REINVENT 2.0 – an AI tool for de novo drug design

Thomas Blaschke[§], *Josep Arús-Pous*[§⊥], *Hongming Chen*[¥], *Christian Margreitter*[§], *Christian Tyrchan*[‖], *Ola Engkvist*[§], *Kostas Papadopoulos*[§], *Atanas Patronov*[§*].

§ Hit Discovery, Discovery Sciences, R&D, AstraZeneca Gothenburg, Pepparedsleden 1, 43183, Mölndal, Sweden

‖ Medicinal Chemistry, Early RIA, Biopharmaceuticals R&D, AstraZeneca, Gothenburg, Pepparedsleden 1, 43183, Mölndal, Sweden

⊥ Department of Chemistry and Biochemistry, University of Bern, Freiestrasse 3, 3012 Bern, Switzerland.

¥ Chemistry and Chemical Biology Centre, Guangzhou Regenerative Medicine and Health-Guangdong Laboratory, Science Park, 510530, Guangzhou, China

Corresponding author: atanas.patronov@astrazeneca.com

Abstract

In the past few years, we have witnessed a renaissance of the field of molecular de novo drug design. The advancements in deep learning and artificial intelligence (AI) have triggered an avalanche of ideas how to translate such techniques to a variety of domains including the field of drug design. A range of architectures have been devised to find the optimal way of generating chemical compounds by using either graph or string (SMILES) based representations. With this application note we aim to offer the community a production-ready tool for de novo design, called REINVENT. It can be effectively applied on drug discovery projects that are striving to resolve either exploration or exploitation problems while navigating the chemical space. It can facilitate the idea generation process by bringing to the researcher's attention the most promising compounds. REINVENT's code is publicly available at https://github.com/MolecularAI/Reinvent

Introduction

The main goal of de novo drug design is to identify novel active compounds that can simultaneously satisfy a constellation of essential optimization goals such as activity, selectivity, physico-chemical and ADMET properties. Because of the sheer number of possible solutions, it is a non-trivial task to optimally satisfy such a multitude of requirements which makes the search process slow and costly even when it is only conducted *in silico*. Therefore, having an efficient solution which enables the navigation of chemical space and generation of relevant ideas is essential. To address such needs the research community has recently turned its focus towards artificial intelligence (AI) based generative models that are capable of proposing promising small molecules. The potential of generative models for chemical space exploration has been demonstrated in numerous studies [1–13]. Various neural network architectures have been engineered and a plethora of AI training strategies have been employed in the race to device more efficient methods for the generation of compounds. A number of architectures, such as Variational Autoencoders (VAEs) [7,14], Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) cells [15], Conditional RNNs or Generative Adversarial Networks have been proven successful in generating molecules by using data representation of molecules either as molecular graphs or SMILES [8,16–18].

Most tools for de novo drug design, regardless of the specifics of their implementation, can be generalized to three main components: search space (SS), search algorithm and a search objective[19]. In this context we can refer to the generative models as the search space. We also observe two main trends of using generative models for de novo design: distribution-learning and goal-directed generation. Distribution-learning efforts are mostly focused on generating ideas that resemble a particular set of molecules. Goal-directed generation methods are typically using search algorithms while aiming to suggest molecules that satisfy the given objective (or objectives) without having to sample the entire search space. In both cases results are ultimately filtered by user defined scoring function (search objective) either during the generation in the goal-driven case or after sampling the entire set of solutions in the distribution-learning scenario.

While a common issue of using goal-directed approach is the narrow set of solutions the opposite approach of using distribution learning leads to screening through a vast variety of irrelevant suggestions. These two extreme scenarios represent the attempt to achieve either exploration or exploitation of the search space.

There is an increasing variety of open-source solutions based on generative models aiming to address these two aspects of de novo design separately [2,9,20,21]. Ideally, users should be allowed to navigate the chemical space efficiently in both exploration and exploitation mode while using the same de novo design tool. For exploitation, users define an area of interest and focus on generating compounds that share similar structural features. In contrast, the exploration mode enables them to obtain compounds that share less structural similarity but still satisfy other desired features. To achieve this in a fashion different from plain distribution-learning approach we need the goal-directed learning to store in memory and adapt to the suggested solutions that have been produced in the course of a single search run. This implies the necessity to utilize not only predictive models and structure similarity/dissimilarity but also various rule-based scoring components to push towards or pull away from specific areas of the chemical space. Moreover, to be able to adapt appropriately to any given drug discovery project at hand, the ability to fine-tune each of these potential scoring function components is paramount.

