A Quantum Chemist Meets Cheminformatics

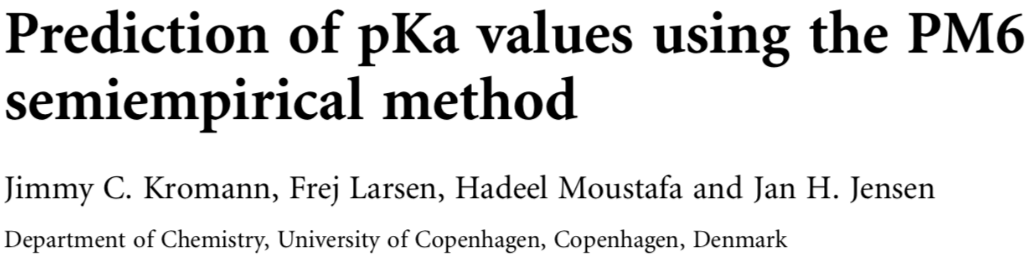
Jan H. Jensen, University of Copenhagen

@janhjensen

Feel free to tweet, record, …
#RDKitUGM2019

How I met RDKit

2016



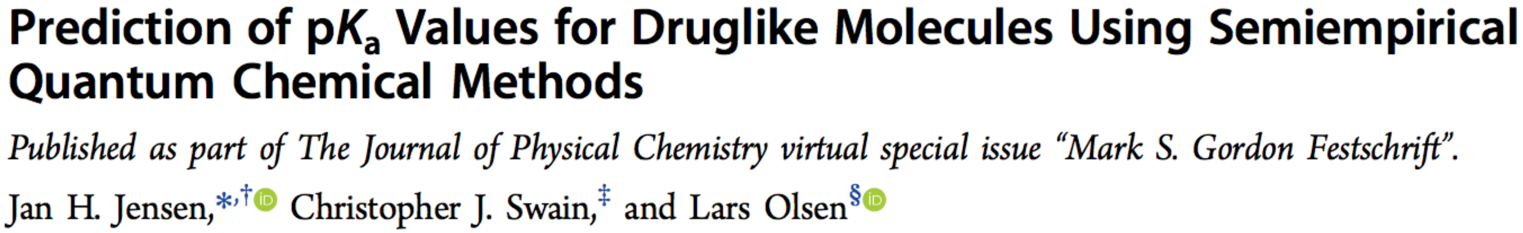
Prediction of pKa values using the PM6 semiempirical method

Jimmy C. Kromann, Frej Larsen, Hadeel Moustafa and Jan H. Jensen

Department of Chemistry, University of Copenhagen, Copenhagen, Denmark

RDKit

2017

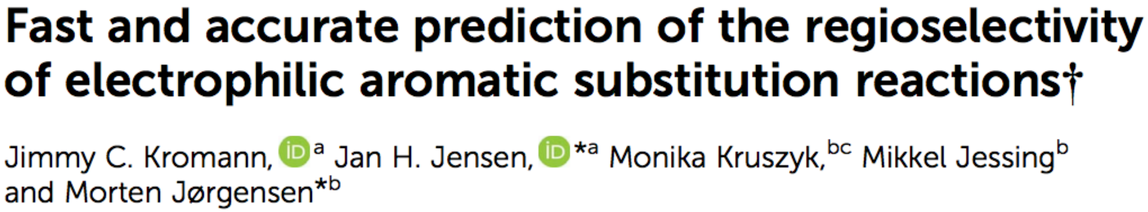


Prediction of pK_a Values for Druglike Molecules Using Semiempirical Quantum Chemical Methods

Published as part of The Journal of Physical Chemistry virtual special issue "Mark S. Gordon Festschrift".

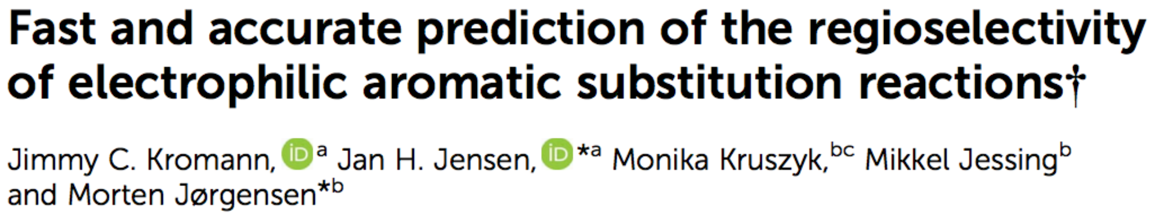
Jan H. Jensen,[*,†] Christopher J. Swain,[‡] and Lars Olsen[§]

