

IUPAC SMILES+

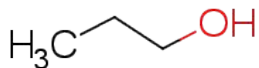
IUPAC Project: [2019-002-2-024](#)
10 minute update

August 23-24, 2019
InChI Symposium
San Diego, CA

Vincent F. Scalfani
The University of Alabama
vfscalfani@ua.edu

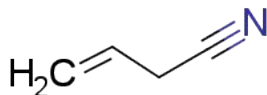
SMILES

SMILES – **S**implified **M**olecular **I**nterface **L**anguage **E**xecutable **S**ystem [1]. Compact line notation for representing molecules and reactions. Four main rules [1-3]:



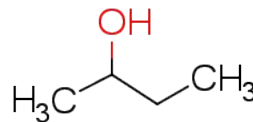
CCCO

1. atomic symbols



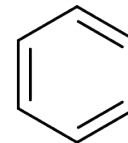
C=CCC#N

2. double '=', triple
bonds '#'



CCC(C)O

3. branching uses
parentheses



C1=CC=CC=C1

4. ring closures
use digits

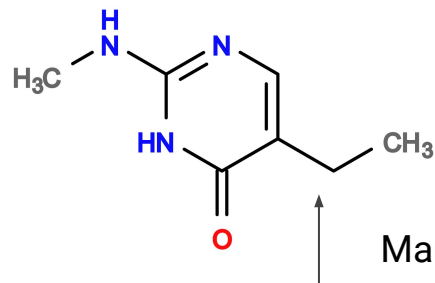
Since 1988, Daylight Chemical Information Systems have developed SMILES [3]. Widely used format in cheminformatics.

[1] Weininger, D. *J. Chem. Inf. Comput. Sci.* **1988**, 28, 31-36. DOI: [10.1021/ci00057a005](https://doi.org/10.1021/ci00057a005); [2] Weininger, D.; Weigniner, A.; Weininger, J.L. *Chem. Des. Autom. News*, **1986**, 1(8), 2-15.; [3] <https://www.daylight.com/dayhtml/doc/theory/>

SMILES vs. InChI? No, SMILES and InChI

SMILES are complementary to InChI, **we need both**. Three main reasons:

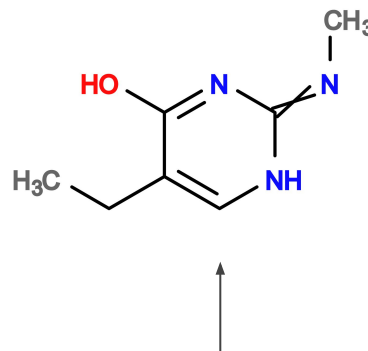
1. InChI is a machine descriptor identifier, powerful at linking information [1]. SMILES are difficult to link, but more closely tied to human (chemist) representation.



Many valid SMILES

```
O=C1NC(NC)=NC=C1CC
CNc1ncc(CC)c(=O)[nH]1
N(C)c1[nH]c(=O)c(cn1)CC
CNc1ncc(c(=O)[nH]1)CC
c1(=O)[nH]c(NC)ncc1CC
n1c([nH]c(=O)c(CC)c1)NC
n1cc(c(=O)[nH]c1NC)CC
...
```

One Standard InChI



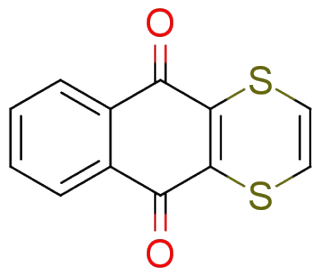
InChI normalization may return representation other than chemist preferred choice (can be lossy without AuxInfo).

```
InChI=1S/C7H11N3O/c1-3-5-4-9-7(8-2)10-6(5)11/
h4H,3H2,1-2H3,(H2,8,9,10,11)
```

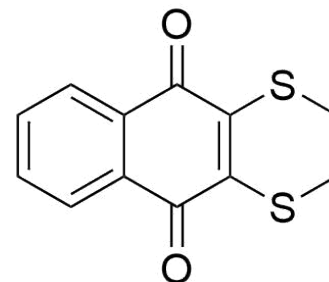
SMILES vs. InChI? No, SMILES and InChI

SMILES are complementary to InChI, **we need both**. Three main reasons:

2. We need to prevent corruption of InChI from SMILES input data (e.g., SMILES → InChI API or SMILES → molfile → InChI)



s1ccsc2=c1c(=O)c1c(c2=O)cccc1



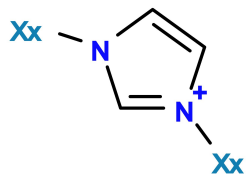
MarvinSketch (ChemAxon JChem) 18.1
JEBDOQPBSBGAP-UHFFFAOYSA-N

ChemDraw 18.1
IVQJELKILULDFK-UHFFFAOYSA-N

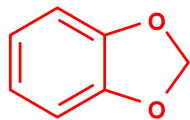
SMILES vs. InChI? No, SMILES and InChI

SMILES are complementary to InChI, **we need both**. Three main reasons:

