provided for rejection.

The impact of exploration algorithms on benchmark performance

The impact of exploration algorithms, such as sampling methods and evolutionary algorithms, must be understood on a case-by-case basis to properly assess generative methods. Deriver was assessed using the previously seen combination of BRICS, naive SELFIES, and Scanner as a generator, while three different exploration algorithms were tested: greedy, linear probabilistic, and Metropolis. Performance on goal-directed benchmarks can be seen in Figure 6.

(Figure 6: Comparison of Selection Method on All Benchmarks)

The greedy, Metropolis, and linear probabilistic sampling methods achieved total scores across all 20 benchmarks of 17.264, 17.440, and 17.758 respectively, and no single exploration algorithm demonstrates consistent improved performance over another. Instead, the key differences between these approaches are in specific benchmark performance, as well as the number of generations required to converge (Supplementary Figure 1). On average (across benchmark tasks), the greedy approach took 47 generations to converge, compared to 83 for Metropolis and 134 for linear, and for 6 out of 7 multiparameter optimization benchmarks the linear approach failed to converge after 200 generations.

Despite converging in far fewer generations than the linear method (and thus sampling fewer molecules overall), the greedy selection scheme gave state-of-the-art results on the Ranolazine (0.9935) and Sitagliptin (0.9258) multi-parameter optimization benchmarks. The benchmark tasks with the largest difference in performance between the three sampling methods are the three rediscovery benchmarks; the linear probabilistic sampling in particular excelled at these tasks (Figure 5, leftmost black box).

Hyperparameter optimization as a necessary step for design challenges

Deriver is a framework for generative methods, molecule filtering, capture of statistics, and ultimately experimentation. Notably, different combinations of generators, discriminators, exploration algorithms, and associated parameters (eg. population size) lead to significantly different results on the same benchmarks (Figure 1, Figure 4). Based on the results observed in the other experiments in this study, we chose a specific set of parameters for Deriver that we expected to be high-performing (but not the highest-possible): 1000 BRICS Deriver candidates, 1000 Jensen SMILES Deriver candidates, and 1000 Jensen SELFIES Deriver candidates with a mutation rate of 0, per generation; the previously described linear probability selection scheme; 200 molecules to seed each generation, selected from a population of 1000 best-seen

molecules.

This combination of methods (Figure 7, Deriver Optimized) demonstrates equal or improved performance on all benchmarks except the Osimertinib MPO, compared to the previous best described by Brown et al. (2019) (Figure 7, graph_GA reported). Any number of tunable hyperparameters (e.g. the combination of generators, the number of selected top molecules per generation, or the selection algorithm used) may have been critical in improving benchmark performance. Comparing this optimized Deriver configuration to CReM (Polishchuk, 2020), CReM does achieve slightly better performance on the Osimertinib MPO, Amlodipine MPO, and Valsartan SMARTS benchmarks (Figure 7, CReM), but it is less consistent across tasks and has a summed performance of 17.92 compared to 18.24 for the optimized Deriver (Supplemental Table 2). The story is similar for Molecule Swarm Optimization (Winter et al., 2019), which has superior performance on Median Molecules 1, Osimertinib MPO, and Perindopril MPO, but a total score of 18.09. It is clear that tuning and optimization can greatly boost both general and task-specific performance, and Deriver was designed to facilitate this process.

(Figure 7: Comparison to Reported Scores)

Discussion

Deriver is designed with an emphasis on inheritance, where new molecules are derived based on a relationship to “parent” molecules provided by the user. It leaves the choice of generators used, the parameters of those generators, and how they are combined to the user. The tunability and modularity of Deriver enables a high degree of user control to balance the many trade-offs inherent in chemical space exploration required from task-to-task. Two specific trade-offs in particular are well-handled: the compromise between designing molecules that fit arbitrary in-silico objectives while remaining pleasing to chemists (Brown et al., 2019), and the compromise between traversing chemical space efficiently while exploring local regions with high granularity (Polishchuk et al., 2020).

Deriver also facilitates the separation of objective optimization and discrimination of chemical ‘quality’, such that there does not need to be any undesired restriction on the available search space. While it is difficult to quantify the relative impact of the algorithmic filters vs satisfaction of other objectives, the delayed filter does greatly improve the percentage of top-scoring molecules that pass filters (100% on all benchmarks except Sitagliptin MPO at 45%), compared to the unfiltered molecules (40.35%), while only minimally impacting scores (Supplemental Table 1).

Efficiency of chemical space exploration is not only a concern for computational expense, but

also for the tractability of a problem. Enumerating and scoring all members of drug-like chemical space, an estimated 10^33 members which could ever be synthesized (Polishchuk et al., 2013) is not feasible, so limiting the search space in a useful way is highly desirable. For Deriver, combining generators with differing granularity can lead to both more rapid convergence and local exploration close to the (goal-specific) optimum. Similarly, combinations with coarser methods like the BRICS Deriver can enable escape from local optima, a known concern for fine-grained methods (Hartenfeller & Schneider, 2011). While in this report the same generator settings were used consistently across all generations of a given experiment, it is also possible to change the generators dynamically over time (e.g. becoming progressively finer grained, or modifying parameters such as fragment complexity or mutation probabilities).

Many hyperparameter optimization approaches (such as sequential model-based optimization (Bergstra et al., 2011)) could be applied to Deriver to automatically determine high-performance settings, either generally across benchmark tasks, or on a task-specific basis. Furthermore, as Deriver functions in part as a wrapper to other published generator methods, it is possible to extend functionality to include new generators such as CReM (Polishchuk, 2020) or incorporate other methods, like MSO (Winter et al., 2019). In principle, any system which generates valid molecules could be added as a generator in Deriver and combined with complementary approaches for more effective chemical space searches. The same extensibility applies to filtering methods, which can be expanded to include any set of SMARTS patterns, or other discriminative function. Deriver represents a philosophy for de novo drug design that is centered around inheritance of molecular structure, modularity, extensibility, and separation of concerns, while maximizing ease of use and modification. All the code to repeat the experiments in this study, as well as the source code for Deriver, are available on Github (https://github.com/cyclica/deriver), and Deriver can be installed easily via pip and the Python Package Index (pypi): https://pypi.org/project/deriver/ .

Figures

Figure 1

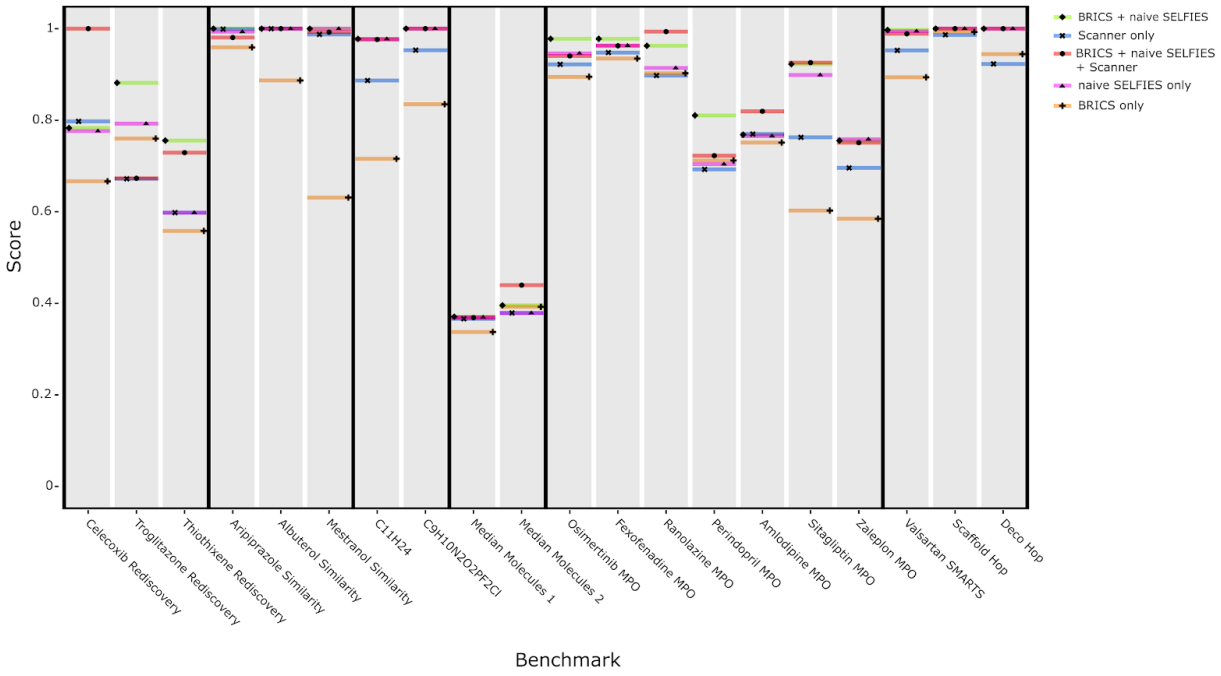


Figure 1: Comparison of Individual and Combined Generators with Greedy Sampling. The final score assigned by Guacamol to the returned sub-population is plotted on the y-axis, for each of the 20 standard goal-directed Guacamol benchmarks, shown on the x-axis. The combinations of Deriver generators used in each case are indicated by colored lines: BRICS only (orange), scanner only (blue), naive SELFIES only (magenta), BRICS + naive SELFIES (lime), and BRICS + naive SELFIES + Scanner (red). The benchmarks are additionally divided by black vertical lines into 6 categories provided by Guacamol: rediscovery, similarity, isomer, median, multi-parameter optimization, and multi-parameter optimizations including SMARTS.