2018



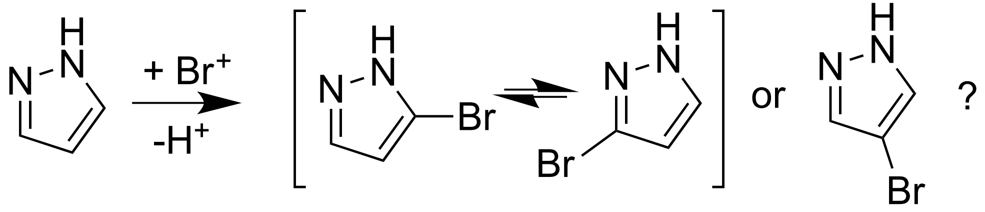
Fast and accurate prediction of the regioselectivity of electrophilic aromatic substitution reactions†

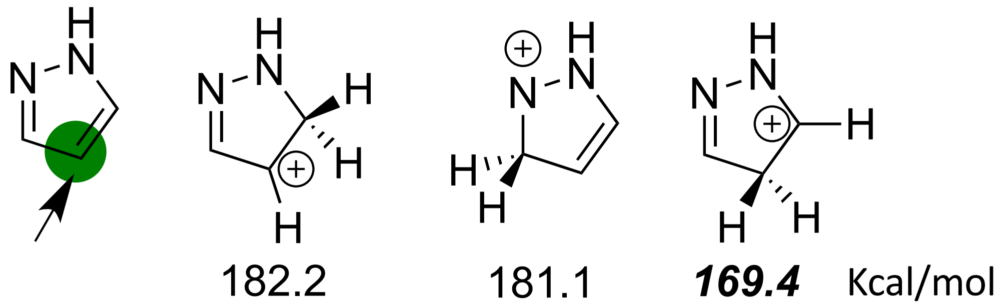
Jimmy C. Kromann,[a] Jan H. Jensen,[*a] Monika Kruszyk,[bc] Mikkel Jessing[b] and Morten Jørgensen[*b]



Fast and accurate prediction of the regioselectivity of electrophilic aromatic substitution reactions†

Jimmy C. Kromann, [a] Jan H. Jensen, *[a] Monika Kruszyk,[bc] Mikkel Jessing[b] and Morten Jørgensen*[b]





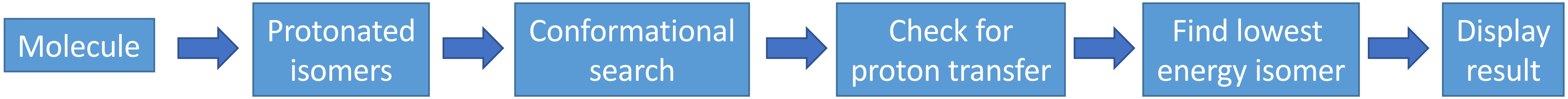
PM3/COSMO heat of formation

90% success rate for 520 compounds

Workflow / Automation

```
('[C;R;H1:1]=[C,N;R;H1:2]>>[CH2:1][*H+:2]')
('[C;R;H1:1]=[C,N;R;H0:2]>>[CH2:1][*+;H0:2]')
```

