Integrating synthetic accessibility with AI-based generative drug design

Maud Parrot[1], Hamza Tajmouati[1], Vinicius Barros Ribeiro da Silva[1], Brian Ross Atwood[1], Robin Fourcade[1], Yann Gaston Mathé[1], Nicolas Do Huu[1], and Quentin Perron[1*]

[1] *Iktos, 65 rue de Prony, 75017, Paris, France*

E-mail: quentin.perron@iktos.com

Abstract

Generative models are frequently used for de novo design in drug discovery projects to propose new molecules. However, the question of whether or not the generated molecules can be synthesized is not systematically taken into account during generation, even though being able to synthesize the generated molecules is a fundamental requirement for such methods to be useful in practice. Methods have been developed to estimate molecule synthesizability, but, so far, there is no consensus on whether or not a molecule is synthesizable. In this paper we introduce the Retro-Score (RScore), which computes a synthetic feasibility score of molecules by performing a full retrosynthetic analysis through our data-driven synthetic planning software Spaya, and its dedicated API: Spaya-API (https://spaya.ai). After a comparison of RScore with other synthetic scores from the literature, we describe a pipeline to generate molecules that validate a list of targets while still being easy to synthesize. We further this idea by performing experiments comparing molecular generator outputs across a range of constraints and conditions. We show that the RScore can be learned by a Neural

Network, which leads to a new score: RSPred. We demonstrate that using the RScore or RSPred as a constraint during molecular generation enables to obtain more synthesizable solutions, with higher diversity. The open-source Python code containing all the scores and the experiments can be found on https://github.com/iktos/generation-under-synthetic-constraint.

Introduction

In small molecule drug discovery projects, generative models can be used to design massive libraries of molecules with specific properties.[1,2] The optimization of an artificial intelligence(AI)-molecular generator to explore a given chemical space and propose new well scored molecules in a Multiparameter Optimization (MPO) project is mostly based on molecular properties and fingerprints.[1,3,4] However, one of the major challenges in any computer-aided drug design (CADD) project is that the molecules need to be synthesized. Generative models are known to sample lots of non accessible molecules,[5,6] and few synthesizability scores are known in the literature to be used in the pipeline of molecular generation.[7–10] Post-processing filters may be applied after the generation to get a smaller selection of molecules more likely to be synthesizable, for instance AstraZeneca filters,[11] include both physicochemical properties and structural filters. No chemical rule is able to completely answer the question of whether a molecule with a valid SMILES can be synthesized or not. Moreover, the evaluation of such scores are challenging, particularly due to the difficulty in interpreting the values. A simple way to define synthesizability is with a binary score denoting synthesizable or not synthesizable. Although a binary score is useful, it has limits, as it does not allow the prioritization of molecules of the same score. Also, a continuous score gives more signal when used as a reward of a de novo drug design algorithm. With the recent efforts of the community, some continuous scores were recently developed to describe synthetic accessibility.[12–15] Those can be based on chemical substructures, domain expertise, or output of models fitting expert

scores. However, as two very similar molecules may have different synthetic routes due to a single functional group, it may be difficult to find a proxy to a true restrosynthetic analysis. The RA score, for retrosynthetic accessibility score,[14] is a predictor of the binary score given by the AiZynthFinder retrosynthesis tool.[15] Its values goes from 0 to 1, and, according to the score, the higher the value the more optimistic the algorithm is regarding the synthesis of the molecule. The SC score, for synthetic complexity score,[12] ranks the molecules and scores them from 1 to 5. Based on the criteria that products are more complex than reactants, a neural network trained on a corpus of reactions was used to build the score. Molecules with lower values have a more optimistic synthesizability profile. Finally, the SA score, for synthetic accessibility score,[13] is based on a heuristics where molecular complexity and fragment contributions are used to evaluate synthetic tractability. Low scores indicate less complex molecules and consequently more feasible compounds. We believe that the features taken into account to compute these scores are not sufficient to encapsulate all of the information about synthesizability.

To address some of these challenges and to aid synthetic, medicinal, and computational chemists, Iktos has developed Spaya,[16] a template-based retrosynthesis AI software that computes synthetic routes and ranks them based on a synthesizability score. In this paper, we describe the Retro-Score (RScore), a synthetic feasibility score derived from the output of a full Spaya retrosynthetic analysis for a given molecule, and we compare it with three other synthesizability scores known in the literature (RA score, SC score and SA score). We highlight the importance of conducting a full retrosynthetic analysis to determine synthesizability. The RScore can be used:

1) To evaluate the synthesizability of molecules given by generative models,

2) Inside the generation itself, to guide the generator to an area of the chemical space where molecules are synthesizable.

Because of the computational costs associated with the computing of a full retrosynthetic analysis needed to obtain the RScore, we also describe a new, easier to compute score called

RSPred. RSPred is obtained by training a Neural Network on the output of the Spaya RScore and performs similarly well to the RScore in a variety of tasks, but can be computed orders of magnitude faster.

Methods

Datasets

The ChEMBL 24[17] dataset was used, with the same post-processing as described in the Guacamol Benchmark experiments.[6] The post-processed ChEMBL dataset can be downloaded following the link.[18]

Another dataset, named 'PI3K/mTOR',[19] was also used. It is a library of 463 structurally homogeneous molecules containing values of IC50 for the two targets Pi3K (pKi measured on the Phosphoinositide 3-Kinase) and mTOR (pKi measured on the mechanistic Target Of Rapamycin), from the Chembl database. After the definition of a threshold of activity, pIC50 Pi3K ≥ 7 and pIC50 mTOR ≥ 8.5, the molecules active for both targets were removed. The dataset is accessible in the GitHub project associated with this paper.[20]

The RScore from Spaya API

The score of a retrosynthesis route in Spaya is a composite of four scores as follows,

$$score(route) = f(d, p, c, a) \tag{1}$$

where:

d = number of reaction steps in the route

p = likelihood of the disconnections of the retrosynthesis route

c = convergence of the route

a = applicability domain estimation of the reaction templates used to make the disconnections

To simplify the use of the algorithm on large batches of molecules, Iktos has recently launched Spaya-API,[16] an API running on Spaya's algorithmic engine for library scoring purposes, which has been used herein to evaluate the synthetic accessibility of newly generated molecules. For a given molecule (m), the RScore is derived from routes proposed by Spaya, but handled in a high throughput manner by Spaya-API. The worst RScore value is 0, when no route is found by Spaya in a given period of time; and the best score is 1, when the route is a one step retrosynthesis matching exactly a reaction in the literature. To score a molecule and obtain its RScore value, Spaya-API performs a retrosynthetic analysis with an early stopping process. The early stopping mode stops the Spaya run when a route with a score above the predefined threshold (set to 0.6 by default) is found or after the defined timeout (set to 1min by default) has elapsed. The RScore of a molecule is defined as:

$$RScore(m) = \max_{\substack{routes\ given\ by\ Spaya \\ with\ early\ stopping}} \Big(score(route(m) \Big) \tag{2}$$

The score is rounded to 1 decimal, and hence can take 11 different values (from 0.0 to 1.0). Spaya-API also returns the number of steps for the best synthetic route found for each input molecule. The list of commercial compounds used for the retrosynthesis is a catalog of 60M commercially available starting materials provided by MCule,[21] Chemspace,[22] eMolecules,[23] and Key Organics.[24] To speed up computation, a default timeout of 1 minute was set when the RScore was used as a synthetic constraint in generative design experiments (RScore1min). In order to better approximate the output that would be obtained from a comprehensive retrosynthetic search, this timeout was increased to 3 minutes when the RScore was used for

scoring molecules in post-processing (RScore3min).

The RScore1min was compared with three synthetic scores previously published in the literature: the RA score,[14] the SC score,[12] and the SA score.[13] The three packages to compute those scores are available on GitHub.[25–27] These scores were computed on a sample of 5000 molecules from the pre-processed ChEMBL dataset, and were compared with the RScore1min in the section *Comparison of synthetic scores*.

Prediction of RScore

The RScore1min computation implies a full retrosynthesis, which is time consuming, with an average of 42 seconds per molecule to trigger the early stopping. For that reason a regression model was built, in the hope to replace the computation of the RScore1min by a simple neural network inference.