Figure 2

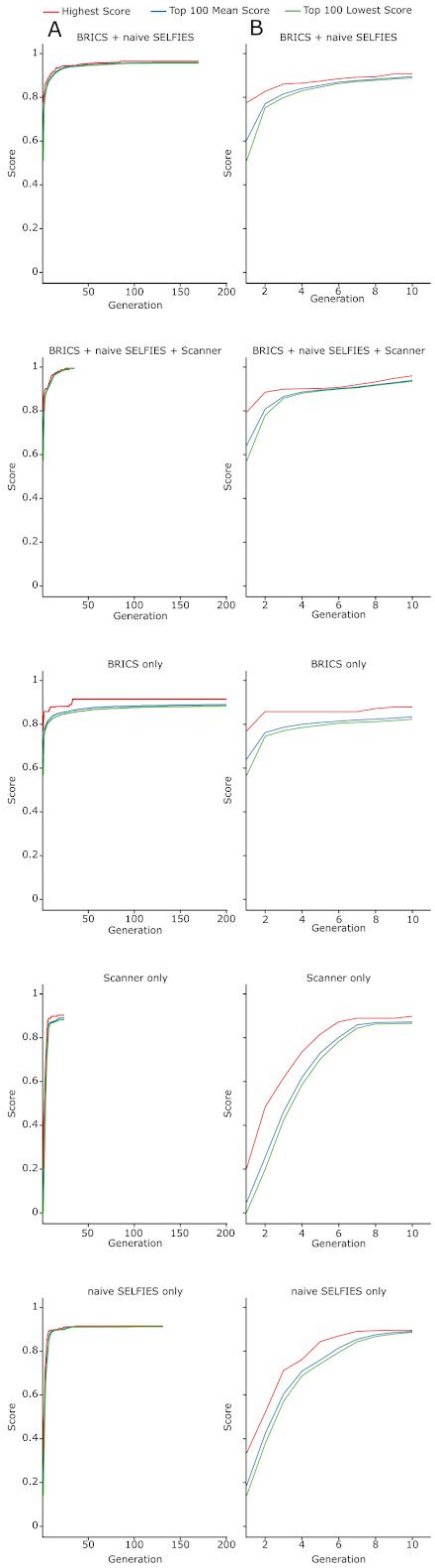


Figure 2: Comparison of Individual and Combined Generators on the Ranolazine MPO Benchmark. The scores (y-axis) of the top 100 molecules at a given generation (x-axis) are shown. The mean (blue line), maximum (red line), and minimum (green line) scores from this subpopulation are shown in each sub-plot representing one or more Deriver generators used alone or in combination. a) shows all generations on the x-axis, while b) provides a focused view of the first 10 generations.

Figure 3

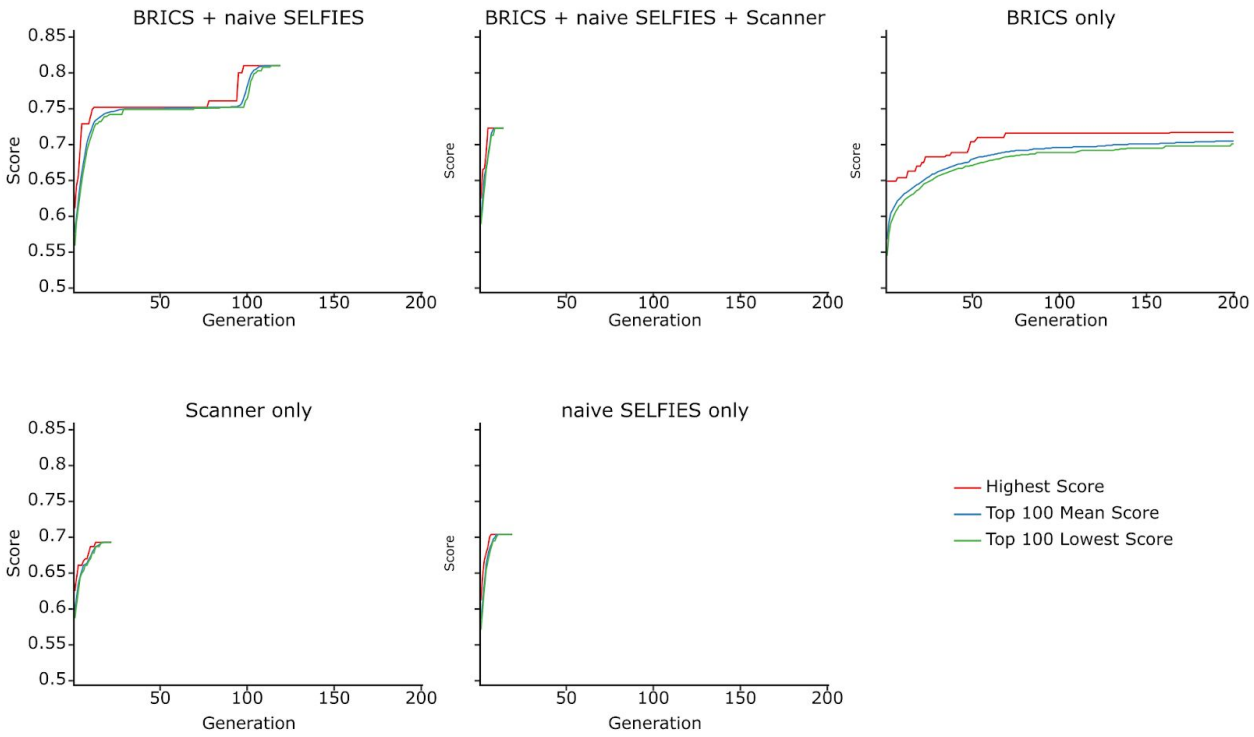


Figure 3: Comparison of Individual and Combined Generators on the Perindopril MPO Benchmark. The scores (y-axis) of the top 100 molecules at a given generation (x-axis) are shown. The mean (blue line), maximum (red line), and minimum (green line) scores from this subpopulation are shown in each sub-plot representing one or more Deriver generators used alone or in combination.

Figure 4

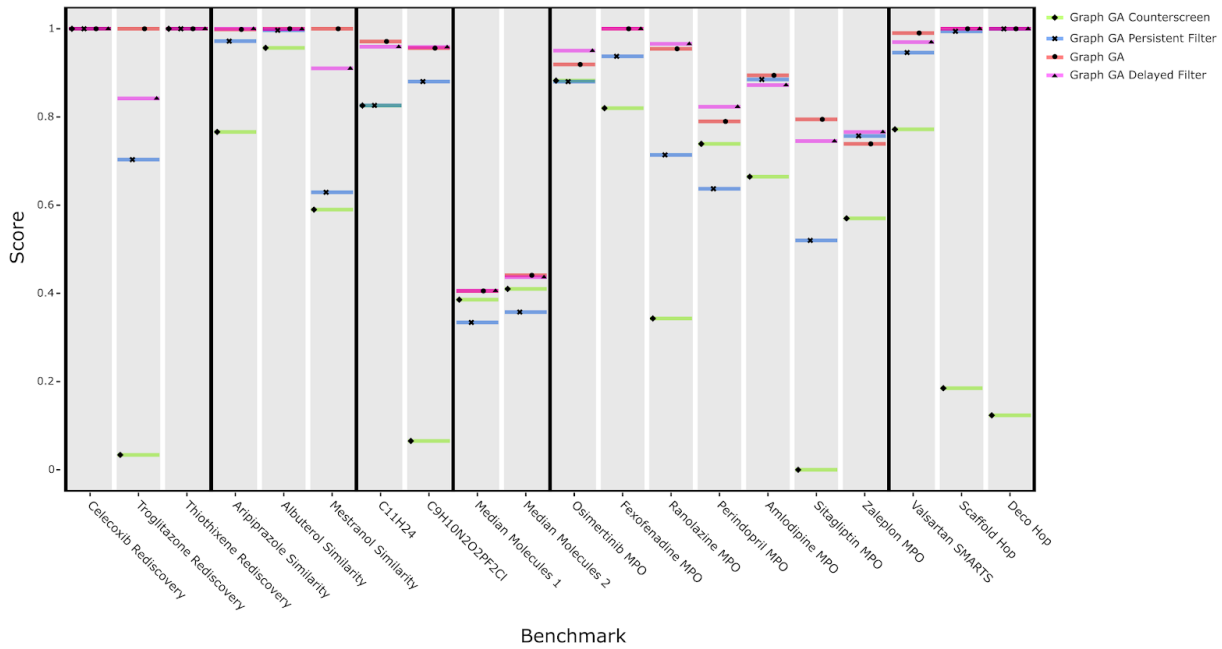


Figure 4: Comparison of Filtering Methods Applied to Graph_GA. The final score assigned by Guacamol to the returned sub-population is plotted on the y-axis, for each of the 20 standard goal-directed Guacamol benchmarks, shown on the x-axis. The filtering methods applied to the graph_GA implementation in Deriver are indicated by colored lines: no filtering (red), counter-screening (lime), persistent filtering (blue), and delayed filtering (magenta). The benchmarks are additionally divided by black vertical lines into 6 categories provided by Guacamol: rediscovery, similarity, isomer, median, multi-parameter optimization, and multi-parameter optimizations including SMARTS.

