JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim

Application Note

AiZynthTrain: Robust, Reproducible, and Extensible Pipelines for Training Synthesis Prediction Models

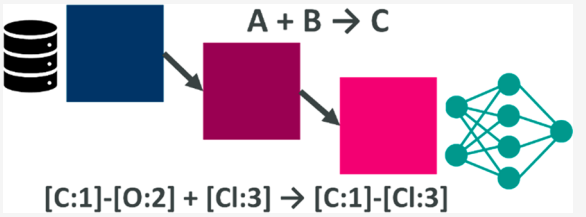
Samuel Genheden,* Per-Ola Norrby, and Ola Engkvist

Cite This: *J. Chem. Inf. Model.* 2023, 63, 1841−1846

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: We introduce the AiZynthTrain Python package for training synthesis models in a robust, reproducible, and extensible way. It contains two pipelines that create a template-based one-step retrosynthesis model and a RingBreaker model that can be straightforwardly integrated in retrosynthesis software. We train such models on the publicly available reaction data set from the U.S. Patent and Trademark Office (USPTO), and these are the first retrosynthesis models created in a completely reproducible end-to-end fashion, starting with the original reaction data source and ending with trained machine-learning models. In particular, we show that employing new heuristics implemented in the pipeline greatly improves the ability of the RingBreaker model for disconnecting ring systems. Furthermore, we demonstrate the robustness of the pipeline by training on a more diverse but proprietary data set. We envisage that this framework will be extended with other synthesis models in the future.



INTRODUCTION

Computer-assisted synthesis planning (CASP) is a field of extensive research, which in the past decade has seen a shift from rule-based manually curated expert systems to data-driven approaches guided by machine-learning (ML) models that take advantage of novel deep learning architectures.[1,2] CASP approaches have the potential to affect the synthesis of novel chemical materials by providing for instance predictions on feasibility of reactions, optimization of reaction conditions, and retrosynthesis planning of novel compounds. Historical reaction data for training ML models are available in open sources such as the extracts from the U.S. Patent and Trademark Office (USPTO) by Lowe,[3] the Open Reaction Database,[4] and SynKB,[5] but the majority of data are available in proprietary sources such as Reaxys,[6] CAS,[7] Pistachio,[8] and in-house electronic lab notebooks. Furthermore, the quality of the reaction data available for modeling might sometimes be questionable, and there is a common lack of negative data, something that has been pointed out in recent publications.[9,10]

Software for synthesis prediction is dominated by commercial programs,[11] although free and sometimes open-source platforms such as ASKCOS from MIT[12] and RXN for Chemistry from IBM[13] are available. We have open sourced the AiZynthFinder software[14] that can be used to predict synthetic routes for novel compounds. It was originally based on a template-based, one-step retrosynthesis model coupled together with a Monte Carlo tree search algorithm[15] but has since been extended to support other types of retrosynthesis models and search algorithms.[16] The models distributed with AiZynthFinder were trained on the USPTO data set, and the codes that was used to train these models were also open sourced.[17] However, the codebase is not in a state that lends itself to updates or extension of the code when the models need to be retrained. Because the majority of models in current CASP tools were trained on historical data, these models should likely be periodically updated. In the case of retrosynthesis models, retraining would give models the opportunity to utilize the latest reaction types when suggesting disconnections. In general, there is a lack of robust and reproducible data pipelines in the CASP community. At best, the training code is scattered throughout several scripts or notebooks, which is not a sustainable software solution.

In this application note, we describe a set of pipelines that enables the training of retrosynthesis models to be performed in a transparent and reproducible fashion. The starting point of the pipelines is a collection of atom-mapped reaction SMILES, and the end point consists of retrosynthesis models that can be integrated into AiZynthFinder or other similar packages. Together with the end-to-end pipelines for the USPTO data set in the recently released *rxnutils* package,[18] this forms a complete set of reproducible pipelines for training retrosynthesis models for the USPTO data set, starting with downloading the source data files and ending at trained models. To show the robustness of the developed pipelines, models trained on internal reaction data are also discussed.

