IUPAC SMILES+ specification: Proposed community effort to advance interoperability of the SMILES chemical structure representation

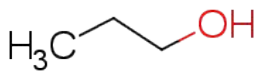
August 26, 2019
258th ACS National Meeting
San Diego, CA

Vincent F. Scalfani, Leah R. McEwen, Christopher Grulke, Evan Bolton, Gregory Landrum, Helen Cooke, Issaku Yamada, John J. Irwin, Jose L. Medina-Franco, Miguel Quirós Olozábal, Oliver Koepler, Susan Richardson

IUPAC Project: 2019-002-2-024
GitHub Repository: github.com/IUPAC/IUPAC_SMILES_plus
Contact: Vincent F. Scalfani, The University of Alabama, vfscalfani@ua.edu

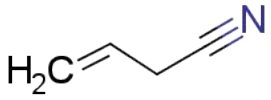
SMILES

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



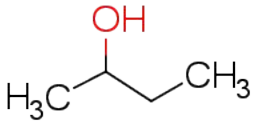
CCCO

1. atomic symbols



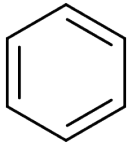
C=CCC#N

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



CCC(C)O

3. branching uses parentheses



C1=CC=CC=C1

4. ring closures use digits

<u>SMILES are human-friendly (and machine processable) molecular structure representations.</u> Since 1988, Daylight Chemical Information Systems have developed SMILES [3].

[1] Weininger, D. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31-36. DOI: 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/

InChI [1]

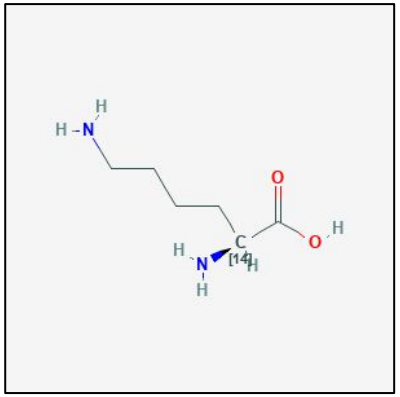
InChI – IUPAC **In**ternational **Ch**emical **I**dentifier [2]. Algorithm normalizes chemical representation. It is an open IUPAC standard and widely used.

InChI=1S/C6H14N2O2/c7-4-2-1-3-5(8)6(9)10/h5H,1-4,7-8H2,(H,9,10)/t5-/m0/s1/i5+2

- InChI is a line notation with layers.
- InChI is designed for machines and information exchange.
- InChIKey is a “hashed” InChI:

KDXKERNSBIXSRK-YDUYVQCESA-N

Version Type
Chemical formula
Connectivity
Charge & proton
Stereochemical
Other (e.g., isotopic)



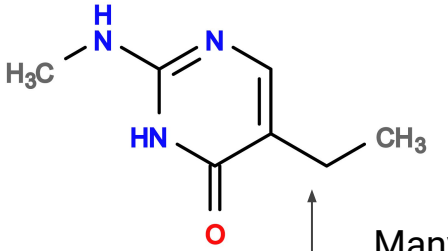
<u>InChI is a machine friendly molecular structure identifier.</u>

[1] *Acknowledgement:* slide adapted from Evan Bolton; [2] Heller et al. *Journal of Cheminformatics*, **2015**, 7:23. DOI: 10.1186/s13321-015-0068-4

SMILES vs. InChI? No, SMILES and InChI

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

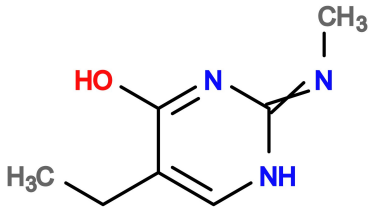
1. InChI is a machine descriptor identifier, powerful at linking information [1]. SMILES are difficult to link [2], but more closely tied to human (chemist) representation [3].



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).