In this application note we are describing REINVENT 2.0 which is a tool for de novo design of small molecules. REINVENT 2.0 draws inspiration from the works of Olivecrona et al. and Sutton et al. for the use of Reinforcement Learning (RL), Arus-Pous et al. for the architecture and the implementation of the generative model, Cummins et al. for the scoring function formulation, and Blaschke et al. for the use of Diversity Filters (DF) in the RL loop to enforce exploration [22–26]. As a de novo design application REINVENT 2.0 covers both distribution-learning and goal-directed scenarios. The goal-directed use case uses a generative model as a SS, RL as a search algorithm and flexible scoring function that can combine the scores from different components to form a reward as a score objective. The calculations in the individual components can be run in parallel. Scores can be also modulated by a diversity filter which penalizes redundancy and rewards diversity in the found solutions thus stimulating exploration. Comprehensive logging is implemented for each use case. Additionally, the option to send logs to a remote REST endpoint is also available which allows to put the application behind a web interface. REINVENT 2.0 can be also used to build generative models from scratch. For details on the generative model, please refer to Arus-Pous et al. [23]. To facilitate the users we supplement the code with pre-built generative model and a range of examples that aim to illustrate some of the most common use-cases. These examples are provided in a separate repository: https://github.com/MolecularAI/ReinventCommunity .

More details on these features are provided in the sections below.

Application Overview

In its core, REINVENT is using a generative model. The generative model has an architecture derived from the work of Arus-Pous et al [23] which in turn is inspired by Segler et al [6]and Olivecrona et al [22]. The model is trained on a dataset derived from ChEMBL [27] and capable of generating compounds in the SMILES format. The architecture of the model is illustrated with **figure *S1***. REINVENT provides different running modes listed in **table *S1***. Different combinations of the running modes allow the users to achieve either exploitation or exploration of the chemical space, see **table *S2***. Further discussion on the general use cases can be found in the supporting materials. One of the key features that allow achieving an exploratory behavior are the Diversity Filters.

Diversity Filters

DF can be regarded as a collection of buckets that are used for keeping track of all generated scaffolds and the compounds that share those scaffolds. A bucket is a collection of compounds that share the same scaffold. Obviously, not all generated compounds are of interest and only those that are ranked by the multi-parameter objective (MPO) score above a certain user-defined threshold will enter the scaffold buckets. When the average score settles above this threshold we have reached **state of productivity**. This means that the majority of the compounds from each step (the ones that score above threshold) will be collected and stored in the memory. Once a compound with a score above the threshold has been generated, its scaffold is extracted and stored in a scaffold registry and the compound enters the corresponding bucket. The buckets have limited capacity and once the limit of compounds in a given bucket has reached the allowed threshold, any subsequent bucket affiliation will be penalized. Every new compound that enters a full bucket will be assigned a score of zero thus informing the agent that this area of chemical space has become unfavorable. It is important to note that compounds will be added to the bucket even if the bucket limit has been exceeded. The only impact will be on the agent, since it will be constantly discouraged from producing similar compounds that share a given scaffold. This will enforce the agent to seek alternative solutions thus achieving in effect chemical space exploration and will prevent the agent from becoming stuck in local minima and generating the same compounds repeatedly. All collected compounds are kept and stored until the end of the RL run and become available as a csv formatted file.

Users can select their diversity strategy by using Topological DF [28], Identical Murcko DF or a Scaffold Similarity DF [29]. The Topological DF is the most restrictive since it is agnostic of the atom types. It is created by removing all side chains and subsequently converting all atoms in the structure to sp3 carbons. The other two DF also remove all side chains but retain the atom types. Identical Murcko DF only checks if there is a bucket with exactly the same scaffold while Scaffold Similarity is more permissive and can include compounds into the bucket if they satisfy a certain threshold of scaffold similarity. The threshold is user defined and is sensitive to the discrete definition of the scoring function. Setting it to higher values would clearly result in less compounds passing the threshold.

