Dealing with organometallic molecules in RDKit

Jan H. Jensen
Department of Chemistry,
University of Copenhagen

@janhjensen

2020 RDKit UGM
2020.10.06

Most homogeneous catalysts are organometallic compounds
Large datasets are becoming available but in xyz format

Most cheminformatics/ML relies on SMILES/graphs
(e.g. substructure searching and graph convolution)

The tmQM Dataset - Quantum Geometries and Properties of 86k Transition Metal Complexes

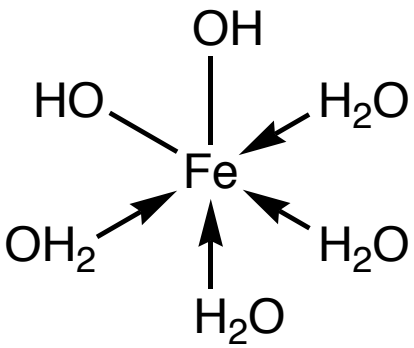
David Balcells[*,†] and Bastian Bjerkem Skjelstad[‡]

chemrxiv.12894818.v1

Using SMILES strings for the description of chemical connectivity in the Crystallography Open Database

Miguel Quirós[1*], Saulius Gražulis[2,3], Saulė Girdzijauskaitė[3], Andrius Merkys[2] and Antanas Vaitkus[2]

the previously published `cif_molecule` program is used to get such image in many cases. The program package *Open Babel* is then applied to get SMILES strings from the CIF files (either those directly taken from the COD or those produced by `cif_molecule` when applicable). The results are then checked and/or fixed by a human editor, in a computer-aided task that at present still consumes a great deal of human time. Even if the procedure still needs to be



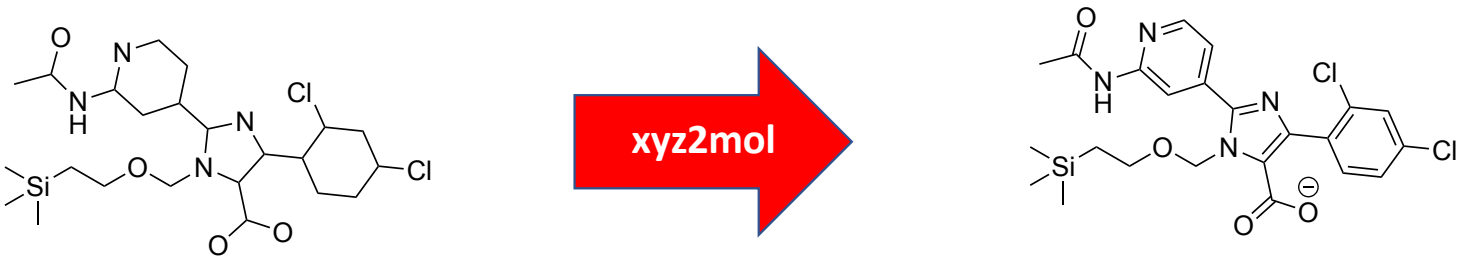
[OH2][Fe](O)(O)([OH2])([OH2]) [OH2]

Not readable by RDKIT

Charge from SMILES often incorrect

J Cheminform (2018)

xyz2mol for organic compounds



xyz2mol converts an xyz file to an RDKit mol object
(needs the molecular charge and hydrogens)

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

Yeonjoon Kim and Woo Youn Kim*

github.com/jensengroup/xyz2mol

Organic examples

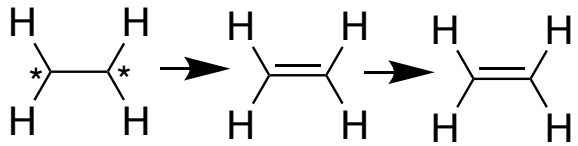
Table 1. Atomic valences.

