Updated 12/03/2020

Supporting Information

Automatic Retrosynthetic Pathway Planning Using Template-Free Models

Kangjie Lin,‡[a] Youjun Xu,‡[a] Jianfeng Pei*[b] and Luhua Lai*[ab]

E-mail: lhlai@pku.edu.cn; jfpei@pku.edu.cn

[a]*BNLMS, Peking-Tsinghua Center for Life Sciences at the College of Chemistry and Molecular Engineering, Peking University, Beijing, 100871, PR China*

[b]*Center for Quantitative Biology, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, 100871, PR China*

‡*Authors contributed equally.*

S1 Key Hyperparameters setting of Transformer architecture

```
hparams.hidden_size = 512
hparams.batch_size = 4096
hparams.max_length = 256
hparams.clip_grad_norm = 0
hparams.optimizer_adam_epsilon = 1e-9
hparams.learning_rate_schedule = "legacy"
hparams.learning_rate_decay_scheme = "noam"
hparams.learning_rate = 0.1
hparams.learning_rate_warmup_steps = 4000
hparams.initializer_gain = 1.0
hparams.num_hidden_layers = 6
hparams.initializer = "uniform_unit_scaling"
hparams.weight_decay = 0.0
hparams.optimizer_adam_beta1 = 0.9
hparams.optimizer_adam_beta2 = 0.98
hparams.num_sampled_classes = 0
hparams.label_smoothing = 0.1
hparams.shared_embedding_and_softmax_weights = True
hparams.symbol_modality_num_shards = 16
hparams.add_hparam("filter_size", 2048)
hparams.add_hparam("num_encoder_layers", 0)
hparams.add_hparam("num_decoder_layers", 0)
hparams.add_hparam("num_heads", 8)
hparams.add_hparam("attention_key_channels", 0)
hparams.add_hparam("attention_value_channels", 0)
hparams.add_hparam("ffn_layer", "dense_relu_dense")
hparams.add_hparam("parameter_attention_key_channels", 0)
hparams.add_hparam("parameter_attention_value_channels", 0)
hparams.add_hparam("attention_dropout", 0.0)
hparams.add_hparam("attention_dropout_broadcast_dims", "")
hparams.add_hparam("relu_dropout", 0.0)
hparams.add_hparam("relu_dropout_broadcast_dims", "")
hparams.add_hparam("pos", "timing")
hparams.add_hparam("nbr_decoder_problems", 1)
hparams.add_hparam("proximity_bias", False)
hparams.add_hparam("causal_decoder_self_attention", True)
hparams.add_hparam("use_pad_remover", True)
hparams.add_hparam("self_attention_type", "dot_product")
hparams.add_hparam("conv_first_kernel", 3)
hparams.add_hparam("attention_variables_3d", False)
hparams.add_hparam("use_target_space_embedding", True)
hparams.add_hparam("moe_overhead_train", 1.0)
hparams.add_hparam("moe_overhead_eval", 2.0)
hparams.moe_num_experts = 16
hparams.moe_loss_coef = 1e-3
hparams.add_hparam("overload_eval_metric_name", "")
hparams.add_hparam("unidirectional_encoder", False)
hparams.add_hparam("hard_attention_k", 0)
hparams.layer_preprocess_sequence = "n"
hparams.layer_postprocess_sequence = "da"
hparams.layer_prepostprocess_dropout = 0.1
hparams.attention_dropout = 0.1
hparams.relu_dropout = 0.1
hparams.learning_rate_warmup_steps = 8000
hparams.learning_rate = 0.2
hparams.optimizer_adam_beta2 = 0.997
hparams.learning_rate_schedule =
        ("constant*linear_warmup*rsqrt_decay*rsqrt_hidden_size")
hparams.learning_rate_constant = 2.0
```

S2 Additional Methods

S2.1 Reaction Classification Algorithm and Results

We took the 80% reactions equally distributed among 10 different reaction classes of USPTO_50K dataset to train a logistic regression (LR) classification model using transformation FP AP3(folded) + Morgan2 FP(agents, folded) as inputs. Below are the results for 10,000 test set reactions equally distributed among 10 different reaction classes. Then we use this LR model to predict the reaction class of reactions in USPTO_MIT dataset.

ID	recall	prec	F-score	reaction class
0	0.9660	0.9593	0.9626	1 Heteroatom alkylation and arylation
1	0.9362	1.0000	0.9670	10 Functional group addition (FGA)
2	0.9773	0.9691	0.9732	2 Acylation and related processes
3	0.9602	0.9645	0.9623	3 C-C bond formation
4	0.8516	0.9281	0.8883	4 Heterocycle formation
5	0.8963	0.9680	0.9308	5 Protections
6	0.9909	0.9831	0.9870	6 Deprotections
7	0.9534	0.9575	0.9554	7 Reductions
8	0.9818	0.9759	0.9789	8 Oxidations
9	0.8591	0.9031	0.8806	9 Functional group interconversion (FGI)
Mean:	0.94	0.96	0.95	

S2.2 Token- and Character-based Preprocessing

Token-based preprocessing splits the reactants and products as tokens using the following regular expression

```
token_regex = "(\[[^\]]+]|Br?|Cl?|N|O|S|P|F|I|b|c|n|o|s|p|\(|\)|\.|= |# |-|\+|\\\\|\/|:|~|@|\?|>|\*|\$|\%[0-9]{2}|[0-9])"
```

Character-based preprocessing splits the reactants and products as characters. Here is an example of the representations of token- and character-based preprocessing.

token: N c 1 n c 2 [nH] c (C C C c 3 c s c (C (= O) O) c 3) c c 2 c (= O) [nH] 1

character: N c 1 n c 2 [n H] c (C C C c 3 c s c (C (= O) O) c 3) c c 2 c (= O) [n H] 1

S2.3 Model architecture

The Transformer architecture follows an encoder-decoder structure using stacked self-attention and point-wise fully connected layers.

The encoder maps an input symbol sequence $(x_1, \ldots, x_n)$ to continuous representations $z = (z_1, \ldots, z_n)$. Given z, the decoder then generates an output symbol sequence $(y_1, \ldots, y_m)$. The encoder and decoder are composed of a stack of N identical layers, each of which contains three sub-modules. The first sub-module is a multi-head self-attention mechanism, which is made of several scaled dot-product attention layers running in parallel. In this attention layer, the input consists of queries (Q), keys (K) of d_k dimension, and values (V) of d_v dimension. The formula of a single attention function is

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

This attention function will yield d_v-dimensional output values. These values are concatenated into the multi-head attention layer using the following formula:

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, \ldots, \mathrm{head_h})W^O$$

where $\mathrm{head_i} = \mathrm{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$ and the trainable parameter matrices

$(W_i^Q \in \mathbb{R}^{d_{\mathrm{model}} \times d_k}, W_i^K \in \mathbb{R}^{d_{\mathrm{model}} \times d_k}, W_i^V \in \mathbb{R}^{d_{\mathrm{model}} \times d_v}$, and $W^O \in \mathbb{R}^{d_{\mathrm{model}} \times hd_k})$

are the linear projections. The number of parallel attention layers or heads is h. For each of these heads, $d_k = d_v = d_{\mathrm{model}}/h$.

The second sub-module is a fully connected feed-forward network, which is applied to each position separately and identically. The transformation function is as follows:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

where W_1, W_2 and b_1, b_2 are learnable weights and biases, respectively.

The third sub-module in the decoder stack is a modified self-attention layer that uses a masking operation to prevent positions from attending to subsequent positions, which ensures that predictions at position i can be only up to the known outputs at position $< i$.

Similar to other sequence models, the embedding layer is added to convert the input tokens and output tokens to d_{model}-dimension vectors. To preserve the order of the sequence, the positional encoding operation is combined into the input embeddings. These encodings have the same dimension d_{model} as the embeddings, so that the two can be summed. The sine and cosine functions of different frequencies are used in the positional encoding as follows:

$$PE_{(pos,2i)} = sin\left(pos/10000^{2i/d_{\mathrm{model}}}\right)$$

$$PE_{(pos,2i+1)} = cos\left(pos/10000^{2i/d_{\mathrm{model}}}\right)$$

where pos is the position and i is the dimension.

