JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

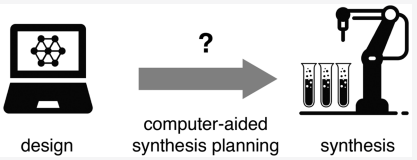
The Synthesizability of Molecules Proposed by Generative Models

Wenhao Gao and Connor W. Coley*

Cite This: J. Chem. Inf. Model. 2020, 60, 5714−5723 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The discovery of functional molecules is an expensive and time-consuming process, exemplified by the rising costs of small molecule therapeutic discovery. One class of techniques of growing interest for early stage drug discovery is *de novo* molecular generation and optimization, catalyzed by the development of new deep learning approaches. These techniques can suggest novel molecular structures intended to maximize a multiobjective function, e.g., suitability as a therapeutic against a particular target, without relying on brute-force exploration of a chemical space. However, the utility of these approaches is stymied by ignorance of synthesizability. To highlight the severity of this issue, we use a data-driven computer-aided synthesis planning program to quantify how often molecules proposed by state-of-the-art generative models cannot be readily synthesized. Our analysis demonstrates that there are several tasks for which these models generate unrealistic molecular structures despite performing well on popular quantitative benchmarks. Synthetic complexity heuristics can successfully bias generation toward synthetically tractable chemical space, although doing so necessarily detracts from the primary objective. This analysis suggests that to improve the utility of these models in real discovery workflows, new algorithm development is warranted.



INTRODUCTION

Molecular design is one of the most fundamental challenges in chemical science and engineering. This task is to identify one or more molecules with a specific set of properties of interest, such as binding affinity and drug-likeness for drug design. High-throughput virtual screening (VS) is one widely used strategy to coarsely optimize a molecular structure using a discretized subspace of the whole chemical space.[1] In VS, we evaluate enumerated candidate molecules in terms of their predicted properties of interest and ranked for follow-up experimental validation. However, because we rarely know *a priori* where the ideal molecule will be within the massive design space of chemical space, there is a trend toward using exceedingly large virtual libraries to increase the likelihood that we will find promising candidates. Modern virtual libraries may comprise hundreds of millions or billions of candidate molecules,[2] often generated through combinatorial enumeration of commercially available building block compounds. Even billions of compounds, however, represent a tiny fraction of theoretically possible, pharmacologically relevant small molecules, often cited as exceeding 10^{60} structures.[3] Brute-force virtual screening screening over a chemical space of this size is clearly computationally intractable.

Recent developments in computer aided drug design (CADD) techniques, especially in *de novo* molecular generation and optimization methods, raise the hope of removing this bottleneck.[4] Generative algorithms are a class of methods that propose molecular structures in a manner that can be tailored toward a specific objective. There is a long history of generative models in chemistry, many based on genetic algorithms[5] and the iterative construction of molecules from molecular fragments.[6] In the past decade, following on the advent of Variational Auto-Encoders (VAEs)[7] and Generative Adversarial Networks (GANs),[8] there has been a flood of new deep learning (DL) methods for this task.[9] Many of these methods learn a mapping from a continuous lower-dimensional real number space to a discrete chemical space. Jointly trained with a structure−property regression, one can obtain novel chemical structures conditioned on desired properties. More usefully, combining generative models with Bayesian optimization (BO), or directly using a heuristic optimization algorithm (e.g., a genetic algorithm (GA) or tree search (TS)), we can bias candidate generation toward the functionality we desire. Deep generative models are trained on a finite set of molecules to learn an underlying distribution of chemical space, where interpolation and extrapolation produce novel chemical structures. Enumerating every candidate molecule is thus unnecessary, and applying these models requires linear computational cost to generate multiple molecular structures once trained. Further, the generative algorithms can explore chemical space beyond the limited beginning pool and provide novel chemical structures with preferential intellectual property (IP) positions, whereas molecules in VS are often pre-existing. In recent years, generative models have been applied to various chemical discovery problems and have shown promise as a useful tool for the problem of molecular design.[10]

Special Issue: Generative Models for Molecular Design

Received: February 18, 2020
Published: April 6, 2020

https://dx.doi.org/10.1021/acs.jcim.0c00174
J. Chem. Inf. Model. 2020, 60, 5714−5723

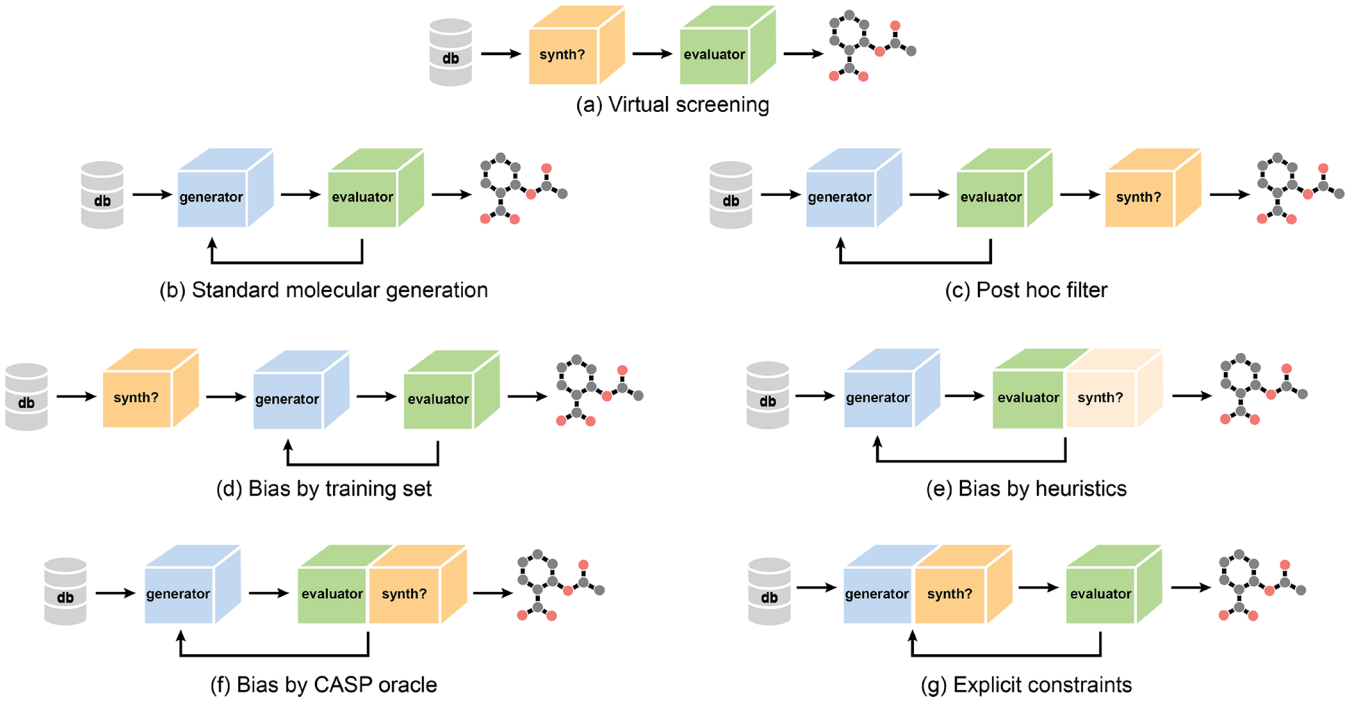


Figure 1. Schematic representation of approaches to address the challenge of synthesizability in molecular optimization: (a) virtual screening can use a filtered database of candidates to ensure that they are all synthetically accessible, (b) standard molecular generation focuses on evaluation of properties without regard for synthesizability, (c) a *post hoc* filter narrows down proposed candidates as a separate step from generation, (d) biasing by training set aims to improve synthesizability by training generative models on synthetically accessible compounds, (e) biasing by heuristics uses simple scalar proxies for synthesizability as part of the objective function, (f) biasing by a CASP oracle runs a full retrosynthetic expansion for proposed molecules to modify the reward function in a reinforcement learning setting, and (g) explicit constraints attempt to restrict chemical space to what is accessible using buyable building blocks and known synthetic transformations.

