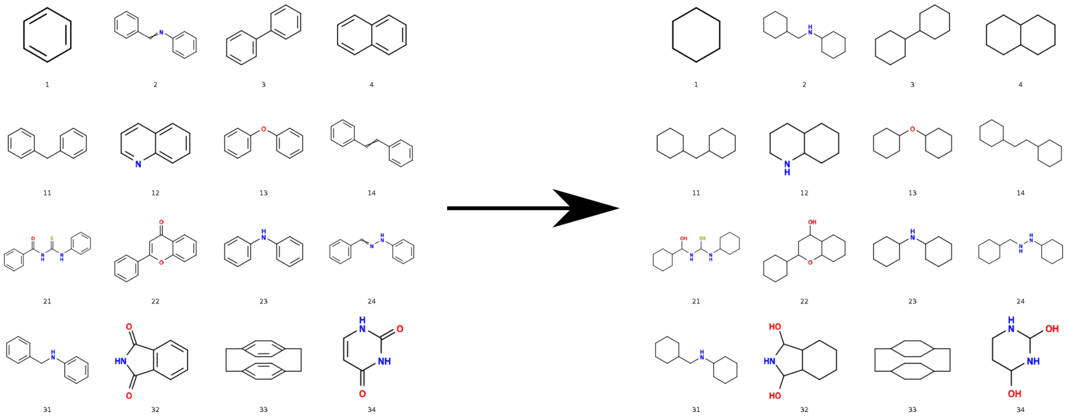
1 Background

The Bemis-Murcko scaffold[1] provided by DataWarrior[2] retains information about bond order and chirality. Sometimes, however, it suffices to retain only atom connectivity, like an assumption «there are only single bonds». Note, DataWarrior equally offers the export of Bemis-Murcko skeleton, however this simplifies e.g. the scaffold about an imidazole into one of cyclopentane.



2 Typical use

The script processes one, or multiple SMILES strings provided in a pattern of

```
python saturate_murcko_scaffolds.py [-h] inputs [inputs ...]
```

Running from the CLI, this translates for example to

```
$ python3 saturate_murcko_scaffolds.py c1ccncc1 c1ccccc1
C1CCNCC1
C1CCCCC1
```

It equally is possible to provide the input as a list of SMILES in a text file. As an example run in Linux Debian 13:

```
$ cat test.smi
c1ccncc1
c1ccccc1
$ python3 saturate_murcko_scaffolds.py test.smi
C1CCNCC1
C1CCCCC1
```

[1] Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39*, 2887–2893 (https://doi.org/10.1021/jm9602928).

[2] Sander, T.; Freyss, J.; Von Korff, M.; Rufener, C. DataWarrior: An Open-Source Program For Chemistry Aware Data Visualization And Analysis. *J. Chem. Inf. Model.* **2015**, *55*, 460–473 (https://doi.org/10.1021/ci500588j). The program, (c) 2002–2024 by Idorsia Pharmaceuticals Ltd., is freely available under http://www.openmolecules.org. For the source code (GPLv3), see https://github.com/thsa/datawarrior.

In a mixed input queue, SMILES strings provided via the CLI are processed prior to SMILES provided via one, or multiple input file(s). If wanted, the output to the CLI can be redirected to (piped into) the input of the next command-line utility, or appended to an already existing permanent record, for instance

```
$ python3 saturate_murcko_scaffolds.py test.smi > output.smi
$ cat output.smi
C1CCNCC1
C1CCCCC1
```

The script requires only functionality provided by the standard library of Python 3. Backed by tests with `pytest` and multiple runner instances GitHub provides, the recommended usage picks any combination of (ubuntu-20.04, ubuntu-22.04, ubuntu-24.04, windows-2019, windows-2022, macos-14) as hosting operating system on one hand, and either Python 3.10, or Python 3.12 as Python interpreter on the other. Anecdotally, the script was observed to equally work in ubuntu 18.04 and Python 3.6.9, too.

3 Example

For a collection of organic materials, the Bemis-Murcko scaffolds were extracted with DataWarrior (then release 5.0.0 for Linux, January 2019) as listing `input.smi` including higher bond orders (see folder `demo`) with a redirect of the output into file `input_sat.smi`. The effect of the «artificial saturation» is easy to recognize while comparing the scaffold lists (fig. 1) in a difference view.