SMILES



Molecule

Protonated isomers

Conformational search

Check for proton transfer

Find lowest energy isomer

Display result

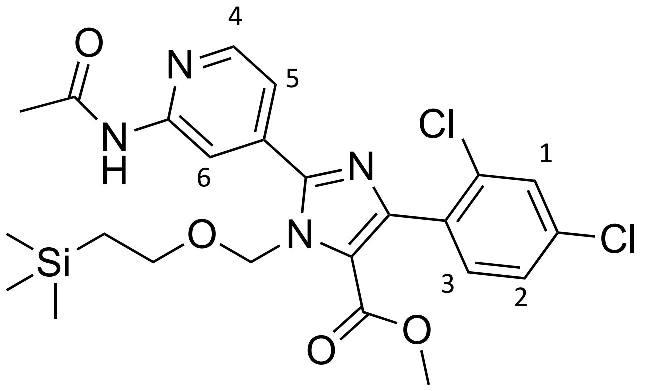
ChemDraw

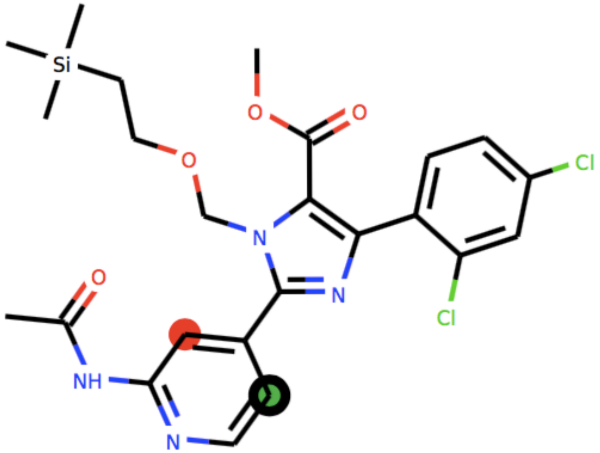
RDKit

RDKit

RDKit

6 isomers x 20 confs





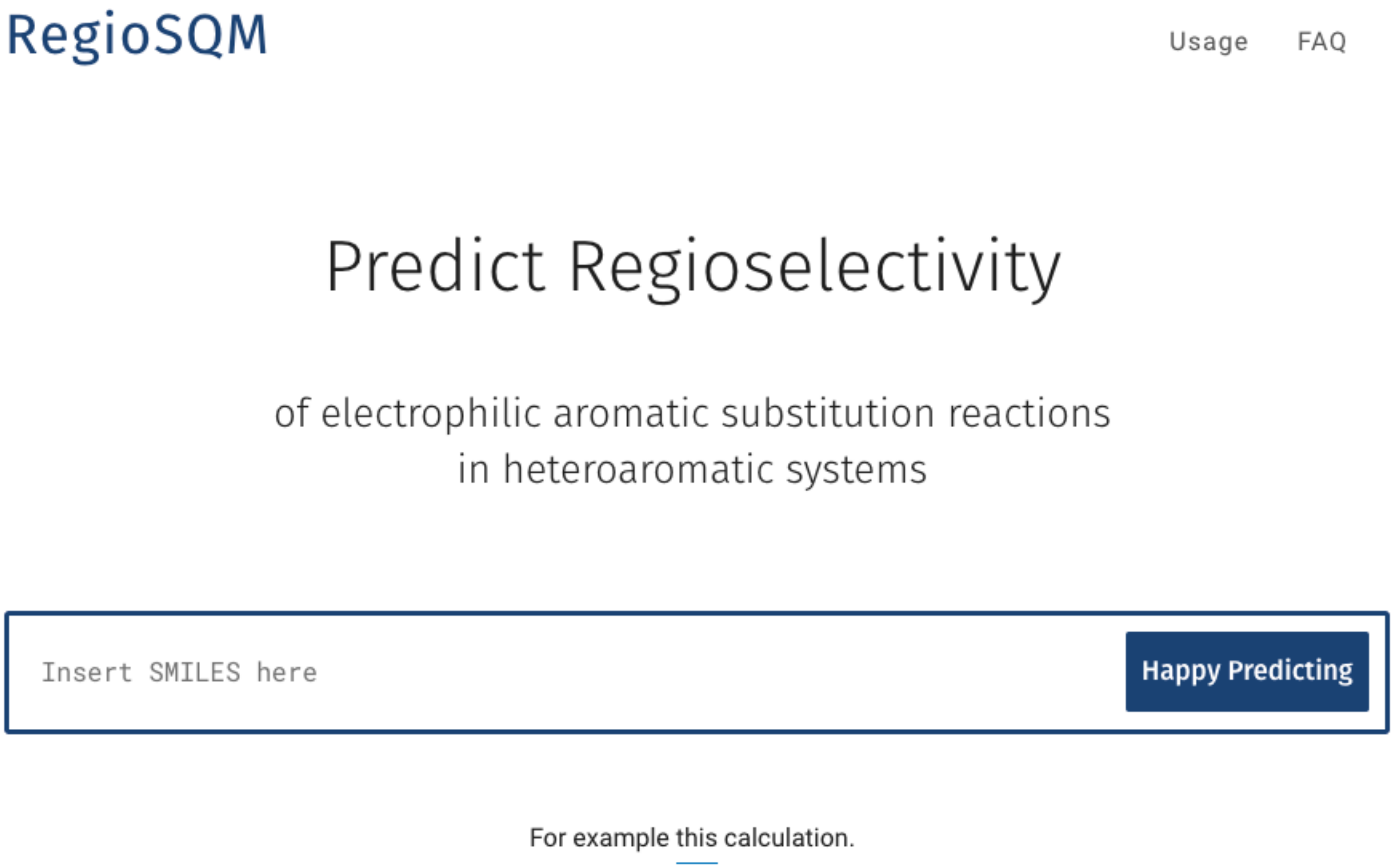
```
c1cnc(cc1c1n(c(c(n1)c1ccc(cc1Cl)Cl)C(=O)OC)COCC[Si](C)(C)C)NC(=O)C
```

