Working with Conformers from Molecular Dynamics Simulations in the RDKit

Sereina Riniker

3rd RDKit UGM, Darmstadt

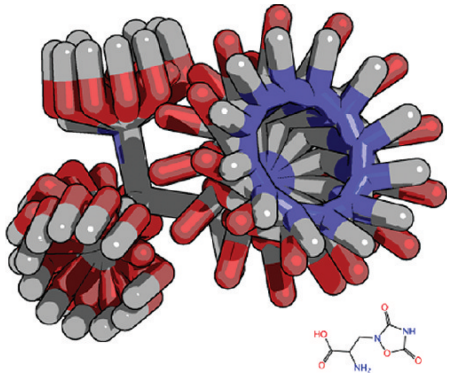
October 22, 2014

Outline

- Motivation
- Introduction to molecular dynamics
- Molecular dynamics trajectories
- ConformerParser in the RDKit
- Examples

Motivation

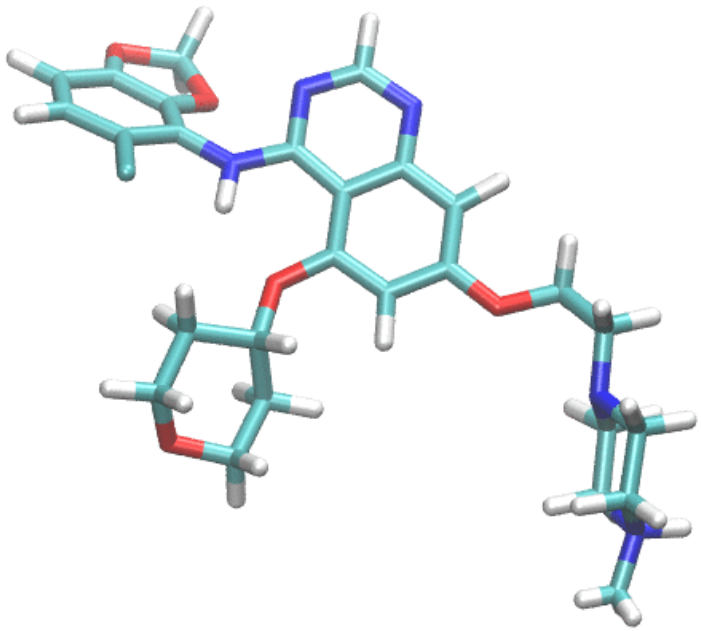
- Conformer generation
  - Systematic: Exhaustive sampling
    - Incrementing torsional angles of all rotatable bonds
  - Stochastic: Random sampling
    - Distance geometry
    - Monte Carlo simulated annealing
    - Genetic algorithms
    - **Molecular dynamics simulation**
  - Knowledge-based methods:
    - Predefined libraries of experimental torsional preferences
    - Systematic or stochastic sampling



J. P. Ebejer *et al.*, *J. Chem. Inf. Model.*, **52**, 1146-1158 (2012).

Motivation

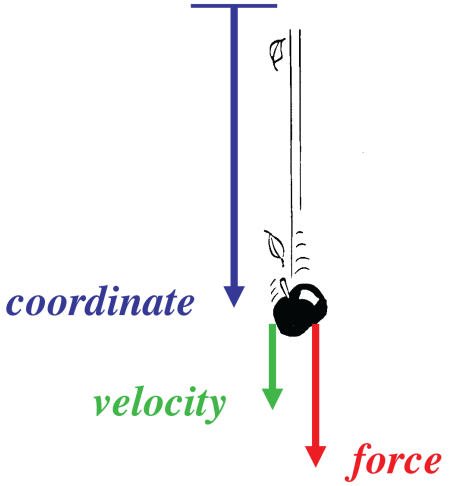
- Molecular dynamics simulation
  - Advantages:
    - Boltzmann weighted ensemble
    - Effect of solvent can be considered (explicit solvent)
    - Effect of binding pocket can be considered (protein-ligand complex)
  - Disadvantages:
    - Computationally very expensive (= slow)
    - Starting structure needed