However, a practical problem that obstructs the usefulness of generative algorithms is that proposed molecular structures may be challenging or infeasible to synthesize. In any realistic discovery scenario, we will need to validate whether a proposed molecule has the property profile we expect; even if our computational models are infallible, we will need to manufacture the molecule in order to apply it (e.g., as a therapeutic, as a catalyst, as a component of a device). Libraries for virtual screening can be constructed from commercially available databases. They are often enumerated using well-characterized reaction templates to *try* to ensure that enumerated molecules are readily synthesizable. Lyu et al. report an 86% successful synthesis rate among 51 top-ranking molecules from a library comprising 99 million structures, consistent with the claims of many chemical vendors.[2]

The situation is quite different in *de novo* molecular design, especially when using deep generative methods. We expect (and want) these models to explore molecular structures beyond the ones they have been trained on, so they may propose nonsensical structures that are unreasonable for pharmaceutical purposes. There have been few studies explicitly examining this problem, but some anecdotal evidence suggests that compounds are not easily synthesizable—many structures reported in papers indeed appear absurd. Bjerrum and Threlfall examined 21 molecules proposed by their recurrent neural network (RNN) model with Wiley's ChemPlanner and found a number of possible selectivity issues in the proposed syntheses, indicating synthetic difficulty.[11] Sumita et al. filter generated molecules by requiring that they be previously reported with at least one synthetic route in SciFinder, which removes these models' ability to propose novel chemical structures.[12] Zhavoronkov et al. select only six molecules from 40 candidate structures based on synthetic accessibility, even after filtering an initial list of 30,000 structures generated by a deep learning model.[10]

Current procedures for quantifying synthesizability are based on (1) structure complexity and similarity or (2) synthetic pathways. The structure-based approach usually involves constructing a heuristic definition based on domain expertise or chemical substructure diversity[13,14] or designing a model that can be fit to expert scores[15−17] or reaction data.[18,19] This kind of method is widely used due to its ease of implementation and low computational cost. However, two similar structures with a single functional group transposition can require substantially different synthetic routes (e.g., due to the selectivity of chemical reactions or availability of specific building blocks), which makes it challenging to fit a good proxy score (see Figures S1 and S2 for one example). The most convincing metric might be a direct scoring from a group of experts on synthetic, medicinal chemistry, which has been used as a ground truth to train models against.[15−18] To have a group of experts large enough to reach a nonbiased and stable value is labor-intensive, hard to replicate, and not scalable.[20]

The second, more nuanced approach to measuring synthesizability is to explicitly plan a synthetic pathway and assess its likelihood of experimental validity. Synthetic pathway-based approaches can incorporate more thorough information about starting materials and chemical reactions, which enables them to overcome the shortcomings of the structure-based analysis. In this approach, a computer-aided synthesis planning (CASP) program[21] can be used to perform the retrosynthetic analysis. The use of an explicit CASP tool

https://dx.doi.org/10.1021/acs.jcim.0c00174
J. Chem. Inf. Model. 2020, 60, 5714−5723

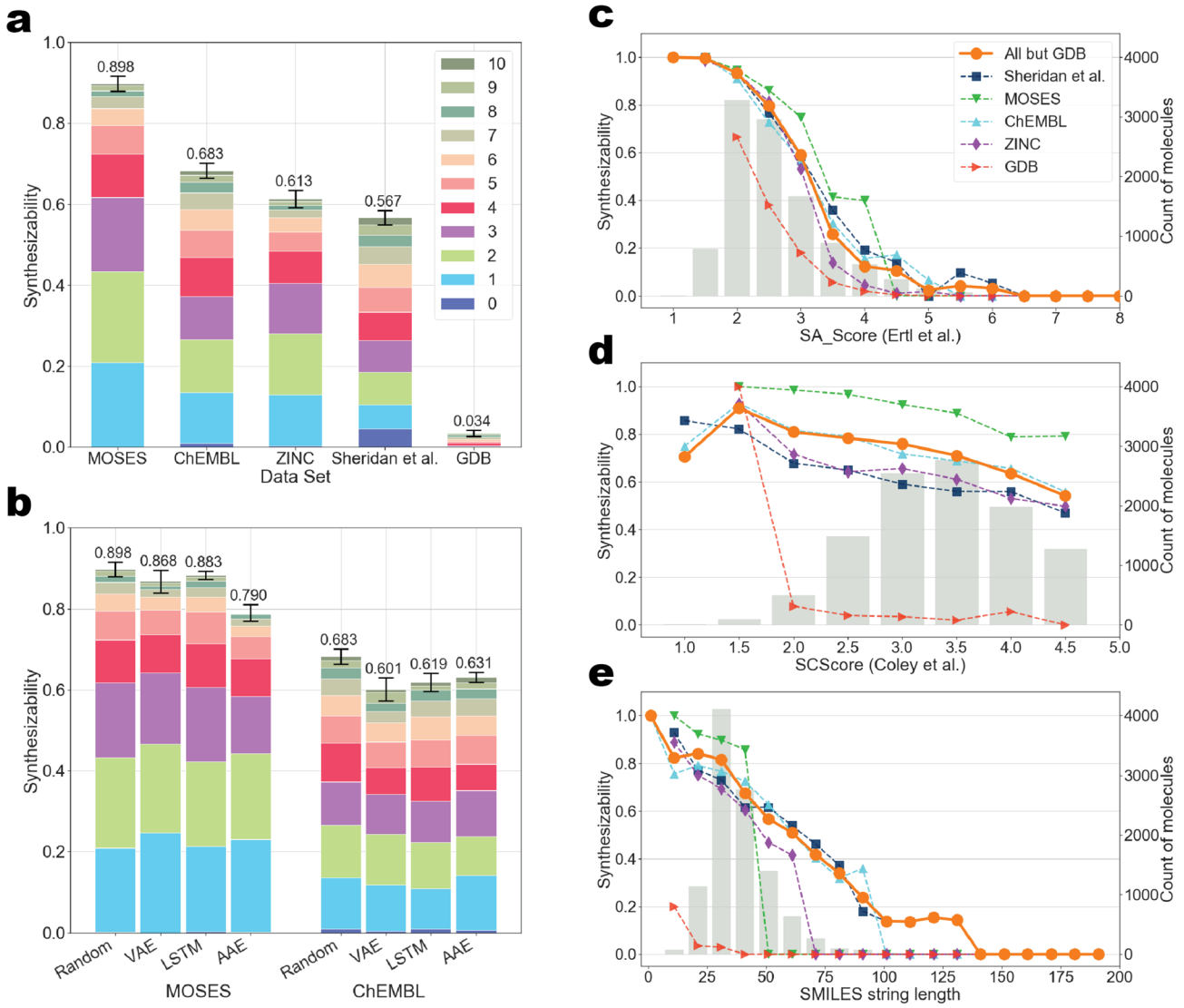


Figure 2. Synthesizability analysis of common data sets, distribution learning, and popular heuristics. (a) The number of synthetic steps required to produce random sampled structures from each data set; error bars represent the standard deviation between random samplings of 300 molecules for each batch, 3000 in total (except Sheridan's data set). (b) The number of synthetic steps required to produce molecules generated by distribution learning algorithms, trained on either MOSES or ChEMBL; error bars represent the standard deviation between 3 batches of nonoverlapping generation, 300 molecules per batch. (c–e) The fraction of synthesizable compounds from each data set binned by heuristic score and the number of molecules scored within each bin (excluding GDB).

