Hard-threshold-Neural-Network based Prediction of Organic Synthetic Outcomes

Haoyang Hu
Tsinghua University
Zhihong Yuan (zhihongyuan@mail.tsinghua.edu.cn)
Tsinghua University

Research article

Keywords: Medicine development, Retrosynthetic analysis, Outcome prediction, Hard-threshold neural network, Combinatorial optimization

Posted Date: November 2nd, 2019

DOI: https://doi.org/10.21203/rs.2.16734/v1

Abstract

Retrosynthetic analysis is the canonical technique to plan the synthesis route of organic molecules in medicine development. In this technique, the screening of synthetic tree branches requires accurate forward reaction prediction, but existing software is still far from completing this step independently. Previous studies have attempted to apply neural network in the forward reaction prediction, but the accuracy is not satisfying. Through using the Edit-based Description and Extended-Connectivity Fingerprints to transform reaction into vector, the presented work focuses on the update of neural network to improve the template-based forward reaction prediction. Hard-threshold activation and target propagation algorithm are implemented by introducing the mixed-convex combinatorial optimization. Comparative tests are conducted to explore the optimal hyperparameter set. Using 15 000 experimental reaction records from granted United States patents, the proposed hard-threshold neural network is systematically trained and tested. The results demonstrate that a higher prediction accuracy is obtained when compared to the traditional neural network with backpropagation algorithm. Indeed, the prediction accuracy of the proposed hard-threshold neural network can reach 73.9% which is higher than Coley's result with 71.8% (Coley et al. ACS Cent. Sci, 2017). Some successfully predicted reaction examples are also briefly discussed.

Background

Medicine development is one of the most important part of the pharmaceutical industry. In recent years, the structure of new medical molecules has been becoming more and more complicated, which dramatically pushed the pharma R&D activities, prolonged discovery-development-deployment cycle, raised the overall cost, and finally lowered the growth rate of the global pharmaceutical market[1]. One of the central points is the synthesis planning, which traditionally faces many challenges, such as low efficiency, poor repeatability and high experimental cost. Therefore, it is imperative to introduce computer-aided synthesis planning for organic products development[1]. Although computer-aided synthesis planning has a long history[1], most existing software have not reached a proper level of automation yet, and still requires chemists to manually edit rules[1]. On the other hand, machine learning and artificial intelligence technologies are in the ascendant nowadays[1]. A variety of machine learning and artificial intelligence methods such as random forest trees, automated reasoning, support vector machines and more recently deep learning are demonstrating their utility for organic molecules discovery, design, and production[1]. Needless to say, the application of the aforementioned new technologies to end-to-end organic molecules discovery and development will be expected to greatly improve the availability of related software, and finally realize fully automatic synthetic synthesis planning.

Retrosynthetic analysis is the canonical technique used to plan the synthesis of small organic molecules for drug discovery. Generally speaking, retrosynthesis analysis consists of four steps: step 1. determine the target compound; step 2. disconnect certain bonds which are thought to be easy to form according to known chemical knowledge in the target compound as the reverse of reaction, and search for possible precursors in this way; step 3. repeat step 2 for all the precursors to form a synthesis tree, and expand it until all precursors are available; step 4. evaluate all the branches of the synthetic tree one by one, then take the most possible one as the optimal route. Step 4 is very critical and difficult among all four steps, since there may be multiple different groups of reactive sites in the same group of precursors, which indicates that there can be difference between real reactions and the expected ones in the branches. Therefore, it is necessary to directly judge the main product based on the reactants in order to evaluate each branch correctly, w.r.t forward reaction prediction. Since there are usually not enough experimental records of properties and reactivities for many precursors, forward reaction prediction cannot be solved by simply searching the chemical knowledge database, but still relies on human assistance at present. Therefore, the presented work attempts to solve this problem by applying machine learning technology to assist forward reaction prediction.

Current research in forward reaction prediction based on machine learning has two main directions: template-based method and template-free method. For template-based method, Coley et al.[11] applied reaction templates as many as possible to the reactant to be predicted to generate a large number of candidate reactions, then trained a neural network to screen the

candidate list. A prediction accuracy with 71.8% was obtained via the above method. For template-free method, Schwaller et al.[12] compared chemical reactions from reactants to products to translations from one language to another, so that forward reaction prediction can be transformed into machine translation and solved by training a seq-to-seq recurrent neural network which directly takes reactant SMILES as input. They achieved a prediction accuracy with 65.4% on a large dataset.

In this work, template-based method is adopted for three reasons: 1. this method has achieved higher prediction accuracy till now; 2. the research target is not to discover new reactions, so it will be difficult to deal with possible results beyond known chemical knowledge in template-free method; 3. reaction rules summarized by experimental results should be fully utilized.

Since the published literatures only used an original fully connected neural network without giving any explanation on how they choose hyperparameter, some room for optimization still exists. Our work will test various optimization methods such as hard-threshold neural network to improve the template-based forward reaction prediction. In detail, hard-threshold activation and target propagation algorithm are implemented by introducing the mixed-convex combinatorial optimization. Comparative tests are conducted to explore the optimal hyperparameter set.

Forward Reaction Prediction

The whole process of template-based forward reaction prediction can be summarized by Fig.1.

Given a group of reactants to predict, templates in the known popular template set are applied to it as many as possible to generate many candidate reactions. Then candidate reactions are converted into vectors, from which the most likely one to occur is selected as the prediction result. The core of this work is to replace the traditional neural network applied in the selection step with hard-threshold neural network to improve the prediction accuracy.

Data augmentation strategy

Considering that existing chemical knowledge database only contains real reactions which can take place in practice, in order to train the neural network to identify the real response, it is necessary to adopt a data augmentation strategy. Specifically, real reactions are first transformed into SMILES, then extracted by a heuristic algorithm to form the template set in SMARTS format. Next, we remove low-frequency ones from this set and re-apply all feasible popular templates to each group of reactants in real reactions to generate a large amount of false reactions which cannot take place in practice. Finally, augmented reaction dataset including real reactions labeled as positive examples along with fake reactions labeled as negative examples are provided to the neural network.

Vectorized description of Reaction

Neural network requires vector format input, so reactions must be converted to vector in an appropriate way. Obviously, the strategy of choosing features to construct the vector will have a significant influence on the prediction accuracy.

