InChI today: all the new stuff!

Gerd Blanke[1], Jonathan Goodman[2]

ACS Fall Meeting 2026

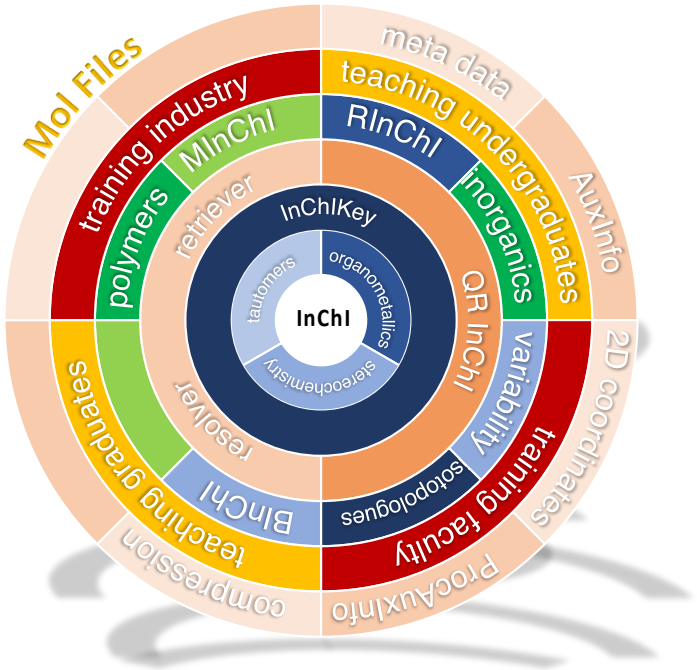
Chicago, Il

24 August 2026



1) StructurePendium Technologies GmbH, Germany
2) Yusuf Hamied Department of Chemistry, University of Cambridge, UK

Agenda



- InChI in the last 15 years
- Molecular inorganics
- MDL enhanced stereochemistry
- Polymers and Polymer InChI
- Application InChIs
- New InChI versioning schema
- InChI educational support
- InChI development and test environments

