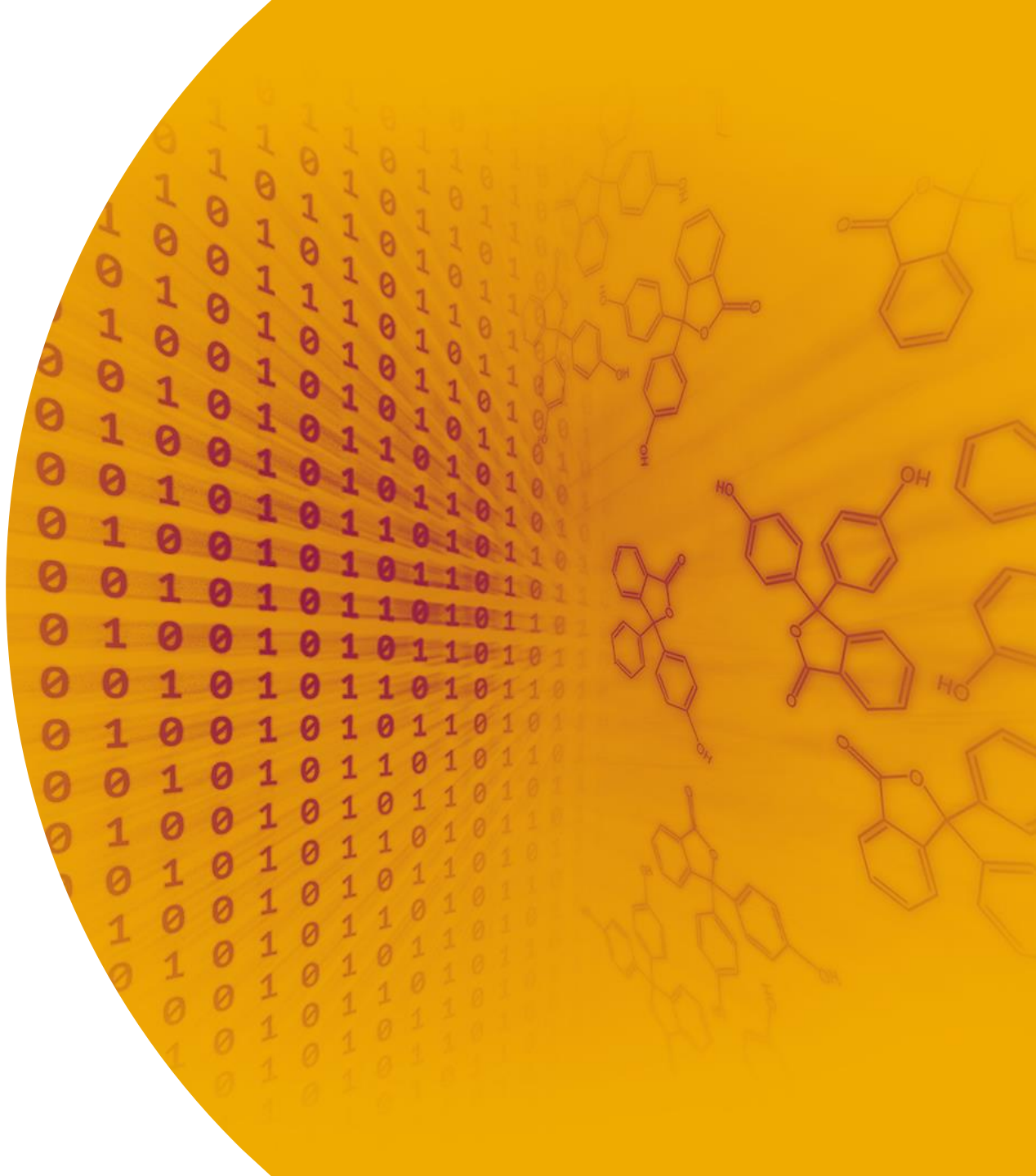
AstraZeneca

Comparing and clustering route predictions

Samuel Genheden

Molecular AI, AstraZenca

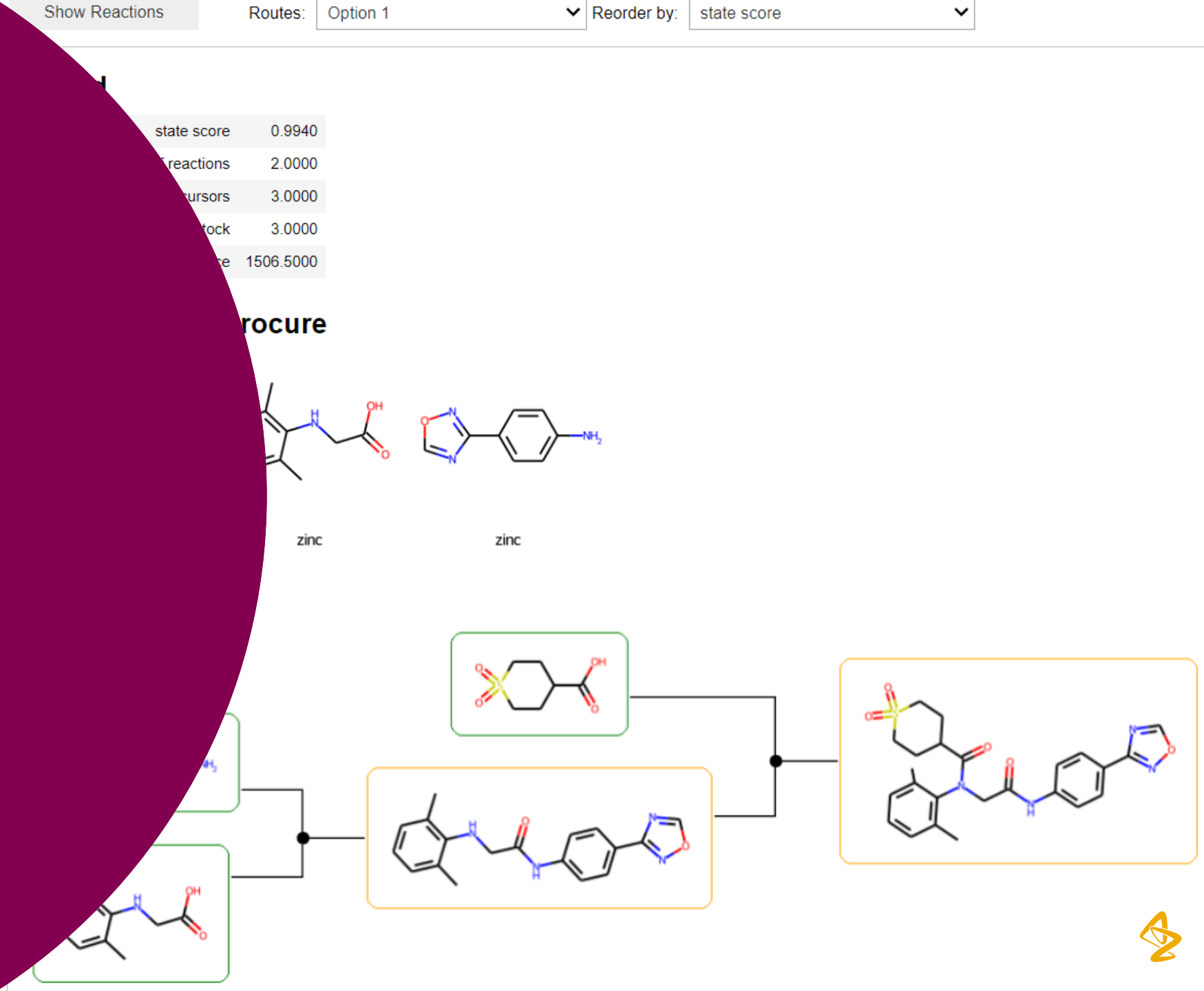
2022-05-12



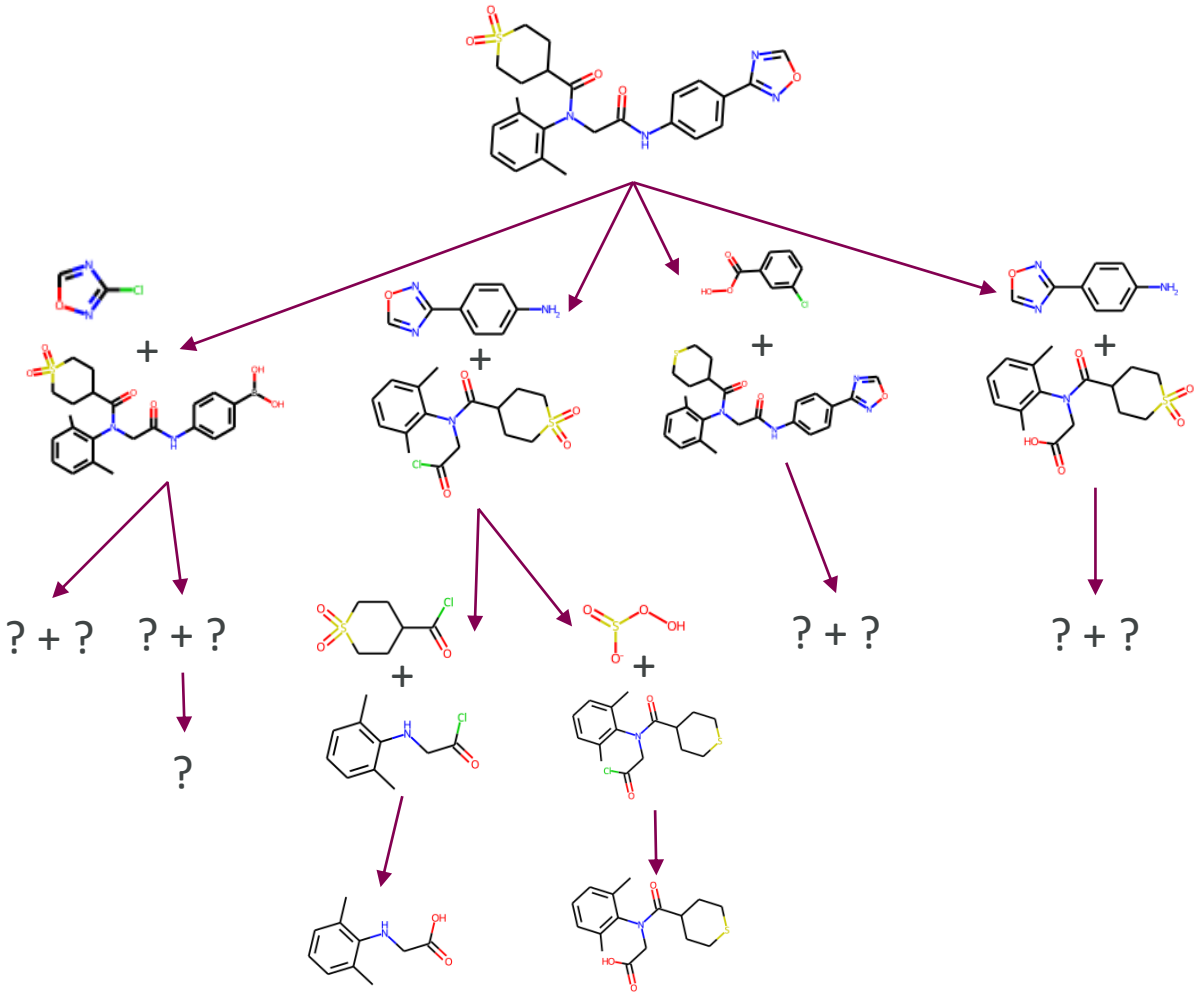
Agenda

- Retrosynthesis predictions
- Route similarity: tree edit distance
- Route similarity: ML model
- Comparing retrosynthesis algorithms
- Concluding remarks

Retrosynthesis predictions



Retrosynthesis problem: an overview



To find routes, we need

- To learn chemistry
  - Define the rules
  - Decide what rules are best
