



Schrödinger

2D Coordinate generation in RDKit

Current state & goals for the future

Rachel Walker

rachel.walker@schrodinger.com

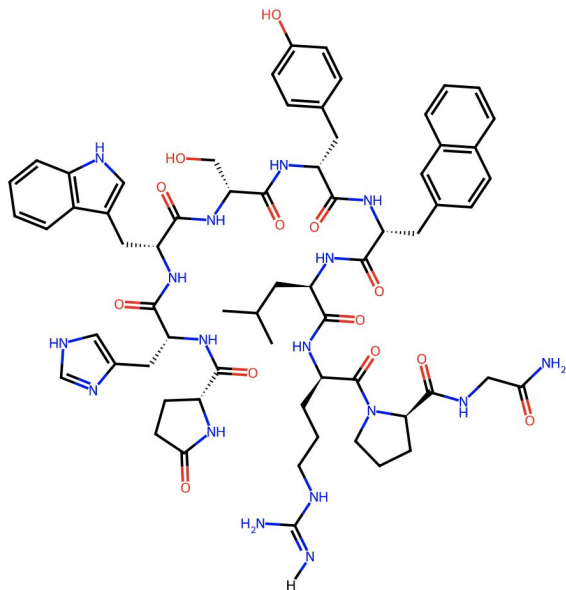
RDKit UGM 2022



Current 2D coordinate generation methods in RDKit

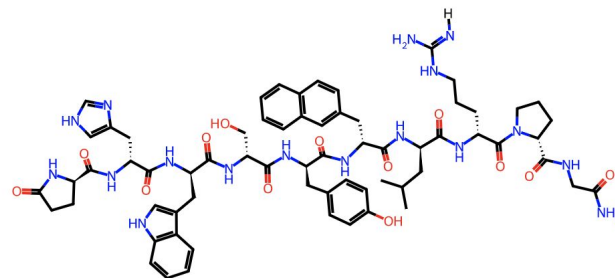
RDKit native method

```
# Default is False  
Chem.rdDepictor.SetPreferCoordGen(False)  
Chem.rdDepictor.Compute2DCoords(m)  
m
```

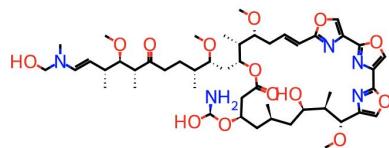


Coordgenlibs

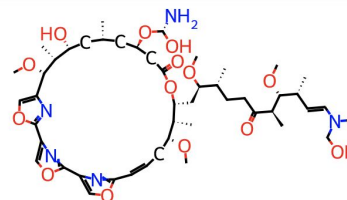
```
Chem.rdDepictor.SetPreferCoordGen(True)  
Chem.rdDepictor.Compute2DCoords(m) # Or Chem.rdCoordGen.AddCoords(m)  
m
```



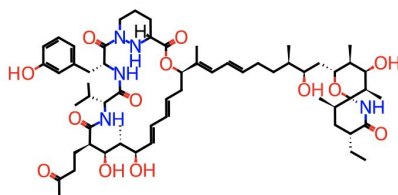
RDKit native method vs Coordgenlibs: Macrocycles



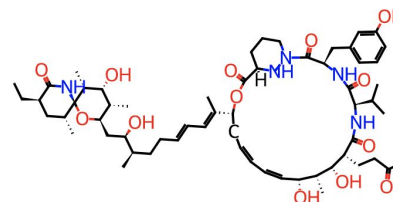
Coordgen



RDKit



Coordgen



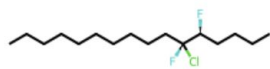
RDKit

RDKit native method vs Coordgenlibs: Chain layout

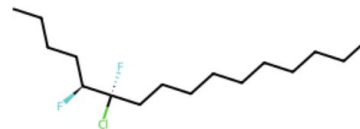


greglandrum commented on Sep 9

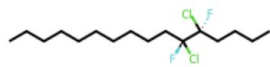
Here are a couple of more examples where the coordgen handling of chains + four coordination is just much better:



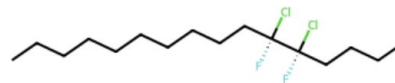
coordgen



rdkit



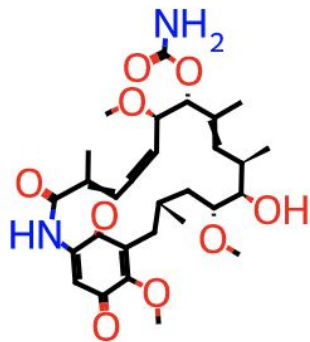
coordgen



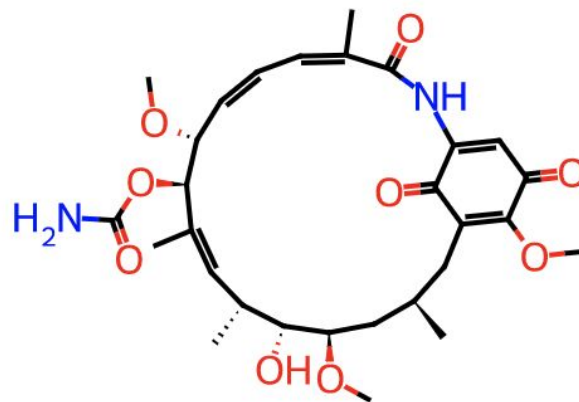
rdkit



RDKit native method vs Coordgenlibs: Macrocycles with cis/trans bonds

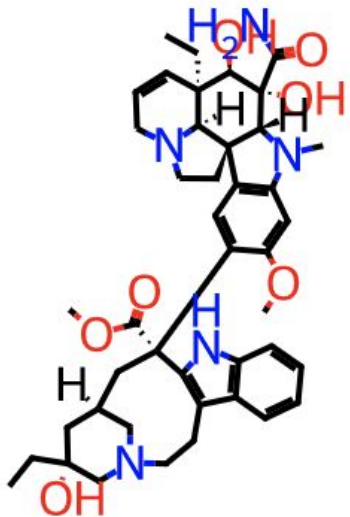


Coordgen

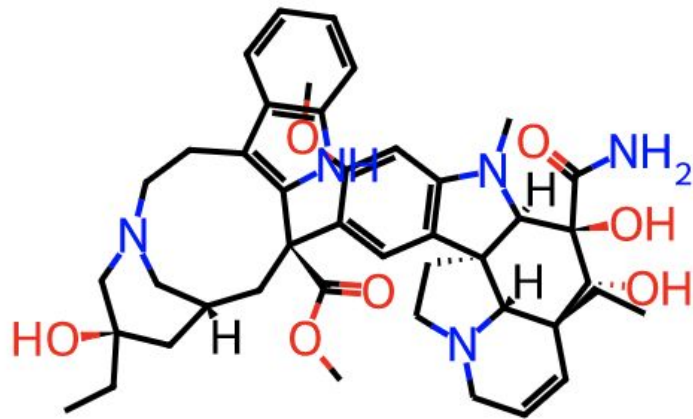


RDKit

RDKit native method vs Coordgenlibs: Handling collisions

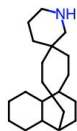


Coordgen



RDKit

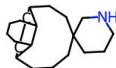
RDKit native method vs Coordgenlibs: Complex ring systems (templating)



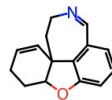
Template #16



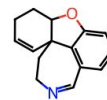
Coordgen



RDKit



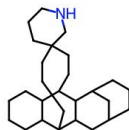
Template #29



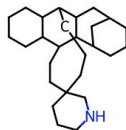
Coordgen



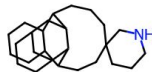
RDKit



Template #17



Coordgen



RDKit



Template #74

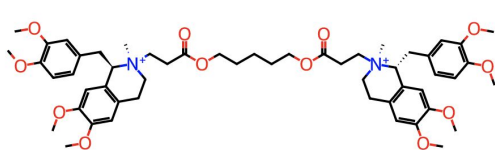


Coordgen

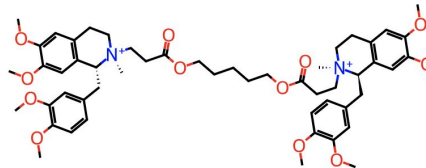


RDKit

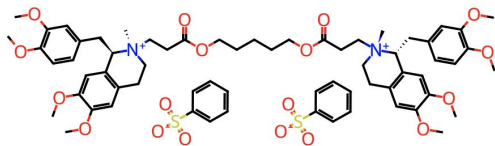
RDKit native method vs Coordgenlibs: Symmetry, multiple molecules



