Electronic Supplementary Material (ESI) for Digital Discovery.

Supporting Information

PaRoutes: towards a framework for benchmarking retrosynthesis route predictions

Samuel Genheden and Esben Bjerrum

Molecular AI, Discovery Sciences, R&D, AstraZeneca Gothenburg, SE-431 83 Mölndal, Sweden

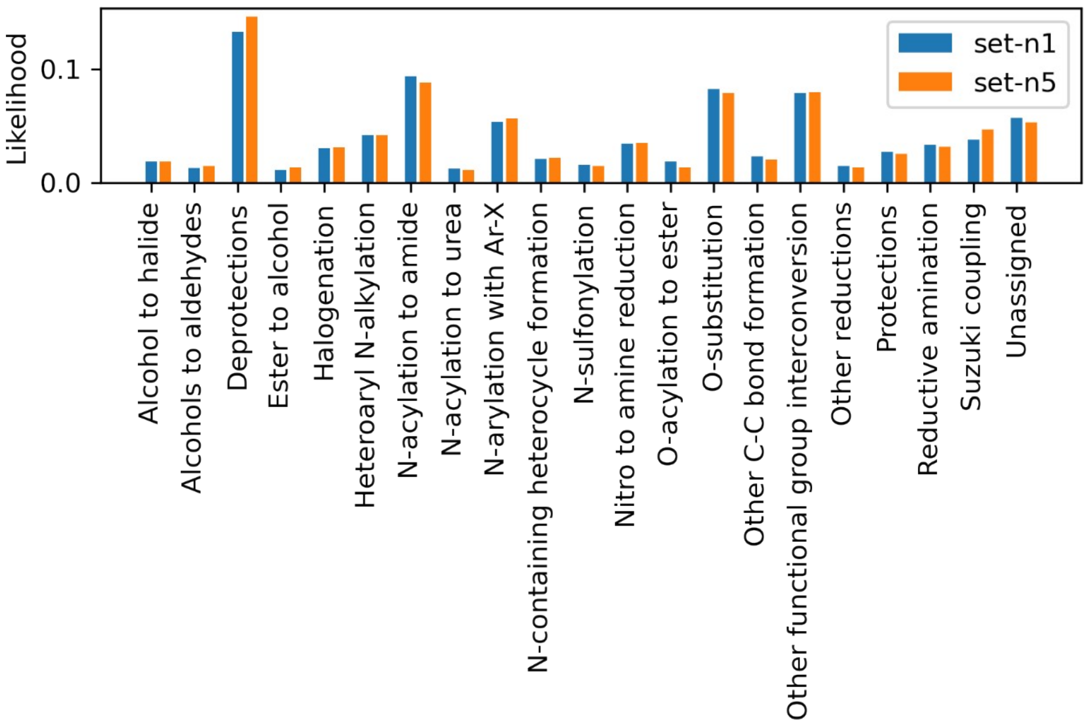

Figure S1 – Top-20 most common reaction classes in the set-n1 and set-n5 reference routes and their likelihood (occurrence divided by the total number of reactions). There are 21 reaction classes in the figure because top-20 is not the same for both sets. There are 28,951 and 36,029 reactions in the set-n1 and set-n5 reference routes, respectively.

.

Table S1 – Cross comparison on ability to find a solution for the set-n1 routes

		Found by			
Method 1	Method 2	Both	Method 1	Method 2	Neither
MCTS	Retro*	0.96	0.01	0.01	0.02
MCTS	DFPN	0.85	0.13	0.00	0.03
Retro*	DFPN	0.85	0.13	0.00	0.03

