ChemBO: Bayesian Optimization of Small Organic Molecules with Synthesizable Recommendations

Ksenia Korovina
Carnegie Mellon University
kkorovin@cs.cmu.edu

Sailun Xu
Carnegie Mellon University
sailunx@andrew.cmu.edu

Kirthevasan Kandasamy
Carnegie Mellon University
kandasamy@cs.cmu.edu

Willie Neiswanger
Carnegie Mellon University
willie@cs.cmu.edu

Barnabás Póczos
Carnegie Mellon University
bapoczos@cs.cmu.edu

Jeff Schneider
Carnegie Mellon University
schneide@cs.cmu.edu

Eric P. Xing
Carnegie Mellon University
epxing@cs.cmu.edu

1 Introduction

In many applications, such as drug discovery and materials optimization, one is interested in designing chemical molecules with desirable properties [1]. For instance, in drug discovery, one wishes to find molecules with high solubility in blood and high potency, but low toxicity. Recently, we have seen a surge of interest in the adoption of machine learning techniques for such tasks, due to their effectiveness in modeling structure-property relations of molecules, as well as due to limitations of traditional methods in computational chemistry methods in effectively exploring the large and complex chemical space. While there have been several strategies for this problem, such as generative modeling, reinforcement learning, and more [2–6], one promising approach is to treat this task as a black-box optimization problem (e.g. [7, 8]). Here, the goal is to find the optimum of an arbitrary function $f(x)$ that measures the goodness of a molecule x for the relevant application, to which we have access through evaluations only. In real world settings, f is typically derived from the results of laboratory experiments. Since conducting such experiments is expensive, it is imperative to find the maximum in as few evaluations as possible.

In this work, we contribute to this line of research by developing ChemBO, a Bayesian optimization (BO) framework for generating and optimizing molecules, focusing on small[1] organic molecules for drug discovery. In doing so, we wish to emulate a real world setting, where an algorithm would recommend new candidate molecules. These molecules *first need to be synthesized*, and then tested for necessary properties. Ideally, the algorithm would not only ensure that the recommended molecule is chemically valid and synthesizable, but also provide a recipe for synthesis and take into consideration the reagents and resources available. Even in cases where the recommended molecules are synthesized manually, providing a recipe can be a helpful guide to the chemist and greatly reduce the amount of manual work required. Combining sequential decision making and synthesis, ChemBO is a first step towards automated molecular optimization. To summarize, our contributions are:

1. We use a Gaussian process (GP) to model structure-property relations in molecules. For the GP kernel, we use prior work on molecular fingerprints [9, 10] and additionally design a new optimal transport based similarity measure between molecules by treating them as graphs.

[1] In contrast with biologics (large molecules), which are protein based.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

2. We use a synthesis graph to navigate the chemical space. On each iteration of BO, ChemBO recommends the molecule on this synthesis graph that is deemed to be the most promising by the GP, i.e. the molecule with the highest *acquisition* value [11]. This approach not only ensures that each recommended molecule is chemically valid, but also provides a synthesis recipe.
3. In our experiments, we demonstrate that ChemBO outperforms simpler alternatives for synthesizeable optimization which do not use a probabilistic model to guide search. The final values for the popular QED [12] and penalized partition coefficient [8] benchmarks achieved by ChemBO are competitive with state-of-the-art methods, while using significantly less data and function evaluations. Our code is released open source at `https://github.com/ks-korovina/chembo`.

2 Method

Gaussian Processes and Bayesian Optimization. In this work, we build a Gaussian process based Bayesian optimization model as in [13]. To design a GP based BO solution for molecular optimization, the two central decisions that we need to make are *choosing a GP kernel* to specify a GP model, and *designing a method to optimize acqusition*. As mentioned previously, when doing so, we will strive to ensure that the recommendations are synthesizable and provide a synthesis recipe.

