Chemical Structure Standardisation

The ChEMBL database and the `standardiser` tool

Francis Atkinson

Chemogenomics Group

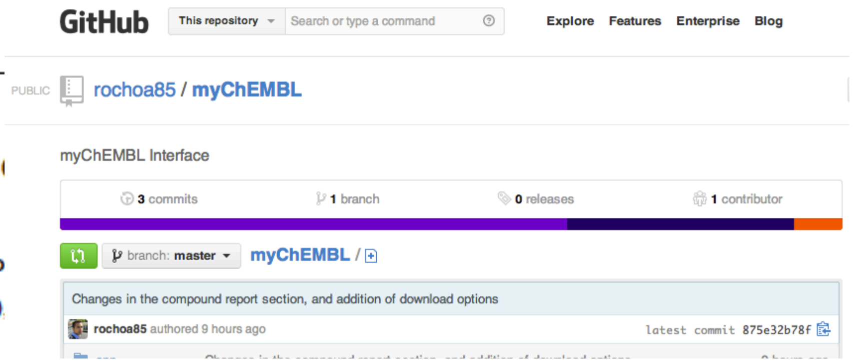
EMBL-EBI

What is ChEMBL?

- A freely-available source of bioactivity data
  - Secure on-line access
  - Downloadable in a variety of formats
- Core is data from key MedChem journals
  - J. Med. Chem, Bioorg. Med. Chem. Lett.
  - Manually extracted by CRO
- Supplemented by…
  - Subset of data from PubChem
  - Deposited datasets
    - NTD consortia, GSK kinase set (PKIS) etc.
- Accelrys Direct chemistry cartridge for Oracle

myChEMBL

- Postgres/RDKit version of ChEMBL
- packaged for distribution as VM



rochoa85 / **myChEMBL**

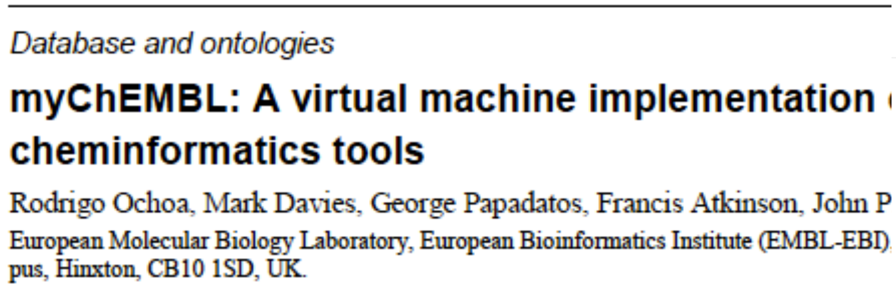
myChEMBL Interface

3 commits | 1 branch | 0 releases | 1 contributor

branch: master | **myChEMBL** /

Changes in the compound report section, and addition of download options

rochoa85 authored 9 hours ago — latest commit 875e32b78f

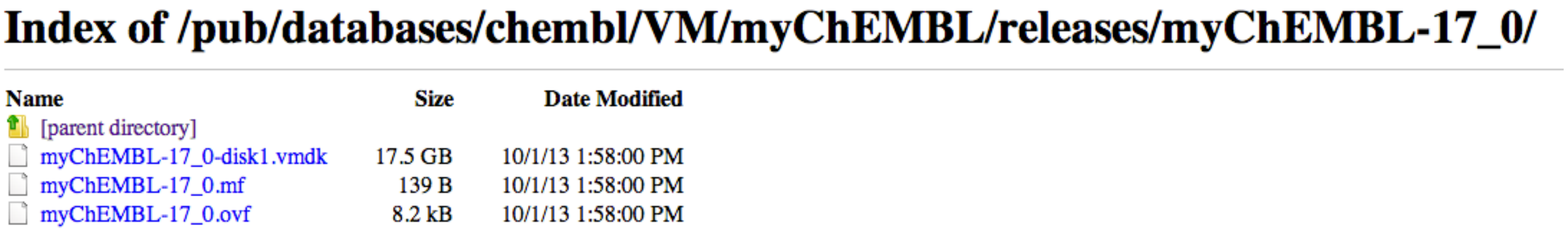


Database and ontologies

myChEMBL: A virtual machine implementation cheminformatics tools

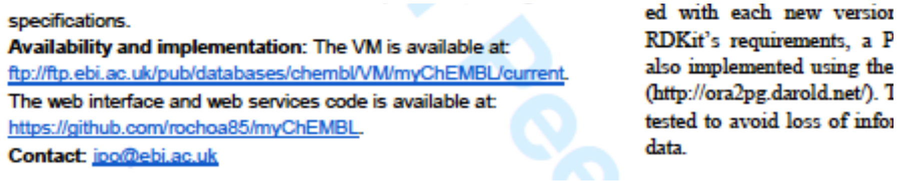
Rodrigo Ochoa, Mark Davies, George Papadatos, Francis Atkinson, John P

European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI) pus, Hinxton, CB10 1SD, UK.