InChI in the last 15 years

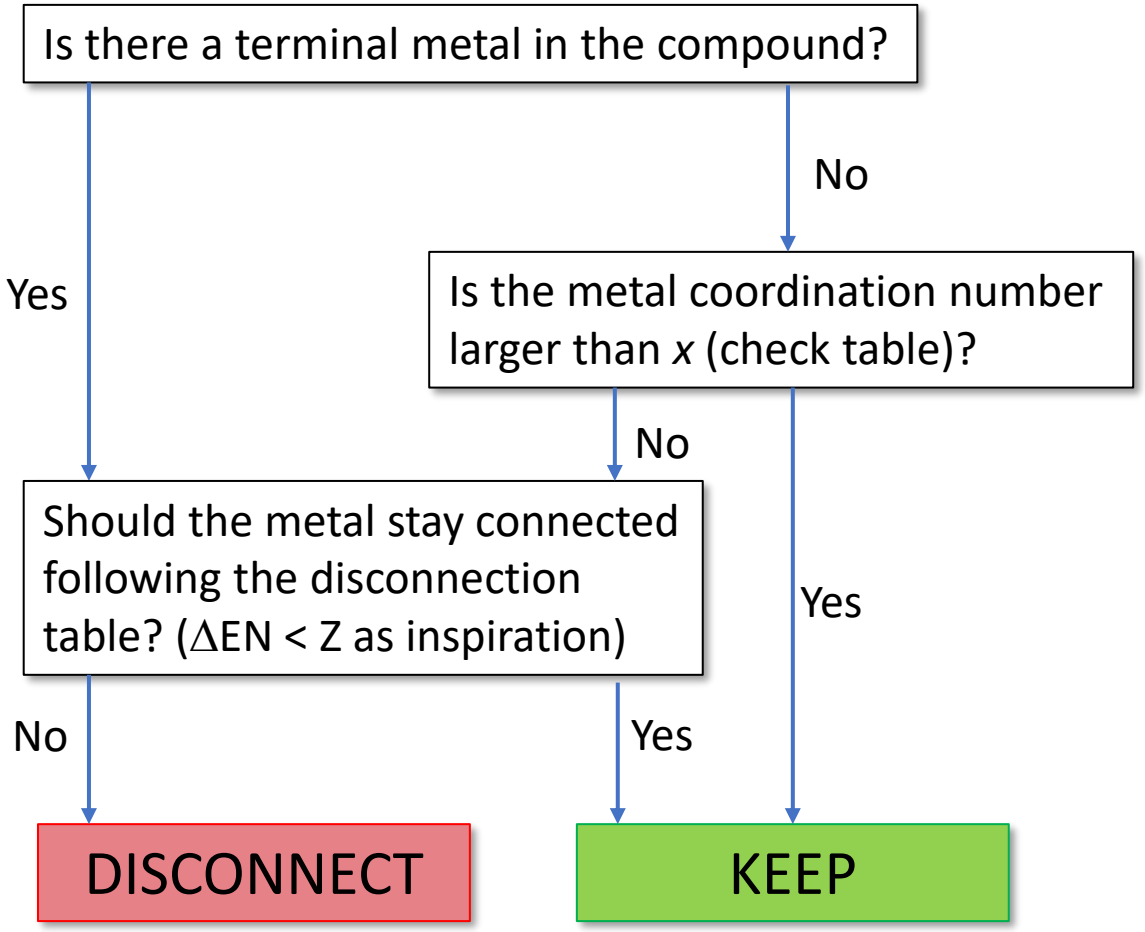
- **Pretty slow development in the 2010th after the release of version 1.04**
  - Version 1.05: introduction of polymers (2017)
    - Primarily focused to the most common polymer brackets (SRU, monomers, copolymers)
  - Version 1.06: resolving some of the issues of the very restrictive polymer implementation approaches of 1.05 (2020)
- **With Igor Pletnev's unexpected death caused by corona in 2021 the heard of the InChI development passed away.**
- **Re-set of the InChI development in 2022**
  - Versions 1.07.x: fix of more than 4,000 minor and major code issues
  - Work on molecular inorganics and the "MDL enhanced stereochemistry"
  - New test environment (based on PubChem substance level)
  - Additional tools for test support and for educational purposes