To build this continuous predictor of the RScore1min, a neural network was trained on features of the molecules. The model was a feed-forward neural network composed of three hidden layers of size 100, with Relu activation function. After each layer, a batch normalization layer was added.[28] A sigmoid was added as the last activation function. The training set was composed of 70K molecules from the pre-processed ChEMBL dataset, and 300K molecules sampled from the generator Guacamol pretrained on ChEMBL. The molecules were represented by real vectors of the ECFP2 fingerprints with a radius of 2, modulo-folded to size 8192 and then ln(x + 1)-preprocessed. For the training part, a dropout[29] with a probability of 0.05 was used, the loss was the mean squared error, the optimizer was the Adam optimizer[30] with an initiate learning rate of 5e−5, the batch size was 2048, and 6 epochs were used for training.

Generations of molecules

For all the generations, the package Guacamol[31] provided by BenevolentAI was used. The generator is a Recurrent Neural Network, containing 3 layers of Long Short-Term Memory

(LSTM) of size 1024. The network was initialized with the weights given by Guacamol on their GitHub project,[31] which was obtained by training on the large dataset ChEMBL 24.[17] For each generation, the reward used was a geometric mean of the different scoring functions on which modifier functions (described in *Score modifiers* section) were applied. The generators were optimized in order to sample molecules that have a good reward. The optimization algorithm used was the Hill Climbing MLE[1] , in which at each step 1024 molecules are being sampled from the generator, then scored, and 2 epochs of teacher forcing[32] are performed on the top scored 152 molecules. The seed was fixed at 42 for all the generations. Overall 46 generations were run : 40 Guacamol generations (for each of the 10 tasks one generation without and 3 with synthetic constraint), and 6 PI3K/mTOR generations (a generation without and 5 with synthetic constraint). The implementations of those generations can be found on GitHub.[20]

Generations without any synthetic accessibility constraint

The first generations were performed using the standardized Guacamol Benchmark. Molecules were generated over 20 epochs on each of the 10 MPO tasks of the Guacamol Benchmark, which are: Osimertinib MPO, Fexofenadine MPO, Ranolazine MPO, Perindopril MPO, Amlodipine MPO, Sitagliptin MPO, Zaleplon MPO, valsartan SMARTS, Deco Hop and Scaffold Hop. Each task is associated with an objective function, the description of each task can be found in the original paper.[6]

The next generation aimed at solving a lead optimization problem on the PI3K/mTOR dataset. The constraints for this task were the Tanimoto similarity of ECFP4 fingerprints to the initial dataset, the Quantitative Estimate of Drug-likeness (QED),[33] and predicted Pi3K and mTor pKi values. For Pi3K and mTOR pKi predicted values, two QSAR models were used as scorers during the ensuing generative procedure. Those were built using ECFP molecular representation with 4096 bits and with radius 4 for mTor and 6 for Pi3K, molecular descriptors, and a ridge regression model. K-fold (K=4) cross validation along with tree-

structured Parzen Estimator was used to select the model and the fingerprints parameters. On a 20% hold out set, the R2 score of the Pi3K model and the mTor model are respectively 0.64 and 0.71. In addition to these scores, a filter was added to enforce a specific substructure within the generated molecules, which pattern corresponds to the following SMARTS and that is drawn on fig. 1. The thresholds for each of the targets can be found in table 1. The objective function associated with this task was the geometric mean of the 5 scoring functions as in eq. (4).

$$smarts_1 = c1cncc(c1)C\#Cc1cncnc1 \tag{3}$$

Figure 1: Imposed structure for PI3K/mTOR generation.

Table 1: Blueprint of the task PI3K/mTOR

criteria	**specification**
Pi3K	>7
mTOR	>8.5
QED	>0.5
Tanimoto similarity	>0.5
contains structure	$\text{smarts}_1(eq.3)$

$$Score(mol) = GeoMean\Big(score_1(mol), ..., score_5(mol)\Big) \tag{4}$$

where:

$$GeoMean(x_1, ..., x_n) = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}} = \sqrt[n]{x_1 x_2 \cdots x_n} \tag{5}$$

The prior model was the one trained on ChEMBL, then 2 steps of transfer learning were run on the PI3K/mTOR dataset in order to stay in the applicability domain of the QSAR regressors. For the training part, the batch size was 1024, the learning rate 1e-3 and the generation run over 250 epochs.

Generations under synthetic accessibility constraint

Generations under synthetic constraint used the same parameters as described above, while incorporating a synthetic accessibility score in the reward. Compared to the previous generations, only the scoring function was changed, and the different synthetic scores were added in the objective function as follows:

$$Score(mol) = GeoMean\Big(score_1(mol), ..., score_k(mol), ScoreSynth(mol)\Big) \tag{6}$$

Where $ScoreSynth$ can be any function that estimates synthetic accessibility: RA, SC, SA, RScore1min, or RSPred, on which a modifier function is applied. The function $GeoMean$ is described in equation 5.

For each of the 10 Guacamol tasks, 3 generations were run with the ScoreSynth being successively SA score, RScore1min and RSPred. For the PI3K/mTOR task, 5 generations were run with the ScoreSynth being successively RA score, SC score, SA score, RScore1min and RSPred.

We conducted a post-processing analysis of the results using the RScore3min. This score was considered the *ground truth* of synthetic accessibility, and the other synthetic scores were evaluated for their relevance as estimates of synthesizability as provided by RScore3min.

In total, 35 generations under synthetic constraint were performed: 30 for the Guacamol Benchmark tasks, and and 5 for the PI3K/mTOR task.

Score modifiers

On each scoring function a modifier function is applied in order to normalize the score into the range [0, 1]. The modifier and its parameters are chosen based on the expected threshold for each target, and are well described in the literature.[6] The two modifiers used are MaxGaussian and MinGaussian:

- MinGaussian(μ, σ): the right half of a Gaussian function. Values smaller than μ are given full score, and values larger than μ decrease continuously to zero.

- MaxGaussian(μ, σ): the left half of a Gaussian function. Values larger than μ are given full score, and values smaller than decrease continuously to zero.

The modifiers of the Guacamol tasks are specified in the original paper. The modifiers used in PI3K/mTOR task are described in table 2.

Table 2: Modifiers used for the different scoring functions

	modifier
Pi3K	MaxGaussian(7, 1)
mTOR	MaxGaussian(8, 1)
QED	MaxGaussian(0.6, 0.13)
Similarity	MaxGaussian(0.75, 0.25)
RA Score	MaxGaussian(0.7, 0.2)
SC Score	MinGaussian(2.5, 0.4)
SA Score	MinGaussian(2.5, 0.4)
RScore1min	MaxGaussian(0.7, 0.2)
RSPred Score	MaxGaussian(0.7, 0.2)

Results and discussions

This part consists in first comparing the values of the different synthetic scores on molecules from the ChEMBL dataset, then evaluating the performances of the RScore1min predictor (RSPred), and finally analyzing the results of the different generations with and without synthetic constraints.

Comparison of synthetic scores

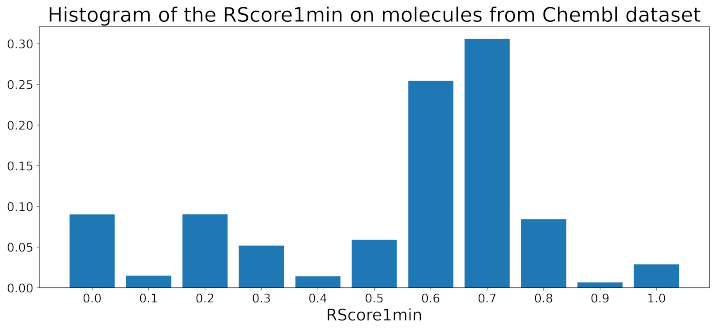


Figure 2: Normalized histogram of the RScore1min on molecules from Chembl dataset.

Based on our experience, we consider the threshold for a good RScore to be 0.5. The total distribution of the RScore1min on a sample of molecules from Chembl 24 is plotted on fig. 2. It can be seen that around 66% of the sample have a good RScore1min (≥ 0.5), that a significant part of the dataset is not solved by Spaya API and that a major mode around 0.7 is observed. The RScore is not directly interpretable but it takes into account the number of synthesis steps, which is a meaningful metric for chemists. The graph fig. 3 is a plot of the correlation between the RScore1min and the number of steps of synthesis found by Spaya for the ChEMBL dataset sample.