This work adopts the “Edit Vector” format proposed by Coley et al[11]. In this format, an atom is described as a 32-dimentional feature vector, and a bond is described with a 4-dimentional one. A reaction is first decomposed into four kind of basic Edits including hydrogen loss, hydrogen gain, bond loss and bond gain, and then described as the combination of feature vectors. For example, a hydrogen loss is described as a corresponding atom feature vector (32-D), and a bond loss is described as two

corresponding atom feature vectors and a corresponding bond feature vector (32-D + 4-D + 32-D = 68-D). This format considers both the chemical environment of the reaction core and the whole molecular, and is also easy to calculate, which is conducive to the processing of large-scale dataset.

To exactly explain how Edit Vector describes the reaction, based on the equations expressed by Fig.2, an simplified but general example is shown as followed: the atom feature vector has 6 dimensions [is_carbon, is_nitrogen, is_oxygen, is_chlorine, num_Hs, num_non-H_atoms] and the bond feature vector has 4 dimensions [is_single, is_aromatic, is_double, is_triple].

The above reaction can be decomposed into three Edits: atom 1 loses a hydrogen, atom 2 and 3 loses a bond and atom 1 and 3 gain a bond. Next, compute the feature vectors of the three atoms and the two bonds involved:

[Please see Supplementary Files for formulas]

The Edit Vectors of hydrogen loss and gain are taken as the feature vector of the corresponding atom, and those of the bond loss and gain are taken as the connection of the feature vectors of corresponding atoms and the bonds:

[Please see Supplementary Files for formulas]

Where "/" represents the connection of feature vectors, and the outermost brackets represent that Edit Vector is the combination of all the vectors of basic Edit in a reaction.

These four Edit Vectors can completely specify the reaction "an sp^2 hybridized carbon is aminated by a secondary amine upon loss of chlorine", which stands for the nature of this reaction, and thus can be used as an input to neural network.

Candidate reaction selection

The selection step uses a complex neural network consisting of several subnetworks, as shown in Fig.3. For each candidate reaction, four Edit Vectors mentioned above are calculated and input to four corresponding subnetworks, whose outputs are summed and input to the final integrating subnetwork to produce scalar probability scores. The above steps are repeated for all candidate reactions, and then all the probability scores are normalized by the softmax method to estimate the probability of occurrence of each candidate reaction. Finally, all candidate reactions are sorted by the probability score, and the first one is output as prediction result. The prediction is correct if the result has the same product SMILES as the recorded one, and vice versa.

Besides the Edit Vector based model, a hybrid model that uses both the Edit Vector and Extended-Connectivity Fingerprint (ECFP) [14]is also trained in our work. The only difference between two models is that an extra subnetwork without hidden

layer which evaluates the ECFP is added to the hybrid model. The output of ECFP network is multiplied by ε when subnetwork outputs are summed before input to the final integrating subnetwork, where ε is the mixing factor.

Hard-threshold Neural Network

Neural network, especially deep neural network learning, is currently the most popular machine learning algorithm with powerful fitting capabilities. However, with the ceaseless expansion of the size of the neural network, a series of problems have emerged, in which gradient vanishing and gradient explosion are particularly serious. To get rid of this dilemma, hard threshold neural networks come into being.

Constructing hard-threshold neural network

“Hard-threshold neural network” means neural network with hard-threshold activation, including Step activation (left) and Staircase activation (right), which is shown in Fig.4, where the latter one is the sum of many former ones. Hard-threshold activation has a constant derivative as 0, which can effectively avoid gradient vanishing and gradient explosion. Besides, the scale of output is almost fixed and insensitive to the scale of the input, which helps avoid certain abnormal propagation and simplify the computation. However, the zero derivative of hard-threshold activation also prevents it from being trained with traditional backpropagation.

Target propagation algorithm

As discussed above, a new backpropagation algorithm is required to train a hard-threshold neural network bypassing the zero derivative of hard-threshold activation.

In 2017, based on the “target propagation” concept proposed by LeCun et al.[14], Friesen et al.[15] proposed a new target propagation algorithm named FTPROP-MB. According to Friesen, since perceptron with Step activation is trainable, hard-threshold neural network could also be trainable if it can be decomposed into perceptron. Specifically, a target vector t_d is introduced to represent what the d^{th} layer is supposed to output for all hard-threshold activation layers. After the normal forward propagation procedure, for each layer, FTPROP-MB determines t_d firstly, then introduces a layer loss L_d and uses it to compute gradient just like training perceptron so that weights can be updated.

Now it is an important problem for FTPROP-MB on how to determine t_d. Considering that the output of the hard-threshold activation is a set of discrete values, this problem can be taken as a combinatorial optimization problem, that is, how to optimize t_d regarding overall loss and layer loss, which can be expressed in standard form as followed:

[Please see supplementary files for formulas]

The search space is large and discrete because all the components of t_d is restricted to 0 and 1, which makes it hard for common search algorithms to find the optimal solution in acceptable time complexity. Because layer loss is usually convex, FTPROP-MB heuristically determines the target vector with the sign of negative gradient which can be formulated as:

[Please see supplementary files for formulas]

When layer loss function is convex, the negative partial derivative of L_{d+1} on h_{dj} points to the global minimal of L_{d+1}. Let's take $h_{dj}=-1$ as an example, if $r(h_{dj})=-1$, which indicate that the partial derivative of L_{d+1} is positive, it is clear that L_{d+1} will increase by making $h_{dj}=+1$ when fixing other components of $\boldsymbol{h_d}$, so there is no doubt that $t_{dj}=r(h_{dj})=-1$. On the other hand, when $r(h_{dj})=+1$, which means the partial derivative of L_{d+1} is negative, we don't know exactly whether L_{d+1} would increase or decrease by making $h_{dj}=+1$. But the difference between h_{dj} and $r(h_{dj})$ indicates that the current value of h_{dj} is lack of confidence, so a natural choice is to lead z_{dj} to 0 by adjusting t_{dj} to make it more possible for h_{dj} to flip, w.r.t $t_{dj}=r(h_{dj})=+1$.

In summary, the training process of a n-layer hard-threshold neural network has both optimization problem on weights and convex-combinatorial optimization problem on target vectors, so a mixed convex-combinatorial optimization problem is formed as a result. The block diagram of target propagation algorithm is shown in Fig.5.

Layer loss function

Till now we are still facing a problem of choosing layer loss function. According to related work[16], it is acceptable to adopt soft hinge loss and weighing according to the gradient, which is shown in Fig.6.