- To search effectively

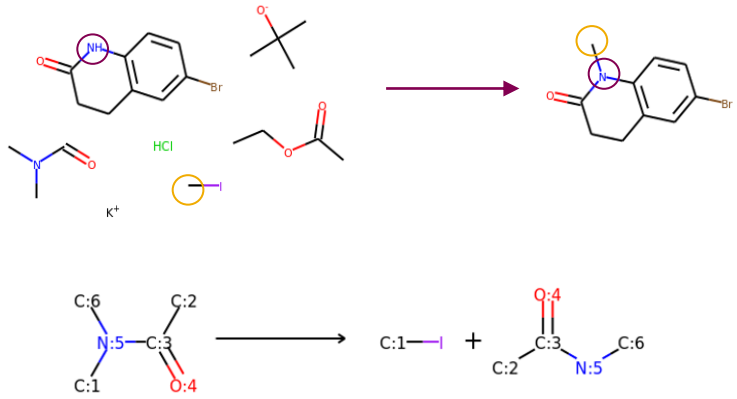
To analyze routes, we need

- To compare routes
- To group routes

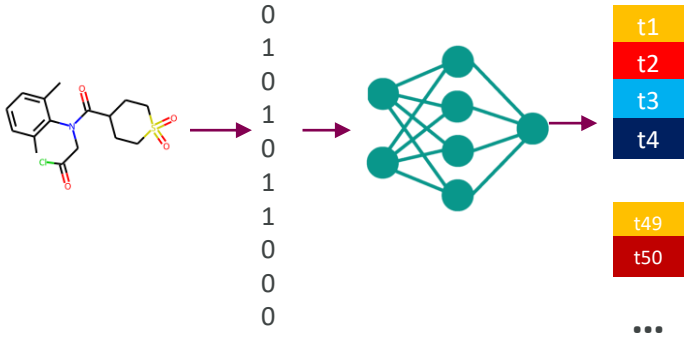
One-step models

Template-based approach

1. Extract reaction rules from known reactions

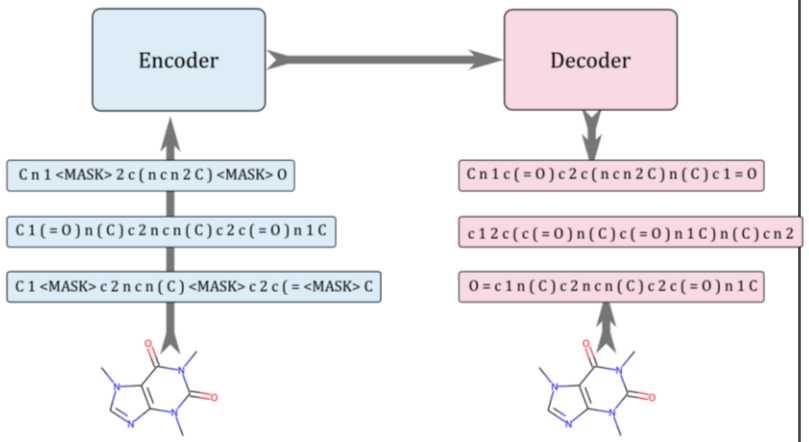


2. Train a recommendation model

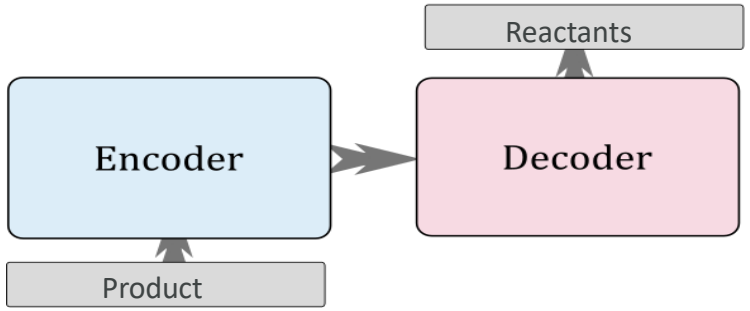


Template-free approach

1. Pretrain a *transformer* model to learn the SMILES language



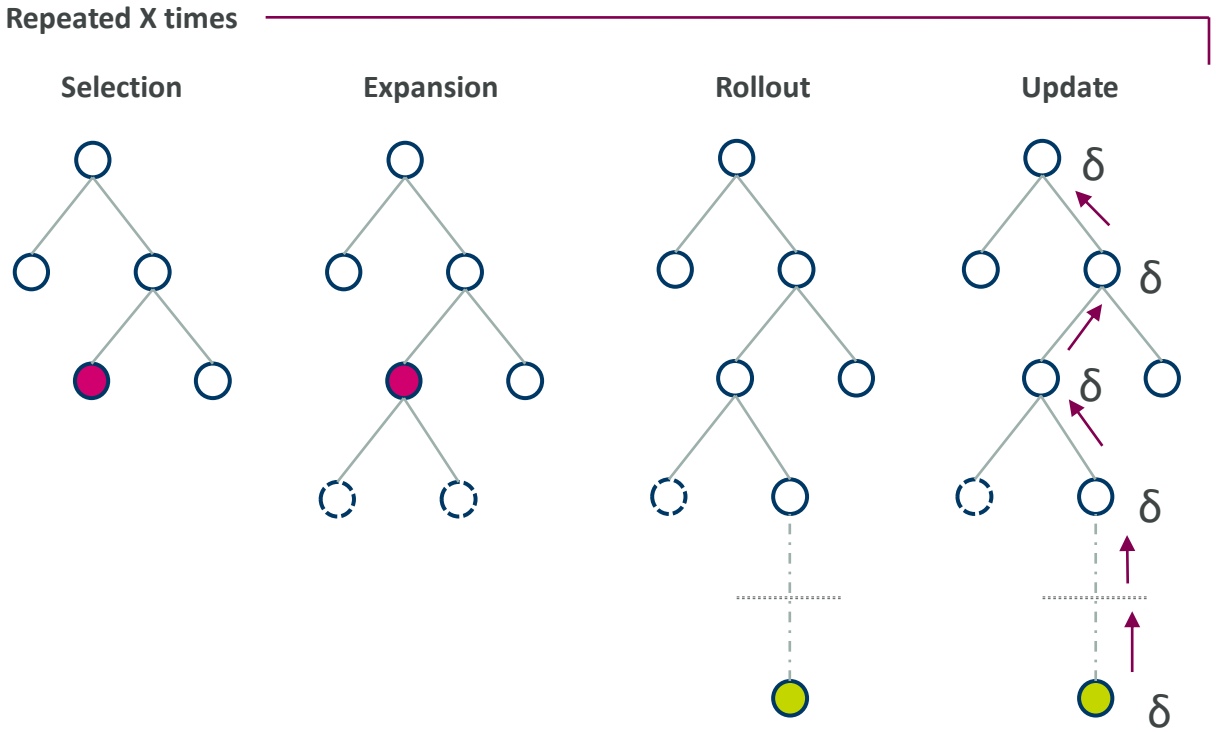
2. Fine-tune a model to perform retrosynthesis