makes it possible not only to capture the high "nonlinearity" of synthesizability with respect to chemical structure but to recommend actionable synthetic pathways. We see this as a form of interpretability to verify *why* the molecule is believed to be synthesizable, with which building blocks, and in how many steps. Only a handful of studies have used a retrosynthetic planning tool to analyze synthesizability.[11,22−24] Its practical application in molecular design is not widespread yet. Therefore, here, we analyze synthesizability of compounds proposed through generative algorithms using our open-source computer-aided retrosynthesis analysis tool, ASKCOS.[25]

We divide our analysis of the synthesizability of molecules generated by *de novo* generative algorithms into evaluations of *distribution learning* and *goal-directed generation* tasks—unoptimized and optimized molecules, respectively. Distribution learning models are meant to interpolate within a chemical space comprised of a training set of molecules and to generate new molecules with similar properties. Goal-directed generation instead tries to generate new molecules that maximize a black-box scoring function. There have been an increasing number of algorithms of these two categories proposed in recent years and a small number of studies that benchmark these algorithms in terms of their ability to generate novel, optimal molecules.[26,27]

We categorize the approaches one might take to ensure that computationally designed molecules are able to be synthesized in Figure 1. These represent combinations of (i) a database of known or enumerated compounds, (ii) an evaluator, which estimates the properties we are trying to optimize, (iii) a generator function, which can propose new candidate molecules, (iv) a synthesizability oracle that determines whether it is straightforward to synthesize a given molecule, and/or (v) a heuristic synthesizability estimator that provides a computationally inexpensive scalar measure of synthesizability. In this study, we focus on three major approaches to solving the synthesizability problem: *post hoc* filtering (Figure 1c),

https://dx.doi.org/10.1021/acs.jcim.0c00174
J. Chem. Inf. Model. 2020, 60, 5714−5723

imposing *a priori* differences in training sets (Figure 1d), and heuristic biasing (Figure 1e).

RESULTS

Synthesizability of Common Databases According to ASKCOS. We first validate that the information returned by ASKCOS is usefully correlated with synthesizability by analyzing molecules from several standard compound libraries: MOSES,[26] ChEMBL,[28] ZINC,[29] Sheridan et al.,[16] and GDB17[30] (see Methods for detailed descriptions of each data set and the settings used for retrosynthetic analysis, including the evaluation of commercial availability of building blocks). Figure 2a shows the predicted number of synthetic steps required to produce a random set of 3000 molecules from each data set. The MOSES data set has the highest rate of perceived synthesizability at 89.8%, consistent with its focus on small lead molecules and exclusion of compounds with "structural alerts." Its parent set, ZINC, has a lower synthesizability rate of 60.8%. The ChEMBL data set has a higher rate of 68.3%; although it contains larger and more complex structures than does ZINC, many have been synthesized previously; among those that cannot be synthesized are natural products that were extracted, not synthesized, and tested for their biological activity. ChEMBL also contains several directly purchasable compounds, second only to Sheridan et al.'s data set of 1730 compounds. Unsurprisingly, the exhaustively enumerated data set, GDB17, has the lowest rate of synthesizability at only 3.5%. We also find that the predicted number of reaction steps is correlated with expert-provided scores (Figure S14). From these trends and the high success rate of the MOSES database, we conclude that ASKCOS's retrosynthetic analyses are largely consistent with our expectations of synthesizability, and it is appropriate to use its predictions to benchmark the evaluation of molecular generation.

Agreement between Synthesizability Heuristics and ASKCOS. We next evaluate the agreement between several heuristic synthesizability scores (length of SMILES, SA_Score,[31] and SCScore[19]) and the results of ASKCOS. Because retrosynthetic analysis can be time-consuming (tens to hundreds of CPU-seconds), we would prefer to bias generation by heuristics rather than by a CASP oracle (cf. Figure 1). Figure 2c−e show the trend of synthesizability of structures in different ranges of SA_Score, SCScore, and SMILES string lengths. None of them can distinguish the synthesizable and unsynthesizable compounds perfectly, but all exhibit a decreasing trend as the heuristic score increases. The trend is clearest for the SA_Score, followed by the SMILES length and then the SCScore. This ordering is quantified in Figure S3, which shows the area under the receiver operating characteristic as if these heuristics were being used for binary classification. The AUC values for the three methods in this order are 0.87, 0.69, and 0.61. The slight shoulder around 5.5−6.0 in Figure 2c is the contribution from the structurally complex but commercially available compounds, highlighting the difference between synthetic complexity and structural complexity as discussed in ref 19. We note that the superiority of the SA_Score over SCScore is not necessarily surprising given their respective definitions (see Methods): the SA_Score penalizes rare substructural fragments that are not widely observed in PubChem, while the SCScore evaluates whether one molecule is more or less likely to be a reactant than another and is trained on reactions from Reaxys. The rarity of a substructure indicates both that it is unlikely to appear in a commercially available reactant and that it is likely challenging to synthesize.

Synthesizability of Unoptimized Generated Molecules. As alluded to above, distribution learning methods are capable of generating "unoptimized" molecules that share properties (in aggregate) with the database used for training. Here, we evaluate methods implemented in the MOSES[26] benchmarking set, which cover diverse approaches to the molecular generation problem: a SMILES long short-term memory (LSTM) model, a variational autoencoder (VAE), and an adversarial autoencoder (AAE; see Methods). There are more deep learning approaches for molecular generation and optimization than can be compared here,[9] so we focus on these top-performing classes of approaches. In this task, we can use *post hoc* filtering or training set biasing by separately training distribution learning models on ChEMBL (less synthesizable) and MOSES (more synthesizable).

Figure 2b shows the fraction of synthesizable molecules from 300 generated by each distribution learning method trained on the ChEMBL and MOSES. We observe that the fraction of synthesizable molecules is comparable to that of the training set, while no method *improves* synthesizability relative to its training set. The stark difference between results using MOSES and ChEMBL suggests that *a priori* biasing by training on a "more synthesizable" data set is a viable approach for distribution learning algorithms. There is no one method that is particularly superior to others. The high fraction of synthesizable results further suggests that *post hoc* filtering is not necessarily a bad approach (i.e., relatively few generated molecules would fail a check for synthesizability). Note these results pertain only to the synthesizability of generated results and do not consider previously evaluated metrics of novelty, uniqueness, and diversity as do Polykovskiy et al.'s analyses.[26] As their and others' analyses have shown, distribution learning methods can successfully mimic the training data set with respect to simple properties like molecular weight, calculated partition coefficient, SA_Score, quantitative estimation of drug-likeness, etc. We hypothesize that these models learn, implicitly, what common functional groups and structural motifs comprise these molecules and which tend to be copresent, which is how they are able both to recapitulate the distribution of scalar descriptors—as others have shown—and to recapitulate synthesizability—as our new results show.