Elements	N_v	Elements	N_v
H	1	F, Cl, Br	1
B	3	N	3 or 4
C	4	P	3, 4, or 5
O	1 or 2	S	2, 4, or 6

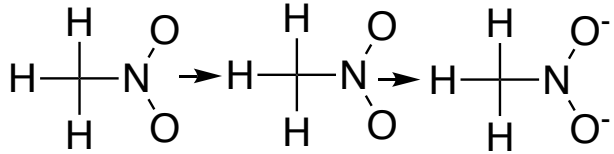
Valence states	Formal charge
Carbon with three single bonds	1/−1 depending on the total charge
Boron	3 − (no. of bonds)
The rest	(no. of valence electrons) − 8 + (no. of bonds)

valence

[4, 4, 1, 1]

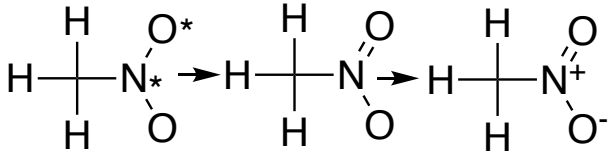


[1, 3, 1, …]



X

[2, 4, 1, …]



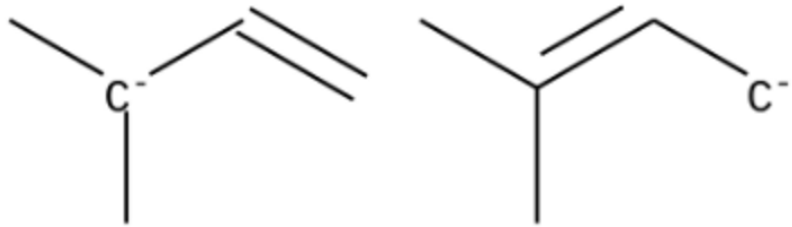
and

$\sum_a q_a = Q_{mol}$?

YES

Sometimes there are more than one solution

xyz2mol will generate one of them arbitrarily



Solution(?): generate all, then filter

Generate all using rdchem.ResonanceMolSupplier*

Create filter that picks “canonical” form

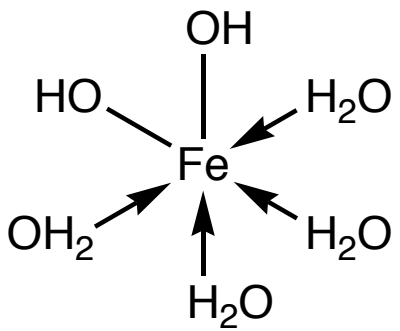
*HT Mads Koerstz

One approach for organometallics

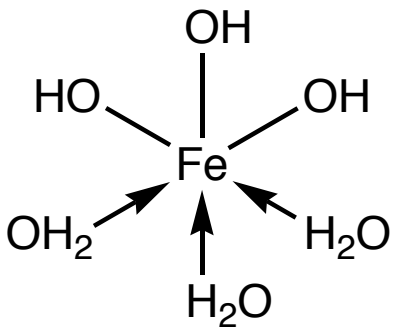
Distinguishing dative from covalent bonds

O->[Fe](O)(O)(<-O)(<-O)<-O

O->[Fe](O)(O)(O)(<-O)<-O



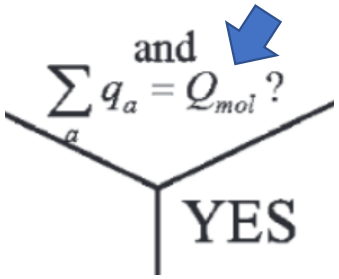
Fe^{+2}



Fe^{+3}

Formal charge on Fe = total charge

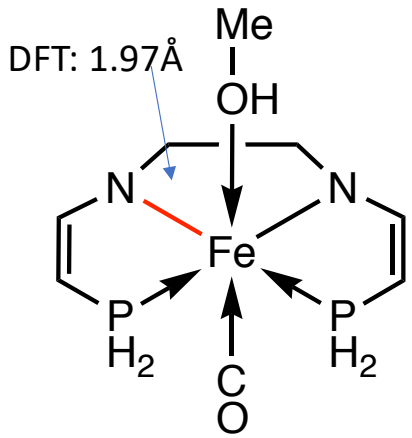
[e.g. $Fe(OH)_2^+$]