State of the art

- New models are proposed on an almost a weekly basis
- Difficult to obtain an overview of what is available
- Benchmarked using subsets of USPTO

Multi-step retrosynthesis



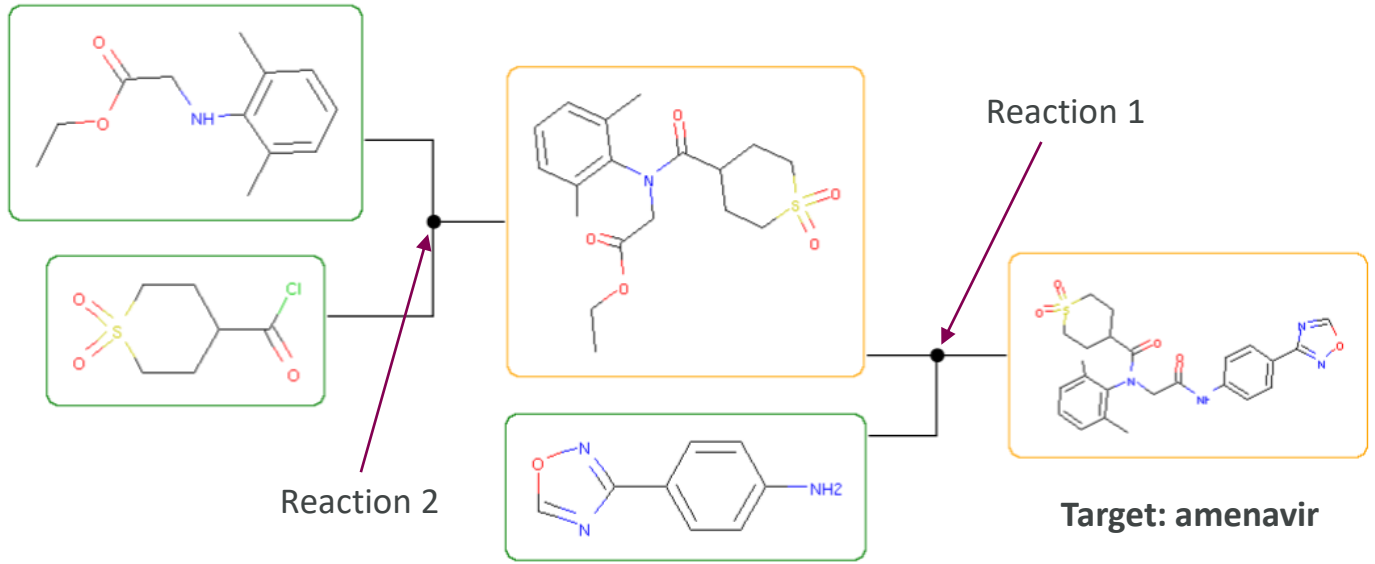
- Use a search algorithm to iteratively find a solution
- Examples: Monte Carlo Tree Search, Proof-Number Search, A*
- Uses one-step model at each iteration to expand the search tree
- Different scores to balance exploitation and exploration
- No standard way to compare algorithms

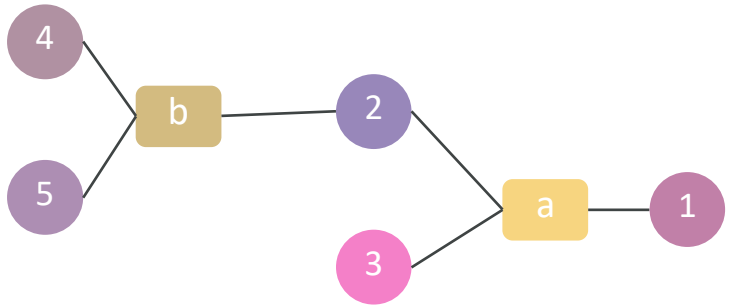
Where are we?

- Route predictions are good at generating ideas for a chemists
- Not sufficiently mature to be blindly trusted
- Outstanding challenges (a selection):
  - Reaction data quality
  - Stereochemistry
  - Complex molecules / cores
  - Human-likeness
- Improving one-step model top-1 accuracy with 2% will not solve this
- Need to evaluate multi-step route predictions

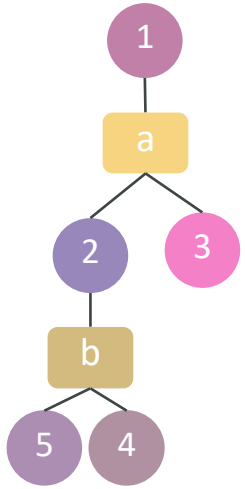
Route similarity: tree edit distance

Routes, synthetic trees – an anatomy





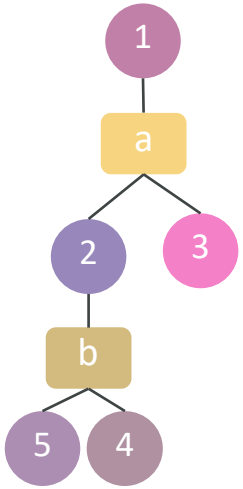
A bipartite tree



Tree edit distance

- Measures the minimum cost of transforming tree A into tree B
- Transformation can be done using 3 operations
  - Add a node
  - Delete a node
  - Replace a node
- Addition and deletion cost set to unity
- Replacement cost set to Tanimoto distance
- For ordered trees there exists a lot of algorithm to solve this problem
- For unordered trees it is NP complete

