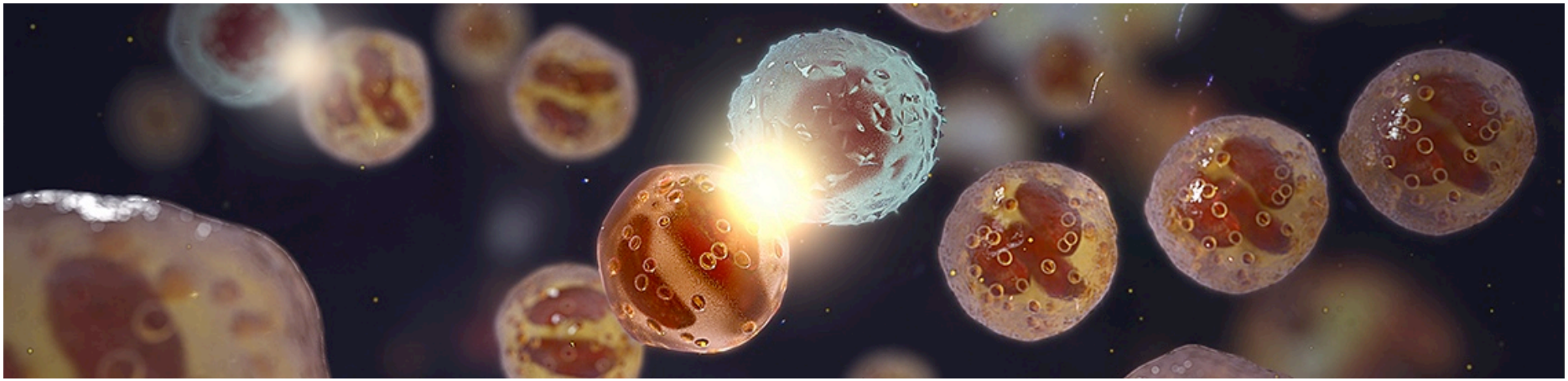
Molecular De Novo Design through Deep Reinforcement Learning

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, Hongming Chen
Hit Discovery, Discovery Sciences, AstraZeneca R&D Gothenburg

05 September 2017



Searching Chemical Space

Goal:

Find structures with certain properties

We need two things:

A way to generate structures

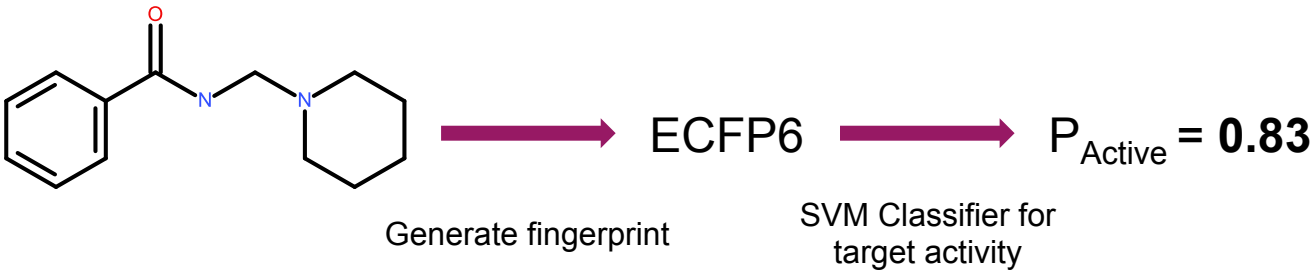
A way to rank structures – scoring function

Ideally, the scoring function should influence the way we generate structures

Scoring function

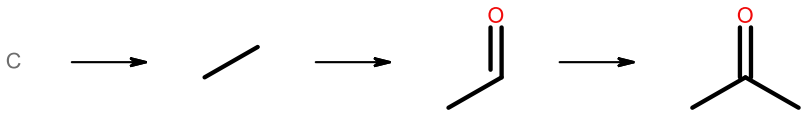
$$S: x \rightarrow y \in \boldsymbol{R}$$

X is the chemical structure that we map to a scalar value y. For example:

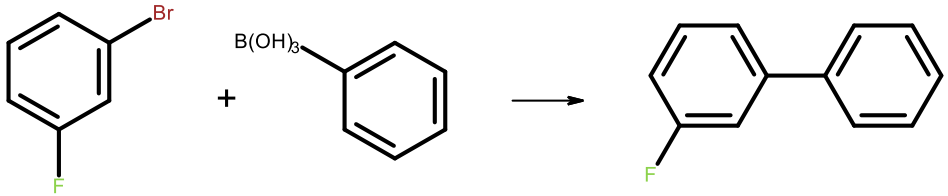


But how do we generate structures?

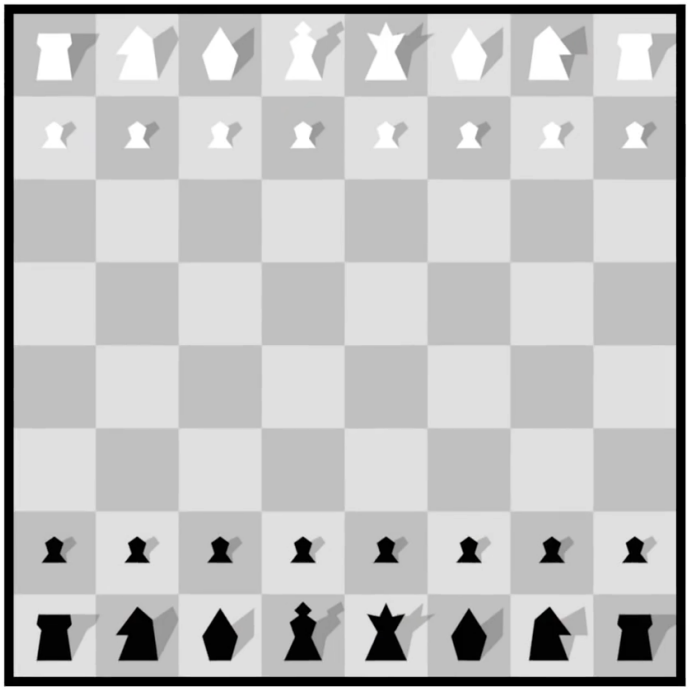
Build structures atom by atom?



Combine fragments through reaction rules?



