chem-bla-ics

/me is having Bioclipse/XMPP/RDF fun

Egon Willighagen

Published May 7, 2009

Citation
Willighagen, E. (2009). /me is having Bioclipse/XMPP/RDF fun. In *chem-bla-ics*. chem-bla-ics. https://doi.org/10.59350/cprj5-xnk74

Keywords
Bioclipse, Rdf, Xmpp

Abstract
Johannes asked me what the Lipinski Rule of Five for farnesol is, in reply to the matching XMPP cloud service.

chem-bla-ics

Johannes asked me what the Lipinski Rule of Five for farnesol is, in reply to the matching XMPP cloud service. Thanx to DBPedia for providing a machine readable form of the wikipedia entry:

Here's the solution (yes, suboptimal, but since we were hacking on XMPP support in Bioclipse) which shows the structure in JChemPaint and Jmol as bonus (gist:107507):

```
// Today, Johannes challenged me to use Bioclipse and XMPP to calculate the Lipinski Rule
// http://en.wikipedia.org/wiki/Farnesol
query = "Farnesol"

// Zero: clear the console
js.clear();
js.print("Query: " + query + "\n");

// One: connect to the XMPP hive, and make contact with the CDK descriptor service here i
xmpp.connect();
var service = xmpp.getService("descriptor.ws1.bmc.uu.se");
service.discoverSync(5000);
service.getFunctions();
var func = service.getFunction("LipinskiRuleOfFive");

// Two: take advantage of RDF, DBPedia
store = rdf.createStore()
rdf.importURL(store, "http://dbpedia.org/data/" + query + ".rdf")
rdf.importURL(store, "http://dbpedia.org/data/" + query + "/section1/Chembox_Identifiers.

// Three: run the SPARQL query and extract the SMILES from the List<List<String>>, and re
// the '@en' suffix
var sparql = "PREFIX dbprop: <http://dbpedia.org/property/> SELECT ?o WHERE { ?s dbprop:s
smiles = rdf.sparql(store, sparql).get(0).get(0)
smiles = smiles.substring(0, smiles.length()-3)

// Four: create a CML document
propane = cdk.fromSMILES(smiles);
js.print("Molecule SMILES: " + smiles + "\n");

// Five: call the function
result = func.invokeSync(propane.getCML(), 900000);
cmlReturned = xmpp.toString(result);

// Six: tune the CML so that the Bioclipse CML reader is happy
cmlReturned = cmlReturned.replace("xsd:int", "xsd:integer")

// Seven: extract the Lipinski Rule of Five score
```

chem-bla-ics

```
propertyList = cml.fromString(cmlReturned);
value = propertyList.getPropertyElements().get(0).
  getScalarElements().get(0).getValue()
js.print("Lipinski Rule of Five: " + value + "\n")

// Eight: while at it, let's create a 2D and open in JChemPaint
service = xmpp.getService("cdk.ws1.bmc.uu.se");
service.discoverSync(5000);
service.getFunctions();
func = service.getFunction("generate2Dcoordinates");
mol = cdk.fromSMILES(smiles)
result = func.invokeSync(mol.getCML(), 900000);
cmlReturned = xmpp.toString(result);
mol2d = cdk.fromCml(cmlReturned);
ui.open(mol2d)

// Nine: oh, and a 3D model in Jmol
func = service.getFunction("addExplicitHydrogens");
result = func.invokeSync(mol.getCML(), 900000);
mol = cdk.fromCml(xmpp.toString(result));
func = service.getFunction("generate3Dcoordinates");
result = func.invokeSync(mol.getCML(), 900000);
mol3d = cdk.fromCml(xmpp.toString(result));
file = "/Virtual/foo.cml";
ui.remove(file)
cdk.saveCML(mol3d, file);
ui.open(file)
```