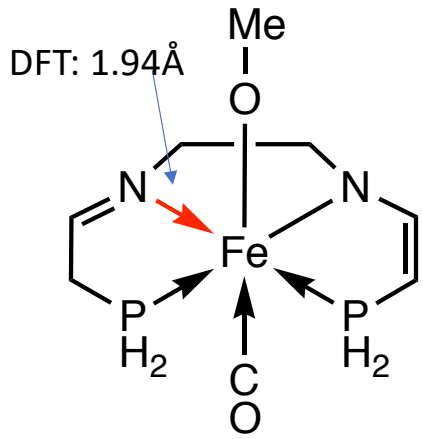


Main problem

Distinguishing dative from covalent bonds

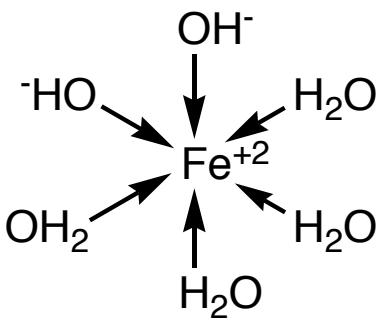
before bond orders are assigned

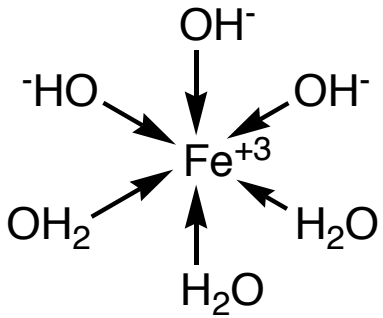




Another approach

Only dative bonds





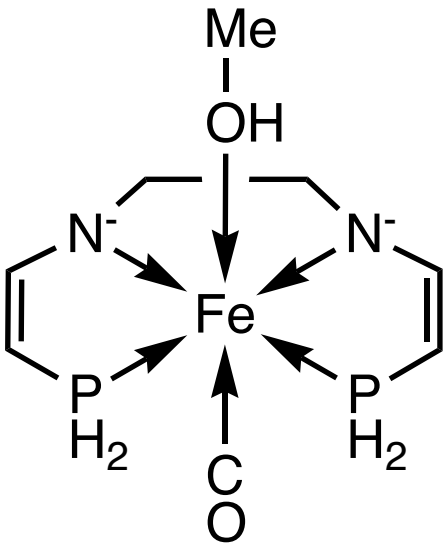
Fe^{+2} Fe^{+3}

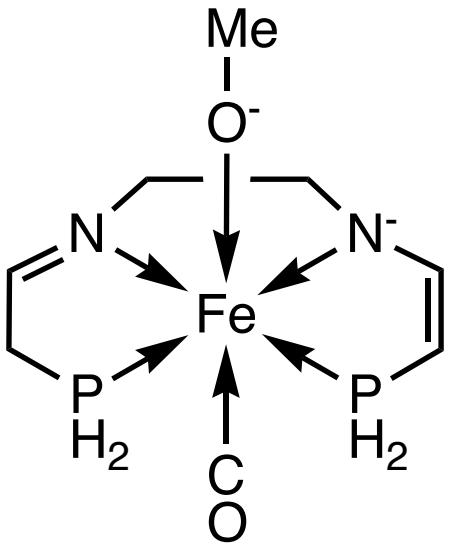
Formal charge in Fe = total charge + ∑ charge on ligands

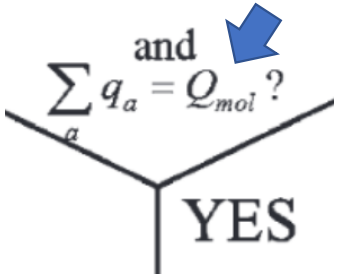
Also not sensitive to presence of bond(s)