Figure 5

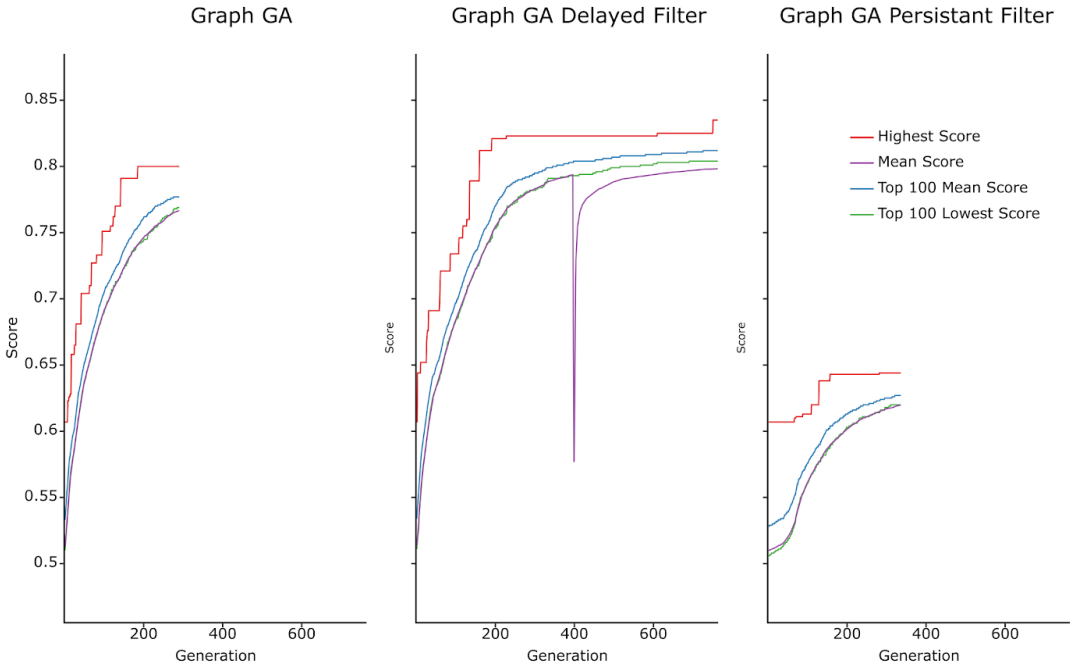


Figure 5: Comparison of Filtering Methods on the Perindopril MPO Benchmark. The scores (y-axis) of the top 100 molecules at a given generation (x-axis) are shown. The mean (blue line), maximum (red line), and minimum (green line) scores from this subpopulation are shown in each sub-plot representing one filtering scheme applied to the graph_GA Deriver implementation.

Figure 6

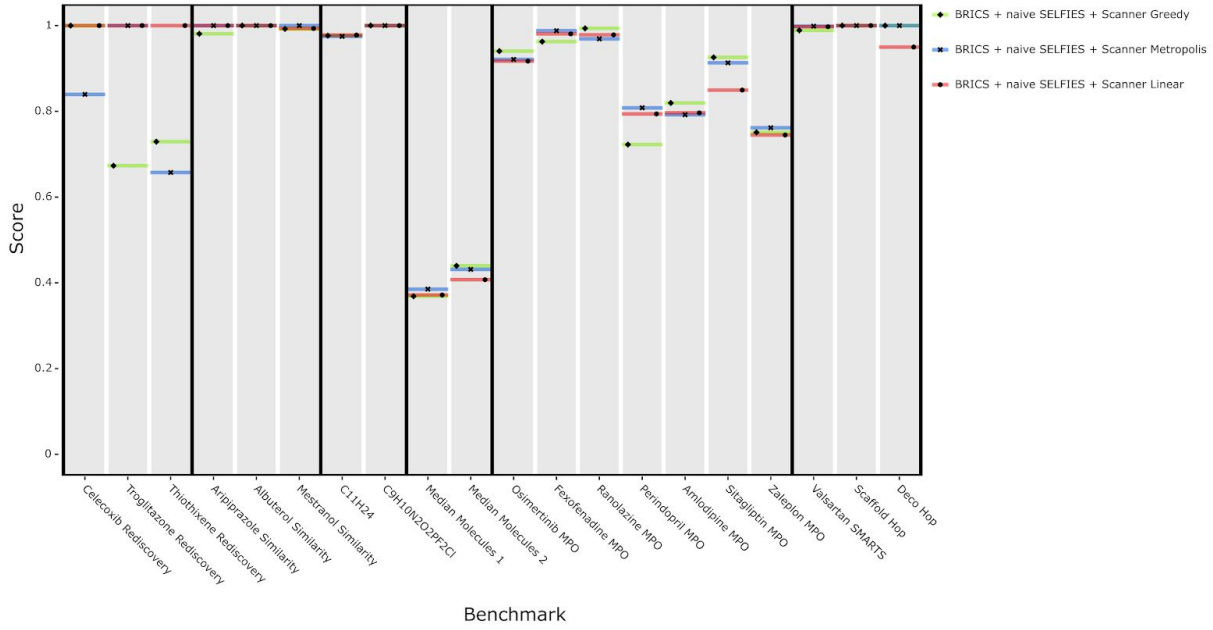


Figure 6: Comparison of Exploration Algorithms using the same Generator. The final score assigned by Guacamol to the returned sub-population is plotted on the y-axis, for each of the 20 standard goal-directed Guacamol benchmarks, shown on the x-axis. The sampling methods applied to the generator combination of BRICS, naive SELFIES, and Scanner are indicated by colored lines: greedy sampling (lime), Metropolis sampling (blue), and linear probabilistic sampling (red)). The benchmarks are additionally divided by black vertical lines into 6 categories provided by Guacamol: rediscovery, similarity, isomer, median, multi-parameter optimization, and multi-parameter optimizations including SMARTS.