An Optimal Transport Based Kernel. A natural option would be to simply use one of the existing molecular kernels. Indeed, molecular fingerprint based kernels are known to work well for several applications, and we use that of Ralaivola et al. [9] in ChemBO. Moreover, we develop a new dissimilarity measure $d : \mathcal{X}^2 \to \mathbb{R}_+$ to capture more graphical information from molecules. Given such a measure, $\kappa = e^{-\beta d}$ where $\beta > 0$, is a similarity measure which can be used as a kernel. The graphical structure of a molecule determines many of its chemical properties, and as such, our measure will view molecules as graphs $M = (A, B)$, A is a set of atoms (vertices) with labels $\ell_{\mathsf{a}}(a)$, B is the set of bonds (edges) with labels $\ell_{\mathsf{b}}(b)$. We will define this dissimilarity measure via a matching scheme which attempts to match the assigned weights $w_{\mathsf{a}}(a)$ of atoms in one molecule to another. The matching will only permit matching identical atoms, i.e. carbon atoms can only be matched to carbon atoms, but we will incur penalties for matching atoms with different bond types.

Given two molecules $M_1 = (A_1, B_1), M_2 = (A_2, B_2)$ with n_1, n_2 atoms respectively, let $U \in \mathbb{R}_+^{n_1 \times n_2}$ denote the matching matrix, i.e. $U(i, j)$ is the weight matched between $i \in M_1$ and $j \in M_2$. The dissimilarity measure is the solution of the following program.

$$\begin{aligned} \underset{U}{\text{minimise}} \quad & \varphi_{\text{at}}(U) + \varphi_{\text{st}}(U) + \varphi_{\text{nm}}(U) \\ \text{subject to} \quad & \sum_{j \in A_2} U(i,j) \leq w_{\mathsf{a}}(i), \ \sum_{i \in A_1} U(i,j) \leq w_{\mathsf{a}}(j), \ \forall i, j \end{aligned} \tag{1}$$

Here, the first term is the atom type penalty φ_{at} which only permits matching similar atoms, i.e. C atoms can only be matched to other C atoms and not H or O atoms. Accordingly, it is defined as $\varphi_{\text{at}}(U) = \langle C_{\text{at}}, U \rangle = \sum_{i \in A_1} \sum_{j \in A_2} C_{\text{at}}(i,j) U(i,j)$, where $C_{\text{at}}(i,j) = 0$ if $\ell_{\mathsf{a}}(i) = \ell_{\mathsf{a}}(j)$ and ∞ otherwise. The second term is the bond type penalty term, which, similar to φ_{at}, is given by $\varphi_{\text{st}}(U) = \langle C_{\text{st}}, U \rangle$, where $C_{\text{st}}(i,j)$ is the penalty for matching unit weight from atom $i \in A_1$ to atom $j \in A_2$. We let $C_{\text{st}}(i,j)$ to be the fraction of dissimilar bonds in the union of all bonds. If the atom type and bond type penalties are too large or infinite, we can choose to not match the atoms from one molecule to another. However, we will incur a penalty via the non-matching penalty term φ_{nm}. We set this term to be the sum of weights unassigned in both graphs, i.e. $\varphi_{\text{nm}}(U) = \sum_{i \in A_1} (w_{\mathsf{a}}(i) - \sum_{j \in A_2} U(i,j)) + \sum_{j \in A_1} (w_{\mathsf{a}}(j) - \sum_{i \in A_1} U(i,j))$. For two given molecules M_1, M_2, we will denote the resulting dissimilarity measure, i.e. the solution of (1), by d.

Exploring the Space of Synthesizable Molecules and Optimizing the Acquisition. Our proposal to optimize the acquisition randomly explores the space of synthesizable molecules and picks the one with the highest acquisition – this can be viewed as performing a random walk on a *synthesis graph*[2]. For this, consider a setting in a laboratory or an automated experimentation apparatus, where we have access to a limited library of reagents $\mathcal{S}$ and process conditions $\mathcal{Q}$. We will assume that we have access to an oracle SYNTHESIZE which can take as input a set of compounds and process

[2] A synthesis graph is a directed graph where each node is a molecule, and the parents of this node are the reagents, which when combined, produce the child molecule.