Molecular inorganics

Molecular inorganics

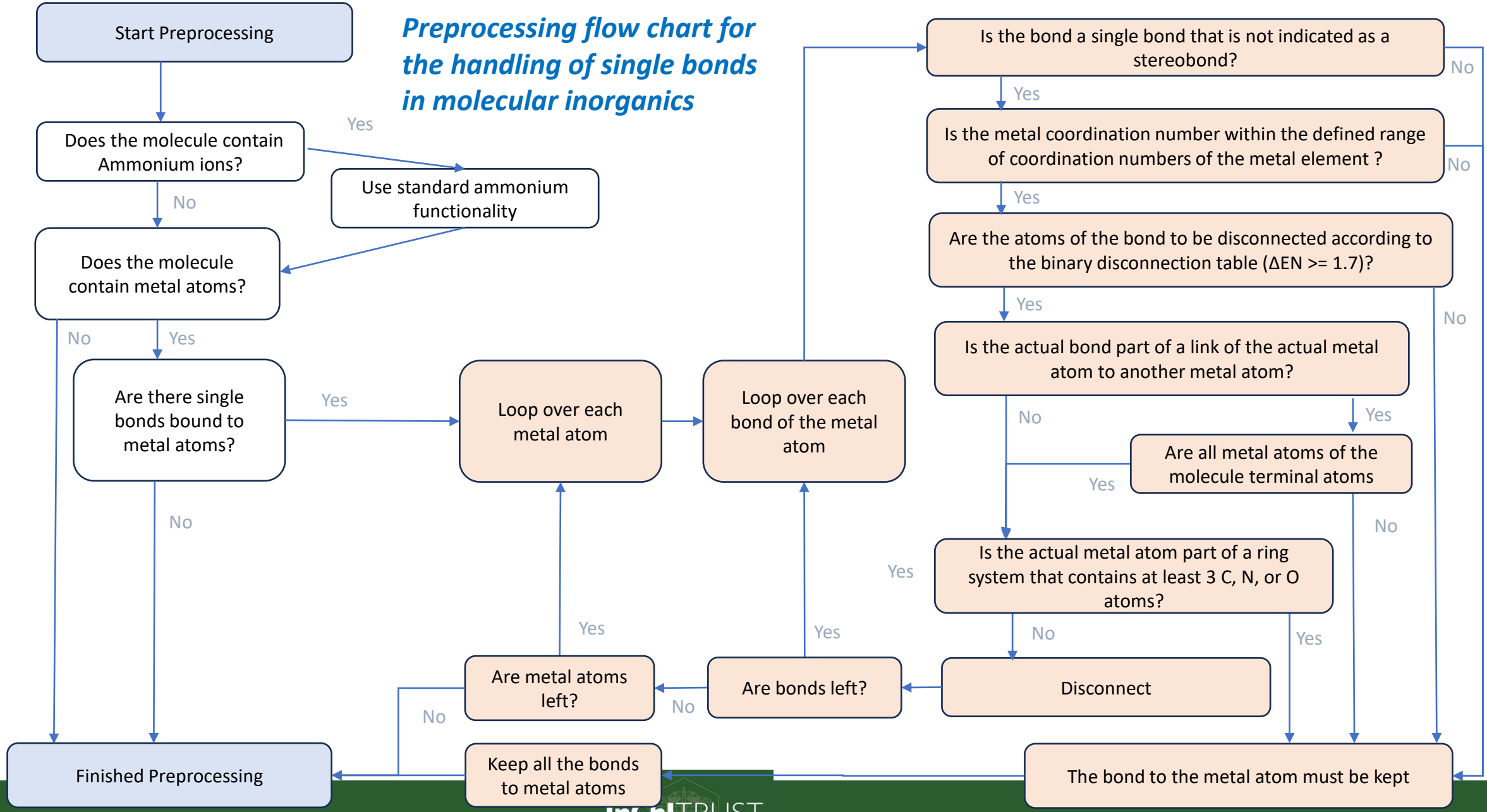
- **Up to standard InChI 1.07.5 bonds, pointing to a metal atom are deleted and the metal atom is isolated.**
  - The disconnection of metal bonds reduced the usability of InChIs in the scope of molecular inorganics.
- **Goals of the first release to enhance the usability of the InChI**
  - Rebuild disconnection rules in order to enhance the identification of molecular inorganic compounds
  - Introduce metal specific bond types
    - Coordinative/dative bonds (bond type 9)
    - Haptic bonds
  - Adapt default valences of the main group metal atoms of the periodic table.

Preprocessing flow chart for inorganic compounds



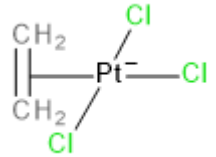
To be noted:

- This flow chart has to be run for every metal.
- The „Disconnection Table“ step has to be run for every bond to a metal.
- If atoms are not connected in the input, they stay disconnected. No new bonds are formed that are not already in the input.

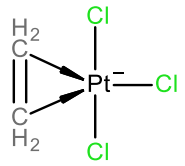
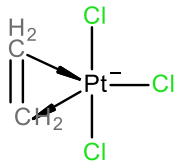
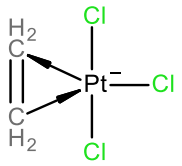
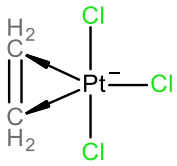
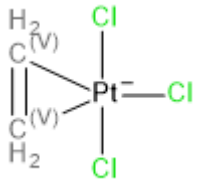
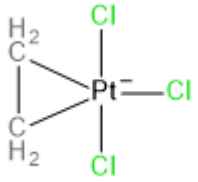

Preprocessing flow chart for the handling of single bonds in molecular inorganics
Start Preprocessing
Does the molecule contain Ammonium ions?
Yes
Use standard ammonium functionality
No
Does the molecule contain metal atoms?
No
Yes
Are there single bonds bound to metal atoms?
Yes
No
Loop over each metal atom
Loop over each bond of the metal atom
Is the bond a single bond that is not indicated as a stereobond?
No
Yes
Is the metal coordination number within the defined range of coordination numbers of the metal element ?
No
Yes
Are the atoms of the bond to be disconnected according to the binary disconnection table (ΔEN >= 1.7)?
No
Yes
Is the actual bond part of a link of the actual metal atom to another metal atom?
No
Yes
Are all metal atoms of the molecule terminal atoms
Yes
No
Is the actual metal atom part of a ring system that contains at least 3 C, N, or O atoms?
Yes
No
Yes
Disconnect
Are bonds left?
Yes
No
Are metal atoms left?
Yes
No
Finished Preprocessing
Keep all the bonds to metal atoms
The bond to the metal atom must be kept

