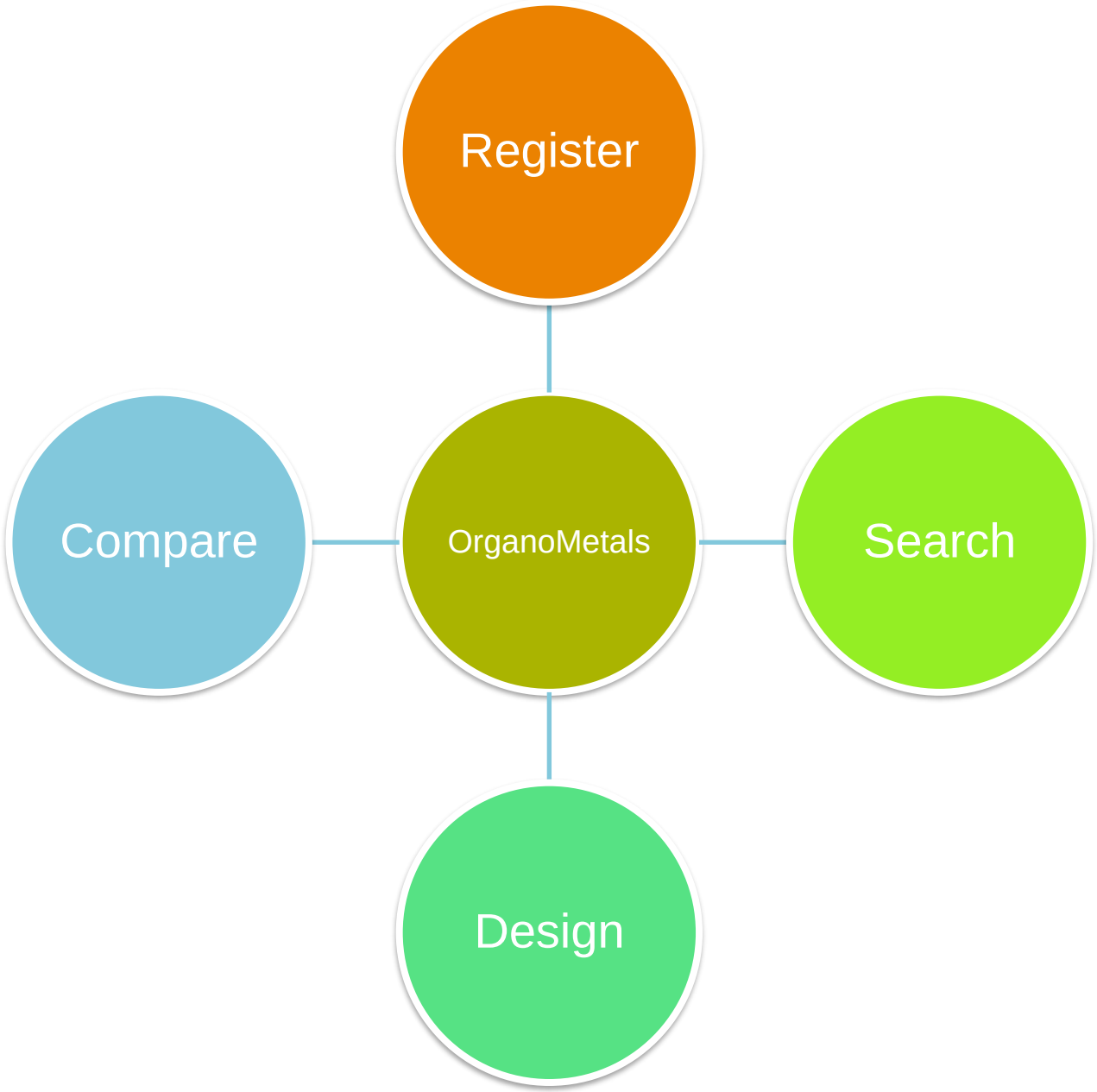
marco.stenta@syngenta.com

CATLAS: Catalyst Atlas

Marco Stenta (Syngenta Crop Protection)
on behalf of the
Data 4 Synthesis Project team and
Computer-Aided Synthesis Team

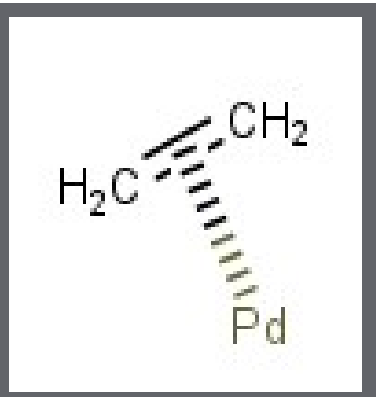
12.09.2024



Lean on Open-Source Technology and support OM handling

```
Mrv2115 10102311132D

  0  0  0     0  0            999 V3000
M  V30 BEGIN CTAB
M  V30 COUNTS 4 2 0 0 0
M  V30 BEGIN ATOM
M  V30 1 C -39.4583 13.4167 0 0
M  V30 2 C -38.1247 14.1867 0 0
M  V30 3 * -38.7915 13.8017 0 0
M  V30 4 Pd -37.9477 11.8652 0 0
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 2 1 2
M  V30 2 9 3 4 ENDPTS=(2 1 2) ATTACH=ALL
M  V30 END BOND
M  V30 END CTAB
M  END
```