SMILES

github.com/jensengroup/RegioSQM

Web server

regiosqm.org



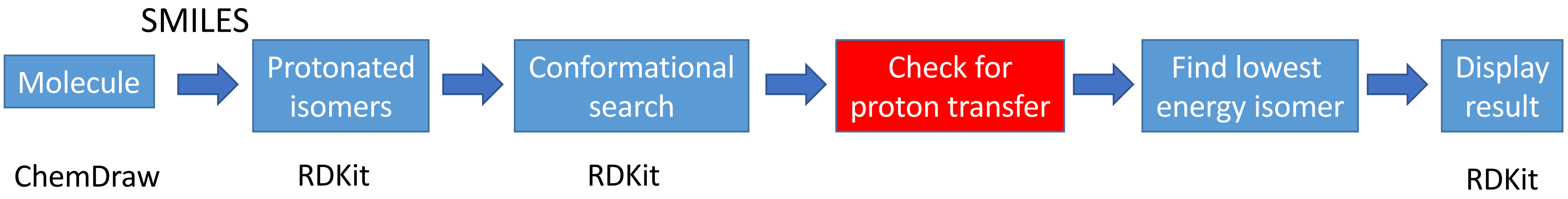
RegioSQM

Predict Regioselectivity

of electrophilic aromatic substitution reactions
in heteroaromatic systems

For example this calculation.

xyz2mol


SMILES

Molecule

Protonated isomers

Conformational search

Check for proton transfer

Find lowest energy isomer

Display result

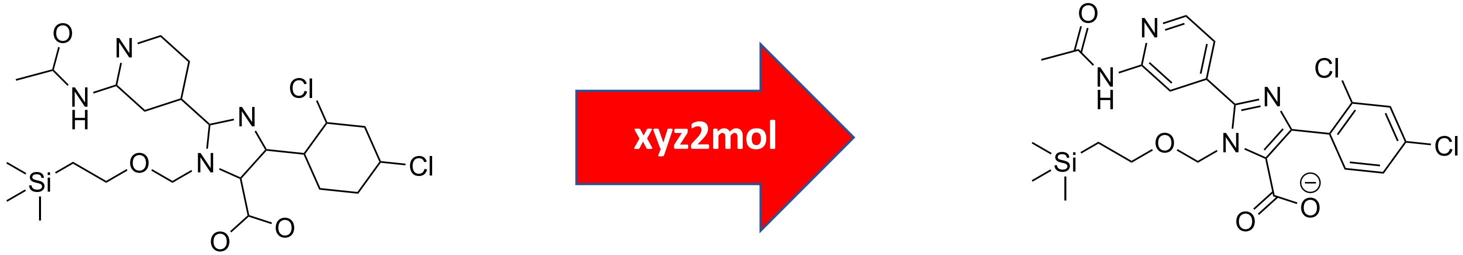
ChemDraw

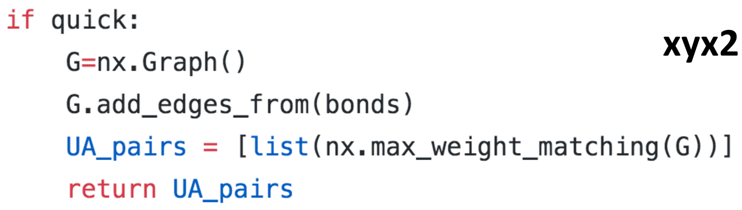
RDKit

RDKit

RDKit

6 isomers x 20 confs

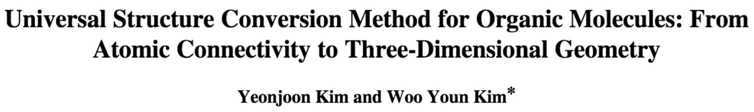




```
if quick:
    G=nx.Graph()
    G.add_edges_from(bonds)
    UA_pairs = [list(nx.max_weight_matching(G))]
    return UA_pairs
```

xyx2mol converts an xyz file to an RDKit mol object
(needs the molecular charge)

github.com/jensengroup/xyz2mol

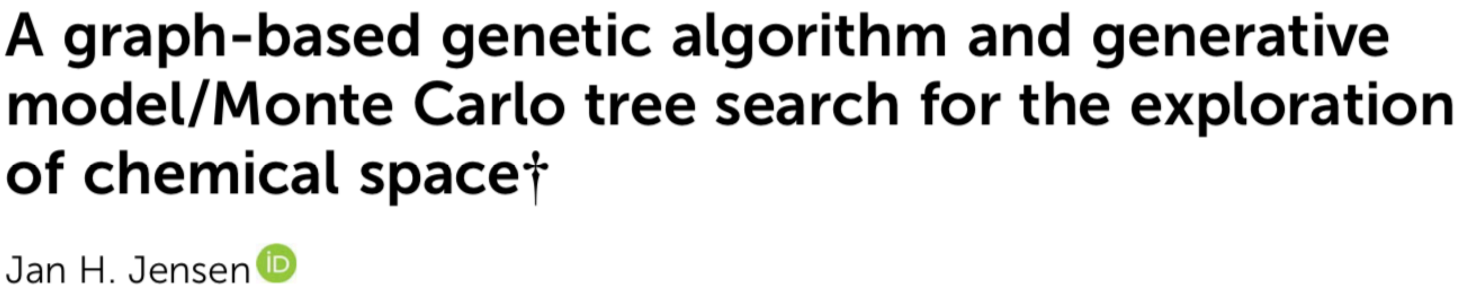


Universal Structure Conversion Method for Organic Molecules: From Atomic Connectivity to Three-Dimensional Geometry