Coordinative/dative bonds – example: Zeise salt

- **Zeise Salt (Ethylene perpendicular to $PtCl_3$ plane)**



- **Coordinative representations:**






and


deliver identical (new) InChIs

InChI (old)

```
InChI=1S/C2H4.3ClH.Pt/c1-2;;;;/h1-2H2;3*1H;/q;;;;+2/p-3
```

InChI (new)

```
InChI=1/C2H4Cl3Pt/c3-6(4,5)1-2-6/h1-2H2/q-1
```

```
ACCLDraw02102500192D

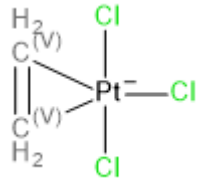
 0  0  0     0  0               999 V3000
M  V30 BEGIN CTAB
M  V30 COUNTS 6 6 0 0 0
M  V30 BEGIN ATOM
M  V30 1 Cl 3.6197 -9.1711 0 0
M  V30 2 Cl 4.8008 -7.99 0 0
M  V30 3 Cl 3.6197 -6.8089 0 0
M  V30 4 C 2.3258 -8.5637 0 0
M  V30 5 C 2.3258 -7.3826 0 0
M  V30 6 Pt 3.6197 -7.99 0 0
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 2 4 5
M  V30 2 9 5 6
M  V30 3 9 4 6
M  V30 4 1 6 1
M  V30 5 1 6 2
M  V30 6 1 6 3
M  V30 END BOND
M  V30 END CTAB
M  END
```

Bond type 9 -> 1

Increase valence at non metal atoms by +1

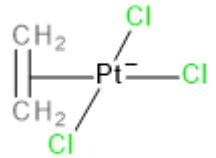
```
ACCLDraw02102500192D

 0  0  0     0  0               999 V3000
M  V30 BEGIN CTAB
M  V30 COUNTS 6 6 0 0 0
M  V30 BEGIN ATOM
M  V30 1 Cl 3.6197 -9.1711 0 0
M  V30 2 Cl 4.8008 -7.99 0 0
M  V30 3 Cl 3.6197 -6.8089 0 0
M  V30 4 C 2.3258 -8.5637 0 0 VAL=5
M  V30 5 C 2.3258 -7.3826 0 0 VAL=5
M  V30 6 Pt 3.6197 -7.99 0 0 CHG=-1
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 2 4 5
M  V30 2 1 5 6
M  V30 3 1 4 6
M  V30 4 1 6 1
M  V30 5 1 6 2
M  V30 6 1 6 3
M  V30 END BOND
M  V30 END CTAB
M  END
```

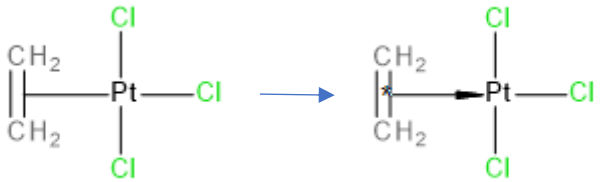


Haptic bonds – example: Zeise salt

- **Zeise Salt (Ethylene perpendicular to $PtCl_3$ plane)**



- **Haptic representation**



Displayed structure

Internal representation with * atom and coordinative bond

```
ACCLDraw03112521502D

 0  0  0     0  0            999 V3000
M  V30 BEGIN CTAB
M  V30 COUNTS 7 5 0 0 0
M  V30 BEGIN ATOM
M  V30 1 C 10.5938 -22.8781 0 0
M  V30 2 C 10.5938 -21.6969 0 0
M  V30 3 * 10.6875 -22.2875 0 0
M  V30 4 Pt 12.2104 -22.2594 0 0 CHG=-1
M  V30 5 Cl 12.2259 -23.4588 0 0
M  V30 6 Cl 13.5509 -22.2678 0 0
M  V30 7 Cl 12.1729 -21.0858 0 0
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 2 1 2
M  V30 2 9 4 3 ENDPTS=(2 2 1) ATTACH=ALL
M  V30 3 1 4 5
M  V30 4 1 4 6
M  V30 5 1 4 7
M  V30 END BOND
M  V30 END CTAB
M  END
```