Alex Clark *J. Chem. Inf. Modeling* 2011

Must know charge on Fe

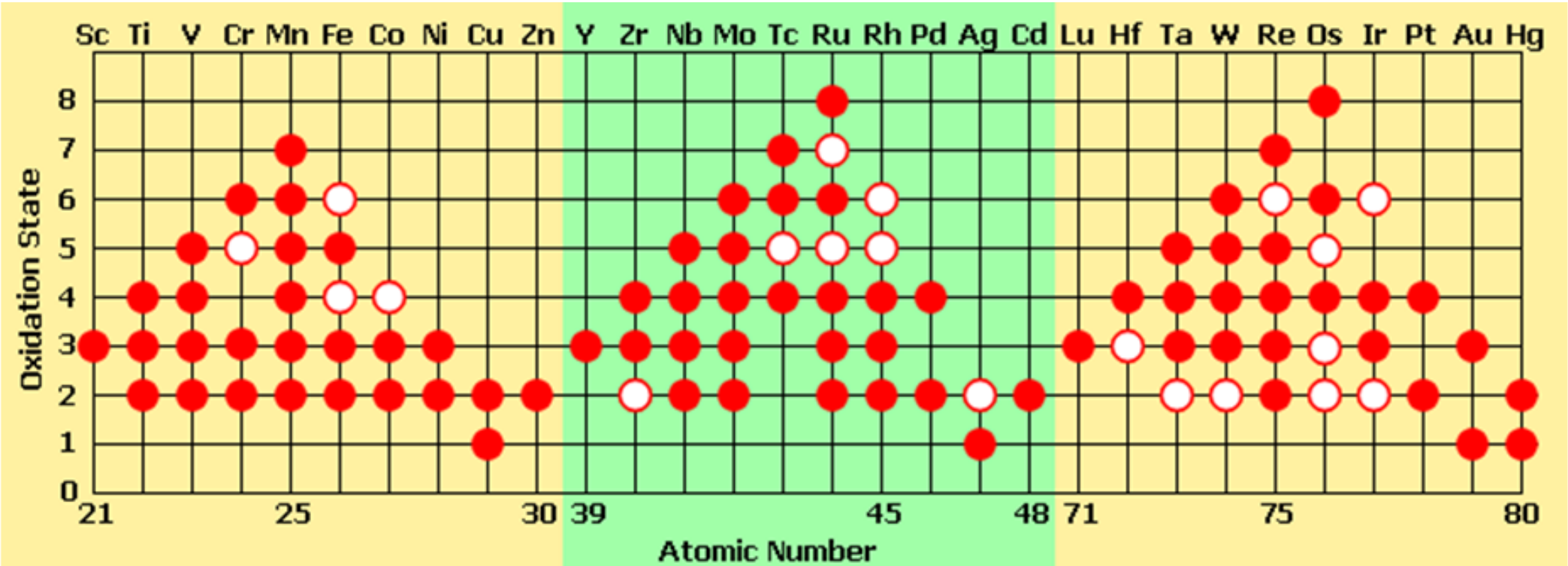






total charge = charge on Fe + ∑ charge on ligands

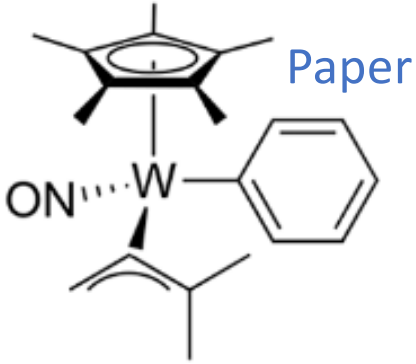
Most TMs have many different oxidation states



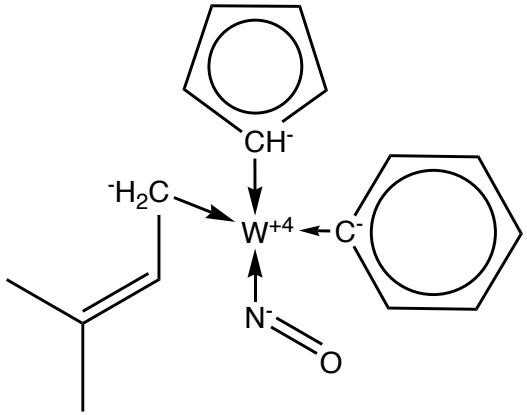
https://byjus.com/chemistry/transition-elements-oxidation-states/

