Molecular Interaction Fields (MIFs) in RDKit

David F. Hahn,[a] Nikolas Fechner[b] and Gregory A. Landrum[b]

[a] *Laboratory of Physical Chemistry, Swiss Federal Institute of Technology ETH, 8093 Zurich, Switzerland*

[b] *Novartis Institute of BioMedical Research, 4056 Basel, Switzerland*

david.hahn@phys.chem.ethz.ch

Abstract

Molecular Interaction Fields (MIFs) were implemented into RDKit [1], a toolkit for cheminformatics developed in C++ with wrappers for Python. The new functionalities include a grid datatype to store and manipulate a molecule's interaction field descriptors and different types of MIF descriptor classes enabling the independent calculation of electrostatic, van der Waals, hydrogen bonding and hydrophilic interaction fields. The resulting fields can be visualized as isosurfaces in chemistry visualization programs and can be used for comparing molecules in three dimensional quantitative structure activity relationship (3D-QSAR) studies.

Introduction

Molecular Interaction Fields (MIFs) are used to describe the interactions of a molecule with its environment. The ability of a molecule to undergo favorable interactions with surrounding molecules are expressed as descriptors on gridpoints around the molecule. The environment is mimicked by probes which can be atoms, group of atoms or charges. There are different types of MIFs, defined describing different types of interactions, e.g. electrostatic (Coulomb) interactions, van der Waals interactions, hydrogen bonding interactions or hydrophilic/hydrophobic fields.

Theory

The interaction energy E_i between a probe and the molecule at the surrounding gridpoint i is calculated by summing up the pairwise atom-probe interactions. There are several MIFs defined, taking into account different types of non-covalent interactions:

- Electrostatic interactions using the Coulomb potential dependent on atom-probe distance $r_{j,probe}$, charges q and vacuum permittivity ϵ_0.

$$E_i^C = \frac{q_{probe}}{4\pi\epsilon_0} \sum_{j=0}^{N_{atoms}} \frac{q_j}{r_{j,probe}} \tag{1}$$

- Van der Waals interactions using a Lennard-Jones potential with parameters $\epsilon_{j,probe}$ and $\sigma_{j,probe}$.

$$E_i^{vdW} = \sum_{j=0}^{N_{atoms}} 4\epsilon_{j,probe} \left[\left(\frac{\sigma_{j,probe}}{r_{j,probe}} \right)^{12} - \left(\frac{\sigma_{j,probe}}{r_{j,probe}} \right)^{6} \right] \tag{2}$$

- Hydrogen Bond Interaction

$$E_i^{HB} = E_{i,r} E_{i,t} E_{i,p} \tag{3}$$

  with $E_{i,r}$ being a distance-dependent Lennard-Jones Potential and $E_{i,t}$ and $E_{i,p}$ being angle-dependent terms taking into account the directionality of a hydrogen bond.

Implementation

- `UniformRealValueGrid3D` for storing and manipulating fields
- Several descriptor functors, which are constructed for specific molecules with various optional arguments (e.g. probe type, softcore parameters, dielectric constants, cutoff radius):
  - Coulomb MIF using either Coulomb potential or a potential scaled by a dielectric as in GRID descriptors. [2]
  - Van der Waals MIF using parameters and functional forms: UFF [3] or MMFF94 [4] force fields, which are already available in RDKit.
  - Hydrogen bonding MIF using parameters and functional forms adapted from GRID descriptors. [5]
- Function `calculateDescriptors(UniformRealValueGrid3D, Functor)` to efficently calculate MIF on the given `UniformRealValueGrid3D`
- Python wrappers to C++ MIF library

Example code

In the following example (IPython [6] notebook) a MIF of the angiotensin-converting enzyme (ACE) inhibitor enalapril [7] is calculated and visualized as an isosurface in PyMol [8] (Figure 1).

```
In [1]: import rdkit.Chem.AllChem as AllChem
        from rdkit.Chem.Draw import IPythonConsole
        import rdkit.Chem.rdMIF as Mif
```

Load structure from SDF file to mol.

```
In [2]: mol = AllChem.SDMolSupplier('mif_rdkit_tutorial/enalapril.sdf',   removeHs=False)
        mol = mol.next()
```

Construct grid around molecule (using default values for spacing and margin).

```
In [3]: grd = Mif.ConstructGrid(mol)
```

Construct a van der Waals descriptor class, calculate descriptor values on the grid and save it to a CUBE file.

```
In [4]: vdw = Mif.ConstructVdWaalsUFF(mol)
        Mif.CalculateDescriptors(grd, vdw)
        Mif.WriteToCubeFile(grd, mol, "mif_rdkit_tutorial/vdw.cube")
```

