

RDKit Applications in Cheminformatics: Analysis of Natural Product Databases



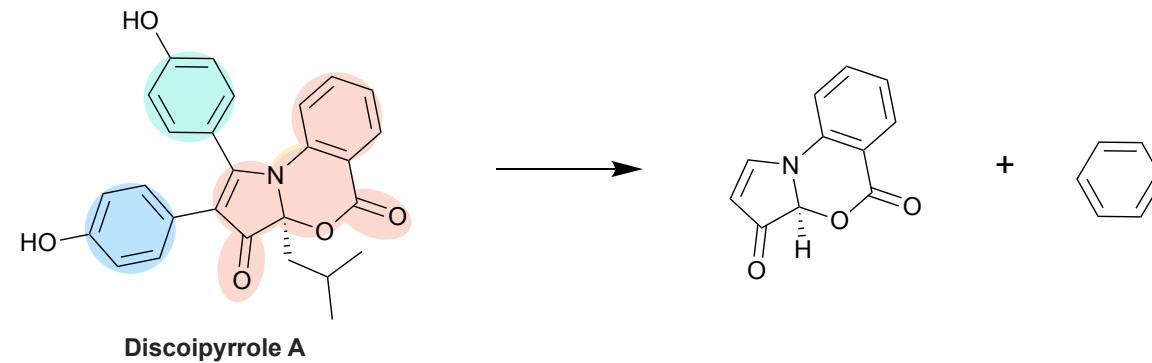
- > 50% of modern small-molecule drugs linked to natural products (NPs)¹
- Cheminformatics approaches are universally applied in NP research²
- Challenges in working with NPs:
 - Ring systems, which are abundant and can be complex³
 - Stereochemistry, which is often insufficiently described³
 - etc...

1. D. J. Newman and G. M. Cragg, *J. Nat. Prod.*, 2020, 83, 770–803

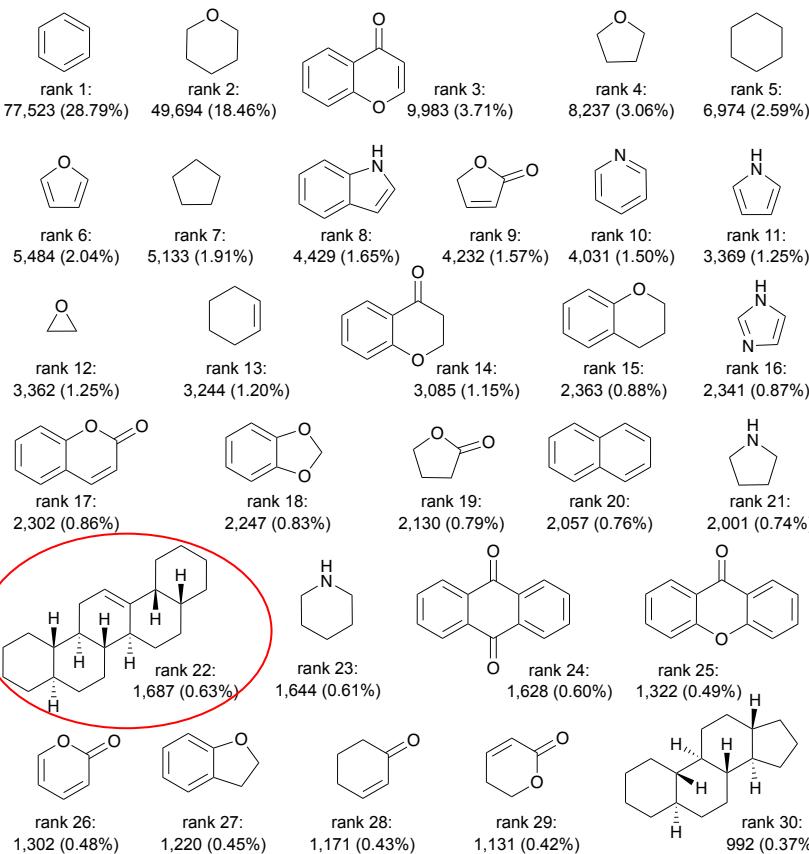
2. Y. Chen and J. Kirchmair, *Mol. Inf.*, 2020, 39, 2000171