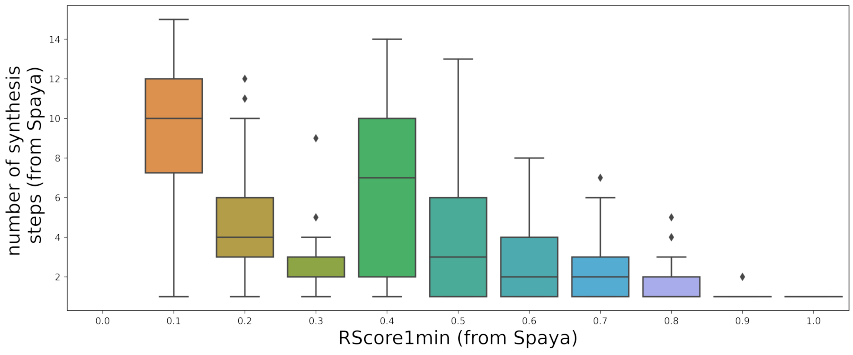


Figure 3: Correlation between the RScore1min and the number of synthetic steps given by Spaya API on a sample from Chembl dataset.

It can be seen that few synthesis steps (less than 6) is a necessary condition for having a good RScore, though the contrary is not true. For instance a 2 steps route may have a bad

score due to low probability disconnections. Indeed, the scoring function in eq. (1) considers other elements than the number of steps to evaluate the route.

As previously discussed, existing literature scores designed to estimate synthesizability do not perform a full retrosynthetic analysis of the target molecule. Those scores were compared on a bench of molecules in order to analyze to what extent they agree with each other.

On the ChEMBL dataset sample, the RA score (fig. 4) often predicts a score of almost 1. Hence, this score is not useful to measure the difficulty of synthesis of feasible compounds (fig. 5). This can be explained by the fact that the model computing the RA score was trained on a subset of ChEMBL. The SA score is significantly correlated to the RScore1min (fig. 6). Having a good SA score seems to be a sufficient condition to have a good RScore, while the contrary is not true : molecules with complex fragments will often have a bad SA score, even if they are synthesizable. As an example, the molecules in fig. 12 contain original and complex fragments, but are easy to synthesize through Spaya.The SC score has no correlation at all with the RScore1min. (fig. 7).

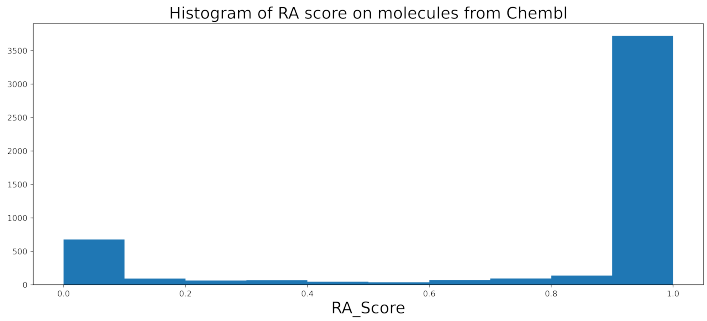


Figure 4: histogram of RA score on the Chembl dataset

RSPred

In hopes of replacing a full retrosynthetic analysis with a prediction, a deep learning model was trained to predict the RScore1min obtained with Spaya-API. The performance of the neural network was evaluated on a hold out test set. The box plot of the predictions made by the neural networks with regards to the true RScore is shown in fig. 8. With a Pearson

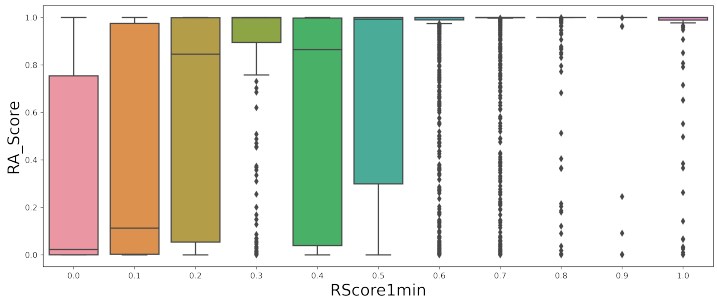


Figure 5: Correlation between RA score and RScore1min on Chembl dataset

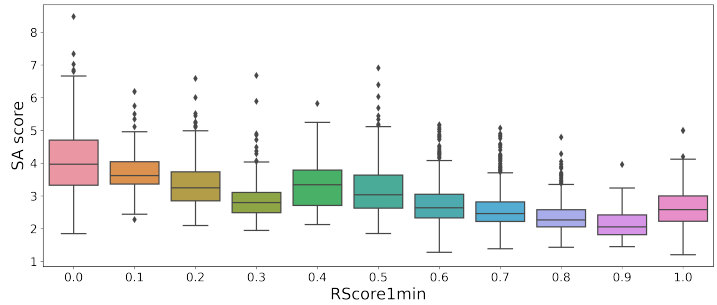


Figure 6: Correlation between SA score and RScore1min on Chembl dataset

correlation of 0.75, the results are quite satisfying. For this reason, the prediction of the neural network is considered as a new synthetic score, RSPred, used as an additional constraint in generation.

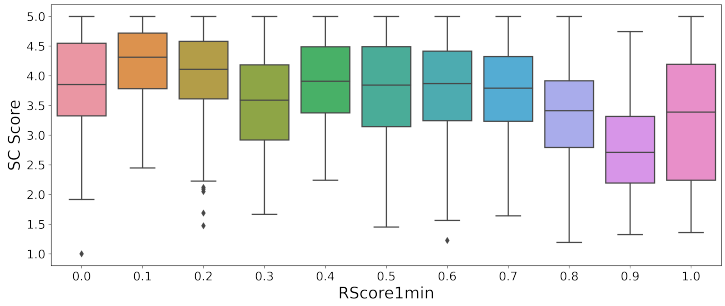


Figure 7: Correlation between SC score and RScore1min on Chembl dataset

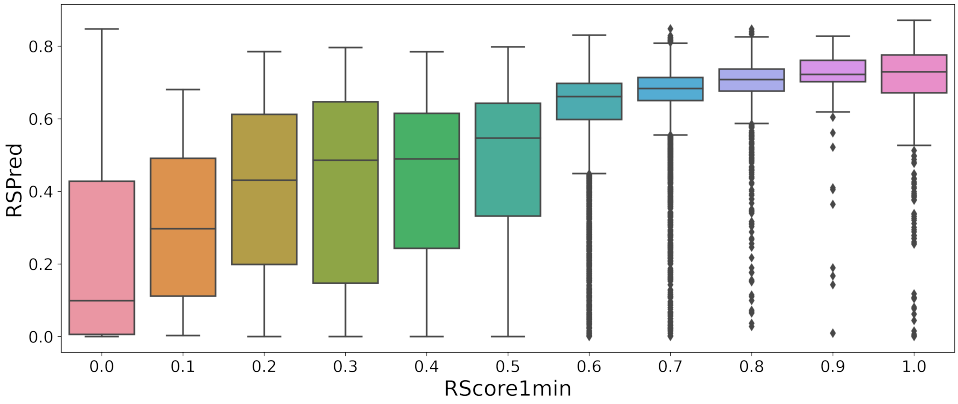


Figure 8: Correlation between the RScore1min and the values predicted from the neural network on a test set

Table 3: Computing time per molecule for the different synthetic scores.

Synthetic score	**Time per molecule (ms)**
RA score	28
SC score	241
SA score	2
RScore	40000
RSPred	1

Computing time

Computing time is an essential attribute of a score as it may limit its usage on large scale data sets. Table 3 displays computing time estimates of the different synthetic scores. The RScore1min, being obtained through a full retrosynthesis, is by far the most time consuming score. Thanks to its scalability, Spaya-API accelerates RScore computation on batches of molecules. The prediction of the latter, RSPred, is the fastest score to compute, only 1ms per molecule, 40 000 time faster than the RScore1min. The SA score closely follows with 2ms per molecule, the RA score is one order of magnitude slower while the SC score is two orders of magnitude slower.

Evaluations of generations on 10 Guacamol tasks

Table 4: RScore3min and reward of the top 100 molecules of the Guacamol generations without any synthetic accessibility constraint.