Figure 7

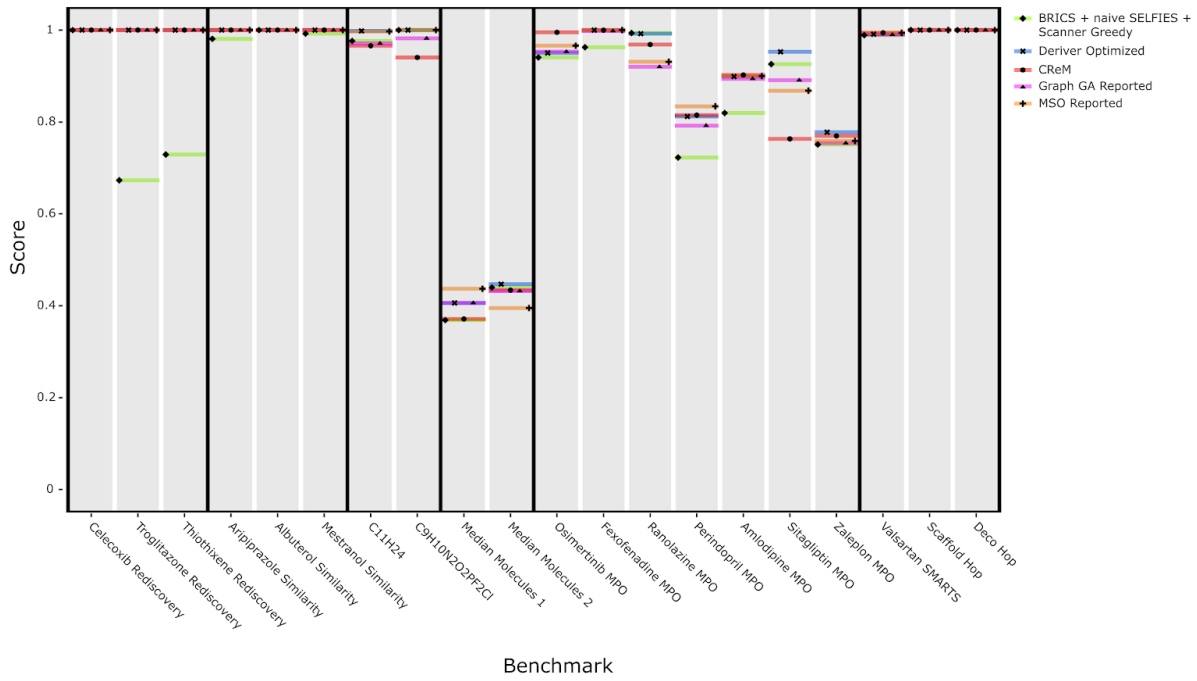


Figure 7: Comparison of Selected Deriver Schemas to Literature Results. The final score assigned by Guacamol to the returned sub-population is plotted on the y-axis, for each of the 20 standard goal-directed Guacamol benchmarks, shown on the x-axis. The reported-in-literature or observed (Deriver) result are indicated by colored lines: BRICS + naive SELFIES + Scanner with Greedy sampling (lime), an optimized Deriver schema (blue), CReM (Polishchuk, 2020) (red), graph_GA (Brown et al., 2019) (purple) and Molecule Swarm Optimization (Winter et al, 2019) (orange). The benchmarks are additionally divided by black vertical lines into 6 categories provided by Guacamol: rediscovery, similarity, isomer, median, multi-parameter optimization, and multi-parameter optimizations including SMARTS.

Tables

Table 1

Property	Default Minimum	Default Maximum
Molecular Weight (Da)	100	600
logP	-3	6
Hydrogen Bond Acceptors		10
Hydrogen Bond Donors		5
Total Polar Surface Area		180
Rotatable Bonds		30
Number of Rings		6
Ring Size		18
Number of Carbons	3	35
Number of Heteroatoms	1	15
Hydrogen/Carbon Ratio	0.1	1.1
Number of charges		4
Charge	-4	4
Number of Chiral Centers		2

Table 1: Default physicochemical property filters for Deriver. Each row represents a physicochemical property which is measured (via RDKit) and can be used to screen molecules. The columns indicate the allowed bounds (outside of which a molecule will be filtered) when this kind of filtering is enabled and it is not manually overwritten; if no lower (minimum) bound is specified, it can be assumed to be zero. Default parameters are chosen such that approximately 90% of FDA-approved drugs pass the filters.

Table 2

	Med. Chemist 1	Med. Chemist 2	Med. Chemist Consensus	Algorithmic Filters	Full Consensus	N
counterscreen	17	30	17	30	17	30
graph_GA_delayed_filter	40	86	40	100	40	100
graph_GA_filtered	40	85	37	100	37	100
unfiltered	32	65	31	30	17	100
total unique	98	202	94	194	80	264
percent passing	37.1	76.5	35.6	73.5	30.3	

Table 2: Counts of Acceptable Molecules by Medicinal Chemists and Algorithmic Filters. Three columns (Med. Chemist 1, Med. Chemist 2, Algorithmic Filters) show the number of "acceptable" molecules generated by a given experiment (row) which were acceptable to either a medicinal chemist or passed the algorithmic filters. The consensus columns correspond either to the number of molecules both chemists agreed to accept (Med. Chemist Consensus) or both chemists and algorithmic filtering agreed to accept (Full Consensus). The column N lists the total number of molecules from each experiment. The last two rows: total unique corresponds to the total number of unique molecules for each column; percent passing shows the percentage of total unique molecules (out of 264) which were accepted by the related discriminator.

Table 3

Pearson correlation	Med 1 vs SMARTS	Med 2 vs SMARTS	Med 1 vs Med 2
graph_GA_delayed_filter			0.32943
graph_GA_filtered			0.1715
unfiltered	0.34617	0.48038	0.45844
all	0.18782	0.40165	0.33013

Table 3: Pearson Correlation between Discriminative Mechanisms. Each entry represents the Pearson correlation between binary vectors representing whether to accept or reject each molecule, comparing the chemists' decisions with each other and with the SMARTS filtering for a given experiment (row). Blank entries exist when SMARTS filters accepted all corresponding molecules.

References

Altae-Tran, H., Ramsundar, B., Pappu, A. S., & Pande, V. (2017). Low Data Drug Discovery with One-Shot Learning. *ACS Central Science*, *3*(4), 283–293. https://doi.org/10.1021/acscentsci.6b00367

Arús-Pous, J., Blaschke, T., Ulander, S., Reymond, J.-L., Chen, H., & Engkvist, O. (2019). Exploring the GDB-13 chemical space using deep generative models. *Journal of Cheminformatics*, *11*(1), 20. https://doi.org/10.1186/s13321-019-0341-z

Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. *Neural Information Processing Systems*, 9. https://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf

Brereton, A., & Windemuth, A. (2020). *deriver: Deriver: for all your molecule generation needs.* (2.3.4) [Computer software]. Retrieved July 2, 2020, from https://pypi.org/project/deriver/

Brown, N., Fiscato, M., Segler, M. H. S., & Vaucher, A. C. (2019). GuacaMol: Benchmarking Models for de Novo Molecular Design. *Journal of Chemical Information and Modeling*, *59*(3), 1096–1108. https://doi.org/10.1021/acs.jcim.8b00839

Brown, N., McKay, B., Gilardoni, F., & Gasteiger, J. (2004). A Graph-Based Genetic Algorithm and Its Application to the Multiobjective Evolution of Median Molecules. *Journal of Chemical Information and Computer Sciences*, *44*(3), 1079–1087. https://doi.org/10.1021/ci034290p

De Cao, N., & Kipf, T. (2018). MolGAN: An implicit generative model for small molecular graphs. *ArXiv:1805.11973 [Cs, Stat]*. http://arxiv.org/abs/1805.11973

Degen, J., Wegscheid-Gerlach, C., Zaliani, A., & Rarey, M. (2008). On the Art of Compiling and Using "Drug-Like" Chemical Fragment Spaces. *ChemMedChem*, *3*(10), 1503–1507. https://doi.org/10.1002/cmdc.200800178

Dijkstra, E. W. (1982). On the role of scientific thought. In *Selected writings on Computing: A Personal Perspective* (pp. 60–66). Springer-Verlag New York, Inc.

Ertl, P., Lewis, R., Martin, E., & Polyakov, V. (2018). In silico generation of novel, drug-like chemical matter using the LSTM neural network. *ArXiv:1712.07449 [Cs, q-Bio]*. http://arxiv.org/abs/1712.07449

Green, D. V. S., Pickett, S., Luscombe, C., Senger, S., Marcus, D., Meslamani, J., Brett, D., Powell, A., & Masson, J. (2019). BRADSHAW: A system for automated molecular design. *Journal of Computer-Aided Molecular Design*. https://doi.org/10.1007/s10822-019-00234-8

Grisoni, F., Moret, M., Lingwood, R., & Schneider, G. (2020). Bidirectional Molecule Generation with Recurrent Neural Networks. *Journal of Chemical Information and Modeling*, *60*(3), 1175–1183. https://doi.org/10.1021/acs.jcim.9b00943

Hartenfeller, M., & Schneider, G. (2011). Enabling future drug discovery by *de novo* design. *WIREs Computational Molecular Science*, *1*(5), 742–759. https://doi.org/10.1002/wcms.49

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*(1), 97-109. https://doi.org/10.1093/biomet/57.1.97

Jensen, J. H. (2019). A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science*, *10*(12), 3567–3572. https://doi.org/10.1039/C8SC05372C

Jin, W., Barzilay, R., & Jaakkola, T. (2019). Junction Tree Variational Autoencoder for Molecular Graph Generation. *ArXiv:1802.04364 [Cs, Stat]*. http://arxiv.org/abs/1802.04364

Kadurin, A., Nikolenko, S. I., Khrabrov, K., Aliper, A., & Zhavoronkov, A. (2017). druGAN: An Advanced Generative Adversarial Autoencoder Model for de Novo Generation of New Molecules with Desired Molecular Properties in Silico. *Molecular Pharmaceutics*. https://doi.org/10.1021/acs.molpharmaceut.7b00346

Krenn, M., Häse, F., Nigam, A., Friederich, P., & Aspuru-Guzik, A. (2020). Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation. *ArXiv:1905.13741 [Physics, Physics:Quant-Ph, Stat]*. http://arxiv.org/abs/1905.13741

Kusner, M. J., Paige, B., & Hernández-Lobato, J. M. (2017). Grammar Variational Autoencoder. *ArXiv:1703.01925 [Stat]*. http://arxiv.org/abs/1703.01925