2022/10/13 3. Y. Chen, C. Rosenkranz, S. Hirte and J. Kirchmair, *Nat. Pro. Rep.*, 2022, 39, 1544-1556

- Ring systems: All atoms forming a ring, plus any proximate exocyclic atom(s) connected via any type of bond other than a single bond

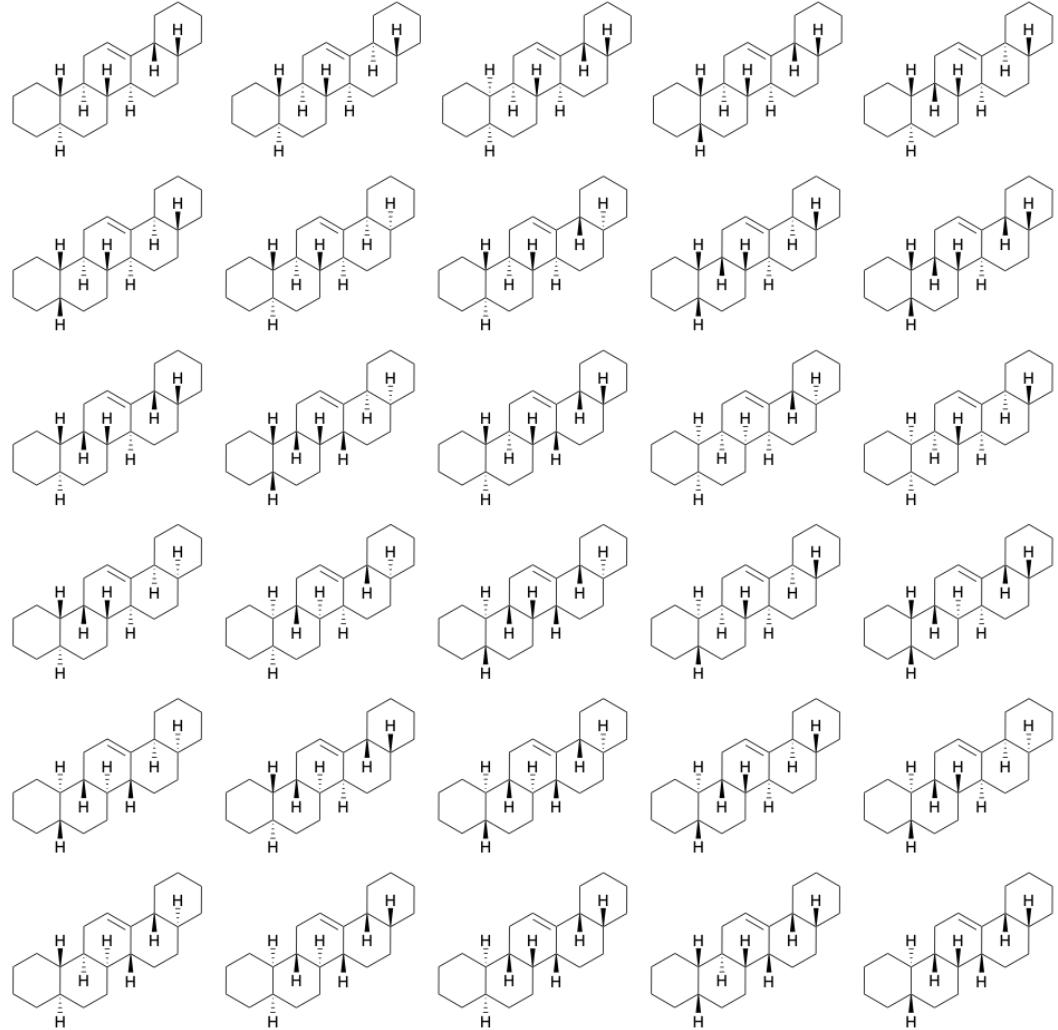
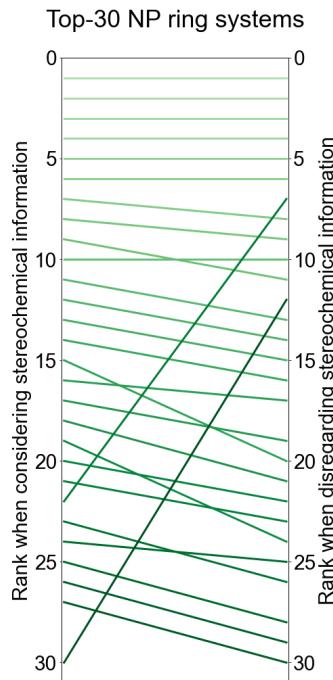