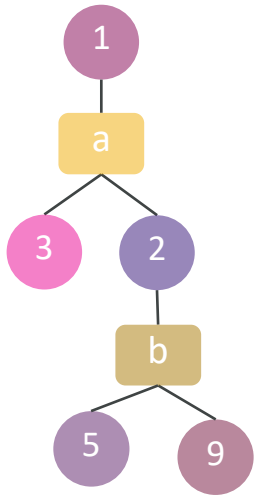
Tree edit distance



Replace *9 with 4*

Cost: 0.05

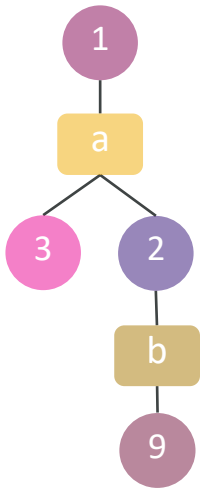
Sum: 4.45



Insert *5*

Cost: 1

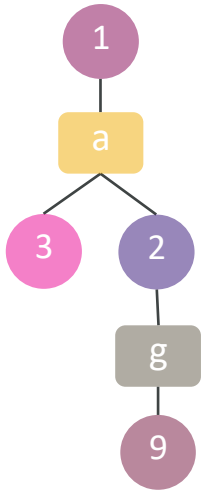
Sum: 4.4

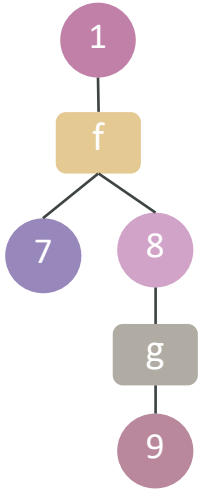


Replace *g* with *b*

Cost: 0.5

Sum: 3.4

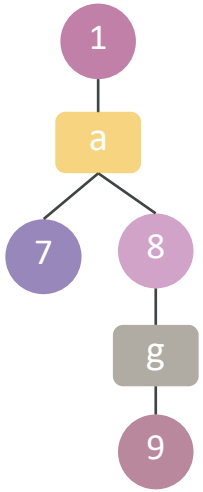




Replace *f* with *a*

Cost: 0.8

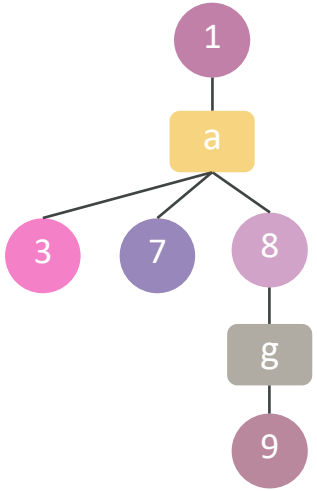
Sum: 0.8



Insert 3

Cost: 1

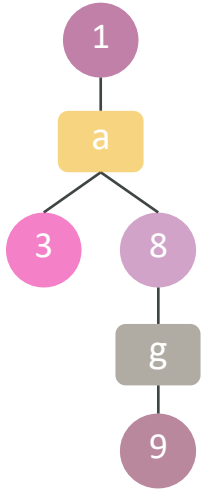
Sum: 1.8



Delete *7*

Cost: 1

Sum: 2.8



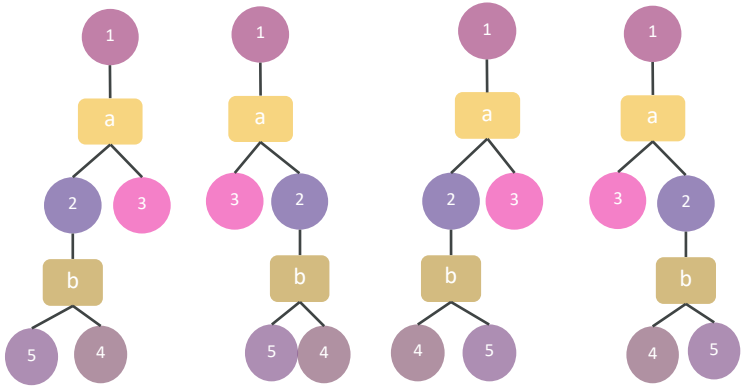
Replace *8* with *2*

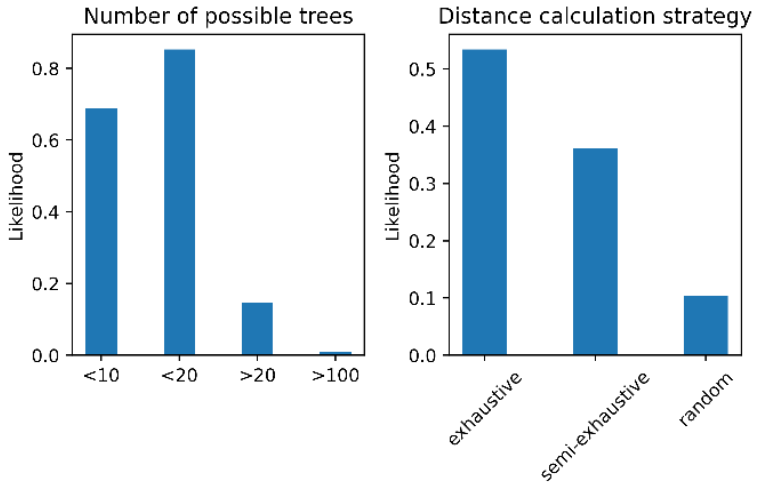
Cost: 0.1

Sum: 2.9

Heuristics on top of TED

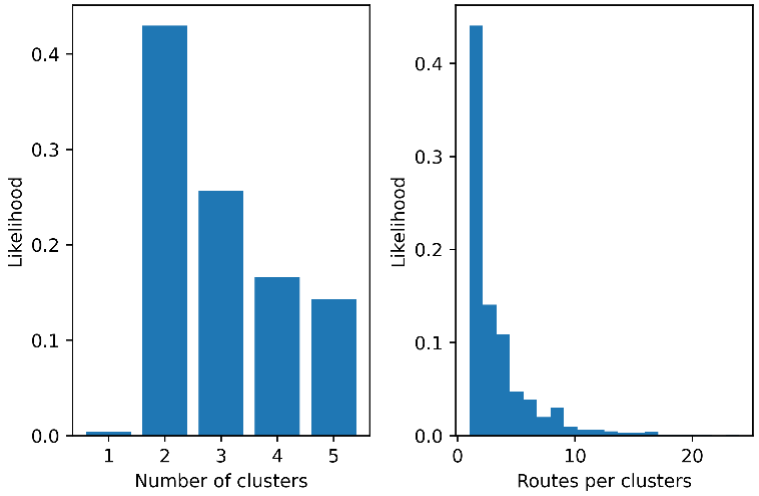
- Synthetic trees are not ordered trees
- Synthetic trees are typically short and each nodes have few children
- For small trees we can enumerate all possible trees
- This is basis for 3 TED calculation strategies: exhaustive, semi-exhaustive and random

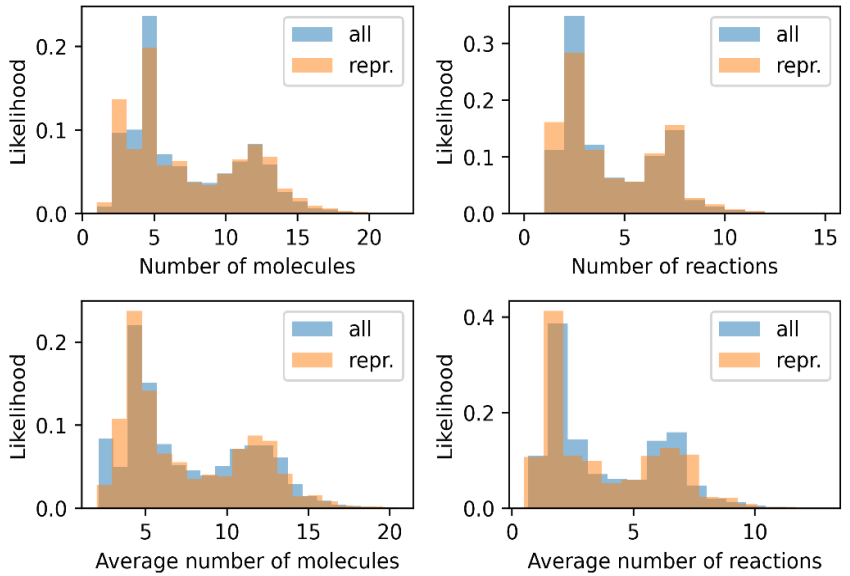




Route clustering

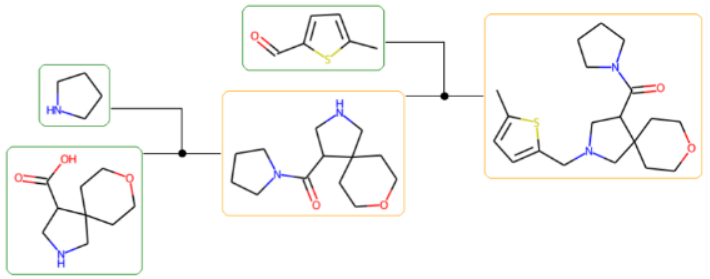
- Use TED pair-wise distances as basis for clustering
- Use hierarchical clustering with single linkage
- Optimize number of clusters using Silhouette approach
- Benchmark on 5,000 random compounds from ChEMBL