Received: December 1, 2022
Published: March 24, 2023

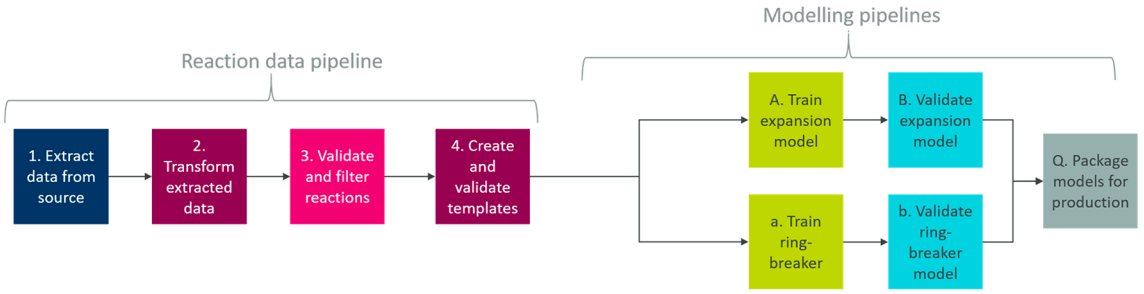


Figure 1. Overview of pipelines. Steps 1−4 form a data pipeline producing cleaned reaction SMILES and reaction templates. Steps A and B, as well as steps a and b, form the existing modeling pipelines.

SOFTWARE OVERVIEW

The AiZynthTrain is a Python 3 package that consists of a number of routines that can be used to train synthesis prediction models. The most convenient approach to use these routines is by using one or more pipelines that utilize the appropriate routines in the correct order to train the models. The pipelines will also produce a human-readable report at each key step of the process, allowing a scientist to gauge the state of the data, identify problems, and evaluate success metrics. Currently, the package has two pipelines that together train a template-based one-step retrosynthesis model, and these pipelines are illustrated in Figure 1. The first pipeline typically requires a lot of memory and can be easily parallelized across many CPUs but needs no GPUs, whereas the second pipeline requires less memory and needs to be run on a GPU. Therefore, the process is divided into two pipelines. For instance, for our internal reaction data set, we needed more than 80 GB of memory to load the entire data set for analysis in the first pipeline, whereas the second pipeline only required about 32 GB of memory. These numbers would of course differ between data sets. Each step of the pipelines is explained in more detail below.

Data Extraction. To function, the pipeline needs a tab-separated values file (TSV) with the appropriate columns, and the first step will ensure that such a file is created from a particular source. Not all columns are needed for the modeling but are there for displaying statistics. Most importantly, the pipeline needs an atom-mapped reaction SMILES. We have previously provided a set of pipelines for the USPTO data set that creates atom-mapped reaction SMILES using the *rxnmapper* tool,[18,19] and the output of those pipelines can be used as input for AiZynthTrain. Alternatively, one could use the Open Reaction Database[4] as a starting point. For internal use, we import data from our in-house solution which is a combination of Reaxys,[6] Pistachio,[8] and ELN (Electronic Lab Notebook) data. These reactions are atom-mapped using the NextMove *namerxn* software[20] when possible, otherwise by Biovia.[21]