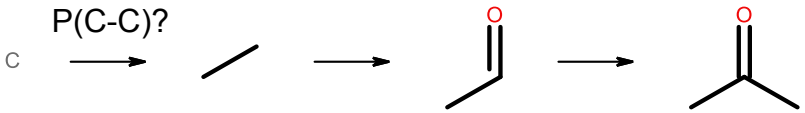
Overcoming the curse of dimensionality



But how do we generate structures?

Atom by atom

Learn transition probabilities



The model should represent a probability distribution over chemical space

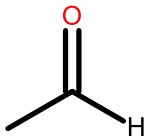
Natural language processing and SMILES

- How to represent molecular structure?
- Can borrow concepts from language processing and apply to SMILES

The → grass → is → ?

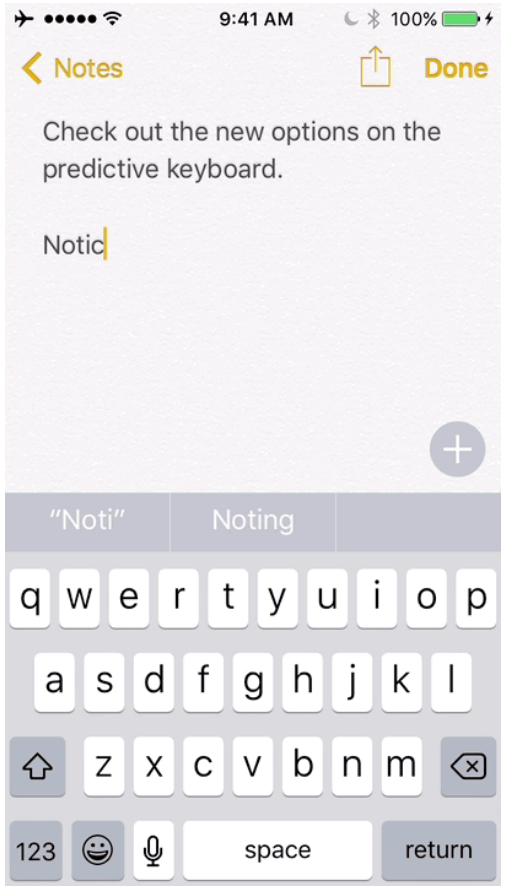
- Conditional probability distributions given context
- $P(green \mid is, grass, The)$

C → C → = → ?