Get # of * atom (here 3)
Get # of metal atom (here 4)
Get endpoints of coordinative bond between * and metal atom

- **Here (2 2 1): 2 end points with atom # = 1 and 2**
- **Increase valence of non-metal atoms by 1**
- **Renumber all atoms sequentially**

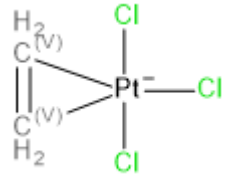
Build new single bonds between end point atoms and metal atom

- **Single bond between atom 1 and atom 4**
- **Single bond between atom 2 and atom 4**

Delete coordinative bond between atom 3 and 4
Delete * atom (here atom 3)
Renumber bonds sequentially
Adapt counts of atoms and bonds

```
ACCLDraw03112521502D

 0  0  0     0  0            999 V3000
M  V30 BEGIN CTAB
M  V30 COUNTS 6 6 0 0 0
M  V30 BEGIN ATOM
M  V30 1 C 10.5938 -22.8781 0 0 VAL=5
M  V30 2 C 10.5938 -21.6969 0 0 VAL=5
M  V30 3 Pt 12.2104 -22.2594 0 0 CHG=-1
M  V30 4 Cl 12.2259 -23.4588 0 0
M  V30 5 Cl 13.5509 -22.2678 0 0
M  V30 6 Cl 12.1729 -21.0858 0 0
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 2 1 2
M  V30 2 1 3 4
M  V30 3 1 3 5
M  V30 4 1 3 6
M  V30 5 1 1 3
M  V30 6 1 2 3
M  V30 END BOND
M  V30 END CTAB
M  END
```



Default valences

- **Each atom has one or more default valences. The number of implicit hydrogens at an atom is equal to the allowed valence minus the number of bonds to non-hydrogen atoms, up to the next allowed valence**.
  - For example, a sulfur atom with one bond to a non-hydrogen atom has one implicit hydrogen, and a sulfur atom with two bonds has zero implicit hydrogens, because the next highest valence is 2. A sulfur atom with three bonds has one implicit hydrogen, because the next highest valence is 4.

The table that follows shows allowed default valences for neutral main group elements:

1a		2a		3a		4a		5a		6a		7a		8	
H	(1)													He	(0)
Li	(1)	Be	(2)	B	(3)	C	(4)	N	(3,5)	O	(2)	F	(1)	Ne	(0)
Na	(1)	Mg	(2)	Al	(3)	Si	(4)	P	(3,5)	S	(2,4,6)	Cl	(1,3,5,7)	Ar	(0)
K	(1)	Ca	(2)	Ga	(3)	Ge	(4)	As	(3,5)	Se	(2,4,6)	Br	(1,3,5,7)	Kr	(0)
Rb	(1)	Sr	(2)	In	(3)	Sn	(2,4)	Sb	(3,5)	Te	(2,4,6)	I	(1,3,5,7)	Xe	(0)
Cs	(1)	Ba	(2)	Tl	(1,3)	Pb	(2,4)	Bi	(3,5)	Po	(2,4,6)	At	(1,3,5,7)	Rn	(0)
Fr	(1)	Ra	(2)												

MDL Chemical Representation 2007

For transition metals, lanthanides, and actinides, any valence is allowed. Consequently, these atoms do not display implicit hydrogens unless you specify an explicit valence