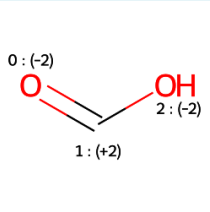
Extend RDKit Support to organometallic structures

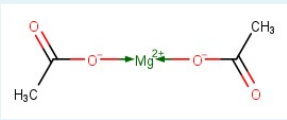
RDKit 2023.3

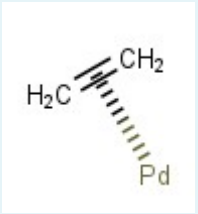
David Cosgrove
CozChemIx

Greg Landrum

Oxidation state calculation
Support for multicenter bonds (molv3000)
Extended support for dative bonds across the periodic table
Metal disconnection algorithm



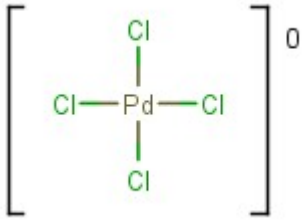




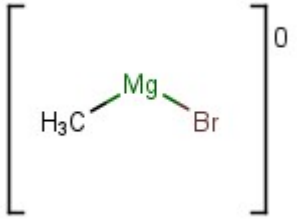
Input Structure: consistency and correctness over familiarity

The charge on M depends
of the overall charge:
need to specify it or use rules

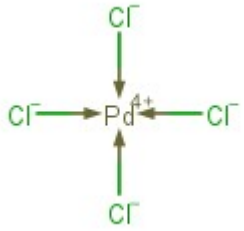
Covalent



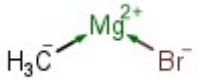
Cl[Pd](Cl)(Cl)Cl

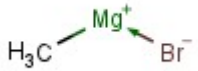


Ionic



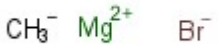
[Cl-]->[Pd+4](<-[Cl-])(<-[Cl-])<-[Cl-]



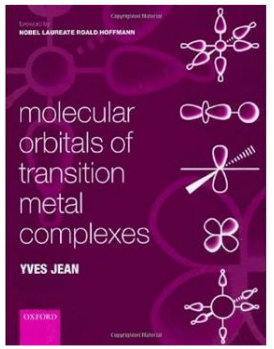


Disconnected

[Cl-].[Cl-].[Cl-].[Cl-].[Pd]



- Separate the requirements for inputting a structure to those of outputting them
  - Input **consistency**
  - Output: **freedom** (depiction)



Automatic Annotation from Chemical Structure

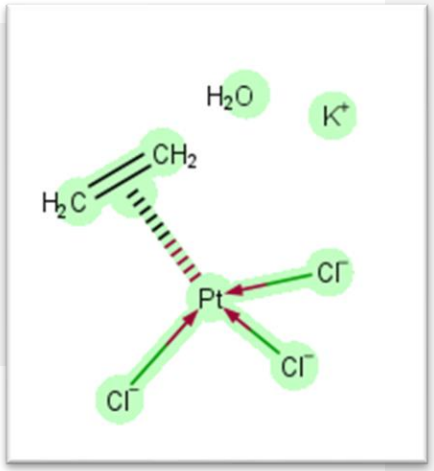
```
# read in a molecule from a string using RDKit
rdmol = rdmol_from_string(input_string=molblock, inp_fmt="molblock")

# inspect the molecular structure using the Inspector class
# this will return a StructuralAnalysis instance
inspector = Inspector(rdmol=rdmol)
structural_analysis = inspector.structural_analysis
structural_assessment = inspector.structural_assessment

# construct the organometallic molecule using the Constructor class
constructor = Constructor(structural_analysis=structural_analysis)
organometal = constructor.construct()
```