Reaction Data Transformation and Validation. The next steps ensure that we have appropriately formatted reaction SMILES that can be parsed with RDKit[22] and that we have a reasonable level of confidence in the validity of the data. Identifying erroneous reaction SMILES is an ongoing process, and not all reactions that are included in the modeling might be correct. Furthermore, it should be noted that we do not attempt to fix erroneous SMILES; we simply remove them from the data set. To begin with, we remove all molecules from all SMILES that cannot be sanitized with RDKit. Second, we perform role assignment based on the atom-mapping; all reactants should share at least one atom-mapping number with a product. We strip the atom-mapping of the reagents and make sure they are placed in the middle portion of the reaction SMILES, so that it follows the format reactants > reagents > products. Then, we neutralize the molecules using RDKit routines (that will attempt to add or remove hydrogens whenever is possible) before we remove any product that also is identical to a reactant. These steps provide the cleanest reaction SMILES we can obtain, and we can start to remove reactions based on certain criteria, which are outlined in Table S1. In the end, we remove duplicate reactions based on reactants and products, because only one example is needed for template extraction. This step also includes identifying reactions suitable for the RingBreaker model.[23] Compared to the original publication, we identify ring-forming reactions based on three criteria: (1) The number of rings in the products should be greater than the number of rings in the reactants (only criteria used in the original publication). (2) A bond in a ring should have been made. (3) The size of the new ring should be between 3 and 7.

Template Extraction and Validation. Next, we extract templates using the RDChiral package.[24] We use a radius 1, except for ring-forming reactions, where we use a special logic: first we take all the atoms at radius 0, i.e., the atoms changed during the reactions, and augment them with all the atoms that are in the formed ring. Finally, we include all heteroatoms bonded to these atoms in the template. This logic gives a template that is more specific than using radius 0 but more generic than using radius 1. All templates are extended with special groups, as originally implemented in RDChiral[24] and later extended by Thakkar et al.[17] The template extraction step also creates a unique identifier for each template based on a fingerprint of the template. The extraction is followed by a validation, where we remove templates that cannot reproduce the recorded reactants when applied to the recorded product. We also remove templates that have more than one product, i.e., where the extraction produced two disconnected fragments. These templates are typically hard to apply to novel compounds. We only keep templates that have more than $N = 3$ example reactions for the USPTO data set or $N = 10$ for our internal data set. Finally, we create the appropriate input for training an expansion policy, both a general model based on all data and a RingBreaker model[23] based on only ring-breaking templates.

Expansion Policy Preprocessing and Training. The expansion policy pipeline starts with producing a template data set that is necessary for template application in AiZynthFinder, as well as a lookup data set that can be used to find database examples based on the template identifier. We also preprocess

https://doi.org/10.1021/acs.jcim.2c01486
J. Chem. Inf. Model. 2023, 63, 1841−1846

Table 1. Statistics on Reaction and Template Data

	USPTO-based data	AZ-based data
# reactions extracted from source	3,521,677	39,764,075
# reactions after filtering	3,285,790	34,218,273
# unique reactions after filter	1,198,554	18,697,432
# RingBreaker reactions	274,093	4,4793,374
# extracted templates	1,194,703	18,603,069
# templates after filtering and selection	819,797	10,168,247
# unique templates with occurrence ≥3 or 10	42,554	179,128
# unique RingBreaker templates	5282	39,198

the training data to create the neural network input and output data, i.e., the product fingerprint and template class index, instead of producing these on the fly in the training. Furthermore, the data are split into training, validation, and test sets. First, we include all reactions in the 20K PaRoutes reference routes[16] in the test partition; the creation of these routes are described below. Second, we perform a stratified split of the remaining templates into training and validation: for each template class, we take 90% of the data as training and 10% of the data as validation. Looking at the entire data set, this stratification leads to a little less than 90% of the data taken as training data. We then train a Keras/Tensorflow model using the same hyperparameters as originally proposed.[17] Because of the expensive training due to the large model size, we cannot afford a hyperparameter optimization and thus rely on the original optimization.