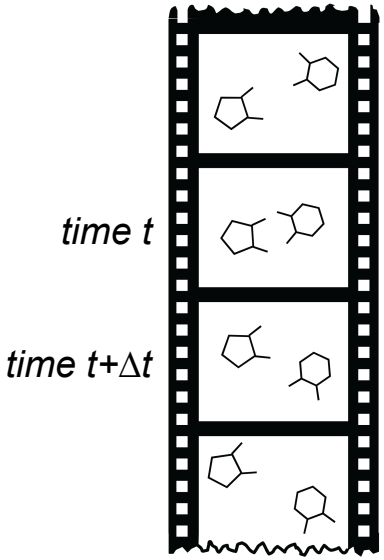
Molecular Dynamics

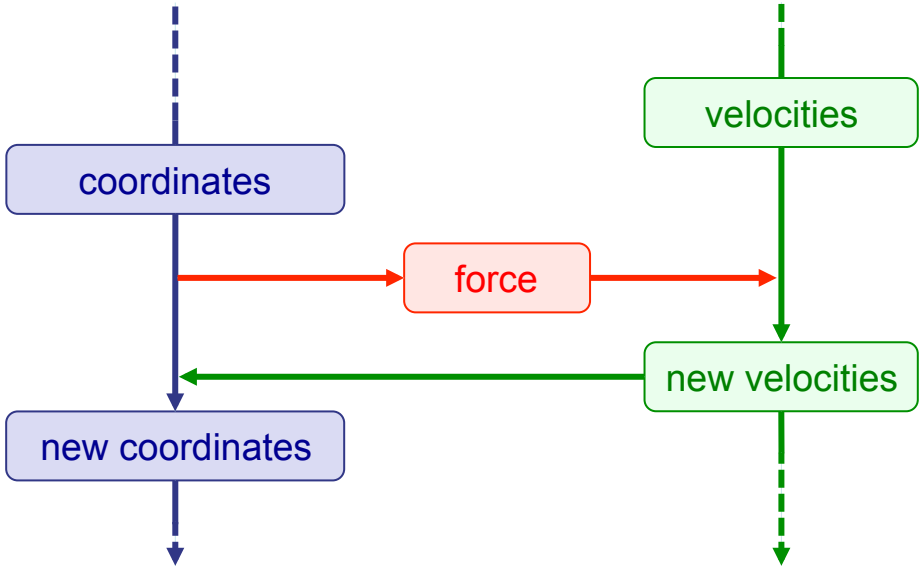
- Classical MD simulations:
  - Atoms = point charges (no electrons)
  - Numerical solution of Newton's equations of motion

$$\mathbf{f}_i(t) = m_i \mathbf{a}_i(t) = -\frac{\partial V(\mathbf{r}^N(t))}{\partial \mathbf{r}_i(t)}$$



  - Generating a "movie" of a molecular system by integrating the equations of motion





MD Trajectories

- Trajectory:
  - Snapshots (coordinates) of the system at regular time intervals
- Problem:
  - Each MD program has its own file format
- Most common MD programs:
  - AMBER trajectory format: trx (ASCII)
  - CHARMM trajectory format: dcd (binary) or crd (ACSII)
  - NAMD / LAMMPS: like CHARMM
  - GROMOS trajectory format: trc or g96 (ACSII)
  - GROMACS trajectory format: xtc (binary) or GROMOS format
  - OpenMM: PDB format

MD Trajectories

- Example: AMBER trajectory (trx)
  - X, Y, Z coordinates of all atoms (10 values in each row)
  - Units: Ångström
  - Snapshots not separated (except line break)

```
Cpptraj Generated trajectory
  37.364  29.875  35.099  46.553  20.451  33.831  47.349  21.693  33.980  46.760
  22.652  32.964  47.122  24.168  33.153  47.014  24.658  34.592  47.273  26.114
  34.740  46.448  27.048  33.795  45.084  26.678  33.984  44.045  27.562  33.627
  42.728  27.129  33.681  41.654  27.975  33.370  40.394  27.384  33.379  39.956
  26.168  34.058  38.472  26.435  34.431  37.793  25.118  34.945  38.050  24.079
  34.004  39.497  23.731  33.899  40.084  24.928  33.161  41.919  29.310  33.012
  40.901  30.233  32.740  39.572  30.061  32.977  38.415  30.727  32.647  38.132
  31.233  31.366  38.974  31.249  30.325  38.348  32.139  29.381  36.980  32.285
  29.733  36.968  31.814  31.001  35.866  31.731  31.839  35.985  31.132  33.075
  37.285  30.609  33.482  41.113  31.531  32.450  42.384  31.782  32.310  43.432
  31.027  32.522  43.200  29.737  32.860  44.262  28.858  33.160  47.769  23.710
  35.482  47.306  22.217  35.470  45.544  20.609  34.215  46.421  20.222  32.772
  47.183  19.713  34.329  48.338  21.523  33.735  47.183  22.332  32.010  45.678
  22.593  33.088  48.144  24.326  32.800  46.425  24.703  32.503  47.308  26.449
  35.780  48.274  26.277  34.333  46.596  28.061  34.178  46.675  27.047  32.726
  42.615  26.061  33.846  40.632  25.962  34.892  38.420  27.202  35.207  37.974
  26.889  33.571  38.316  24.886  35.876  36.726  25.180  35.173  39.966  23.548
  34.869  39.475  22.823  33.289  39.569  25.063  32.208  41.146  24.750  32.981
  39.377  29.171  33.420  38.777  33.129  29.166  38.442  31.641  28.405  34.936
  32.197  31.522  35.090  30.980  33.673  42.599  32.837  32.168  45.221  29.359
  33.051  48.841  23.749  35.273  47.626  24.106  36.490  46.247  22.201  35.733
  47.857  21.583  36.166
  37.250  30.226  34.512  47.674  21.037  33.052  47.902  22.472  33.501  47.228
  23.544  32.620  47.371  25.019  33.058  46.846  25.203  34.493  47.135  26.571
  35.007  46.421  27.571  34.151  44.995  27.240  34.059  43.980  28.077  33.637
  42.742  27.535  33.520  41.670  28.287  33.266  40.488  27.737  33.148  39.937
```

MD Trajectories

- Example: GROMOS trajectory (trc)
  - X, Y, Z coordinates of all atoms (one atom per row)
  - Units: nm (→ will be converted to Ångström)
  - Block structure (NAME – content – END):
    - Title block (only once), each snapshot has timestep/positions/(velocities)/ box-info blocks