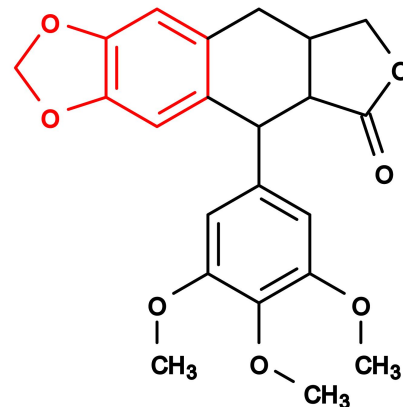
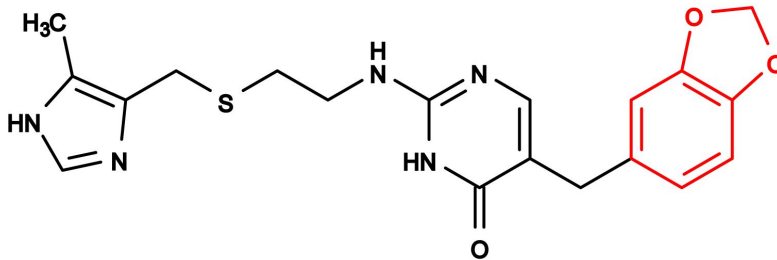
3. Variability handling [*] and SMARTS (a superset of SMILES) application, a popular substructure/pattern searching method [1].



[*][N+]1=CN([*])C=C1



SMARTS pattern for Benzodioxole
c1cccc-2c1-[#8]-[#6]-[#8]-2



[1] <https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html>

Current SMILES Specification Documents

Unlike InChI, SMILES are not always well defined....

- Daylight's last update to specification was in **2011** [1].
- OpenSMILES, a Blue Obelisk community driven effort created a non-proprietary open specification of SMILES (**2007**) [2].
- OpenSMILES clarified some ambiguities in the Daylight SMILES specification.

OpenSMILES specification

Craig A. James
version 1.0, 2016-05-15
Current specification

www.opensmiles.org

Copyright © 2007-2016, Craig A. James

Content is available under [GNU Free Documentation License 1.2](#)

Contributors: Richard Apodaca, Noel O'Boyle, Andrew Dalke, John van Drie, Peter Ertl, Geoff Hutchison, Craig A. James, Greg Landrum, Chris Morley, Egon Willighagen, Hans De Winter, Tim Vandermeersch, John May

1. Introduction

"... we cannot improve the language of any science, without, at the same time improving the science itself; neither can we, on the other hand, improve a science, without improving the language or nomenclature which belongs to it ..."

[Antoine Lavoisier_1787](#)

1.1. Purpose

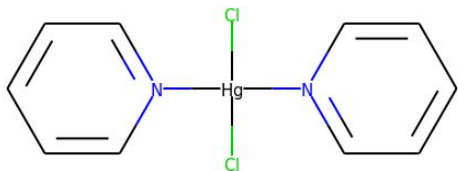
This document formally defines an [open specification](#) version of the [SMILES](#) language, a typographical [line notation](#) for specifying chemical structure. It is hosted under the banner of the [Blue Obelisk](#) project, with the intent to solicit contributions and comments from the entire computational chemistry community.

[1] daylight.com/dayhtml/doc/theory/index.html

[2] opensmiles.org/opensmiles.html

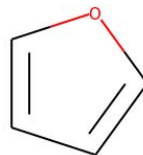
Many SMILES Extensions Exist

Documentation from toolkit providers often extend Daylight and OpenSMILES specification with additional features:



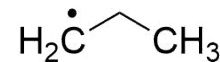
Cl[Hg]23Cl.c1ccn->2cc1.c1ccn->3cc1

[1] RDKit dative bonds, -> and <-



c%(1000)occc%(1000)

[2] Ring closure notation > 100, %(nnn). (Jmol, Open Babel, RDKit)



CCc

[3] Open Babel radical centers via lowercase symbols

[1] rdkit.org/docs/RDKit_Book.html#dative-bonds


[2] Hanson, R.M. *J. Cheminform.* **2016**, 8:50. DOI: 10.1186/s13321-016-0160-4

[3] openbabel.org/docs/current/Features/Radicals.html

SMILES Interoperability

Compatibility and interoperability issues can exist in SMILES reading. Examples:

1. Reading aromatic SMILES and disagreement with SMILES valence models [1].
2. SMILES support (e.g., higher order stereochemistry) and extension symbols and support varies across toolkits.