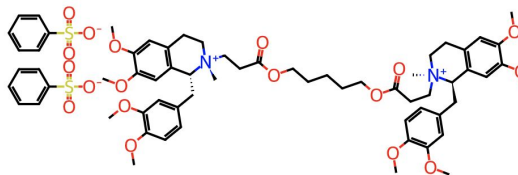
Coordgen



RDKit



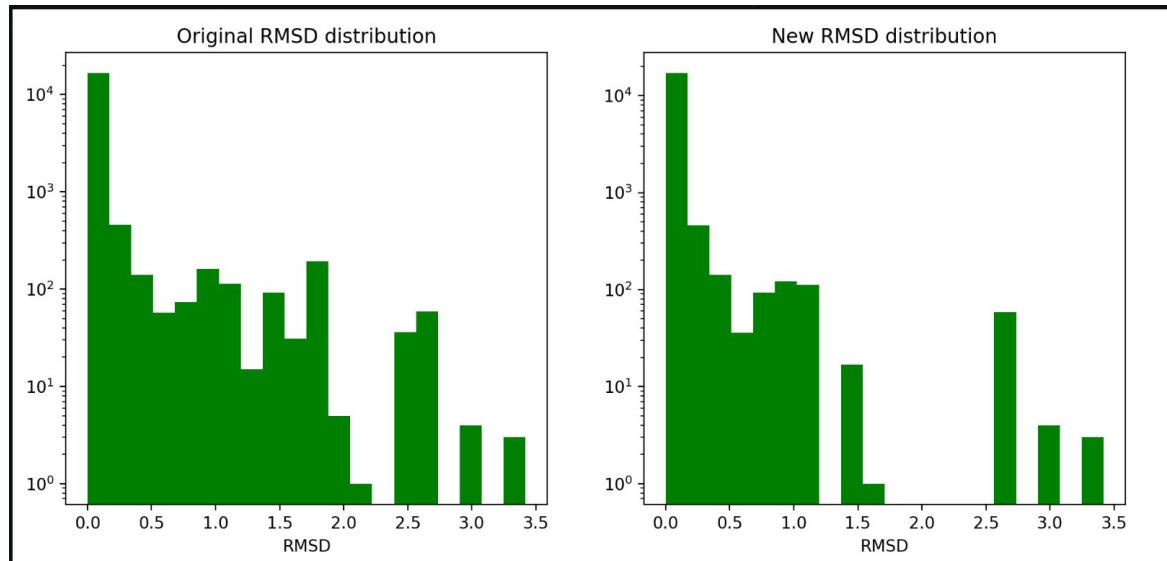
Coordgen



RDKit

Recent-ish Coordgenlibs Improvements

- PR #95: Early stopping in “force field” minimization (2x performance improvement)
- PR #98: Fixing some distorted bridged ring systems
- PR #93: Improving alignment to provided templates
- Some other small bug fixes, general maintenance & adding templates



PR #93: Aligning
to substructures

Issues with current 2D coordinate generation methods

RDKit native method

- Macrocycle layout (and handling cis/trans bonds)
- Chain layout
- No default templating
- General quality

Coordgenlibs

- Performance can be very poor for large structures (especially with macrocycles)
- For smaller structures, usually ~10x slower than RDKit
- Difficult to maintain, would be very nice to use RDKit mol
- No strict alignment with user-provided template

Future goals for 2D coordinate generation

1. Collapse coordinate generation methods: for easier maintainability, use RDKit mol
2. Performance improvement: closer to performance of RDKit coordinate generation (especially for non-macrocycles)
3. Overall: get the “best of both worlds” from previous slide

Why does Schrodinger care?

- We do a LOT of 2D coordinate generation – performance is very important & 2D coordinate generation can be a bottleneck (LiveDesign, 3D conformer generation)
- Coordgenlibs is great but has been difficult to maintain. Now that it is used in LiveDesign we have been seeing a lot more bugs, and it is a roadblock in goal to open source sketcher

Acknowledgements & other notes

Nic Zonta

Dan Nealschneider

John Mayfield

Greg Landrum

Paolo Tosco

Chris Von Bargen

Ricardo Rodriguez-Schmidt

& other contributors to RDKit

and coordgenlibs





Schrödinger

Thank you!