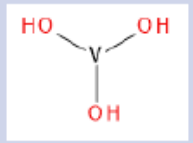
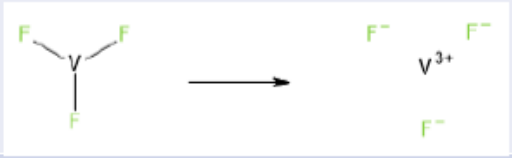
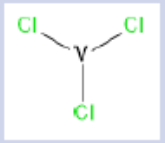
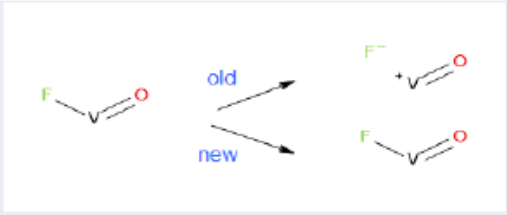
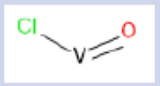
The table that follows shows allowed default valences for neutral main group elements:

1a		2a	3a		4a		5a		6a		7a		8	
H	(1)												He	(0)
			B	(3)	C	(4)	N	(3)	O	(2)	F	(1)	Ne	(0)
					Si	(4)	P	(3,5)	S	(2,4,6)	Cl	(1,3,5,7)	Ar	(0)
							As	(3,5)	Se	(2,4,6)	Br	(1)	Kr	(0)
									Te	(2,4,6)	I	(1,3,5,7)	Xe	(0)
											At	(1,3,5,7)	Rn	(0)

Accelrys Chemical Representation 9.1, 2016

Implicit hydrogens are never added to metal atoms, with the exception of Al(-1) which has a default valence of 4.

Examples

Structure	Standard InChI	Transformation	Mol. Inorganics InChI	Comments
	InChI=1S/3H2O.V/h3*1H2;/q;;;+3/p-3		InChI=1/3H2O.V/h3*1H2;/q;;;+3/p-3	Coord# = 3 in OrgMet4 for Vanadium ΔEN (O-V) >= 1.7
	InChI=1S/3FH.V/h3*1H;/q;;;+3/p-3		InChI=1/3FH.V/h3*1H;/q;;;+3/p-3	Coord# = 3 in OrgMet4 for Vanadium ΔEN (F-V) >= 1.7
	InChI=1S/3ClH.V/h3*1H;/q;;;+3/p-3		InChI=1/Cl3V/c1-4(2)3	Coord# = 3 in OrgMet4 for Vanadium ΔEN (Cl-V) < 1.7
	InChI=1S/FH.O.V/h1H;;/q;;+1/p-1		Old: InChI=1/FH.OV/c;1-2/h1H;/q;+1/p-1 New: InChI=1B/FOV/c1-3-2	Coord# = 3 in OrgMet4 for Vanadium; ΔEN (F-V) >= 1.7 but V=O stops bond break between V and F
	InChI=1S/ClH.O.V/h1H;;/q;;+1/p-1		InChI=1/ClOV/c1-3-2	Coord# = 3 in OrgMet4 for Vanadium; ΔEN (Cl-V) < 1.7 but V=O

Molecular inorganics: bond handling

- **Status**
  - Programing finished
  - Release testing against the substances of PubChem (about 350 million structures) is ongoing
  - Next beta version released most likely at the end of September 2026

Stereochemistry of molecular inorganics

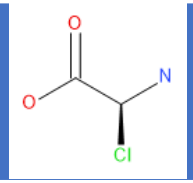
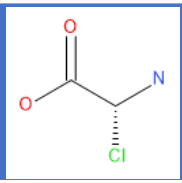
- **Introduction of the new layer "/x" describing the stereochemistry**
- **Necessary input**
  - 2.5D depictions of molecular inorganics
    - Use up and down bonds to mark the stereochemistry in 2D structure depiction
    - Up and down bonds are used to add 3D coordinates (z coordinate) to the atoms around the stereocenter.
  - 3D coordinates of the molecular inorganic structure
    - The determination of the stereochemistry of the metal atom is based on the 3D coordinates of the ligand atoms (= atoms directly bound to the metal atom)
      - Steps
        - Identify polyhedron type of the metal/ligand atom group and identify the preferred z-axis for a spherical coordination system
        - Transfer the 3D cartesian coordinates into spherical coordinates
        - Order the spherical coordinates and transfer the related atom numbers into the /x layer
- **Status: under development including prototyping.**