Yeonjoon Kim and Woo Youn Kim*

Last example

2019

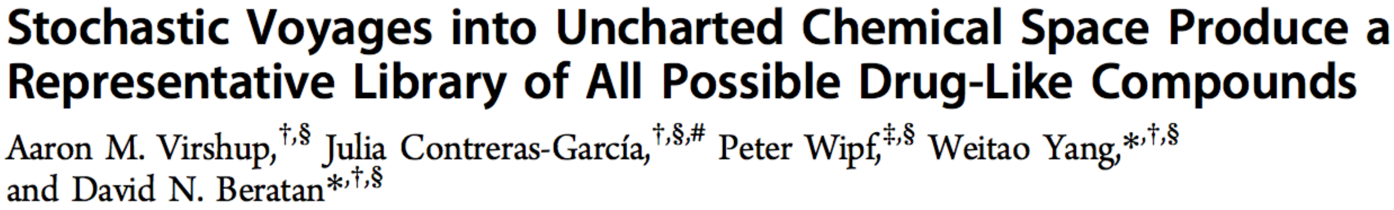


A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space[†]

Jan H. Jensen

An RDKit implementation of

2013

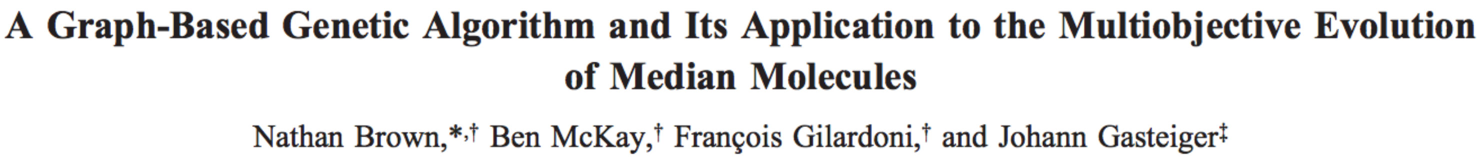


Stochastic Voyages into Uncharted Chemical Space Produce a Representative Library of All Possible Drug-Like Compounds

Aaron M. Virshup,[†,§] Julia Contreras-García,[†,§,#] Peter Wipf,[‡,§] Weitao Yang,[*,†,§] and David N. Beratan[*,†,§]

and

2004



A Graph-Based Genetic Algorithm and Its Application to the Multiobjective Evolution of Median Molecules

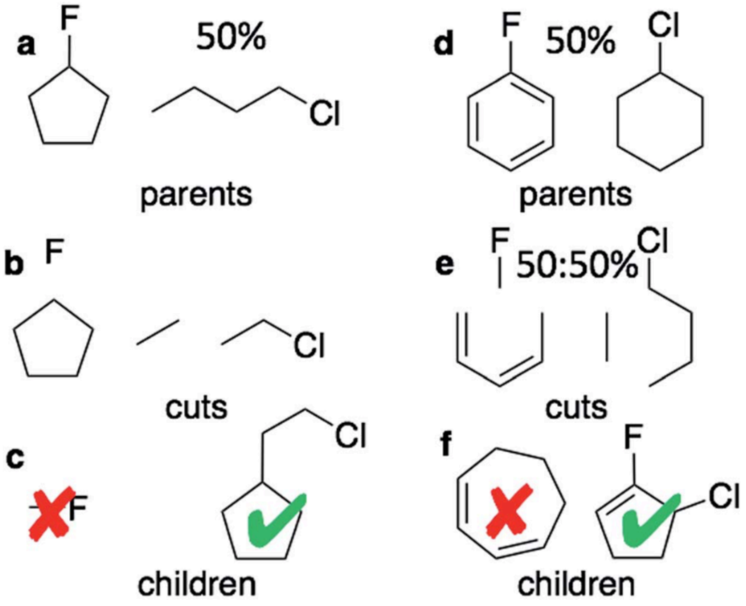
Nathan Brown,[*,†] Ben McKay,[†] François Gilardoni,[†] and Johann Gasteiger[‡]