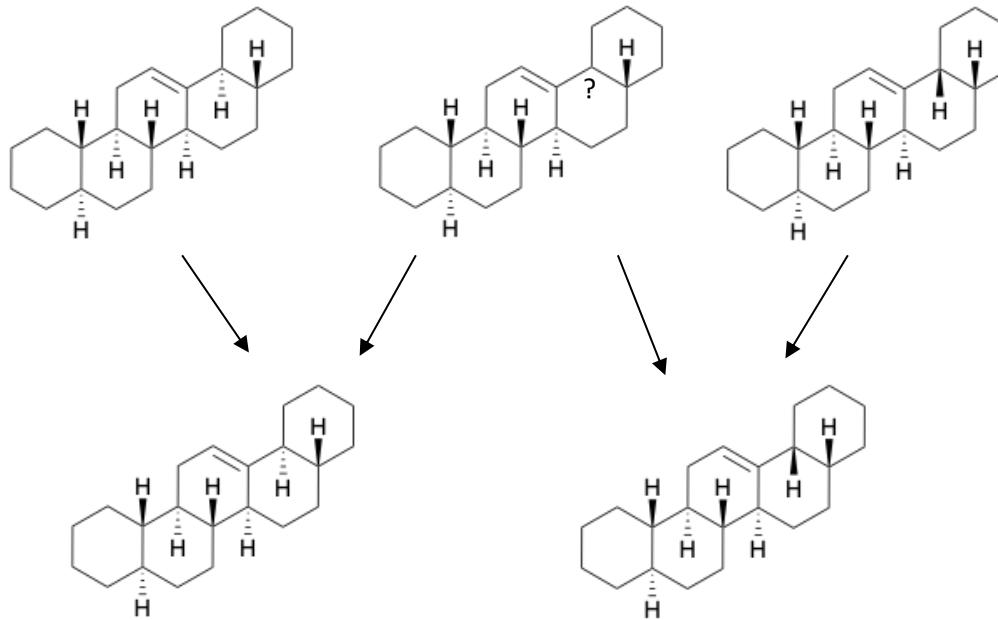
1. Identify individual rings (with the RDKit function *ringInfo*). Result in ring atom sets
2. If two ring atom sets share at least one atom the sets are fused
3. The processed ring atom sets are extended by all atoms directly connected to the ring via any type of bond other than a single bond
4. Cut bonds between ring systems and all other substituents are replaced by hydrogen atoms