Methods

Preparing reaction database

The reaction database used in this work comes from the 1976-2013 USPTO dataset complied by Lowe[17], which is augmented and filtered by Coley et al. [17] using the popular template set extracted by Law et al[17]. and Bogevig et al[17]. It contains 15000 groups of reactants corresponding to 15000 real reactions, approximately 5 million reactions including real and fake ones, and is stored in MongoDB format.

Structure of Edit Vector

Atom features used in this work is much more complex than the simplified example mentioned above, while bond features are the same. Specific structure is shown in Tab.1.

Tab.1 Structure of Edit Vector

Object	Index	Feature
Atom	0	Crippen logP contribution
	1	Crippen MR contribution
	2	TPSA contribution
	3	Labute ASA contribution
	4	Estate index
	5	Gasteiger partial charge
	6	Gasteiger H partial charge
	7–17	atomic number (1-hot)
	18–23	number of neighbors (1-hot)
	24–28	number of hydrogens (1-hot)
	29	formal charge
	30	is in ring
	31	is aromatic
Bond	0	is single bond
	1	is aromatic bond
	2	is double bond
	3	is triple bond

Structure of ECFP for hybrid model

Molecular fingerprint is also a common method for vectorizing molecules. A fingerprint is usually a 0-1 vector with an adjustable dimension and equivalent to the hash of a molecule. The Extended Connectivity Fingerprints (ECFP) proposed by Rogers et al.[13] in 2010 is a circular fingerprint based on the Morgan algorithm, which has become the de facto standard in the industry. In this paper, the ECFP of the reactants and products is used with a radius of 2 and a dimension of 1024 as a supplement to the Edit Vector to construct the hybrid model, where the radius and dimension are determined by convention.

Building training platform

PyTorch is one of the most popular deep learning frameworks, and its high customizability brings much convenience to implement all kinds of "non-standard" backpropagation algorithm. Therefore, this work uses PyTorch and a NVIDIA GeForce GTX 1070 to finish all experiments.

The reaction set containing 15000 groups of reactants were divided by convention: the latter 20% (3000 groups) is the test set that is not involved in training or validation but only in evaluation after training; as for the former 80% (12000 groups), 1/8 (1500 groups) is randomly taken as the validation set before the training of each model, and the remaining 7/8 (10500 groups) are taken as the training set.

Hyperparameters used by Coley [17] are: the hidden nodes structure of four subnetwork evaluating Edit Vector is [200/100/50], while that of integrating subnetwork is [50/1]; the activation is Tanh, and the optimizer is AdaDelta (ρ=0.95); each batch contains 20 groups of reactants, and each model is trained for 85 epochs. This work will apply hard-threshold neural network based on these hyperparameters and find a better combination.

Results And Discussion

By observing the prediction accuracy history on validation set in primary test, it can be found that the model tends to be stable after 100 epochs, and the AdaM optimizer makes training more stable compared to AdaDelta. Therefore, the following experiments will be performed in these two conditions.

Edit Vector based model

The core of the hard-threshold neural network is activation, so its influence on the model is examined first. The results are shown in Tab.2, where the "(Soft/Hard)" in the first column means the function has soft/hard threshold, and the prefix of the Staircase activation represents the order, that is, the number of "stairs" on its image.

Tab.2 Influence of hard-threshold activation

on Edit Vector based model

Activation	Train Acc	Val Acc	Test Acc
Tanh(Soft)	80.0%	71.1%	70.0%
Step(Hard)	72.7%	71.5%	69.1%
3-Staircase(Hard)	76.4%	69.3%	69.8%
5-Staircase(Hard)	78.0%	68.9%	68.8%
7-Staircase(Hard)	**80.1%**	**70.0%**	**71.2%**
10-Staircase(Hard)	80.0%	69.8%	70.8%

Although innocent Step activation works even worse than Tanh activation, the prediction accuracy gradually improves as the order of the hard-threshold activation increases, which can be seen from Tab.2. When the order reaches 7, Staircase activation has achieved a higher prediction accuracy than traditional activation, indicating that the target propagation does have an advantage. However, prediction accuracy decreases on the contrary when the order continues to increase to 10, indicating that 10 is unnecessary and 7 is proper for the order, and thus following experiments will be performed with 7-Staircase activation.

Tab.3 Influence of subnetwork structure

on Edit Vector based model

Subnetwork structure	Train Acc	Val Acc	Test Acc
200/100/50	**80.1%**	**70.0%**	**71.2%**
250/100/50	**80.0%**	**72.0%**	**71.3%**
250/125/50	78.8%	71.7%	70.9%
300/100/50	78.7%	69.8%	69.8%
100/100/50/50/50	76.0%	70.4%	69.9%
100/100/50/50/50	74.7%	70.1%	69.5%

Furthermore, test results in term of subnetwork structure is shown in Tab.3. It can be seen that the prediction accuracy cannot be significantly improved via deepen or to widen the structure of subnetworks, sometimes, the prediction accuracy even may decrease, which means that the subnetwork structure of 200/100/50 is complicated enough in this task. In other words, what limits the accuracy is overfitting rather than underfitting, and more hidden nodes will only disturb training. Therefore, the following experiments will focus on how to avoid overfitting using original subnetwork structure.

Dropout is a common and convenient strategy to avoid overfitting. The original idea of dropout is very simple: in each forward propagation step, forcing some hidden nodes to output zero just like “killing” them, where “killed” ones are randomly chosen with a pre-set dropout rate before each forward propagation step starts. In this way, hidden nodes are prevented from connecting incorrectly and overfitting can thus be avoided. It should be noticed that the dropout rate must be set carefully: a

too low dropout rate cannot avoid overfitting obviously, while a too high one will lead to underfitting instead because it hurts the neural network too much. Test results of dropout rate are shown in Tab.4.

Tab.4 Influence of dropout rate on Edit Vector based model

Dropout rate	Train Acc	Val Acc	Test Acc
0	80.1%	70.0%	71.2%
0.01	77.3%	69.8%	70.1%
0.02	**79.9%**	**72.5%**	**72.7%**
0.1	75.4%	69.7%	70.8%

As shown in Tab.4, high dropout rate (such as 0.1) damage the model too much, while low dropout rate (such as 0.01) cannot solve the overfitting problem, and the prediction accuracy does not improve in both cases. A dropout rate of 0.02 can achieve a balance between the above two cases, in other words, it can moderate overfitting while not damaging the model too much.

