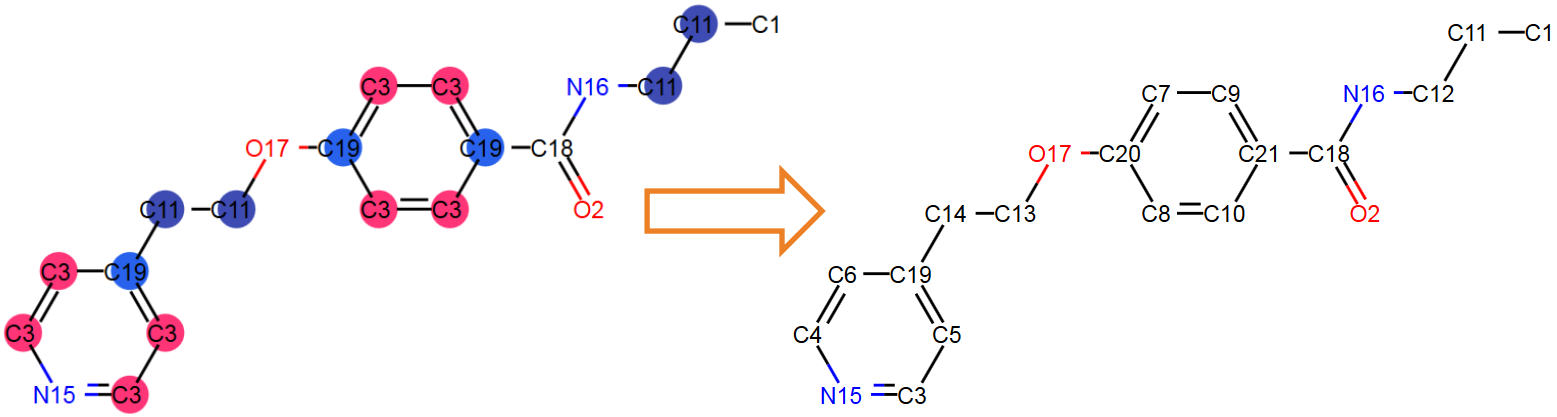


Get your atoms in order

Reducing graph canonicalization to a sorting problem

Nadine Schneider

Postdoctoral fellow NIBR Informatics

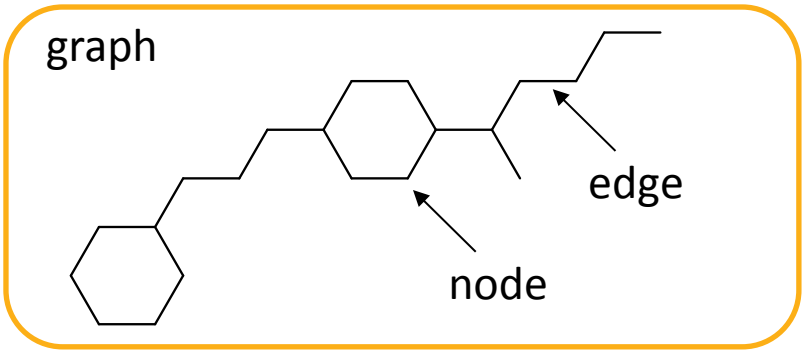
4th RDKit UGM

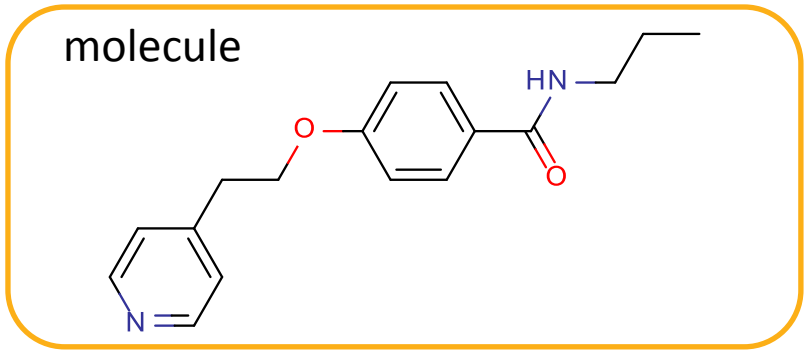
3rd September 2015

NOVARTIS

What does graph canonicalization mean?

From graph theory to molecules





- Generating a unique order of the nodes of a graph
- For molecules: Finding a canonical order of the atoms (based on topology)
- Pre-requisite for generating a unique representation of the molecule
- Can be used to generate a unique SMILES
  - → search compound databases or the web
  - → identify duplicates in databases

Canonicalization approaches date back to 1965

So why do we need yet another one?

- First canonicalization approaches date back to 1965, like the well-known Morgan algorithm[1] and many refinements of that[2-8]
- Morgan introduced the **extended connectivity**: equivalence classes of the immediate neighbors of an atom are employed to update its equivalence class
- Problem: non-equivalent atoms can be assigned the same extended connectivity values[4]
- Solutions, e.g., the products of primes[2], may lead to numerical problems when molecules become large (integer overflow)
- Many of the limitations arise from the FORTRAN implementation of these early versions which only allowed the sorting of numbers

[1] Morgan HL. J Chem Doc 1965, 5:107−113
[2] Weininger et al. *J Chem Inf Comput Sci* 1989, **29**:97−101
[3] Wipke WT, Dyott TM. J Chem Soc 1974, 96:4834−4842
[4] Randić M. J Chem Inf Comput Sci 1975, 15: 105-108.
[5] Jochum C, Johann G. J Chem Inf Comput Sci 1977, 17: 113-117.
[6] McKay BD. Congressus Numerantium 1981, 30:45-87.
[7] Randic M et al. J Chem Inf Comput Sci 1982, 21: 52-59.
[8] Faulon JL. *J Chem Inf Comput Sci* 1998, **38**: 432-444.