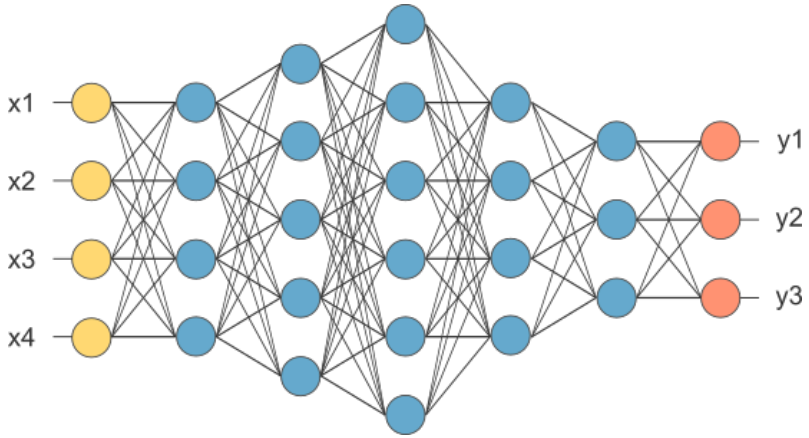
- $P(O \mid =, C, C)$

Natural language processing and SMILES



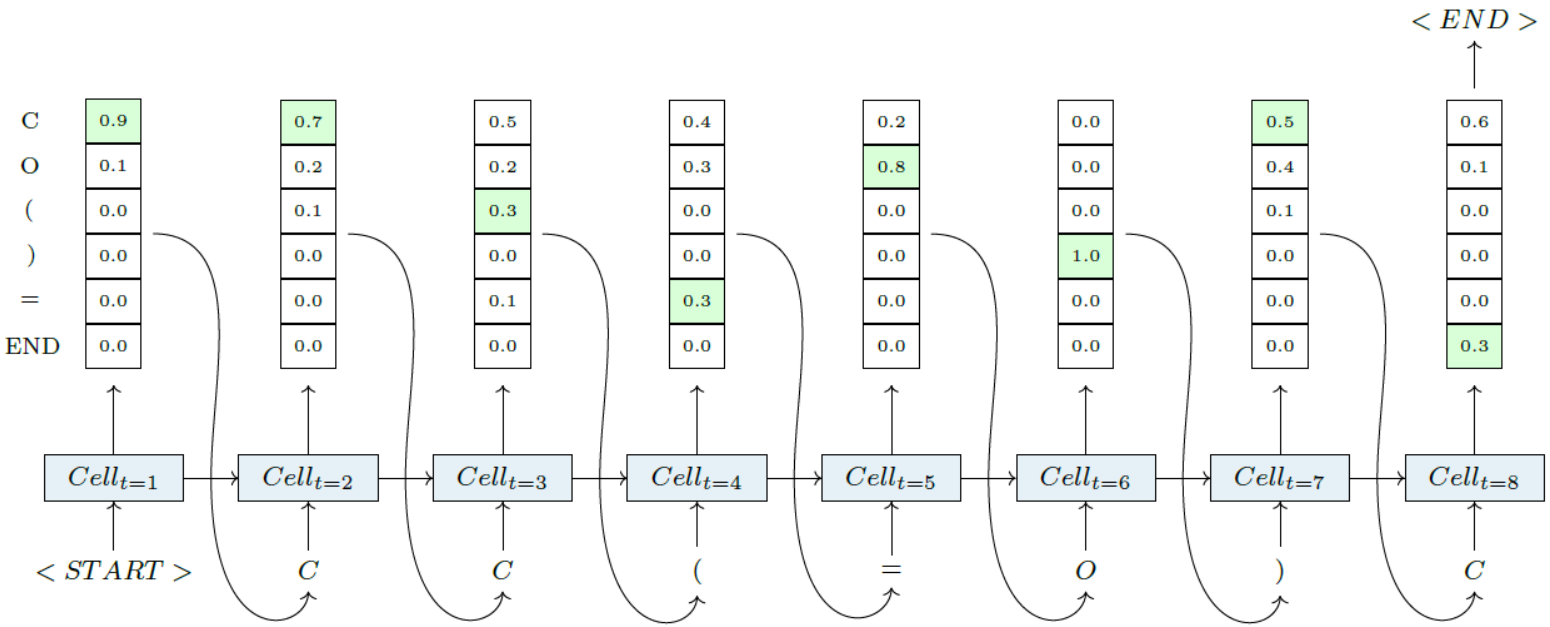
Neural networks

A neural network is a function approximator



Powerful due to flexibility

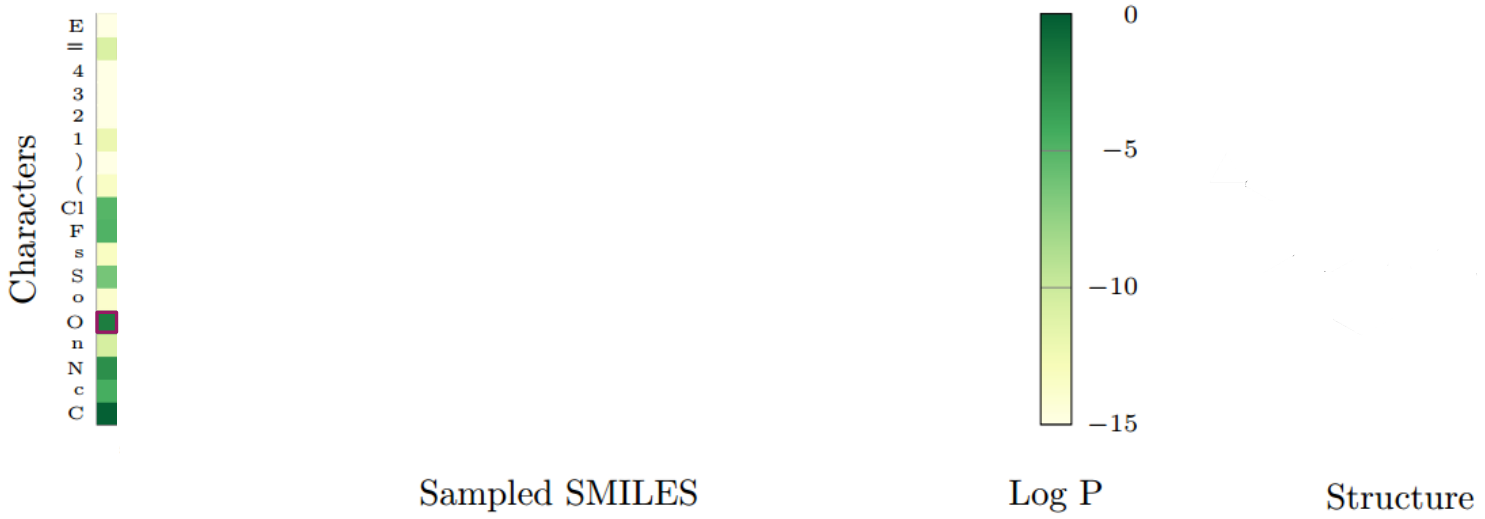
Recurrent Neural Networks



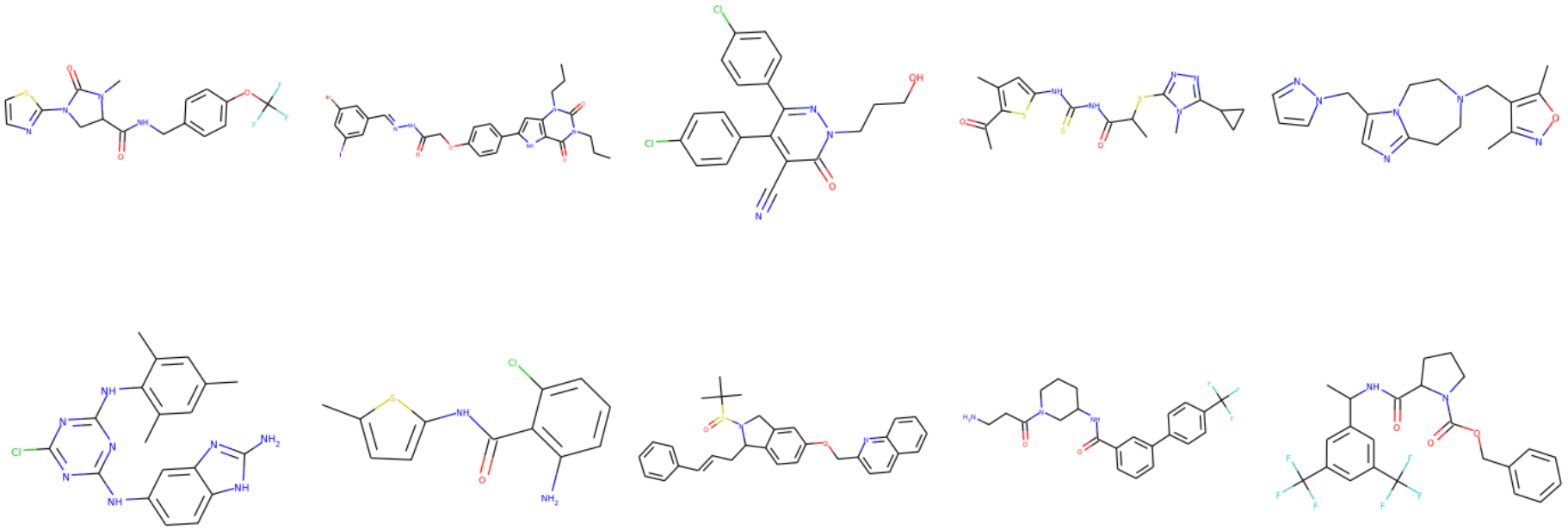
The prior RNN

- Restrain structures to 10 to 50 heavy atoms
- And elements H, B, C, N, O, F, P, S, Cl, Br, I
- 1.5 million SMILES from ChEMBL
- Canonicalized using RDKit

The generative process



Structures generated by the Prior



Randomly selected

Searching Chemical Space

We need two things:

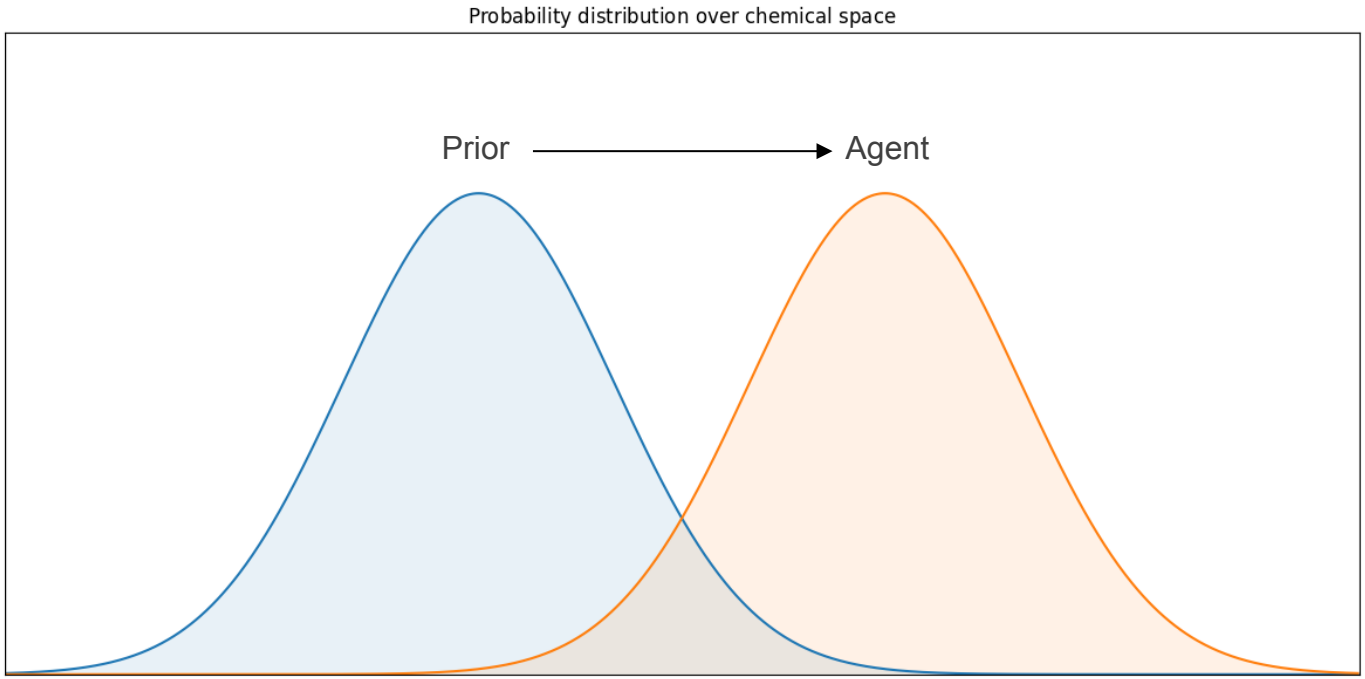
A way to generate structures

A way to rank structures – scoring function

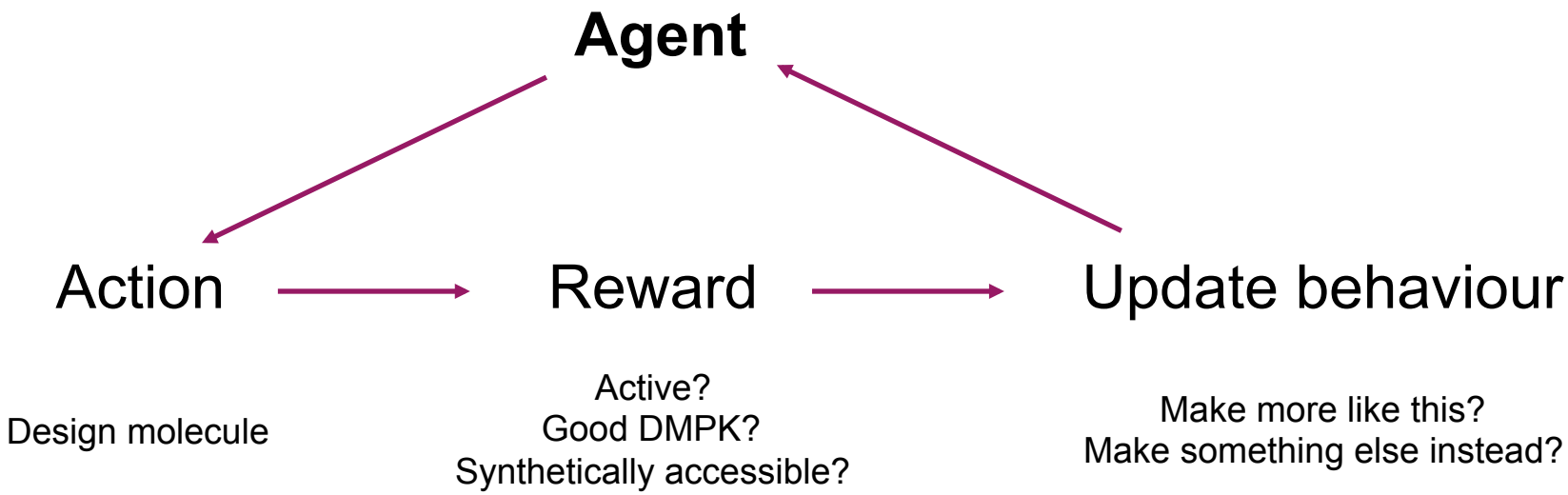
Ideally, the scoring function should influence the way we generate structures