Integrating the adjustments in this section, the prediction accuracy of the final model is shown with bold line in Tab.4. The test accuracy is 72.7%, which is higher than that of Coley's Edit Vector based model (68.5%).

Hybrid model

As mentioned in Section 1.3, the mixing factor ε determines the proportion of the ECFP subnetwork output in the summation subnetwork outputs, which directly determines how much the model relies on ECFP. According to the results in Section 4.2, a more complex sub-network is meaningless for prediction, so only the influence of mixing factor and dropout rate is examined in this section, and the results are shown in Tab.5.

Tab.5 Effect of mixing factor ε and deactivation rate

on hybrid model

ε	Dropout rate	Train Acc	Val Acc	Test Acc
0	0	80.1%	70.0%	71.2%
1.0000	0	99.9%	63.1%	61.6%
0.1000	0	99.7%	65.1%	66.9%
0.0200	0	98.8%	71.0%	70.8%
0.0010	**0**	**85.3%**	**71.9%**	**72.5%**
0.0008	**0**	**85.5%**	**75.3%**	**72.6%**
0.0005	0	83.6%	70.1%	72.3%
0.0010	0	85.3%	71.9%	72.5%
0.0010	0.01	85.7%	73.8%	73.0%
0.0010	**0.02**	**85.4%**	**73.7%**	**73.9%**
0.0010	0.05	83.4%	73.1%	73.1%
0.0010	0.1	82.6%	70.9%	72.7%
0.0010	0.2	77.9%	68.7%	70.3%

As shown in Tab.5, when mixing factor is large (such as 1), the model shows a very serious overfitting. However, when mixing factor is gradually reduced to around 0.001, there is almost no obvious overfitting in the model, and the prediction

accuracy is even better than the Edit Vector based model (ε=0), indicating that the extra information introduced by the ECFP does help the prediction. Moreover, if the mixing factor continues to decrease, the hybrid model will lose its meaning. The influence of dropout rate on the hybrid model is nearly the same as that on the Edit Vector based model, which means a dropout rate of 0.02 can give a balance to the model.

Integrating the adjustments of mixing factor and dropout rate, the prediction accuracy of the final model is shown with bold line in Tab.5. The test accuracy is 73.9%, which is higher than that of Coley's hybrid model (72.8%).

Prediction examples

Neural network is often criticized for its "black box" feature (the output cannot be explained to human exactly), but it can still be observed through examples. Here we take the final Edit Vector based model in Section 4.1 (referred to as "optimized model") as an example to compare with the model trained with the same hyperparameters from literature (referred to as "original model"), and give two prediction examples.

For the reaction in Fig.7, the substitution should occur on the pyrazole ring due to the strong electron withdrawing effect of the nitro group. The optimized model assigned a probability of 33.1% to the true product. On the other hand, the original model assigned a probability of 1.7% to the true product, and a probability of 31.6% to the wrong product.

For the reaction in Fig. 8, since the hydrochloric acid-pyridine condition is weakly acidic, the imine hydroxyl group on the product should not dehydrate to form a cyano group. The optimized model assigned a probability of 70.1% to the true product. On the other hand, the original model assigned a probability of 47.1% to the true product, and a probability of 48.5% to the wrong product.

Conclusion

In this work, we implemented the vectorized description of reaction by using the Edit Vector and ECFP, and applied the hard-threshold neural network with target propagation algorithm to the template-based forward reaction prediction thereby. In this way, we achieved a higher prediction accuracy than the traditional neural network with backpropagation algorithm on the same dataset, and thus provided a new idea for computer-aided template-based forward reaction prediction.

Abbreviations

SMILES: Simplified Molecular Input Line Entry Specification

SMARTS: SMiles ARbitrary Target Specification

ECFP: Extended-Connectivity FingerPrint

Declarations

- Availability of data and material

Reaction database can be downloaded from:

https://figshare.com/articles/MongoDB_dump_compressed_/4833482

An implementation of FTPROP-MB can be found in:

https://github.com/afriesen/ftprop

- Funding

The authors gratefully acknowledge financial support from the National Scientific Foundations of China (NSFC, Grant No. 21706143) and the State Key Laboratory of Chemical Engineering (Grant No. SKL-ChE–18T01)

- Authors' contributions

Dr. ZY conceived and guided the project, and contributed to writing the manuscript. Mr. HH built the model, performed all experiments, and wrote the manuscript. All authors have read and approve of the final manuscript.

- Acknowledgements

We would like to acknowledge the contributions of Zhaofeng Si for introducing PyTorch to Mr. HH and Yu Cheng for providing organic chemistry support.

- Competing Interests

The authors declare no competing financial interest.

References

1. Scannell J W, et al. Diagnosing the decline in pharmaceutical R&D efficiency[J]. Nature Reviews Drug Discovery. 2012, 11: 191-200.
2. Butler K T, et al. Machine learning for molecular and materials science[J]. Science. 2018, 559: 547-555.
3. Corey E J, Wipke W T. Computer-Assisted Design of Complex Organic Syntheses[J]. Science, 1969, 166(3902): 178-192.
4. Schneider G. Automating drug discovery[J]. Nature Reviews Drug Discovery. 2018, 17: 97-113.
5. Vamathevan J, et al. Applications of machine learning in drug discovery and development[J]. Nature Reviews Drug Discovery. 2019, 18: 463-477.
6. Ekins S, et al. Exploiting machine learning for end-to-end drug discovery and development[J]. Nature Materials. 2019, 18: 435-441.
7. Button A, et al. Automated de novo molecular design by hybrid machine intelligence and rule-driven chemical synthesis[J]. Nature machine intelligence. 2019, 1: 307-315.
8. Segler M, Preuss M, Waller M. Planning chemical syntheses with deep neural networks and symbolic AI[J]. Nature. 2018, 555: 604-610.
9. Ahneman D, et al. Predicting reaction performance in C-N cross-coupling using machine learning[J]. Science. 2018, 360: 186-190.
10. Coley C, et al. A robotic platform for flow synthesis of organic compounds informed by AI planning[J]. Science. 2019, 365: eaax1566.
11. Coley C W, Barzilay R, Jaakkola T S, et al. Prediction of Organic Reaction Outcomes Using Machine Learning[J]. ACS central science, 2017, 3(5): 434-443.
12. Schwaller P, Gaudin T, Lanyi D, et al. "Found in Translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models[J]. Chemical Science, 2018, 9(28): 6091-6098.
13. Rogers D, Hahn M. Extended-connectivity fingerprints[J]. Journal of Chemical Information and Modeling, 2010, 50(5): 742-754.
14. Le Cun Y. Learning process in an asymmetric threshold network[M]//Disordered systems and biological organization. Springer, Berlin, Heidelberg, 1986: 233-240.