Reinforcement Learning

It is often necessary to direct the generative model towards relevant areas in the chemical space that contain compounds of interest. We achieve this by subjecting it to a RL [25] scenario while aiming to satisfy a set of user-defined requirements that reflect the most important features of the desired compounds. In other words, the generative model will try to maximize the outcome of a scoring function that contains multiple components/parameters, thus computing an MPO score [30]. To generate compounds from a specific part of the chemical space, REINVENT employs a composite scoring function consisting of different user-defined components. Each component is responsible for a simple target property. The feedback from the scoring function is used in a RL loop with a policy iteration as described by Olivecrona et al. [22].

The components of the RL loop used in REINVENT are shown in ***figure S2***. Commonly the RL setup consist of an actor and an environment in which the actor takes a set of actions and receives a reward. The reward reflects how well the actor solved the problem at hand. The set of actions is referred to as policy, and the reward after completing the policy is known as a policy iteration. In our case, the actions are the individual steps necessary for building sequences of tokens which translate into SMILES. The role of the environment is played by the score modulating block in ***figure S2*** and the actor is denoted as an “agent”. After the agent samples a batch of smiles the reward is influenced by several components: scoring function, “prior” and a diversity filter.

The “prior” is a generative model which shares identical architecture and vocabulary with the agent. It possesses a great generative capacity and the potential to sample compounds from a comparably vast area of the chemical space. Essentially, the prior is the same as the agent at the beginning of the RL. There is, however, a use case where the agent might be subjected to initial transfer learning in which case the models will have different weights. Further details are described in workflow “E” ***table S2***. The role of the prior is to serve as a reference point for the likelihood of sampling a given SMILES. For every batch of SMILES generated by the agent, the prior calculates the negative log-likelihood denoted as *NLL (eq. 1)*. *NLL* reflects how likely it is to sample a sequence **S** from the model. $P(X_i = T_i | X_{i-1} = T_{i-1} \dots X_1 = x_1)$ is the probability of sampling a token T_i at step X_i given the previously sampled tokens. The minus sign is to account for the fact that large positive values are actually corresponding to a low probability.

$$NLL(S) = -\sum_{i=1}^{N} ln\, P(X_i = T_i | X_{i-1} = T_{i-1} \dots X_1 = x_1)$$

(1)

Analogously the NLL(S) for the given string ***S*** is also calculated by the agent. The SMILES string is also evaluated by the scoring function which we denote as a multi-parameter objective (MPO). MPO is a value in the range [0,1]. At this step the DF is used to evaluate whether the SMILES string has been sampled before or whether it satisfies the DF policy. The MPO score will be set to 0 if the DF filters determine that the provided compound already exists or if there are too many compounds of the same scaffold and their number exceeds the user defined threshold. For more details on the types of DF please consult with the supporting materials ***table S4***. The resulting MPO score is combined with the prior's likelihood and used to form the augmented likelihood (eq 2). The MPO score is multiplied by **σ** which is a scalar value used for scaling up the scoring function output to the same order of magnitude as the NLL. Otherwise, the low MPO score ranging between [0,1] will have no impact whatsoever. The higher MPO score translates into higher augmented likelihood values. Ultimately, the loss is calculated as the squared difference between the agent's likelihood and the augmented likelihood (eq 3).

$$NLL(\boldsymbol{S})_{Augmented} = NLL(\boldsymbol{S})_{Prior} - \boldsymbol{\sigma} * MPO(\boldsymbol{S})_{score}$$

(2)

$$loss = \left[NLL(\boldsymbol{S})_{Augmented} - NLL(\boldsymbol{S})_{Agent}\right]^2$$

(3)

The final component of the RL loop as shown on ***figure S2*** is inception. The purpose of inception is to keep track of previously well scored compounds and to randomly expose a subset of them to the agent thus helping to direct the learning. More details about inception are provided in the supporting information. Finally, after including the compounds from inception's memory to the batch the loss is propagated back and only the agent is updated thus receiving the feedback from its interaction with the environment. The environment is represented by the score modulating block on ***figure S2***. The prior on the other hand does not undergo any changes.

The duration of RL is pre-defined by the user in terms of number of RL steps to be performed. This is very case specific and is normally determined by the complexity of the problem at hand. In terms of computational cost the scoring function is the costliest element of the RL loop since it may contain a variable number of components including slow predictive models, docking and/or pharmacophore similarity (the latter two are not included in the current release).