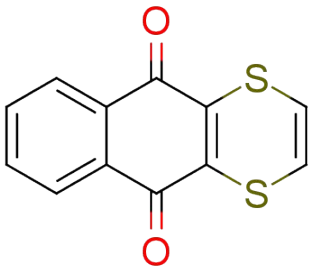
```
InChI=1S/C7H11N3O/c1-3-5-4-9-7(8-2)10-6(5)11/
h4H,3H2,1-2H3,(H2,8,9,10,11)
```


[1] Heller et al. *Journal of Cheminformatics*, **2015**, 7:23. DOI: 10.1186/s13321-015-0068-4
[2] Exception: O'Boyle, N.M. *Journal of Cheminformatics* **2012**, 4:22. DOI: 10.1186/1758-2946-4-22.
[3] Weininger, D. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31-36. DOI: 10.1021/ci00057a005

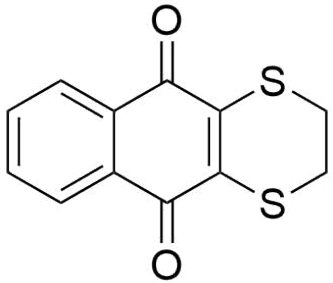
SMILES vs. InChI? No, SMILES and InChI

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
JEBDOQPBSPBGAP-UHFFFAOYSA-N

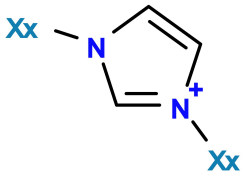
ChemDraw 18.1
IVQJELKILULDFK-UHFFFAOYSA-N

Example SMILES (#32408) from: github.com/nextmovesoftware/smilesreading

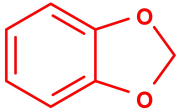
SMILES vs. InChI? No, SMILES and InChI

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

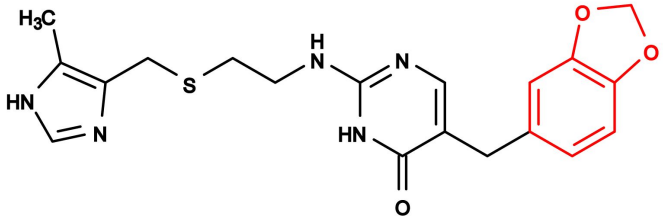
3. SMARTS (a superset of SMILES) substructure/pattern searching [1]. InChI is not designed for this, however a connectivity “skeleton” search is possible with IK hash.

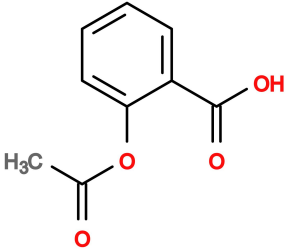


[*][N+]1=CN([*])C=C1

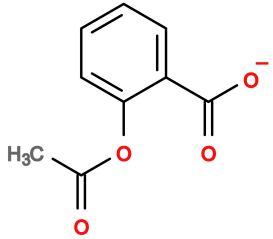


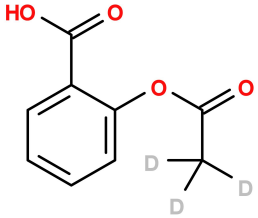
SMARTS pattern for Benzodioxole
c1cccc-2c1-[#8]-[#6]-[#8]-2





BSYNRYMUTXBXSQ-UHFFFAOYSA-N





BSYNRYMUTXBXSQ-UHFFFAOYSA-M

BSYNRYMUTXBXSQ-FIBGUPNXSA-N

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

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

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 Lavoiser, 1787

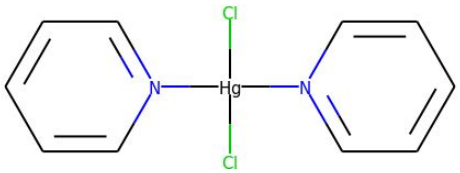
1.1. Purpose

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:

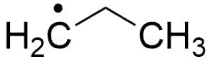


Cl[Hg]23Cl.c1ccn->2cc1.c1ccn->3cc1

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

c%(1000)occc%(1000)

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

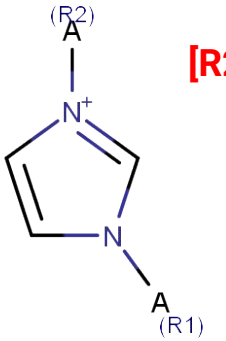


CCc

[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

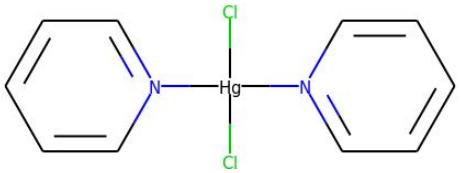
SMILES Extension Notation Can Vary



[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
[3] docs.eyesopen.com/toolkits/python/oechemtk/SMILES.html
[4] https://www.rdkit.org/docs/RDKit_Book.html#dative-bonds

Useful to document these all in one place so we can avoid conflicts.