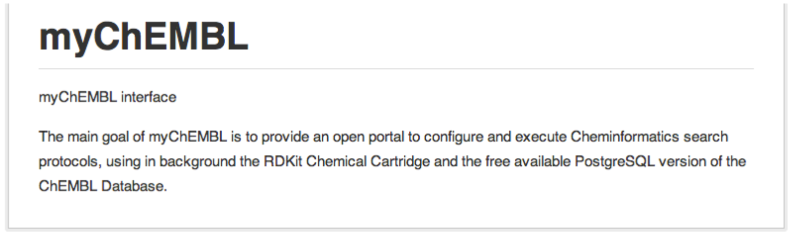
Index of /pub/databases/chembl/VM/myChEMBL/releases/myChEMBL-17_0/

Name	Size	Date Modified
[parent directory]		
myChEMBL-17_0-disk1.vmdk	17.5 GB	10/1/13 1:58:00 PM
myChEMBL-17_0.mf	139 B	10/1/13 1:58:00 PM
myChEMBL-17_0.ovf	8.2 kB	10/1/13 1:58:00 PM



specifications.
Availability and implementation: The VM is available at: ftp://ftp.ebi.ac.uk/pub/databases/chembl/VM/myChEMBL/current. The web interface and web services code is available at: https://github.com/rochoa85/myChEMBL.
Contact: jpo@ebi.ac.uk

ed with each new versio RDKit's requirements, a P also implemented using the (http://ora2pg.darold.net/). T tested to avoid loss of info data.



myChEMBL

myChEMBL interface

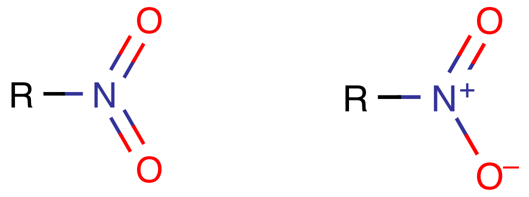
The main goal of myChEMBL is to provide an open portal to configure and execute Cheminformatics search protocols, using in background the RDKit Chemical Cartridge and the free available PostgreSQL version of the ChEMBL Database.

Heterogeneous sources of structures

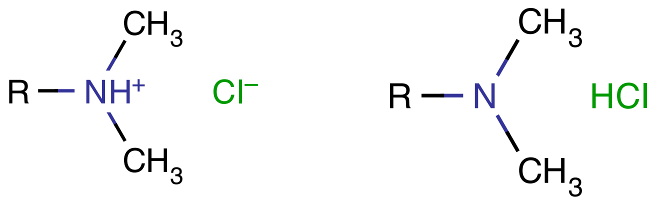
- Chemical Literature
  - Redrawn by CRO from depictions in articles
- PubChem
- Other bioactivity databases
- Data depositions
- Typically received as MDL molfiles

Multiple representations

- Hypervalent *vs.* charge-separated



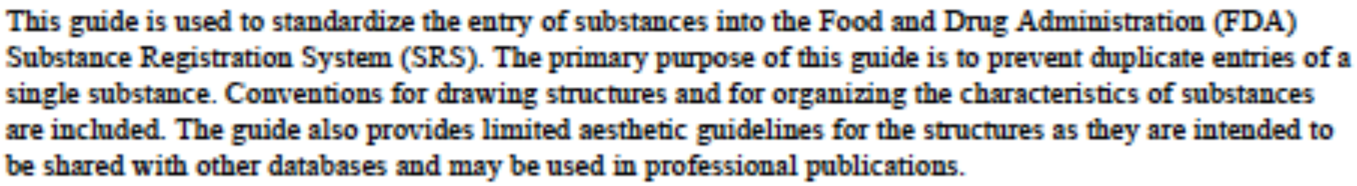
- Charged *vs.* neutral salts



- Chemical cartridge will treat these as different
  - Could cause SSS to fail if alternative depiction used as query
  - Standardization is required

Standardisation

- The FDA have published a compound depiction SOP for their Substance Registration System



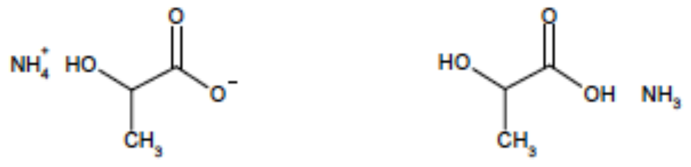
This guide is used to standardize the entry of substances into the Food and Drug Administration (FDA) Substance Registration System (SRS). The primary purpose of this guide is to prevent duplicate entries of a single substance. Conventions for drawing structures and for organizing the characteristics of substances are included. The guide also provides limited aesthetic guidelines for the structures as they are intended to be shared with other databases and may be used in professional publications.