NOVARTIS

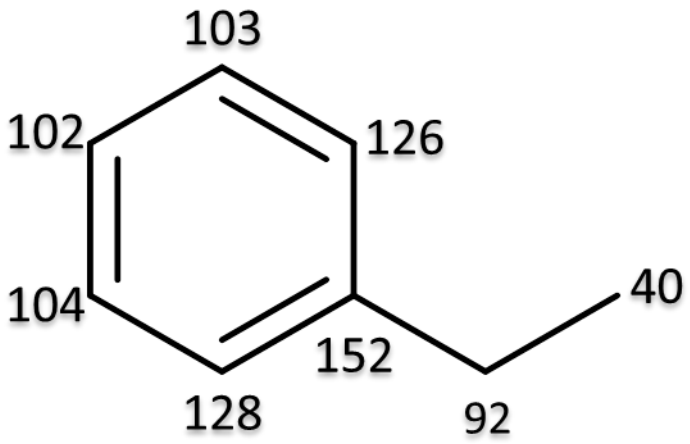
Graph canonicalization algorithm

Three steps to reach the goal

- All canonicalization algorithm have basically three steps:

1. **Initialization**: initial assignment of node equivalence classes (EC) based on topological features of the node (for atoms also chemical features)
2. **Relaxation**: refinement of the initial equivalence classes by incorporating the equivalence classes of the immediate neighbors
3. **Tie-breaking/Symmetry class assignment**: when no further differentiation is possible a tie-breaking is performed or for very elaborate versions a systematic test for symmetry classes is done

Example Morgan:
Final result after relaxation and two tie-breaking steps



Extended connectivity:
$EC\ atom_i = EC\ neighbor_{i-1} + EC\ neighbor_{i+1}$

NOVARTIS

Yet another canonicalization approach

...or not?

- Uses a **stable index**[1] for equivalence classes and implements an **efficient partition-based graph relaxation**[2] method
- **Two special invariants** were developed which allow construction of a canonical atom order for highly symmetrical molecules
- The new approach has proven to be robust and fast on a large, diverse set of molecules

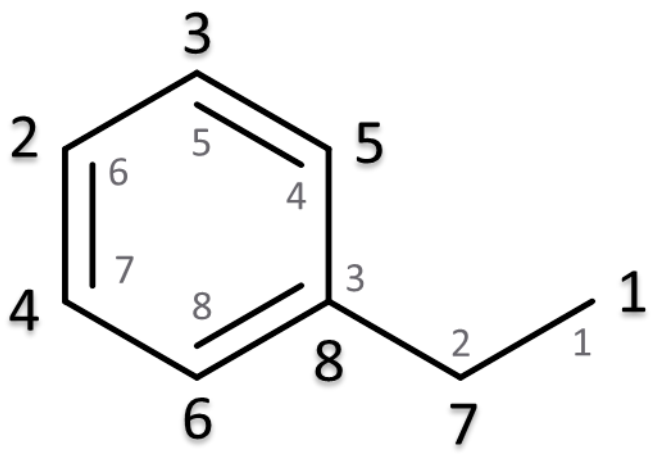
[1] Rohde B: GM-Search. A System for Stereochemical Substructure Search. PhD Thesis. Philosophical Faculty II, University of Zürich; 1988.
[2] Paige R, Tarjan RE: Three partition refinement algorithms, SIAM J Comput, 1987, 16(6):973-989.

Canonicalization using a stable index and sorting

Reducing the number of re-examinations of atoms

- The new algorithm has also basically three steps:

1. **Initialization**: atom invariant is based on common atom properties: degree, atomic number, isotope, total number of hydrogen atoms attached, charge, atom stereochemistry, and bonds (bond type, bond stereo)
2. **Relaxation**: using a **stable index and a standard stable-sorting algorithm**, if ties exist apply **two new invariants**
3. **Tie-breaking**



Atom	1	2	3	4	5	6	7	8	Index after:
Index	1	1	1	1	1	1	1	1	Initialization
1	1	7	8	2	2	2	2	2	Initial atom invariant
2	1	7	8	5	3	2	3	5	Including neighbors
3	1	7	8	5	3	2	4	6	Tie breaking

Stable index: $i_r = i_{r-1} + n_{r-1}$

i_r = index of atoms in class r,
n_r = number of atoms in class r

NOVARTIS

Partition-based graph relaxation

An alternative to avoid re-sorting of all atoms in each iteration

- Allows re-evaluation of only those partitions which contain atoms whose neighbors' indices have been updated in the previous pass and the number of elements to sort is reduced

  → FASTER

- Any stable sorting algorithm can be used to determine the order of the atoms within a partition

  → "Easy" to re-implement

- In the RDKit we implemented a customized sort algorithm which is based on the mergesort "Towers of Hanoi" algorithm[1,2]