Try all charges and save cases for which

total charge = charge on TM + ∑ charge on ligands



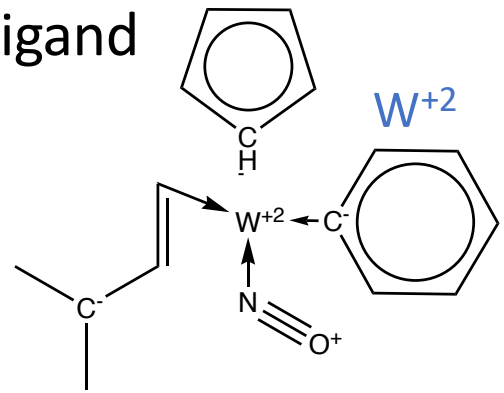
Paper



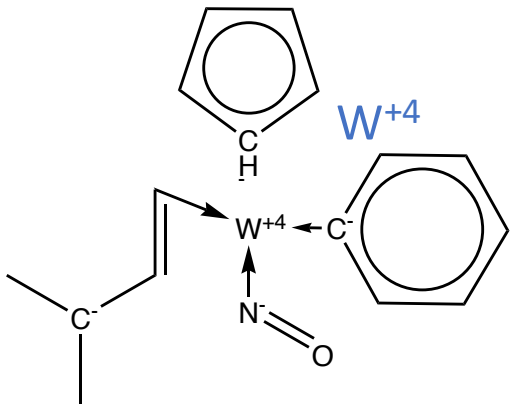
Expect (W^{+4})

Missing bond to W
"wrong" resonance
form of ligand

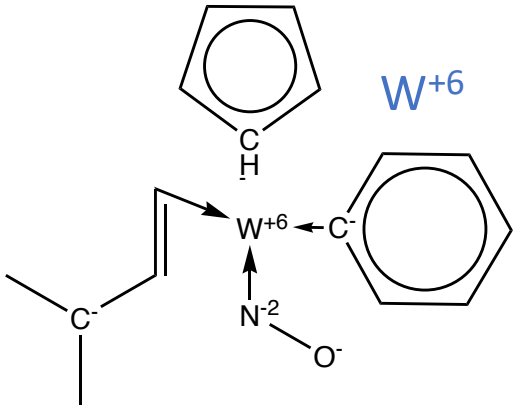
Get



W^{+2}



W^{+4}



W^{+6}

Some issues

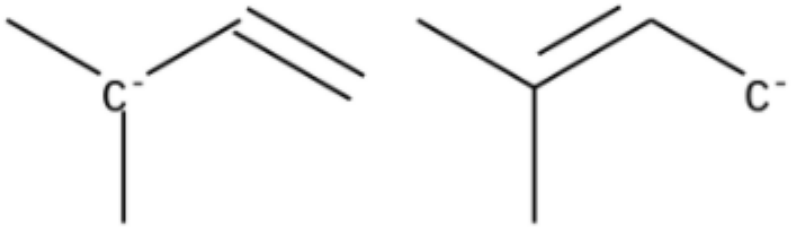
Not all bonds to TM are found
(uses RDKit Hückel reduced overlap population)

Other resonance forms of ligands

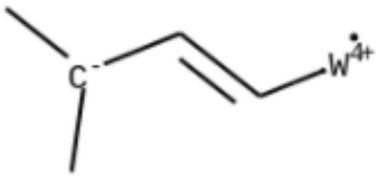
Hydrides

Dative bond limits ResonanceMolSupplier

```
m = Chem.MolFromSmiles('C[C-](C)C=C')
ms = rdchem.ResonanceMolSupplier(m)
Draw.MolsToGridImage(ms,molsPerRow=5,legends=None,subImgSize=(200,200))
```




```
m = Chem.MolFromSmiles('C[C-](C)C=C->[W+4]')
ms = rdchem.ResonanceMolSupplier(m)
Draw.MolsToGridImage(ms,molsPerRow=5,legends=None,subImgSize=(200,200))
```



Fragment -> Resonance structures -> combine(?)

