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 – Simplified Molecular Input Line-Entry System [1]. Compact line notation for representing molecules and reactions. Four main rules [1-3]:

1. atomic symbols
2. double ‘=’, triple bonds ‘#’
3. branching uses parentheses
4. ring closures use digits

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

SMILES vs. InChl? No, SMILES and InChl

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.

   ![InChI Structure](image1)

   ![SMILES Structure](image2)

   Many valid SMILES

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

   ![InChI Normalization](image3)

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

   One Standard InChI

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)

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

\[
* \text{N}^+ = \text{CN}[^*])\text{C}=\text{C}1
\]

SMARTS pattern for Benzodioxole c1ccccc-2c1-[#8]-[#6]-[#8]-2

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.

Many SMILES Extensions Exist

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

\[
\text{Cl[Hg]23Cl.c1ccn->2cc1.c1ccn->3cc1} \quad \text{c%(1000)occc%(1000)} \quad \text{CCc}
\]

[1] RDKit dative bonds, -> and <-  
[2] Ring closure notation > 100, % (nnn). (Jmol, Open Babel, RDKit)  
[3] Open Babel radical centers via lowercase symbols

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

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/vfscalfani/IUPAC_SMILES_plus
## Progress: Survey of Toolkit Docs

Survey of 10 toolkit docs:

<table>
<thead>
<tr>
<th>Toolkit</th>
<th>CXSMILES</th>
<th>R Groups [Z] or [R]</th>
<th>[%e]</th>
<th>Quadruple Bond $</th>
<th>Ring Closures &gt; 100 (% (nnn))</th>
</tr>
</thead>
<tbody>
<tr>
<td>CACTVS v3.4.8.3</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CDK v2.2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ChemAxon 2019</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OEChem 2.2.0</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Open Babel v3.0.0rc1</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RDKit v2019.03.1</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Stereochemistry**

**Aromaticity models**

**Extensions**

[IUPAC_SMILES_plus Toolkit Comparison](#)
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].

Process again through toolkits

[1] github.com/nextmovesoftware/smilesreading
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 Olozábal, 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
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?
Your initial questions and feedback

...(e.g., anything you hope to discuss in this session?)
InChI provides a quick way to identify tautomers using Standard InChI. This can be more difficult to handle with SMILES [1].

![Diagram of tautomers](image)

**e.g., explicit Isocytosine tautomers [2]**

with SMILES

---


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)

SMILES/InChI Use Case 3: Database Linking

![Ciprofloxacin](image)

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)

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

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)

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

SMARTS pattern for Benzodioxole:
\[
\text{c1cccc-2c1-}\left[\#8\right]-\left[\#6\right]-\left[\#8\right]-2
\]

SMARTS pattern for Benzodioxole:
\[
\left[\text{N}\right]=\text{CN}\left(\text{[*]}\right)\text{C}=\text{C}1
\]
Are there other high level complementary use cases we should be thinking about with SMILES and InChI??
SMILES Extension Notation Can Vary

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

[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

Extension notation is not always interoperable. It would be great if supported extensions were standardized.
Several SMILES extensions (beyond Daylight spec) are already well adopted [1]:

**CXSMILES, SMILES_String |<feature1>,<feature2>,...|**
Example multicenter S-group: \(\text{Cl}^*\cdot\text{Cl}^*\cdot\text{c}1\text{ccc(cc1)-c1cccccc1 m:1:6.5.4.9.8.7,3:10.11.12.13.14.15}\)
Supported in 4 toolkits

**R Group notation, [Z] or [R]**
Example: \([Z].[R1]\text{N}1\text{C}=\text{C}[\text{N+}](\text{[R2]})=\text{C}1\)
Supported in 5 toolkits

**Aromatic [te]**
Example: \(\text{OC}(=\text{O})\text{c}1\text{[te]ccc1}\)
Supported in 6 toolkits

**Quadruple Bonds $ (in OpenSMILES spec)**
Example: \([\text{Rh-}](\text{Cl})(\text{Cl})(\text{Cl})(\text{Cl})\$[\text{Rh-}](\text{Cl})(\text{Cl})(\text{Cl})\text{Cl}\)
Supported in 4 toolkits

**Ring Closures > 100, % (nnn)**
Example: \(\text{c}%(1000)\text{occc%(1000)}\)
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]:

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?

Aromaticity

Different algorithms for aromaticity perception. Consider 4-pyridone:

<table>
<thead>
<tr>
<th>Toolkit</th>
<th>Aromatic?</th>
</tr>
</thead>
<tbody>
<tr>
<td>DayLight [1]</td>
<td>yes</td>
</tr>
<tr>
<td>OpenEye [1]</td>
<td>yes</td>
</tr>
<tr>
<td>MDL [1]</td>
<td>no</td>
</tr>
<tr>
<td>Tripos [1]</td>
<td>no</td>
</tr>
<tr>
<td>ChemAxon Basic</td>
<td>no</td>
</tr>
<tr>
<td>ChemAxon General</td>
<td>yes</td>
</tr>
<tr>
<td>RDKit Default</td>
<td>yes</td>
</tr>
</tbody>
</table>

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

Kekule SMILES - (O=C1C=CNC=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?

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

Process again through toolkits

[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?

Can IUPAC SMILES+ borrow ideas from InChI?

Example, mark the notation? [1]:

IUPAC_SMILES+/1S=c1ccccc1

Or (tab) after SMILES:

c1ccccc1 IUPAC_SMILES+/1S

What other lessons from InChI should we consider?

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!!!
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!
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