S2.4 Reward in MCTS

If a state is unsolved during rollout, it will receive a reward of -1. If a state exists, its reward is calculated by the following steps: All scaffolds of the target molecule and all the starting molecules in building block database are calculated; Then the maximum common subscaffold (MaxSubscaffold) of the target and starting molecules is calculated; Finally, a score is defined as the similarity between the MaxSubscaffold and candidate terminal molecule, and a reward (between 0 and 1) will be given according to the score.

The formulas are shown below:
(1) MaxSubscaffold=*MaxCommonSubScaffold*(target molecule, building blocks)
(2) score=*DiceSimilarity*(MaxSubscaffold, candidate terminal molecule)
(3) reward=(a*score)/(1.0+a*score), where a is a parameter between 0 and 1.

S3 Additional Results

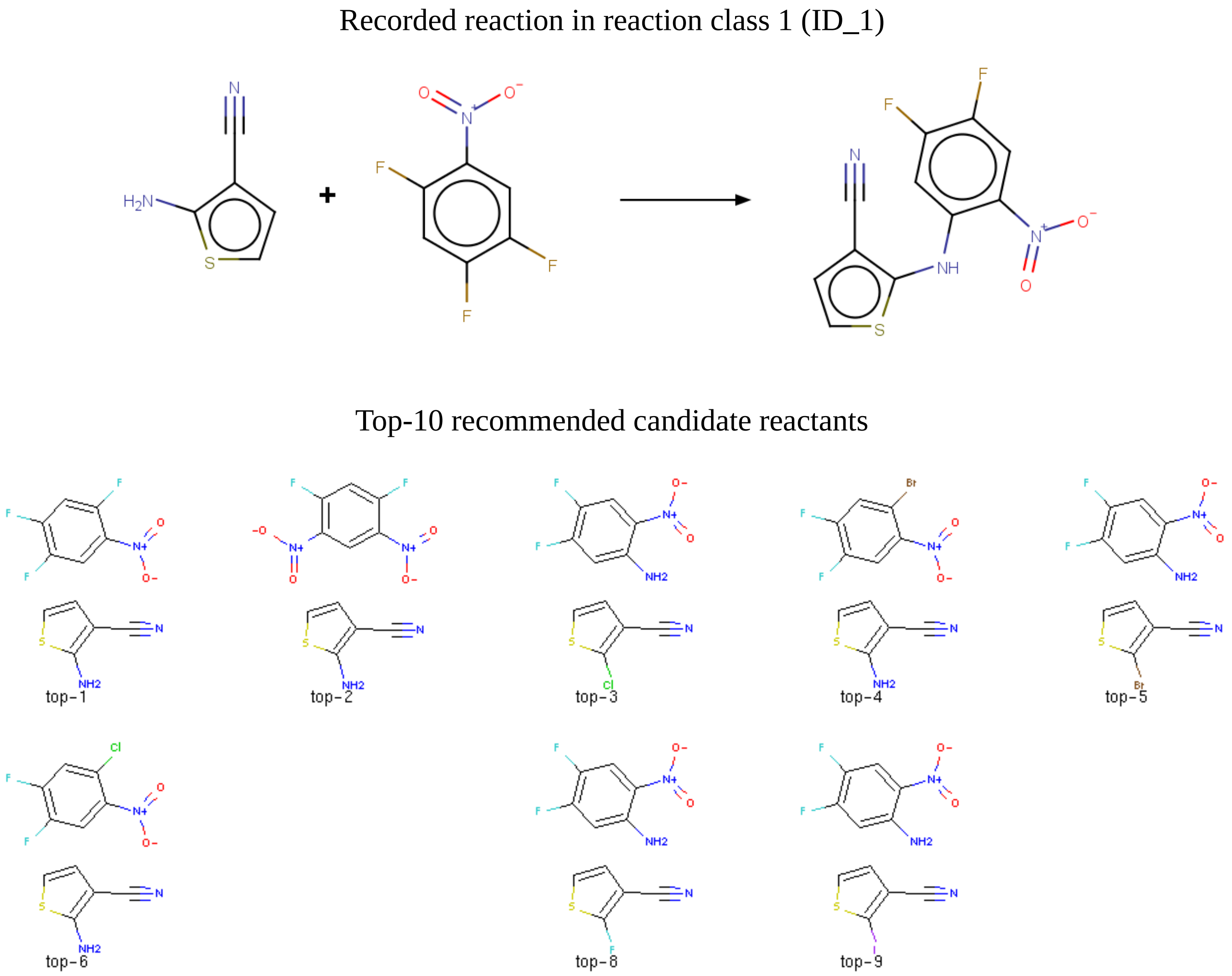
S3.1 Single-Step Retrosynthetic Top-10 Accuracy Within Each Reaction Class

Table S1. Model top-10 accuracy for 10 reaction classes

	Reaction class, top-10 accuracy (%)									
model (dataset)	1	2	3	4	5	6	7	8	9	10
Liu et al. template+class (USPTO_50K)[1]	77.2	84.9	53.4	54.4	6.2	26.9	74.7	68.4	46.7	73.9
Liu et al. LSTM+class (USPTO_50K)[1]	57.5	74.6	46.1	27.8	80.0	62.8	67.8	69.1	47.3	56.5
Coley et al. Similarity+class (USPTO_50K)[2]	86.7	94.2	74.6	67.0	97.1	95.5	88.3	98.8	71.2	91.3
Our Transformer+char+class (USPTO_50K)	83.1	90.4	76.2	60.0	92.3	88.6	88.2	86.4	73.9	82.6
Our Transformer+char+class (USPTO_MIT)	88.2	91.2	81.9	67.8	75.4	86.6	87.1	88.5	73.5	66.7

Key: “+class” means that reaction class information was provided to the model; “+char” means that char-based preprocessing was applied.

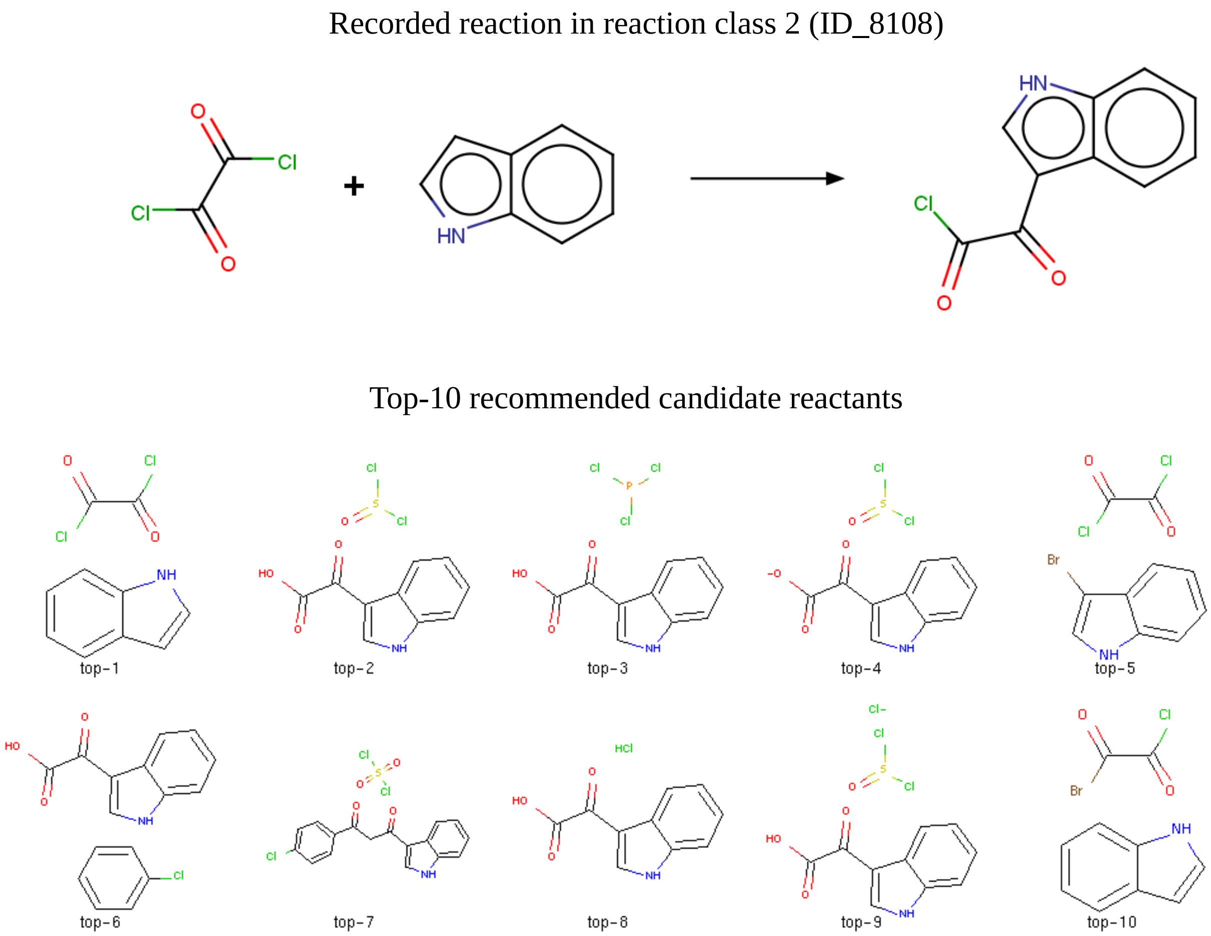
S3.2 Single-Step Retrosynthetic Result Within Each Reaction Class