the 30 most frequent ring systems in NPs

NPs with 105 recorded stereoisomers



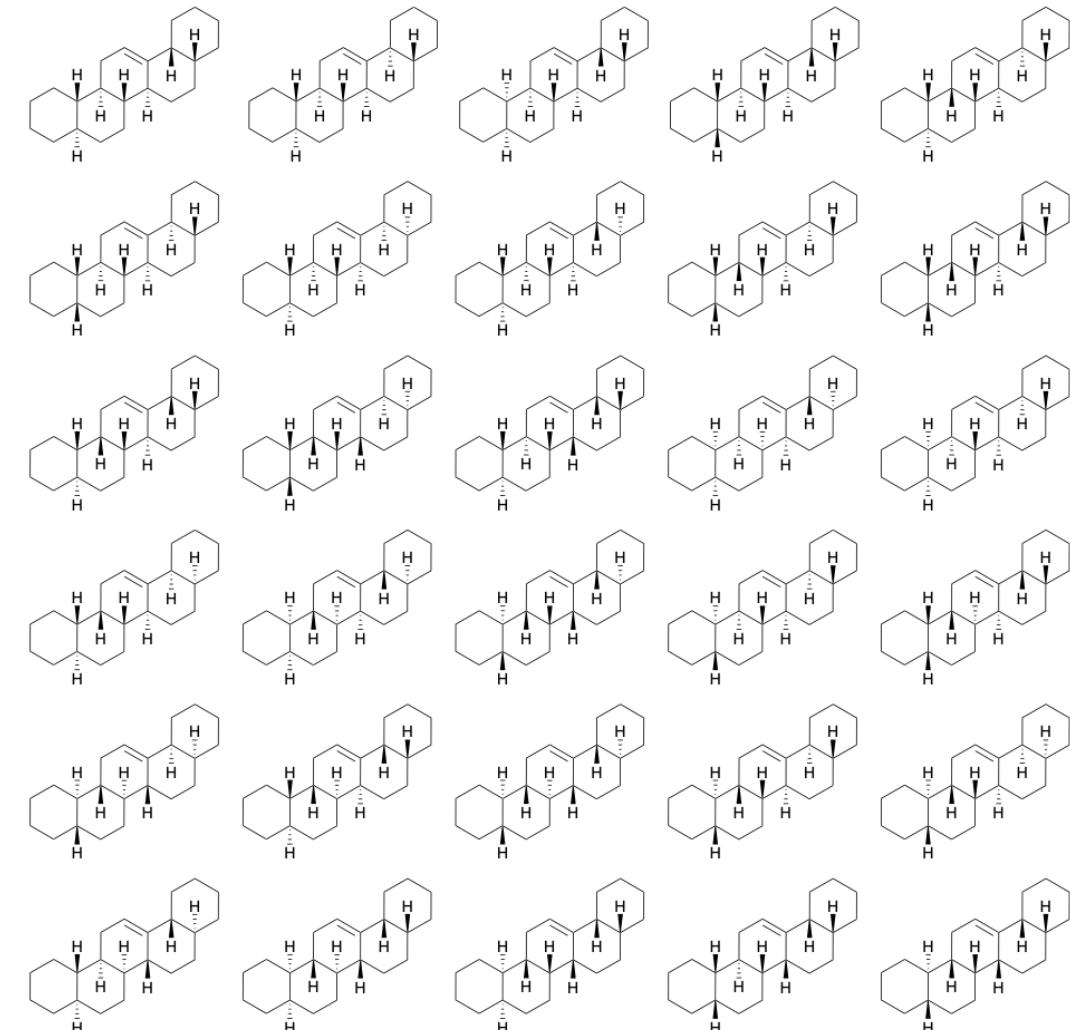


The RDKit 2022.03.1 documentation » Python API Reference » [rdkit package](#) » [rdkit.Chem package](#) »

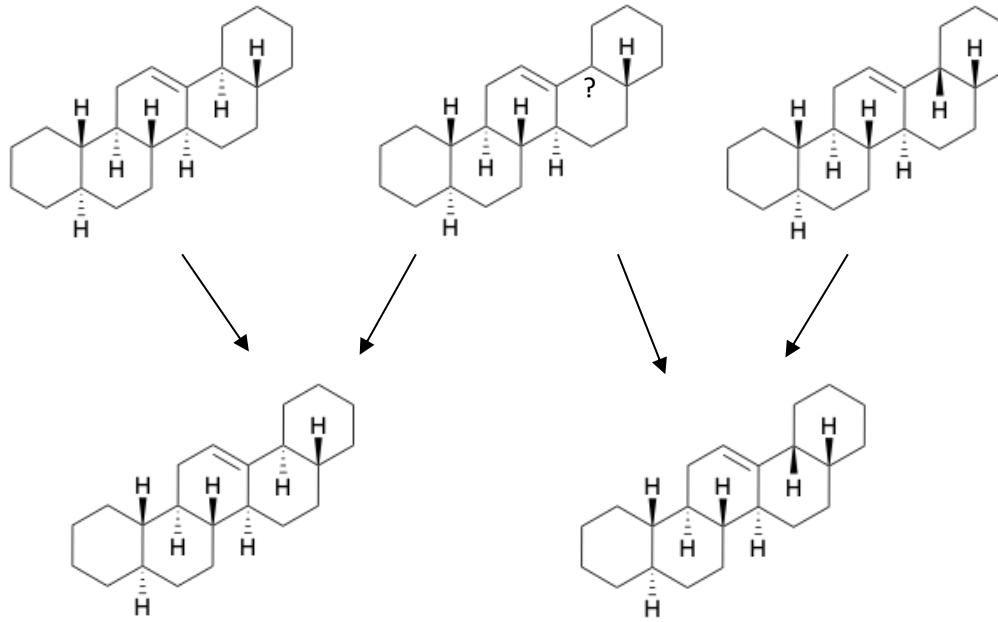
rdkit.Chem.EnumerateStereoisomers module

```
rdkit.Chem.EnumerateStereoisomers.EnumerateStereoisomers(m, options=  
<rdkit.Chem.EnumerateStereoisomers.StereoEnumerationOptions object>, verbose=False)
```