Avalon	Cl2 Cl4 Br2 Br4 I2 I4	<p>"Happy valence models are all alike; every unhappy valence model is unhappy in its own way." ...with apologies to Tolstoy</p> <p>'9.5'/15 correct now. When I started, it was 6/15.</p> 
BIOVIA Draw	Cl2 Cl4 Br2 Br4 I2 I4	
Cactvs	N4 P4 S3 S5 (or none*)	
CDK		
CEX (Weininger)		
ChemDoodle		
ChemDraw		
Indigo†		
iwtoolkit	N4 Cl2 Cl3 Cl4 Cl5 Br2 Br3 Br4 I2 I4 (or P4 S3 S5*)	
JChem		
KnowItAll		
OEChem		
Open Babel		
OpenChemLib	N4 Cl2 Cl4 Br2 Br4 I2 I4	
RDKit†	P6 I3 I4	

* If the default options are modified
† Results exclude 17 atom types rejected by Indigo, and 19 rejected by RDKit

[1] O'Boyle, N.M.; Mayfield, J. W.; Sayle, R. A. A De Facto Standard or a Free-for-all? A Benchmark for Reading SMILES.
https://github.com/rdkit/UGM_2018/blob/master/Presentations/OBoyle-SMILESBenchmark.pdf

IUPAC SMILES+ Project

A formalized recommended up-to-date open specification of the SMILES format that articulates standard interpretation of SMILES.

Primary goal is documentation that facilitates:

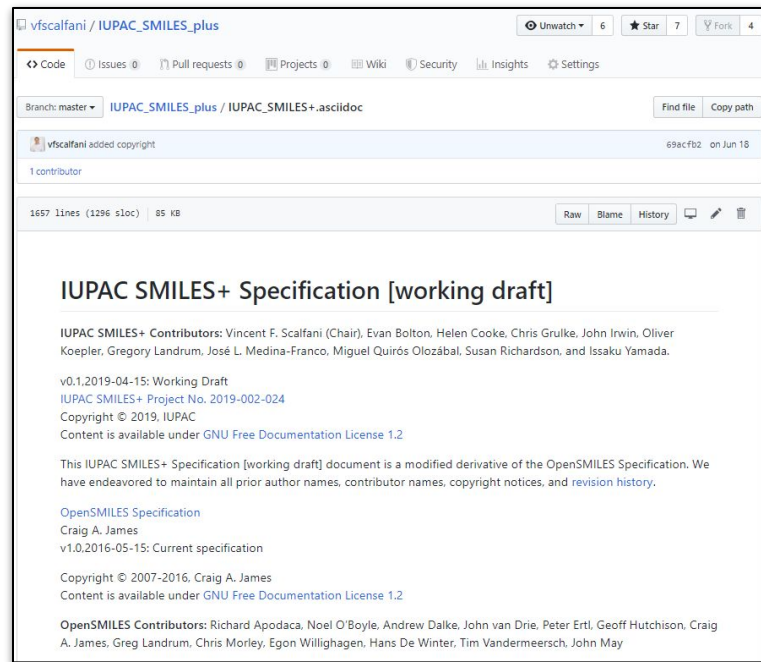
1. Consistent *reading* of SMILES between toolkits
2. Mechanism for community “approved” edits and extensions
3. A validation suite to test compatibility and show what a set of SMILES “means”

Project Phases of IUPAC SMILES+

- Phase 1** Establish dedicated communication channels with stakeholders
- Phase 2** Collect SMILES documentation and use cases. Start from OpenSMILES
- Phase 3** Identify SMILES edge cases where there are different toolkit interpretations and use this data to identify ambiguities within SMILES
- Phase 4** Write version 1 of IUPAC SMILES+ (w/lots of community input)
- Phase 5** Discuss implementation of IUPAC SMILES+ with toolkit developers (throughout)
- Phase 6** Outline an ongoing maintenance procedure with IUPAC and community

Progress: GitHub Repository for Working Docs

- Open workflow on GitHub for the IUPAC SMILES+ project.
- Made a copy of the OpenSMILES documentation to start from.
- Anyone can open a new “Issue”, comment, or Pull Request to suggest a change as work progresses.



https://github.com/vpscalfani/IUPAC_SMILES_plus

Progress: Survey of Toolkit Docs

Survey of 10
toolkit docs:

Stereochemistry

**Aromaticity
models**

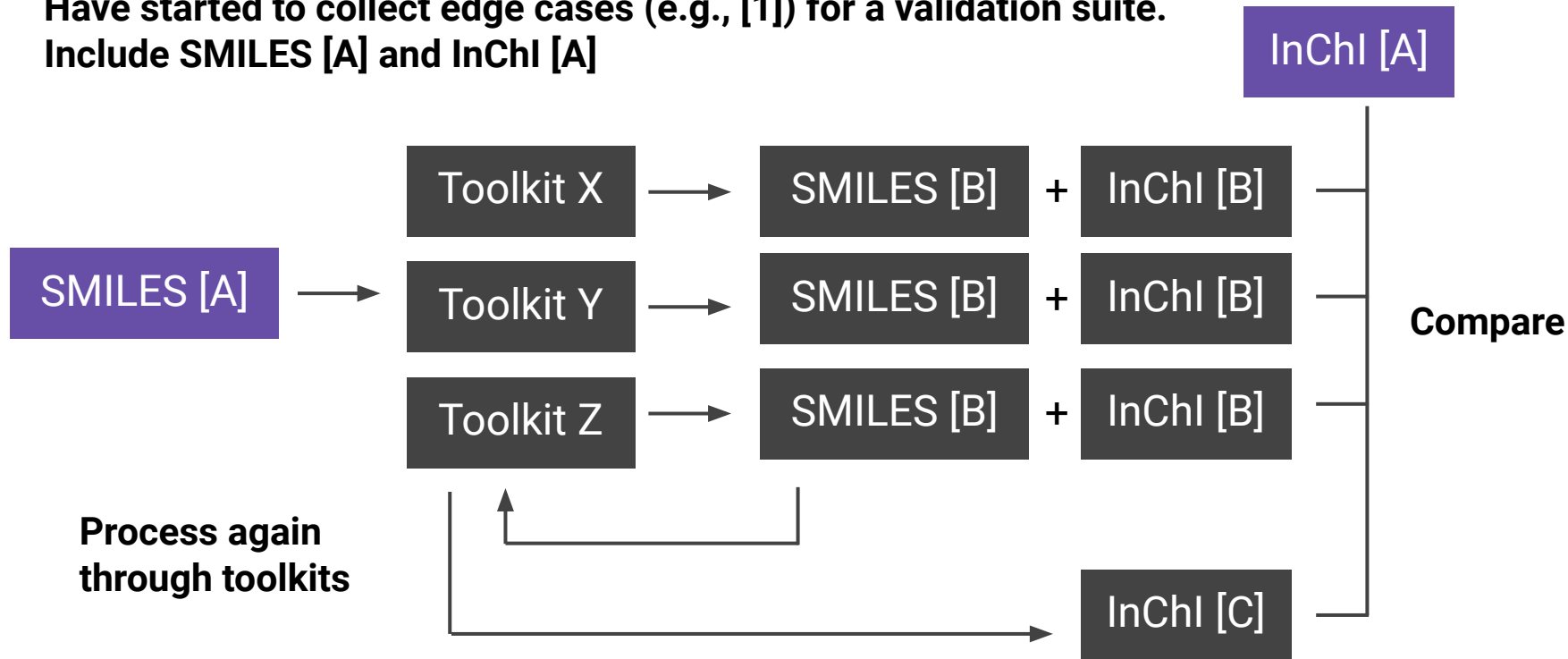
Extensions

The image shows a stack of four small comparison tables, one for each toolkit. Each table has the same columns: Toolkit, CXSMMILES, R Groups [Z] or [R], [te], Quadruple Bond \$, and Ring Closures > 100 (% (nnn)). The tables are partially overlapping, showing the top portion of each.

Toolkit	CXSMMILES	R Groups [Z] or [R]	[te]	Quadruple Bond \$	Ring Closures > 100 (% (nnn))
CACTVS v3.4.8.3	-	✓	✓	-	-
CDK v2.2	✓	✓	✓	-	-
ChemAxon 2019	✓	✓	-	-	-
OEChem 2.2.0	-	✓	✓	✓	-
Open Babel v3.0.0rc1	-	-	✓	✓	✓
RDKit v2019.03.1	✓	-	✓	-	✓

Progress: Edge Cases for a Validation Suite

Have started to collect edge cases (e.g., [1]) for a validation suite.
Include SMILES [A] and InChI [A]



[1] github.com/nextmovesoftware/smilesreading

Other Outputs in Near Future...

1. A FAQ and project overview in *Chemistry International*
2. Technical report outlining complementary use cases of SMILES and InChI (aiming to submit to *Pure And Applied Chemistry*)
3. Start editing IUPAC SMILES+ specification document

IUPAC SMILES+ Team

Vincent F. Scalfani (Chair), University of Alabama

Evan Bolton, NIH/NLM/NCBI

Chris Grulke, EPA

Gregory Landrum, KNIME AG

Susan Richardson, Royal Society of Chemistry

José L. Medina-Franco, Universidad Nacional
Autónoma de México

Helen Cooke, RSC CICAG Committee Member

Issaku Yamada, The Noguchi Institute

Miguel Quirós Olozabal, Universidad de Granada

John Irwin, University of California San Francisco;

Oliver Koepler, German National Library of Science
and Technology



...and the community!

Acknowledgements

- IUPAC
- IUPAC SMILES+ Team (see previous slide)
- InChI Community
- All cheminformatics toolkit developers and contributors [1]
- The University of Alabama Libraries

[1] It is a lot of fun using these wonderful tools, and we benefit from them everyday!

Contact:

Vincent F. Scalfani

The University of Alabama

vfscalfani@ua.edu

IUPAC Project: [2019-002-2-024](#)

GitHub Link: https://github.com/vfscalfani/IUPAC_SMILES_plus

IUPAC SMILES+ Breakout Session Topics

1. Your initial questions and feedback
2. SMILES and InChI complementary use cases
3. Prioritizing SMILES extensions
4. How to handle Daylight decisions (e.g. valence, aromaticity).
5. Validation suite specifications
6. What can we learn from InChI to help IUPAC SMILES+?
7. What can IUPAC SMILES+ deliver for InChI?