Visualize grid as an isosurface in PyMol. PyMol has to be running with the XML-RPC server active (-R argument to PyMol on launch).

```
In [5]: from rdkit.Chem import PyMol
        v = PyMol.MolViewer()
        v.ShowMol(mol)
        v.LoadFile("~/notebooks/mif_rdkit_tutorial/vdw.cube", "vdw")
        v.server.do("isosurface surf, vdw, 0")
        v.server.do('set transparency, 0.4, surf')
        v.GetPNG(h=500)
```

Results

Van der Waals Field (Figure 1)

- Parameters and functional forms of Universal Force Field (UFF)
- Probe O.3, sp^2 oxygen (eg. in alcohol or ether)
- Isosurface at 0.0 kJ mol^{-1}

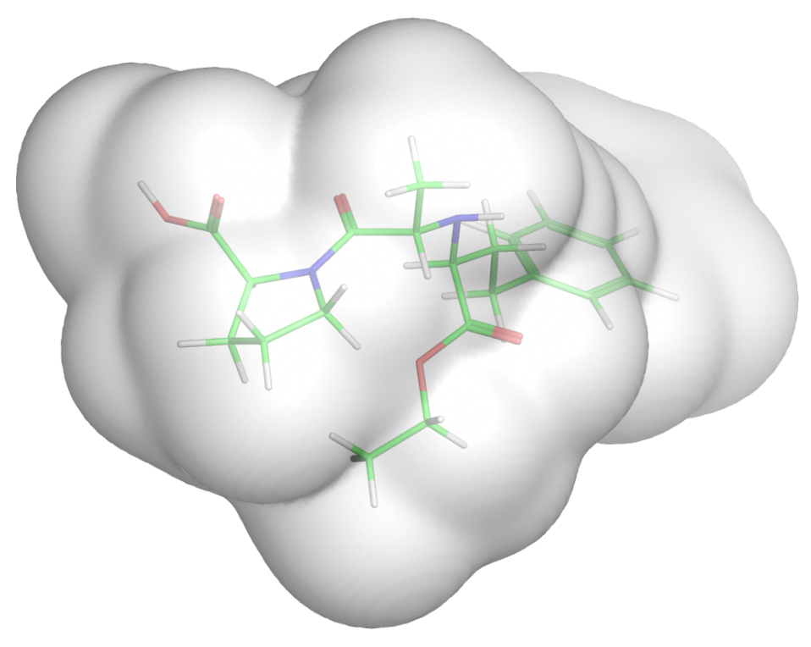

Figure 1: Isosurface of a van der Waals MIF.

Coulomb Field (Figure 2)

- Coulomb potential in vacuum.
- Positive Charge (+1e)
- Isosurface as in Figure 1
- Colored by a Coulomb MIF colormap ranging from -25.0 kJ mol^{-1} (blue) via 0.0 kJ mol^{-1} (white) to $+25.0$ kJ mol^{-1} (red)

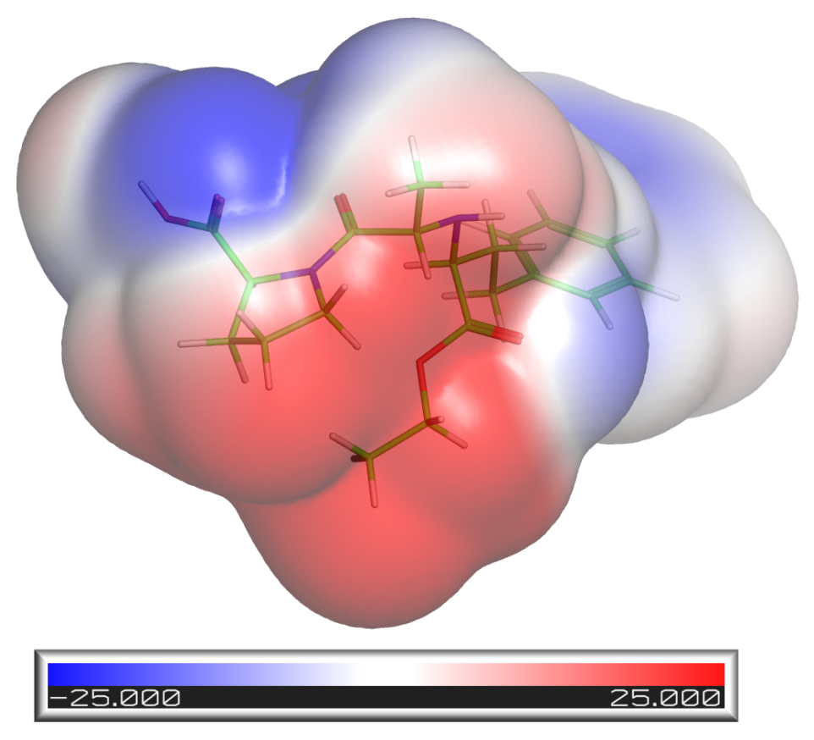

Figure 2: Isosurface of van der Waals MIF colored by colormap of a Coulomb field.