Recorded reaction in reaction class 1 (ID_1)

Top-10 recommended candidate reactants

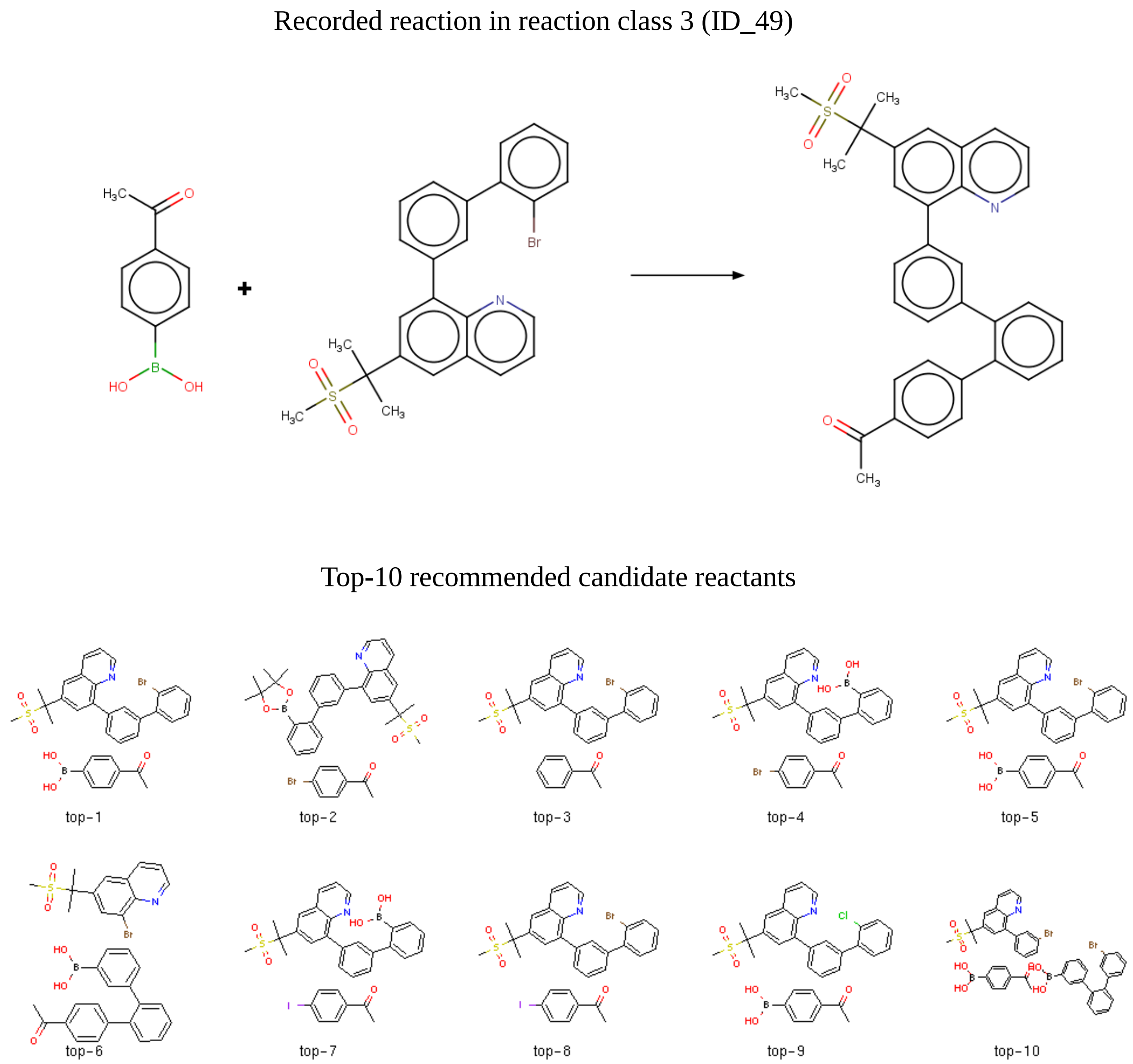
Figure S1: Randomly-selected example from class 1 (heteroatom alkylation and arylation). Our approach proposes the recorded reactants with rank 1. The results containing invalid SMILES are left blank.



Recorded reaction in reaction class 2 (ID_8108)

Top-10 recommended candidate reactants

Figure S2: Randomly-selected example from class 2 (acylation and related processes). Our approach proposes the recorded reactants with rank 1. The results containing invalid SMILES are left blank.



Recorded reaction in reaction class 3 (ID_49)

Top-10 recommended candidate reactants

Figure S3: Randomly-selected example from class 3 (C-C bond formation). Our approach proposes the recorded reactants with rank 1. The results containing invalid SMILES are left blank.