Kutchukian, P. S., Vasilyeva, N. Y., Xu, J., Lindvall, M. K., Dillon, M. P., Glick, M., Coley, J. D., & Brooijmans, N. (2012). Inside the Mind of a Medicinal Chemist: The Role of Human Bias in Compound Prioritization during Drug Discovery. *PLOS ONE*, *7*(11), e48476. https://doi.org/10.1371/journal.pone.0048476

Landrum, G. (2006). *RDKit: Open-source cheminformatics*. http://rdkit.org/

Lowe, D., & 2019. (2019, September 4). *Has AI Discovered a Drug Now? Guess.* SCience Translational Medicine. https://blogs.sciencemag.org/pipeline/archives/2019/09/04/has-ai-discovered-a-drug-now-guess

Lessel, U., & Lemmen, C. (2019). Comparison of Large Chemical Spaces. *ACS Medicinal Chemistry Letters*, *10*(10), 1504–1510. https://doi.org/10.1021/acsmedchemlett.9b00331

Neil, D., Segler, M., Guasch, L., Ahmed, M., Plumbley, D., Sellwood, M., & Brown, N. (2018). *Exploring Deep Recurrent Models with Reinforcement Learning for Molecule Design*. https://openreview.net/forum?id=Bk0xiI1Dz

Olivecrona, M., Blaschke, T., Engkvist, O., & Chen, H. (2017). Molecular de-novo design through deep reinforcement learning. Journal of Cheminformatics, 9(1), 48. https://doi.org/10.1186/s13321-017-0235-x

Polishchuk, P. (2020). CReM: Chemically reasonable mutations framework for structure generation. *Journal of Cheminformatics*, *12*(1), 28. https://doi.org/10.1186/s13321-020-00431-w

Polishchuk, P. G., Madzhidov, T. I., & Varnek, A. (2013). Estimation of the size of drug-like chemical space based on GDB-17 data. *Journal of Computer-Aided Molecular Design*, *27*(8), 675–679. https://doi.org/10.1007/s10822-013-9672-4

Renz, P., Van Rompaey, D., Wegner, J. K., Hochreiter, S., & Klambauer, G. (2020). On failure modes of molecule generators and optimizers. ChemRxiv. Preprint. https://doi.org/10.26434/chemrxiv.12213542.v1

Segler, M. H. S., Kogej, T., Tyrchan, C., & Waller, M. P. (2017). Generating Focussed Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ArXiv:1701.01329 [Physics, Stat]*. http://arxiv.org/abs/1701.01329

Verhellen, J. and Van den Abeele, J. (2020). Illuminating Elite Patches of Chemical Space. 10.26434/chemrxiv.12608228.v1

Virshup, A. M., Contreras-García, J., Wipf, P., Yang, W., & Beratan, D. N. (2013). Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. Journal of the American Chemical Society, 135(19), 7296–7303. https://doi.org/10.1021/ja401184g

Weininger, D. (1988). SMILES, a chemical language and information system. *Journal of Chemical Information and Computer Sciences*, *28*(1), 31–36. https://doi.org/10.1021/ci00057a005

Winter, R., Montanari, F., Steffen, A., Briem, H., Noé, F., & Clevert, D.-A. (2019). Efficient multi-objective molecular optimization in a continuous latent space. *Chemical Science*, *10*(34), 8016–8024. https://doi.org/10.1039/C9SC01928F

Zhavoronkov, A., Ivanenkov, Y. A., Aliper, A., Veselov, M. S., Aladinskiy, V. A., Aladinskaya, A. V., Terentiev, V. A., Polykovskiy, D. A., Kuznetsov, M. D., Asadulaev, A., Volkov, Y., Zholus, A., Shayakhmetov, R. R., Zhebrak, A., Minaeva, L. I., Zagribelnyy, B. A., Lee, L. H., Soll, R., Madge, D., … Aspuru-Guzik, A. (2019). Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nature Biotechnology*, *37*(9), 1038–1040. https://doi.org/10.1038/s41587-019-0224-x

Acknowledgements

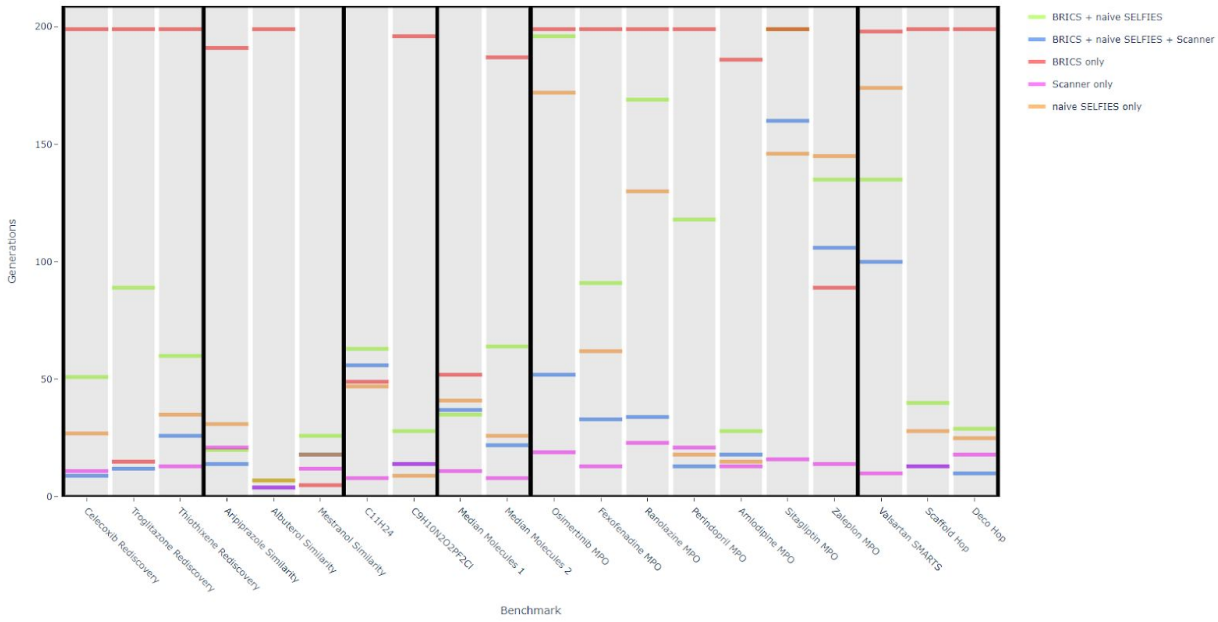
We thank Mario Krenn, Florian Hase, Pascal Friederich, Akshat Nigam, and Alán Aspuru-Guzik for their advice and their support on bugfixes and improvements to SELFIES. We thank Harsh Patel for contributions to the BRICS fragmentation code used by Deriver. We thank Naheed Kurji, Shoshana Wodak, and the Cyclica team for their unwavering support and advice. We thank everyone who is wearing masks, staying home, and generally taking the COVID-19 pandemic seriously.

Supplemental Information

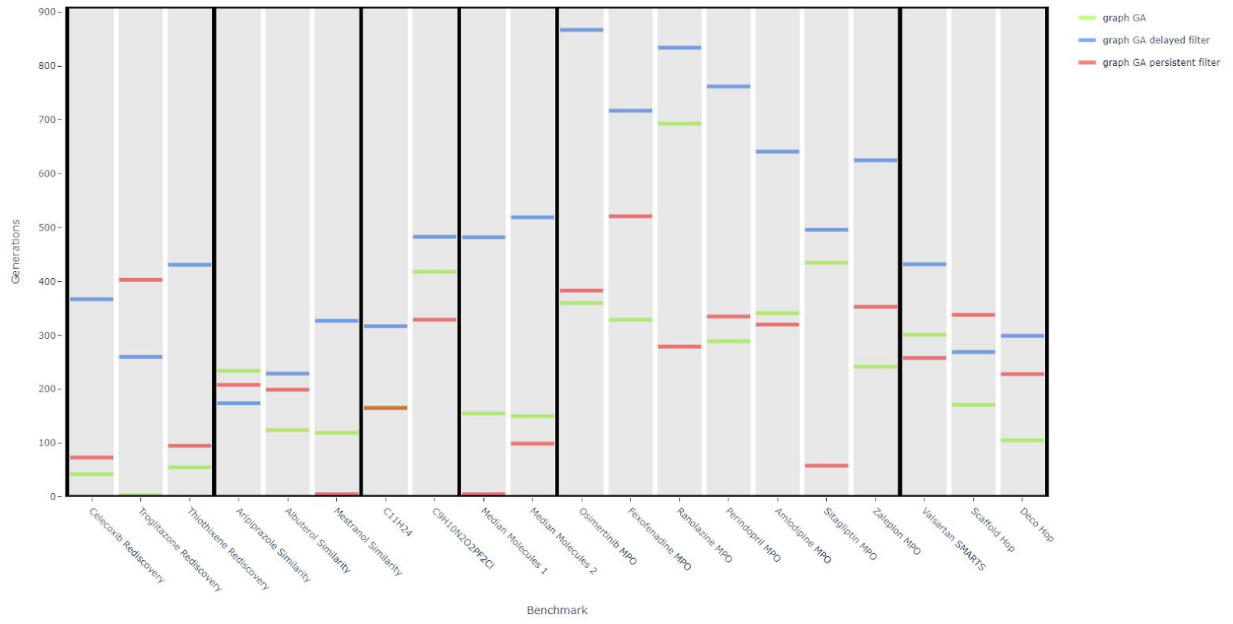
Supplementary Figure 1: Total Number of Generations for each Experiment Set

The following 5 figures display the total number of generations required to reach convergence (or be truncated at maximum allowed generations) corresponding to each experiment performed, as seen in the results section.The total generations are shown on the y-axis, for each of the 20 standard goal-directed Guacamol benchmarks, shown on the x-axis. Different coloured bars represent distinct Deriver parameter configurations.