returns a generator that yields possible stereoisomers for a molecule



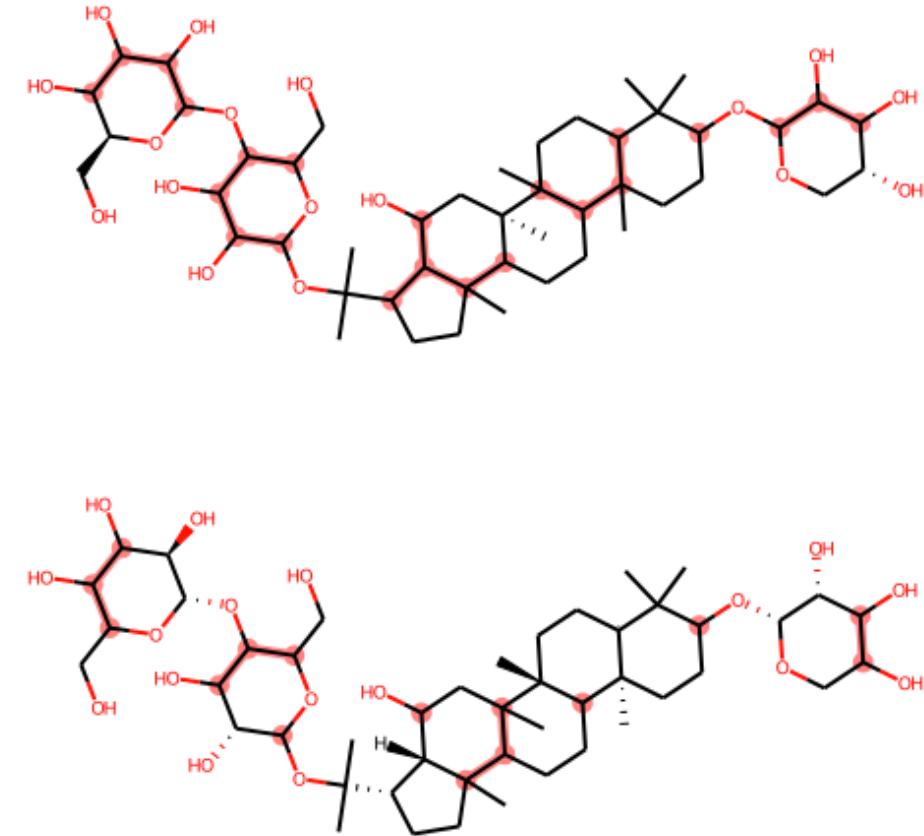
...



The RDKit 2022.03.1 documentation » Python API Reference » [rdkit package](#) » [rdkit.Chem package](#) »