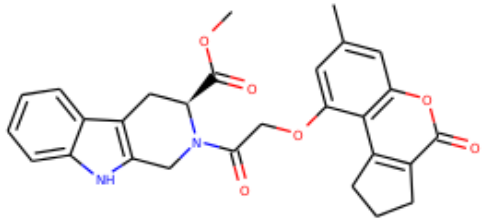
```
TITLE
FRAME_00001.trc
END
TIMESTEP
               0         0.000000000
END
POSITIONRED
   -0.009221286    0.430625522    0.177084634
    0.087544495    0.427158037    0.229267340
    0.074678072    0.415003918    0.337824947
    0.147502634    0.516050023    0.204745642
    0.155019242    0.308074406    0.187852728
    0.180409394    0.287032527    0.055899114
    0.170838628    0.373960055   -0.030603882
    0.232042261    0.144556661    0.034769232
    0.241829699    0.100464986    0.135069435
    0.152371394    0.047717833   -0.041953505
#        10
    0.374569942    0.157989010   -0.019217432
    0.370268275    0.202619708   -0.119668100
    0.428453486    0.228275079    0.046029355
    0.221164555   -0.079439594   -0.068629918
    0.182887072   -0.150328161    0.006279667
    0.192391097   -0.116143318   -0.168264258
    0.372763191   -0.083896511   -0.058422836
```

ConformerParser in RDKit

- Idea:
  - Read snapshots from MD trajectories in as conformers of a molecule
  - Get access to cheminformatics methods in RDKit
- Requirement:
  - RDKit-molecule with the same order of the atoms (!)
- Checks performed during reading:
  - Correct number of atoms
- Currently supported formats:
  - AMBER
  - GROMOS (GROMACS)
- Location:
  - in rdkit/Contrib/ConformerParser
  - Flag in rdkit/CMakeLists.txt: `RDK_BUILD_CONTRIB = ON` (default: OFF)
  - Load module: `from rdkit.Chem import rdConformerParser`

Examples

- MD simulation of a ligand in explicit water with GROMOS



➢ Some simulation details:
  - Ligand (62 atoms, uncharged) in 1196 SPC water molecules
  - Temperature: 298 K
  - Length: 200 ps, coordinates written every 2 ps → 100 snapshots

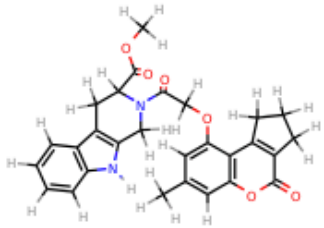
➢ Preparation of trajectory:
  - Extraction of the ligand coordinates using a GROMOS helper program
  - Initial ligand structure converted to PDB → for reference molecule

Examples

- Reference molecule from PDB:

In [6]:
```
# load pdb
ref1 = Chem.MolFromSmiles('Cc1cc2c(c(c1)OCC(=O)N3Cc4c(c5ccccc5[nH]4)C[C@H]3C(=O)OC)c6c(c(=O)o2)CCC6')
ref1 = AllChem.AddHs(ref1, addCoords=True)
mol1 = Chem.MolFromPDBFile('ligand_zinc_start.pdb', removeHs=False)
mol1 = AllChem.AssignBondOrdersFromTemplate(ref1, mol1)
mol1.RemoveAllConformers()
mol1
```

Out[6]:



- Read MD conformers:

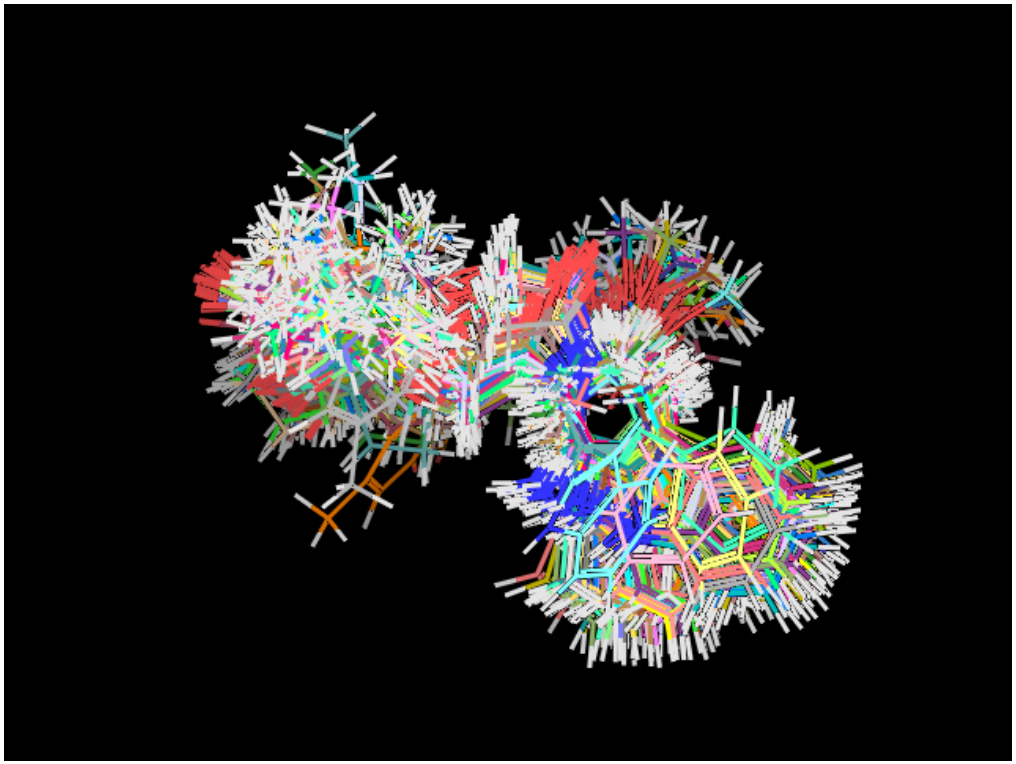
In [7]:
```
from rdkit.Chem import rdConformerParser
cids1 = rdConformerParser.AddConformersFromGromosTrajectory(mol1, 'ligand_zinc_gath.trc')
print len(cids1)
```

```
100
```