Expansion Policy Validation. Validating the trained expansion model is the final step of the pipeline, and we perform validation of both the model itself but also on the performance of the multistep retrosynthesis when the retrained model is used. For evaluating the expansion model, we produce statistics of the training and apply the expansion model on a subset of the test set (reactions in the PaRoutes reference routes). This subset was created by sampling 1000 products from the test set and ensuring that all reaction classes in the test set are represented. For evaluating the multistep retrosynthesis performance, we first check how well the search finds synthetic routes where all the starting materials are in stock for 50 random ChEMBL[25] compounds. Next, we check how well the search recovers 20 randomly selected PaRoute reference routes. The number of target molecules evaluated automatically is sufficient to compare the performance from one training session to another but is probably not sufficient to judge how well the model performs on a larger set. Nevertheless, the validation is summarized in a report that can be used to gauge if the model is sufficiently good to use for production.

MODEL TRAINING DETAILS AND VALIDATION

We made a version 2.0 of the PaRoutes benchmark set from the selected USPTO reactions. All the reactions corresponding to the selected reaction templates were used as a basis for extracting routes as previously described.[16] This produced two new sets of reference routes, with 10k routes each, for retrosynthesis benchmarking and a larger set of 450k routes for training machine-learning models. We consider this to be a better version of the PaRoutes benchmark set because the data were derived in a more transparent and reproducible fashion. It is still based on the USPTO data set as the previous version, but the atom-mapping is more consistent and the reaction data cleaner. Furthermore, since we exclude the reactions in the PaRoutes 2.0 reference routes from the training of the retrosynthesis models (see above), we obtain a self-consistent procedure to derive models and extracted benchmarking routes. Any reference to PaRoutes below is referring the new version.

Using the pipelines in the AiZynthTrain package, we trained two one-step retrosynthesis models based on the USPTO data set: one based on all extracted and selected templates and one based on a subset of those templates that correspond to the RingBreaker model. To prove the robustness of the developed pipelines, we also trained the corresponding models but based them on our internal database covering Reaxys, Pistachio, and AstraZeneca Electronic Lab Notebooks (henceforth referred to as the AZ-based model). To evaluate the general retrosynthesis models on a wide set of molecules, we selected three sets of target molecules: one set consisting of 10,000 random molecules from ChEMBL, one set consisting of 5000 molecules designed for various AstraZeneca projects, and 13,000 molecules generated by the Reinvent software[26] in various AstraZeneca projects. For these three sets, we evaluate how well AiZynthFinder finds routes where all the starting material is in stock. For the model based on USTPO, we used a stock that was a combination of ZINC[27] and eMolecules,[28] and for the AZ-based model, we used an AstraZeneca stock database. Furthermore, we also evaluated the models on the PaRoutes benchmark set to see how well the models are in recovering reference routes and how diverse the set of predicted routes are (see Supporting Information).[16]

RESULTS AND DISCUSSION

Data Processing. The AZ-based model is based on more than ten times as many reactions as in the USPTO data set, see Table 1. Here, 68% of the reactions used for the AZ-based model come from Reaxys, 27% from Pistachio, and the rest from AZ ELNs. For the USPTO-based model, 93% of the reactions pass the filters in Table S1, for the AZ-based model, it is a little less, 86%. Based on the unique reactions (counting only reactants and products), we were able to extract approximately 1 and 18.6M reaction templates, respectively. After validating them and only keeping the ones with a sufficient number of examples, we are left with approximately 820k templates for USPTO-based data and 10M for the AZ-based data. The numbers of unique templates, i.e., the output size of the expansion model, are approximately 43k for the USPTO-based model and 180k for the AZ-based model. The sizes of the training, validation, and test sets are listed in Table S2. The full reports from the pipelines of the USPTO-based model are available as Supporting Information.

RingBreaker Evaluation. The RingBreaker model was developed because the general one-step model was rarely able to break rings because such templates were rarely predicted.[23]

https://doi.org/10.1021/acs.jcim.2c01486
J. Chem. Inf. Model. 2023, 63, 1841−1846

Table 2. RingBreaker Performance on Approximately 1000 Ring-Forming Reactions[a]

	USPTO-based models		AZ-based models	
Model version	% expected reactants found	% rings broken	% expected reactants found	% rings broken
Thakkar2020[23]	69.20	88.20	30.15	80.78
AiZynthTrain	95.10	99.70	95.73	100.00

[a]Ringbreaker models are evaluated on (a) for how many of the products did the model predict the true reactants and (b) for how many of the products did the model break a ring.