Scoring Functions

REINVENT offers two general scoring function formulations (eqs 4 and 5). The individual components of the scoring function can be either combined as a weighted sum or as a weighted product [24]. The individual score components can have different weight coefficients reflecting their importance in the overall score. Score contribution from each component can vary in a [0,1] range. As a result, the overall score is also within the same [0,1] range. In the equations below the score for sequence ***x*** is denoted ***S*** and is either a weighted product (eq 5) or a weighted sum (eq 4). The user-selected components are denoted as ***p*** in both equations and the corresponding weights are denoted as ***w***. Weights can vary in the range of [1, +∞).

$$S(x) = \left[\prod_i p_i(x)^{w_i}\right]^{1/\sum_i w_i} \tag{4}$$

$$S(x) = \frac{\sum_i w_i * p_i(x)}{\sum_i w_i} \tag{5}$$

Both formulations are provided for user convenience and flexibility. More details on the scoring functions and a full list of components included in this release is provided in ***table S3*** and in the supporting materials.

Transfer Learning (TL)

As an alternative to the goal-directed generation, distribution-learning is also supported in REINVENT. This approach requires a pre-trained generative model with the generative capacity and the potential to sample compounds from a rather vast area of the chemical space. We refer to this generative model as the prior. This prior is subjected to transfer learning with a smaller set of compounds which are relevant for a given project. For example, if we aim to maximize a predictive model among the other components we would use all the compounds that are considered as active by this model. If we aim towards certain subseries of compounds we would only use those that share the series-specific features (for example scaffold). This will result in a model that produces compounds similar to the target dataset with a higher probability. We refer to that model to as “focused prior”. The user can subsequently sample this model and score the generated compounds by using the scoring mode in REINVENT.

As an alternative we could also use the resulting “focused prior” as an agent in the RL loop. The resulting generative model from distribution-learning is a suitable starting point for goal-directed generation [31]. This pre-focusing of the prior can speed up the overall RL process since the chance of producing compounds of relevance will be much higher compared to using a general, unfocused prior as an agent. Once focused, the agent will have an increased probability of sampling a chemical subspace of interest thus reaching a **state of productivity** sooner. Both use cases are further illustrated in **figure S3** and **table S2** within the supporting information.

Logging

Essential for monitoring of the learning process is the availability of a comprehensive logging system. In REINVENT we utilize Tensorboard [32] to provide information about the evolution of the agent during TL by sampling after each step and displaying the likelihood distribution for the sampled data. Stats on validity of the smiles and the most frequently encountered molecules are also shown. For RL we are plotting the evolution of the scoring function and the individual scoring component contributions to the overall score. We are also displaying the highest scoring compounds after each RL step. As an alternative, we also provide the implementation used by us for remote logging which can be set up to post the logging results to a custom REST endpoint.

Implementation

REINVENT is an open-source Python application. It uses PyTorch 1.3.0 [33] as a deep learning engine and RDKit version 2019.03.3.0 [34] as a chemistry engine. It works exclusively with scikit-learn based machine learning models and for the detailed logging of the chemical space navigation process, it makes use of Tensorboard's implementation in PyTorch.

Conclusion

We have described a production-ready, open-source application for de novo generation of small molecules. It can be used to address both exploration and exploitation type of problems while allowing a flexible formulation of complex MPO scores. Examples of various use cases are provided with the code repository and in https://github.com/MolecularAI/ReinventCommunity.

Apart from providing a ready-to-use solution, with releasing the code, we are hoping to facilitate the research on using generative methods for drug discovery. We also hope that it can be used as an interaction point for future scientific collaborations.

Funding

TB has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 676434, „Big Data in Chemistry" („BIGCHEM", http://bigchem.eu). The article reflects only the authors' view and neither the European Commission nor the Research Executive Agency (REA) are responsible for any use that may be made of the information it contains.

References

(1) Arús-Pous, J.; Patronov, A.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J.-L.; Chen, H.; Engkvist, O. SMILES-Based Deep Generative Scaffold Decorator for De-Novo Drug Design. **2020**. https://doi.org/10.26434/CHEMRXIV.11638383.V1.

(2) 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* (9), 1038–1040. https://doi.org/10.1038/s41587-019-0224-x.