Task name	Average RScore3min	% with RScore3min ≥ 0.5	Average reward
Amlodipine MPO	0.55	77	0.86
Deco HOP	0.66	97	0.99
Fexofenadine MPO	0.66	95	0.89
Osimertinib MPO	0.50	74	0.50
Perindopril MPO	0.59	94	0.59
Ranolazine MPO	0.39	50	0.39
Scaffold Hop	0.58	81	0.58
Sitagliptin MPO	0.60	82	0.60
Valsartan SMARTS	0.62	90	0.62
Zaleplon MPO	0.69	99	0.68

In this section, we evaluate how synthesizable are the most optimal generated molecules from the 10 Guacamol tasks. We then consider the impact of adding a synthetic constraint during the generation. The results are analyzed based on the initial objective functions as well as the synthetic feasibility.

Table 4 contains the reward and RScore3min (3 minutes timeout) of the top 100 molecules generated without any synthetic constraint for each task. The ranking is performed based on the reward of each task. It can be noticed that the top 100 molecules are already good in terms of feasibility and reward: an average of 98% of molecules are synthesizable according

to Spaya-API, and a large majority even have a good RScore3min (above 0.5).

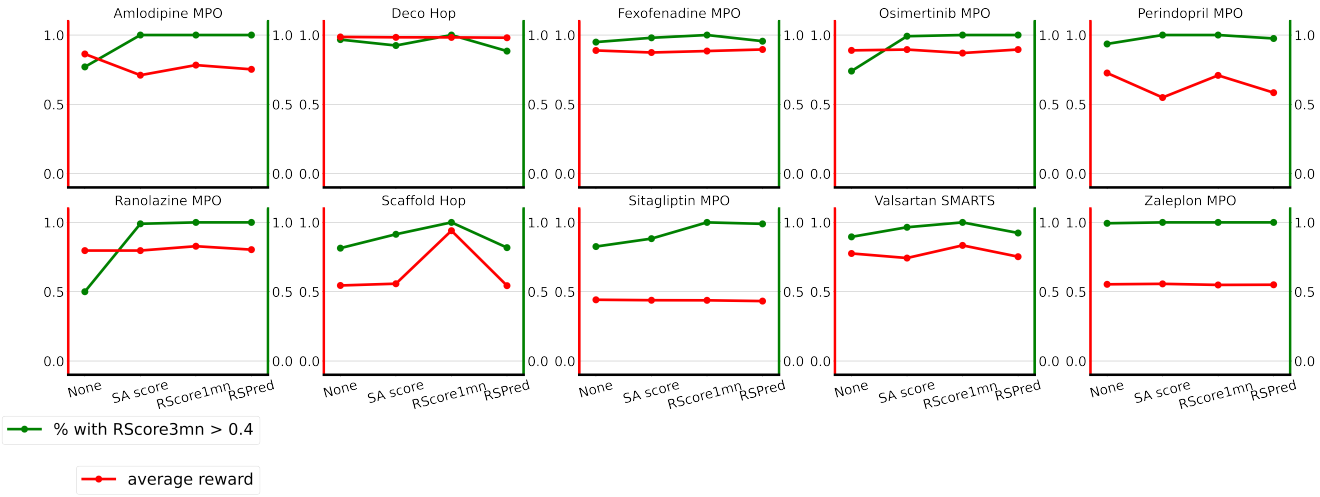


Figure 9: Reward and accessibility of the top 100 molecules for each task and with different synthetic constraints. The red line is the average reward (without the synthetic score) on the top 100 molecules of the generation. The green line is the % of the top 100 molecules with a RScore3min $\geq$ to 0.5

For each of the 10 benchmarks, in addition to the generations without any synthetic constraint, 3 generations were run with:

1) SA score constraint;

2) RScore1min constraint;

3) RSPred constraint;

All those generations are compared based on 2 metrics: the RScore3min on the top 100 molecules, and the average reward on the top 100 molecules, where the top 100 are selected based on their score on the initial objective function. The plots in fig. 9 summarize the results of the different generations. As previously stated, even without any synthetic constraint in the scoring function, the top 100 molecules of these generations have a reasonably good RScore3min. The SA score constraint improves the RScore3min of the top molecules, and the RScore1min and RSPred constraints improve it even more. Importantly, the reward is generally not degraded by the synthetic score constraint.

Being too easy, these tasks may be insufficient to evaluate the impact of adding a synthetic constraint during generation. Indeed, real-life drug design projects suffer more from synthetic

accessibility issues as they are harder to solve. The intuition is that when the generator struggles to find a solution, it designs more and more awkward structures to satisfy the goal criteria, resulting in molecules which are likely not synthesizable. Hence, the generation under synthetic constraint is a potential solution as it keeps orienting the generative model in a chemical space of feasible molecules. This is the motivation behind the "PI3K/mTOR experiment": it is a more realistic model of a real-life drug design project and reflects better the impact of using a constraint on synthetic accessibility.

Evaluations of generations on PI3K/mTOR dataset

This task is a generation around a library of 463 structurally homogeneous PI3K and mTOR inhibitors. The objective and targets can be found in table 1. This dataset serves as a simplified proxy for a real life MPO in a lead optimization project with four objectives to be optimized (table 1). Six generations were run based on this dataset: one without any synthetic score constraint, and 5 with synthetic score constraints (RA, SC, SA, RScore1min, and RSPred). When looking at the evolution of each component of the score among epochs, it can be noted that the reward increases and saturates systematically around the epoch 60 (fig. 13).

Synthetic feasibility of generated molecules in the blueprint

Here, the main metric to evaluate the quality of a generation method is the number of generated molecules validating all the constraints which also have a good RScore3min. The number of generated molecules for each generation (table 5) is roughly constant (+/- 3%), but the number of distinct molecules is more variable. A molecule is said to be in the blueprint when the computed value of each objective is in the desired range. The graph fig. 10 shows for each of the 5 generations how many molecules validate the thresholds, and their RScore3min range.

First, the generation without synthetic constraint and the one with the RA constraint

Table 5: Number of molecules generated for each generation; the first column indicates what synthetic score constraint was used.

synthetic constraint	n molecules	n distinct molecules
None	80399	16085
RAscore	80663	14509
SCscore	80291	14635
SAscore	78612	12775
RScore1min	83215	11081
RSPred	79861	12703

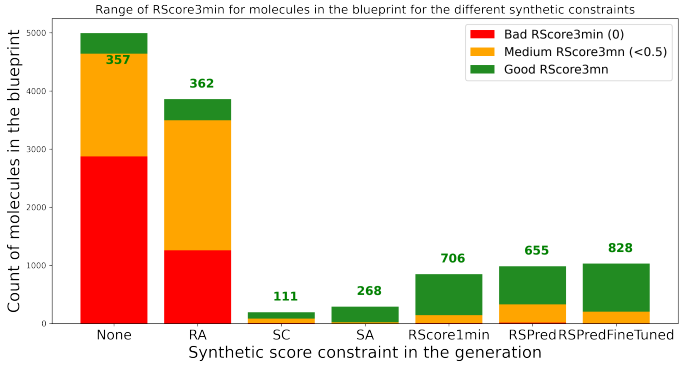


Figure 10: Molecules that validate all the constraints for each generation, with indication on their RScore3min range.

both contain a high percentage of non-synthesizable molecules, which would be problematic in a real-life project. For the other generations, almost all molecules in the selection are synthesizable. The generation with the RScore1min constraint gives, unsurprisingly, the best results, with more than twice as many optimal molecules compared to the SA generation. The RSPred generation gives almost as many easy to make molecules as the RScore1min generation. It seems that RSPred was sufficient to lead the generative algorithm towards the generation of synthesizable molecules. Finally, the generation under SC constraint gave very few molecules in the blueprint (fig. 10).

Diversity of generated molecules

Table 6 displays information about the generated molecules which met all objectives described in table 1. It shows that generations under RScore1min or RSPred constraint enabled to find 2 to 4 times more easy-to-make molecules than the other generation methods. We

Table 6: Some statistics about the molecules in the blueprint for the 6 generations. Standard Murcko and Generic Murcko : the number of different Murcko. Feasible/good RScore : number of molecules with RScore > 0 / ≥ 0.5. Unique : # of molecules that are only in this generation (and not any of the 5 others). All the columns after 'good RScore3min' refer to the molecules in the blueprint with a good RScore3min.

synth constraint	count	average RScore3min	feasible	good RScore3min	standard Murckos	generic Murckos	Unique
None	5005	0.08	1959	282 (6%)	59 (21%)	36 (13%)	34
RA	3574	0.11	2660	360 (10%)	79 (22%)	47 (13%)	64
SC	211	0.35	202	127 (60%)	19 (15%)	14 (11%)	64
SA	311	0.56	311	286 (92%)	40 (14%)	31 (11%)	145
RScore1min	850	0.49	843	706 (83%)	69 (10%)	46 (7%)	314
RSPred	985	0.46	971	655 (66%)	104 (16%)	73 (11%)	357

notice that the generations with no synthetic constraint and under RA constraint give more molecules in the blueprint but few of those have a good RScore3min, while the generations under SC and SA constraints give less molecules in the blueprint, and less molecules in the blueprint with a good RScore3min.