- Use OPEN3DAlign to align the conformers (first as reference):

In [8]:
```
from rdkit.Chem import rdMolAlign
for i in range(1, mol1.GetNumConformers()):
    pyO3A = rdMolAlign.GetO3A(mol1, mol1, prbCid=i, refCid=0)
    pyO3A.Align()
```

Examples



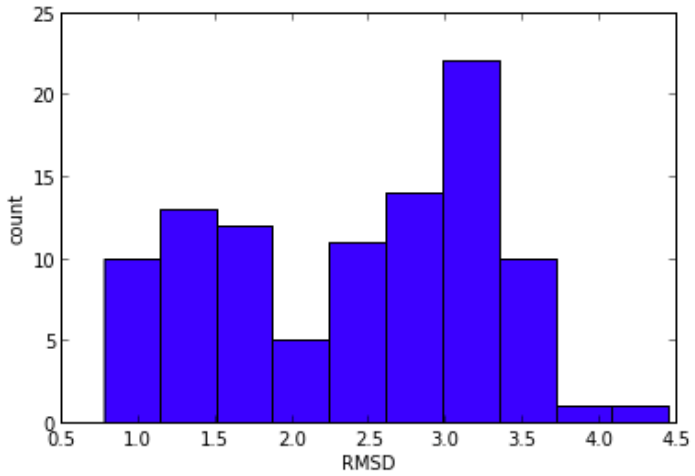
- What can we do with the conformers?
  - Diversity picking or clustering
  - Based on RMSD, Shape-Tanimoto distance, 3D-pharmacophore fingerprint similarity

Examples

- RMSD (first conformer as reference)

In [9]:
```
rmsdl = []
for cid in cids1[1:]:
  rmsdl.append(AllChem.GetConformerRMS(mol1, 0, cid, prealigned=True))
```

In [10]:
```
plt.xlabel('RMSD')
plt.ylabel('count')
hist(rmsdl)
plt.show()
```



- RMSD matrix

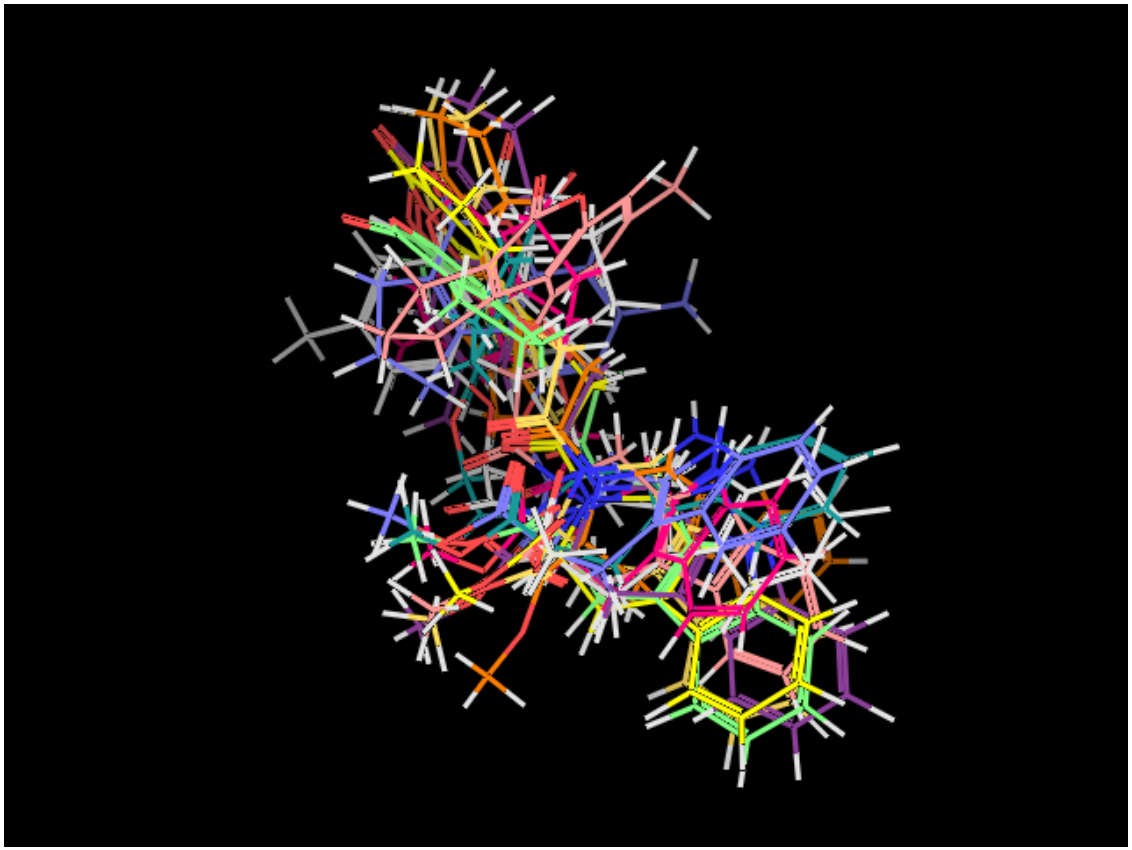
In [11]:
```
rmsdmat1 = AllChem.GetConformerRMSMatrix(mol1, prealigned=True)
```