Hydrides and SMILES (RDKit 2020.03.5)

```
mol = Chem.MolFromSmarts('[#26]')
rwMol = Chem.RWMol(mol)
rwMol.AddAtom(Chem.Atom(1))
rwMol.AddBond(1,0,Chem.BondType.DATIVE)
mol = rwMol.GetMol()
mol.GetAtomWithIdx(0).SetFormalCharge(2)
mol.GetAtomWithIdx(1).SetFormalCharge(-1)
print('SMILES and total charge', Chem.MolToSmiles(mol),Chem.GetFormalCharge(mol))
print('allHsExplicit', Chem.MolToSmiles(mol, allHsExplicit=True))
print('number of atoms', mol.GetNumAtoms())
print('charge on Fe and H', mol.GetAtomWithIdx(0).GetFormalCharge(), mol.GetAtomWithIdx(1).GetFormalCharge())
```

```
SMILES and total charge [HH2-]->[Fe+2] 1
allHsExplicit [HH2-]->[Fe+2]
number of atoms 2
charge on Fe and H 2 -1
```

Hydrides and SMILES (RDKit 2020.03.5)

SMILES does not give mol object with correct charge

```
mol = Chem.MolFromSmiles('[HH2-]->[Fe+2]')
print('SMILES and total charge', Chem.MolToSmiles(mol),Chem.GetFormalCharge(mol))
```

```
SMILES and total charge [FeH+2] 2
```

```
mol = Chem.AddHs(mol)
print('SMILES and total charge', Chem.MolToSmiles(mol),Chem.GetFormalCharge(mol))
```

```
SMILES and total charge [H][Fe+2] 2
```

`[HH2-]->[Fe+2]` becomes `[FeH+2]` or `[H][Fe+2]`

Hydrides and SMILES (RDKit 2020.03.5)

Greg found a workaround

```
parse_ps = Chem.SmilesParserParams()
parse_ps.removeHs=False
remove_ps = Chem.RemoveHsParameters()
remove_ps.removeHydrides = False
m = Chem.RemoveHs(Chem.MolFromSmiles('[HH2-]->[Fe+2]',parse_ps),remove_ps)
print(Chem.MolToSmiles(m),Chem.GetFormalCharge(m) )
```

```
[HH2-]->[Fe+2] 1
```

Another option is to treat TM-hydride bonds as covalent and reduce TM charge

`[FeH+1]` instead of `[HH2-]->[Fe+2]`

Summary

Prototype generates RDKit readable SMILES
for organometallic compounds w/o human intervention

But …

Not all bonds to TM are found
(uses RDKit Hückel reduced overlap population)

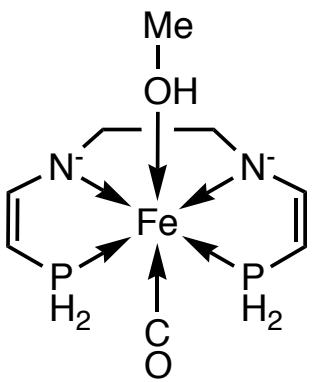
Non-unique oxidation states/resonance forms
(filter/”canonicalization”?)

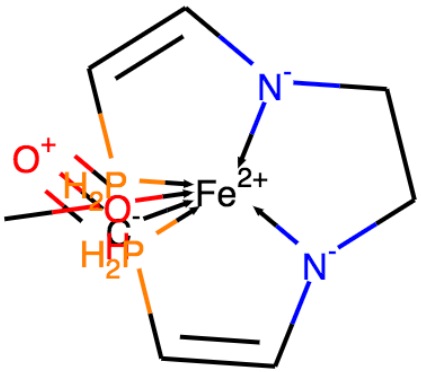
Hydrides charge bug for MolFromSmiles

How to automatically test code?