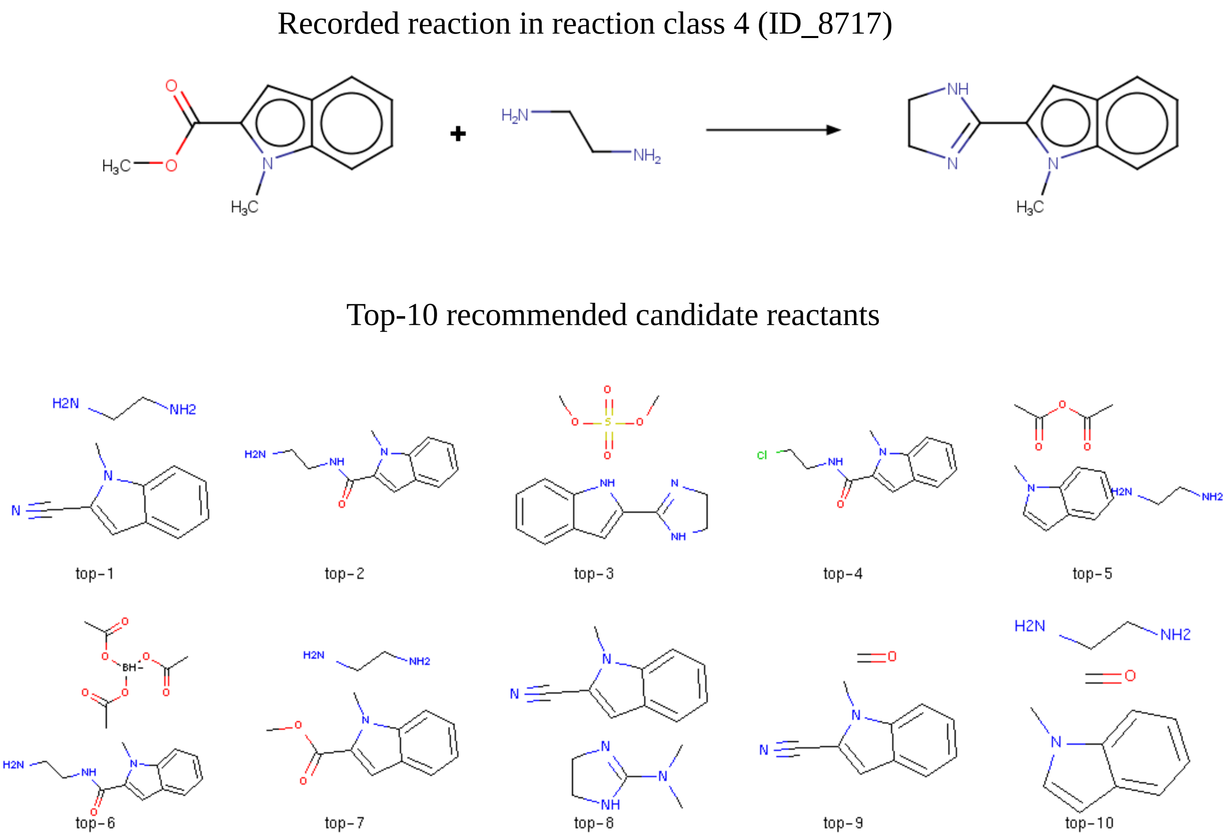
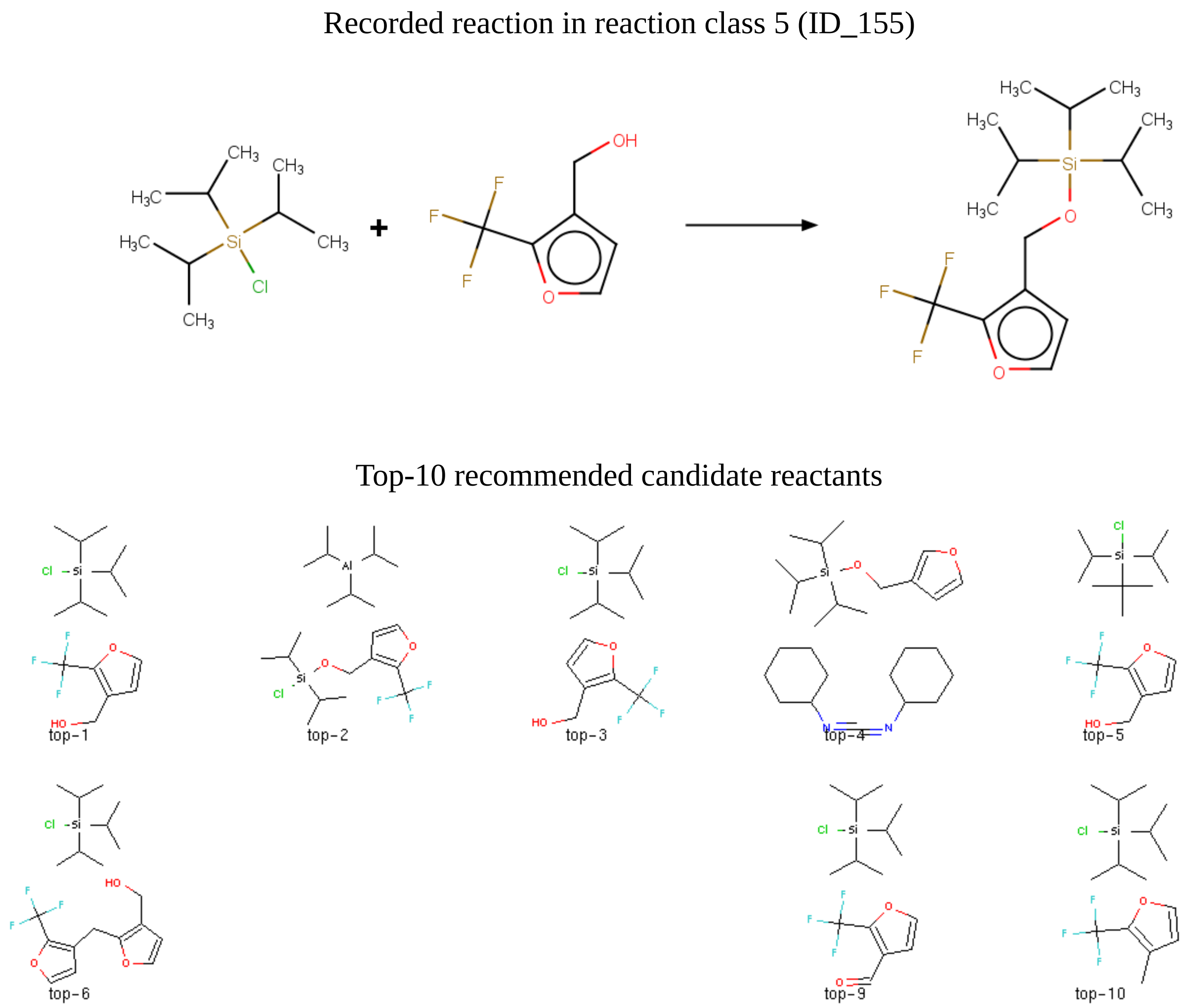

Recorded reaction in reaction class 4 (ID_8717)
Top-10 recommended candidate reactants
top-1
top-2
top-3
top-4
top-5
top-6
top-7
top-8
top-9
top-10

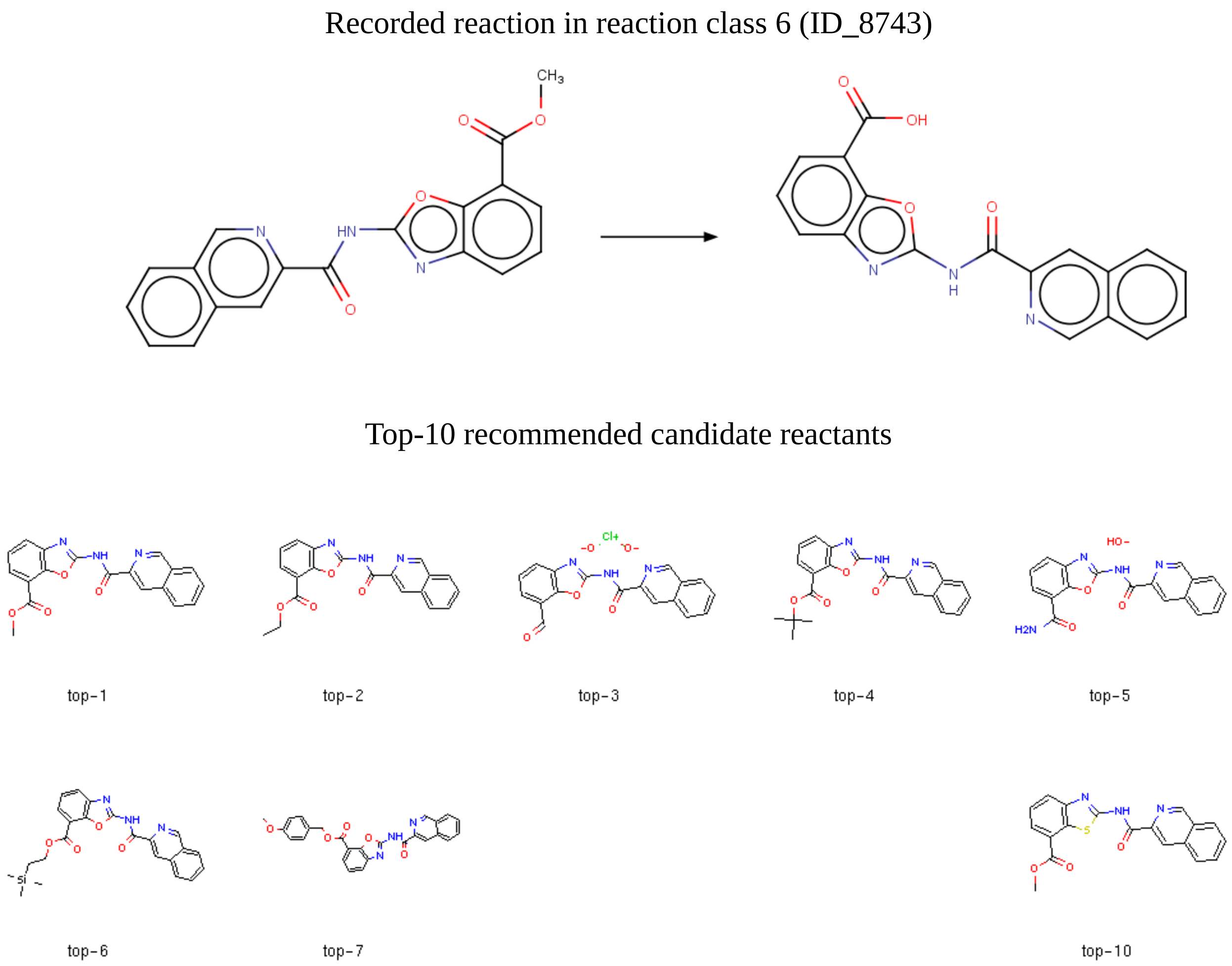
Figure S4: Randomly-selected example from class 4 (heterocycle formation). Our approach proposes the recorded reactants with rank 7. The results containing invalid SMILES are left blank.