Examples

- Pick 10 most diverse conformers:

```
In [16]: from rdkit import SimDivFilters
         mmp = SimDivFilters.MaxMinPicker()
         dids1 = mmp.Pick(numpy.array(rmsdmat1),mol1.GetNumConformers(),10)
         print [d for d in dids1]

[37, 75, 61, 81, 10, 44, 84, 59, 18, 3]
```



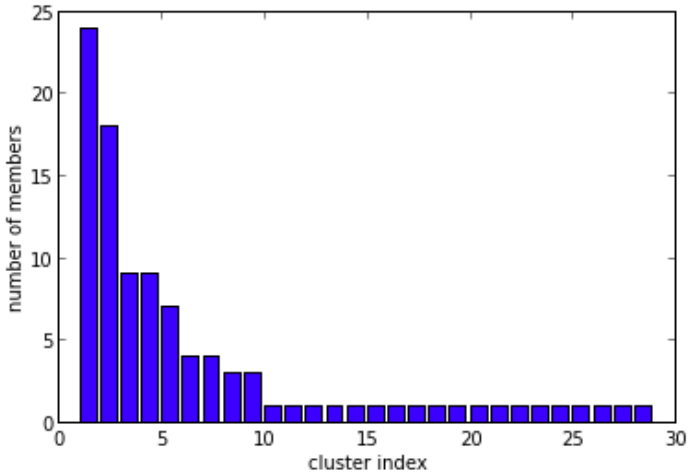
Examples

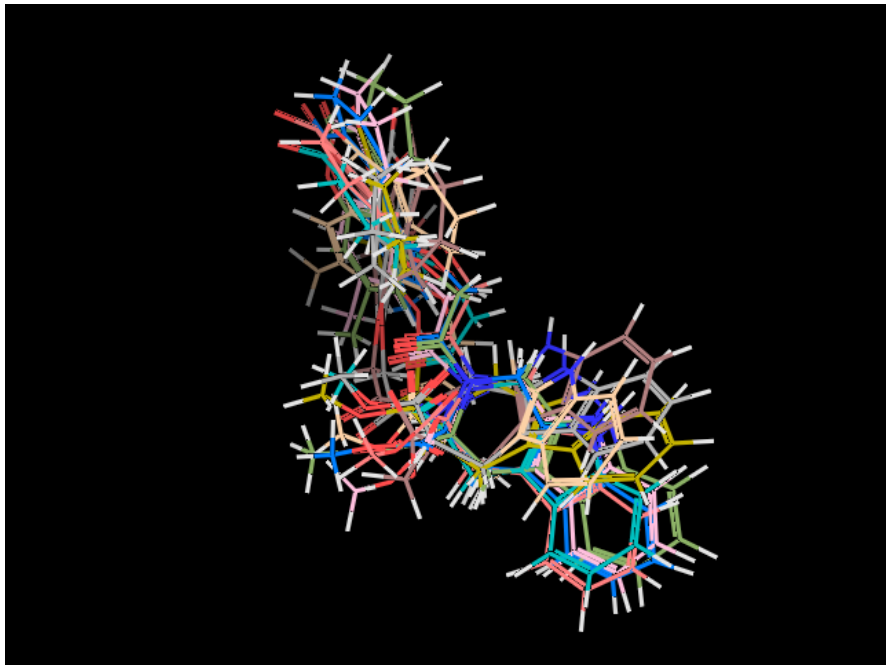
- RMSD-based clustering (with reordering):

```
In [20]: from rdkit.ML.Cluster import Butina
         clusters1 = Butina.ClusterData(rmsdmat1, mol1.GetNumConformers(), 1.0, isDistData=True, reordering=True)
         print len(clusters1)

         28

In [21]: plt.xlabel('cluster index')
         plt.ylabel('number of members')
         bar(range(1,len(clusters1)+1), [len(c) for c in clusters1])
         plt.show()
```