OpenBabel[3] is used to illustrate the work of the script. The instructions to the CLI follow the pattern of

```
obabel -ismi test_input.smi -O test_input_color.svg -xc10 -xr12 -xl --addinindex
```

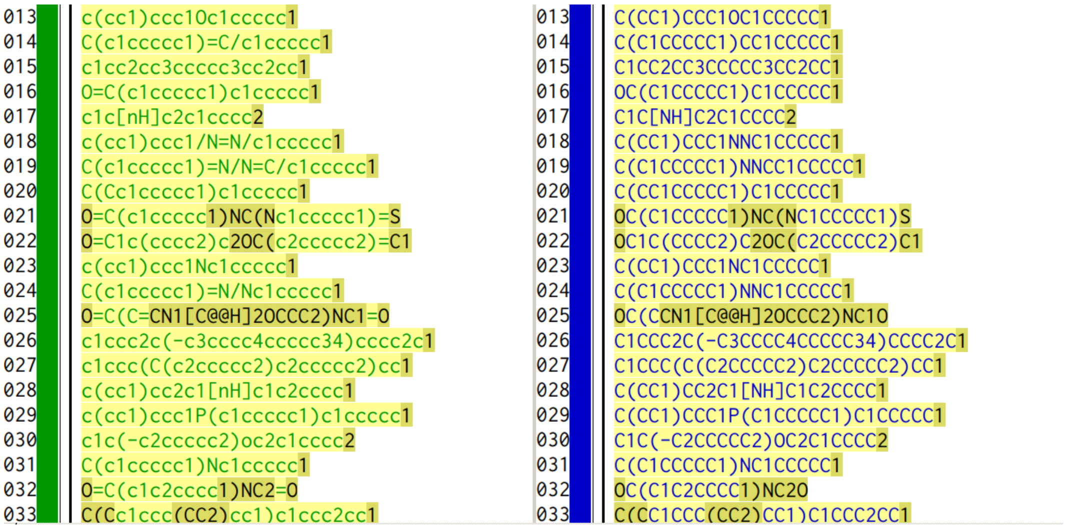
to generate a `.svg` file (vector representation), or

```
obabel -ismi test_input_sat.smi -O test_input_sat_color.png -xc10 -xr12 -xl
↪  --addinindex -xp 3000
```

to generate a bitmap `.png` with structure formulae depicted in a grid of 10 columns by 12 rows. Script `series.py` automates the generation of the illustrations about both structure data sets.

It is remarkable how well OpenBabel's displays the molecular structures with advanced motifs. In addition to those shown in the first illustration of this guide, see sub-folder `test_data` for a more extensive survey (e.g., the scaffold of cyclophane [entry #33], sparteine [#38], or adamantane [#50]).

[3] https://github.com/openbabel/openbabel For the most recent documentation, see https://open-babel.readthedocs.io/en/latest/



```
013 c(cc1)ccc1Oc1ccccc1
014 C(c1ccccc1)=C/c1ccccc1
015 c1cc2cc3ccccc3cc2cc1
016 O=C(c1ccccc1)c1ccccc1
017 c1c[nH]c2c1cccc2
018 c(cc1)ccc1/N=N/c1ccccc1
019 C(c1ccccc1)=N/N=C/c1ccccc1
020 C(Cc1ccccc1)c1ccccc1
021 O=C(c1ccccc1)NC(Nc1ccccc1)=S
022 O=C1c(cccc2)c2OC(c2ccccc2)=C1
023 c(cc1)ccc1Nc1ccccc1
024 C(c1ccccc1)=N/Nc1ccccc1
025 O=C(C=CN1[C@@H]2OCCC2)NC1=O
026 c1ccc2c(-c3cccc4ccccc34)cccc2c1
027 c1ccc(C(c2ccccc2)c2ccccc2)cc1
028 c(cc1)cc2c1[nH]c1c2cccc1
029 c(cc1)ccc1P(c1ccccc1)c1ccccc1
030 c1c(-c2ccccc2)oc2c1cccc2
031 C(c1ccccc1)Nc1ccccc1
032 O=C(c1c2cccc1)NC2=O
033 C(Cc1ccc(CC2)cc1)c1ccc2cc1
```

```
013 C(CC1)CCC1OC1CCCCC1
014 C(C1CCCCC1)CC1CCCCC1
015 C1CC2CC3CCCCC3CC2CC1
016 OC(C1CCCCC1)C1CCCCC1
017 C1C[NH]C2C1CCCC2
018 C(CC1)CCC1NNC1CCCCC1
019 C(C1CCCCC1)NNCC1CCCCC1
020 C(CC1CCCCC1)C1CCCCC1
021 OC(C1CCCCC1)NC(NC1CCCCC1)S
022 OC1C(CCCC2)C2OC(C2CCCCC2)C1
023 C(CC1)CCC1NC1CCCCC1
024 C(C1CCCCC1)NNC1CCCCC1
025 OC(CCN1[C@@H]2OCCC2)NC1O
026 C1CCC2C(-C3CCCC4CCCCC34)CCCC2C1
027 C1CCC(C(C2CCCCC2)C2CCCCC2)CC1
028 C(CC1)CC2C1[NH]C1C2CCCC1
029 C(CC1)CCC1P(C1CCCCC1)C1CCCCC1
030 C1C(-C2CCCCC2)OC2C1CCCC2
031 C(C1CCCCC1)NC1CCCCC1
032 OC(C1C2CCCC1)NC2O
033 C(CC1CCC(CC2)CC1)C1CCC2CC1
```