- Used as basis for ChEMBL standardisation rules
  - with some modifications, *e.g.* …

8. Salts Formed by the Reaction of Ammonia with Acids

A salt formed by the reaction of ammonia with an acid is represented as the ammonium ion and the conjugate base of the acid.

Example: AMMONIUM LACTATE

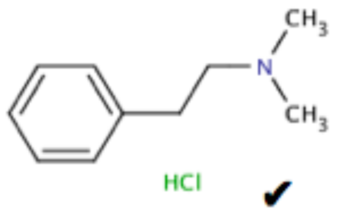


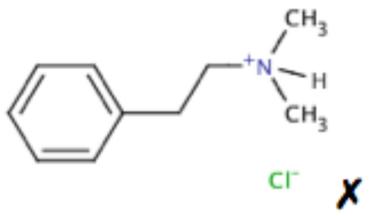
Correct Incorrect

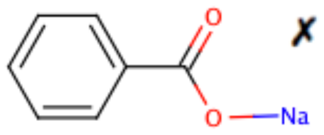
ChEMBL would treat these the same as other salts, *i.e.* they would be neutralized

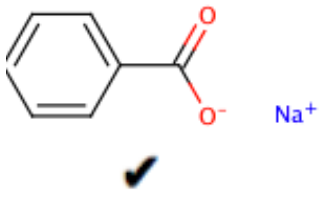
Examples of rules: charges

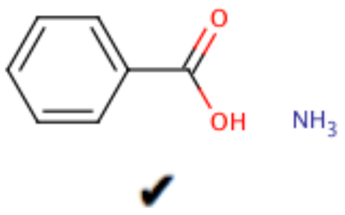
- Consistent rules for drawing salts

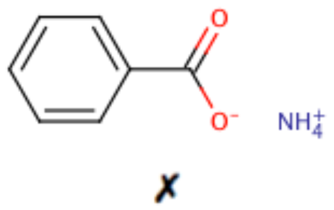


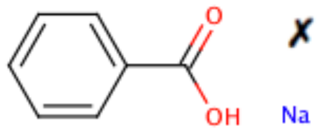








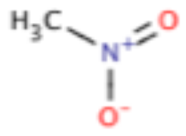




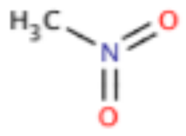
- Compounds to be charge-neutral if possible
  - quaternary N an exception if counterion unknown

Example of rules: functional groups

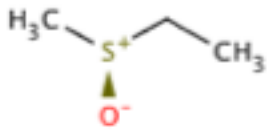
- Charge-separated preferred over hypervalent



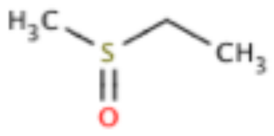
✔



✗

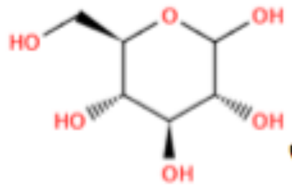


✔

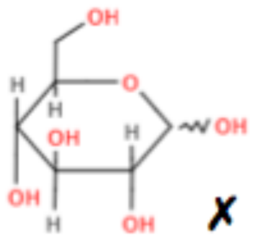


✗

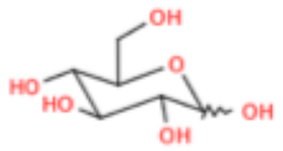
- Stylistic rules for sugars, peptides, steroids *etc.*