To evaluate the diversity we counted the number of Murcko scaffolds and generic Murcko scaffolds[34] among the molecules in the blueprint with a good RScore3min. We observe that diversity is not significantly different among the different methods with RScore1min and RSPred generations producing more scaffolds than the other methods. Also, those methods enabled to generate a significant number (more than 300) of compounds which could not be found with he other methods. This seems to imply that the synthetic constraint in the RScore and RSPred generations led the generative algorithm to explore an unseen area of the chemical space and to identify solutions meeting the blueprint and the synthetic feasibility constraint that could not be found with other methods.

Discussion

To summarize our results, in the PI3K/mTOR experiment that we conducted, RScore1min appeared to the best synthetic feasibility score to use as a synthetizability constraint integrated in an MPO generation, as it outperformed the other methods in generating a high

number of compounds in the blueprint with a good synthetic feasibility score (RScore3min). The other important point is that RSPred is a good proxy of the RScore1min with a much lower computational cost. The SA score has some correlation to the RScore, but the generation under SA constraint outputs less than half as many interesting molecules than the generation under RScore1min or RSPred constraint. The other synthetic constraints were not very useful in this experiment: the RA score has a very bad precision, meaning that among the molecules well scored by RA score, very few actually have a good RScore3min, and when included in the reward of a generation, almost all molecules get a high reward and the generator can't be optimized towards easier to make molecules; the SC score has no correlation to the RScore3min, so it comes as no surprise that the generation under SC score constraint fails to optimize the RScore3min during the generation, and gives poor results. In order to illustrate the output of those generations, we show in fig. 15 to fig. 20 the top 10 molecules of each generation, where the selection process was the following: after filtering on the molecules validating the 4 thresholds, the top 10 molecules regarding the optimized synthetic score were selected. In fig. 14 are exhibited some molecules generated under RScore1min constraint that may be interesting according to a chemist. An example of a synthetic route can be found in fig. 11. This route contains 4 commercial compounds and 4 synthesis steps (A, B, C and D).

This work has two main limitations: firstly, the MPO experiment was conducted only on one dataset, the PI3K/mTOR dataset. Reason for this is the difficulty to find adequate publicly available datasets which are representative of the challenges of multi-parametric optimization in real-life lead optimization projects. We found the MPO datasets and tasks available in the Guacamol benchmark way too easy to solve, and therefore not adequate for our purpose. Additional work has been performed by Iktos on other MPO datasets (not disclosed), showing similar results and conclusions than with the PI3K/mTOR experiment; secondly, in all our experiments of generations under synthetic constraint, we consider the "ground truth" of synthetic accessibility to be the RScore3min, which creates a strong bias,

since by construction the RSCore1min is strongly correlated to the RScore3min. It is therefore not surprising that generations under RScore1min perform better to generate molecules with good RScore3min than other synthetic constraints. The reason for such choice in the design of our experiments is that there is no known computational score which could be considered as an objective measure of the synthetic feasibility of molecules and chemists themselves may sometimes disagree on the ease of synthesis of a given molecule, which may also vary depending on the building blocks available. We conducted experiments (data not shown) showing that indeed, the distribution of the RScore for a sample of molecules shifts when only simpler and cheaper building blocks are used in the retrosynthesis. Finally, the major advantage of the RScore is that it derives from the output of a real retrosynthetic analysis, and our internal experience leads to trust the RScore as a good measure of synthetic feasibility, as our internal chemists tend to agree in most cases with the relevance of the score.

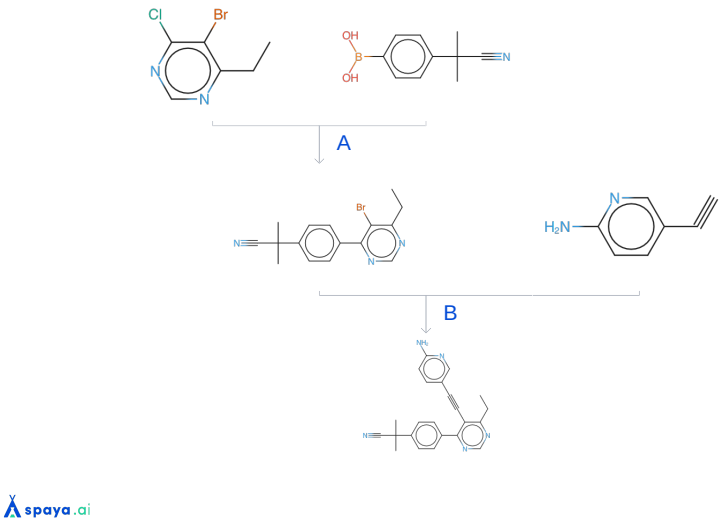


Figure 11: Example of a synthesis route obtained by Spaya.

Conclusion

Molecular generation methods are known to produce unrealistic structures which can be unsynthesizable and known synthetic scores fail to address that issue. In this paper, we introduce a new synthetic accessibility score, RScore, derived from Spaya,[16] a data-driven synthetic planning software developed by Iktos. The main advantage that distinguishes RScore from other synthetic scores is that it is computed from the output of a full retrosynthetic analysis performed by Spaya. Also, the tool can be designed to better adapt to a use case, thanks to customization options: the user can impose intermediate products to be in the routes, limit the number of steps and customize the list of starting materials. For relatively simple molecular generation tasks, it was demonstrated that the RScore as a post-processing filter can be sufficient to ensure synthetic accessibility of optimal solutions. For more problematic cases, where generations without synthetic constraint tend to suggest complex structures, adding an explicit synthetic constraint during the generation enables the design of synthetically accessible compounds by the generative algorithm. The computational complexity of the RScore is a limitation, hence a predictor of the RScore was built in order to accelerate the scoring. In a relatively difficult MPO task, generations under constraint of different synthetic scores were compared, and RScore and RSPred constrained generations gave the best results. Hence RSPred, in addition to being fast to compute, produced good results. However, just as any machine learning models, the predictor RSPred has a certain applicability domain, that is around the Chembl dataset. In the PI3K/mTOR experiment, the initial chemical space was already in the applicability domain of the predictor, which may explain the success of the generation under RSPred constraint. But in other cases, when the chemical space is far from Chembl, the predictor may have poor results and might lead the generation to an area of false positives. To address that issue, a preliminary fine tuning of the predictor on the chemical space of the generation might be helpful, if not necessary, to make sure the predictor's performance is still sufficient. Investigations are on going regarding that topic.

References

(1) Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Central Science* **2018**, *4*, 120–131, PMID: 29392184.

(2) Perron, Q.; Mirguet, O.; Tajmouati, H.; Skiredj, A.; Rojas, A.; Gohier, A.; Ducrot, P.; Bourguignon, M.-P.; Sansilvestri-Morel, P.; Do Huu, N.; et al., Deep Generative Models for Ligand-based de Novo Design Applied to Multi-parametric Optimization. *ChemRxiv* **2021**,

(3) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics 9*, 48.

(4) Gmez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernndez-Lobato, J. M.; Snchez-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 Central Science* **2018**, *4*, 268–276, PMID: 29532027.

(5) Renz, P.; Van Rompaey, D.; Wegner, J. K.; Hochreiter, S.; Klambauer, G. On failure modes in molecule generation and optimization. *Drug Discovery Today: Technologies* **2019**, *32-33*, 55–63, Artificial Intelligence.

(6) Brown, N.; Fiscato, M.; Segler, M. H.; Vaucher, A. C. GuacaMol: Benchmarking Models for de Novo Molecular Design. *Journal of Chemical Information and Modeling* **2019**, *59*, 1096–1108.

(7) Bradshaw, J.; Paige, B.; Kusner, M. J.; Segler, M. H. S.; Hernández-Lobato, J. M. A Model to Search for Synthesizable Molecules. *CoRR* **2019**, *abs/1906.05221*.

