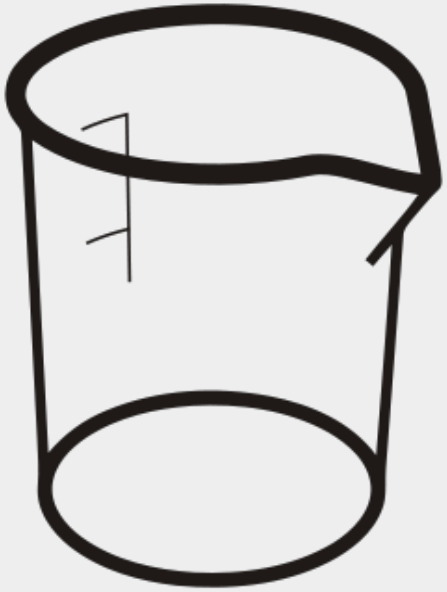

BEAKER

RDKit and OSRA in Bottle on Tornado

Michał Nowotka
ChEMBL Group
EMBL-EBI

OVERVIEW

1. Motivation
2. Ingredients
3. Overview
4. Examples
5. Future

~$ whoami:

- ChEMBL group staff member
- Web Applications Developer
- RDKit enthusiast end everyday user
- Available via email, skype, SO, github, etc.

DISCLAIMER

- This is a pet-project
- Developed in free time
- Unfinished
- Proof-of-concept

MOTIVATION

RDKit installation process:

Expectations:

```
activate rdkit-virtualenv
pip install rdkit
```

Reality:

- 859 hits for *build* on `rdkit-discuss`
- 498 hits for *install*

MOTIVATION

- *Virtualenv* is **essential** tool for python developers
- Remember *PIL* and *Pillow* case?
- VMs and Doker are not (yet?) an answer

MOTIVATION

Beaker is to ***RDkit*** like ***Sorl*** to ***Lucene***.

Beaker is to ***RDkit*** like ***aquarium*** to ***fish***.

- Server platform and RDKit container
- Provides cheminformatics tools
- REST-like HTTP API
- Easy to use from any programming language
- Install Beaker on one machine instead of installing RDKit on many hardware/software configurations

MOTIVATION

- More and more people are using RDKit as a service
- Why not to try to standardise it?
- Maybe even include in distribution...
- So we don't have to reinvent the wheel all the time

MOTIVATION

I'm into webservices anyway:
https://www.ebi.ac.uk/chemblws2

So why something different? **Software stack!**

- Django ORM
- Tastypie
- Kilolines of code
- And RDKit webservices can be dead simple!

RDKIT AND OSRA TOGETHER?

- Complementary Cheminformatics libraries
- OSRA extends possible RDKit input formats
- Both are **Open**
- OSRA is even harder to install and available only in C++

INGREDIENTS

- **OSRA** - utility designed to convert graphical representations of chemical structures.
- **RDKit** - Cheminformatics and Machine Learning Software.
- **Bottle** - fast, simple and lightweight WSGI micro web-framework for Python.
- **Tornado** - Python web framework and asynchronous networking library.

WHY THIS CHOICE?

- This needs to be lightweight.
- But fast and efficient.
- With small number of small dependencies.
- Well known, standard, virtualenv-friendly dependencies
- Small and simple codebase.
- Generic, elegant, robust API

OVERVIEW

Format conversion:

- `ctab2smiles / smiles2ctab`
- `ctab2inchi / inchi2ctab`
- `ctab2image / image2ctab`
- `inchi2inchiKey`

OVERVIEW

- All methods implemented as `POST` and some (`x2ctab`) as `GET`
- For `GET`, parameters have to be `base64` encoded
- All methods support batch processing

DEMO

```
curl -X POST http://beaker/ctab2image -F filedata="@out.mol" > o
ut.png

curl -X POST http://beaker/image2ctab --data-binary "@aspirin.jp
g" > out.mol

curl -X POST http://beaker/smiles2ctab --data-binary "@a.smi"
curl -X POST http://beaker/smiles2ctab --data "CNc1ncnc2c1ncn2Cc
1cccc(F)c1F

curl -X POST http://beaker/inchi2inchiKey --data "InChI=1S/C2H6O
/c1-2-3/h3H,2H2,1H3"
```

Better example: **Clippy**

POTENTIAL USE CASES

- Access from languages like java script, ruby
- Webapplications
- Mobile apps (camera + OSRA + RDKit)
- Small desktop apps (clippy)

FUTURE

- Different output formats: json, jsonp, xml
- Implement stub methods
- Compound descriptors: logP, TPSA, molWt, etc.
- Editing molecules: addHs, removeHs, kekulize
- Ring information, SSSR, sanitize…
- IUPAC names
- Pymol, matplotlib

CODE!

Beaker code is available as github repository:
https://github.com/mnowotka/chembl_beaker

Presentation code has its own repository:
https://github.com/mnowotka/beaker-presentation

THANK YOU!

Questions?