15. Friesen A L, Domingos P M. Deep Learning as a Mixed Convex-Combinatorial Optimization Problem[J]. international conference on learning representations, 2018.
16. Wu Y, Liu Y. Robust Truncated Hinge Loss Support Vector Machines[J]. Journal of the American Statistical Association, 2007, 102(479): 974-983.
17. Lowe D M. Extraction of chemical structures and reactions from the literature[D]. University of Cambridge, 2012.
18. Law J, et al. A retrosynthetic analysis tool utilizing automated retrosynthetic rule generation[J]. Journal of Chemical Information and modeling, 2009, 49: 593-602.
19. Bogevig A, et al. Route design in the 21st century: the ICSYNTH software tool as an idea generator for synthesis prediction[J]. Organic Process Research & Development. 2015, 19: 357-368.
20. Srivastava N, et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting[J]. Journal of Machine Learning Research. 2014, 15: 1929-1958.

Figures

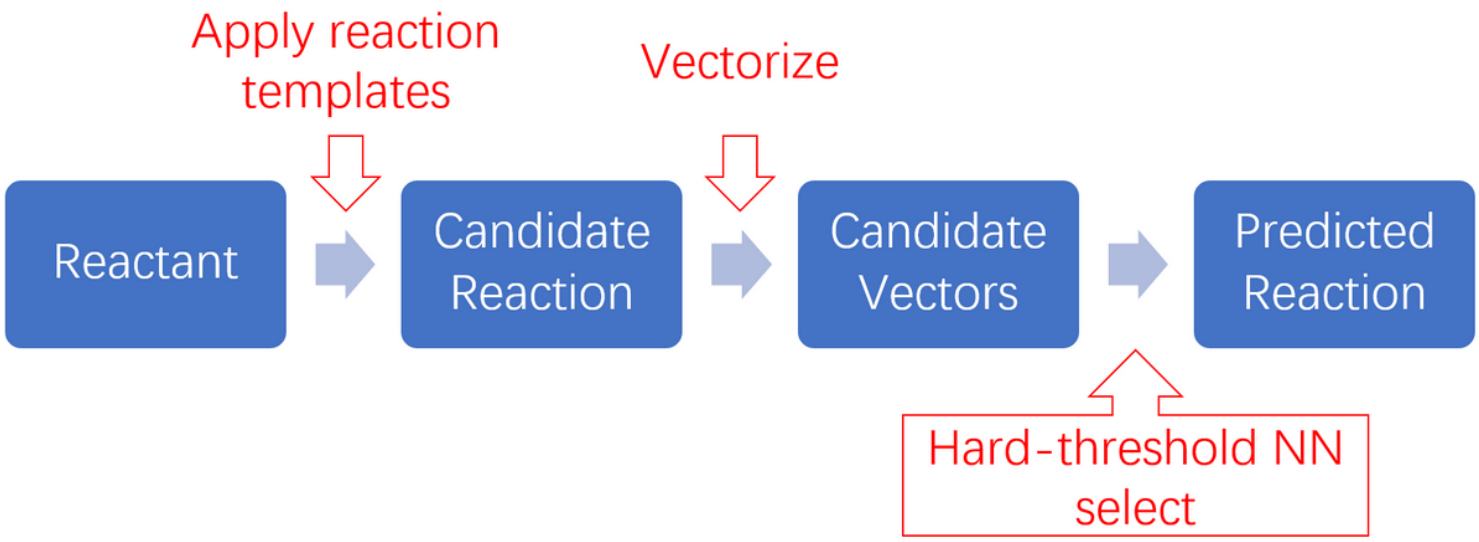


Figure 1

The illustration of template-based forward reaction prediction process

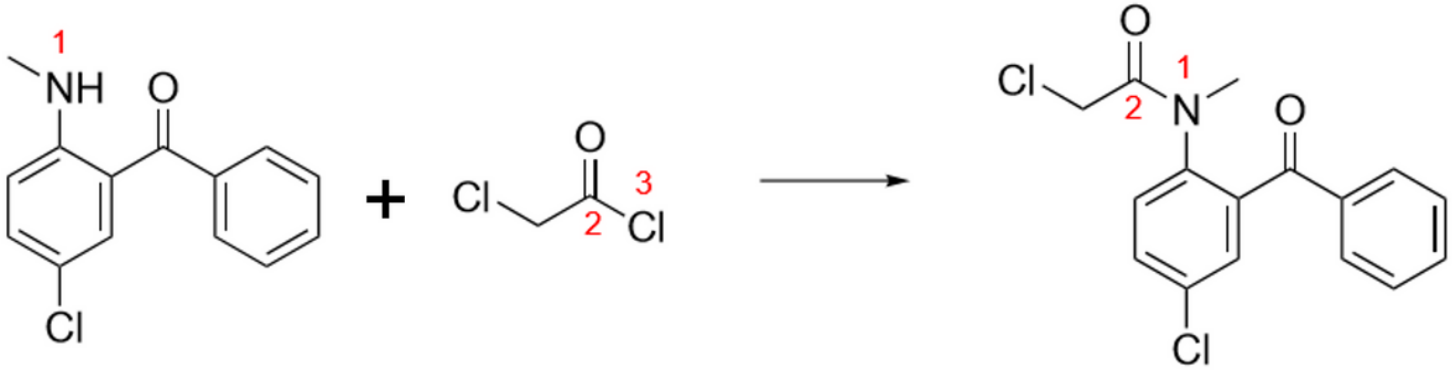


Figure 2

Reaction between chloroacetyl chloride and 2-methylamino-5-chlorobenzophenone.