Augmented Likelihood

$$Augmented\ Likelihood = Prior\ Likelihood + \sigma \times Score$$

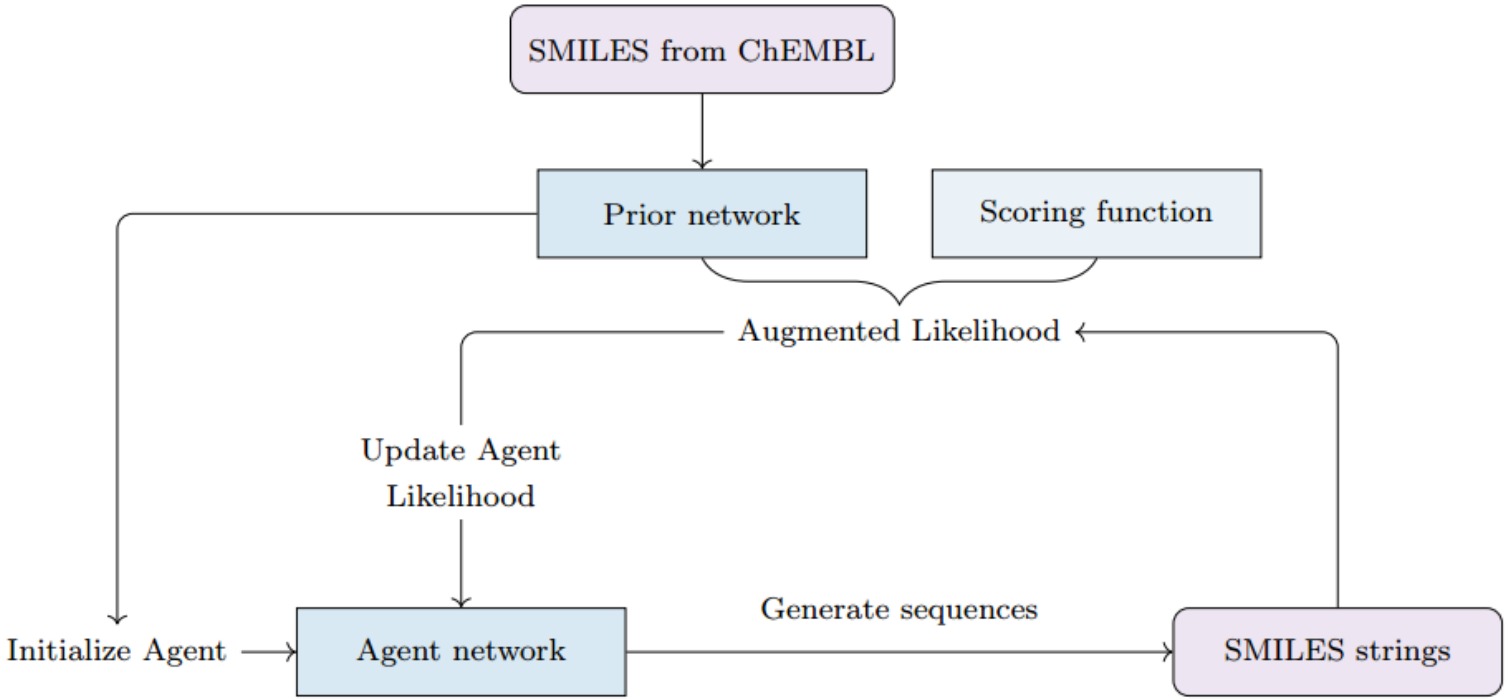


Reinforcement learning



- Learning from doing

The Agent


SMILES from ChEMBL
Prior network
Scoring function
Augmented Likelihood
Update Agent
Likelihood
Initialize Agent
Agent network
Generate sequences
SMILES strings

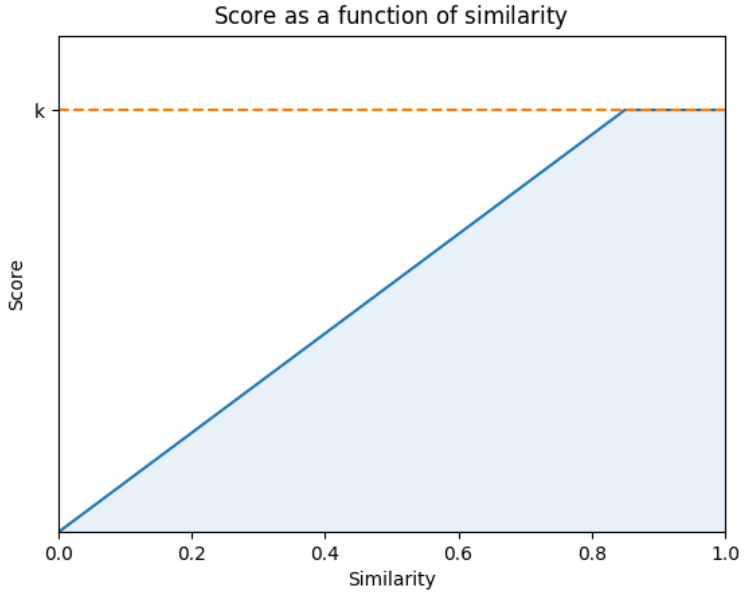
Results

Example 1 – Structure similarity

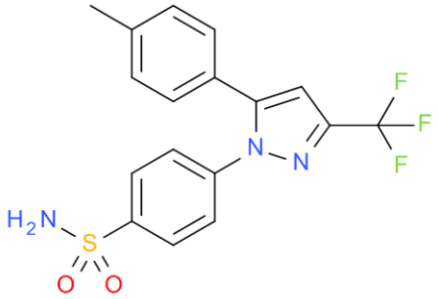
- Learn to generate analogues

$$S(A) = -1 + 2 \times \frac{\min\{J_{i,j}, k\}}{k}$$

- *J* is the Tanimoto similarity
- FCFP4 fingerprints
- *k* is the upper limit of *J*

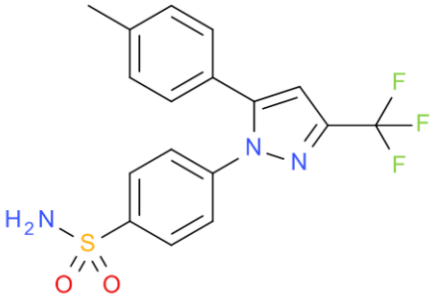


Generating only Celecoxib

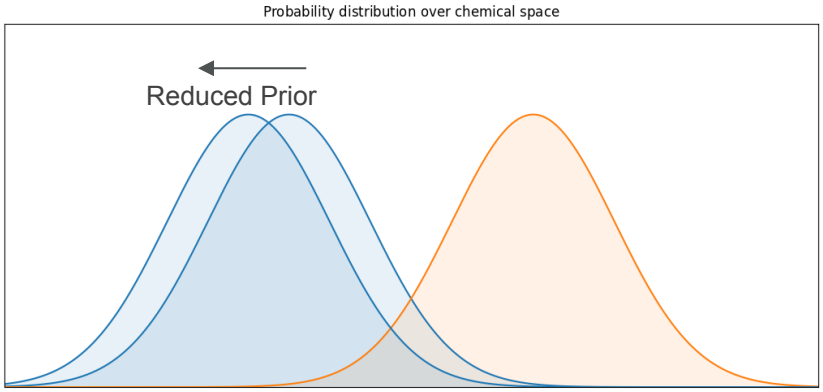


- We chose Celecoxib as a query structure
- First explored if we could recover Celecoxib itself
- $k = 1$ and $\sigma = 15$
- After 200 training steps, the model generates only Celecoxib
- Even when everything with $J > 0.5$ was removed from the Prior ("Reduced Prior"), Celecoxib was recovered