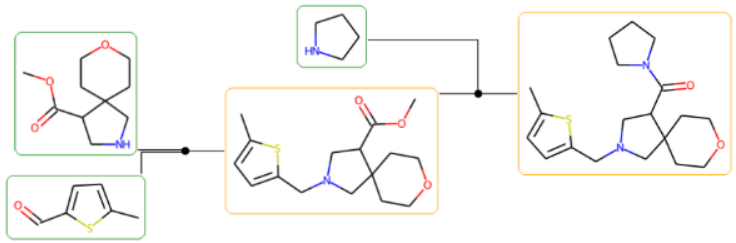


Route clustering – qualitative evaluation

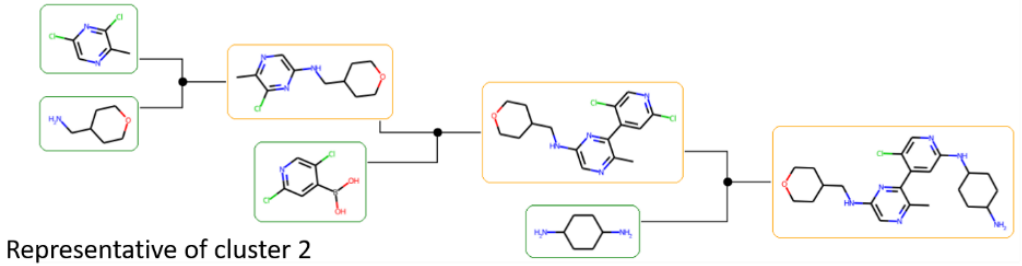
Representative of cluster 1



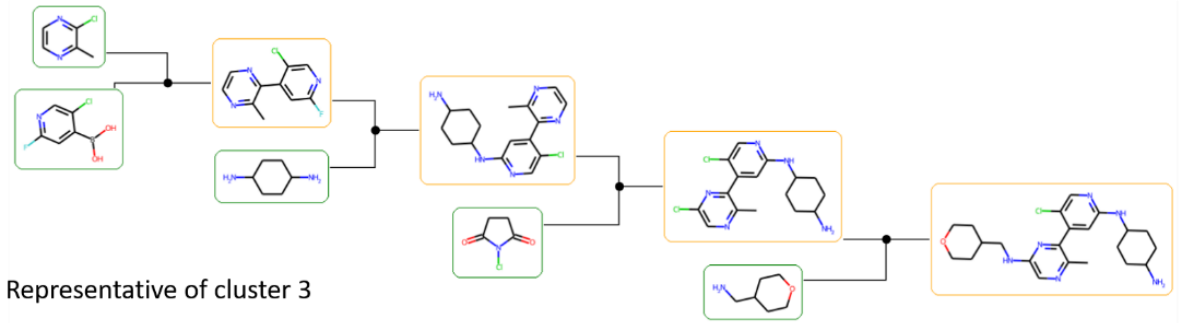
Representative of cluster 2



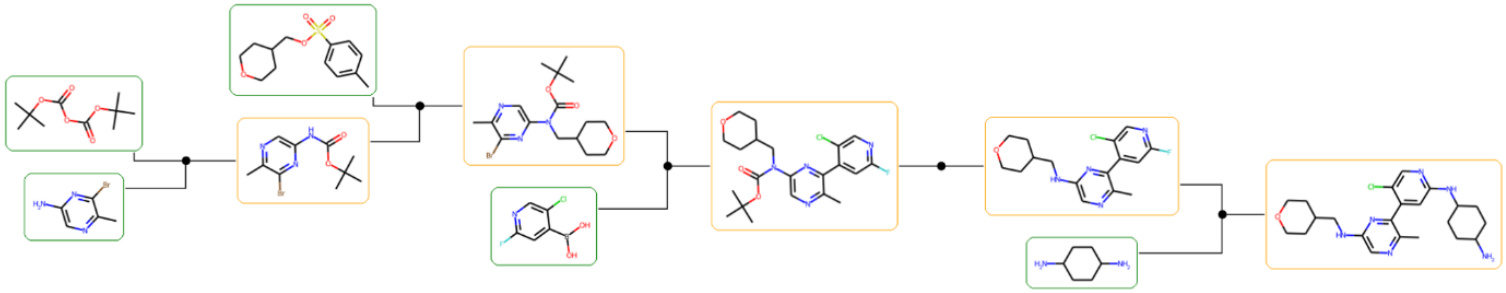
Representative of cluster 1



Representative of cluster 2

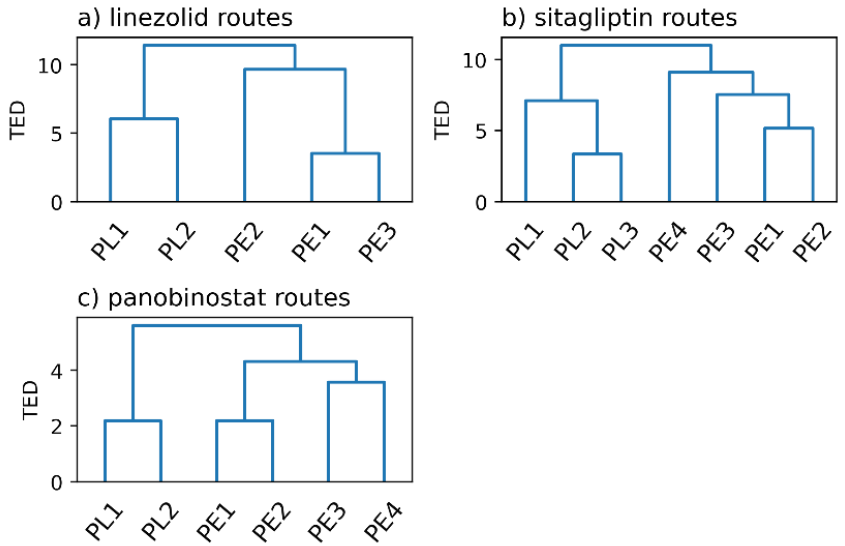


Representative of cluster 3



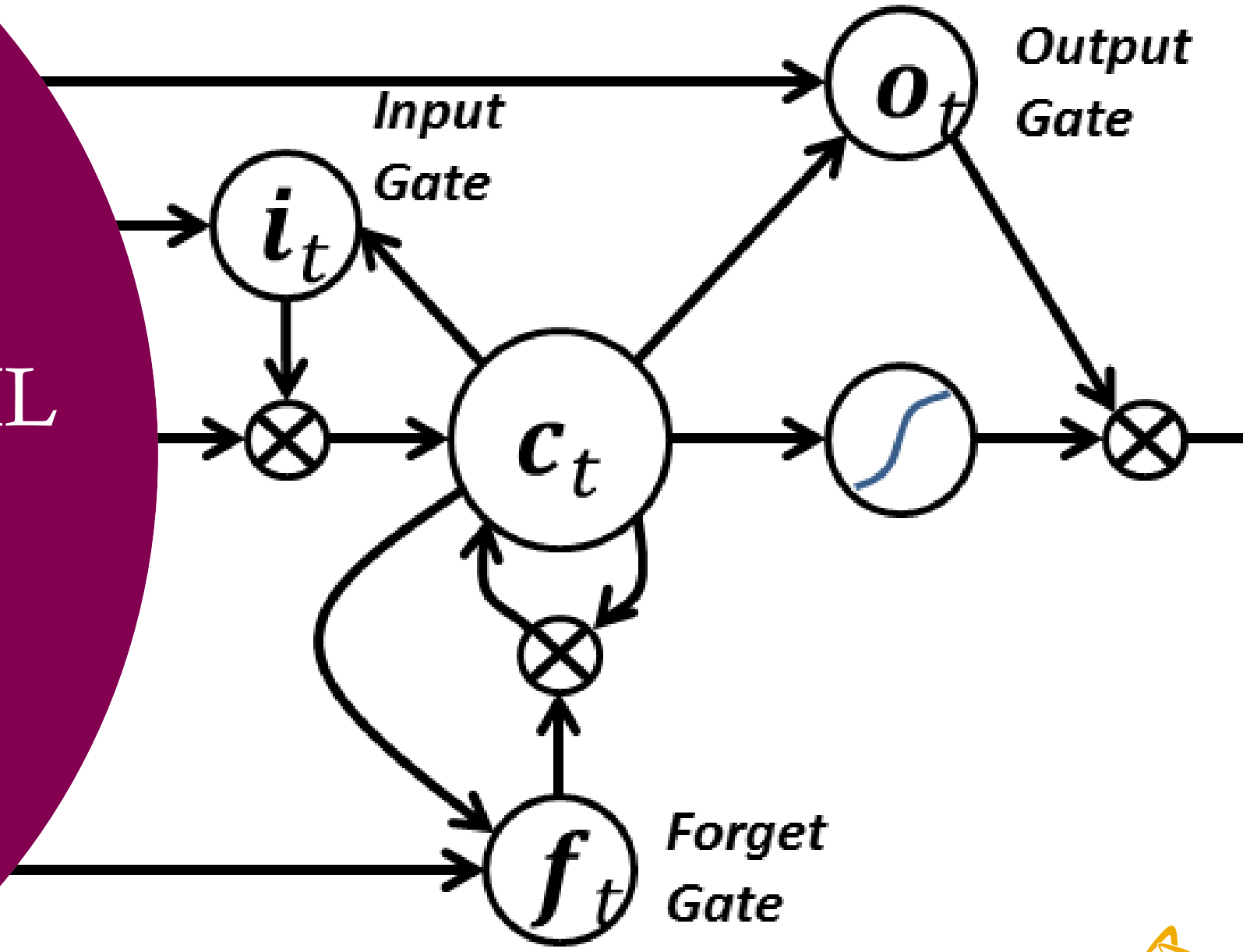
Patent evading routes

- Routes predicted with Chematica software
- Routes manually designated patent-like (PL) or patent-evading (PE)
- Routes clustered with TED approach



Molga, K.; Dittwald, P.; Grzybowski, B. A. Navigating around Patented Routes by Preserving Specific Motifs along Computer-Planned Retrosynthetic Pathways. *Chem* **2019**, *5*, 460–473.

Route similarity: ML model

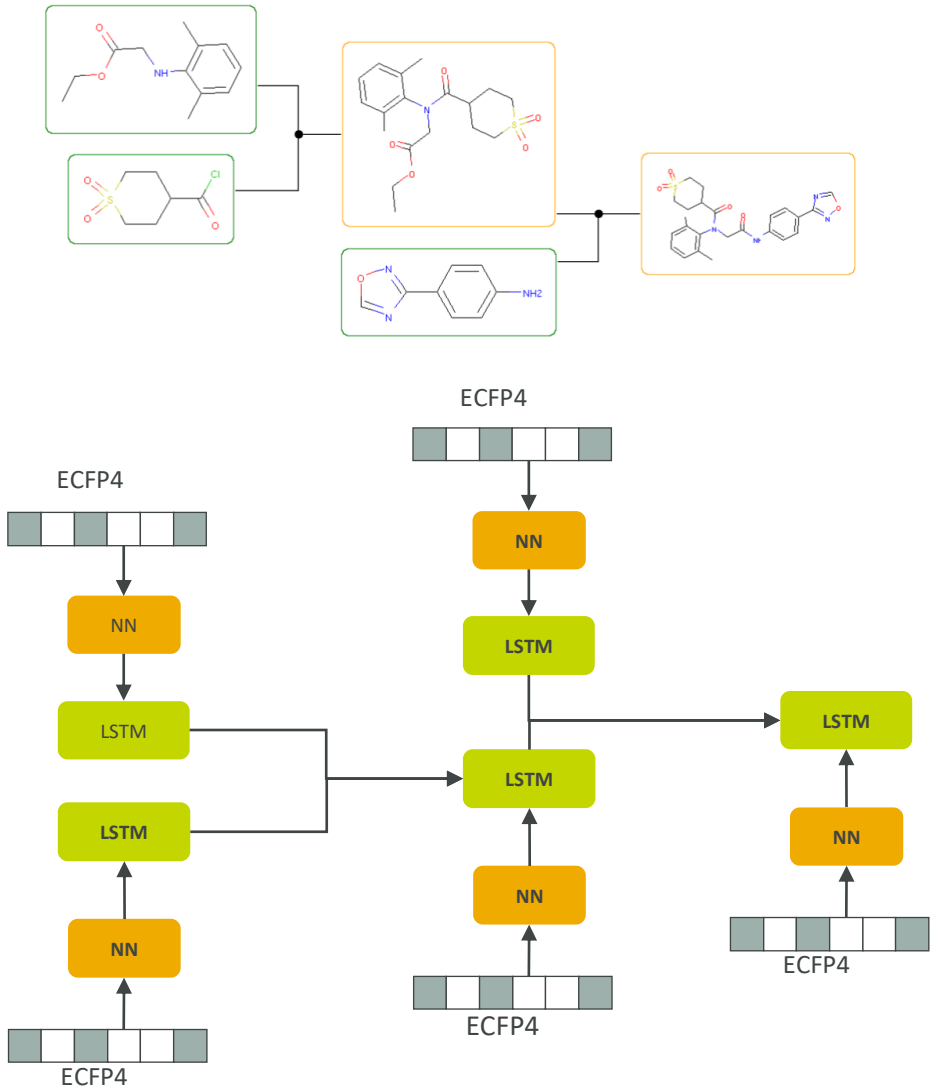


The problem with TED

- Tree edit distance calculations is unfortunate slow in worst case
- Solution: train a machine learning model to reproduce TED