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.

DISAGREEMENTS WITH SMILES VALENCE MODEL

Avalon	Cl2 Cl4 Br2 Br4 I2 I4
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

"Happy valence models are all alike; every unhappy valence model is unhappy in its own way."
...with apologies to Tolstoy

'9.5'/15 correct now.
When I started, it was 6/15.

** 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+

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”

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 Olozábal, Universidad de Granada
John Irwin, University of California San Francisco;
Oliver Koepler, German National Library of Science and Technology

...and the community!

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

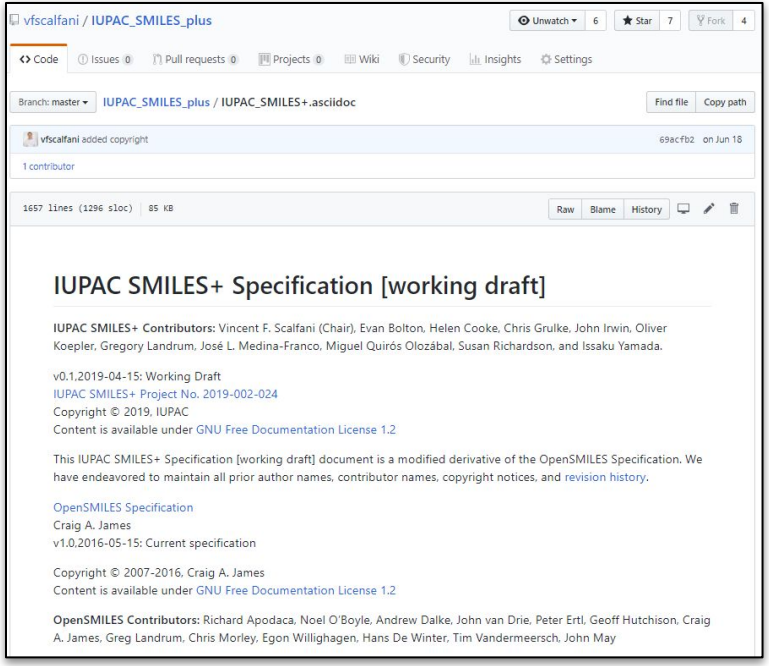
Phase 4 Write version 1 of IUPAC SMILES+ (w/lots of community input)

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 specification to start from.
- Anyone can open a new “Issue”, comment, or Pull Request to suggest a change as work progresses.



https://github.com/IUPAC/IUPAC_SMILES_plus

Progress: GitHub Repository for Working Docs

How we hope to engage the broader community.

Targeted “Issues” in GitHub to get feedback from community.

https://github.com/vfscalfani/IUPAC_SMILES_plus/issues/4

Should select SMILES extensions become part of the core IUPAC SMILES+ specification? #4

Edit | New issue

Open | vfscalfani opened this issue 4 days ago · 0 comments

vfscalfani commented 4 days ago | Owner

Based on a high-level survey of current SMILES extensions and support across toolkits (via their documentation), there are several SMILES extensions that are somewhat well-adopted already (i.e., >3 toolkits support them).

These extensions include: ChemAxon CXSMILES, R group notation `[Z]` or `[R]`, aromatic `[te]`, ring closures > 100 `%(nnn)`, and quadruple bonds `$`.

Note that the current IUPAC SMILES+ Specification [working draft] (based off OpenSMILES) already has support for quadruple bonds in the core specification.

So, a few questions to consider:

1. When does it make sense to incorporate a SMILES extension into the core specification, rather than leaving it as an add-on extension? What are the important considerations?
2. Are there any extensions such as those listed above that you would recommend becoming a part of the core specification?
3. If yes to question 2, would incorporating SMILES extensions limit toolkit adoption of an IUPAC SMILES+ specification? How do we lower the barrier here?

One important consideration is to avoid conflicting extension notation, and we'll have to be careful and thoughtful about this before incorporating anything into the core specification.

Vin

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications | Customize
Unsubscribe
You're receiving notifications because you're watching this repository.