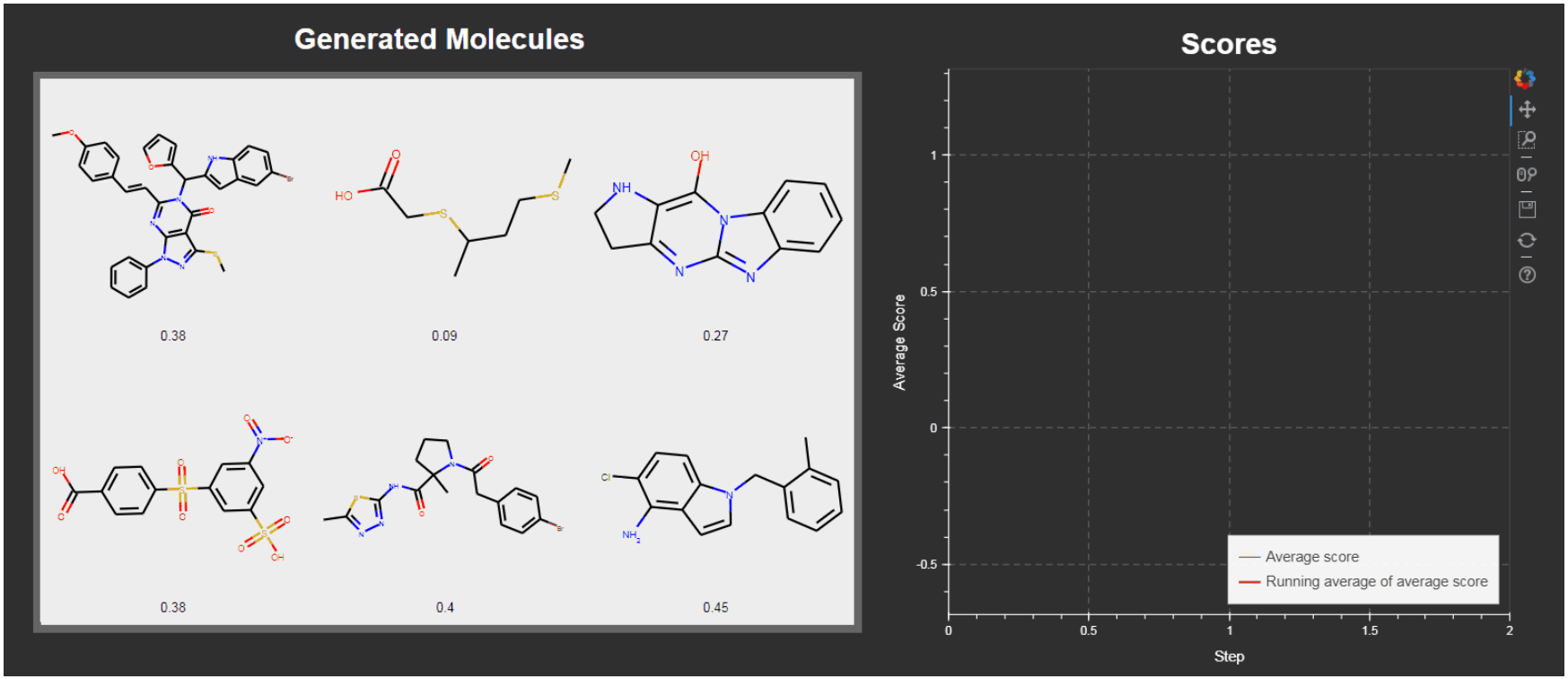
Generating analogues to Celecoxib

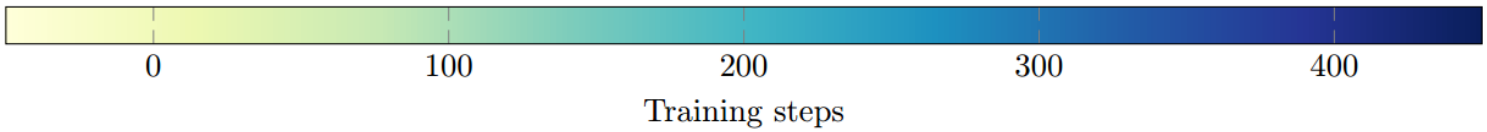


- But we want analogues
- $k=0.7$ and $\sigma=12$
- For canonical Prior learns well
- Reduced Prior requires a higher $\sigma=15$ to offset the lower prior likelihood



Example 1 – Structure similarity




0
100
200
300
400
Training steps

Example 2 – Biological target activity

- Using activity model (SVM model) as the scoring function
- Uncalibrated probability of being active $P\downarrow active$

$$Score = -1 + 2 \times P\downarrow active$$

- Train for 3000 steps with $\sigma = 7$
- Also remove all actives from ChEMBL and train a reduced Prior
- Then train reduced Agent

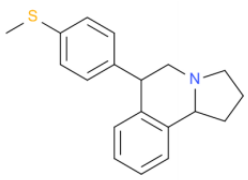
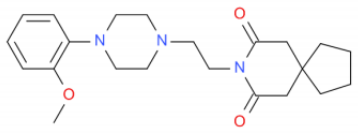
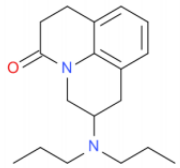
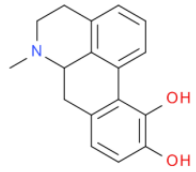
Enrichment of predicted actives

Model	Prior	Agent	Prior†	Agent†
Fraction predicted actives	0.03	0.97	0.02	0.96
Fraction similar to train active	0.02	0.79	0.02	0.75
Fraction similar to test active	0.01	0.46	0.01	0.38
Fraction of test actives recovered	0.00	0.13	0.00	0.07
Probability of generating a test set active ($\times 10^{-3}$)	0.17	40.2	0.05	15.0