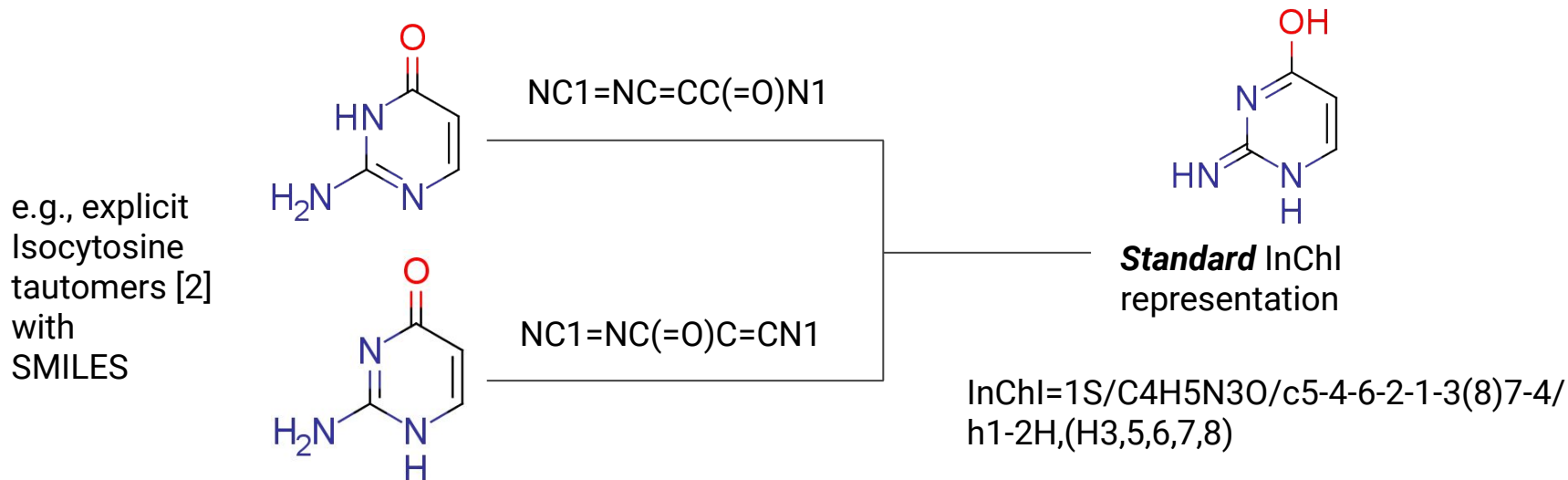
IUPAC SMILES+ Breakout Session

Your initial questions and feedback

...(e.g., anything you hope to discuss in this session?)

SMILES/InChI Use Case 1: Tautomers

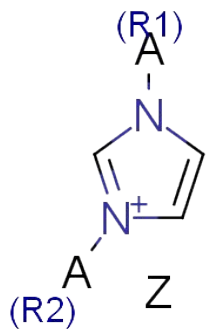
InChI provides a quick way to identify tautomers using Standard InChI. This can be more difficult to handle with SMILES [1].



[1] New software can help: NextMove Software MolHash: github.com/nextmovesoftware/molhash

[2] Milletti, F. et al. *J. Chem. Inf. Model.* **2010**, *50*, 1062.

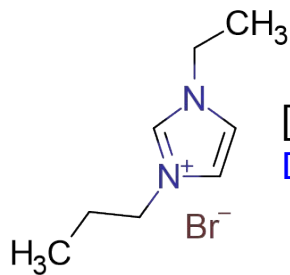
SMILES/InChI Use Case 2: Enumerate/Deduplication with InChI



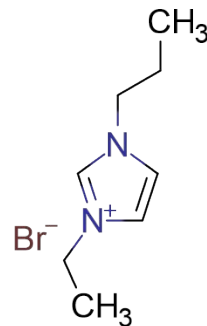
Direct concatenation[1]: [Z].[R1]N1C=C[N+](R2)=C1

Ring Closure notation[2]: [Z].N(%90)1C=C[N+](%91)=C1.[R1]%90.[R2]%91

Straightforward to combine SMILES strings programmatically and create libraries. InChI is then incredibly useful to quickly remove duplicates (e.g., mesomeric structures)



[Br-].CCN1C=C[N+](CCC)=C1
DWBYGJUDBYCOIN-UHFFFAOYSA-M

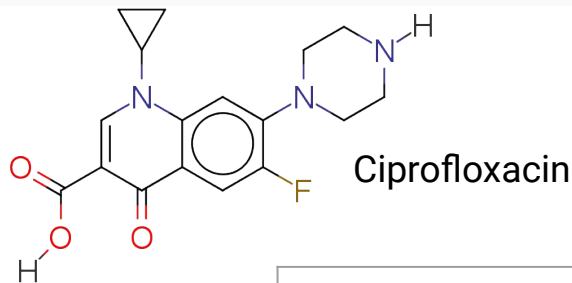


[Br-].CCCN1C=C[N+](CC)=C1
DWBYGJUDBYCOIN-UHFFFAOYSA-M

[1] Scalfani, V.F. et al. *Ind. Eng. Chem. Res.* **2018**, 57, 15971

[2] See Andrew Dalke's Blog: [Combinatorial Library Generation with SMILES](#)