Recorded reaction in reaction class 5 (ID_155)

Top-10 recommended candidate reactants

Figure S5: Randomly-selected example from class 5 (protections). Our approach proposes the recorded reactants with rank 1. The results containing invalid SMILES are left blank.



Recorded reaction in reaction class 6 (ID_8743)

Top-10 recommended candidate reactants

Figure S6: Randomly-selected example from class 6 (deprotections). Our approach proposes the recorded reactants with rank 1. The results containing invalid SMILES are left blank.

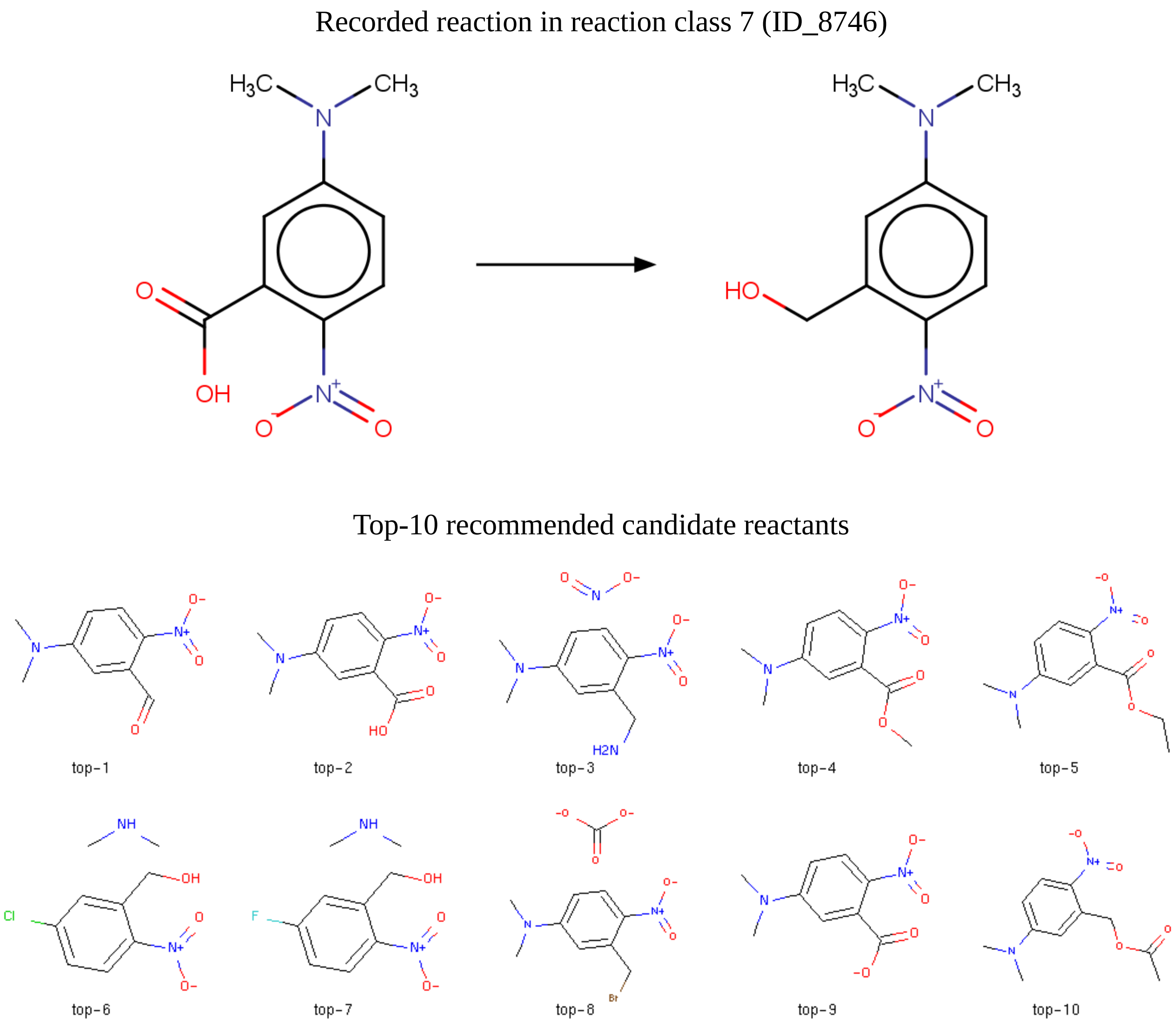
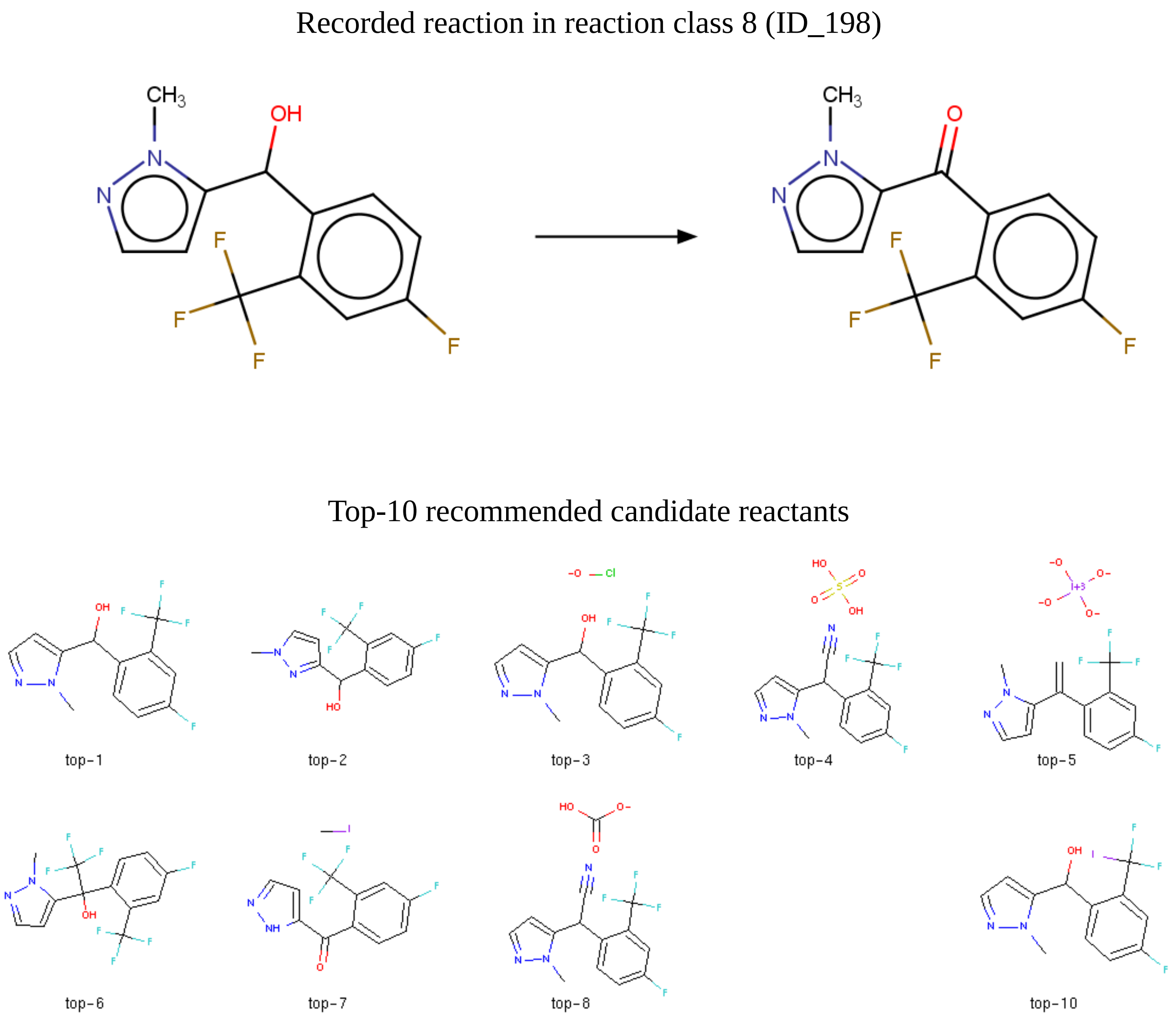