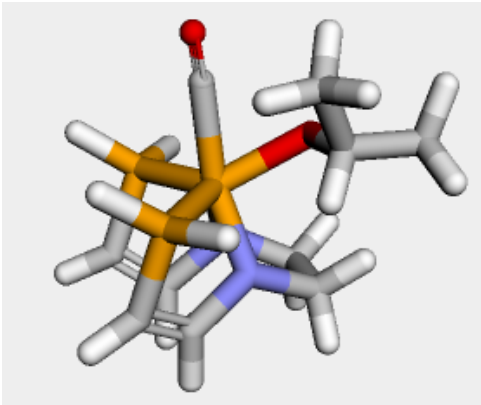
Additional RDKit issues

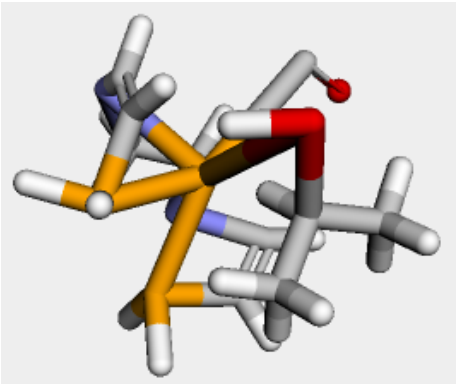
Depiction of octahedral compounds not helpful





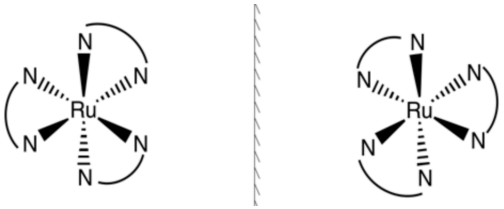
Embedding/UFF optimization not working

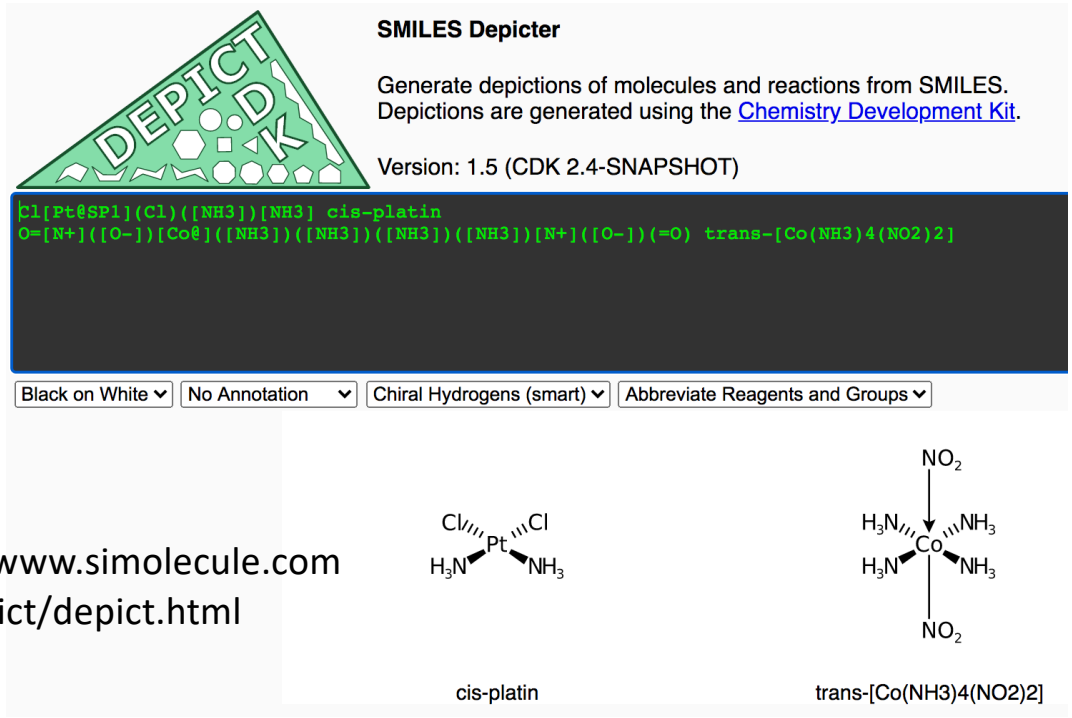




Additional issues continued

Specifying stereochemistry





https://www.simolecule.com/cdkdepict/depict.html

Experimental branch can be found here

Feedback welcome

https://github.com/jensengroup/xyz2mol/tree/tm_comb