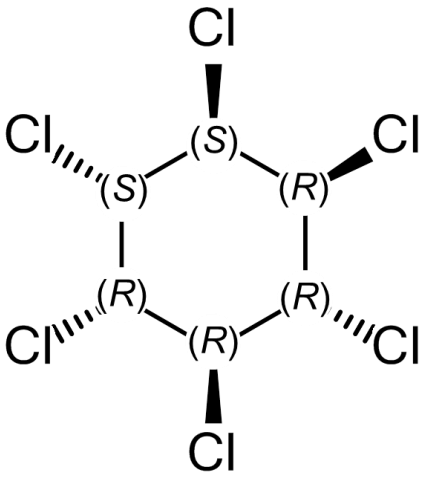
  → Faster in sorting partially ordered lists containing duplicated keys

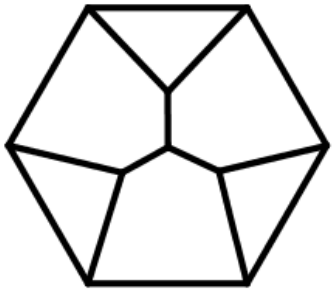
[1] Towers of Hanoi Mergesort: https://sourceware.org/ml/libc-alpha/2002-01/msg00218.html
[2] Knuth DE: The Art of Computer Programming: Sorting and Searching, Addison-Wesley Publishers, 1998.

Two new invariants

Allows canonicalization of highly symmetrical molecules

- Tie breaking of non-equivalent atoms results in a non-canonical labeling
- Highly symmetrical molecules, especially those exhibiting dependent chirality, are not distinguishable with the standard atom invariant
- Non-local properties play a role for these molecules
- Alternatively all possible numberings of the “equivalent” atoms have to be tested for equality to guarantee a canonical labeling

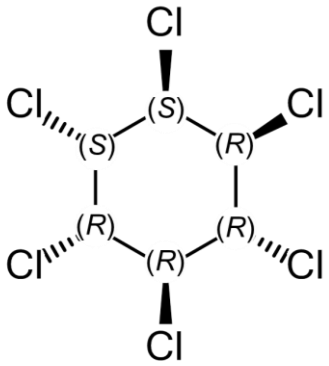


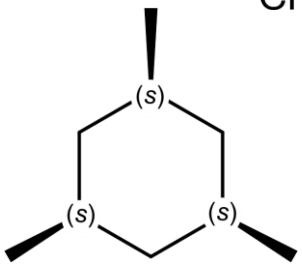


Two new invariants

Handling of molecules with dependent chirality

- Explicit CIP chirality assignment of the molecules is not necessary
- Dependent chirality is only determined by a different constitution of the neighbors
- Chiral centers need to be marked as such and the constitution (clockwise or anticlockwise) must be given





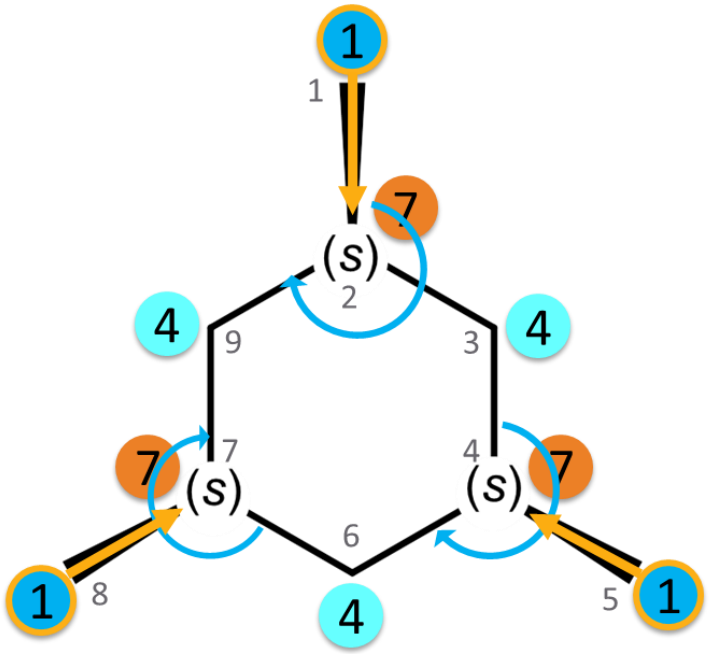
Algorithm:

- Compare original constitution of the chiral atom to the current constitution defined by the current index of the neighbors
- Calculate the number of swaps required to interconvert from the original constitution to the current constitution
- Number of swaps is even → retain constitution otherwise invert

NOVARTIS

Two new invariants

Handling of molecules with dependent chirality



Atom 1: Origin constitution: 1 3 9 (@)
Current constitution: 1 3 9
Swaps: 0
Final constitution: @

Atom 5: Origin constitution: 3 5 6 (@)
Current constitution: 5 3 6
Swaps: 1
Final constitution: @@

Atom 8: Origin constitution: 6 8 9 (@)
Current constitution: 8 6 9
Swaps: 1
Final constitution: @@