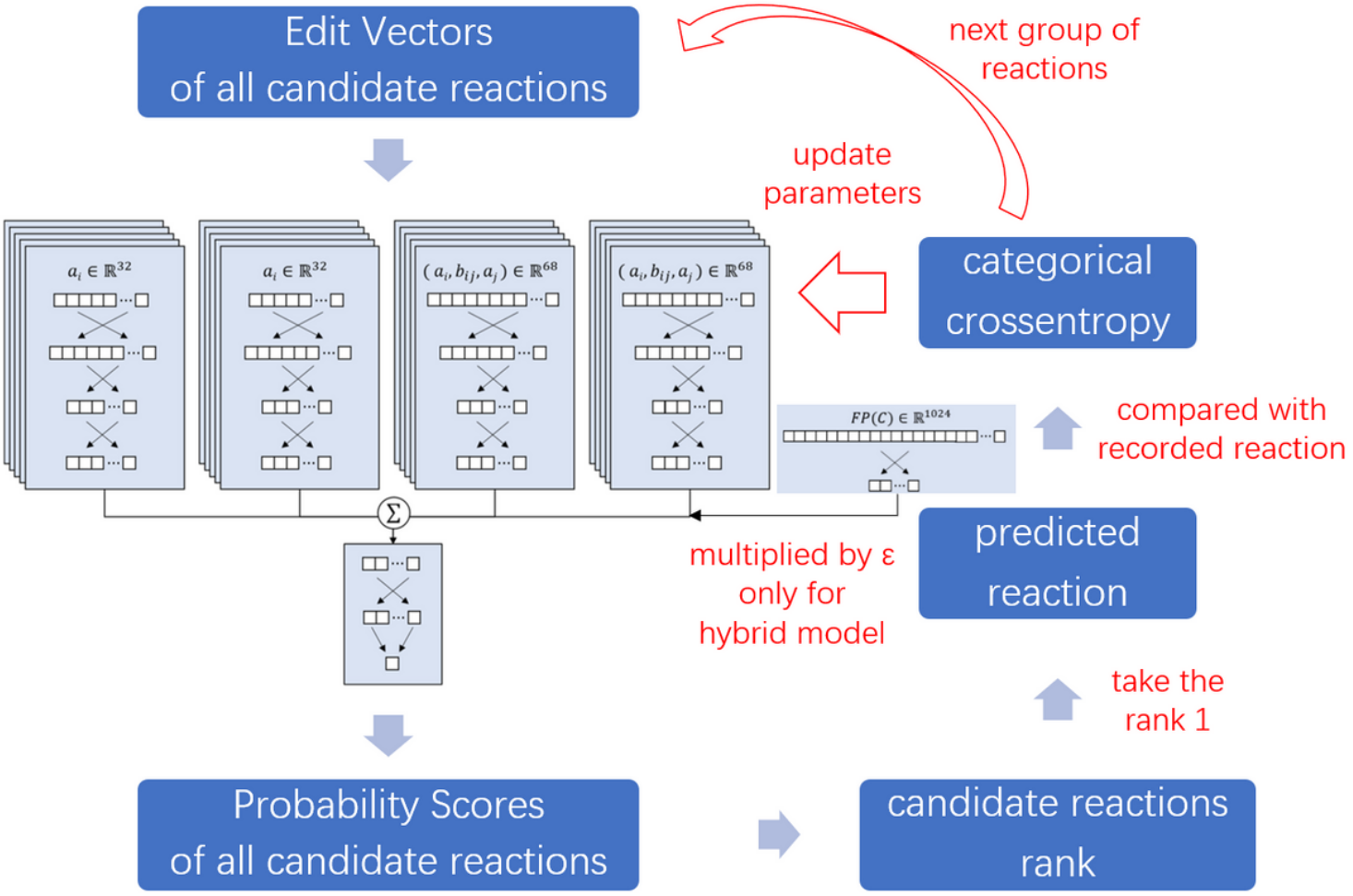


Figure 3

The illustration of selection process.

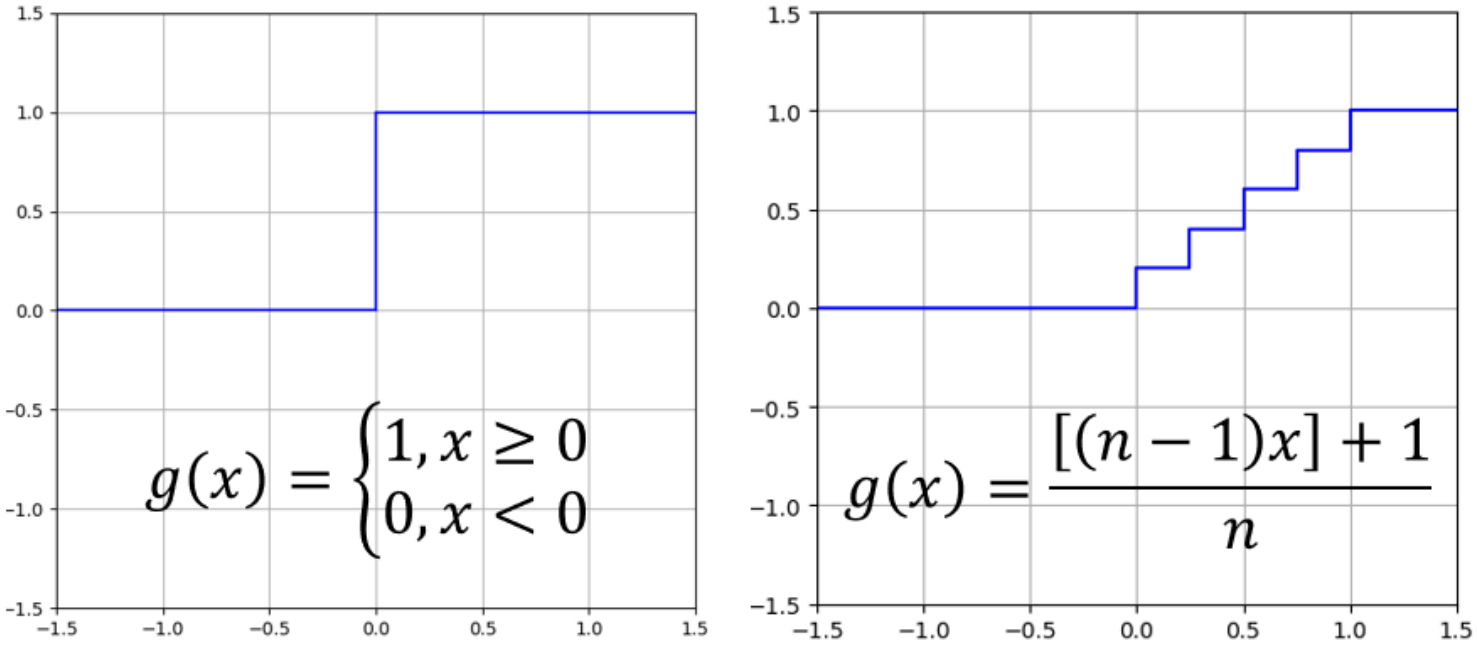


Figure 4

Hard-threshold activation.