SMILES/InChI Use Case 3: Database Linking



Canonical SMILES are toolkit dependent for comparison [1], standard InChI is not (i.e., fine to process SMILES locally, but can't reliability link databases with SMILES)

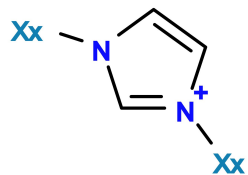
Toolkit	Canonical SMILES
ChemAxon 18.1	<chem>OC(=O)C1=CN(C2CC2)c2cc(N3CCNCC3)c(F)cc2C1=O</chem>
Open Babel 2.4.1	<chem>Fc1cc2c(cc1N1CCNCC1)n(cc(c2=O)C(=O)O)C1CC1</chem>
CACTVS (via CIR)	<chem>OC(=O)C1=CN(C2CC2)c3cc(N4CCNCC4)c(F)cc3C1=O</chem>
RDKit (v2019.03.2)	<chem>O=C(O)c1cn(C2CC2)c2cc(N3CCNCC3)c(F)cc2c1=O</chem>

InChI=1S/C17H18FN3O3/c18-13-7-11-14(8-15(13)20-5-3-19-4-6-20)21(10-1-2-10)9-12(16(11)22)17(23)24/h7-10,19H,1-6H2,(H,23,24)

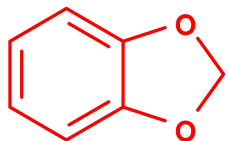
[1] Exception w/ Universal SMILES: O'Boyle, N.M. *Journal of Cheminformatics* **2012**, 4:22. [DOI: 10.1186/1758-2946-4-22](https://doi.org/10.1186/1758-2946-4-22).

SMILES/Use Case 4: Substructure/variability and IK First Hash

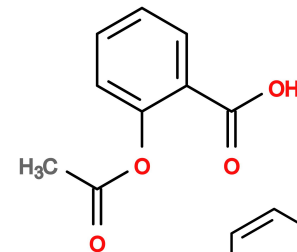
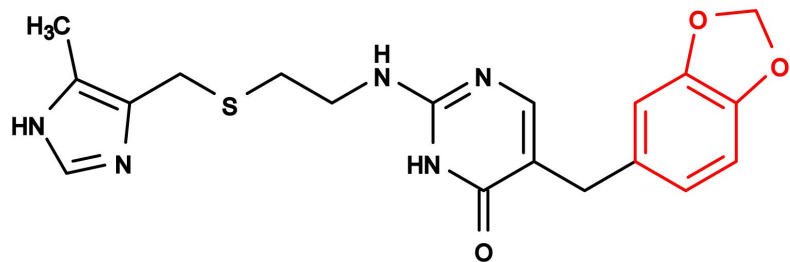
SMILES can handle variability and SMARTS substructure/pattern searching [1]. InChI is not designed for this, however a connectivity “skeleton” search is possible.



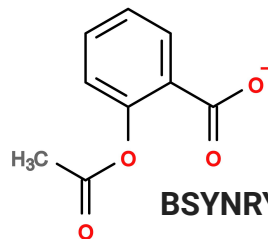
[*][N+]1=CN([*])C=C1



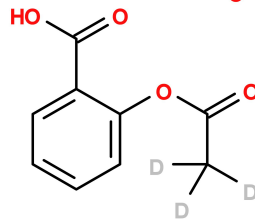
SMARTS pattern for Benzodioxole
c1cccc-2c1-[#8]-[#6]-[#8]-2



BSYNRYMUTXBXSQ-UHFFFAOYSA-N



BSYNRYMUTXBXSQ-UHFFFAOYSA-M



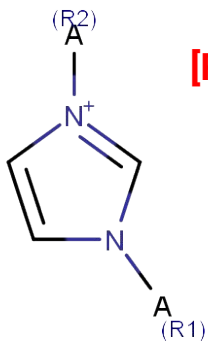
BSYNRYMUTXBXSQ-FIBGUPNXSA-N

[1] <https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html>

Discussion

Are there other high level complementary use cases we should be thinking about with SMILES and InChI??

SMILES Extension Notation Can Vary



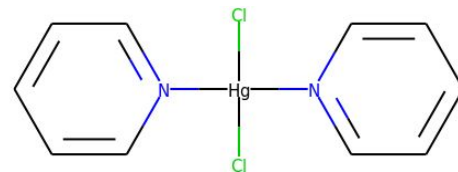
[R2][N+]1=CN([R1])C=C1

R groups can be one of the following depending on toolkit [1-3]

[R], [R1], [R2]

[Z]

&n



Dative bonds can be either [1,4]
-> **and** <-

|

Cl[Hg]23Cl.c1ccn->2cc1.c1ccn->3cc1

Cl[Hg]23Cl.c1ccn|2cc1.c1ccn|3cc1

[1] docs.chemaxon.com/display/docs/SMILES

[2] [CDK 2.2 API](https://cdk.github.io/CDK/2.2/API/)

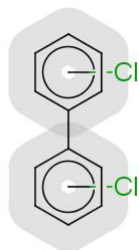
[3] docs.eyesopen.com/toolkits/python/ochemtk/SMILES.html

[4] https://www.rdkit.org/docs/RDKit_Book.html#dative-bonds

Extension notation is not always interoperable. It would be great if supported extensions were standardized.