(3) Winter, R.; Montanari, F.; Steffen, A.; Briem, H.; Noé, F.; Clevert, D. A. Efficient Multi-Objective Molecular Optimization in a Continuous Latent Space. *Chem. Sci.* **2019**, *10* (34), 8016–8024. https://doi.org/10.1039/c9sc01928f.

(4) Merk, D.; Friedrich, L.; Grisoni, F.; Schneider, G. De Novo Design of Bioactive Small Molecules by Artificial Intelligence. *Mol. Inform.* **2018**, *37* (1). https://doi.org/10.1002/minf.201700153.

(5) Grebner, C.; Matter, H.; Plowright, A. T.; Hessler, G. Automated De Novo Design in Medicinal Chemistry: Which Types of Chemistry Does a Generative Neural Network Learn? *J. Med. Chem.* **2020**. https://doi.org/10.1021/acs.jmedchem.9b02044.

(6) Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci.* **2018**, *4* (1), 120–131. https://doi.org/10.1021/acscentsci.7b00512.

(7) 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* (2), 268–276. https://doi.org/10.1021/acscentsci.7b00572.

(8) Li, Y.; Zhang, L.; Liu, Z. Multi-Objective de Novo Drug Design with Conditional Graph Generative Model. *J. Cheminform.* **2018**, *10* (1), 33. https://doi.org/10.1186/s13321-018-0287-6.

(9) Popova, M.; Isayev, O.; Tropsha, A. Deep Reinforcement Learning for de Novo Drug Design. *Sci. Adv.* **2018**, *4* (7), eaap7885. https://doi.org/10.1126/sciadv.aap7885.

(10) Sattarov, B.; Baskin, I. I.; Horvath, D.; Marcou, G.; Bjerrum, E. J.; Varnek, A. De Novo Molecular Design by Combining Deep Autoencoder Recurrent Neural Networks with Generative Topographic Mapping. *J. Chem. Inf. Model.* **2019**, *59* (3), 1182–1196. https://doi.org/10.1021/acs.jcim.8b00751.

(11) Skalic, M.; Jiménez, J.; Sabbadin, D.; De Fabritiis, G. Shape-Based Generative Modeling for de Novo Drug Design. *J. Chem. Inf. Model.* **2019**, *59* (3), 1205–1214. https://doi.org/10.1021/acs.jcim.8b00706.

(12) Grisoni, F.; Moret, M.; Lingwood, R.; Schneider, G. Bidirectional Molecule Generation with

Recurrent Neural Networks. *J. Chem. Inf. Model.* **2020**, *60* (3), 1175–1183. https://doi.org/10.1021/acs.jcim.9b00943.

(13) Méndez-Lucio, O.; Baillif, B.; Clevert, D. A.; Rouquié, D.; Wichard, J. De Novo Generation of Hit-like Molecules from Gene Expression Signatures Using Artificial Intelligence. *Nat. Commun.* **2020**, *11* (1), 1–10. https://doi.org/10.1038/s41467-019-13807-w.

(14) Blaschke, T.; Olivecrona, M.; Engkvist, O.; Bajorath, J.; Chen, H. Application of Generative Autoencoder in *De Novo* Molecular Design. *Mol. Inform.* **2018**, *37* (1–2), 1700123. https://doi.org/10.1002/minf.201700123.

(15) Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9* (8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

(16) Weininger, D. SMILES, a Chemical Language and Information System: 1: Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28* (1), 31–36. https://doi.org/10.1021/ci00057a005.

(17) Prykhodko, O.; Johansson, S. V.; Kotsias, P.-C.; Arús-Pous, J.; Bjerrum, E. J.; Engkvist, O.; Chen, H. A de Novo Molecular Generation Method Using Latent Vector Based Generative Adversarial Network. *J. Cheminform.* **2019**, *11* (1), 74. https://doi.org/10.1186/s13321-019-0397-9.

(18) Kotsias, P.-C.; Arús-Pous, J.; Chen, H.; Engkvist, O.; Tyrchan, C.; Bjerrum, E. J. Direct Steering of de Novo Molecular Generation Using Descriptor Conditional Recurrent Neural Networks (CRNNs). **2019**. https://doi.org/10.26434/CHEMRXIV.9860906.V2.