Recorded reaction in reaction class 7 (ID_8746)
Top-10 recommended candidate reactants
top-1
top-2
top-3
top-4
top-5
top-6
top-7
top-8
top-9
top-10

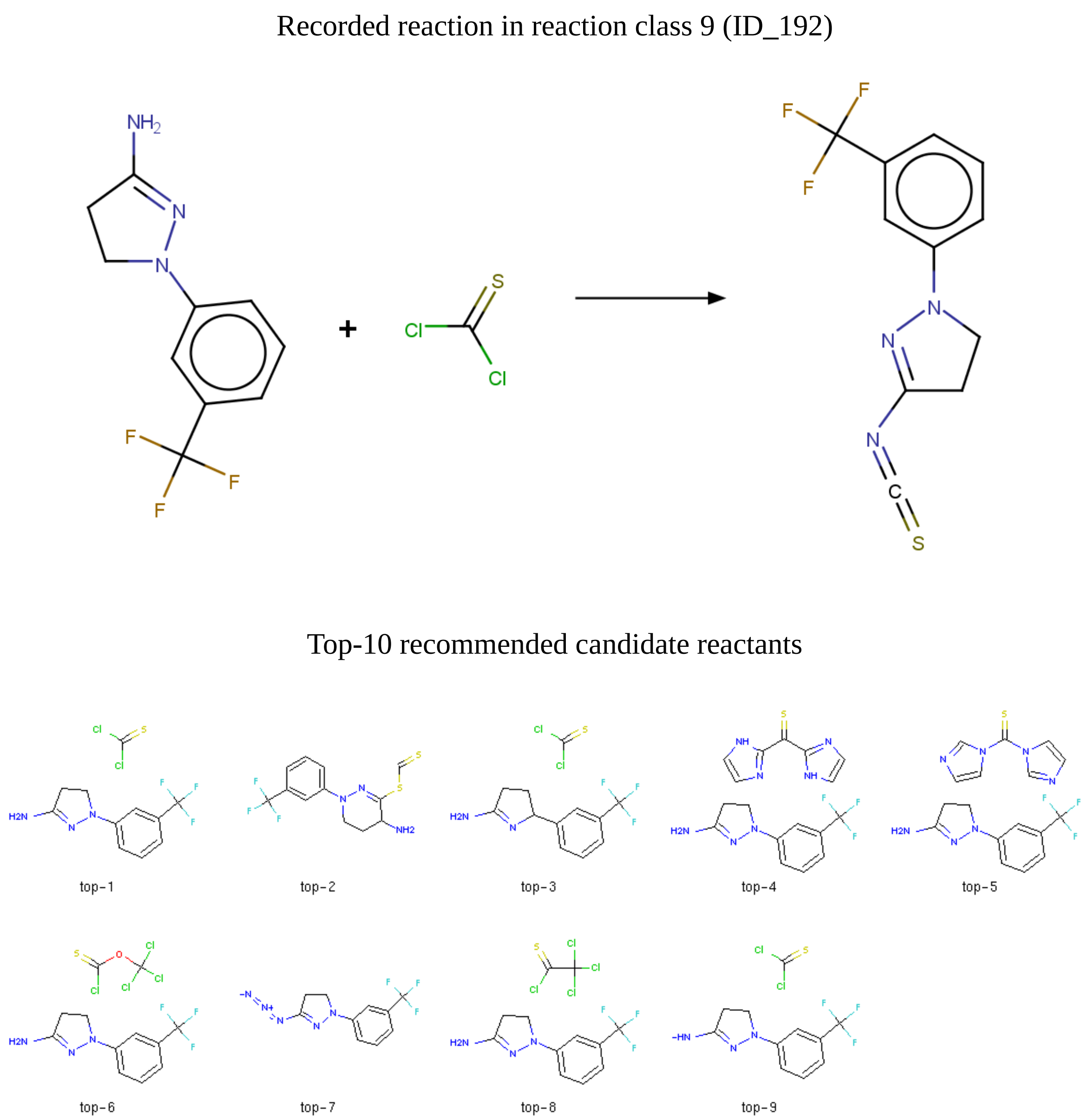
Figure S7: Randomly-selected example from class 7 (reductions). Our approach proposes the recorded reactants with rank 2. The results containing invalid SMILES are left blank.



Recorded reaction in reaction class 8 (ID_198)

Top-10 recommended candidate reactants

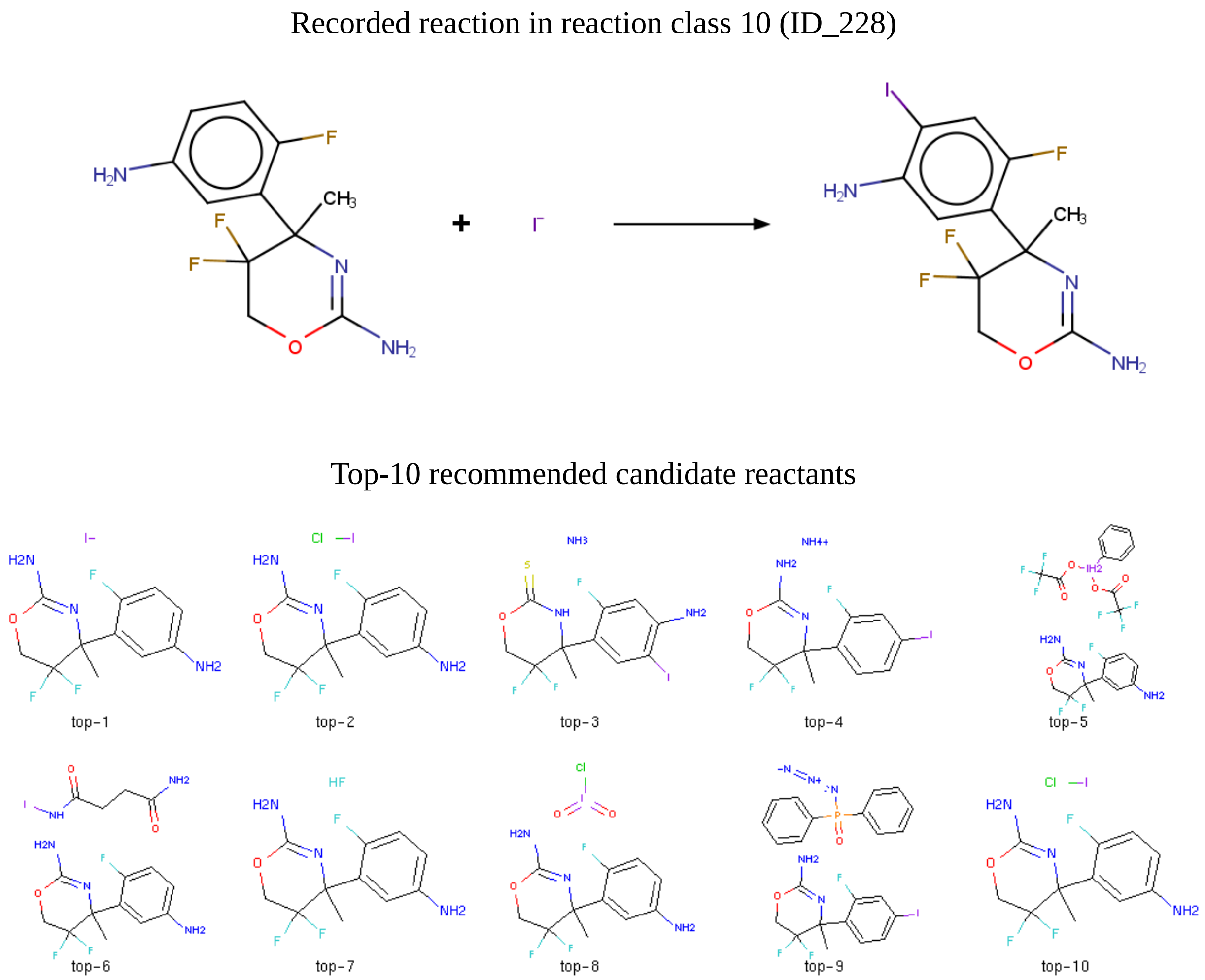
Figure S8: Randomly-selected example from class 8 (oxidations). Our approach proposes the recorded reactants with rank 1. The results containing invalid SMILES are left blank.