Synthesizability of Optimized Generated Molecules. Our next analyses focus on *goal-directed benchmarks*, which reflect the actual use-case for generative models. Here, we re-evaluate the methods and objective functions evaluated by Brown et al.'s Guacamol[27] in terms of their synthesizability. As detailed in the Methods, this includes three generative algorithms (SMILES LSTM, SMILES GA, and Graph GA) and 14 multiproperty objective functions (MPOs) that convert a molecular structure to a scalar fitness score. As a baseline method, we include a virtual screening approach, "Best from Data," where all candidates from either ChEMBL or MOSES are evaluated to identify the top performers. In addition to *post hoc* filtering and training set biasing, we can also bias generation by modifying the objective function with a heuristic synthesizability score. We multiply the original objective functions (normalized between 0 and 1) with a quantitative synthesizability metric (SA_Score or SCScore, also normalized between 0 and 1). More details can be found in the Methods section.

https://dx.doi.org/10.1021/acs.jcim.0c00174
J. Chem. Inf. Model. 2020, 60, 5714−5723

We evaluate the effects of heuristic biasing both in terms of the synthesizability of suggested molecules and in terms of the primary objective function value. Figure 3 shows four examples

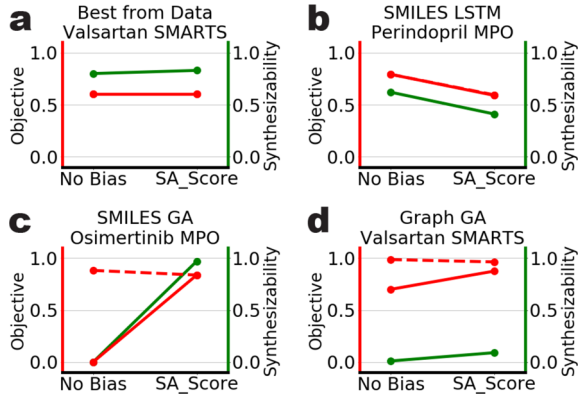


Figure 3. Dependence of goal-directed optimization performance on heuristic biasing by the SA_Score using ChEMBL as the training database, for four exemplary method-objective combinations. In each plot, the green solid line represents the change of fraction of synthesizable compounds in the top 100. Red solid lines represent the change in the objective function value of the top *synthesizable* molecule, while the dashed red line represents the change in objective function value of the top molecule, regardless of its synthesizability; dashed red lines may be occluded by solid red lines.

of how these metrics change when biasing with SA_Score, initially trained on ChEMBL (Figure S5 shows the full results of all methods, biasing strategies, training data sets, and objective functions). Figure 3a shows a case where there is no improvement in the objective function value of the top synthesizable molecule, as the majority of molecules selected from the data set are already synthesizable. Figure 3b shows a case where heuristic biasing actually decreases the synthesizability of proposed molecules as well as their objective function values. However, Figure 3c shows a dramatic success of heuristic biasing, where *none* of the proposed compounds are synthesizable in the absence of biasing; biasing leads to almost all proposed compounds being synthesizable and, despite doing so, does not significantly detract from the objective function value (see the top compounds for this case in Figure 4a). The fourth case in Figure 3d shows only an incremental increase in synthesizability upon biasing, but one that results in a moderate increase of objective function value for the top synthesizable compound (see the top compounds for this case in Figure 4b).

In the full results (Figure S5), we find that the synthesizability varies significantly between different methods and objectives. Indeed, the total fraction of synthesizable compounds in all methods for "hard" objectives without biasing is 30.2% with ChEMBL and 32.7% with MOSES (see Figure S6 for more details), excluding the direct sampling from data set. Compared to distribution learning, the goal-directed generation methods are less sensitive to the starting set of molecular compounds. For several tasks (Figure S6), very few or no compounds in the top 100 are synthesizable in the absence of heuristic biasing, particularly when using the genetic algorithms, illustrating the risk of relying on a *post hoc* filtering strategy.

Examples in Figure 4a,b illustrate cases where no molecule in the top 100 is synthesizable and heuristic biasing is required to generate even a single feasible candidate. The compounds remaining after filtering for synthesizability, if any, may have low objective function values. In extreme cases, particularly the unbiased proposal in Figure 4a, generated molecules contain substructures that could easily be flagged by an expert-encoded set of substructural rules. Such substructure filters are commonly employed in virtual screening pipelines to identify compounds with "undesirable motifs" (e.g., those that are likely to interfere with biological assays and lead to unreliable results[32,33]) and could be extended to include the most frequently proposed "unsynthesizable" motifs.

Most cases in Figure S5 show that the synthesizability of the top 100 compounds after biasing is quite high, often exceeding the rate for ChEMBL. Generally speaking, the SA_Score performs better than SCScore: the overall synthesizability for hard objectives was improved from 30.2% to 80.2% or 55.4% when biasing by SA_Score or SCScore, respectively, originally trained on ChEMBL (Table 1). The superiority of the SA_Score over the SCScore is not surprising given the trends in Figure 2d,e. Nevertheless, the successful increase in synthesizability validates the approach shown in Figure 1e, but the increased synthesizability comes at the expense of the objective function value of the top candidate. For some tasks, there are decreases of over 0.2—a significant difference for these benchmark tasks. However, we note that the value of an *in silico* objective function is completely inconsequential if the molecule cannot be made and experimentally tested.

A fairer comparison can be made between the objective function values of the top *synthesizable* candidates, i.e., after *post hoc* filtering. Figure 4c,d shows two examples where the objective of the top-1 candidate decreased, but the value of the top-1 synthesizable candidate increased. That this is observed in some cases (also see Figure S13) suggests a practical workflow for molecular optimization: if only a few synthesizable candidates (1−10) are desired, first optimize without biasing and filter unsynthesizable suggestions; if the top synthesizable candidates are worse than the top unsynthesizable candidates, repeat the optimization while biasing with the SA_score.

Discussion of Other Approaches. As described in Figure 1, there are more ways to improve synthesizability of *de novo* molecular generation algorithms. One promising approach is to bias the generation using a full CASP tool to evaluate synthesizability, instead of a proxy score (Figure 1f). The advantages are already described above; the disadvantage is the computational expense. While ASKCOS finds pathways in a few seconds for some molecules, we spend up to 1 min evaluating each molecule to reduce the number of false negatives.