(8) Bradshaw, J.; Paige, B.; Kusner, M. J.; Segler, M. H. S.; Hernández-Lobato, J. M.

Barking up the right tree: an approach to search over molecule synthesis DAGs. *CoRR* **2020**, *abs/2012.11522*.

(9) Liu, C.; Korablyov, M.; Jastrzebski, S.; Wlodarczyk-Pruszynski, P.; Bengio, Y.; Segler, M. H. S. RetroGNN: Approximating Retrosynthesis by Graph Neural Networks for De Novo Drug Design. *CoRR* **2020**, *abs/2011.13042*.

(10) Gao, W.; Coley, C. W. The Synthesizability of Molecules Proposed by Generative Models. *Journal of Chemical Information and Modeling* **2020**, *60*, 5714–5723, PMID: 32250616.

(11) Cumming, J. G.; Davis, A. M.; Muresan, S.; Haeberlein, M.; Chen, H. Chemical predictive modelling to improve compound quality. *Nature Reviews Drug Discovery 12*, 948–962.

(12) Coley, C. W.; Rogers, L.; Green, W. H.; Jensen, K. F. SCScore: Synthetic Complexity Learned from a Reaction Corpus. *Journal of Chemical Information and Modeling* **2018**, *58*, 252–261, PMID: 29309147.

(13) Ertl, P.; Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics 1*, 8.

(14) Thakkar, A.; Chadimov, V.; Bjerrum, E. J.; Engkvist, O.; Reymond, J.-L. Retrosynthetic accessibility score (RAscore) rapid machine learned synthesizability classification from AI driven retrosynthetic planning. *Chem. Sci.* **2021**, *12*, 3339–3349.

(15) Genheden, S.; Thakkar, A.; Chadimov, V.; Reymond, J.-L.; Engkvist, O.; Bjerrum, E. AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning. *Journal of Cheminformatics 12*, 70.

(16) Website Spaya. `https://spaya.ai/`.

(17) Mendez, D. et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research 47*, D930–D940, _eprint: https://academic.oup.com/nar/article-pdf/47/D1/D930/27437436/gky1075.pdf.

(18) Post-processed ChEMBL datasets. https://figshare.com/ projects/GuacaMol/56639, Accessed: Nov 20,2018.

(19) J. A. Engelman, Nature Reviews Cancer, 2009, 9, 550-562; A. Carnero, Expert Opin. Investig. Drugs, 2009, 18, 1265-1277 and P. Liu et al., Nature Reviews Drug Discovery, 2009, 8, 627-64.

(20) Iktos, GitHub containing the code reproducing the paper. `https://github.com/iktos/generation-under-synthetic-constraint/`.

(21) MCule. `https://mcule.com/`.

(22) Chem-space. `https://chem-space.com/`.

(23) e-molecule. `https://www.emolecules.com/`.

(24) Key Organics. `https://www.keyorganics.net/`.

(25) RA score repository. `https://github.com/reymond-group/RAscore`.

(26) SC score repository. `https://github.com/connorcoley/scscore`.

(27) SA score repository. `https://github.com/EricTing/SAscore`.

(28) Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 9.

(29) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overtting. 30.

(30) Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* **2014**,

(31) BenevolantAI, Guacamol github. `https://github.com/BenevolentAI/guacamol/`.

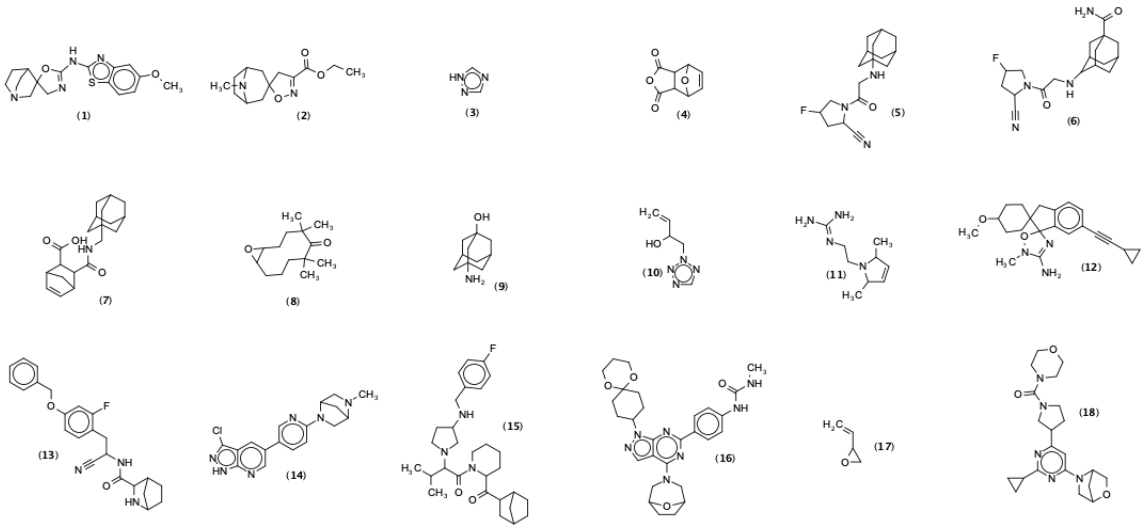
(32) Lamb, A.; Goyal, A.; Zhang, Y.; Zhang, S.; Courville, A.; Bengio, Y. Professor Forcing: A New Algorithm for Training Recurrent Networks.

(33) Bickerton, R.; Paolini, G.; Besnard, J.; Muresan, S.; Hopkins, A. Quantifying the chemical beauty of drugs. *Nature chemistry* **2012**, *4*, 90–8.

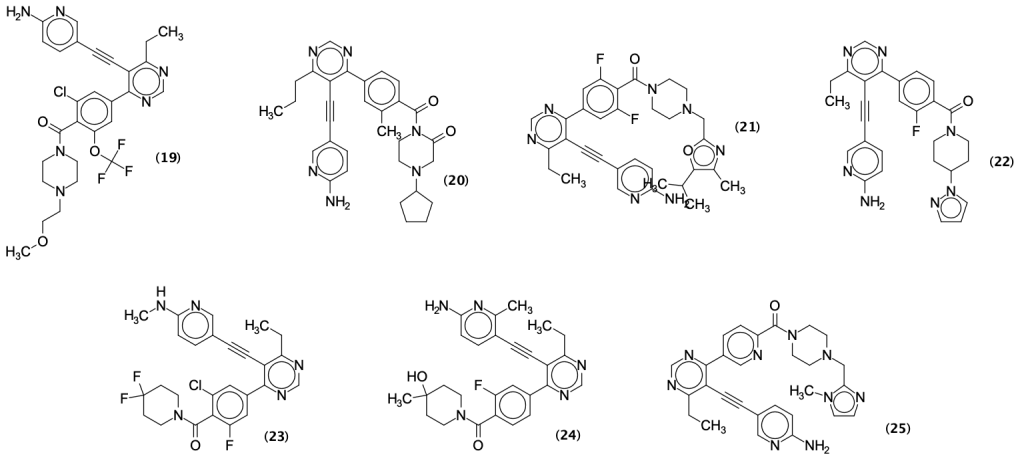
(34) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *Journal of Medicinal Chemistry* **1996**, *39*, 2887–2893, PMID: 8709122.

Supporting Information Available

All code and data can be found at https://github.com/iktos/generation-under-synthetic-constraint.



(a) Molecules around Chembl dataset.



(b) Molecules around PI3K/mTOR dataset.

Figure 12: Example of molecules with a bad SA score (>3.5) but a good RScore (>0.4).