†DRD2 actives witheld from the training of the Prior

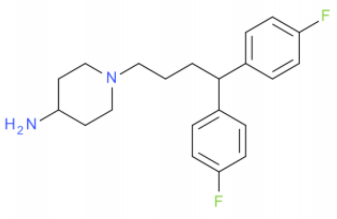
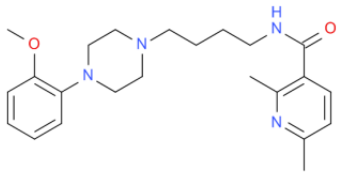
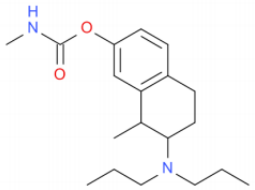
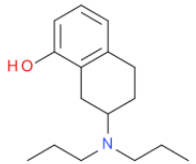
- Enrichment of 250 times for both Agents
- Withholding actives affects similarity to actives, but not fraction of predicted actives

Structures generated by reduced Agent

P_{active}	0.95	0.95	0.73	0.66

Recovered test actives

P_{active}	1.00	0.99	0.98	1.00

Randomly selected

Conclusion

Acknowledgements

- Thanks to Hongming, Thomas, and Ola

- Thanks to Thierry Kogej and Christian Tyrchan for valuable discussion

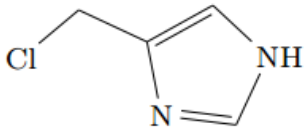
Confidentiality Notice

Backup slides

Tokenization of SMILES

- Tokenize combinations of characters like “Cl” or “[nH]”
- Represent the characters as one-hot vectors

Graph:



SMILES: ClCc1c[nH]cn1

One-hot encoding:

	Cl	C	c	1	c	nH	c	n	1
C	0	1	0	0	0	0	0	0	0
c	0	0	1	0	1	0	1	0	0
n	0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	1
nH	0	0	0	0	0	1	0	0	0
Cl	1	0	0	0	0	0	0	0	0

Augmented Likelihood

- Do not want to forget the prior probability distributions
- Instead modify it using a scoring function
- Scoring function rates desirability of a molecule (e.g. bioactivity)

$$Augmented\ Likelihood = Prior\ Likelihood + \sigma \times Score$$

- Likelihood is here reported as the natural logarithm
- Sigma is a constant signifying the tradeoff between score and prior likelihood

Augmented Likelihood

$$Augmented\ Likelihood = Prior\ Likelihood + constant \times Score$$

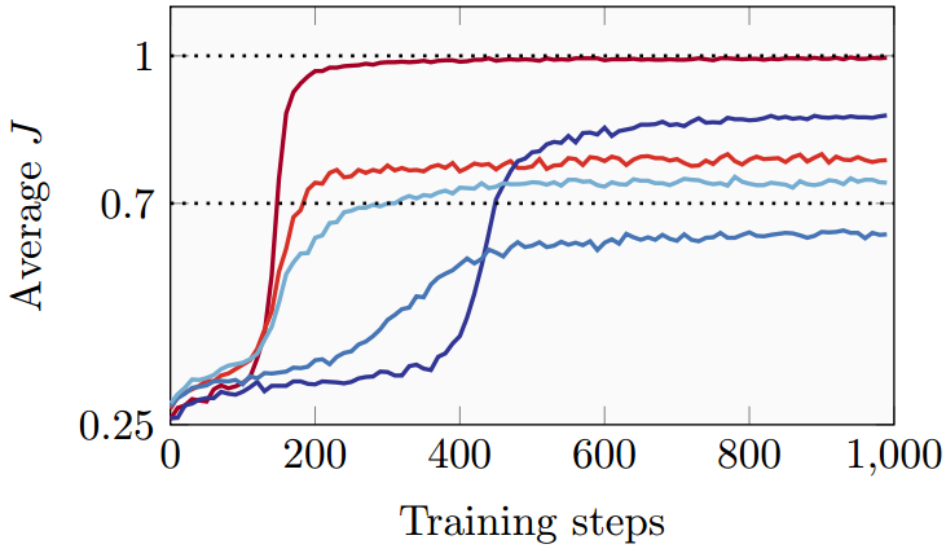
- For example:
  - $Molecule = Ethanol$
  - $Scoring\ function = \text{“}-1\ if\ aromatic\ else\ 1\text{”}$
  - $Score = 1$
  - $Constant\ \sigma = 3$
  - $Prior\ Likelihood = -15$

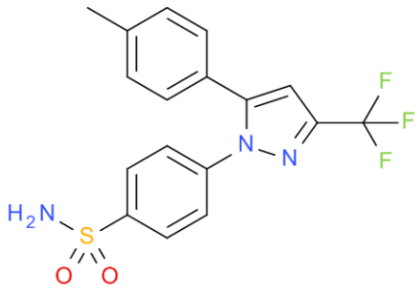
Prior P	constant	Score	Augmented P
-15	3	1	$-15 + 3 \times 1 = -12$

The prior RNN

- 94% of the generated sequences are valid SMILES
- 90% are novel – not part of the 1.5 million training SMILES
- Most common error is not closing an opened ring or branch