Table 3. Route Finding Capabilities of Retrosynthesis Models

		USPTO-based models		AZ-based models	
Molecule set	Model version	% solved	time to first solution (s)	% solved	time to first solution (s)
ChEMBL	Thakkar2020[17]	70.74	4.65	73.16	4.74
ChEMBL	AiZynthTrain	71.61	4.56	81.18	3.41
AZ designs	Thakkar2020[17]	51.20	6.74	61.58	5.42
AZ designs	AiZynthTrain	51.68	6.72	71.82	5.98
Reinvent	Thakkar2020[17]	42.89	6.07	45.90	7.09
Reinvent	AiZynthTrain	48.37	7.27	69.81	8.92

However, over time, we noticed that even the RingBreaker model was unable to provide satisfactory disconnections for many molecules. Often, we found that a radius of 1 leads to the inclusion of atoms adjacent to the ring of interest, thus making it more difficult to apply the template to novel compounds. Therefore, we redesigned the selection criteria for identifying ring-forming reactions as well as the template extraction procedure (see above). In Table 2, we show that this had a significant impact on the performance of the RingBreaker model. We subsampled the test set for the RingBreaker model to do an analysis of approximately 1000 ring-forming reactions. First, we see that the Thakkar2020 models were only able to recover the expected reactants in 69% and 30% of the cases, for the USPTO- and AZ-based model, respectively. For the retrained model, we can recover the expected reactant in 95% of the cases for both models. To be useful, the RingBreaker model needs to disconnect rings, but it does not necessarily need to do it using a specific reaction mechanism. Therefore, we also investigated how often the models could break the ring in the product, regardless of what the predicted reactants were. We see that both of the Thakkar2020 models have a success rate in this regard that is below 90%, whereas the AiZynthTrain models can virtually break up all of the rings in the test set. This improvement makes the RingBreaker model derived with AiZynthTrain much more useful.

Route-Finding Capability. We evaluated the general one-step retrosynthesis models on the task of finding solved routes, i.e., routes where all the starting material is in stock. This gives a strong indication of how good the models are at generating novel synthesis ideas. We compared the models trained using the pipelines in the AiZynthTrain package with the models trained by Thakkar et al.[17] in order to see if the retrained models are better at finding solved routes. As seen in Table 3, for the USPTO-based models, we find solutions for approximately 70% of the ChEMBL compounds, 51% of the AZ designs, and between 43% and 49% of the Reinvent compounds. The only significant difference between the Thakkar2020 and AiZynthTrain models is seen for the Reinvent compounds, where we see an increase of approximately 6%. The difference between the Thakkar2020 and AiZynthTrain models does not only lay in the data preprocessing but also in the atom-mapping tool; NextMove[20] for the Thakkar2020 model versus the *rxnmapper*[19] for the AiZynthTrain model. We also created a USPTO-based model based on the AiZynthTrain pipelines but with the reactions mapped with the NextMove software[20] where it was possible. This model was able to solve approximately 1%−3% more routes compared to the model based on the *rxnmapper* (see Table S3), showing that the atom-mapping tools have a significant effect on retrosynthesis modeling.