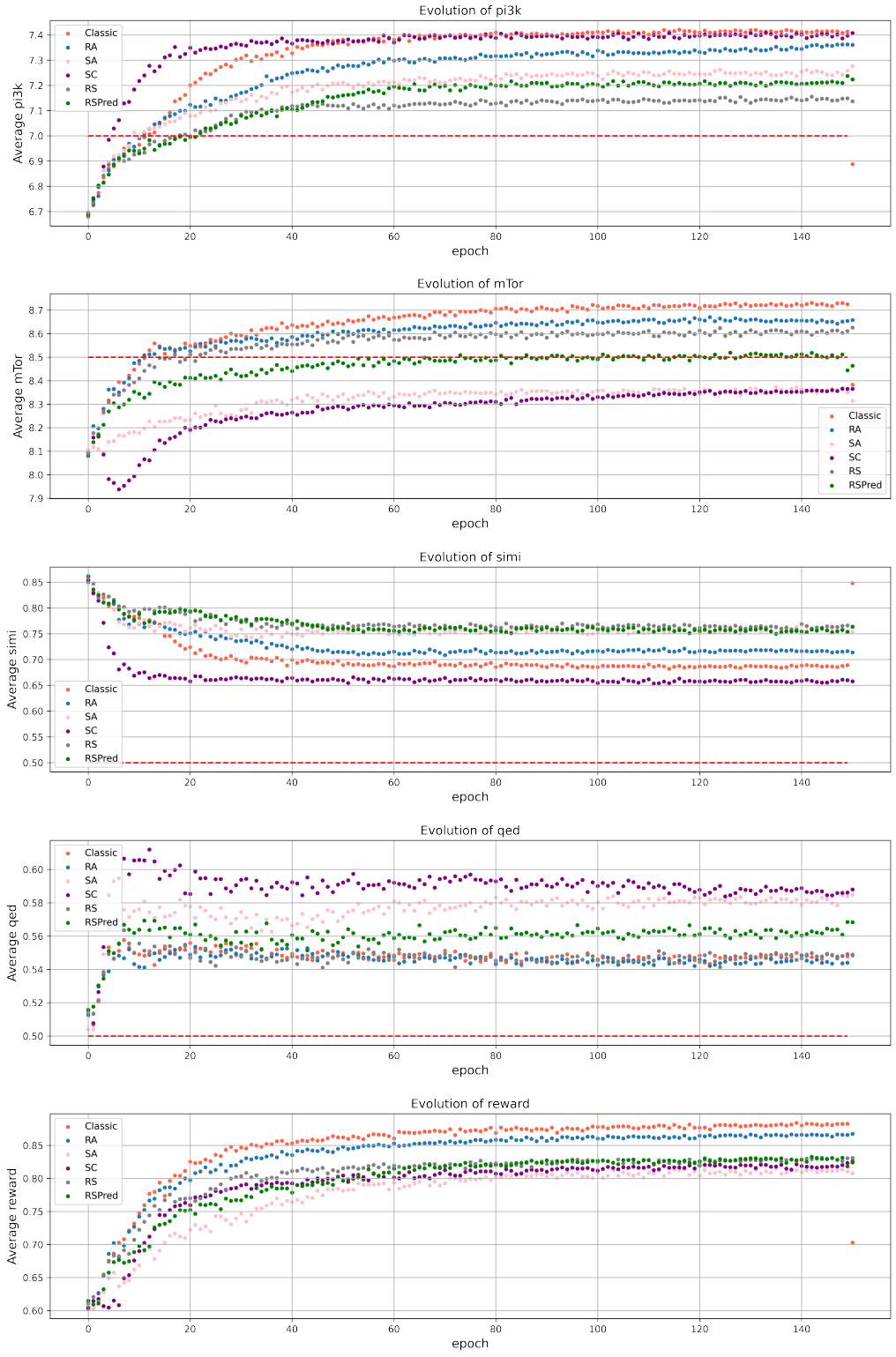


Figure 13: Evolution of 4 scoring functions : PI3K, mTOR, similarity, QED among epochs for 6 different generations around PI3K/mTOR dataset.

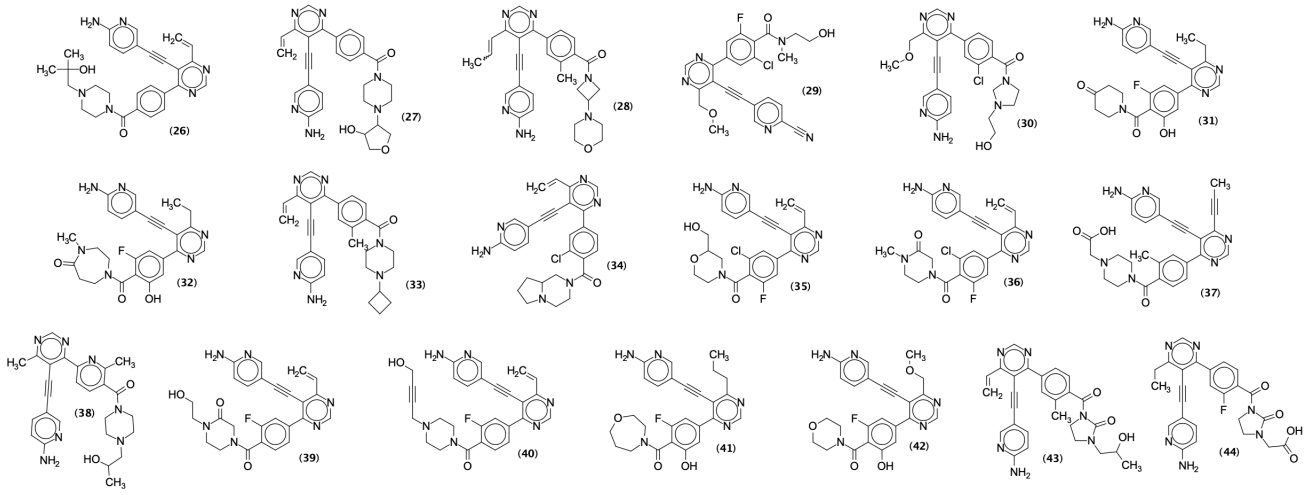


Figure 14: Molecules generated during RScore constrained generation.

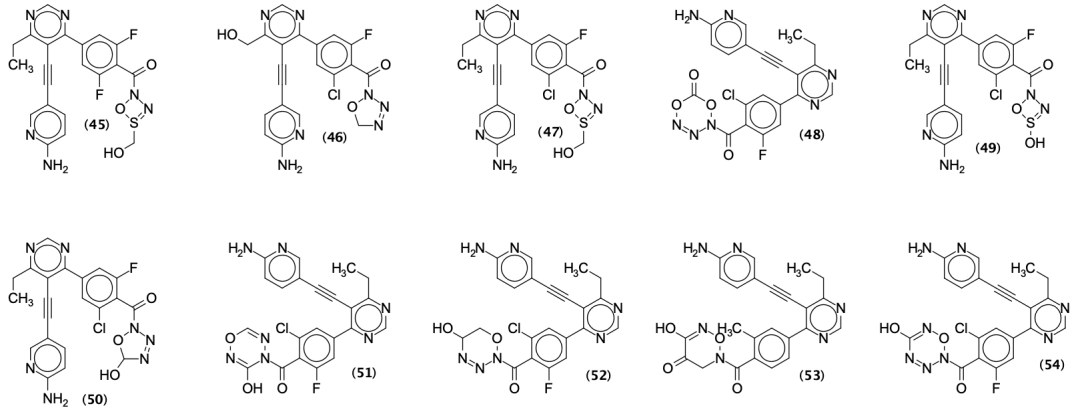


Figure 15: Top 10 molecules from PI3K/mTOR generation without any synthetic constraint.

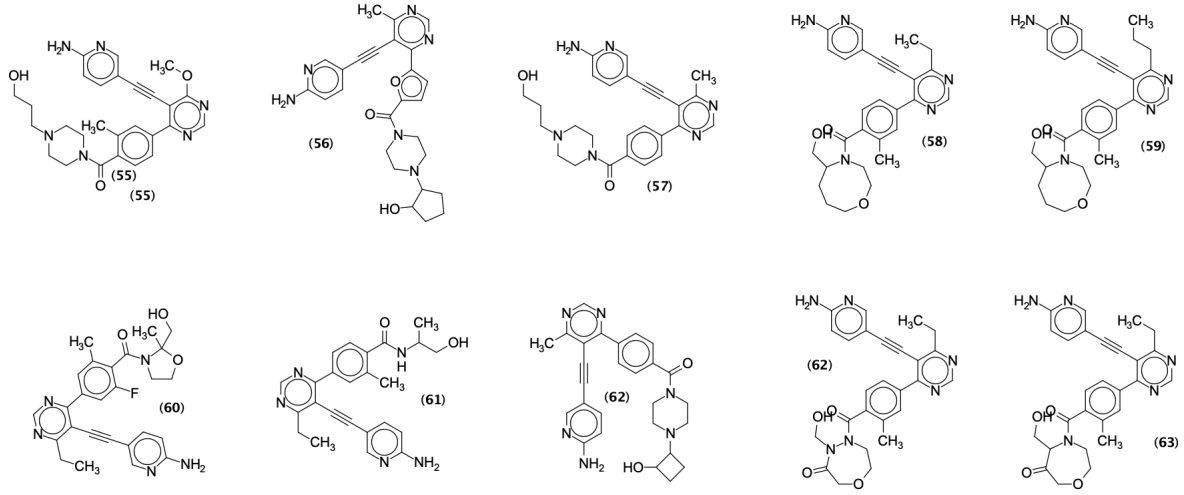


Figure 16: Top 10 molecules from PI3K/mTOR RA score constrained generation.