Algorithm 1 Random Walk Explorer

1: **Input:** n, $\mathcal{S}$, $\mathcal{P}$, $\mathcal{D}$ ▷ # steps n, Initial molecules $\mathcal{S}$ and conditions $\mathcal{P}$, Past evaluations D
2: $k = 0$
3: **while** $k \leq n$ **do**
4: $S \leftarrow$ RAND-SELECT$(\mathcal{S})$ ▷ Select a subset of molecules as reaction inputs
5: $Q \leftarrow$ RAND-SELECT$(\mathcal{Q})$ ▷ Select a subset of process conditions
6: $M \leftarrow$ SYNTHESIZE(S, Q) ▷ Predict reaction product
7: **if** $M \neq$ NULL **And** $M \backslash D \neq \varnothing$ **then** ▷ $M \backslash D$ is set difference.
8: $k \leftarrow k + 1$
9: $\mathcal{S} \leftarrow \mathcal{S} \cup M \backslash D$ ▷ Add outcomes to the pool
return $\arg\max_{m \in \mathcal{S}} \varphi(m)$

conditions and tell us the set of molecules M produced if these compounds are reacted in the given conditions. In the event, a reaction cannot be effected, it will output NULL. Our procedure, described in Algorithm 1, operates as follows. As input, it takes $\mathcal{S}$, $\mathcal{P}$, the number of evaluations n and a set D of evaluations where we have already conducted experiments. First it randomly samples a few molecules S and a few process conditions Q from $\mathcal{S}$ and $\mathcal{Q}$ respectively. It passes them to SYNTHESIZE to generate a set of outputs M. If the synthesis was successful, i.e. if we could generate new molecules that were not evaluated before, they are added to the pool $\mathcal{S}$. It repeats this for n successful steps. At the end, we return the maximizer $\operatorname{argmax}_{m \in \mathcal{S}} \varphi(m)$ of the acquisition φ.

The above procedure relies crucially on the SYNTHESIZE oracle, which can perfectly predict the outcomes of reactions. Alas, no such oracle exists[3]. Fortunately however, there have been several advances in computational chemistry to predict outcomes of chemical reactions, which can be used in place of the oracle. In our work we use Rexgen [14], a recent method based on graph neural networks. Developing synthesis predictors is an active area of research [15, 16], and as such methods become more reliable, so will the efficacy of our framework. Moreover, an inaccurate or inefficient recipe can still be a useful guide to a chemist (who might choose to modify it), and in most cases is better than expecting the chemist to develop a recipe of her own from scratch.

3 Experimental evaluation

Optimization Objectives: We evaluate our methods on two of the most common molecular property functions found in the literature: the QED score (Quantitative Estimate of Drug likeliness) [12], and Pen-logP (penalized octanol-water partition coefficient). Although these metrics may not be most relevant in the drug discovery problem, they provide good benchmarks to compare different optimization methods.

Methods: We compare two instantiations of ChemBO: one using a molecular fingerprint kernel (fingerprint) and the other using the OT-based dissimilarity metric (ot-dist). The fingerprint based kernel computes Tanimoto similarity between topological (path-based) fingerprints of given molecules [17]. In addition, we also also compare to the random walk explorer (rand) in Algorithm 1, which operates exactly as described except returns the maximum of the function f in step 9 (instead of the acquisition) This can be viewed as a simple random search baseline over the synthesis graph.

Experimental Set Up: As stated previously, we wish to emulate a setting where a chemist has to work with the reagents and process conditions available to her. We choose 20 randomly chosen molecules from the openly available ChEMBL database as our initial set of reagents. The maximum QED score of the initial pool was 0.858 (when QED > 0.9, it is typically considered high). As the process conditions for the random explorer, we use all the process conditions available in Rexgen. We bootstrap all three methods listed above by evaluating the metric (QED or Pen-logP) on this initial set, and then execute the methods for 80 iterations, totaling 100 evaluations of f.

Results & Discussion: In Figure 1, we plot the number of iterations against the optimal found value by each method over 80 function evaluations for both QED and Pen-logP. The ChemBO methods, fingerprint and ot-dist, both outperform the naive random walk strategy on both tasks, validating the use of Bayesian strategies for this task. ot-dist does better than fingerprint on the QED score while vice versa on Pen-logP, indicating that the choice of the kernel can be important for the application.