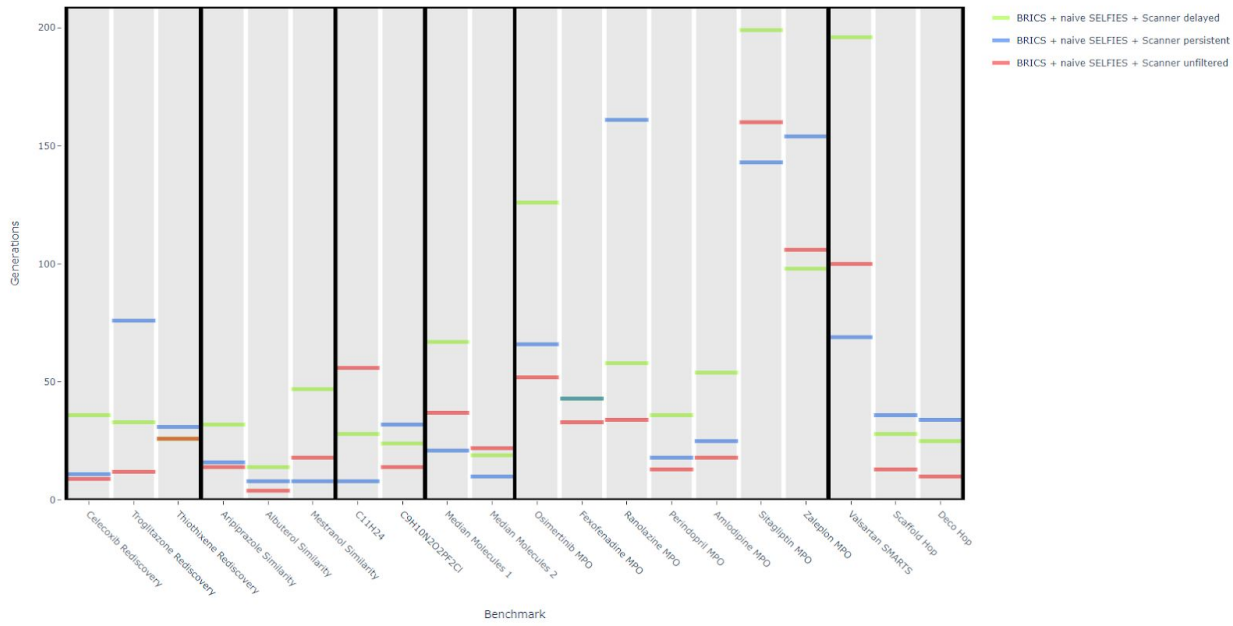
a) Total Number of Generations for each Combination of Generators



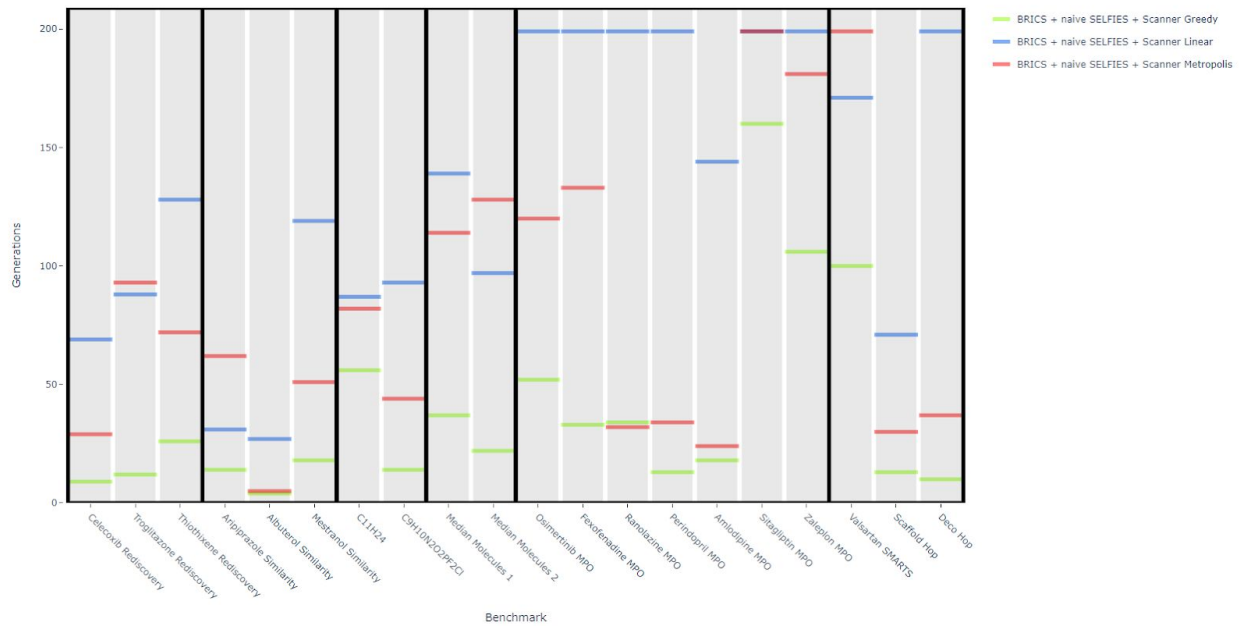
b) Total Number of Generations for Graph_GA Filter Experiment



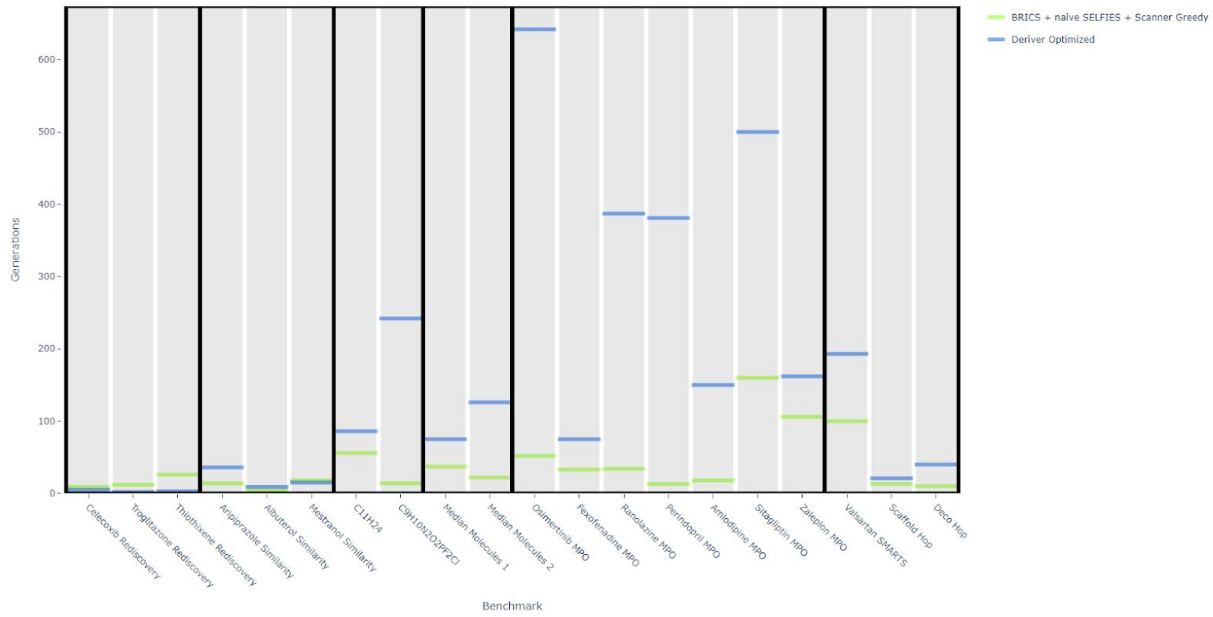
c) Total Number of Generations for Mixed-method Filter Experiment