One of the conclusions of Thakkar et al.[17] was that the USPTO-based model was sufficient for finding routes for compounds of interest to medicinal chemistry. This conclusion is upheld if we compare the performance of the two Thakkar2020 models on the ChEMBL and Reinvent data sets; for these compounds, we only see a small effect when training the retrosynthesis model on the much larger reaction data set from AZ. However, for the AZ designs, we see an improvement of 10% when using the Thakkar2020 model and 20% if we are using the AiZynthTrain retrained model. For the AiZynthTrain AZ-based model, we see an increase in performance of 10% and 20% for the ChEMBL and Reinvent compounds, respectively. Thus, it is clear from these results that if the task is to find solved routes and thereby generate synthesis ideas, it is better to use a model trained on a large reaction data set. Furthermore, we see a significant difference when comparing the Thakkar2020 and AiZynthTrain AZ-based models for all target sets. The rationale for this difference is harder to pick apart because we cannot reproduce the original modeling pipeline of Thakkar et al. Nevertheless, one factor that we can easily investigate is the temporal factor, i.e., that the AiZynthTrain model was trained more than three years after the Thakkar2020 models. To this end, we excluded all reactions in Reaxys, Pistachio, and AZ ELNs recorded after the first of January 2019 when processing the data set and then retrained the model. For the ChEMBL data set, this model outperforms the model trained on later data, but only by 1% (see Table S3). For the AZ design and the Reinvent molecules, the model trained on older data finds solutions to 3% and 1% fewer targets, respectively, than the model trained on all data. Thus, it seems that there is a limited effect when including the reactions from the last three years, and most of the improvement observed in the AiZynthTrain models is related to better-curated data and modeling pipelines. However, it might be valuable to investigate the temporal effect further in

https://doi.org/10.1021/acs.jcim.2c01486
J. Chem. Inf. Model. 2023, 63, 1841−1846

the future since it is one of the main motivations to have robust pipelines for (re)training models.

CONCLUSION AND OUTLOOK

We have developed the AiZynthTrain package to train synthesis prediction models in an end-to-end fashion. The pipelines incorporate advanced techniques that we have iteratively refined by working with large reaction data sets and synthesis predictions over several years. Therefore, we believe that the models we have trained are based on the most reliable data set that we currently have available. We have also shown that the retrained AZ-based models give a substantial increase in the capability of AiZynthFinder to find routes that could be of interest to medicinal chemists. The package is straightforward to use and extensible to other pipelines. We provide the models trained for this paper on the USPTO data set for downloading.

We provide the AiZynthTrain package as open source to encourage adoption, scrutiny, and extension by the large computer-assisted synthesis planning (CASP) community. We envisage that the package will be extended by pipelines for quick forward filters as well as with pipelines for other retrosynthesis models. Currently, it is difficult to train available one-step models on other data sources in a reproducible fashion, and we believe that AiZynthTrain offers a solution to this issue. Furthermore, even though we focus on models to be used with our own retrosynthesis engine, the models created with the pipelines can be used with any retrosynthesis software that supports machine-learning models. With sufficient experience of training synthesis prediction models and more elaborate approaches to evaluate them, we also envisage that the models can be pushed automatically to production after an automated system has checked the reports produced by the pipelines. In summary, we believe that AiZynthTrain provides a solution to an important unmet need in the CASP community, and we envisage that it will become very useful for furthering the dissemination of transparent research.

ASSOCIATED CONTENT

Data Availability Statement

The AiZynthTrain package is released on GitHub under the Apache 2.0 license: https://github.com/MolecularAI/aizynthtrain. The USPTO models and the PaRoutes reference data are available from Zenodo: https://zenodo.org/record/7341155. The ChEMBL 10k data set was previously released and is available from Zendo: https://zenodo.org/record/4925903. The ZINC stock was previously released and is available here: https://figshare.com/files/23086469. The eMolecules stock can be downloaded from the eMolecule homepage (https://www.emolecules.com/). The AZ designs and Reinvent molecule data sets, the AstraZeneca compound stock, and the AZ-based models are all proprietary or derived from proprietary data and software and cannot be publicly released.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.2c01486.