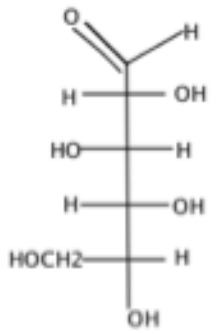
✔



✗



✗



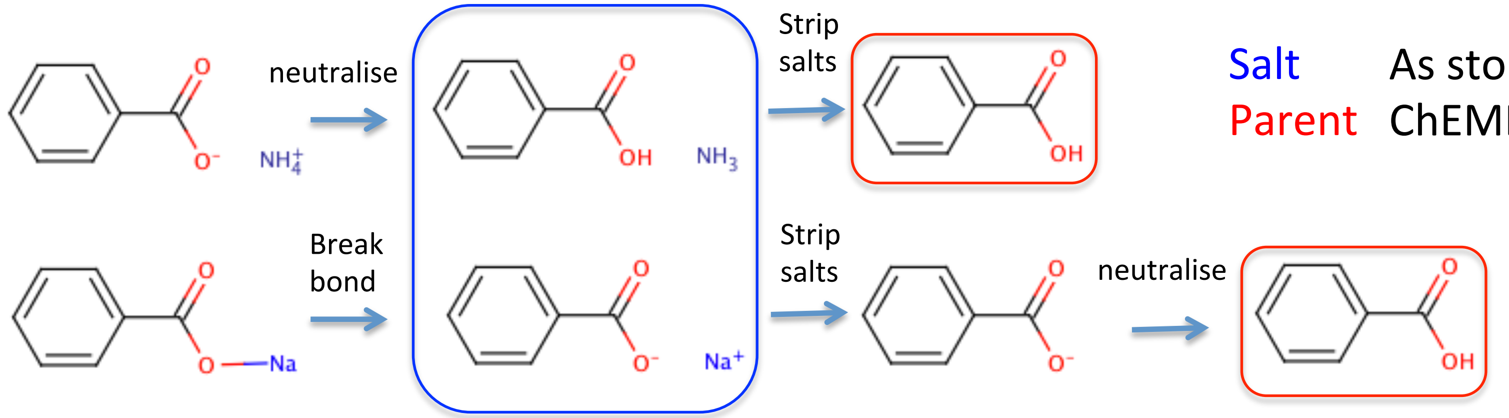
✗

Salt stripping

- Salt/solvate components should not affect the biological activity of a compound
  - There can be exceptions
    - *e.g.* salt form may affect solubility
  - However, any 3D or QSAR modelling should be done using only the bioactive ‘parent’
- It is desirable to be able to view compounds with a common parent together
- It can be appropriate to aggregate data for the parent compound

Salt stripping (2)

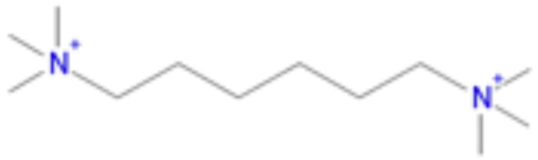
- Counterions and solvent molecules are removed using a custom ‘salt dictionary’
  - Acids with inorganic cations may require a further round of charge-neutralization…



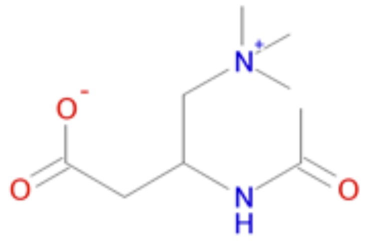
- The parent is registered as a separate structure
  - The parent and original salt are linked *via* the ‘molecule hierarchy’ table

Curator intervention

- Structures with permanent charges
  - *i.e.* counterion not recorded
  - Zero charge used as check on neutralization steps



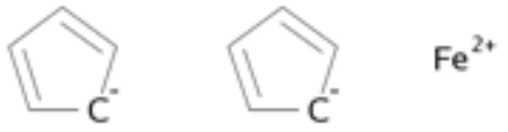
- Zwitterions
  - Naïve neutralization could introduce errors



- These types automatically routed for inspection
  - Possible manual standardisation

Inorganics & organometallics

- Handled poorly by current chemoinformatics tools…
  - multicenter bonding
  - coordination complexes
  - Non tetrahedral stereochemistry
    - Cannot distinguish *cis*- and *trans*-platins



ferrocene ?!

- Charge-balancing, salt-stripping *etc.* are difficult
- Cannot be properly searched for by cartridge
- Not put through main standardisation process
  - May be redrawn for clarity in some cases
    - *e.g.* approved drugs such as the platins
- Structures now excluded (post-CHEMBL17)
  - ~3200 structures affected
  - Bioactivities are retained

Implementation

- Rules implemented using Pipeline Pilot
  - Run by ChEMBL's chemical curator
  - Some manual intervention in difficult cases
- Protocols not currently publically available
  - Tightly coupled with data-loading pipeline
  - Subject to change as new issues identified
- Pipeline Pilot is commercial software
- We have a interest in an FOSS equivalent…
  - *e.g.* for loading proprietary data into myChEMBL