(19) Brown, N.; Fiscato, M.; Segler, M. H. S.; Vaucher, A. C. GuacaMol: Benchmarking Models for de Novo Molecular Design. *J. Chem. Inf. Model.* **2019**, *59* (3), 1096–1108. https://doi.org/10.1021/acs.jcim.8b00839.

(20) Moret, M.; Friedrich, L.; Grisoni, F.; Merk, D.; Schneider, G. Generative Molecular Design in Low Data Regimes. *Nat. Mach. Intell.* **2020**, *2* (3), 171–180. https://doi.org/10.1038/s42256-020-0160-y.

(21) Liu, X.; Ye, K.; van Vlijmen, H. W. T.; IJzerman, A. P.; van Westen, G. J. P. An Exploration Strategy Improves the Diversity of de Novo Ligands Using Deep Reinforcement Learning: A Case for the Adenosine A2A Receptor. *J. Cheminform.* **2019**, *11* (1), 35. https://doi.org/10.1186/s13321-019-0355-6.

(22) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular De-Novo Design through Deep Reinforcement Learning. *J. Cheminform.* **2017**, *9* (1), 48. https://doi.org/10.1186/s13321-017-0235-x.

(23) Arús-Pous, J.; Johansson, S. V.; Prykhodko, O.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J. L.; Chen, H.; Engkvist, O. Randomized SMILES Strings Improve the Quality of Molecular Generative Models. *J. Cheminform.* **2019**, *11* (1). https://doi.org/10.1186/s13321-019-0393-0.

(24) Cummins, D. J.; Bell, M. A. Integrating Everything: The Molecule Selection Toolkit, a System for Compound Prioritization in Drug Discovery. *J. Med. Chem.* **2016**, *59* (15), 6999–7010. https://doi.org/10.1021/acs.jmedchem.5b01338.

(25) Sutton, R. S.; Barto, A. G. *Reinforcement Learning: An Introduction*; 1998.

(26) Blaschke, T.; Engkvist, O.; Bajorath, J.; Chen, H. *Memory-Assisted Reinforcement Learning for*

Diverse Molecular de Novo Design; ChemRxiv, 2020. https://doi.org/10.26434/CHEMRXIV.12693152.V1.

(27) Gaulton, A.; Hersey, A.; Nowotka, M. L.; Patricia Bento, A.; Chambers, J.; Mendez, D.; Mutowo, P.; Atkinson, F.; Bellis, L. J.; Cibrian-Uhalte, E.; Davies, M.; Dedman, N.; Karlsson, A.; Magarinos, M. P.; Overington, J. P.; Papadatos, G.; Smit, I.; Leach, A. R. The ChEMBL Database in 2017. *Nucleic Acids Res.* **2017**, *45* (D1), D945–D954. https://doi.org/10.1093/nar/gkw1074.

(28) Xu, Y. J.; Johnson, M. Algorithm for Naming Molecular Equivalence Classes Represented by Labeled Pseudographs. *J. Chem. Inf. Comput. Sci.* **2001**, *41* (1), 181–185. https://doi.org/10.1021/ci0003911.

(29) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39* (15), 2887–2893. https://doi.org/10.1021/jm9602928.

(30) Wager, T. T.; Hou, X.; Verhoest, P. R.; Villalobos, A. Moving beyond Rules: The Development of a Central Nervous System Multiparameter Optimization (CNS MPO) Approach to Enable Alignment of Druglike Properties. *ACS Chem. Neurosci.* **2010**, *1* (6), 435–449. https://doi.org/10.1021/cn100008c.

(31) Renz, P.; Rompaey, V.; Wegner, K.; Hochreiter, S.; Klambauer, G.; Rompaey, D. Van; Wegner, J. K.; Unter Klambauer, G. On Failure Modes of Molecule Generators and Optimizers. *ChemRxiv* **2020**. https://doi.org/10.26434/chemrxiv.12213542.v1.

(32) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; Zheng, X.; Research, G. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*.

(33) Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; Facebook, Z. D.; Research, A. I.; Lin, Z.; Desmaison, A.; Antiga, L.; Srl, O.; Lerer, A. *Automatic Differentiation in PyTorch*.

(34) G., L. RDKit http://www.rdkit.org/ (accessed Feb 12, 2020).