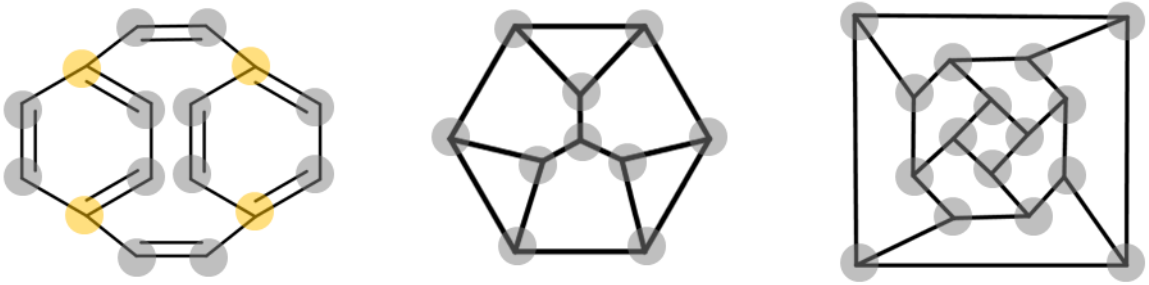
Atom	1	2	3	4	5	6	7	8	9	Index after:
Index	1	7	4	7	1	4	7	1	4	Refinement
Tie is broken	1	7	4	7	2	4	7	2	4	1. iteration special invariant
	1	7	4	8	2	6	9	3	5	Refinement with special invariant

NOVARTIS

Two new invariants

Handling of highly symmetrical molecules

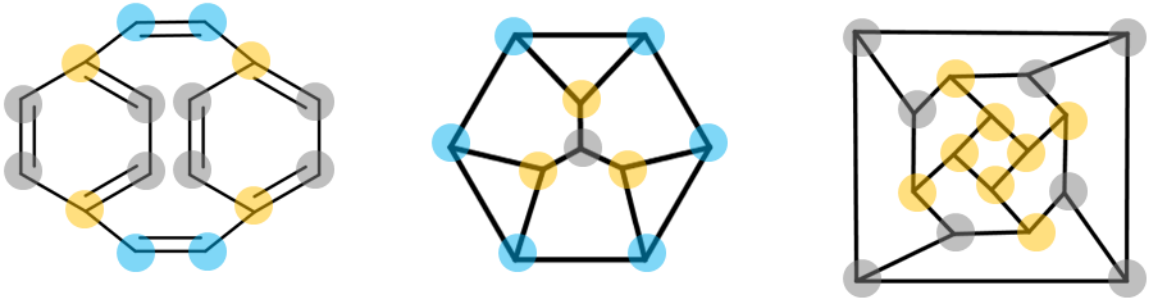
Before



Different colors = different equivalence classes

- To calculate this invariant a breadth-first search (BFS) is conducted from each ring node in the graph
- BFS is terminated when a neighbor is not a ring member
- BFS-traversal the **number of newly found neighbors** at each level is recorded as well as the **number of neighbors that are revisited**

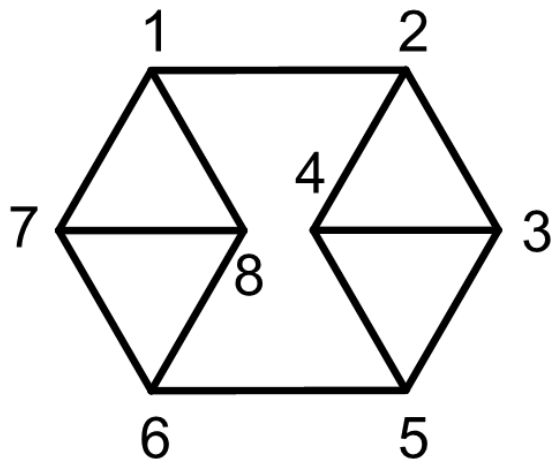
After



Two new invariants

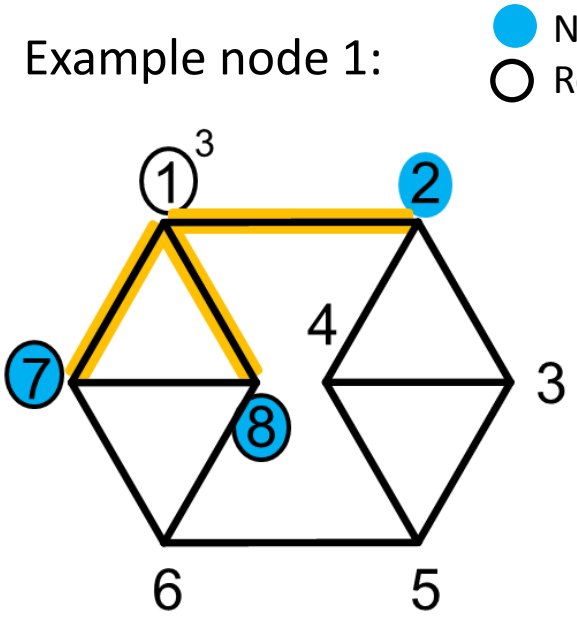
Handling of highly symmetrical molecules

- Calculate number of **newly found neighbors** (NN) at each level and number of **neighbors that are revisited** (RN) using BFS search



Using the standard atom invariant all atoms are the same

Example node 1:



1. Iteration: Node 1:

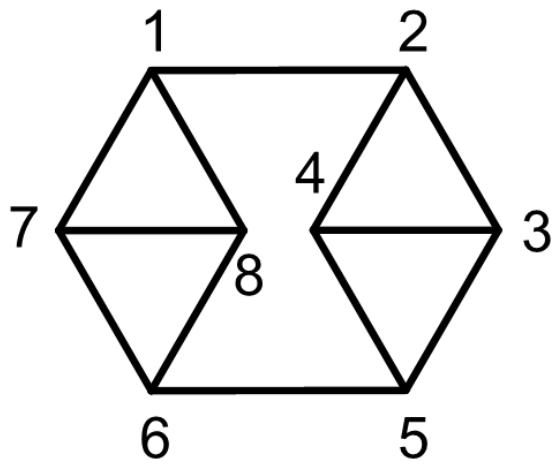
- # new neighbors: 3
- # of rev. neighbors: 1,1,3