Enhanced stereochemistry

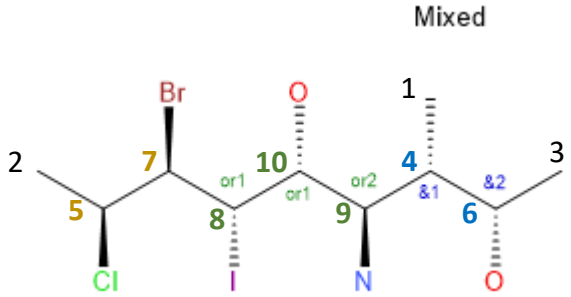
Current InChI stereo parameters

Currently available InChI parameters

- **All parameters are valid for the entire molecule (no combinations).**
- **SAbs (standard InChI)**
  - Tetrahedral center is absolutely described
- **SRel**
  - Tetrahedral center is relatively described
- **SRac**
  - Tetrahedral center is racemic (mixture of both enantiomers)
- **SUCF (Use Chiral Flag)**
  - With Chiral flag = absolute
  - Without Chiral flag = racemic

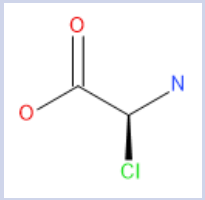
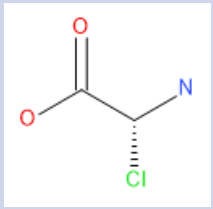
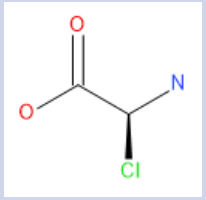
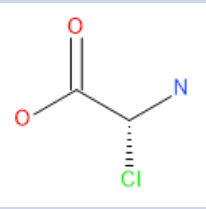
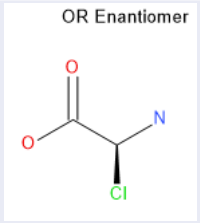
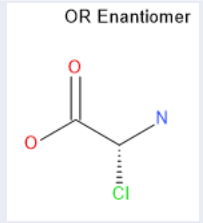
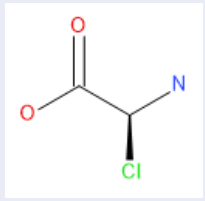
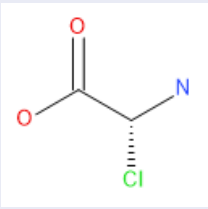
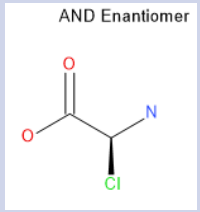
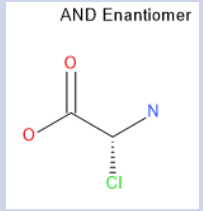
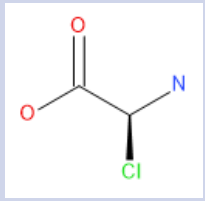
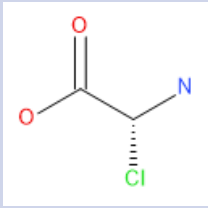
	
InChI=1S/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6) **/t1-/m1/s1**	InChI=1S/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6) **/t1-/m0/s1**
InChI=1/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6) **/t1-/s2**	InChI=1/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6) **/t1-/s2**
InChI=1/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6**)** **/t1-/s3**	InChI=1/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6) **/t1-/s3**
InChI=1/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6) **/t1-/m1/s1** (with Chiral flag, s3 without flag)	InChI=1/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6) **/t1-/m0/s1** (with Chiral flag, s3 without flag)