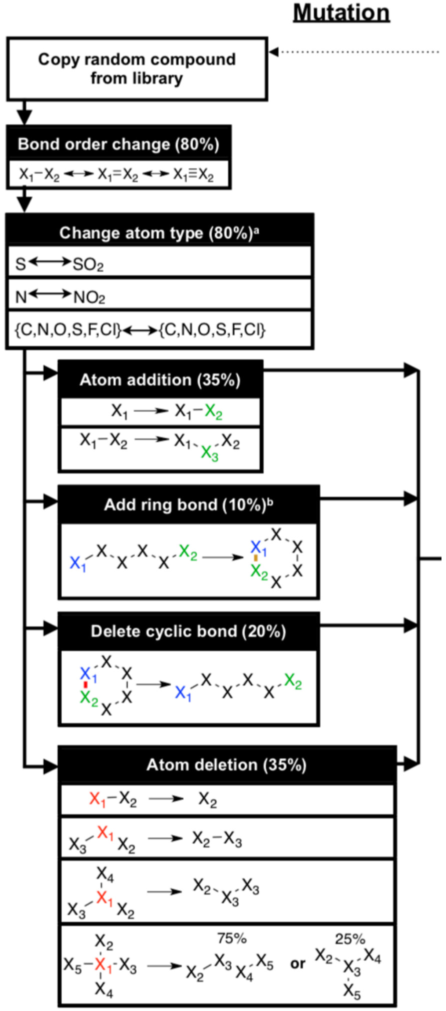
github.com/jensengroup/GB-GA

A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space†

Jan H. Jensen

crossover





Mutation

Copy random compound from library

Bond order change (80%)

$X_1{-}X_2 \leftrightarrow X_1{=}X_2 \leftrightarrow X_1{\equiv}X_2$

Change atom type (80%)[a]

$S \leftrightarrow SO_2$

$N \leftrightarrow NO_2$

{C,N,O,S,F,Cl} ↔ {C,N,O,S,F,Cl}

Atom addition (35%)

$X_1 \rightarrow X_1{-}X_2$

$X_1{-}X_2 \rightarrow X_1{-}X_3{-}X_2$

Add ring bond (10%)[b]

Delete cyclic bond (20%)

Atom deletion (35%)