`standardiser`

- Tool to pre-process structures for modelling
- Molecular representation can affect results…
  - Descriptors, forcefields, QM
- Need to standardise representation
  - Common to both training and application
- 'Inspired by' ChEMBL curation & InChI preprocessing
  - Only interested in parent (bioactive) component
  - Standardise tautomers
    - *i.e.* hydroxy-pyridine -> pyridone
    - *Not* tautomer canonicalization
  - No manual intervention
- Implemented in Python & RDKit
  - Runs under Python2.6+
  - Fully open-source

Procedure

- Break bonds to Group I or II metals
- Neutralize charges by adding/removing protons
- Apply standardization rules
- Neutralize any charges exposed by rules
- Discard any salt/solvate components
- Return standardized parent

Limitations

- Rule set could be expanded
  - *e.g.* will review RCS / CVSP rules
- No attempt to handle inorganics
  - Impractical with current tools
- No attempt to produce ‘canonical’ tautomer
  - Could flag tautomeric molecules for inspection?
- re-charging not handled

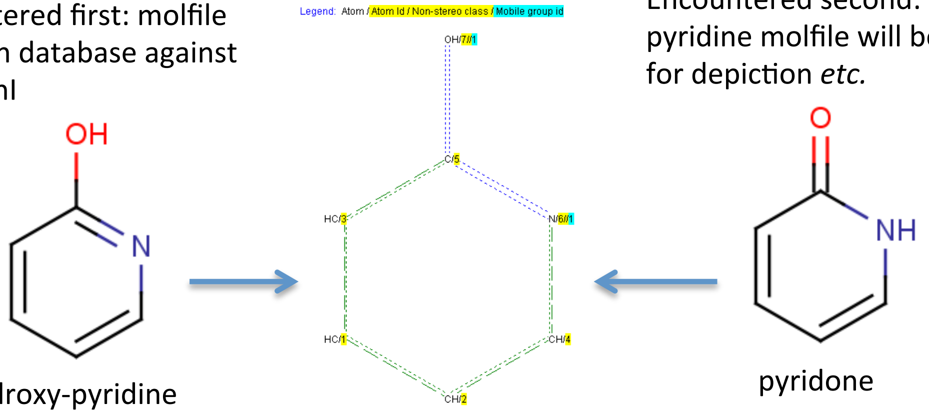
Key differences to ChEMBL

- ChEMBL makes no attempt to standardise tautomers
- InChI codes are used for registration (*i.e.* assignment of database identifiers/keys)
  - *i.e.* two molecules are the if they have the same InChI
- For any molecule, the first-encountered tautomer will always be used
  - to generate images, for searching and in downloadable SDF files *etc.*
- Thus, for example: if the first time a molecule encountered it is shown as the hydroxy-pyridine, this tautomer will be stored (as a molfile) and will be used for the depiction of this molecule even where an alternative tautomer is encountered in a later document
- By contrast, standardiser *does* attempt to standardise tautomers
  - *e.g.* the pyridone is preferred, as it is likely to be the lower-energy tautomer

ChEMBL

Encountered first: molfile stored in database against this InChI

Encountered second: hydroxy-pyridine molfile will be used for depiction *etc.*

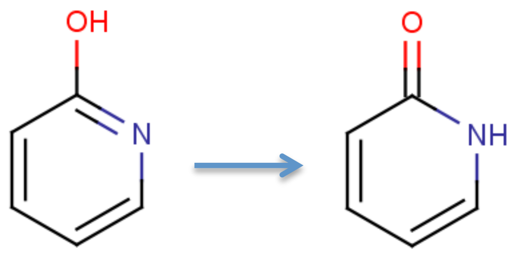


hydroxy-pyridine

pyridone

InChI=1S/C5H5NO/c7-5-3-1-2-4-6-5/h1-4H,(H,6,7)

`standardiser`



input

output

Example

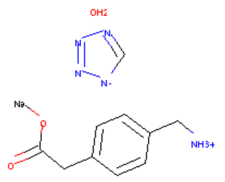
- Provided as a Python package
  - Compatible with python2.6 (for etoxws VM)
- A simple driver program for batch processing is included
  - Accepts SDF or SMILES input

```
In [2]: from standardise import standardise
```

Examples

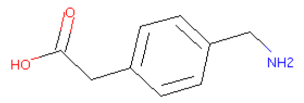
```
In [3]: mol = Chem.MolFromSmiles("[Na]OC(=O)Cc1ccc(C[NH3+])cc1.c1nnn[n-]1.O")
        mol
```

Out[3]:



```
In [4]: parent = None

        try:

            parent = standardise.apply(mol)

        except standardise.StandardiseException as e:

            logging.warn(e.message)

        parent
```