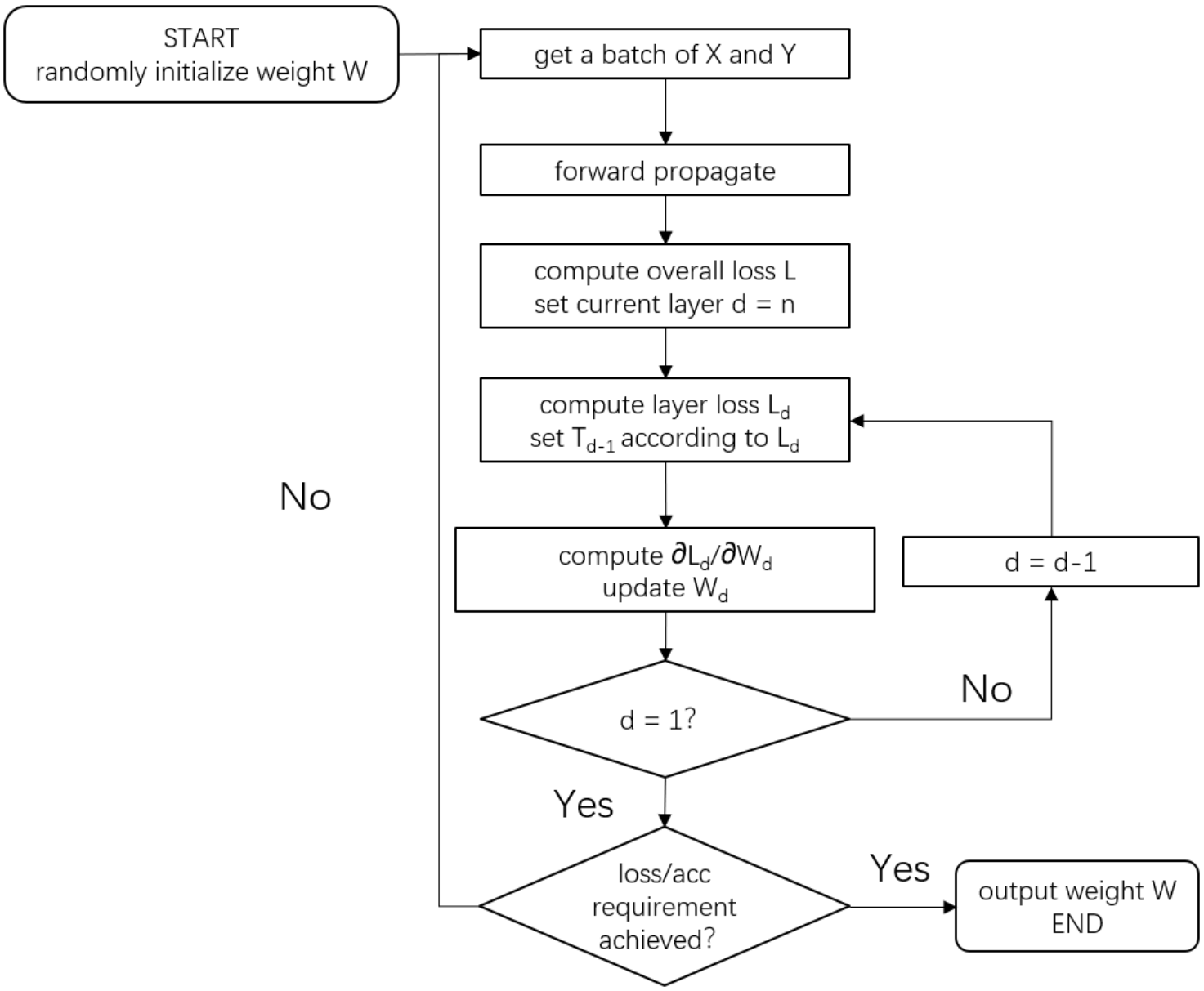
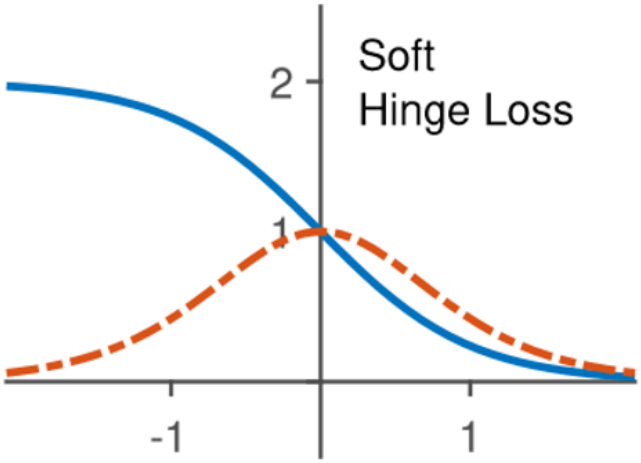


Figure 5

Block diagram of target propagation algorithm.



$$soft\_hinge(t_{dj}z_{dj}) = tanh(-t_{dj}z_{dj}) + 1$$

$$L_d = \sum_j [soft\_hinge(z_{dj}, t_{dj}) \cdot \left|\frac{\partial L_{d+1}}{\partial h_{dj}}\right|]$$

Figure 6

Weighted soft hinge layer loss function.