Benchmarking for molecular optimization, in addition to neglecting synthesizability, has largely neglected the number of objective function calls and computational expense. When using genetic algorithms for molecular optimization, we would first select high scoring synthesizable compounds as the initial set to propagate from a pool of up to millions of structures ($\sim 10^6$) and then score, at each of hundreds of iterations ($\sim 10^3$), hundreds of child compounds ($\sim 10^3$). In total, we would require millions or at least hundreds of thousands of calls to the CASP oracle. Reinforcement-learning-based optimization methods that outperform Bayesian optimization when using VAEs require one oracle call per iteration but require hundreds of thousands or millions of iterations (e.g., MolDQN reports the use of 200k function calls[34]). One study

https://dx.doi.org/10.1021/acs.jcim.0c00174
J. Chem. Inf. Model. 2020, 60, 5714−5723

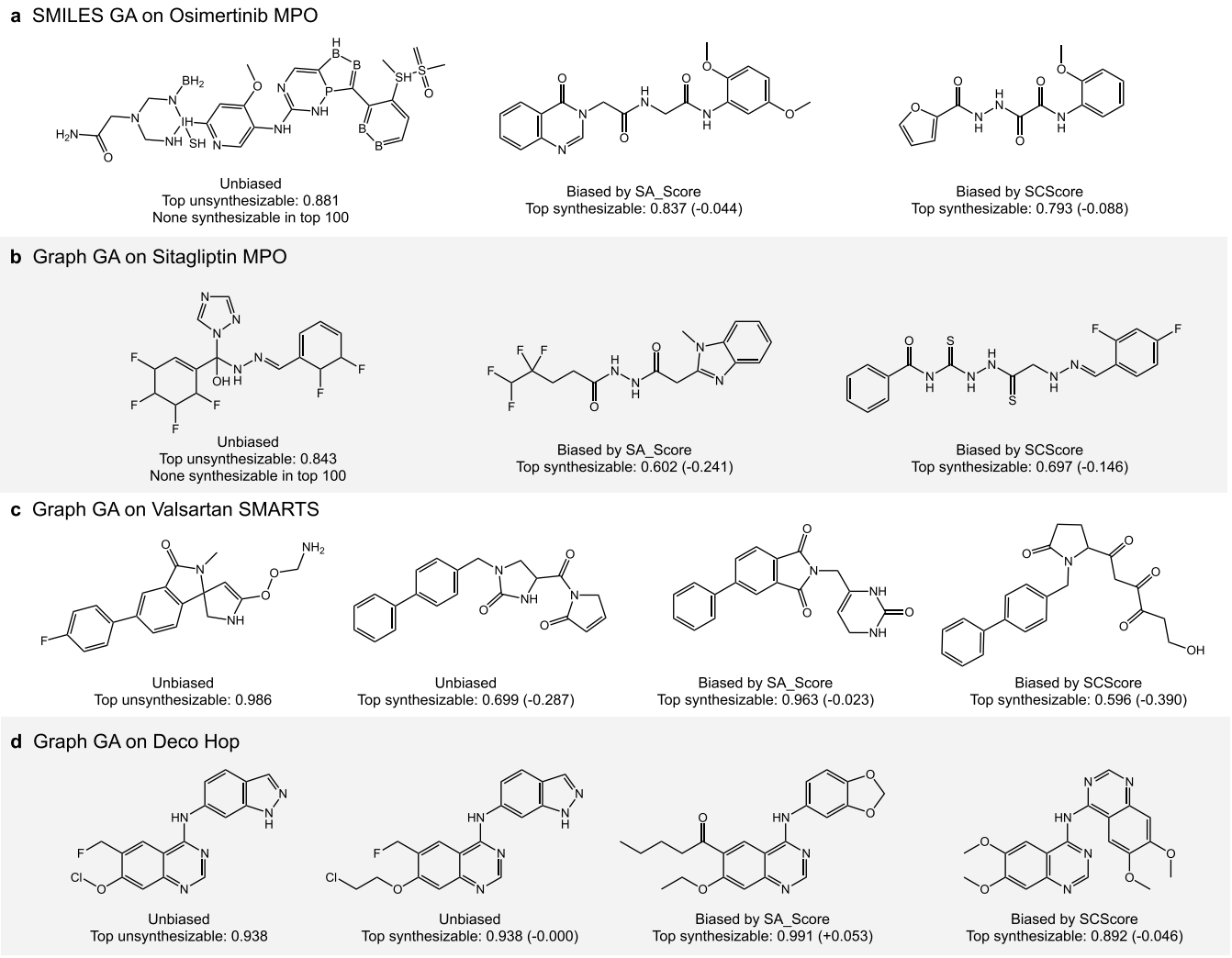


Figure 4. Examples of molecules from goal-directed optimization that were improved by heuristic biasing. Scores shown are the objective function values that have been normalized to the interval [0, 1]. (a,b) Cases where no synthesizable compounds were found in the top 100 suggestions without biasing, but at least one was found with either SA_Score or SCScore biasing. (c,d) Cases where the top structure found without biasing was perceived as unsynthesizable and the use of heuristic biasing improved the objective function value of the top *synthesizable* structure.

Table 1. Fraction of Synthesizable Compounds in the Top-100 Candidates Across All Goal-Directed Optimization Tasks and All Methods, Demonstrating Successful Heuristic Biasing

training database	task difficulty	unbiased	biased by SA_Score	biased by SCScore
ChEMBL	trivial	60.1%	91.0%	77.9%
	hard	30.2%	80.2%	55.4%
MOSES	trivial	63.5%	92.2%	78.8%
	hard	32.7%	77.2%	58.0%

by Korovina et al., who propose a method described in the next paragraph, highlight several existing methods that all require ≥5000 evaluations for a single task compared to their 100. On the basis of the machine learning community's broader interest in improving the sample efficiency of reinforcement learning algorithms[36] (thus fewer times calling the oracle) and CASP tools becoming faster, the use of an explicit retrosynthetic planner *during* optimization may become a computationally viable strategy.

The final approach (Figure 1g) is to embed synthesizability constraints in the generation algorithm itself, i.e., constrain the search space to molecules that can be produced from available building blocks. As early as 2003, Vinkers et al. described the iterative optimization of molecular structure by selecting building blocks to react with a growing molecular structure.[37] More recently, Bradshaw et al.[38] proposed a model called MoleculeChef that generates a bag of reactants and uses a forward reaction prediction software to obtain the final products. Korovina et al.'s ChemBO similarly treats molecular generation as a random walk on a directed (synthetic) graph where each node is a molecule, and the parents of this node are the reagents that produce the child molecule when combined.[35] These techniques are philosophically aligned with our use of retrosynthetic analysis to evaluate synthesizability—both try to use our collective knowledge of chemical reactivity to dictate what reactions are possible—but operate in the *forward* synthetic direction. This makes them subject to the same caveats that any CASP tool is subject to: their validity is entirely dependent on the accuracy of their forward reaction prediction engine, which can use either hand-coded rules or algorithmically inferred rules. The greater the number of synthetic steps we allow, the lower the chances that each reaction will proceed as predicted. As this is essentially how

https://dx.doi.org/10.1021/acs.jcim.0c00174
J. Chem. Inf. Model. 2020, 60, 5714−5723

virtual libraries are constructed, we would expect a similar rate of success (anecdotally, 85% successful delivery of compounds from a library enumerated with a single synthetic step). Nevertheless, as the search space is directly constrained by these rules, they may enable a more efficient exploration of chemical space. We expect such algorithms to rapidly grow in popularity as the accuracy of reaction prediction tools improves.[39,40]

CONCLUSION

In this paper, we describe an analysis of the synthesizability of *de novo* generative algorithms. We first examined common chemical compound libraries and used ASKCOS to evaluate their synthesizability. We next evaluated molecules proposed by distribution learning and goal-directed generation methods, with and without biasing by heuristic synthesizability metrics. Distribution learning methods, provided they can learn the chemical distribution of the training set well, seem to generate molecules that are synthesizable with a similar frequency to their training set. Goal-directed generation methods have a significant risk of proposing unsynthesizable structures as their top suggestions, particularly using the SMILES GA or Graph GA methods, but occasionally there may be enough high-performing, synthesizable molecules in the top 100 that *post hoc* filtering (Figure 1c) is a viable strategy. In other cases, the proposed molecules are so absurd that one immediately recognizes why benchmarking these methods solely in terms of their objective function value is insufficient (e.g., Figures S10 and S11). Biasing generation by training set synthesizability (Figure 1d) works for distribution learning but does not have a noticeable effect on goal-directed optimization tasks. For some tasks, modifying the objective function with the SA_Score leads to candidates that outperform those obtained through *post hoc* filtering (Figure 4c,d and Figure S13). This heuristic biasing (Figure 1e) almost always improves the synthesizability of generated candidates but necessarily detracts from the main objective function.