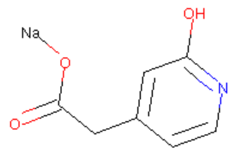
Out[4]:



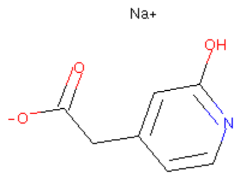
Modules within package

- Modules implementing different steps may be called independently
  - could be incorporated into different workflows

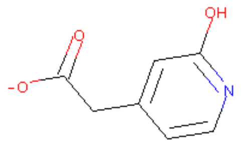
```
In [13]: mol = Chem.MolFromSmiles("[Na]OC(=O)Cc1cc(O)ncc1")
         mol

Out[13]:
```



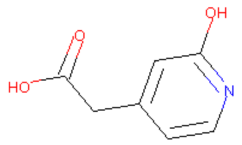
```
In [16]: mol = break_bonds.apply(mol)
         mol

Out[16]:
```



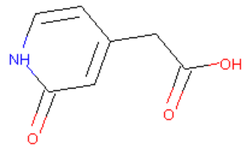
```
In [25]: mol = [x for x in Chem.GetMolFrags(mol, asMols=True) if not unsalt.is_nonorganic(x)][0]
         mol

Out[25]:
```



```
In [26]: mol = neutralize.apply(mol)
         mol

Out[26]:
```

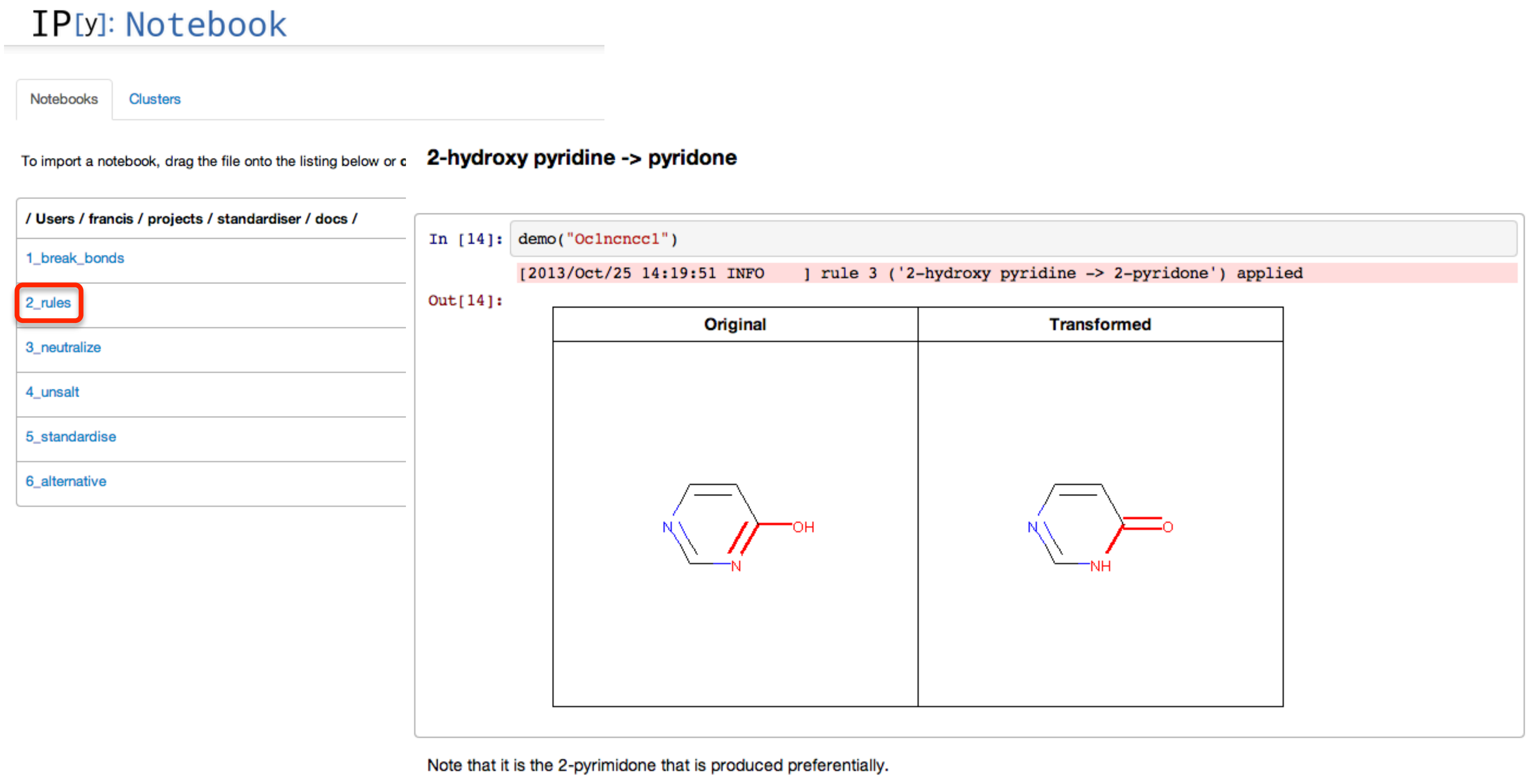


```
In [28]: mol = rules.apply(mol)
         mol

Out[28]:
```