rdkit.Chem.EnumerateStereoisomers module

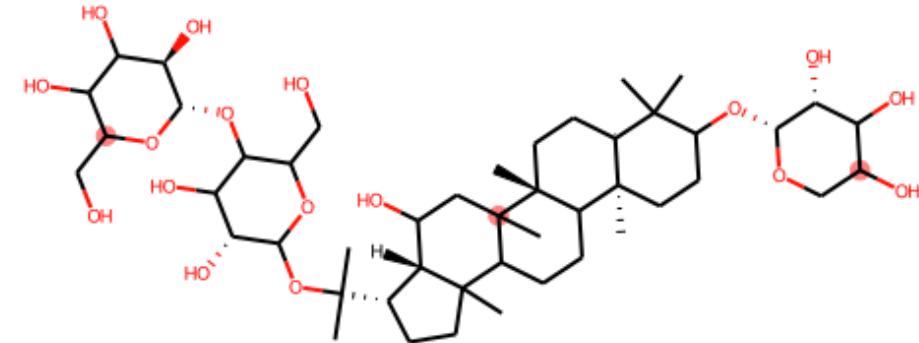
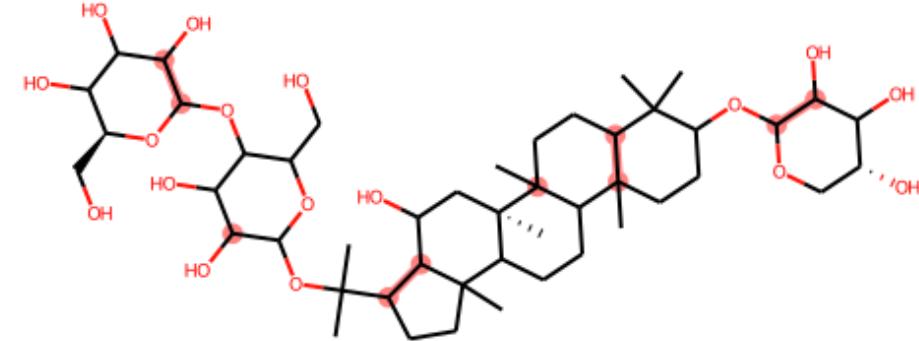
```
rdkit.Chem.EnumerateStereoisomers.EnumerateStereoisomers(m, options=  
    <rdkit.Chem.EnumerateStereoisomers.StereoEnumerationOptions object>,verbose=False)  
    returns a generator that yields possible stereoisomers for a molecule
```





Solution:

In a pair of molecules, enumerate the configurations only of atoms for which the configuration is defined in exactly one molecule



Solution:

In a pair of molecules, enumerate the configurations only of atoms for which the configuration is defined in exactly one molecule

```
def superpose(m1, m2):
    """
    this algorithm works well if both molecules m1 and m2 have a large number of unspecified stereo centers
    bond stereochemistry is not considered, so the input molecules should have all removed bond stereo
    ...
    # quick exit: are both molecules the same if we remove all stereo information?
    if MolToSmiles(m1, isomericSmiles=False) != MolToSmiles(m2, isomericSmiles=False):
        return False

    for match in m1.GetSubstructMatches(m2, uniquify=False, useChirality=False):
        m1_copy = Mol(m1)
        m2_copy = Mol(m2)

        # gather all atom pairs where only one atom has a specified stereo center
        relevant_atom_pairs = [(a1, a2) for a1, a2 in zip([m1_copy.GetAtomWithIdx(idx) for idx in match],
                                                       m2_copy.GetAtoms())
                               if (a1.GetChiralTag() == ChiralType.CHI_UNSPECIFIED or
                                   a2.GetChiralTag() == ChiralType.CHI_UNSPECIFIED) and
                                  a1.GetChiralTag() != a2.GetChiralTag()]

        # make sure that each tuple starts with the unspecified atom
        relevant_atom_pairs = [(a1, a2) if a1.GetChiralTag() == ChiralType.CHI_UNSPECIFIED
                               else (a2, a1)
                               for (a1, a2) in relevant_atom_pairs]

        # we want to enforce that m1 and m2 have the same CIP label on the stereo centers
        # unfortunately, we don't know which chiral tag (@, @@) corresponds to which CIP label
        # for that reason, we have to try all combinations
        for chiral_tag_combination in itertools.product((ChiralType.CHI_TETRAHEDRAL_CW,
                                                          ChiralType.CHI_TETRAHEDRAL_CCW),
                                                       repeat=len(relevant_atom_pairs)):

            # assign chiral tag combination to atoms
            for (a1, a2), flag in zip(relevant_atom_pairs, chiral_tag_combination):
                # we made sure that the first atom is unspecified
                a1.SetChiralTag(flag)

            if m1_copy.HasSubstructMatch(m2_copy, useChirality=True):
                return True
    return False
```

Acknowledgements

- Assoc.-Prof. Dr. Johannes Kirchmair
- Cara Rosenkranz
- Steffen Hirte
- Comp3D group



Check the publication:

Y. Chen, C. Rosenkranz, S. Hirte and J. Kirchmair, Nat. Pro. Rep., 2022, 39, 1544-1556

Check the code:

<https://github.com/anya-chen/RingSystems>

Thank you for your attention!