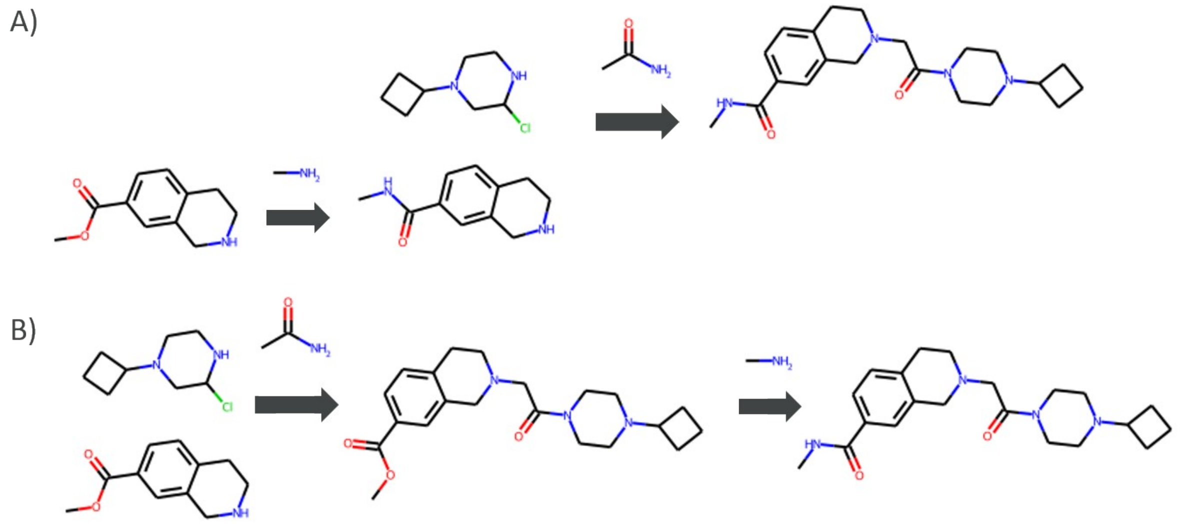

Figure S2 – Example of top-ranked route for a target where Retro* and DFPN found the reference route but MCTS was unable to. A) Route to synthesize the target as extracted from the US20070232591A1 patent, which was recovered by Retro* and DFPN. B) Alternative route to synthesize the target predicted by both MCTS. The one-step model probability and rank of the template applied on the target in route A) was $\sim 3^{-3}$ and 18, respectively whereas the probability and rank of the template applied on the target in route B) was $\sim 2^{-2}$ and 7, respectively. Thus Retro* and DFPN were able to utilize a template with much lower rank and could recover the reference route, whereas MCTS deemed this template to be too unlikely to continue exploring.