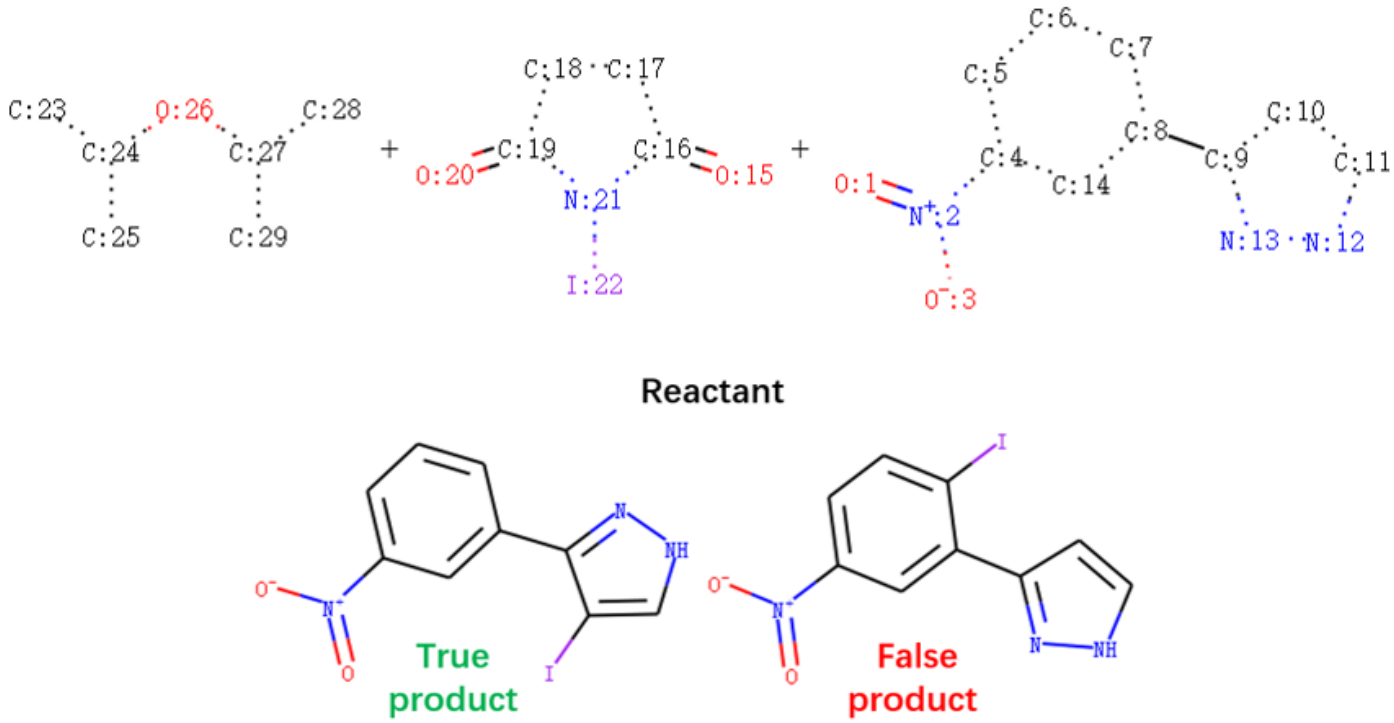


Figure 7

Iodine reaction in 2-propan-2-yloxypropane between NIS and 3-(3-Nitrophenyl)-1H-pyrazole

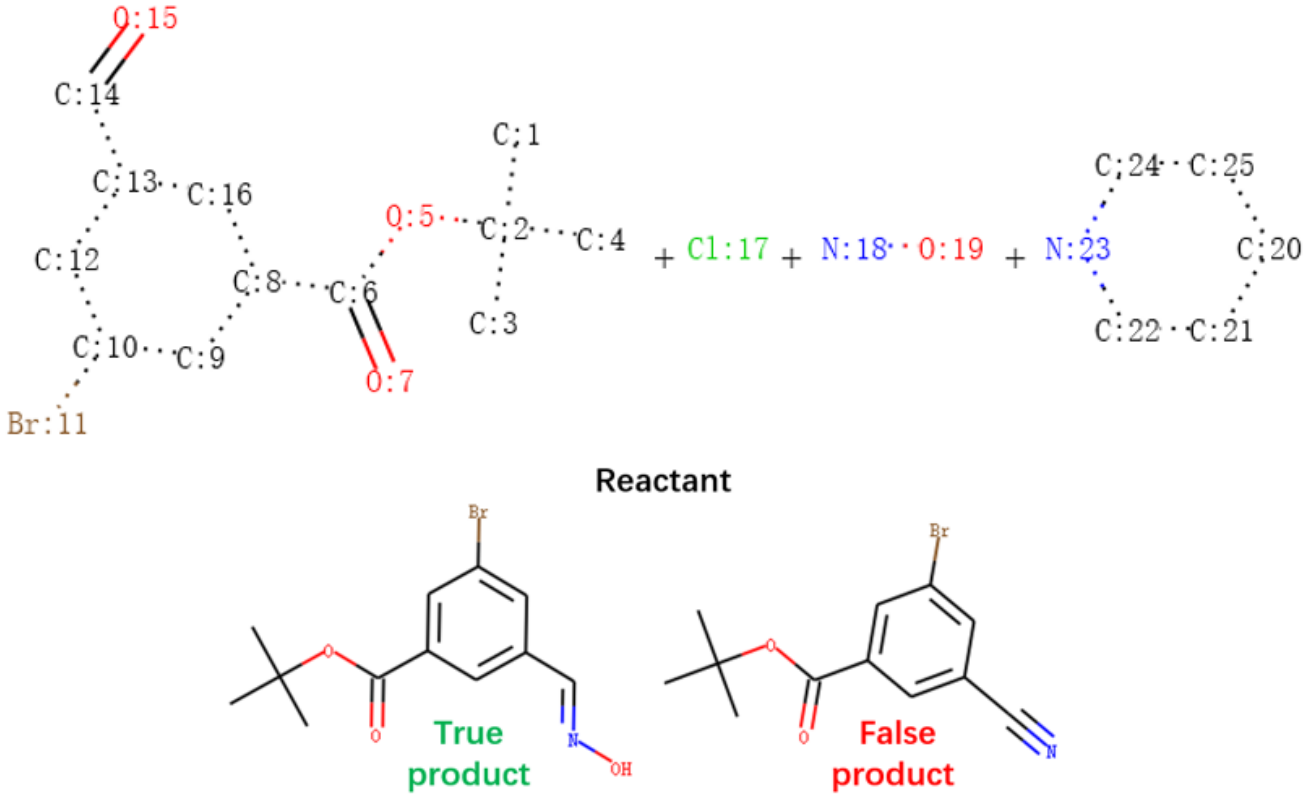


Figure 8

Amination of hydrazines with aromatic aldehydes under hydrochloric acid-pyridine condition.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- formulas.docx