		TED	
	Mean # pairs	mean time	worst time
ChEMBL-5k	63.90	6.45	213.95
ChEMBL-10k	64.75	5.54	211.89
GDB-MedChem	50.76	6.71	156.83
GDB-ChEMBL	52.97	6.82	157.30
ChEMBL-5k (100 routes)	4770.86	377.03	4215.92
ChEMBL-5k (all solved routes)	1217.81	72.62	3806.78

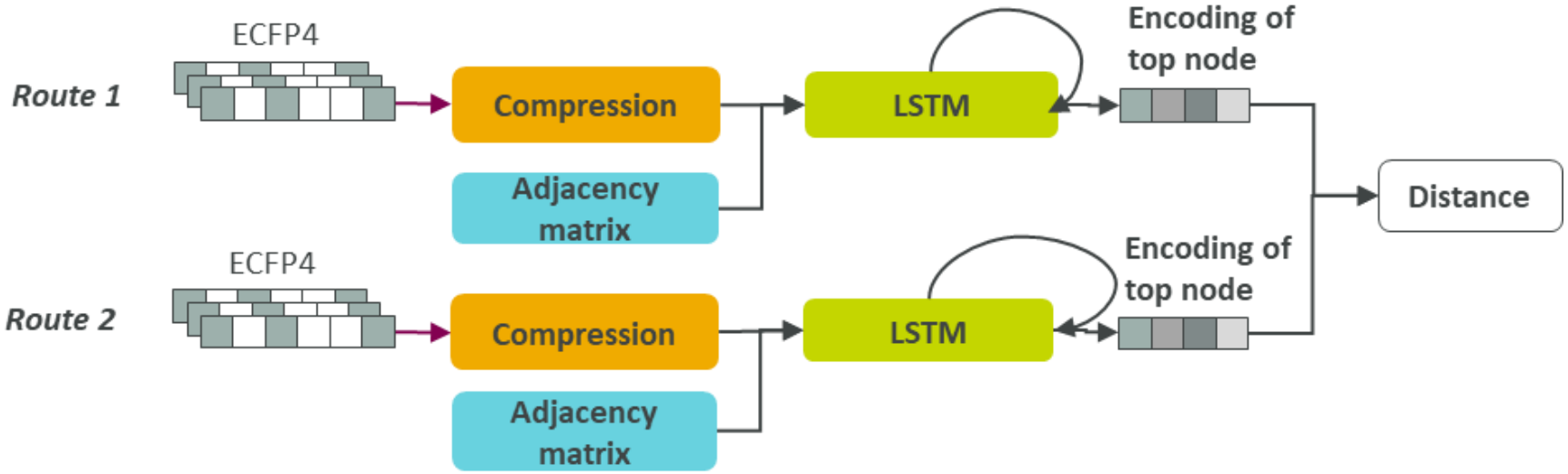
LSTM-based model



- At each molecule, a long short-term memory node is placed
- Takes input from children LSTM nodes as well as a compressed fingerprint
- LSTM output of top-node (target molecule) is an encoded representation of the route
- Reactions are ignored

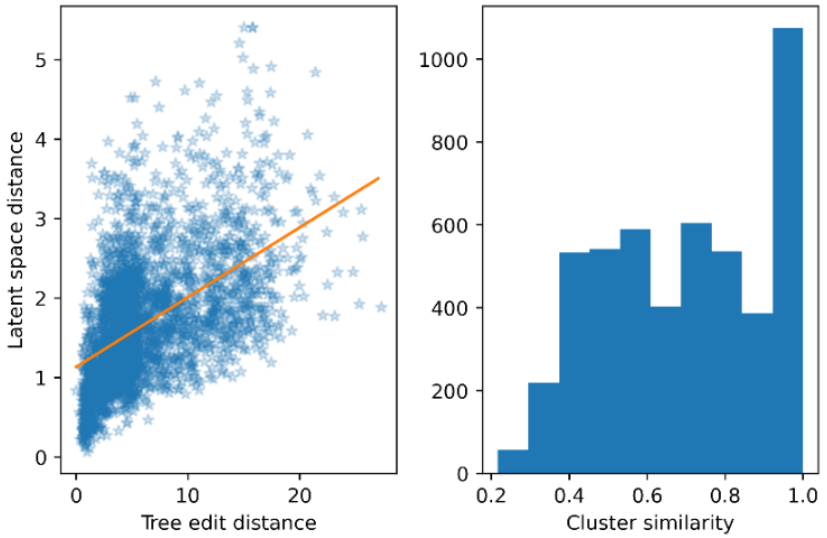
Twin network training

- A twin network architecture was trained to reproduce TED
- First route is feed through the LSTM model, followed by the second route
- Euclidean distance between encodings of top-nodes is the distance between the routes



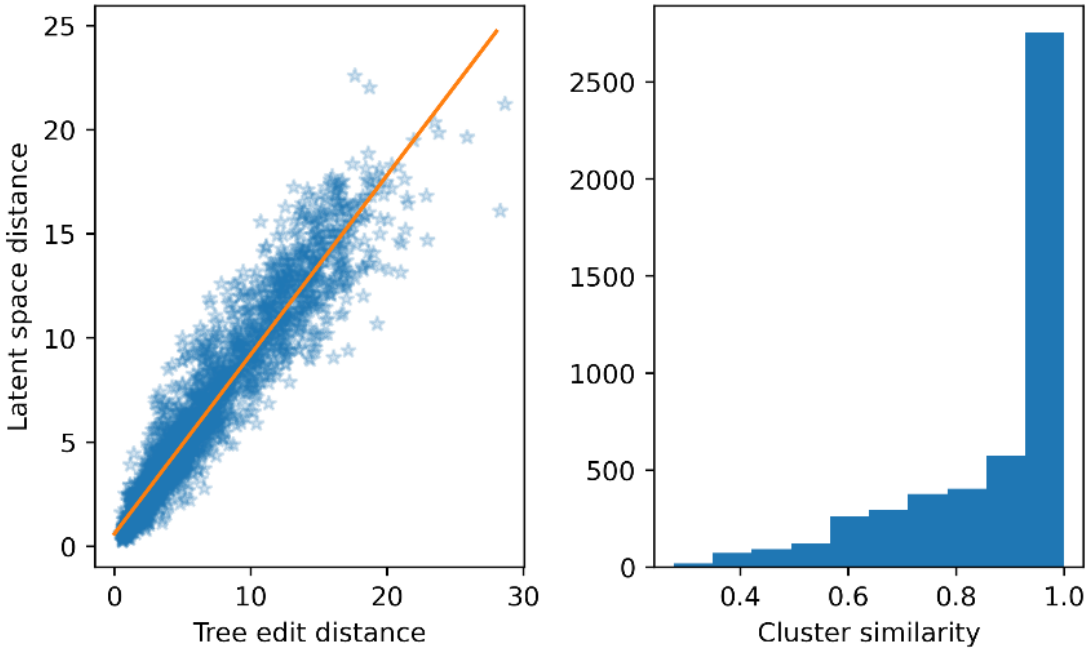
Side-bar: MIT model

- MIT proposed a similar model for representing synthetic routes
- They trained the model to distinguish between human and predicted routes
- Not guaranteed that model produce good clusters