Documentation

- Provided as IPython Notebooks
  - Also available as static web pages



IP[y]: Notebook

Notebooks Clusters

To import a notebook, drag the file onto the listing below or c

/ Users / francis / projects / standardiser / docs /

1_break_bonds

2_rules

3_neutralize

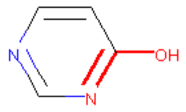
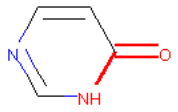
4_unsalt

5_standardise

6_alternative

2-hydroxy pyridine -> pyridone

```
In [14]: demo("Oc1ncncc1")
         [2013/Oct/25 14:19:51 INFO     ] rule 3 ('2-hydroxy pyridine -> 2-pyridone') applied
Out[14]:
```

Original	Transformed
	

Note that it is the 2-pyrimidone that is produced preferentially.

Futher Documentation

- Also included are some pages describing issues encountered
  - Intended to stimulate debate on future directions of tool

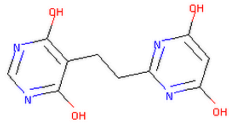
Issue 01

The molecule below is a simplified version of real examples. It illustrates a problem with the original (very simple) version of the algorithm.

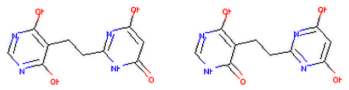
While the code has been fixed to cope with cases like this, it should be noted that it is not really clear how often problems of this nature would occur in practice, and thus how much the extra complexity needed to deal with them is warranted.

It would also be possible to deal with this issue by rewriting the SMARTS transforms to be more specific (*i.e.* by excluding cases such as the present one). However, tt was felt that this would likely end up being more compolicated than the chosen method, as mutiple complicated SMARTS could become necessary to represent fairly simple transforms.

```
In [2]:  mol = Chem.MolFromSmiles("Oc1nc(CCc2c(O)ncnc2O)nc(O)c1")
         mol

Out[2]:
```

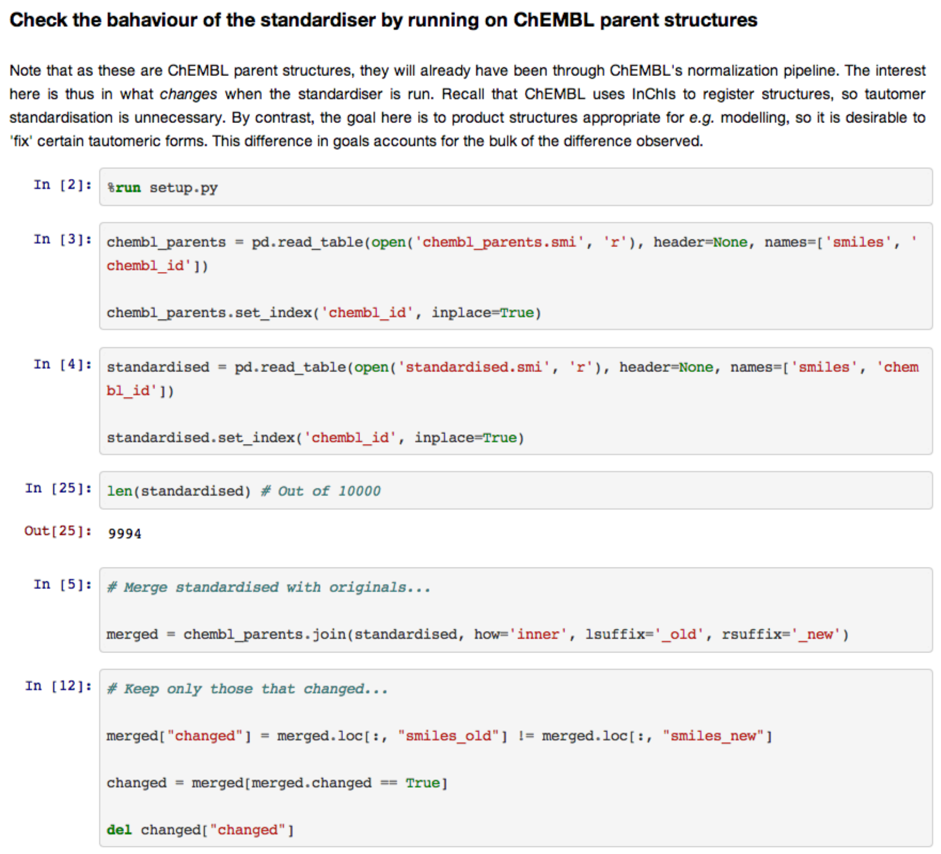


```
In [109]: rxn = AllChem.ReactionFromSmarts("[OX2H1:1]-[c:2]:[nX2:3]>>[OH0:1]=[c:2]:[nH1:3]") # 2-hydroxy pyridine -> 2-pyridone

In [111]: # Run transform...
          # NB As the transorm is a one-component reaction, only the first component from each product tuple will exist

          products = [x[0] for x in rxn.RunReactants((mol,))]

          # Sanitize product mols...

          for x in products: Chem.SanitizeMol(x)

          # Keep unique products (i.e. tautomers) only

          products = {Chem.MolToSmiles(x): x for x in products}.values()

          # Depict...

          Draw.MolsToGridImage(products)

Out[111]:
```