SMILES Extensions

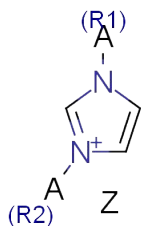
Several SMILES extensions (beyond Daylight spec) are already well adopted [1]:



CXSMILES, SMILES_String |<feature1>,<feature2>,...|

Example multicenter S-group: Cl*.Cl*.c1ccc(cc1)-c1ccccc1 |m:1:6.5.4.9.8.7,3:10.11.12.13.14.15|

Supported in 4 toolkits



R Group notation, [Z] or [R]

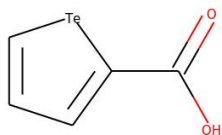
Example: [Z].[R1]N1C=C[N+](A)([R2])=C1

Supported in 5 toolkits

Quadruple Bonds \$ (in OpenSMILES spec)

Example: [Rh-](Cl)(Cl)(Cl)(Cl)\$[Rh-](Cl)(Cl)(Cl)Cl

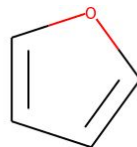
Supported in 4 toolkits



Aromatic [te]

Example: OC(=O)c1[te]ccc1

Supported in 6 toolkits



Ring Closures > 100, %(nnn)

Example: c%(1000)occc%(1000)

Supported in 3 toolkits

[1] [Toolkit Doc Comparisons](#)

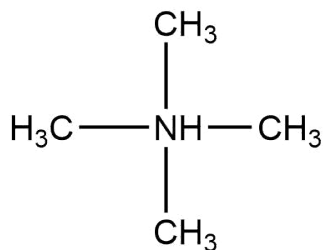
Discussion

1. Should “well-adopted” SMILES extensions be part of a core IUPAC SMILES+ specification?
2. If so, what criteria should we use for adoption into a core specification?

Daylight Decisions...

How should IUPAC SMILES+ approach Daylight decisions? Should it always be how Daylight handled it (to the best of our knowledge)?

Example with Nitrogen valence [1]:



N(C)(C)(C)C

WeiningerCEX_132 toolkit says HN(CH₃)₄.

Some other toolkits disagree or reject for bad valence.

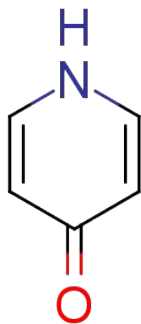
Both Daylight theory manual and OpenSMILES specify 3 or 5 valence for N, so it is correct based on the specification.

...Do we continue these choices?

[1] O'Boyle, N.M.; Mayfield, J. W.; Sayle, R. A. A De Facto Standard or a Free-for-all? A Benchmark for Reading SMILES.
https://github.com/rdkit/UGM_2018/blob/master/Presentations/OBoyle-SMILESBenchmark.pdf

Aromaticity

Different algorithms for aromaticity perception. Consider 4-pyridone:



O=C1C=CN=C1

Toolkit	Aromatic?
DayLight [1]	yes
OpenEye [1]	yes
MDL [1]	no
Tripos [1]	no
ChemAxon Basic	no
ChemAxon General	yes
RDKit Default	yes

How to handle in a specification where we want to maximize interoperability?