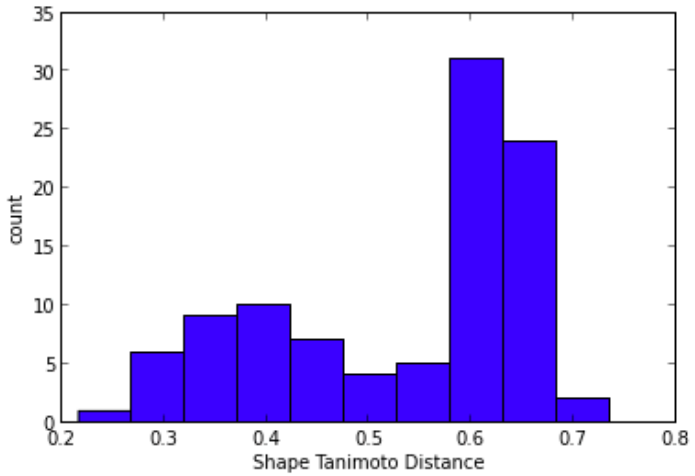
Cluster centers of clusters with more than one member

Examples

- Shape Tanimoto distance (first conformer as reference):

In [63]:
```
tani1 = []
for cid in cids1[1:]:
  tani1.append(AllChem.ShapeTanimotoDist(mol1, mol1, confId1=0, confId2=cid))
```

In [68]:
```
plt.xlabel('Shape Tanimoto Distance')
plt.ylabel('count')
hist(tani1)
plt.show()
```



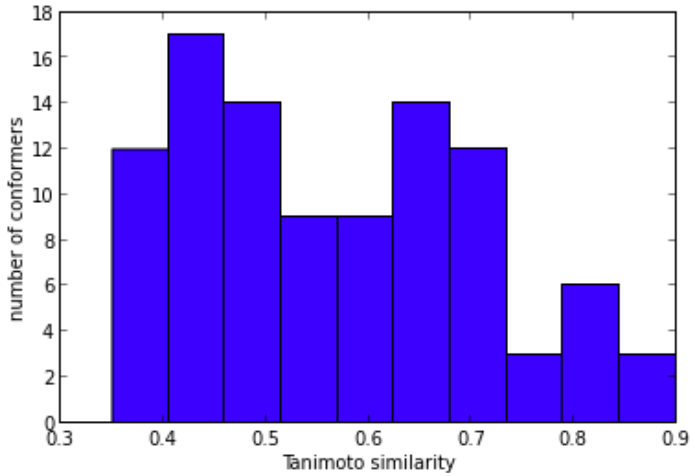
Examples

- 3D-pharmacophore fingerprint similarity (first conformer as reference):

In [83]:
```
from rdkit.Chem.Pharm2D import Gobbi_Pharm2D, Generate
factory = Gobbi_Pharm2D.factory
pharm3D = []
for cid in cids1:
    dm = Chem.Get3DDistanceMatrix(mol1, confId=cid)
    pharm3D.append(Generate.Gen2DFingerprint(mol1, factory, dMat=dm))
```

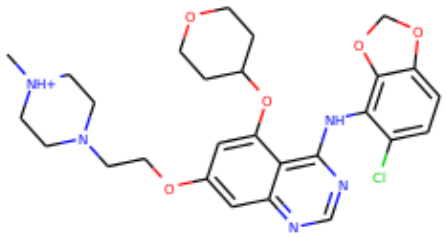
In [84]:
```
from rdkit import DataStructs
# similarity distribution
sims1 = []
p1 = pharm3D[0] # first conformer as reference
for p in pharm3D[1:]:
    sims1.append(DataStructs.TanimotoSimilarity(p1, p))
```

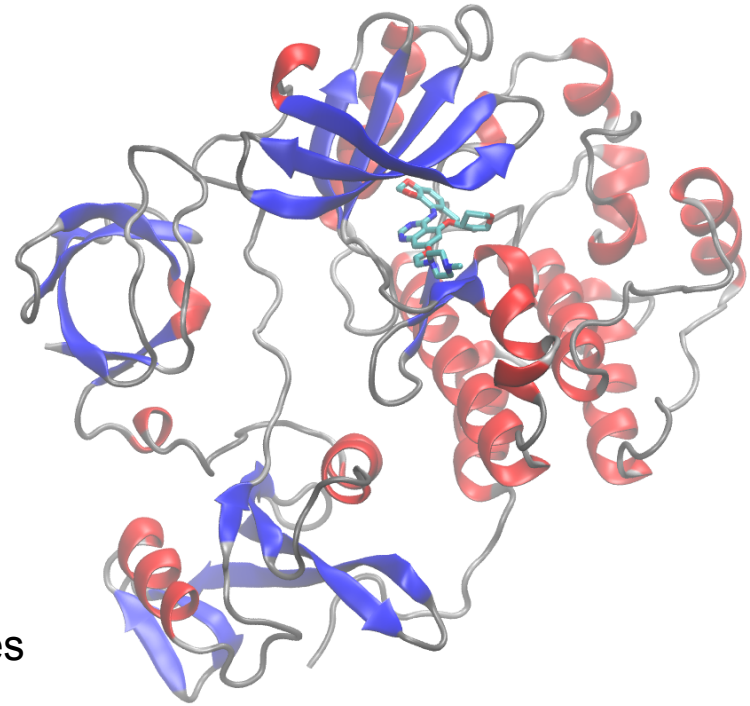
In [85]:
```
plt.xlabel('Tanimoto similarity')
plt.ylabel('number of conformers')
hist(sims1)
plt.show()
```