Recorded reaction in reaction class 9 (ID_192)

Top-10 recommended candidate reactants

Figure S9: Randomly-selected example from class 9 (functional group interconversion). Our approach proposes the recorded reactants with rank 1. The results containing invalid SMILES are left blank.



Recorded reaction in reaction class 10 (ID_228)

Top-10 recommended candidate reactants

Figure S10: Randomly-selected example from class 10 (functional group addition). Our approach proposes the recorded reactants with rank 1. The results containing invalid SMILES are left blank.

S3.3 Reproduction of baseline experiments

(i) For Liu et al.'s seq2seq model[1], we evaluate their model by retraining their open-source code using the USPTO_MIT dataset. We use the same hyper parameters of theirs during training.

(ii) For Segler et al.'s Neural Symbolic approach, we evaluate their approach by retraining the Coley's reproduced code (https://github.com/connorcoley/retrotemp). In the case of USPTO_50K, since our raw data comes from Liu et al.'s work[1] and does not have atom mapping, we used the dataset with atom mapping that have been used by Coley et al.[2] for Segler et al's approach. The two datasets both originated from the dataset used by Schneider et al.[3] We follow the same data split like Coley et al. In the case of USPTO_MIT, we use the our original data split. Using Coley et al.'s template extraction method[2] in both USPTO_50K and USPTO_MIT, we have fine-tuned the hyper parameters and change the model architecture of Coley's reproduced model as FC(1024)-elu-dropout (keep_prob = 0.9), and the final results are shown in Table 3.

S3.4 Automatic Retrosynthetic Pathway Planning Within Four Examples

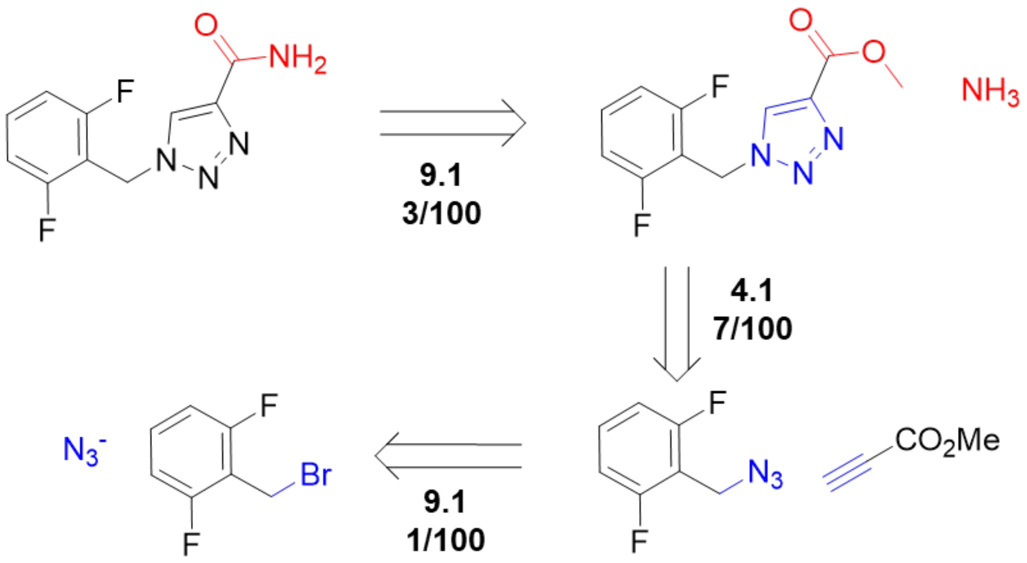


Figure S11: Automatic retrosynthetic pathway planning of Rufinamide. Routes are constructed by automatic searching via MCTS coupled with heuristic scoring function. The suggested disconnections are consistent with published pathways.

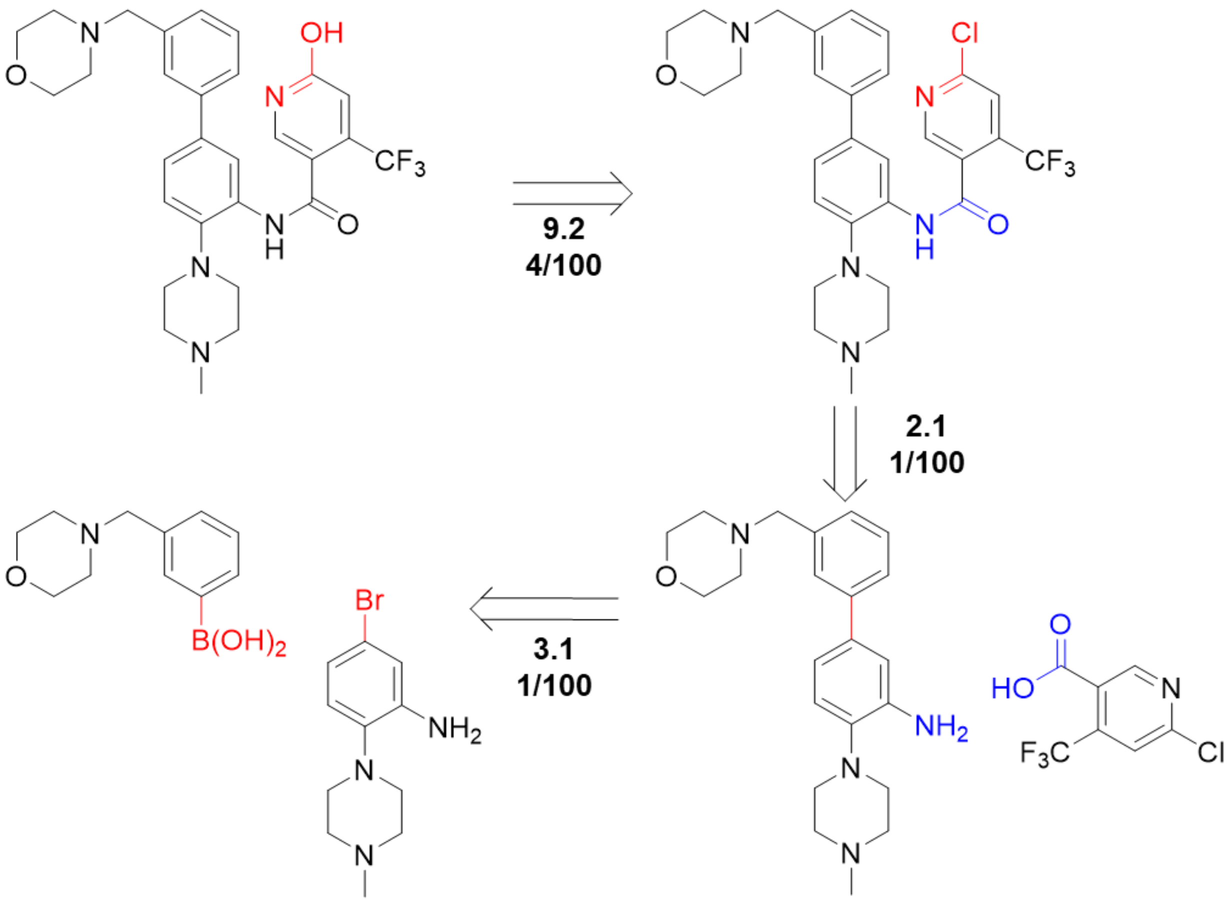


Figure S12. Automatic retrosynthetic pathway planning of an antagonist of the interaction between WDR5 and MLL1, from the examples of Grzybowski et al. Routes are constructed by automatic searching via MCTS coupled with heuristic scoring function. The suggested disconnections are consistent with published pathways.