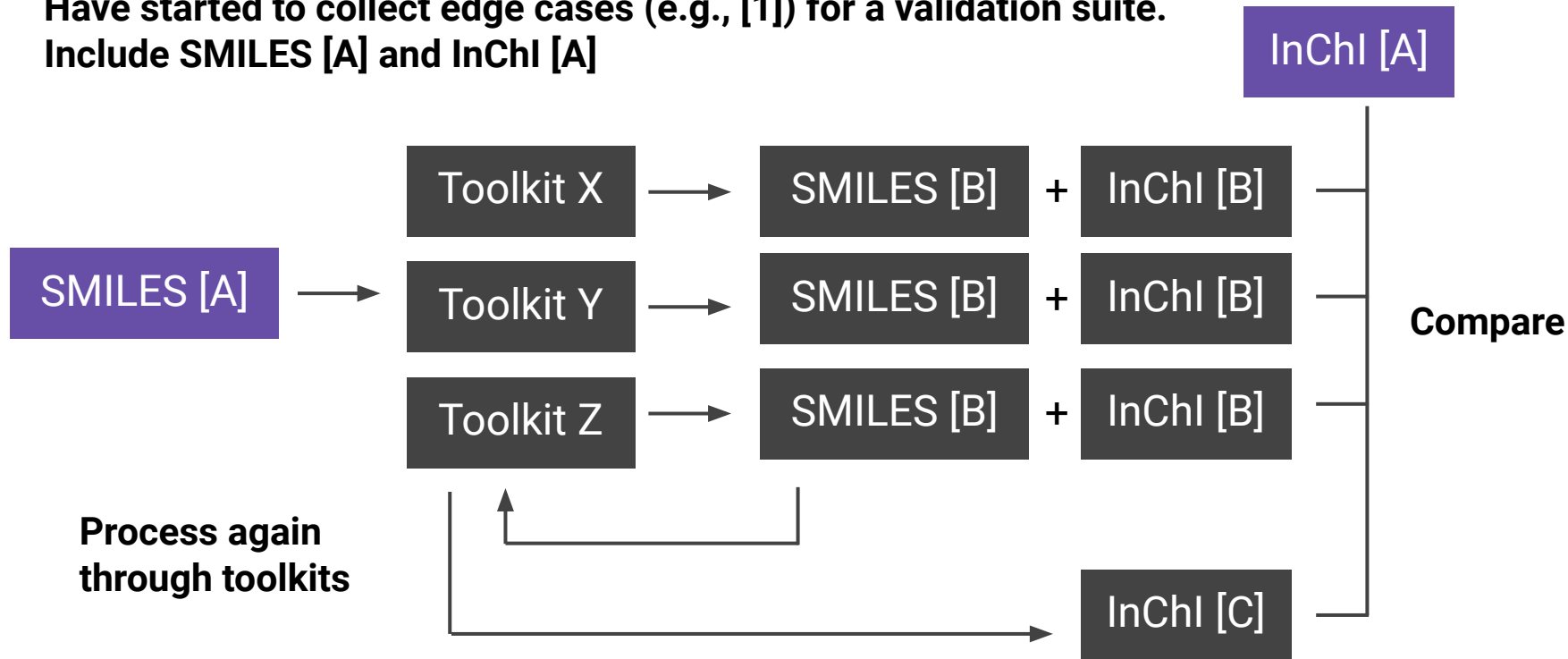
Kekule SMILES - (O=C1C=CN=C1) More interoperable, good representation of chemical compound.

Aromatic SMILES - (O=c1cc[nH]cc1) Better if consistent aromatic assignment is desired. Good representation of molecular graph allowing downstream processing [2].

OpenSMILES specifies the aromatic form is preferred, is this what is best?

What Should a Validation Suite Look Like: Start with InChI...

Have started to collect edge cases (e.g., [1]) for a validation suite.
Include SMILES [A] and InChI [A]



[1] github.com/nextmovesoftware/smilesreading

Do we need a specific “Validation only” Format?

For example, something that can tell us exactly what the SMILES string “means” in a lossless format. Two ways:

1. SMILES → **JSON** (e.g., [1])
2. **SMILES** → Depiction/ image dataset

What key requirements should we think about for a useful SMILES validation suite?

```
{
  "id": "CID6324",
  "name": "ethane",
  "atoms": [
    {"z": 6, "impHs": 3},
    {"z": 6, "impHs": 3}
  ],
  "bonds": [
    {"type": 1, "atoms": [0, 1]}
  ]
}
```

CommonChem JSON [1].

[1] github.com/CommonChem/CommonChem

What Can we Learn from InChI?

Can IUPAC SMILES+ borrow ideas from InChI?

Example, mark the notation? [1]:

IUPAC_SMILES+/1S=c1ccccc1

Or (tab) after SMILES:

c1ccccc1 IUPAC_SMILES+/1S

Maybe even specify the
toolkit/aromaticity model used?

What other lessons from InChI should we consider?

[1] originally proposed by Greg Landrum in 2007: <https://sourceforge.net/p/blueobelisk/mailman/message/843245/>

Conversely, what can IUPAC SMILES+ deliver for InChI?

1. Do we need a direct SMILES input ----> InChI conversion in InChI software?
Could this extend use of InChI?
2. What outcomes from the IUPAC SMILES+ project may help further advance InChI?

(e.g., using InChI as a validation tool extends utility of InChI)

GitHub setup?

Thanks for the discussion!!!

IUPAC SMILES+ Team

Vincent F. Scalfani (Chair), University of Alabama

Evan Bolton, NIH/NLM/NCBI

Chris Grulke, EPA

Gregory Landrum, KNIME AG

Susan Richardson, Royal Society of Chemistry

José L. Medina-Franco, Universidad Nacional
Autónoma de México

Helen Cooke, RSC CICAG Committee Member

Issaku Yamada, The Noguchi Institute

Miguel Quirós Olozabal, Universidad de Granada

John Irwin, University of California San Francisco;

Oliver Koepler, German National Library of Science
and Technology



...and the community!

Acknowledgements

- IUPAC
- IUPAC SMILES+ Team (see previous slide)
- InChI Community
- All cheminformatics toolkit developers and contributors [1]
- The University of Alabama Libraries

[1] It is a lot of fun using these wonderful tools, and we benefit from them everyday!

Contact:

Vincent F. Scalfani

The University of Alabama

vfscalfani@ua.edu

IUPAC Project: [2019-002-2-024](#)

GitHub Link: https://github.com/vfscalfani/IUPAC_SMILES_plus