d) Total Number of Generations for each Selection Method



e) Total Number of Generations for Optimized Deriver



Supplementary Table 1

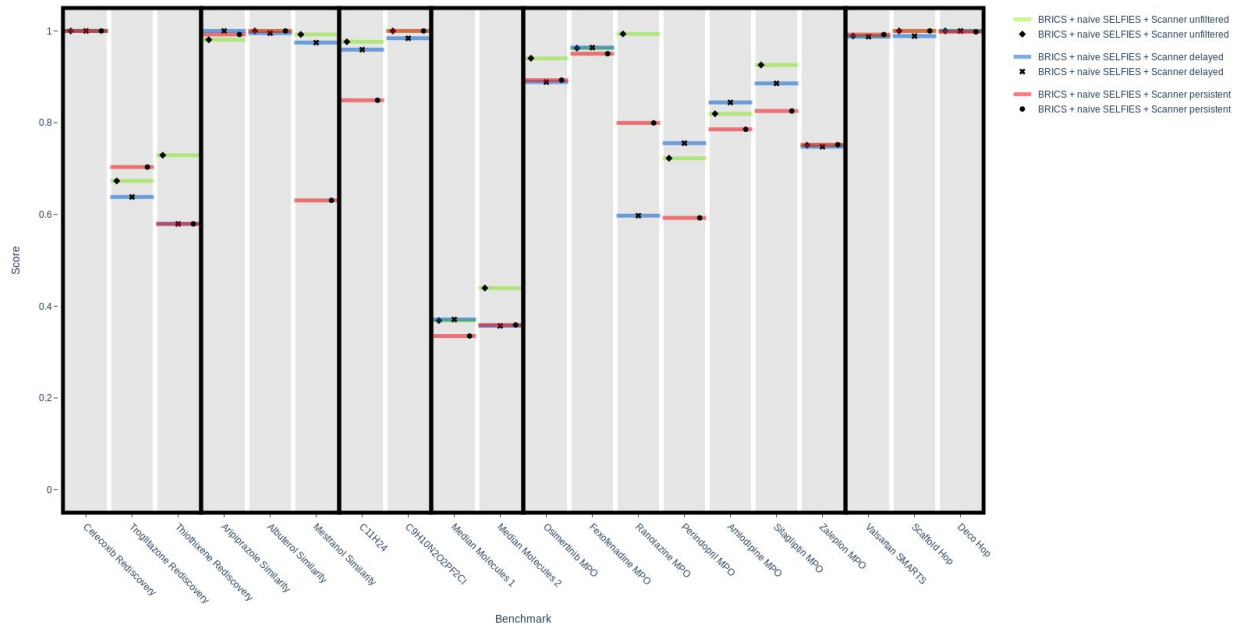
	unfiltered			filtered			delayed			counterscreen
	score	% selected	% top 100	score	% selected	% top 100	score	% selected	% top 100	score
Celecoxib rediscovery	1	100	47	1	100	95	1	100	100	1
Troglitazo	1	0	29	0.703	100	96	0.842	100	100	0.034

ne rediscovery										
Thiothixene rediscovery	1	100	54	1	100	100	1	100	100	1
Aripiprazole similarity	0.999	30	30	0.972	100	100	1	100	100	0.766
Albuterol similarity	1	87	87	0.997	99	99	0.999	100	100	0.957
Mestranol similarity	1	7	7	0.629	77	77	0.91	100	100	0.59
C11H24	0.971	84.9	88	0.826	95.6	97	0.959	100	100	0.826
C9H10N2O2PF2Cl	0.956	6.8	7	0.88	99.6	100	0.959	100	100	0.065
Median molecules 1	0.406	85	85	0.334	77	77	0.406	100	100	0.386
Median molecules 2	0.441	78	78	0.358	100	100	0.437	100	100	0.41
Osimertinib MPO	0.919	88	88	0.88	100	100	0.95	100	100	0.883
Fexofenadine MPO	1	46	46	0.938	100	100	1	100	100	0.82
Ranolazine MPO	0.955	3	3	0.726	100	100	0.966	100	100	0.343
Perindopril MPO	0.786	82	82	0.637	100	100	0.823	100	100	0.739
Amlodipine MPO	0.894	25	25	0.885	100	100	0.872	100	100	0.665
Sitagliptin MPO	0.795	0	0	0.52	87	87	0.745	45	45	0
Zaleplon MPO	0.739	30	30	0.73	100	100	0.766	100	100	0.57
Valsartan SMARTS	0.99	33	33	0.946	100	100	0.97	100	100	0.772
Scaffold Hop	1	0	0	0.994	53	53	1	100	100	0.185
Deco Hop	1	1	1	1	44	44	1	100	100	0.123

Supplementary Table 1: Quality of Graph_GA Molecules Generated Under Different Filtering Conditions. For each of the 20 Guacamol Benchmarks (rows) and filtering settings (major columns), the score of the selected sub-population returned to Guacamol (score), the percentage of these molecules which passed filters selected by Brown et al., 2019 (% selected)

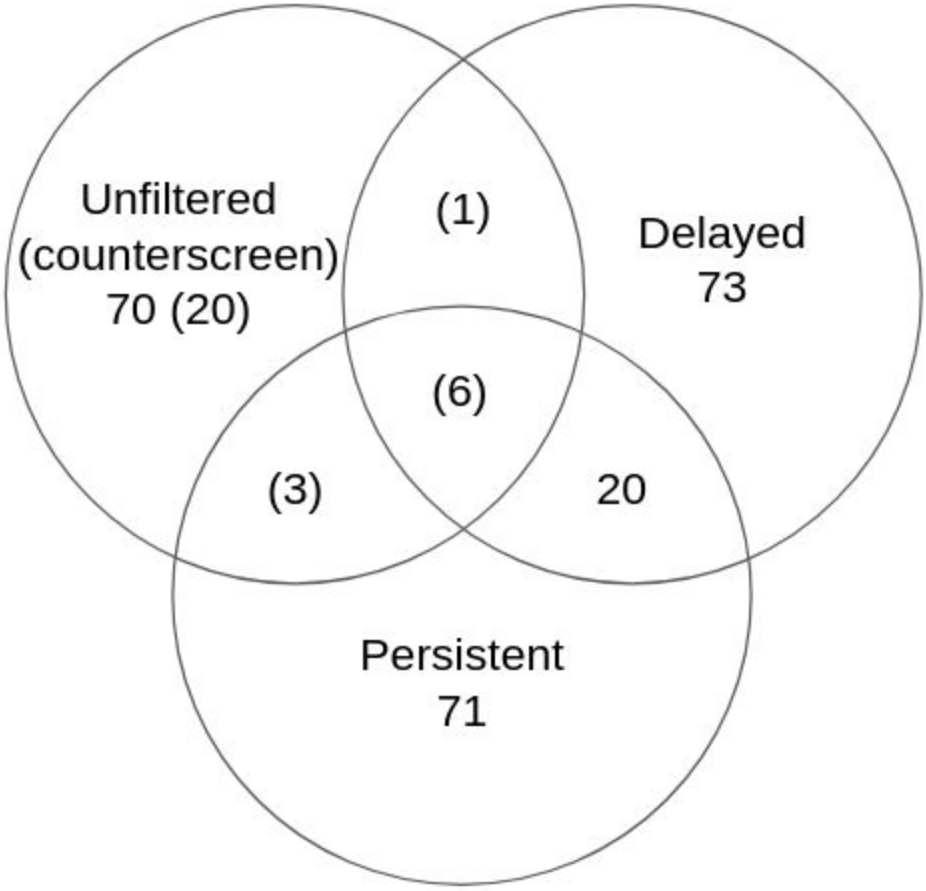
and the percentage of the top 100 highest scoring molecules in the final population which passed these filters (% top 100) are displayed. The % top 100 corresponds to the quality metric originally reported by Brown et al., 2019 for other baselines algorithms. The percentage passing columns are not shown for counter screening, since by design 100% of molecules pass.