1 participant

Lock conversation

https://github.com/IUPAC/IUPAC_SMILES_plus

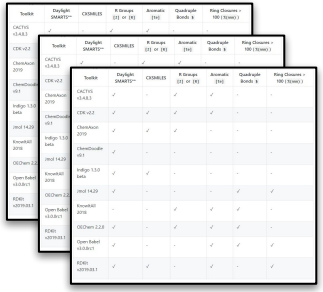
Progress: Survey of Toolkit Docs

Survey of 10 toolkit docs:

Stereochemistry

Aromaticity models

Extensions



Toolkit	CXSMILES	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	✓	-	✓	-	✓

IUPAC_SMILES_plus Toolkit Comparison

Progress: Collecting sets of SMILES for Validation Tests

Started to collect lists of SMILES for future validation tests [1-3]:

Aromatic SMILES

Kekule SMILES

Valance Model

Elements

Nonstandard SMILES

And more...

cdk_...

File Edit Format View Help

```
C1CN2C(=C)N(CC2C1)C 30
C1CCN2C1CN(C2=N)C 3064
N12C(=O)CC(C1)CC=C2 30
[n+]1(cccn1C)Cl 3066
O1CC2CCCC(C1)C2 3067
N1C(=O)CC=CC1=O 3068
C12N(CCO1)C(=O)C2 3069
C1=CC(=O)CCC1=C 3070
n1c[nH]c2c1cco2 3071
C1CN(CN1C)[SH5] 3072
c1cc(=O)oc(=O)[nH]1 30
C1N=C2N(CCCC2)CC1 3074
C1CCNc2cnnn21 3075
O1c2cscc2OCC1 3076
C1=CC(=O)C2CC1CC2 3077
c12c(COCC1)scc2 3078
C1=CN2C(CC2=O)CN1 3079
c1cc2c([nH]1)nco2 3080
c12c(scn1)[nH]cc2 3081
C12CCC(=N)CC1C2 3082
C1N=CC2C(C1)CCCC2 3083
C1C=NN(C2C1OCC2)C 3084
C1[N+](CCOC1)(C)C 3085
C1c2c(CCN1)nc[nH]2 308
C1c2n[nH]cc2CNC1 3087
```

smile...

File Edit Format View Help

```
B B0
BC B1
B(C)C B2
B(C)(C)C B3
B(C)(C)(C)C B4
C C0
CC C1
C(C)C C2
C(C)(C)C C3
C(C)(C)(C)C C4
C(C)(C)(C)(C)C C5
N N0
NC N1
N(C)C N2
N(C)(C)C N3
N(C)(C)(C)C N4
N(C)(C)(C)(C)C N5
N(C)(C)(C)(C)(C)C N6
O O0
OC O1
O(C)C O2
O(C)(C)C O3
P P0
PC P1
P(C)C P2
```

tetr...

File Edit Format View Help

```
Br[C@@](Cl)(F)I 1
Br[C@@](F)(I)Cl 1
Br[C@@](I)(Cl)F 1
Br[C@](Cl)(I)F 1
Br[C@](F)(Cl)I 1
Br[C@](I)(F)Cl 1
[C@@](Br)(Cl)(F)I 1
[C@](Br)(Cl)(I)F 1
[C@](Br)(F)(Cl)I 1
[C@@](Br)(F)(I)Cl 1
[C@@](Br)(I)(Cl)F 1
[C@](Br)(I)(F)Cl 1
[C@](Cl)(Br)(F)I 1
[C@@](Cl)(Br)(I)F 1
[C@@](Cl)(F)(Br)I 1
[C@](Cl)(F)(I)Br 1
[C@](Cl)(I)(Br)F 1
[C@@](Cl)(I)(F)Br 1
[C@@](F)(Br)(Cl)I 1
[C@](F)(Br)(I)Cl 1
[C@](F)(Cl)(Br)I 1
[C@@](F)(Cl)(I)Br 1
[C@@](F)(I)(Br)Cl 1
[C@](F)(I)(Cl)Br 1
[C@](I)(Br)(Cl)F 1
```

non...

File Edit Format View Help

```
C(/Br)=C\\I
C((C))O
C((C))O
(N1CCCC1)
[Na+]..[Cl-]
.CCO
CCO.
C1CCC
C/C=C
C/C=CC
CC/=C/C
C/C(\F)=C/C
CccccC
Ccc
C1.C1
C%00CC%00
C(C.C)C
C(C)1CC1
C(.C)
C()
(CO)=O
(C)
.c
C..C
C.
```

File Edit Format View Help

```
[H]
[He]
[Li]
[Be]
[B]
[C]
[N]
[O]
[F]
[Ne]
[Na]
[Mg]
[Al]
[Si]
[P]
[S]
[Cl]
[Ar]
[K]
[Ca]
[Sc]
[Ti]
[V]
[Cr]
```