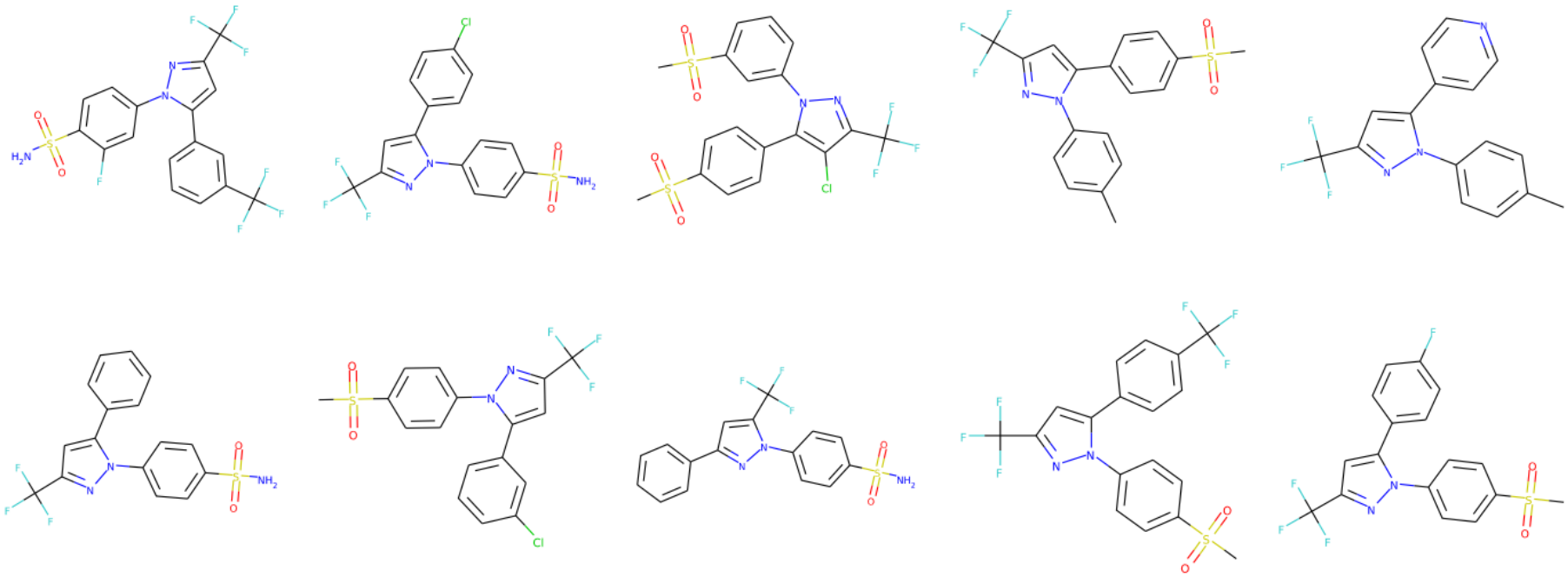
Example 1 – Structure similarity





Canonical prior	$k = 1$	$\sigma = 15$
Canonical prior	$k = 0.7$	$\sigma = 12$
Reduced prior	$k = 1$	$\sigma = 15$
Reduced prior	$k = 0.7$	$\sigma = 12$
Reduced prior	$k = 0.7$	$\sigma = 15$

Example 1 – Structure similarity



Example 2 – SVM model

- DRD2 activity data from ExCAPE-DB
- 7218 actives and 340 000 inactive compounds
- Actives were clustered using the Butina algorithm (ECFP6 cutoff of 0.4)
- Clusters were split into train (4/6), validation(1/6), and test(1/6) sets
- 4526, 1287, 1405 actives respectively

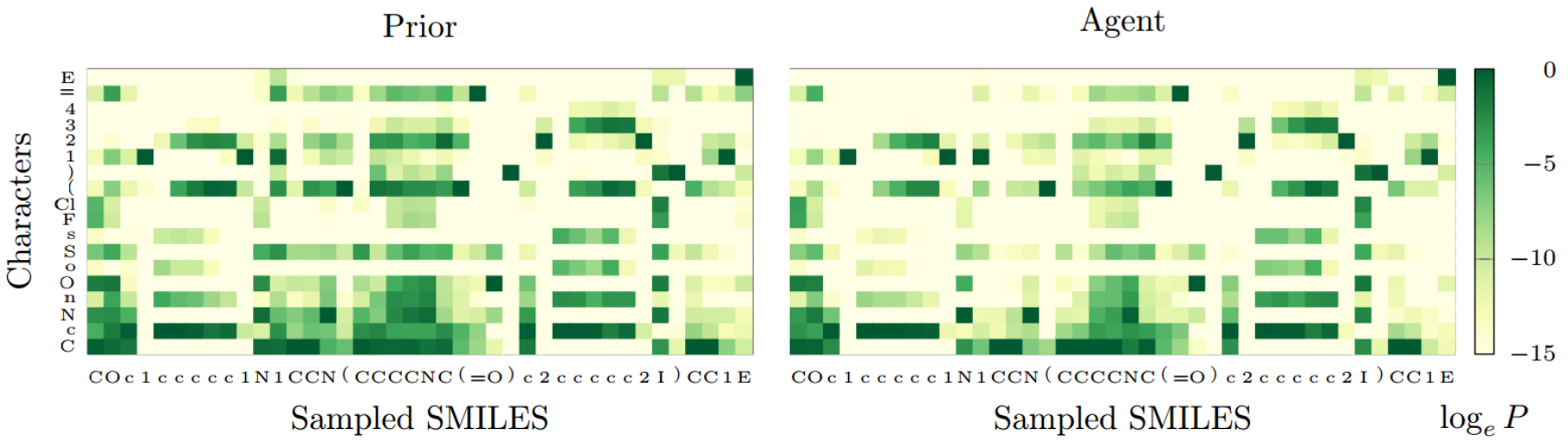
Example 2 – SVM model

- Used a SVM classifier in RDKit with $C=2^{7}$ and $gamma=2^{-6}$

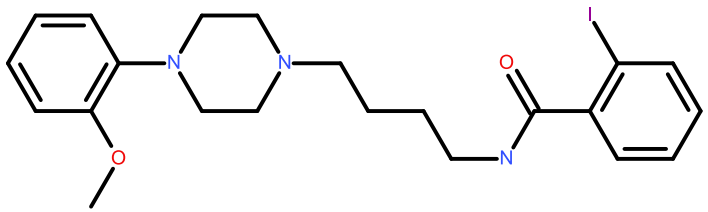
Table 2 Performance of the DRD2 activity model

Set	Training	Validation	Test
Accuracy	1.00	0.98	0.98
ROC-AUC	1.00	0.99	1.00
Precision	1.00	0.96	0.97
Recall	1.00	0.73	0.82

Example 2 – Probability distributions



- Reduced Prior and Agent
- Small changes overall
- But a large change at even one step may significantly change the structures generated



Example 3 – Learn to avoid sulphur

- Toy example – learn to generate structures that do not contain sulphur

$$S(A) = \begin{cases} & 1 \text{ if valid and no S} \\ & 0 \text{ if not valid} \\ - & 1 \text{ if contains S} \end{cases}$$

- Train for 1000 steps with σ=2 and a batch size of 128

Example 3 – Learn to avoid sulphur

Model	Prior	Agent
Fraction of valid SMILES	0.94	0.96
Fraction without sulphur	0.68	**0.98**
Average molecular weight	371 ± 1.70	367 ± 3.30
Average cLogP	3.36 ± 0.04	3.37 ± 0.09
Average NumRotBonds	5.39 ± 0.04	5.41 ± 0.07
Average NumAromRings	2.26 ± 0.02	2.26 ± 0.02

Example 3 – Learn to avoid sulphur