Examples

- MD simulation of a ligand in SRC kinase in explicit water with AMBER





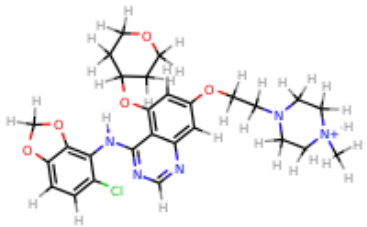
  - Some simulation details:
    - Starting structure: PDB 2H8H
    - Ligand (71 atoms, charge +1) with protein in 16129 TIP3P water molecules
    - Temperature: 298 K
    - Length: 500 ps, coordinates written every 5 ps → 100 snapshots
  - Preparation of trajectory:
    - Extraction of the ligand coordinates using an AMBER helper program
    - Initial ligand structure converted to sdf → for reference molecule

Examples

- Reference molecu

In [10]:
```
# PDB 2H8H
mol2 = Chem.MolFromMolFile('lig_2H8H.sdf', removeHs=False)
mol2.RemoveAllConformers()
mol2
```

Out[10]:



- Read MD conformers and align them with OPEN3DAlign:

In [12]:
```
cids2 = rdConformerParser.AddConformersFromAmberTrajectory(mol2, 'ligand.trx')
print len(cids2)
```

```
100
```

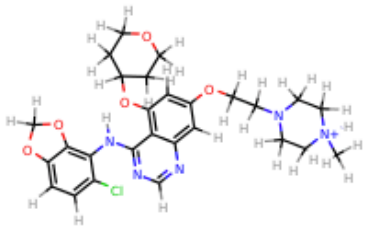
In [13]:
```
for i in range(1, mol2.GetNumConformers()):
    pyO3A = rdMolAlign.GetO3A(mol2, mol2, prbCid=i, refCid=0)
    pyO3A.Align()
```

Examples

- Reference molecu

```
In [10]: # PDB 2H8H
         mol2 = Chem.MolFromMolFile('lig_2H8H.sdf', removeHs=False)
         mol2.RemoveAllConformers()
         mol2
```

Out[10]:

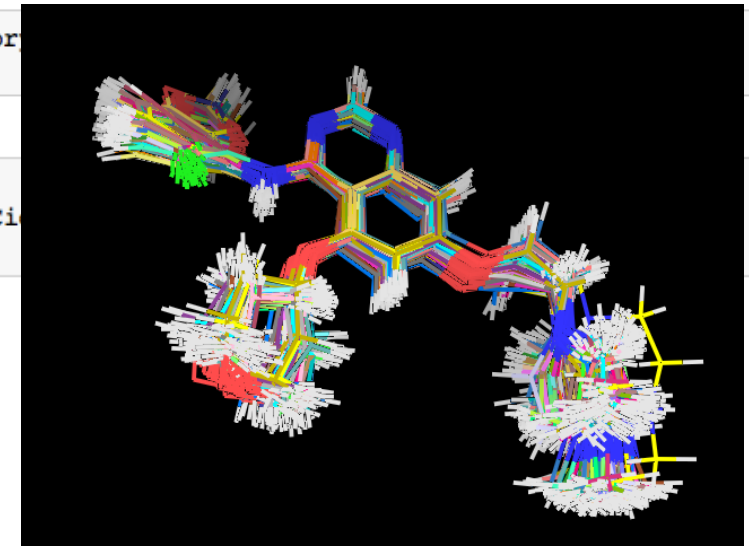


- Read MD conformers and align them with OPEN3DAlign:

```
In [12]: cids2 = rdConformerParser.AddConformersFromAmberTrajector
         print len(cids2)
```

```
100
```

```
In [13]: for i in range(1, mol2.GetNumConformers()):
             pyO3A = rdMolAlign.GetO3A(mol2, mol2, prbCid=i, refCi
             pyO3A.Align()
```

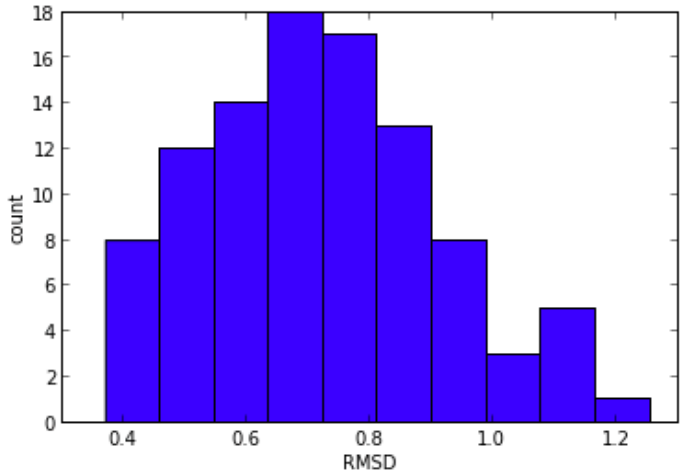


Examples

- RMSD (first conformer as r

In [35]:
```
rmsd2 = []
for cid in cids2[1:]:
  rmsd2.append(AllChem.GetConformerRMS(mol2, 0, cid, prealigned=True))
```