[3] If it did, the entire field of organic chemistry would be just a massive graph search problem.

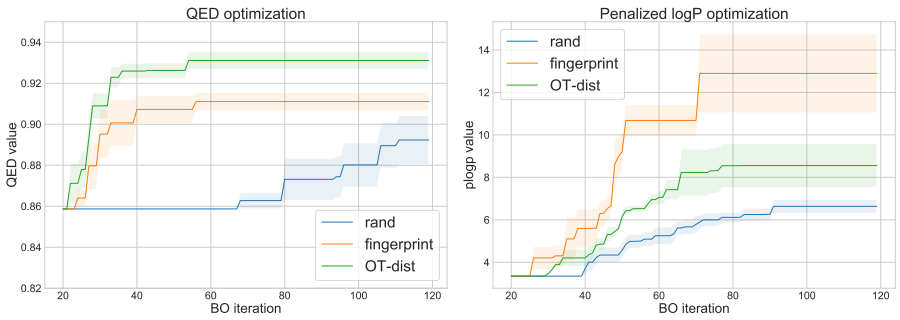

Figure 1: Results comparing the three methods described in the beginning of Section 3. We plot the number of iterations (after initialization) against the highest found QED (left) and Pen-LogP (right) values (higher is better for both). All curves are averages over 5 independent runs, shaded regions indicate one standard error.

Reliability of synthesis paths: A thorough validation of the synthesis paths proposed by ChemBO would require performing actual synthesis in lab conditions. However, we can compute the minimum synthetic accessibility score [18] over the synthesis graph as a proxy for plausibility of the entire synthesis recipe (higher is better). For ChemBO, this score mean/std is 2.498 ± 0.442 (and the average over synthesis path is 3.772 ± 1.455), while the ChEMBL dataset statistics are 2.733 ± 0.650. This means that on average the synthesis operations on ChemBO pathss are easier than in ChEMBL generally, and worst operations are not significantly worse than average.

Novel Molecules: During the execution of ChemBO, we compute the fraction of molecules that do not appear in the entire ChEMBL dataset. For ot-dist optimizing QED, on average 95.64% molecules are novel, for fingerprint 96.84%; and for Pen-logP 78% and 87.67%, respectively. This indicates that ChemBO is able to explore the chemical space well, despite the constraints on synthesizability.

Comparison with existing work: In Table 1, we compare ChemBO to state-of-the-art methods adopting reinforcement learning or generative modeling techniques [3, 5, 19, 20]. We use the same evaluation strategy as in these works, reporting top scores across several runs. It is interesting to compare the number of QED/Pen-LogP evaluations required by some of these methods. It should be emphasised that the above methods are not designed to keep the number of QED/Pen-logP evaluations to a minimum, and in fact, are tools developed for very different settings. Yet, it speaks to the efficiency of ChemBO, that we were able to obtain better or comparable values than the above work in significantly fewer evaluations, given more stringent conditions on synthesizability.

	ORGAN [19]	JT-VAE [3]	GCPN [5]	MolDQN [20]	ChemBO
QED	0.896	0.925	0.948	0.948	0.941
Pen-logP	3.63	5.30	7.98	11.84	18.39
# evaluations	$\geq 5K$	$275K$	$\geq 25K$	$\geq 25K$	100

Table 1: The best QED and Pen-LogP scores reported from prior work. For ChemBO, we use the best value obtained across the 5 trials for both fingerprint and ot-dist.

Virtual screening baseline: We translate the virtual screening experiment (keeping the number of evaluations fixed) into a computational simulation as follows. We start with the pool, and then sample compounds *outside* of that pool from the rest of the dataset. This corresponds to a situation where the experimenter purchases the compounds randomly in addition to the ones that he/she has; in theory, this could lead to a higher optimum due to a larger search space. Simulating such experiment, we got 0.922 ± 0.0128 for QED and 5.34 ± 0.973 for Pen-logP over 10 iterations - even using more data, these values are worse than the numbers in Figure 1.