We acknowledge that the identification of a synthetic pathway by ASKCOS is not a necessary or sufficient condition for synthesizability, nor would the generation of molecular candidates through forward synthesis prediction be a guarantee that those reactions would work experimentally. CASP tools for retrosynthesis and forward synthesis are imperfect. They do not capture our entire knowledge of chemical reactivity and may occasionally produce overly optimistic suggestions (e.g., with respect to selectivity). Further, the ability of CASP programs to find pathways is sensitive to the precise database of chemicals considered buyable and the settings one chooses for the retrosynthetic expansion. Even with an imperfect CASP tool like ASKCOS, however, we can obtain a meaningful analysis of synthesizability of generated molecules.

Generative models have a tremendous potential to accelerate molecular discovery. As we improve their ability to propose synthesizable molecules—whether by improving CASP tools for *post hoc* filtering, developing new heuristics for synthesizability, efficiently sampling a CASP oracle to bias generation with reinforcement learning, or designing new algorithms explicitly constrained by predictions of chemical reactivity—their utility and relevance to practical discovery projects will only increase.

METHODS

ASKCOS. ASKCOS is an open-source software framework that integrates efforts to generalize known chemistry to new substrates by learning to apply retrosynthetic transformations, to identify suitable reaction conditions, and to evaluate whether reactions are likely to be successful when attempted experimentally.[25,41] Data-driven models within ASKCOS are trained on millions of reactions from the U.S. Patent and Trademark Office (USPTO) and Reaxys databases. The core retrosynthetic capabilities rely on the recursive application of algorithmically extracted reaction templates encoded as SMARTS patterns. Expansion is parallelized using an upper confidence bound tree search as detailed in the original publication. Importantly, ASKCOS has both programmatic and graphical interfaces to enable thousands of compounds to be processed without human intervention. The program makes extensive use of RDKit.[42]

While the program offers flexible stopping criteria, we require starting materials to be commercially available according to a 2018 database of molecules from eMolecules or Sigma-Aldrich with prices no greater than $100 per gram; the full list is available in the ASKCOS codebase. This is a very strict price limit in the context of drug discovery, so it warrants two additional comments. First, one could consider most molecules to be "commercially available," in that some supplier or contract research organization will agree to produce them at some cost given sufficient lead time. Second, it is straightforward to modify the database of molecules considered commercially available depending on each user's price tolerance and available chemical inventory.

To determine whether a molecule is "synthesizable," we run a retrosynthetic expansion using ASKCOS with the following expansion settings: the maximum search depth—longest linear sequence—is 9; the maximum branching ratio—number of unique precursors to consider at each disconnection—is 25; the maximum wall time of expansion is 60 s; the maximum cumulative probability for the target is 0.999; the maximum number of templates to apply is 1000; the maximum price for starting materials is $100/g as described above; the minimum plausibility of reactions—evaluated by a binary classifier as a "sanity check"—is 0.1. We terminate the search as soon as a pathway is found, rather than continuing to search for a more optimal (e.g., shorter, cheaper) pathway. All retrosynthetic analyses were carried out in an ASKCOS server on a debian virtual machine running on Google Cloud with eight cores, 52 GB of memory, and no other background tasks.

Compound Databases. MOSES[26] is an open database included in the MOSES benchmarking platform that evaluates distribution learning algorithms for drug discovery. The database of 1.94 million structures represents a subset of the 4.6 million in the ZINC Clean Leads collection with molar masses of 250−350 g/mol, fewer than eight rotatable bonds, and a maximum XLogP of 3.5. Polykovskiy et al. filtered out molecules containing charged atoms; atoms besides C, N, S, O, F, Cl, Br, and H; cycles longer than eight atoms; and molecules containing "structural alerts" from medicinal chemistry filters and PAINS filters.

ChEMBL[28] is a regularly updated, open access database containing a large number of biologically relevant compounds and associated assays (e.g., binding and ADMET). In our experiments, we use ChEMBL release 24, which contains 15.2 million activity measurements for 1.8 million compounds.

https://dx.doi.org/10.1021/acs.jcim.0c00174
J. Chem. Inf. Model. 2020, 60, 5714−5723

ZINC[43] is an open database of commercially available (not in-stock) compounds for virtual screening. ZINC contains over 230 million purchasable compounds in ready-to-dock, 3D formats. We sampled molecules from ZINC-250k, which is a widely used subset of ZINC12[43] from Gómez-Bombarelli et al.[44]

Sheridan et al.[16] refers to a set of 1730 unique and parseable compounds taken from the 2575 unique molecules released by Merck in their paper exploring a crowd-sourced definition of molecular complexity. These molecules were drawn from various public and Merck-internal sources as described in the original publication.

GDB17[30] is an open database containing 166.4 billion enumerated molecules with up to 17 heavy atoms of C, N, O, S, and halogens. The enumeration started from mathematical graphs to form skeletons, aiming to cover size ranges containing many drugs and typical for lead compounds. We are sampling from its "Lead-like Set" of 800,000 compounds with molar masses of 100−350 g/mol, CLogP of 1−3, and without three- or four-membered rings.

Molecular Generation Algorithms. *Random sampler* is a baseline approach to molecular generation and optimization that randomly samples molecules (with replacement) from a "training set" of known compounds.

Best from data represents the virtual screening approach to molecular optimization, where all molecules from a "training set" of known compounds are evaluated to identify the ones with the highest scores.

LSTM[45] refers to a Long−Short-Term Memory[46] neural network that is widely used in natural language processing. The model is trained in an autoregressive way to predict the next character of a simplified molecular-input line-entry (SMILES) string. It can be iteratively fine-tuned to optimize molecules toward a specific objective using a hill-climbing algorithm. We evaluated the implementation in ref 27.

VAE[44] refers to a variational autoencoder architecture that learns to construct a bidirectional mapping between SMILES represented chemical space and a finite-dimensional continuous latent space. The architecture is devised to learn a probabilistic generative model as well as its posterior, respectively known as decoder and encoder. The two parts are trained simultaneously by maximizing the evidence lower bound (ELBO) of the marginal likelihood

$$\mathrm{ELBO}(\phi, \theta) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x) \| p(z))$$

where ϕ and θ are differential parameters and KL is the Kullback−Leibler (KL) divergence. We evaluated the implementation from ref 26.

AAE[47] is another approach to train a SMILES-based encoder−decoder architecture. Instead of KL regularization, AAE is trained with an adversarial learning regularization that matches the posterior distribution to a prior distribution. We evaluated the implementation from ref 26.

SMILES GA[48] is a population-based grammar evolution algorithm. We evaluated Yoshikawa et al.'s model that adopted a "chromosome" with context-free grammar of the SMILES string so that crossover and mutation happen at the level of SMILES tokens. Each "chromosome" can be decoded to a SMILES string, and checked validity can be used. We evaluated the implementation from ref 27.