Benchmarking of LSTM-based model

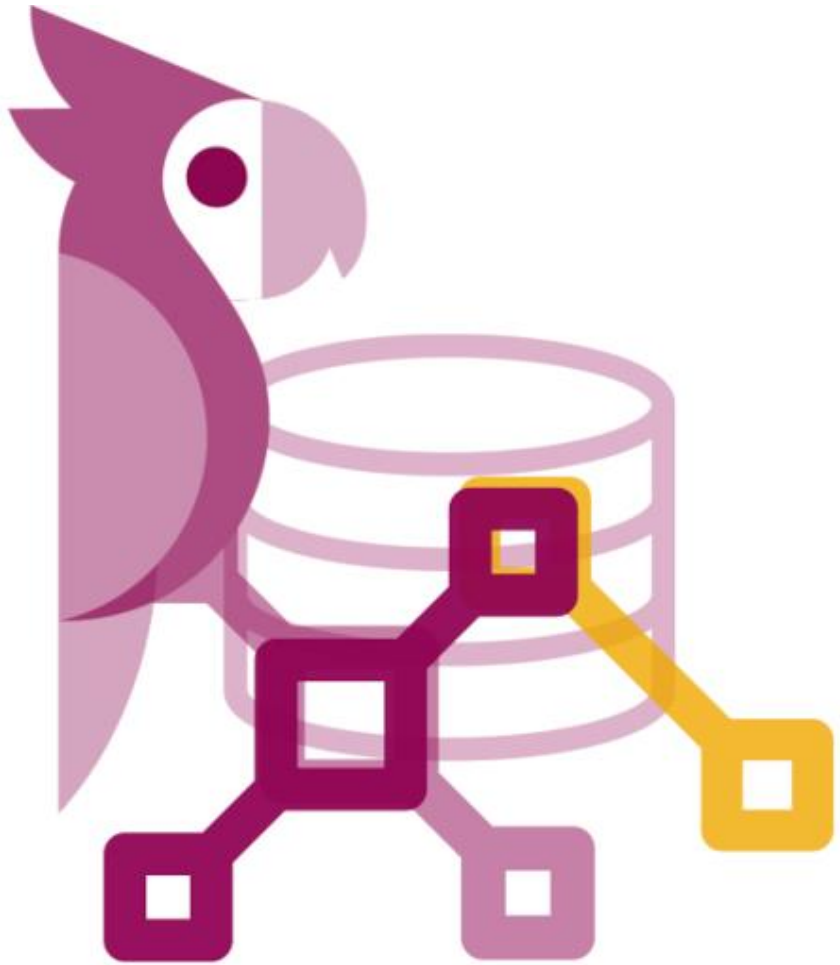
- Trained two models based on routes for either 5k or 10k ChEMBL compounds
- Validate on various sets of molecules
- The latent-space distance is correlated with the tree edit distance
- Cluster similarity is high



Benchmarking (2)

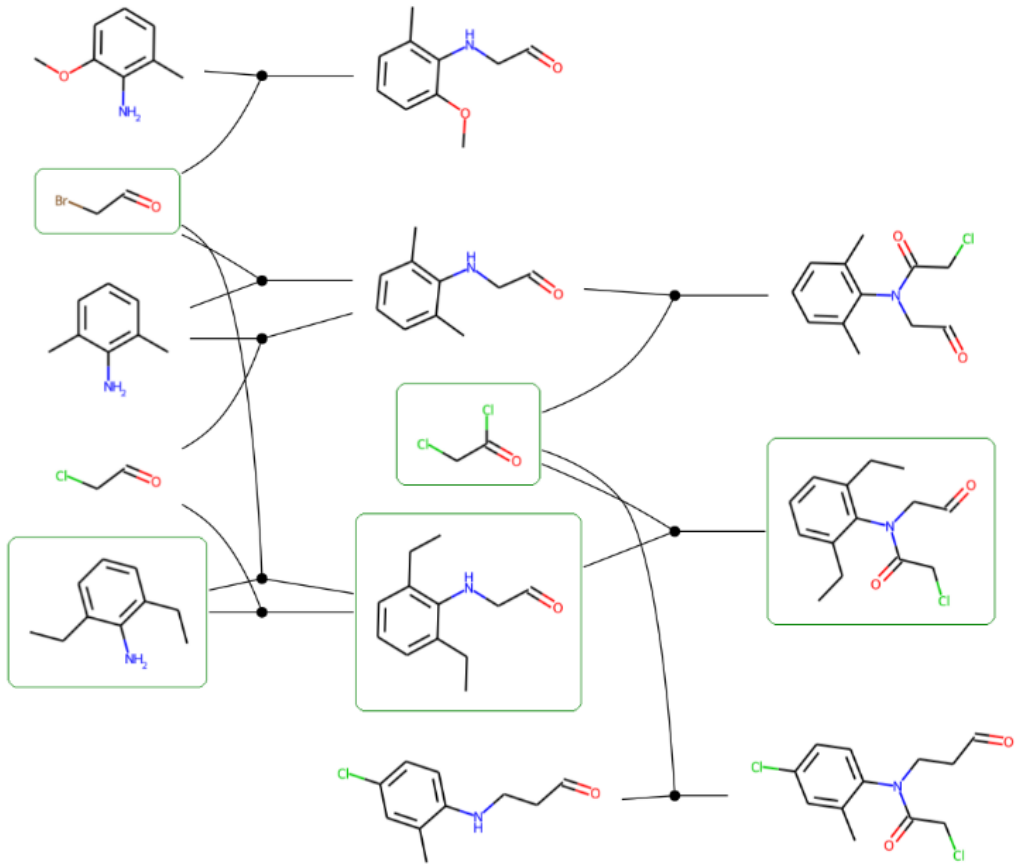
Compound set	Expansion policy	Distance		Cluster	
		R	MAE	Mean similarity	Median similarity
ChEMBL-5k	USPTO	0.95	0.92	0.88	0.97
ChEMBL-10k	USPTO	0.95	1.03	0.87	0.95
GDB-MedChem	USPTO	0.92	1.66	0.87	0.96
GDB-ChEMBL	USPTO	0.92	1.61	0.87	0.95
GDB-MedChem	Reaxys	0.92	1.55	0.88	1.00
GDB-ChEMBL	Reaxys	0.92	1.53	0.88	1.00

Comparing retrosynthesis algorithms



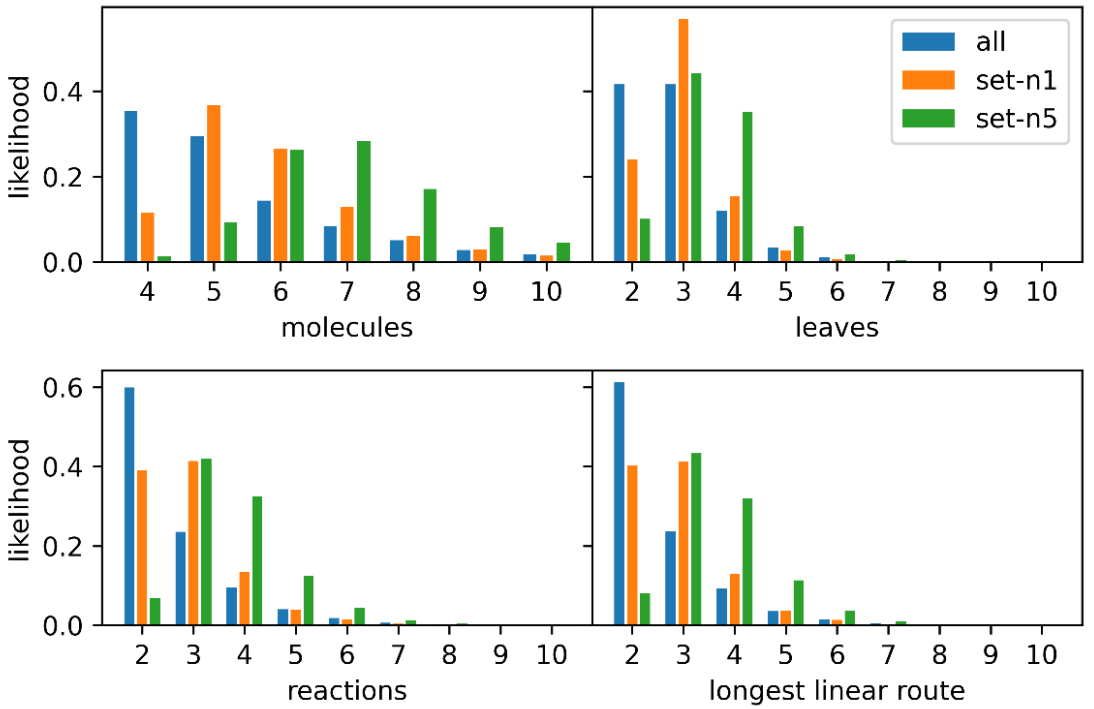
Route extraction from USPTO

- Start from curated UPSTO data from Thakkar et al. (2020)
- Extract routes from networks made for each individual patent
- Discard routes with single leaf (transformations)
- Keep *n* routes from each patent
- Discard routes that overlap
- Select 10,000 diverse routes



Route extraction from USPTO (2)

- Extracted 150,000 routes with more than one starting material (leafs)
- Extracted 2 sets of 10,000 routes for benchmarking with *n*=1 and *n*=5 respectively
- Set-n5 is enriched in longer *and* convergent routes
- **Stock taken as leaves / starting material of the 10,000 reference routes**



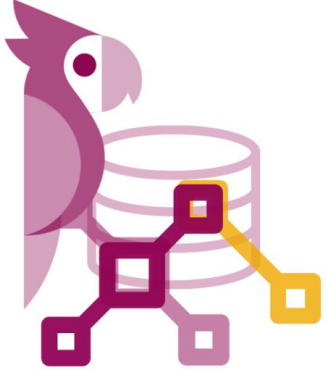
Quality metrics

- Search time
- Number of solved routes, i.e., routes that have all starting material in stock

- Route quality metric based on TED between predictions and reference route
  - Measures the smallest TED between reference route and top-*n* predictions
  - When TED = 0, the prediction is identical to reference route
  - Average of TED = 0 gives top-n accuracy

- Route diversity metrics based on optimal number of clusters
  - Calculate pair-wise distances using fast ML-model
  - Optimize number of clusters with Silhouette method