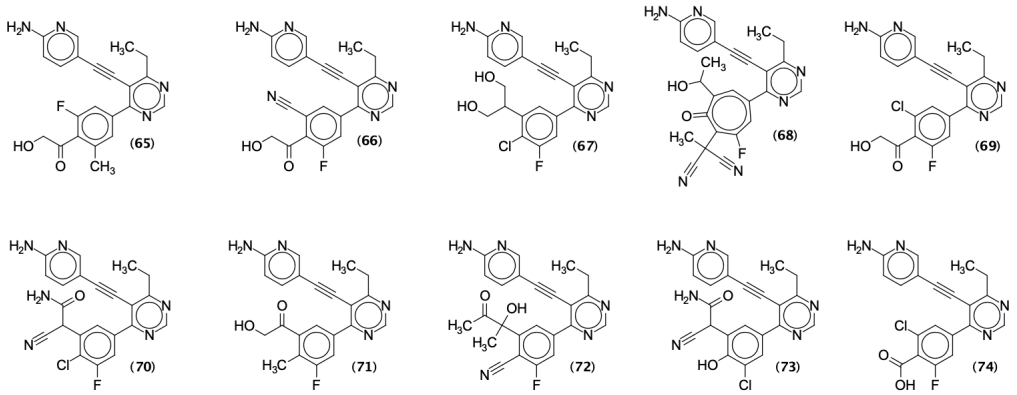


Figure 17: Top 10 molecules from PI3K/mTOR SC score constrained generation.

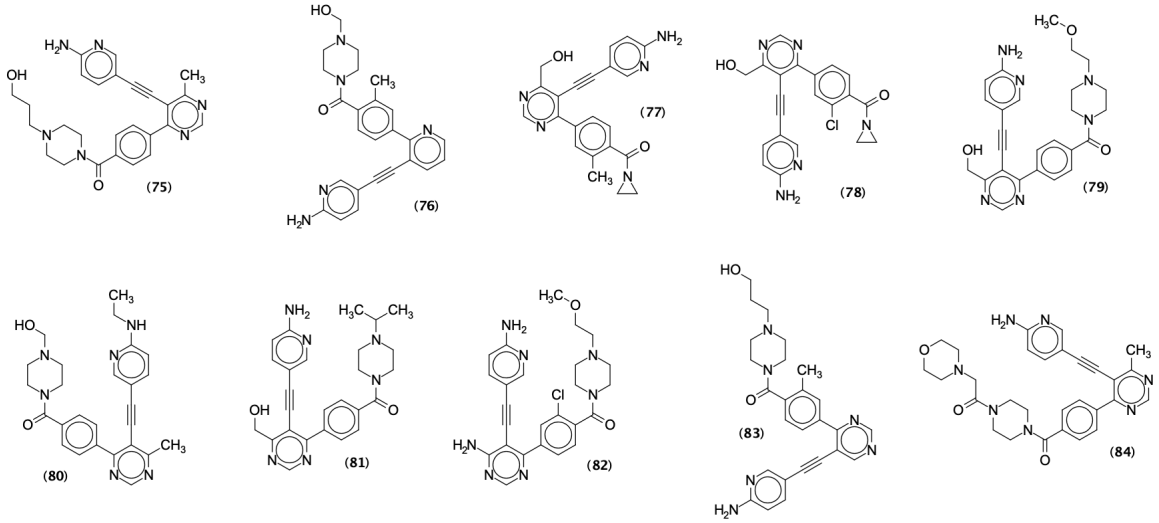


Figure 18: Top 10 molecules from PI3K/mTOR SA score constrained generation.

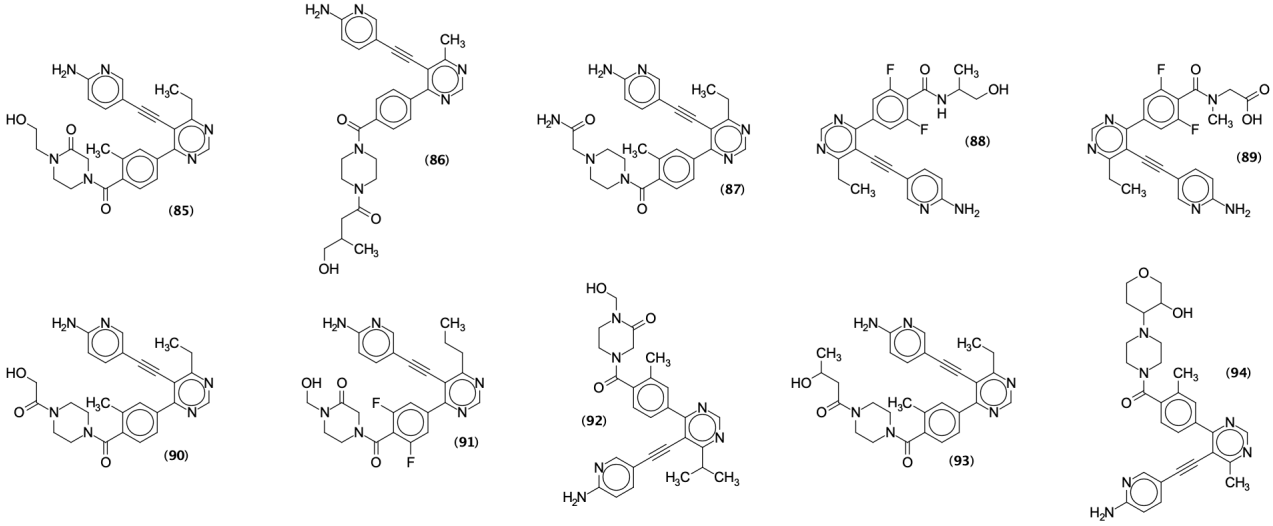


Figure 19: Top 10 molecules from PI3K/mTOR RScore constrained generation.

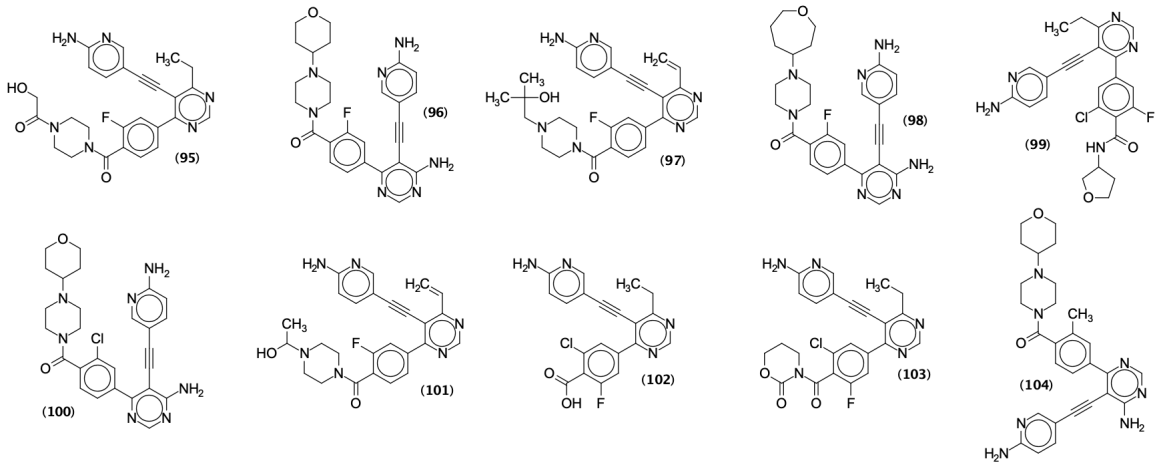


Figure 20: Top 10 molecules from PI3K/mTOR RSPred constrained generation.