Supplementary Figure 2



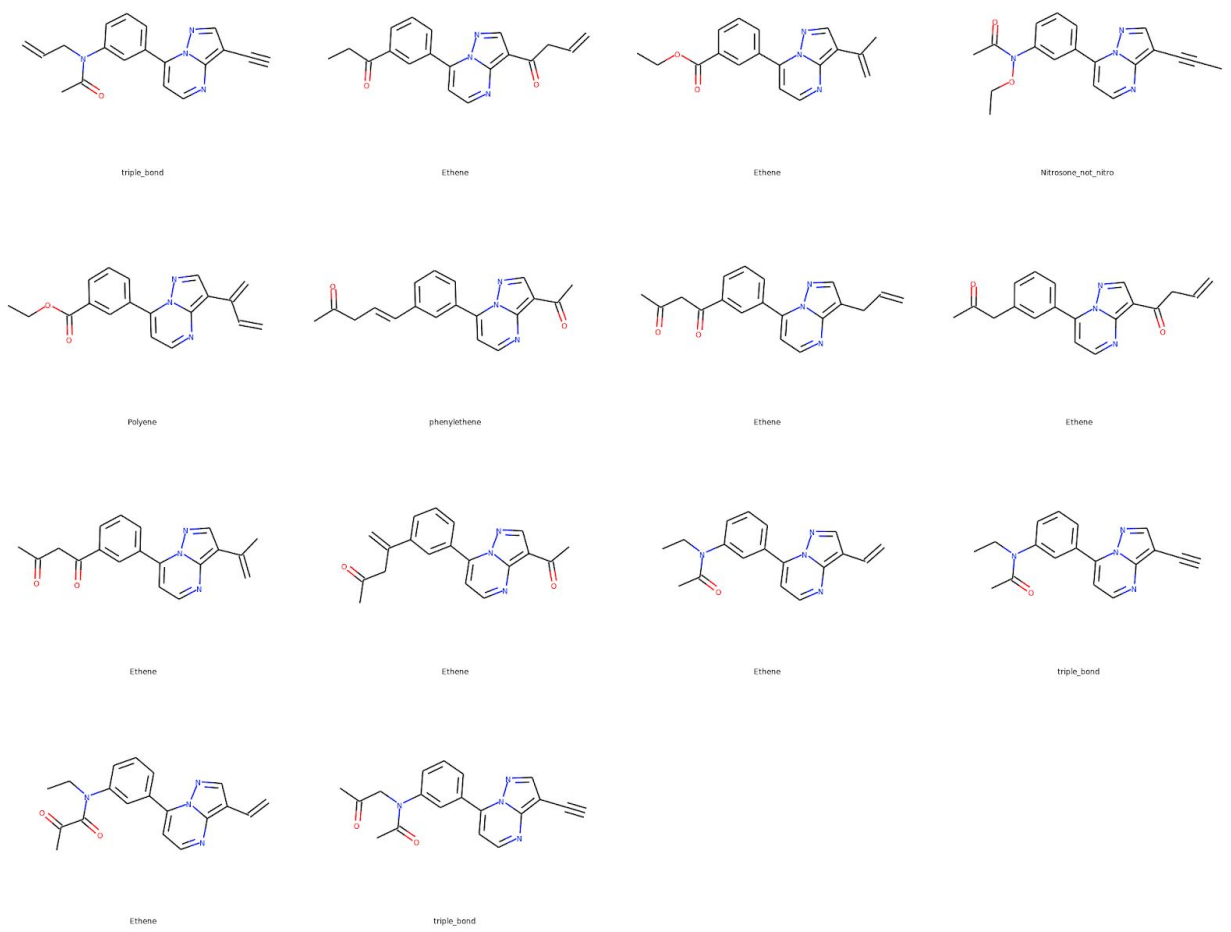
Supplementary Figure 2: Comparison of Filtering Methods Applied to a Mixed-method Generator. The final score assigned by Guacamol to the returned sub-population is plotted on the y-axis, for each of the 20 standard goal-directed Guacamol benchmarks, shown on the x-axis. The filtering methods applied to the combination of BRICS, naive SELFIES, and Scanner with greedy sampling as seen in Figure 1. The coloured bars divide the filtering methods: no filtering (lime), delayed filtering (blue), and persistent filtering (red). The benchmarks are additionally divided by black vertical lines into 6 categories provided by Guacamol: rediscovery, similarity, isomer, median, multi-parameter optimization, and multi-parameter optimizations including SMARTS.

Supplementary Figure 3



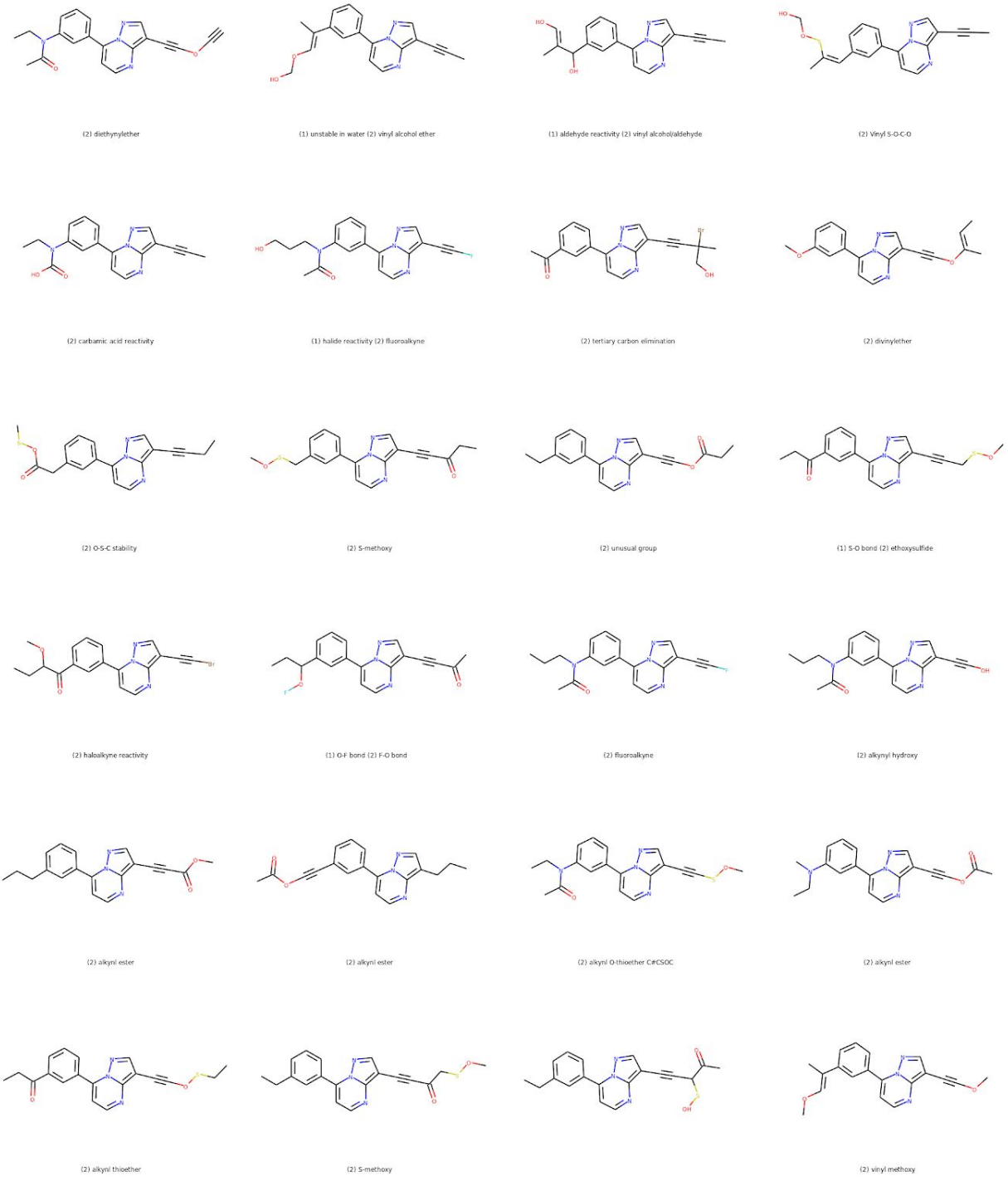
Supplementary Figure 3: Overlap Between Graph_GA Zaleplon MPO Molecules Created in Different Filtering Schemes. This Venn Diagram displays, from the 264 total unique molecules produced when each of the filtering methods are applied to graph_GA on the Zaleplon MPO Guacamol benchmark, which method or methods generated them. Molecules generated by the unfiltered method, but which passed the counter screen, are shown in parenthesis. Each method, aside from the counter screen, produced 100 molecules.

Supplementary Figure 4



Supplementary Figure 4: Molecules Accepted by Medicinal Chemists, but not by algorithmic filters. Shown above are the structures of a subset of all 264 unique molecules generated by different filterings settings applied to graph_GA on the Guacamol Zaleplon MPO benchmark, for which both medicinal chemists accepted the molecules, but they were rejected automatically by the algorithmic filters. The reason for rejection is listed below each molecule.

Supplementary Figure 5



Supplementary Figure 5: Molecules Accepted by algorithmic filters, but not by Medicinal Chemists. Shown above are the structures of a subset of all 264 unique molecules generated

by different experimental filter settings applied to graph_GA on the Guacamol Zaleplon MPO benchmark. These molecules were not rejected by the algorithmic filters, but were rejected by both medicinal chemists. The reason(s) they were rejected by the medicinal chemists are listed below each structure, with the preceding numbers (1,2) indicating which medicinal chemist gave each explanation.