Graph GA[49] is another genetic algorithm that represents molecules as graphs, rather than relying on SMILES strings. The crossovers and mutations are performed by altering a molecular graph directly, i.e., exchanging substructures and hand-written substitution rules for mutation. We evaluated the implementation from ref 27.

Objective Functions for Optimization. The suite of objective functions we use for goal-directed optimization was taken from Brown et al.'s benchmarking function sets.[27] Evaluation is divided into "trivial" tasks and "hard" tasks following the language of the original work. The trivial tasks are named as such because almost all molecular optimization methods can perform exceedingly well on them (thus they are not suitable for the assessment of generative models), whereas the hard tasks show greater variation as a function of the method used. However, all of these objective functions are relatively simple heuristic functions of molecular structure.

The trivial objectives we use include quantitative estimate of drug-likeness (QED),[50] a central nervous system (CNS) MPO,[51] an isomer of $C_7H_8N_2O_2$, and Pioglitazone MPO. The hard objectives we use include Osimertinib MPO, Fexofenadine MPO, Ranolazine MPO, Perindopril MPO, Amlodipine MPO, Ranolazine MPO, Sitagliptin MPO, Zaleplon MPO, Valsartan SMARTS, Scaffold Hop, and Decorator Hop. Some MPO tasks try to identify molecules dissimilar to the titular molecule but with similar properties; other MPO tasks try to identify molecules similar to the titular molecule but with "improved" druglikeness properties. We did not include the benchmarks that measure the similarity to commercial drug molecules and isomer benchmarks in hard tasks because we think they are less meaningful for drug discovery purposes. We refer readers to the list of benchmarks in ref 27 for a full description of these objectives.

Biasing Techniques for Molecular Generation. *Post hoc* filtering is the approach where a CASP tool is used to filter unsynthesizable molecules suggested by an unbiased generation. We evaluate this approach by calculating the fraction of molecules that would pass the ASKCOS filter and their objective function values.

Training set biasing is the approach of starting with a molecule database that has a higher fraction of synthesizable compounds as the training set for deep learning methods or the starting pool for genetic algorithms. In this paper, we use ChEMBL (68.3% as tested) and MOSES (89.8% as tested) as representative data sets with lower and higher synthesizabilities, respectively. This approach can be used in both unoptimized generation and optimized generation.

Heuristic biasing is the approach of modifying the main objective function to penalize the generation of unsynthesizable compounds. We apply a synthesizability function multiplier, ranging from 0 to 1, to a prenormalized objective function (also ranging from 0 to 1). Specifically, we use a form of modified Gaussian and sigmoid function to rescale the heuristic score x:

$$\text{modifier} = \begin{cases} 1 & x < \mu \\ e^{-(x-\mu)^2/2\sigma} & x \geq \mu \end{cases}$$

$$\text{modifier} = 1 - \frac{1}{1 + e^{a(x-b)}}$$

We performed 30 iterations of Tree Parzen Estimator (TPE) Bayesian optimization to determine the hyper parameters for each score. The hyper parameters aimed to maximize the fraction of synthesizable suggestions times the average of the

https://dx.doi.org/10.1021/acs.jcim.0c00174
J. Chem. Inf. Model. 2020, 60, 5714−5723

objective function for the top 10 molecules from a graph genetic algorithm. We tested the biasing effect of SA_Score, SCScore, and length of SMILES string, but meaningful parameters could not be obtained for the SMILES string heuristic. The multipliers we use are shown in Figure S4. This approach can only be used in optimized generation.

SA_Score[31] is a popular heuristic score for quantifying synthesizability. It computes a score using a fragment-contribution approach, where rarer fragments (as judged by their abundance in the PubChem database) are taken as an indication of lower synthesizability.

SCScore[19] is a learned synthetic complexity score computed as a neural network model trained on reaction data from the Reaxys database. It was designed with synthesis planning in mind to operate on molecules resembling not just drug-like products but intermediates and simpler building blocks as well.

SMILES length is a very simple heuristic that associates molecules with longer SMILES strings as an indication of synthetic difficulty. The length of a SMILES string correlates closely with the number of heavy atoms in a molecule (i.e., larger molecules are harder to synthesize) but is further increased by the presence of formal charges, ring closures, and defined stereochemistry.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c00174.

Additional results (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Connor W. Coley − *Department of Chemical Engineering, MIT, Cambridge, Massachusetts 02139, United States; Broad Institute of Harvard and MIT, Cambridge, Massachusetts 02142, United States;* orcid.org/0000-0002-8271-8723; Email: ccoley@mit.edu

Author

Wenhao Gao − *Department of Chemical Engineering, MIT, Cambridge, Massachusetts 02139, United States; Department of Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, Maryland 21218, United States;* orcid.org/0000-0002-6506-8044

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c00174

Notes

The authors declare no competing financial interest.

All code and data can be found at https://github.com/wenhao-gao/askcos_synthesizability.

■ ACKNOWLEDGMENTS

This work was supported by the Machine Learning for Pharmaceutical Discovery and Synthesis consortium. We thank Mike Fortunato and Thomas Struble for assisting with programmatic interfacing of ASKCOS. We also thank Lagnajit Pattanaik and Klavs Jensen for commenting on the manuscript.

■ REFERENCES

(1) Walters, W. P.; Wang, R. New Trends in Virtual Screening. *J. Chem. Inf. Model.* **2019**, *59*, 3603−3604.

(2) Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O'Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K.; Tolmachev, A. A.; Shoichet, B. K.; Roth, B. L.; Irwin, J. J. Ultra-large library docking for discovering new chemotypes. *Nature* **2019**, *566*, 224.

(3) Virshup, A. M.; Contreras-García, J.; Wipf, P.; Yang, W.; Beratan, D. N. Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. *J. Am. Chem. Soc.* **2013**, *135*, 7296−7303.

(4) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018**, *361*, 360−365.

(5) Lameijer, E.-W.; Bäck, T.; Kok, J. N.; Ijzerman, A. P. Evolutionary algorithms in drug design. *Nat. Comput.* **2005**, *4*, 177−243.

(6) Lewis, D. W.; Willock, D. J.; Catlow, C. R. A.; Thomas, J. M.; Hutchings, G. J. De novo design of structure-directing agents for the synthesis of microporous solids. *Nature* **1996**, *382*, 604.

(7) Kingma, D. P.; Welling, M. Auto-Encoding Variational Bayes. arXiv: 1312.6114 [cs, stat], **2014**.

(8) Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc., 2014; pp 2672−2680.

(9) Elton, D. C.; Boukouvalas, Z.; Fuge, M. D.; Chung, P. W. Deep learning for molecular generation and optimization - a review of the state of the art. arXiv: 1903.04388 [physics, stat], **2019**.

(10) 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.; Xing, L.; Guo, T.; Aspuru-Guzik, A. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat. Biotechnol.* **2019**, *37*, 1038−1040.

(11) Bjerrum, E. J.; Threlfall, R. Molecular Generation with Recurrent Neural Networks (RNNs). arXiv: 1705.04612 [cs, q-bio], **2017**.

(12) Sumita, M.; Yang, X.; Ishihara, S.; Tamura, R.; Tsuda, K. Hunting for Organic Molecules with Artificial Intelligence: Molecules Optimized for Desired Excitation Energies. *ACS Cent. Sci.* **2018**, *4*, 1126−1133.

(13) Bertz, S. H. The First General Index of Molecular Complexity. *J. Am. Chem. Soc.* **1981**, *103*, 3599−3601.