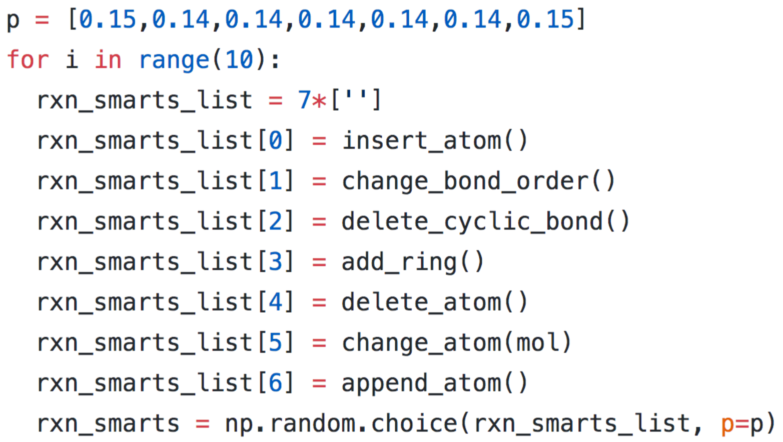
$X_1{-}X_2 \rightarrow X_2$

$X_3{-}X_1{-}X_2 \rightarrow X_2{-}X_3$

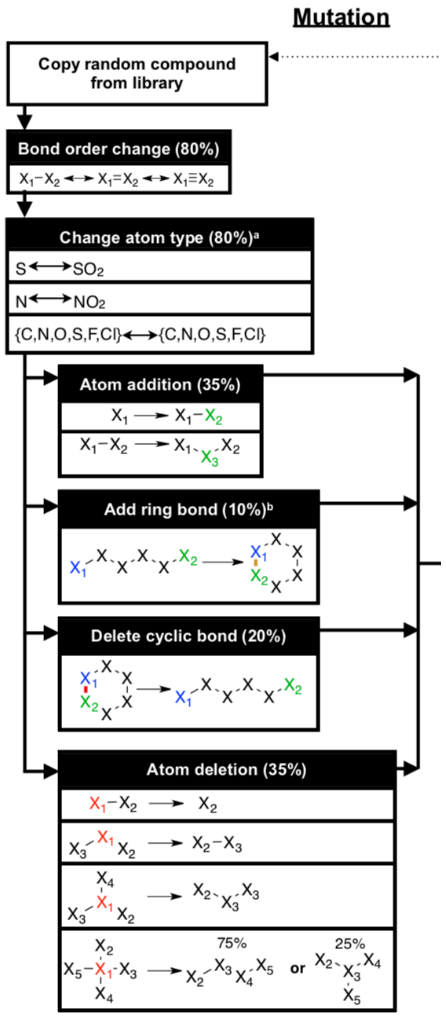
75% or 25%

A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space†

Jan H. Jensen

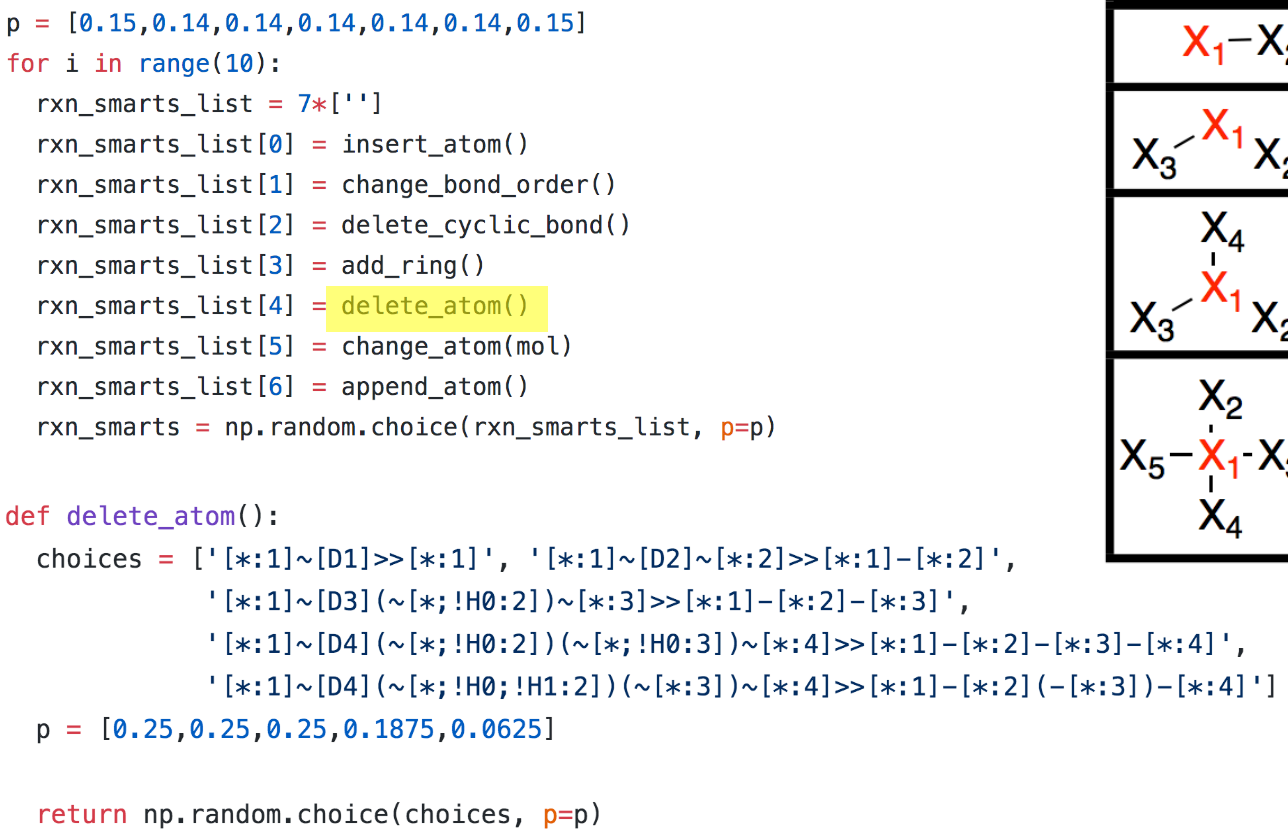


```
p = [0.15,0.14,0.14,0.14,0.14,0.14,0.15]
for i in range(10):
  rxn_smarts_list = 7*['']
  rxn_smarts_list[0] = insert_atom()
  rxn_smarts_list[1] = change_bond_order()
  rxn_smarts_list[2] = delete_cyclic_bond()
  rxn_smarts_list[3] = add_ring()
  rxn_smarts_list[4] = delete_atom()
  rxn_smarts_list[5] = change_atom(mol)
  rxn_smarts_list[6] = append_atom()
  rxn_smarts = np.random.choice(rxn_smarts_list, p=p)
```