Hydrogen Bonding Field (Figure 3)

- Lennard-Jones potential scaled by additional angular terms.
- Red: Isomesh of a hydrogen bond MIF using an acceptor probe (C=O) at level -25.0 kJ mol^{-1}.
- Blue: Isomesh of a hydrogen bond MIF using a donor probe (OH) at level -20.0 kJ mol^{-1}.

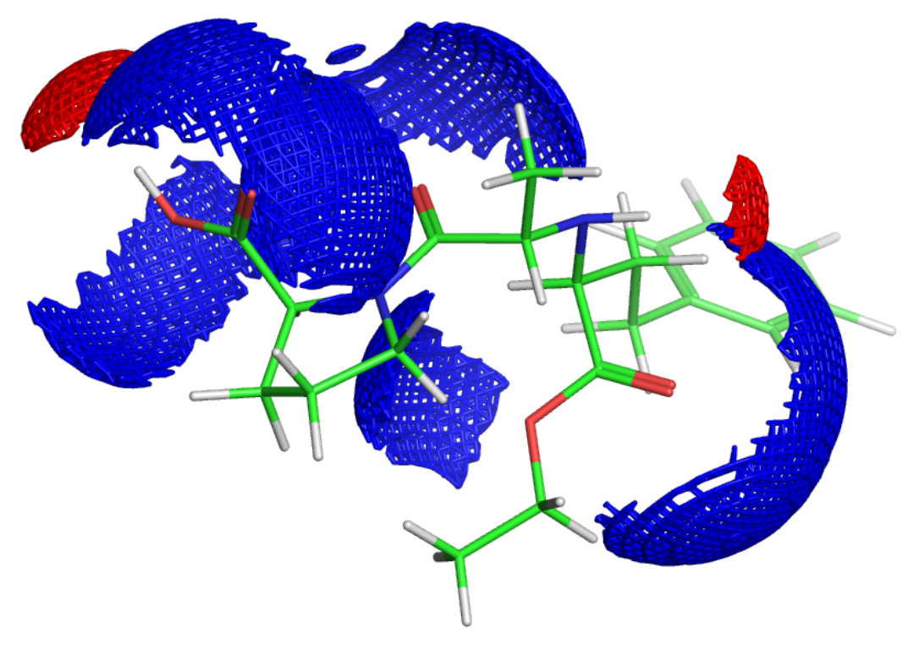

Figure 3: Hydrogen bonding MIFs using acceptor (red) and donor (blue) probe.

Conclusions

The newly developed RDKit functionalities

- can compute van der Waals, Coulomb and Hydrogen bonding MIFs of different molecules.
- can be used in conjunction with all available force fields in RDKit.
- can be used in Python through the newly implemented wrappers to the MIF library.
- open new possibilities for 3D-QSAR studies in RDKit.

References

[1] RDKit: Open-source cheminformatics; http://www.rdkit.org.

[2] P. J. Goodford, *J. Med. Chem.* **1985**, *28*, 849.

[3] A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard III, W. M. Skid, *J. Am. Chem. Soc.* **1992**, *114*, 100024.

[4] T. A. Halgren, *J. Comp. Chem.* **1996**, *17*, 520.

[5] R. C. Wade, K. J. Clark, P. J. Goodford, *J. Med. Chem.* **1993**, *36*, 140. R. C. Wade, Peter J. Goodford, *J. Med. Chem.* **1993**, *36*, 148.

[6] F. Pérez, B. E. Granger, *Comput. Sci. Eng.* **2007**, *9*, 21, *http://ipython.org.*

[7] National Center for Biotechnology Information. PubChem Compound Database; CID=5388962, *https://pubchem.ncbi.nlm.nih.gov/compound/5388962* (accessed Aug. 24, 2015).

[8] The PyMOL Molecular Graphics System, Version 1.7.0 Schrdinger, LLC.

Acknowledgements

NOVARTIS
INSTITUTES FOR
BIOMEDICAL RESEARCH