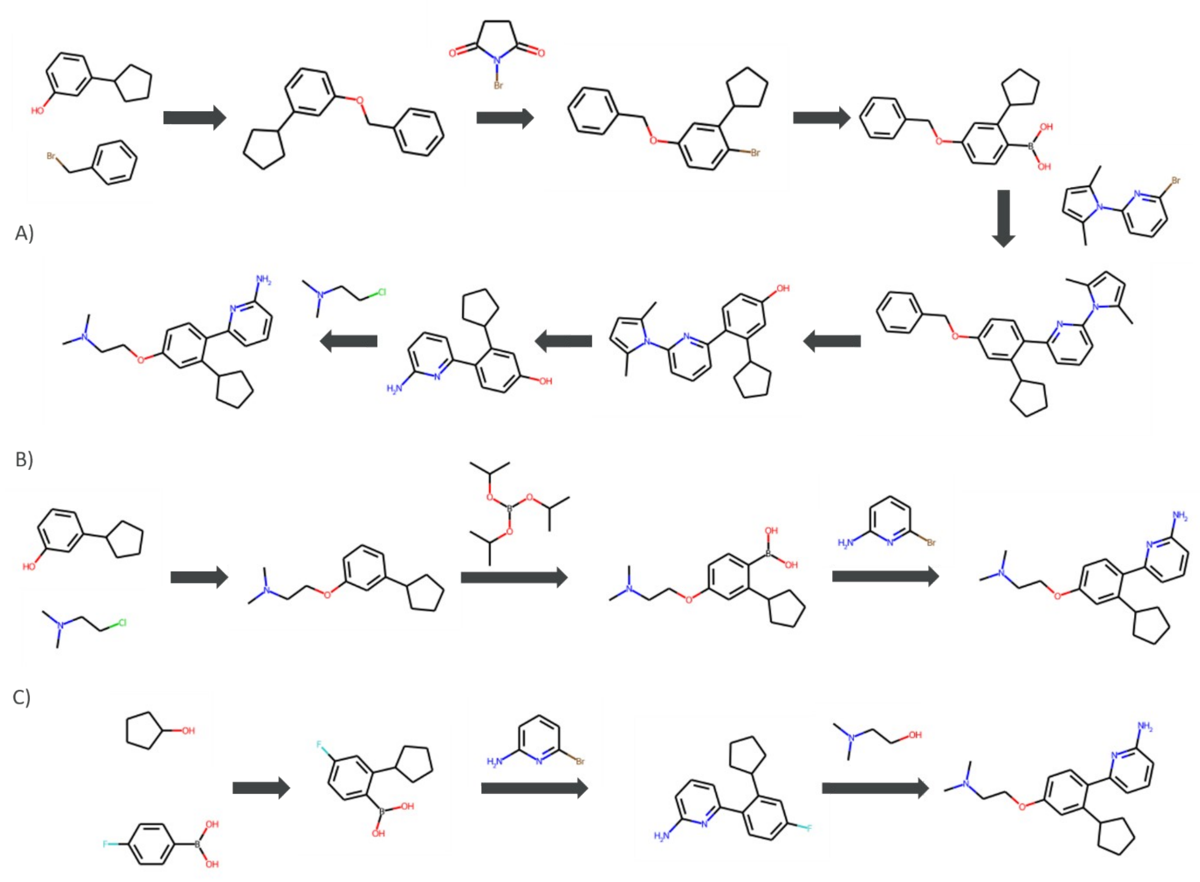

Figure S3 – Example of reference and top-ranked routes for a target where no algorithm could recover the reference route. A) The reference route extracted from patent US20030013720A1, B) top-ranked route from MCTS, C) top-ranked route from Retro*. Both MCTS and Retro* were able to find a shorter route to the target, but both routes will have selectivity issues. In B) there is a selectivity issue with the second step and in C) there is a selectivity issue with the first step.

Table S2 – Overview of the search algorithms tested in this work. The boxes describe the main statements in one iteration of each of the algorithms. For clarity many of the extra checks of node states and details of various calls has been omitted.

MCTS	Retro*	DFPN

`# Select leaf`
`leaf <- root`
`while leaf is expanded and leaf is not terminal`
`  select child with best UCB score`
`  leaf <- child`
 	
`Expand leaf with one-step model, add children nodes`
 	
`# Rollout`
`while leaf is not terminal`
`  select child with best UCB score`
`  expand child with one-step model`
`  leaf <- child`
 	
`backpropagate reward of leaf`
	`Select leaf with minimum estimated cost from all expandable leaf-nodes`
	
	`Expand leaf with one-step model, add AND/OR sub-trees`
	
	`Update the cost of all ancestor nodes of the selected leaf given the cost of the added molecules`
		`if first iteration`
		`  frontier <- root`
		
		`# Select new frontier`
		`while frontier is not expandable`
		`  update proof and disproof numbers of frontier`
		`  if frontier cannot be searched`
		`    frontier <- parent of frontier`
		`  else`
		`    find child with minimum proof number`
		`    frontier <- child`
		
		`# Expanding molecule node`
		`expand frontier, adding reaction nodes`
		`update proof and disproof numbers of frontier`
		`find child with minimum proof number`
		`frontier <- child`
		
		`# Expanding reaction node`
		`expand frontier adding molecule nodes`
		`update proof and disproof numbers of frontier`
		`find child with minimum proof number`
		`frontier <- child`