In [36]:
```
plt.xlabel('RMSD')
plt.ylabel('count')
hist(rmsd2)
plt.show()
```



- RMSD matrix

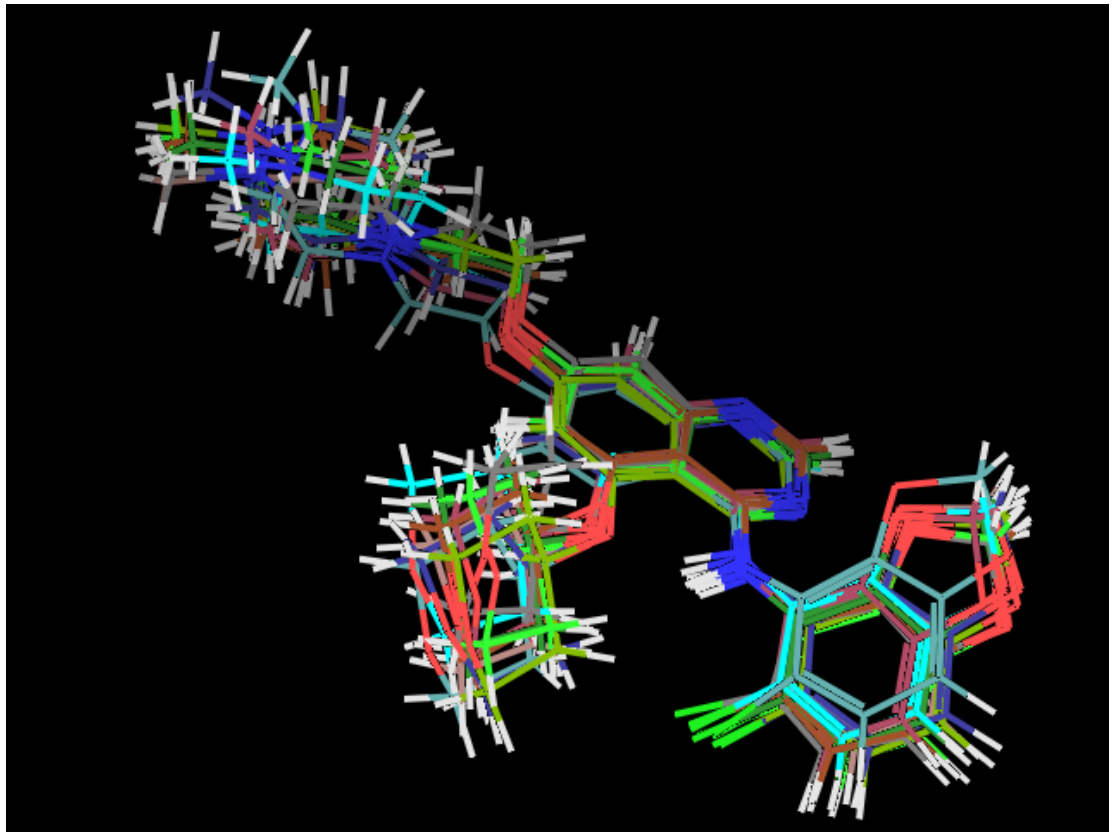
In [44]:
```
rmsdmat2 = AllChem.GetConformerRMSMatrix(mol2, prealigned=True)
```

Examples

- Pick 10 most diverse conformers:

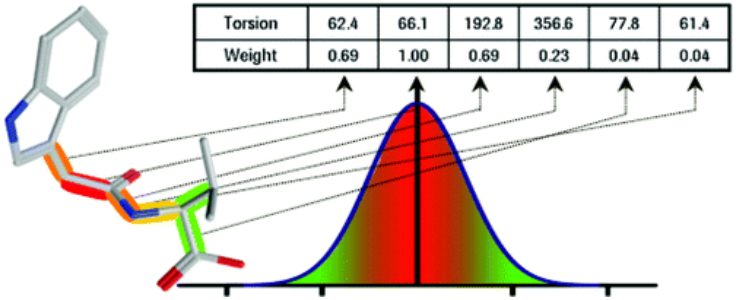
```
In [45]: mmp = SimDivFilters.MaxMinPicker()
         dids2 = mmp.Pick(numpy.array(rmsdmat2), mol2.GetNumConformers(), 10)
         print [d for d in dids2]

[37, 23, 11, 62, 28, 77, 61, 49, 44, 33]
```



Outlook

- Extension to other file formats:
  - CHARMM Cartesian coordinate format (crd)
  - Series of PDB (OpenMM)
  - Others?
- Implementation of Torsion fingerprints
  - T. Schulz-Gasch *et al.*, *J. Chem. Inf. Model.*, **52**, 1499-1512 (2012)



  - Item for the Hackathon on October 24
- Other 3D fingerprints/methods?

Acknowledgements

- Greg Landrum (Novartis)
- Romain Wolf (Novartis)

- Philippe Hünenberger (ETH Zürich)

Acknowledgements

- Greg Landrum (Novartis)
- Romain Wolf (Novartis)

- Philippe Hünenberger (ETH Zürich)