Additional information (ZIP)

Tables (PDF)

AUTHOR INFORMATION

Corresponding Author

Samuel Genheden − *Molecular AI, Discovery Sciences, R&D, AstraZeneca Gothenburg, SE-431 83 Mölndal, Sweden;* orcid.org/0000-0002-7624-7363; Email: samuel.genheden@astrazeneca.com

Authors

Per-Ola Norrby − *Data Science and Modelling, Pharmaceutical Sciences, R&D, AstraZeneca Gothenburg, SE-431 83 Mölndal, Sweden;* orcid.org/0000-0002-2419-0705

Ola Engkvist − *Molecular AI, Discovery Sciences, R&D, AstraZeneca Gothenburg, SE-431 83 Mölndal, Sweden;* orcid.org/0000-0003-4970-6461

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.2c01486

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We acknowledge Anders Egnéus, Aleksandr Savelev, and Nicholas Tomkinson for their work on the Reaction Connect database unifying all internal and external reaction data sources. We acknowledge Jon Paul Janet for compiling the Reinvent dataset and Annie Westerlund and Christos Kannas for proof-reading the manuscript.

REFERENCES

(1) Mervin, L.; Genheden, S.; Engkvist, O. AI for Drug Design: From Explicit Rules to Deep Learning. *Artif. Intell. Life Sci.* **2022**, *2*, 100041.

(2) Schwaller, P.; Vaucher, A. C.; Laplaza, R.; Bunne, C.; Krause, A.; Corminboeuf, C.; Laino, T. Machine Intelligence for Chemical Reaction Space. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2022**, *12*, e1604.

(3) Lowe, D. Chemical reactions from U.S. patents (1976-Sep2016), 2017. https://figshare.com/articles/dataset/Chemical_reactions_from_US_patents_1976-Sep2016_/5104873.

(4) Kearnes, S. M.; Maser, M. R.; Wleklinski, M.; Kast, A.; Doyle, A. G.; Dreher, S. D.; Hawkins, J. M.; Jensen, K. F.; Coley, C. W. The Open Reaction Database. *J. Am. Chem. Soc.* **2021**, *143* (45), 18820−18826.

(5) Bai, F.; Ritter, A.; Madrid, P.; Freitag, D.; Niekrasz, J. SynKB: Semantic Search for Synthetic Procedures*arXiv Preprint*, arXiv:2208.07400, 2022 DOI: 10.48550/arxiv.2208.07400

(6) *Reaxys*. https://www.reaxys.com/ (accessed 2022−09−01).

(7) CAS Content. *CAS*. https://www.cas.org/about/cas-content (accessed 2022−09−01).

(8) Pistachio. *NextMove Software*. https://www.nextmovesoftware.com/pistachio.html (accessed 2022−09−01).

(9) Beker, W.; Roszak, R.; Wolos, A.; Angello, N. H.; Rathore, V.; Burke, M. D.; Grzybowski, B. A. Machine Learning May Sometimes Simply Capture Literature Popularity Trends: A Case Study of Heterocyclic Suzuki-Miyaura Coupling. *J. Am. Chem. Soc.* **2022**, *144* (11), 4819−4827.

(10) Strieth-Kalthoff, F.; Sandfort, F.; Kühnemund, M.; Schäfer, F. R.; Kuchen, H.; Glorius, F. Machine Learning for Chemical Reactivity: The Importance of Failed Experiments. *Angew. Chemie - Int. Ed.* **2022**, *61* (29), e202204647.

(11) Johansson, S.; Thakkar, A.; Kogej, T.; Bjerrum, E.; Genheden, S.; Bastys, T.; Kannas, C.; Schliep, A.; Chen, H.; Engkvist, O. AI-Assisted Synthesis Prediction. *Drug Discovery Today: Technologies* **2019**, *32-33*, 65−72.