(14) Ertl, P.; Schuffenhauer, A. Estimation of Synthetic Accessibility Score of Drug-Like Molecules Based on Molecular Complexity and Fragment Contributions. *J. Cheminf.* **2009**, *1*, 8.

(15) Takaoka, Y.; Endo, Y.; Yamanobe, S.; Kakinuma, H.; Okubo, T.; Shimazaki, Y.; Ota, T.; Sumiya, S.; Yoshikawa, K. Development of a method for evaluating drug-likeness and ease of synthesis using a data set in which compounds are assigned scores based on chemists' intuition. *J. Chem. Inf. Model* **2003**, *43*, 1269−1275.

(16) Sheridan, R. P.; Zorn, N.; Sherer, E. C.; Campeau, L.-C.; Chang, C. Z.; Cumming, J.; Maddess, M. L.; Nantermet, P. G.; Sinz, C. J.; O'Shea, P. D. Modeling a Crowdsourced Definition of Molecular Complexity. *J. Chem. Inf. Model.* **2014**, *54*, 1604−1616.

(17) Baba, Y.; Isomura, T.; Kashima, H. Wisdom of crowds for synthetic accessibility evaluation. *J. Mol. Graphics Modell.* **2018**, *80*, 217−223.

(18) Li, J.; Eastgate, M. D. Current Complexity: a Tool for Assessing the Complexity of Organic Molecules. *Org. Biomol. Chem.* **2015**, *13*, 7164−7176.

(19) Coley, C. W.; Rogers, L.; Green, W. H.; Jensen, K. F. SCScore: Synthetic complexity learned from a reaction corpus. *J. Chem. Inf. Model.* **2018**, *58*, 252−261.

(20) Lajiness, M. S.; Maggiora, G. M.; Shanmugasundaram, V. Assessment of the consistency of medicinal chemists in reviewing sets of compounds. *J. Med. Chem.* **2004**, *47*, 4891−4896.

(21) Feng, F.; Lai, L.; Pei, J. Computational chemical synthesis analysis and pathway design. *Front. Chem.* **2018**, *6*, 199.

(22) Podolyan, Y.; Walters, M. A.; Karypis, G. Assessing synthetic accessibility of chemical compounds using machine learning methods. *J. Chem. Inf. Model.* **2010**, *50*, 979−991.
(23) Huang, Q.; Li, L.-L.; Yang, S.-Y. RASA: a rapid retrosynthesis-based scoring method for the assessment of synthetic accessibility of drug-like molecules. *J. Chem. Inf. Model.* **2011**, *51*, 2768−2777.
(24) Bonnet, P. Is chemical synthetic accessibility computationally predictable for drug and lead-like molecules? A comparative assessment between medicinal and computational chemists. *Eur. J. Med. Chem.* **2012**, *54*, 679−689.
(25) Coley, C. W.; Thomas, D. A.; Lummiss, J. A.; Jaworski, J. N.; Breen, C. P.; Schultz, V.; Hart, T.; Fishman, J. S.; Rogers, L.; Gao, H.; et al. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science* **2019**, *365*, eaax1566.
(26) Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; Kadurin, A.; Nikolenko, S.; Aspuru-Guzik, A.; Zhavoronkov, A. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. arXiv: 1811.12823 [cs, stat], **2018**.
(27) Brown, N.; Fiscato, M.; Segler, M. H. S.; Vaucher, A. C. GuacaMol: Benchmarking Models for De Novo Molecular Design. arXiv: 1811.09621 [physics, q-bio], **2018**.
(28) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40*, D1100−D1107.
(29) Sterling, T.; Irwin, J. J. ZINC 15 − Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324−2337.
(30) Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *J. Chem. Inf. Model.* **2012**, *52*, 2864−2875.
(31) Ertl, P.; Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminf.* **2009**, *1*, 8.
(32) Bruns, R. F.; Watson, I. A. Rules for Identifying Potentially Reactive or Promiscuous Compounds. *J. Med. Chem.* **2012**, *55*, 9763−9772.
(33) Baell, J. B.; Nissink, J. W. M. Seven Year Itch: Pan-Assay Interference Compounds (PAINS) in 2017—Utility and Limitations. *ACS Chem. Biol.* **2018**, *13*, 36−44.
(34) Zhou, Z.; Kearnes, S.; Li, L.; Zare, R. N.; Riley, P. Optimization of molecules via deep reinforcement learning. *Sci. Rep.* **2019**, *9*, 1−10.
(35) Korovina, K.; Xu, S.; Kandasamy, K.; Neiswanger, W.; Poczos, B.; Schneider, J.; Xing, E. P. ChemBO: Bayesian Optimization of Small Organic Molecules with Synthesizable Recommendations. arXiv preprint arXiv: 1908.01425, **2019**.
(36) Ortega, P, et al. Meta-learning of sequential strategies. arXiv preprint arXiv: 1905.03030, **2019**.
(37) Vinkers, H. M.; de Jonge, M. R.; Daeyaert, F. F. D.; Heeres, J.; Koymans, L. M. H.; van Lenthe, J. H.; Lewi, P. J.; Timmerman, H.; Van Aken, K.; Janssen, P. A. J. SYNOPSIS: SYNthesize and OPtimize System in Silico. *J. Med. Chem.* **2003**, *46*, 2765−2773.
(38) Bradshaw, J.; Paige, B.; Kusner, M. J.; Segler, M. H.; Hernández-Lobato, J. M. A Model to Search for Synthesizable Molecules. arXiv preprint arXiv: 1906.05221, **2019**.
(39) Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.* **2019**, *10*, 370−377.
(40) Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Bekas, C.; Lee, A. A. *Molecular Transformer for Chemical Reaction Prediction and Uncertainty Estimation*; 2018.
(41) ASKCOS Software Repository. https://github.com/connorcoley/ASKCOS.
(42) Landrum, G. *RDKit: Open-Source Cheminformatics Software*; T5 Informatics GmbH, 2016.
(43) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: a free tool to discover chemistry for biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757−1768.
(44) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **2018**, *4*, 268−276.
(45) Segler, M. H.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **2018**, *4*, 120−131.
(46) Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Computation* **1997**, *9*, 1735−1780.
(47) Polykovskiy, D.; Zhebrak, A.; Vetrov, D.; Ivanenkov, Y.; Aladinskiy, V.; Mamoshina, P.; Bozdaganyan, M.; Aliper, A.; Zhavoronkov, A.; Kadurin, A. Entangled conditional adversarial autoencoder for de novo drug discovery. *Mol. Pharmaceutics* **2018**, *15*, 4398−4405.
(48) Yoshikawa, N.; Terayama, K.; Sumita, M.; Homma, T.; Oono, K.; Tsuda, K. Population-based de novo molecule generation, using grammatical evolution. *Chem. Lett.* **2018**, *47*, 1431−1434.
(49) Jensen, J. H. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chem. Sci.* **2019**, *10*, 3567−3572.
(50) Bickerton, G. R.; Paolini, G. V.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **2012**, *4*, 90.
(51) Wager, T. T.; Hou, X.; Verhoest, P. R.; Villalobos, A. Central nervous system multiparameter optimization desirability: application in drug discovery. *ACS Chem. Neurosci.* **2016**, *7*, 767−775.

https://dx.doi.org/10.1021/acs.jcim.0c00174
J. Chem. Inf. Model. 2020, 60, 5714−5723