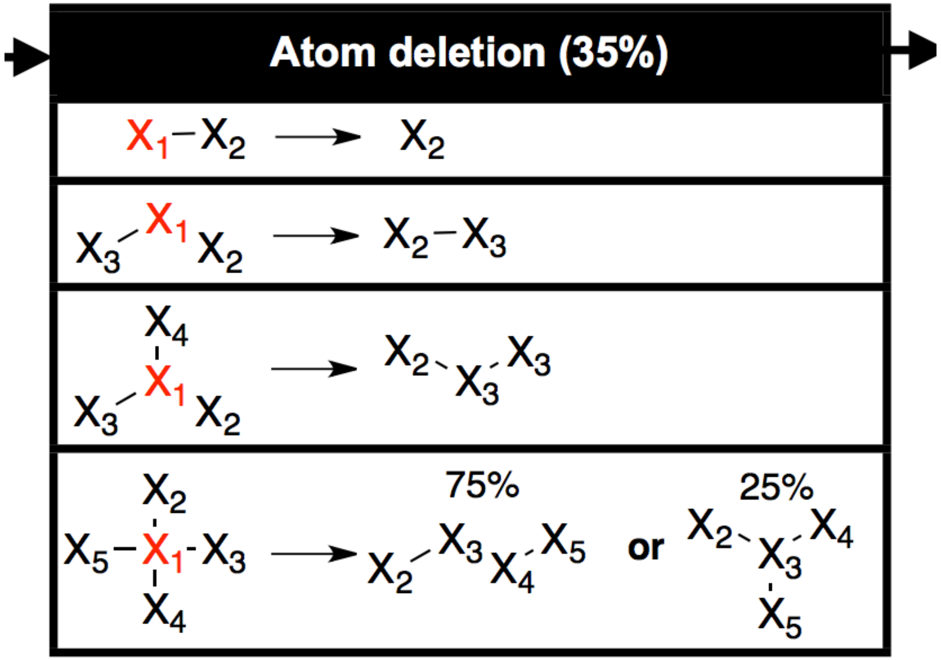


A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space†

Jan H. Jensen

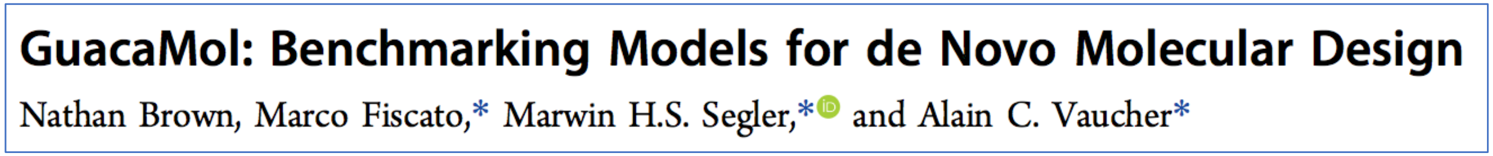


```
p = [0.15,0.14,0.14,0.14,0.14,0.14,0.15]
for i in range(10):
  rxn_smarts_list = 7*['']
  rxn_smarts_list[0] = insert_atom()
  rxn_smarts_list[1] = change_bond_order()
  rxn_smarts_list[2] = delete_cyclic_bond()
  rxn_smarts_list[3] = add_ring()
  rxn_smarts_list[4] = delete_atom()
  rxn_smarts_list[5] = change_atom(mol)
  rxn_smarts_list[6] = append_atom()
  rxn_smarts = np.random.choice(rxn_smarts_list, p=p)

def delete_atom():
  choices = ['[*:1]~[D1]>>[*:1]', '[*:1]~[D2]~[*:2]>>[*:1]-[*:2]',
             '[*:1]~[D3](~[*;!H0:2])~[*:3]>>[*:1]-[*:2]-[*:3]',
             '[*:1]~[D4](~[*;!H0:2])(~[*;!H0:3])~[*:4]>>[*:1]-[*:2]-[*:3]-[*:4]',
             '[*:1]~[D4](~[*;!H0;!H1:2])(~[*:3])~[*:4]>>[*:1]-[*:2](-[*:3])-[*:4]']
  p = [0.25,0.25,0.25,0.1875,0.0625]

  return np.random.choice(choices, p=p)
```



Atom deletion (35%)

Does it work?



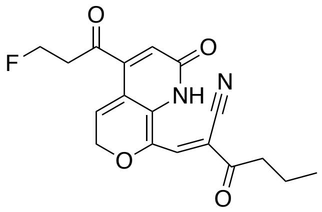
GuacaMol: Benchmarking Models for de Novo Molecular Design

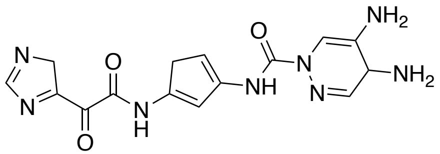
Nathan Brown, Marco Fiscato,* Marwin H.S. Segler,* and Alain C. Vaucher*

Unpublished: finding molecules that absorb at 600 nm

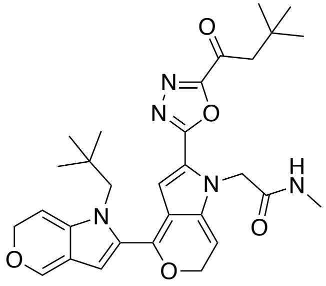
Mating pool size: 20, mutation rate: 0.05, sTDA-xTB, 10 runs

Starting from random molecule in the ZINC data base





	Max gen.	Mean gen.
600 nm		
GB	31	19.9±6.8
SMILES	50(1)	18.2±12.1
DeepSMILES	50(1)	18.5±12.0
SELFIES	50(2)	32.3±14.1



Summary/Outlook

RDKit changed my research life

Quantum chemical studies need to be automated

Manual work replacing CPU power as rate limiting step

Mistakes become increasingly common

QM students need to learn Python/RDKit

“QM-needs” for RDKit

Conf search for finding global minimum

Generalized Born solvation model