(12) Coley, C. W.; Thomas, D. A.; Lummiss, J. A. M.; Jaworski, J. N.; Breen, C. P.; Schultz, V.; Hart, T.; Fishman, J. S.; Rogers, L.; Gao, H.; Hicklin, R. W.; Plehiers, P. P.; Byington, J.; Piotti, J. S.; Green, W. H.; Hart, A. J.; Jamison, T. F.; Jensen, K. F. A Robotic Platform for Flow Synthesis of Organic Compounds Informed by AI Planning. *Science* **2019**, *365* (6453), eaax1566.
(13) Schwaller, P.; Petraglia, R.; Zullo, V.; Nair, V. H.; Haeuselmann, R. A.; Pisoni, R.; Bekas, C.; Iuliano, A.; Laino, T. Predicting Retrosynthetic Pathways Using Transformer-Based Models and a Hyper-Graph Exploration Strategy. *Chem. Sci.* **2020**, *11* (12), 3316−3325.
(14) Genheden, S.; Thakkar, A.; Chadimová, V.; Reymond, J. L.; Engkvist, O.; Bjerrum, E. AiZynthFinder: A Fast, Robust and Flexible Open-Source Software for Retrosynthetic Planning. *J. Cheminform.* **2020**, *12* (1), 70.
(15) Browne, C. B.; Powley, E.; Whitehouse, D.; Lucas, S. M.; Cowling, P. I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; Colton, S. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games.* **2012**, *4*, 1−43.
(16) Genheden, S.; Bjerrum, E. PaRoutes: Towards a Framework for Benchmarking Retrosynthesis Route Predictions. *Digit. Discovery* **2022**, *1* (4), 527−539.
(17) Thakkar, A.; Kogej, T.; Reymond, J. L.; Engkvist, O.; Bjerrum, E. J. Datasets and Their Influence on the Development of Computer Assisted Synthesis Planning Tools in the Pharmaceutical Domain. *Chem. Sci.* **2020**, *11* (1), 154−168.
(18) Kannas, C.; Thakkar, A.; Bjerrum, E.; Genheden, S. rxnutils − A Cheminformatics Python Library for Manipulating Chemical Reaction Data. ChemRxiv, 2022. DOI: 10.26434/CHEMRXIV-2022-WT440-V2.
(19) Schwaller, P.; Hoover, B.; Reymond, J. L.; Strobelt, H.; Laino, T. Extraction of Organic Chemistry Grammar from Unsupervised Learning of Chemical Reactions. *Sci. Adv.* **2021**, *7* (15), eabe4166 DOI: 10.1126/sciadv.abe4166.
(20) NameRxn. *NextMove Software*. https://www.nextmovesoftware.com/namerxn.html (accessed 2022−09−01).
(21) Pipeline Pilot v20.1.0.2208. *Biovia*. https://www.3ds.com/products-services/biovia/products/data-science/pipeline-pilot/
(22) *RDKit*. https://github.com/rdkit (accessed 2022−09−01).
(23) Thakkar, A.; Selmi, N.; Reymond, J. L.; Engkvist, O.; Bjerrum, E. J. "Ring Breaker": Neural Network Driven Synthesis Prediction of the Ring System Chemical Space. *J. Med. Chem.* **2020**, *63* (16), 8791−8808.
(24) Coley, C. W.; Green, W. H.; Jensen, K. F. RDChiral: An RDKit Wrapper for Handling Stereochemistry in Retrosynthetic Template Extraction and Application. *J. Chem. Inf. Model.* **2019**, *59* (6), 2529−2537.
(25) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40* (D1), D1100.
(26) Blaschke, T.; Arús-Pous, J.; Chen, H.; Margreitter, C.; Tyrchan, C.; Engkvist, O.; Papadopoulos, K.; Patronov, A. REINVENT 2.0: An AI Tool for de Novo Drug Design. *J. Chem. Inf. Model.* **2020**, *60* (12), 5918−5922.
(27) Sterling, T.; Irwin, J. J. ZINC 15 - Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **2015**, *55* (11), 2324−2337.
(28) *eMolecules*. https://www.emolecules.com/ (accessed 2022−02−01).

https://doi.org/10.1021/acs.jcim.2c01486
J. Chem. Inf. Model. 2023, 63, 1841−1846