Figure 1: Difference view of the SMILES strings of a Murcko scaffold *prior* (left hand column) and *after* an «artificial saturation» (right hand column). The processing affects explicit bond order indicators, e.g. double bond (equality sign, e.g., line #14), triple bond bond (number sign #, not shown); or about implicit aromatization (lower case to upper case) for atoms of carbon, nitrogen, oxygen (depicted); or phosphorus, sulfur (not depicted). Stereochemical indicators about double bonds will be removed (e.g., slashes in lines #18 and #19). Descriptors of stereogenic centers (@-signs, e.g., line #25) and charges (not shown) are copied verbatim.

4 Known peculiarities

The script provides «saturation» by dropping explicit information related to double and triple bonds which SMILES encode (=, # regarding bond order; / (forward slash), \ (backward slash) regarding (*cis*)-(*trans*) relationship around double bonds). While processing double bonds of e.g., ketones to yield secondary alcohols, the script refrains from the assignment of new CIP priorities and a corresponding label. It then depends on the program used for a visualization, if an explicit wedge is used (e.g., OpenBabel), or the absence of information is highlighted (e.g., as question mark in DataWarrior, or the project of CDK depict[4]) as ambiguous. Absolute configuration of stereogenic centers (indicated in SMILES with the @ sign) already assigned in the input however is retained.

For a selection of elements (C, N, O, P, S), the implicit description of aromatic systems (e.g., as c1ccncc1 in pyridine, c1c[nH]cc1 in pyrrol) is recognized. To offer a «saturation», these characters returned as upper case characters to yield e.g., piperidine (C1CCNCC1) and pyrrolidine (C1C[NH]CC1).

The script equally preserves up to one single negative, or single positive charge of these five elements (e.g., [O-]c1ccccc1 about the phenolate anion, and C[N+](c1ccccc1)(C)C about *N,N,N*-trimethylbenzenaminium cation). Here, it can be sensible to «sanitize» the results this

[4] https://www.simolecule.com/cdkdepict/depict.html For the mentioned annotation of CIP labels, change No Annotation (second pull down menu from the left) to CIP Stereo Label.

script provides by other libraries as e.g. RDKit.[5]

The capitalization of the five characters is constrained to prevent non sensible transformations of e.g., an (implicitly) aromatic atom of tin `[sn]` into the invalid form `[SN]`. Though the script is going to write tin as `[Sn]`, an adjustment of valence for elements written with two characters is beyond the current scope of the script.

A SMILES string may describe more than one molecule. Thus, the concatenation with `"."` (period character) as seen for example in descriptions of co-crystals like about 1,4-benzoquinone and hydroquinone, `C1=CC(=O)C=CC1=O.c1cc(ccc1O)O`, is retained. The example is resolved as `C1CC(O)CCC1O.C1CC(CCC1O)O`.

5 License

Norwid Behrnd, 2019–24, GPLv3.

[5] For an overview about the freely available RDKit library, see www.rdkit.org. An introduction into the topic of «molecular sanitization» is provided in the section of this very title in the on-line RDKit Book.