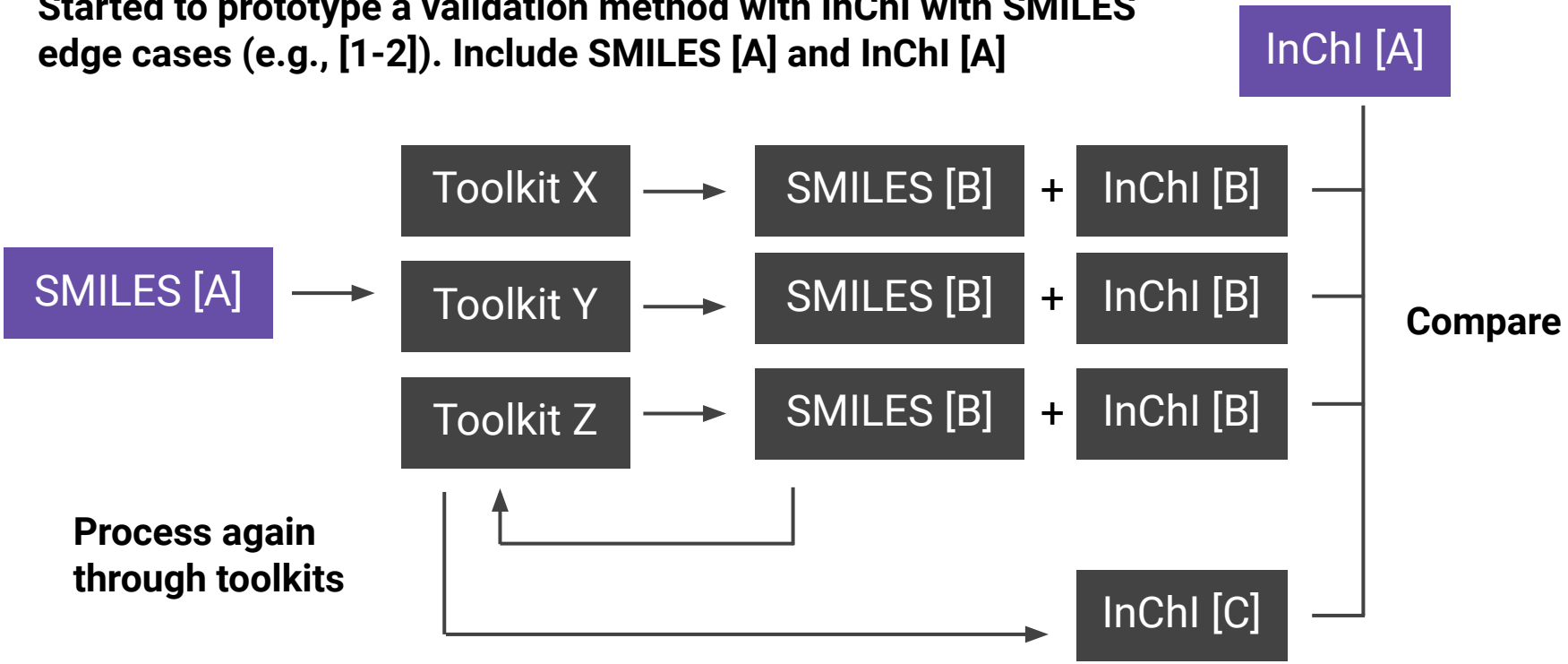

[1] https://github.com/nextmovesoftware/smilesreading

[2] https://sourceforge.net/p/blueobelisk/mailman/blueobelisk-smiles/

[3] https://docs.eyesopen.com/toolkits/python/oechemtk/SMILES.html#chapter-smiles

Progress: Validation Suite

Started to prototype a validation method with InChI with SMILES edge cases (e.g., [1-2]). Include SMILES [A] and InChI [A]



[1] github.com/nextmovesoftware/smilesreading
[2] https://sourceforge.net/p/blueobelisk/mailman/blueobelisk-smiles/

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

Conclusions

1. SMILES and InChI are complementary. We need both.

2. There is a need for a comprehensive up-to-date SMILES reference document.

3. InChI can help us validate SMILES and improve interoperability between toolkits. This further extends the utility of InChI.

4. Through IUPAC, and an open workflow, we hope to create an international SMILES specification that can be responsive to community needs and serve as the document of reference for SMILES.

Acknowledgements

- IUPAC
- IUPAC SMILES+ Team
- 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/IUPAC/IUPAC_SMILES_plus