PaRoutes – a framework



Data

- A subset of the USPTO database with reactions that can be used to train a one-step model
- ~150K routes extracted from the USPTO database, which can be used for machine learning tasks
- set-n1 consisting of a diverse set of 10,000 routes which show a similar distribution in the number of molecules and reactions as the 150K routes
- set-n5 consisting of a diverse set of 10,000 routes that are longer and enriched in convergent routes
- stock-n1 consisting of the 13,633 leaves molecules in set-n1 and should be used as a stock together with set-n1
- stock-n5 consisting of the 13,783 leaves molecules in set-n5 and should be used as a stock together with set-n5

Scripts

- Program to compute top-n accuracies
- Program to calculate optimal number of clusters

https://github.com/MolecularAI/paroutes

Example use of PaRoutes framework

- We didn’t want to benchmark all available algorithms and all possible variations of such algorithms
- We choose three algorithms implemented in the AiZynthFinder tool
  - Monte Carlo tree search
  - Retro* (with some extensions to open-source reference implementation)
  - Depth-first proof-number (DFPN) search (inspiration from a few implementations)
- Use template-based one-step model trained on USPTO data not included in the reference routes

Route prediction comparison 1

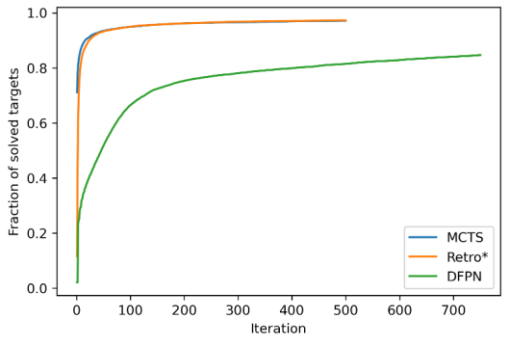
Search method	Route set	Solved targets	Search time	First solution time	One-step model calls	Template applications
MCTS	set-n1	9714	303.3	8.6	3355.6	8658.2
	set-n5	9676	365.7	11.7	3615.3	8953.0
Retro*	set-n1	9726	300.7	7.0	497.4	24281.1
	set-n5	9703	349.2	10.5	498.0	24322.5
DFPN	set-n1	8475	347.3	43.0	404.5	19503.2
	set-n5	7382	297.9	53.2	414.5	19957.6

MCTS and Retro* find solutions for most targets, but DFPN struggles

The MCTS and Retro* are not complementary, they find solutions to the same compounds

MCTS and Retro* finds first solution faster then DFPN

For *set-n5,* it takes longer time



The algorithms differ in how they solve the route-finding problem. MCTS uses the one-step model more extensively.

Route prediction comparison 2

Search method	Route set	Accuracy top-1	Accuracy top-5	Accuracy top-10	Shorter route	Leaves overlap	Routes extracted	Number of clusters
MCTS	set-n1	0.20	0.55	0.61	0.44	0.68	273	68
	set-n5	0.09	0.34	0.42	0.59	0.62	272	77
Retro*	set-n1	0.17	0.48	0.54	0.44	0.68	264	68
	set-n5	0.08	0.30	0.38	0.61	0.63	149	39
DFPN	set-n1	0.19	0.33	0.33	0.45	0.63	6	2
	set-n5	0.08	0.14	0.14	0.65	0.55	6	2

For MCTS, the reference route is top-ranked for 1/5 of the targets

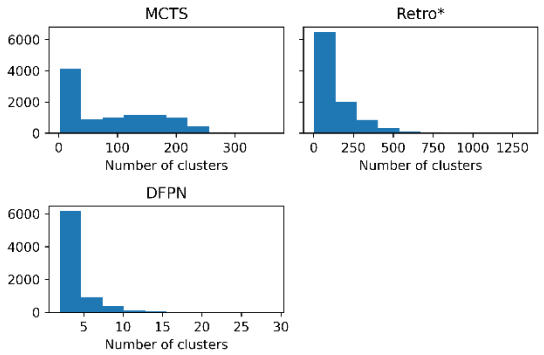
For 60% of the targets, the reference route is found in top-10

All methods are roughly equal on top-1, but differ for higher *n*

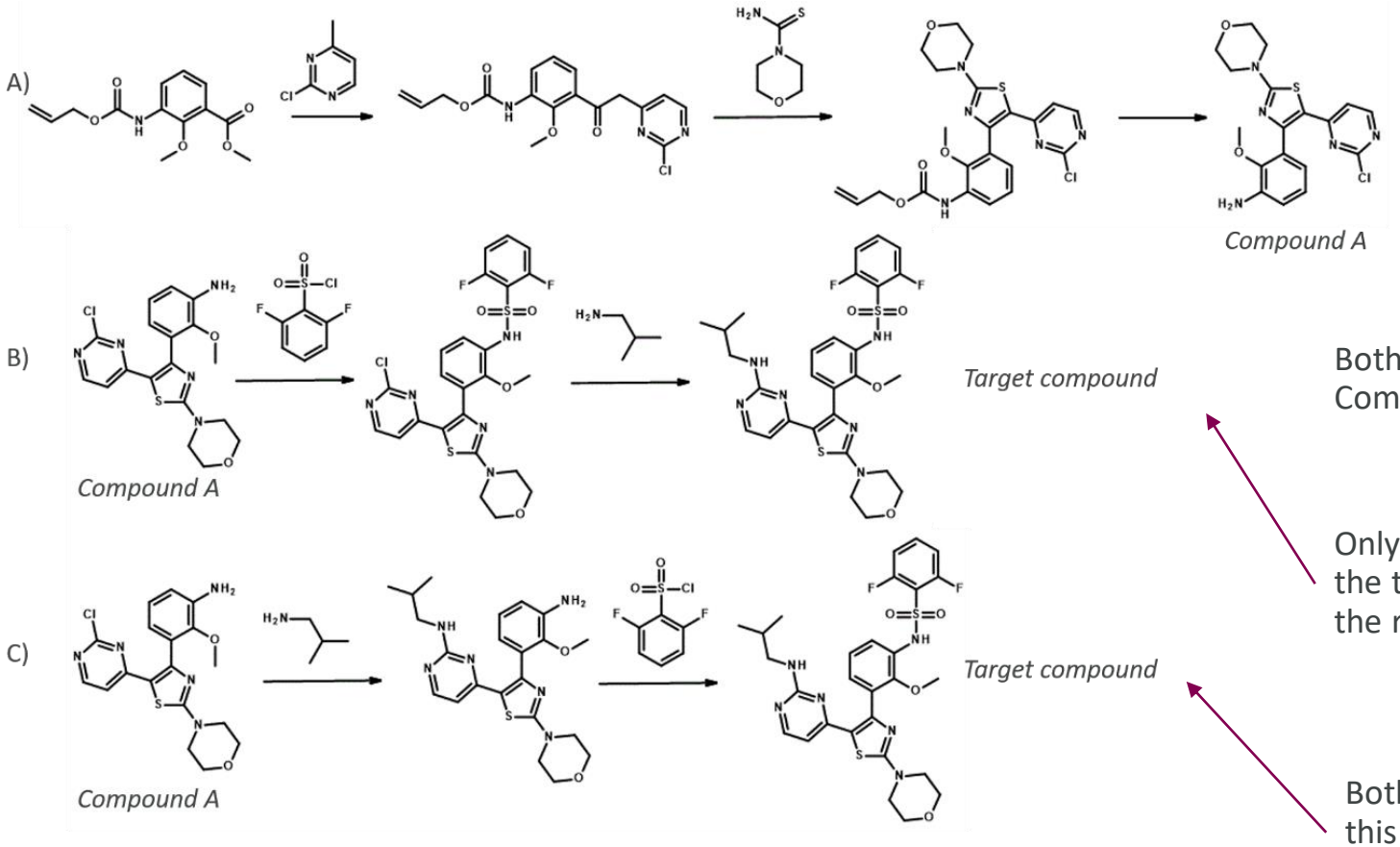
The accuracy for set-n5 is poorer

For set-n1, MCTS and Retro* have the same diversity

With DFPN, very few routes are extracted



Case-study



Both Retro* and MCTS identifies Compound A as an intermediate

Only MCTS identifies this route to the target compound, which is also the reference route

Both Retro* and MCTS identifies this alternative route. For Retro* this is the top-ranked route.

Conclusions and outlook

- In it is current implementation, DFPN cannot be recommended
- MCTS and Retro* are very similar, but MCTS seems to have slightly higher accuracy and produce more diverse output for more targets
- It is hard to declare a clear winner, and that is perhaps beside the point

- Improvements to PaRoutes
  - How to handle multiple feasible reference routes?
  - More metrics (forward prediction scores, accuracy as a function of time)
- Good starting point for benchmarking route preditions

Concluding remarks

Final remarks

- We have developed a model for computing distances between synthetic routes
- Applications include:
  - Clustering output from retrosynthesis tools
  - Identification of patent evading routes
  - Comparing route predictions
- A framework for comparing route predictions will
  - Provide a common baseline for judging improvements
  - Increase transparency of method
  - Help us determine limiting factor(s) for reaching predictive level

Acknowledgements

- Molecular AI department

- Esben Bjerrum

- Ola Engkvist

- Ross Irwin

PaRoutes

Yasmine Nahal

- Users of AiZynthFinder (both internal and external)

Thanks for listening!