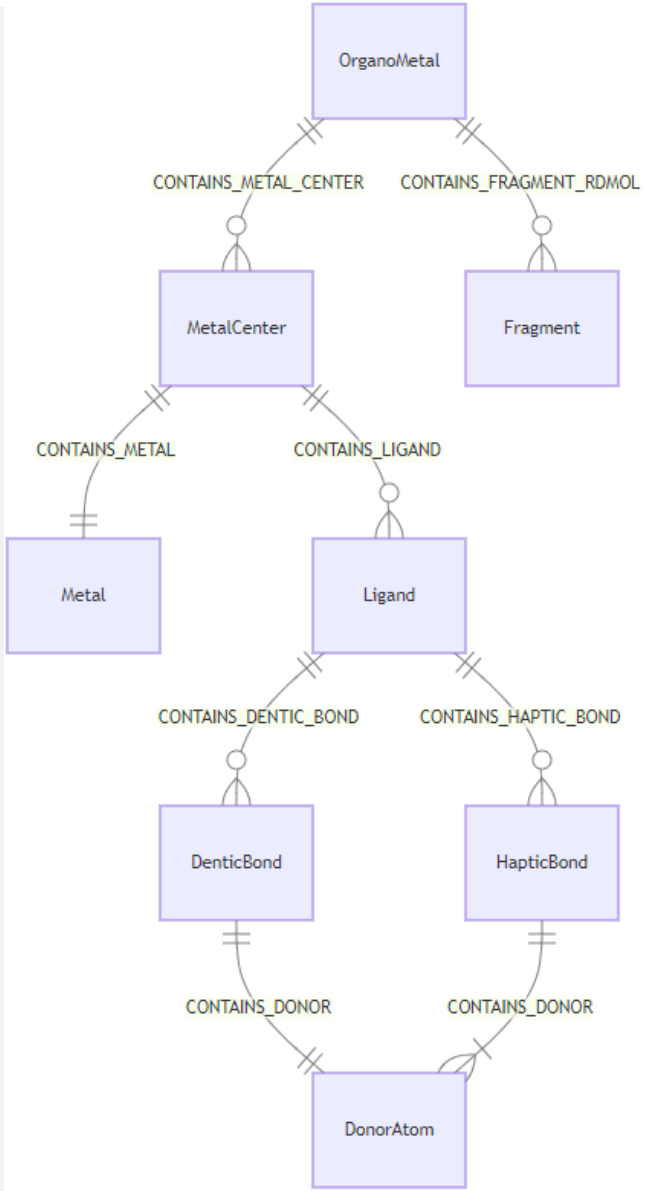
```
pprint.pprint(organometal.metal_centers[0].metal.labels)
pprint.pprint(organometal.metal_centers[1].metal.labels)
```

```
{'atomic_name': ['Platinum'],
 'availability_threat': ['3__rising_threat'],
 'electron_configuration_string': ['[Xe]6s1 4f14 5d9'],
 'metal_group_block': ['transition_metal'],
 'possible_oxidation_states_string': ['+2 +4']}
{'atomic_name': ['Potassium'],
 'availability_threat': ['1__no_threat'],
 'electron_configuration_string': ['[Ar]4s1'],
 'metal_group_block': ['alkali_metal'],
 'possible_oxidation_states_string': ['+1']}
```




```
"fragment_idx": 2,
                "formal_charge": 0,
                "dentic_bonds": [],
                "haptic_bonds": [
                  {
                    "dummy_atom_idx": 4,
                    "donors": [
                      {
                        "atom_idx": 2,
                        "fragment_idx": 2,
                        "fragment_atom_idx": 0,
                        "atomic_symbol": "C",
                        "atomic_number": 6,
                        "smarts_string": "C",
                        "ox_no": -2,
                        "formal_charge": 0,
                        "uid": "99914b932bd37a50b983c5e7c90ae93b"
                      },
                      {
                        "atom_idx": 3,
                        "fragment_idx": 2,
                        "fragment_atom_idx": 1,
                        "atomic_symbol": "C",
                        "atomic_number": 6,
                        "smarts_string": "C",
                        "ox_no": -2,
                        "formal_charge": 0,
                        "uid": "99914b932bd37a50b983c5e7c90ae93b"
                      }
                    ],
                    "uid": "99914b932bd37a50b983c5e7c90ae93b"
                  }
                ],
                "labels": {},
                "uid": "99914b932bd37a50b983c5e7c90ae93b"
              },
```


- OrganoMetal Data Model



Persistence: Register/Retrieve

Element properties

From the Periodic Table

- Atomic Number (z), Atomic Symbol, Atomic Mass
- Chemical Group Block
- Electronic Configuration
- Possible Oxidation states

Metal Labels

Provided by Users

- Natural Abundance/Scarcity
- Environmental/Toxicological risks
- ...

Metal properties

In the context of a specific complex

- Charge / Oxidation State
- Coordination number/geometry
- Geometric descriptors 2D>3D

Ligand Structure

- Substructure (ex by SMARTS patterns)
- Similarity (ex by SMILES query)
- Equivalence (ex by usign RDKit/NexMove MolHashes)
- Chirality (central, axial, partially unspecified, etc)

Ligand Labels

- Common names, synonyms
- Catalog names (ex CASNO)
- User-defined labels and tags

Ligand Binding Properties

In the context of a specific complex

- Denticity (number and nature of “donor” atoms)
- Hapticity (number and nature of “donor” atoms)

Acknowledgements

Edouard Godineau
Process Chemist
S&T Fellow

Serge Parel
Platform Manager &
Technical Lead

Guillaume Berthon
Change Lead Digitization
of Chemical Synthesis

Marco Stenta
Computational Chemist
S&T Fellow

Bringing plant potential to life