Final result node 1:
NN: 3
RN: 113

Two new invariants

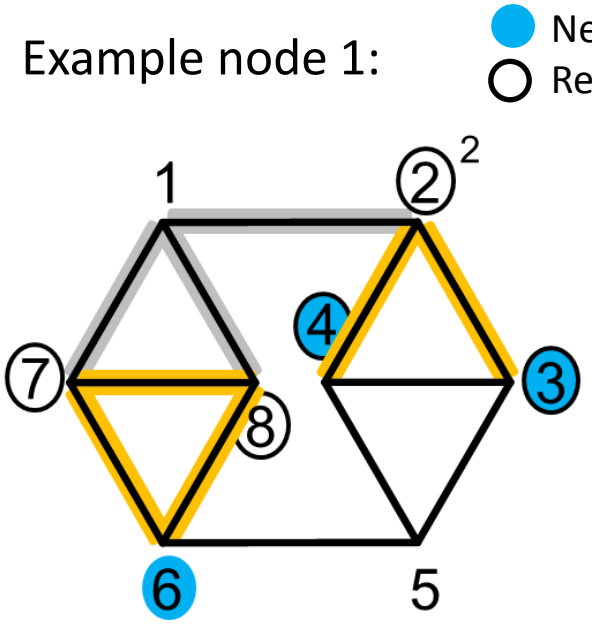
Handling of highly symmetrical molecules

- Calculate number of **newly found neighbors** (NN) at each level and number of **neighbors that are revisited** (RN) using BFS search



Using the standard atom invariant all atoms are the same

Example node 1:



2. Iteration: Node 1:

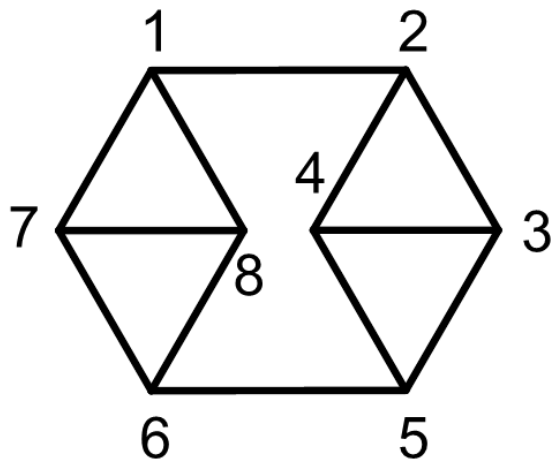
- # new neighbors: 3
- # of rev. neighbors: 1,1,1,1,2

Final result node 1:
NN: 33
RN: 113|11112

Two new invariants

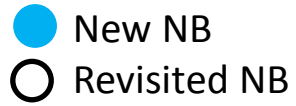
Handling of highly symmetrical molecules

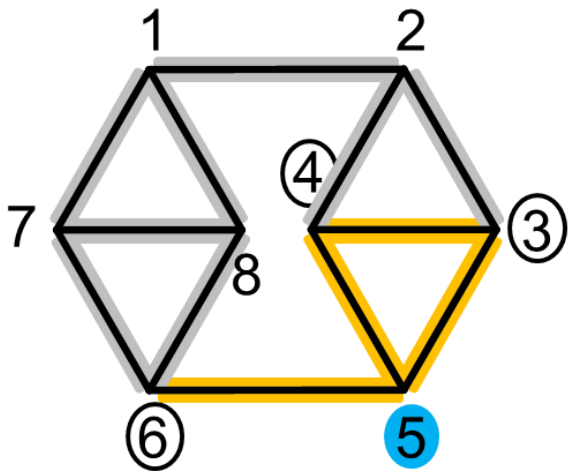
- Calculate number of **newly found neighbors** (NN) at each level and number of **neighbors that are revisited** (RN) using BFS search



Using the standard atom invariant all atoms are the same

Example node 1:





3. Iteration: Node 1:

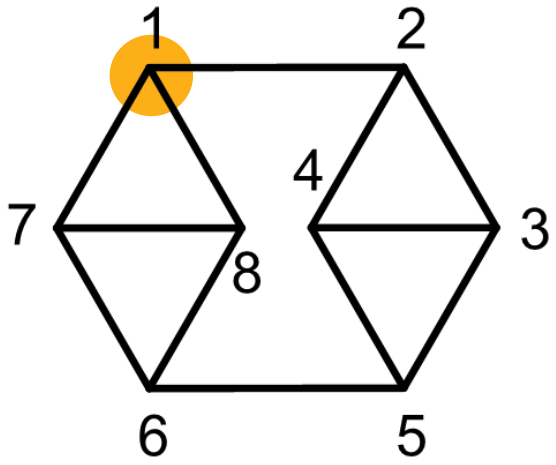
- # new neighbors: 1
- # of rev. neighbors: 1,1,1