Futher Documentation

- The tool will be run on diverse test sets and the results posted
  - Again, designed to detect flaws and promote discussion
  - One example up so far…

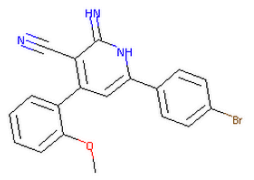
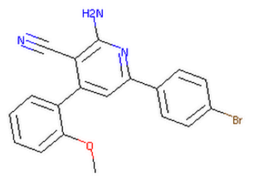
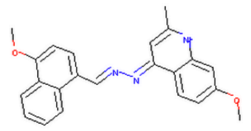
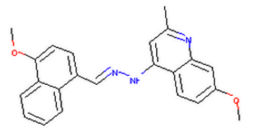
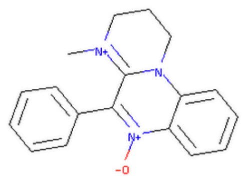
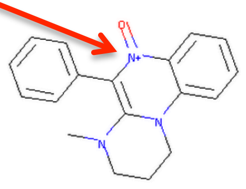


Check the bahaviour of the standardiser by running on ChEMBL parent structures

Note that as these are ChEMBL parent structures, they will already have been through ChEMBL's normalization pipeline. The interest here is thus in what *changes* when the standardiser is run. Recall that ChEMBL uses InChIs to register structures, so tautomer standardisation is unnecessary. By contrast, the goal here is to product structures appropriate for *e.g.* modelling, so it is desirable to 'fix' certain tautomeric forms. This difference in goals accounts for the bulk of the difference observed.

```
In [2]:  %run setup.py

In [3]:  chembl_parents = pd.read_table(open('chembl_parents.smi', 'r'), header=None, names=['smiles', '
         chembl_id'])

         chembl_parents.set_index('chembl_id', inplace=True)

In [4]:  standardised = pd.read_table(open('standardised.smi', 'r'), header=None, names=['smiles', 'chem
         bl_id'])

         standardised.set_index('chembl_id', inplace=True)

In [25]: len(standardised) # Out of 10000

Out[25]: 9994

In [5]:  # Merge standardised with originals...

         merged = chembl_parents.join(standardised, how='inner', lsuffix='_old', rsuffix='_new')

In [12]: # Keep only those that changed...

         merged["changed"] = merged.loc[:, "smiles_old"] != merged.loc[:, "smiles_new"]

         changed = merged[merged.changed == True]

         del changed["changed"]
```

```
In [28]: not_hydroxy_pyridine

Out[28]:
```

	mol_old	mol_new
chembl_id		
CHEMBL570855		
CHEMBL532261		
CHEMBL1186335		

This isn't good…

Future developments

- We would like to open-source this code
  - Tool would improve as more bugs/issues are found
  - *e.g.* establish wiki for community annotation of rules
  - Need to tidy up code and documentation
- More testing on diverse compound sets
- Incorporate into registration tool for myChEMBL
  - Longer term project
  - recall differences between standardisation for modelling and for database registration