4 Conclusion

We have proposed ChemBO, a Bayesian optimization algorithm for design synthesizable recommendations, is a first step towards the ambitious goal of automating the process of molecular optimization. Our experiments indicate that model based Bayesian methods outperform naive alternatives for this problem. In addition, on two benchmark objectives, we are able to get better or competitive

scores than existing work using significantly less evaluations of the objective. While our approach is invariably constrained by current synthesis predictors, it can still be a useful guide to a practitioner.

Improving the reliability of synthesis predictors and developing smarter methods to explore the chemical space are interesting avenues for future research, which will improve the efficacy of our framework. For instance, Bradshaw et al. [21] developed concurrently with us a method for learning representations of synthesizable subsets of molecules, and this could be combined with our method for smarter search. Another direction is to use ChemBO to optimize for the ability to bind with a given target. Separately, one could view the optimization budget not in terms of the number of compounds tested, but rather in terms of the number of synthesis steps – it might be that synthesis is the bottleneck, not testing the compound. This also brings up some new interesting questions for Bayesian optimization. Finally, it would be interesting to extend and test our framework to biologics and other molecular optimization problem in drug discovery and materials science.

References

[1] Daniel C. Elton, Zois Boukouvalas, Mark D. Fuge, and Peter W. Chung. Deep learning for molecular generation and optimization - a review of the state of the art, 2019.

[2] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 2018. doi: 10.1021/acscentsci.7b00572.

[3] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation, 2018.

[4] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de-novo drug design. *Science Advances, 2018, vol. 4, no. 7, eaap7885*, 2017. doi: 10.1126/sciadv.aap7885.

[5] Jiaxuan You, Bowen Liu, Rex Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation, 2018.

[6] Anton O Oliynyk, Erin Antono, Taylor D Sparks, Leila Ghadbeigi, Michael W Gaultois, Bryce Meredig, and Arthur Mar. High-throughput machine-learning-driven synthesis of full-heusler compounds. *Chemistry of Materials*, 28(20):7324–7331, 2016.

[7] Julia Ling, Maxwell Hutchinson, Erin Antono, Sean Paradiso, and Bryce Meredig. High-dimensional materials and process optimization using data-driven experimental design with well-calibrated uncertainty estimates. *Integrating Materials and Manufacturing Innovation*, 6(3):207–217, 2017.

[8] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design, 2017.

[9] Liva Ralaivola, Sanjay J Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural networks*, 18(8):1093–1110, 2005.

[10] Georg Hinselmann, Nikolas Fechner, Andreas Jahn, Matthias Eckert, and Andreas Zell. Graph kernels for chemical compounds using topological and three-dimensional local atom pair environments. *Neurocomputing*, 74(1-3):219–229, 2010.

[11] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[12] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90, 2012.

[13] Kirthevasan Kandasamy, Karun Raju Vysyaraju, Willie Neiswanger, Biswajit Paria, Christopher R. Collins, Jeff Schneider, Barnabas Poczos, and Eric P. Xing. Tuning hyperparameters without grad students: Scalable and robust bayesian optimisation with dragonfly, 2019.

[14] Connor W. Coley, Wengong Jin, Luke Rogers, Timothy F. Jamison, Tommi S. Jaakkola, William H. Green, Regina Barzilay, and Klavs F. Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.*, 2019.

[15] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Costas Bekas, and Alpha A Lee. Molecular transformer for chemical reaction prediction and uncertainty estimation. *arXiv preprint arXiv:1811.02633*, 2018.

[16] Philippe Schwaller, Theophile Gaudin, David Lanyi, Costas Bekas, and Teodoro Laino. "found in translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chemical science*, 9(28):6091–6098, 2018.

[17] Greg Landrum et al. Rdkit: Open-source cheminformatics, 2006.

[18] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8, 2009.

[19] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models, 2017.

[20] Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *arXiv preprint arXiv:1810.08678*, 2018.

[21] John Bradshaw, Brooks Paige, Matt J Kusner, Marwin HS Segler, and José Miguel Hernández-Lobato. A model to search for synthesizable molecules. *arXiv preprint arXiv:1906.05221*, 2019.