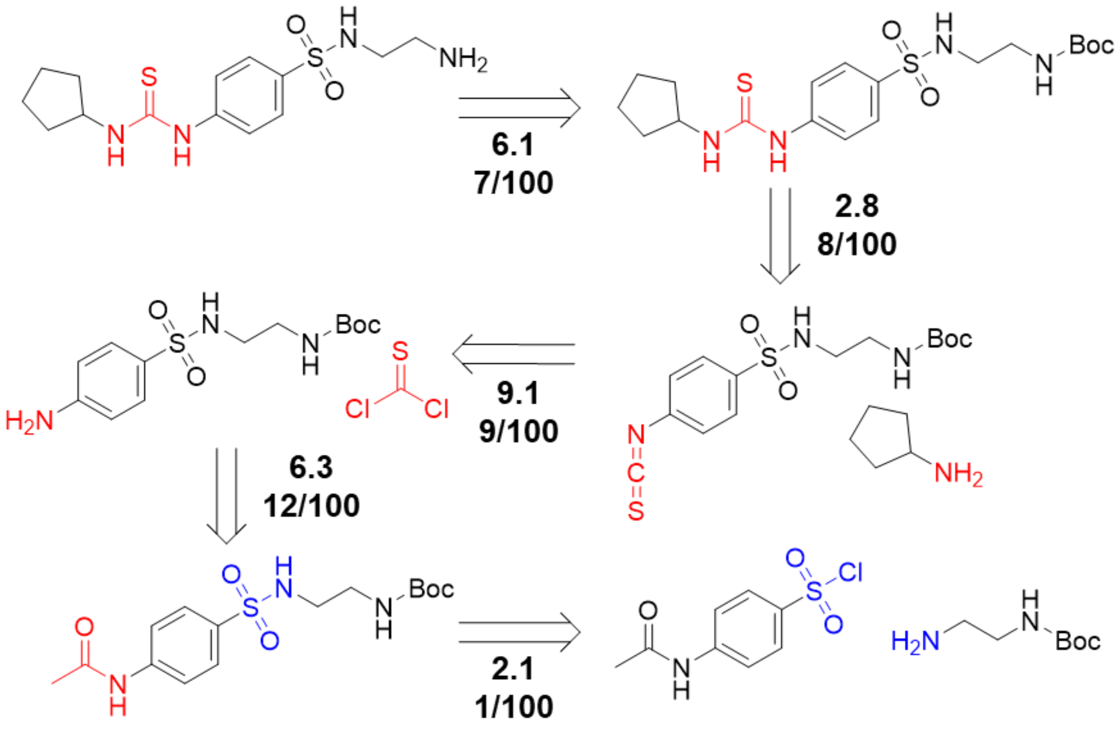


Figure S13. Automatic retrosynthetic pathway planning of an allosteric activator for GPX4. Routes are constructed by automatic searching via MCTS coupled with heuristic scoring function. The suggested disconnections are consistent with published pathways.

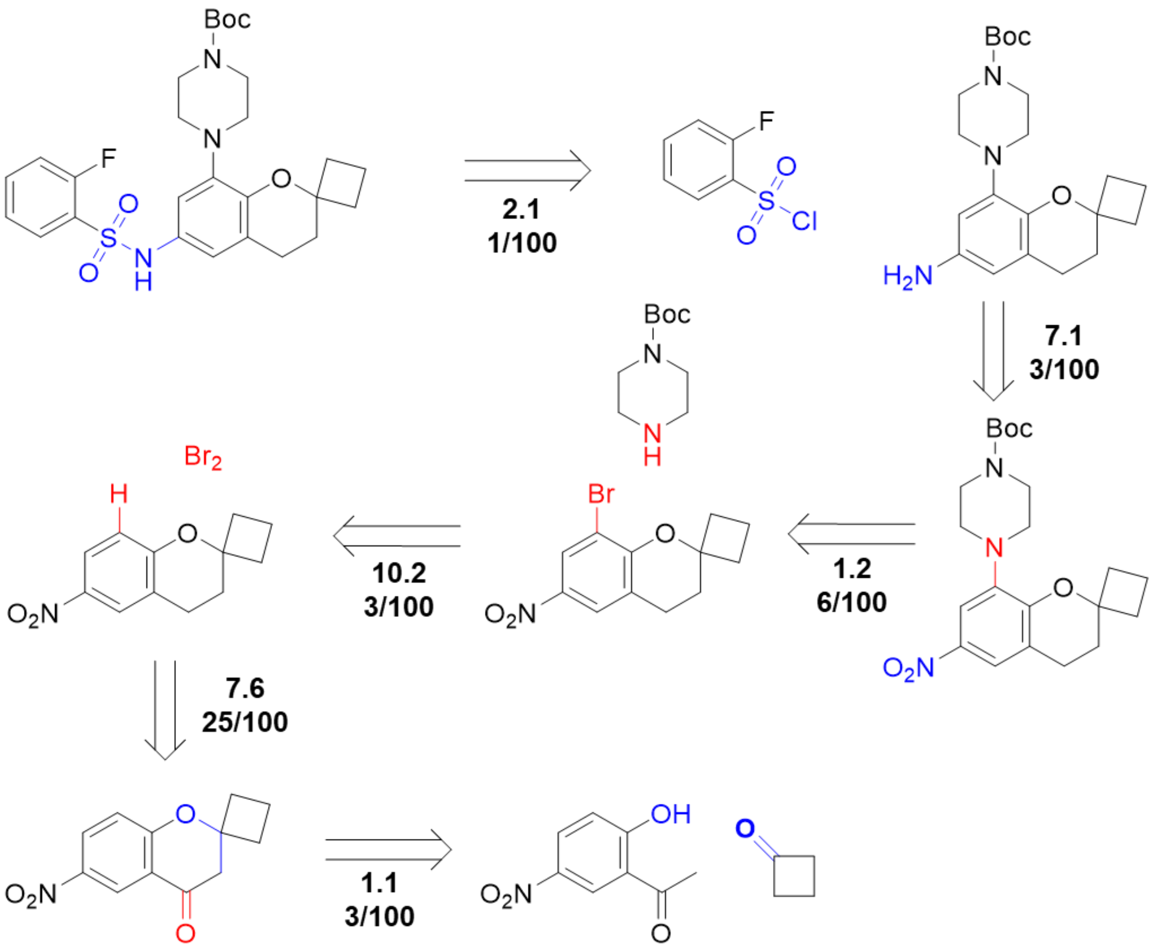


Figure S14. Automatic retrosynthetic pathway planning of an intermediate of drug candidate from the examples of Segler et al. Routes are constructed by automatic searching via MCTS coupled with heuristic scoring function. The suggested disconnections are consistent with published pathways.

References

1. B. Liu, B. Ramsundar, P. Kawthekar, J. Shi, J. Gomes, Q. Luu Nguyen, S. Ho, J. Sloane, P. Wender and V. Pande, *ACS Central Science*, 2017, **3**, 1103-1113.
2. C. W. Coley, L. Rogers, W. H. Green and K. F. Jensen, *ACS Central Science*, 2017, **3**, 1237-1245.
3. N. Schneider, N. Stiefl, G. A. Landrum, *J. Chem. Inf. Model.* 2016, **56**, 2336−2346.