Final result node 1:
NN: 331
RN: 113|11112|111

Two new invariants

Handling of highly symmetrical molecules

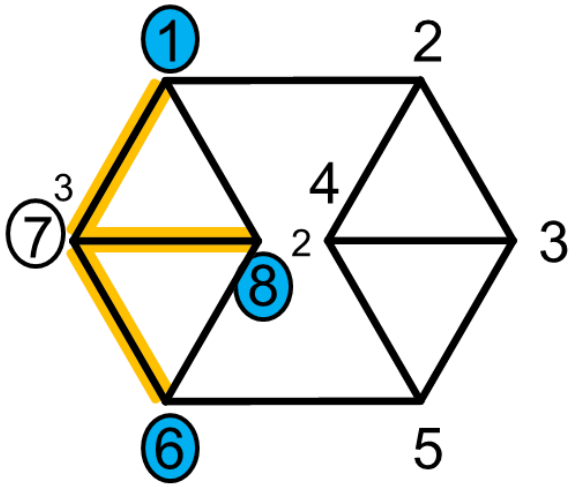
Example node 1:



Final result node 1:
NN: 331
RN: 113|11112|111

Example node 7:

New NB
Revisited NB



1. Iteration: Node 7:

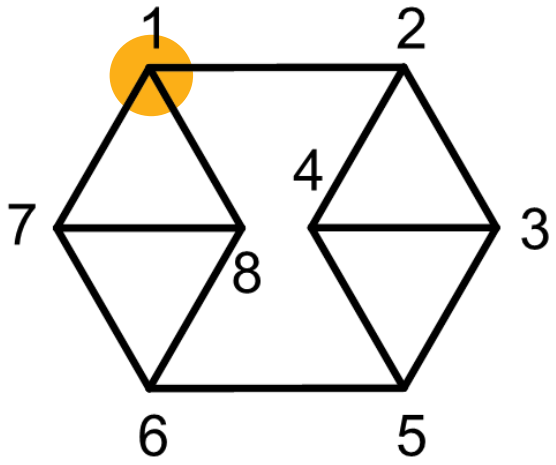
- # new neighbors: 3
- # of rev. neighbors: 1,1,2,3

Final result node 7:
NN: 3
RN: 1123

Two new invariants

Handling of highly symmetrical molecules

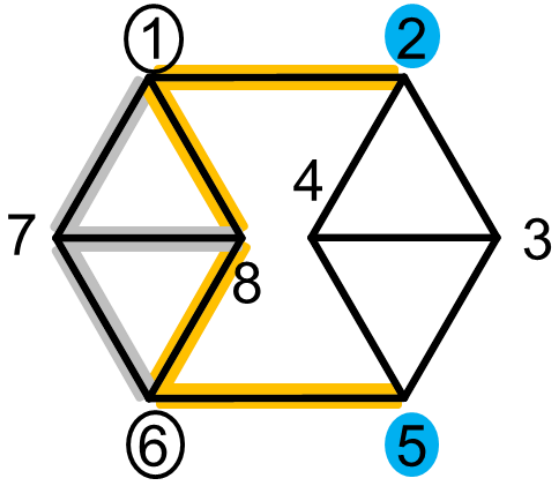
Example node 1:



Final result node 1:
NN: 331
RN: 113|11112|111

Example node 7:

New NB
Revisited NB



2. Iteration: Node 7:

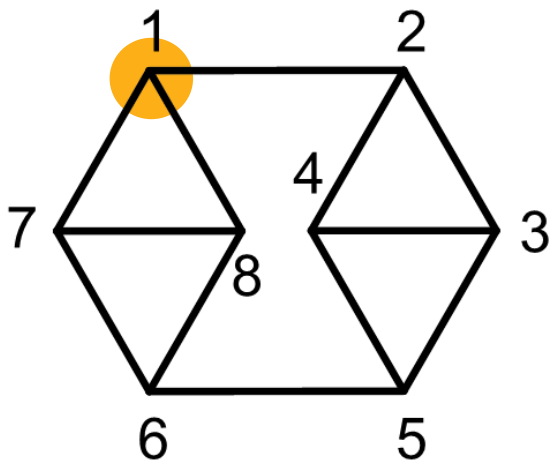
- # new neighbors: 2
- # of rev. neighbors: 1,1

Final result node 7:
NN: 32
RN: 1123|11

Two new invariants

Handling of highly symmetrical molecules

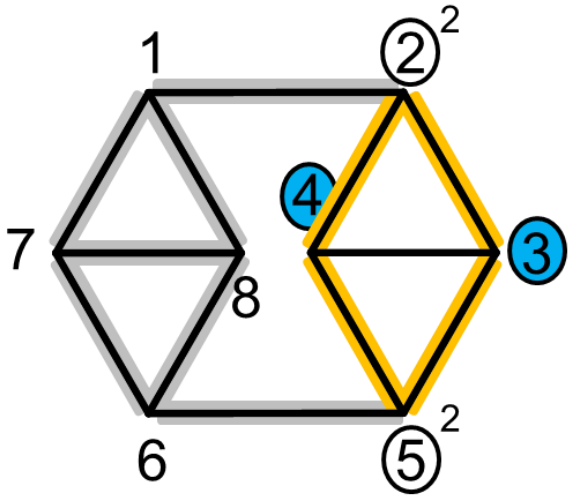
Example node 1:



Final result node 1:
NN: 331
RN: 113|11112|111

Example node 7:

New NB
Revisited NB



3. Iteration: Node 7:

- # new neighbors: 2
- # of rev. neighbors: 1,1,2,2

Final result node 7:
NN: 322
RN: 1123|11|1122

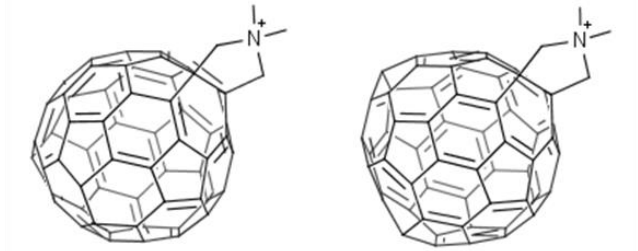
Validation of the robustness and correctness

Renumbering and SMILES round tripping tests

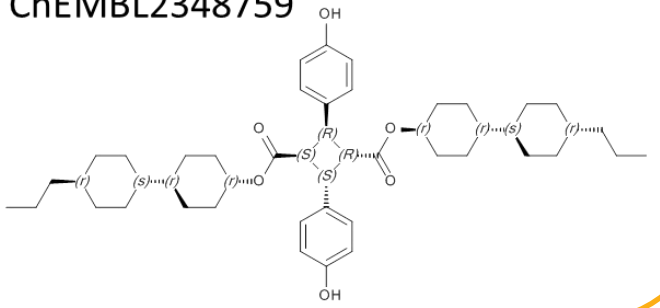
- ChEMBL 20 dataset[1] which consists of 1,456,020 different molecules taken from the medicinal chemistry literature and other public sources
- Random renumbering is repeated 50 times for each molecule

ChEMBL [sdf]	# molecules	# errors
Parsing	1,456,020	256
Canonicalization	1,455,764	2
Previous release 2014.09		
Parsing	1,456,020	257
Canonicalization	1,455,763	**755**

ChEMBL415840



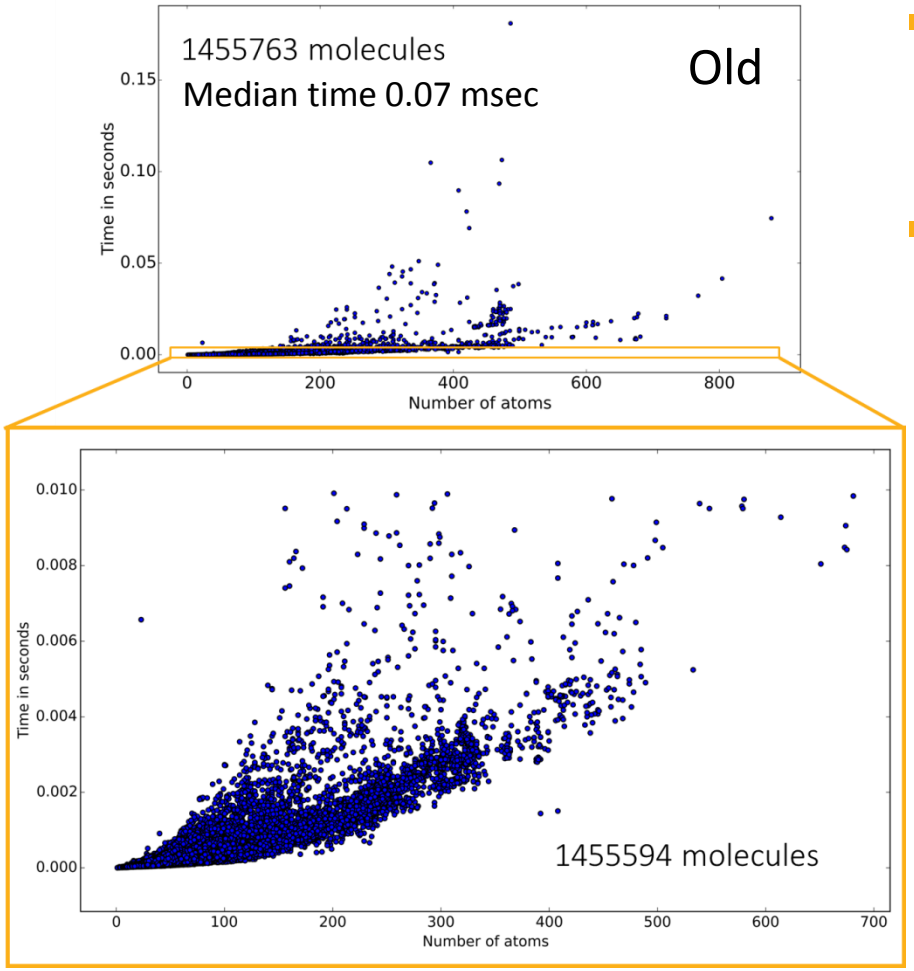
ChEMBL2348759



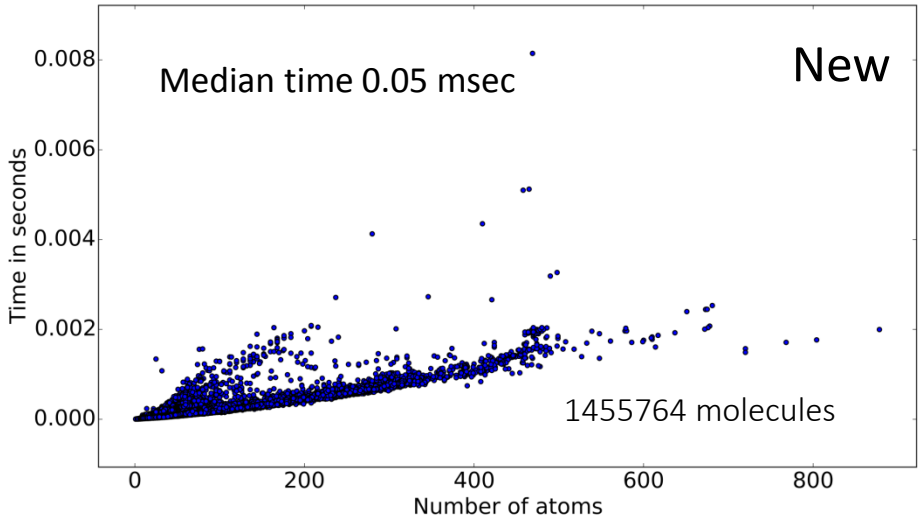
[1] Bento PA, et al.: The ChEMBL bioactivity database: an update. Nucleic acids research 2014, 42.D1: D1083-D1090.

Runtime evaluation

Comparing old and new RDKit canonicalization algorithm



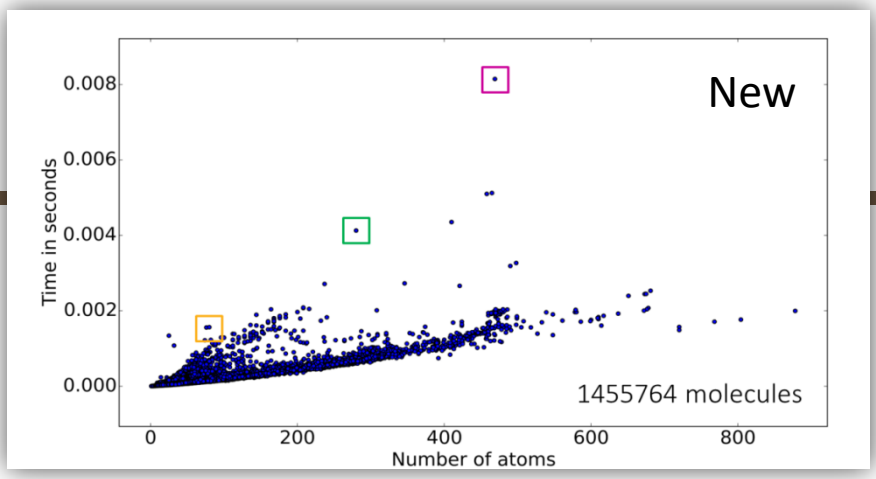
- ChEMBL 20 dataset: represents a broad range of different molecule sizes and complexities
- 40% improvement in runtime:
  - Old: 128 sec, New: 77 sec
  - Less outliers, better scaling



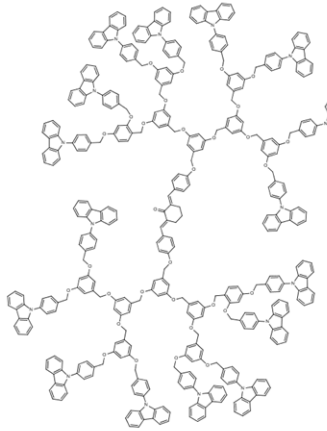
NOVARTIS

Runtime evaluation

Looking at the outliers



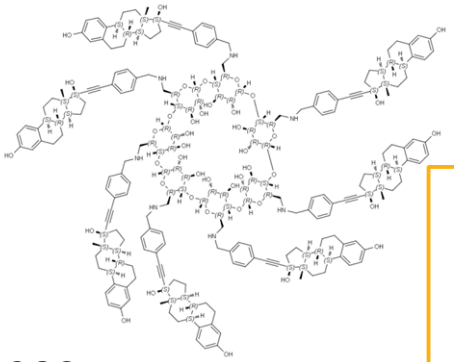
ChEMBL1077020



469 atoms

Tie breaking does not propagate through the molecule

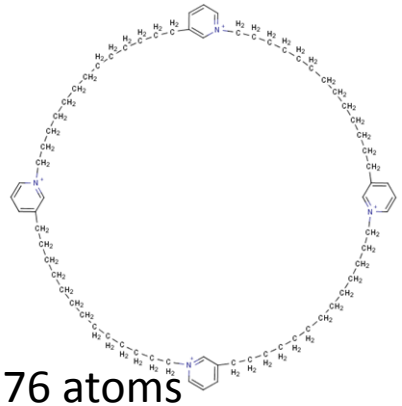
ChEMBL604989



280 atoms

Application of both new expensive invariants

ChEMBL2078656



76 atoms

Application of high-symmetry invariant for all atoms (all atoms are in a ring!)

NOVARTIS

Conclusions and Outlook

Efficient and robust alternative to existing published approaches

- Partition-based graph relaxation algorithm and a stable index of the equivalence classes allows iterative refinement
- Does not require the calculation of unique attributes like polynomials or products of primes
- Two new invariants allow assigning canonical labels even to highly symmetrical molecules
- Better performance particularly for larger molecules, e. g. proteins
- Canonical ranking of the atoms could be used to:
  - Generate a canonical Kekulé form of aromatic rings
  - Allow creating canonical MOL files
  - Construction of a molecular hash code
  - Learn about equivalence classes of the atoms by turning off tie breaking

Acknowledgements

- NIBR
  - Greg Landrum
  - Brian Kelley
  - Bernd Rohde
  - NIBR Education office for funding

- NextMove Software:
  - Roger Sayle

...and Sereina for organizing a great meeting!

NOVARTIS