Enhanced stereochemistry

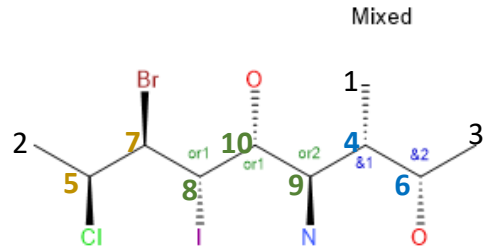


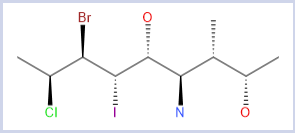
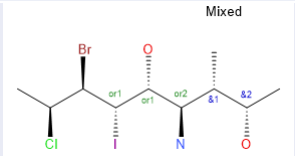
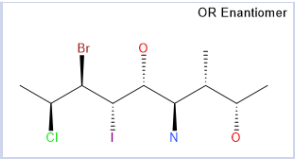
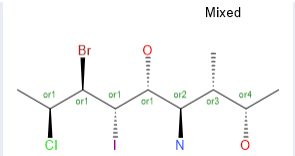
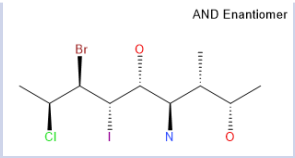
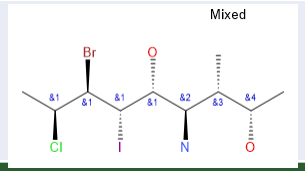
- **Current standard InChI**
  - All centers are seen being abolutely known
  - InChI=1S/C10H20BrClINO2/c1-4(6(3)15)9(14)10(16)8(13)7(11)5(2)12/h4-10,15-16H,14H2,1-3H3/t4-,5+,6+,7-,8-,9-,10+/m1**/s1**
- **InChI with enhanced stereochemistry**
  - Centers are represented according to their stereochemistry
    - Related centers are grouped, e.g. or1, or2
    - Multiple groups are ordered alphanumerically beginning with the lowest number
  - InChI=1B/C10H20BrClINO2/c1-4(6(3)15)9(14)10(16)8(13)7(11)5(2)12/h4-10,15-16H,14H2,1-3H3/t4-,5-,6-,7+,8-,9-,10+/m0**/s1(5,7)2(8,10)(9)3(4)(6)**
- **Note**
  - The string part for the extended sterechemistry is an extension to the stereochemistry layer of the current standard InChI

MDL/Biovia enhanced stereochemistry

	Drawings		Interpretation			
Absolutely known enantiomer				InChI=1B/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6)/t1-/m1/s1		InChI=1B/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6)/t1-/m0/s1
Pure but unknown enantiomer				or		InChI=1B/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6)/t1-/s2
(Racemic) mixture of enantiomers				and		InChI=1B/C2H4ClNO2/c3-1(4)2(5)6/h1H,4H2,(H,5,6)/t1-/s3

MDL enhanced stereochemistry – Edge cases



Drawing	InChI with enhanced stereochemistry
	InChI=1B/C10H20BrClINO2/c1-4(6(3)15)9(14)10(16)8(13)7(11)5(2)12/h4-10,15-16H,14H2,1-3H3/t4-,5+,6+,7-,8-,9-,10+/m1/s1
	InChI=1B/C10H20BrClINO2/c1-4(6(3)15)9(14)10(16)8(13)7(11)5(2)12/h4-10,15-16H,14H2,1-3H3/t4-,5-,6-,7+,8-,9-,10+/m0/s1(5,7)2(8,10)(9)3(4)(6)
	InChI=1B/C10H20BrClINO2/c1-4(6(3)15)9(14)10(16)8(13)7(11)5(2)12/h4-10,15-16H,14H2,1-3H3/t4-,5+,6+,7-,8-,9-,10+/s2
	InChI=1B/C10H20BrClINO2/c1-4(6(3)15)9(14)10(16)8(13)7(11)5(2)12/h4-10,15-16H,14H2,1-3H3/t4-,5-,6-,7+,8+,9-,10-/s2(4)(5,7, 8,10)(6)(9)
	InChI=1B/C10H20BrClINO2/c1-4(6(3)15)9(14)10(16)8(13)7(11)5(2)12/h4-10,15-16H,14H2,1-3H3/t4-,5+,6+,7-,8-,9-,10+/s3
	InChI=1B/C10H20BrClINO2/c1-4(6(3)15)9(14)10(16)8(13)7(11)5(2)12/h4-10,15-16H,14H2,1-3H3/t4-,5-,6-,7+,8+,9-,10-/s3(4)(5,7,8,10)(6)(9)

Enhanced stereochemistry

- **The concept of the enhanced stereochemistry has been transferred to non-tetrahedral stereocenters:**
  - Allenes
  - Atropisomers
  - Double bonds (E/Z)
- **Current status: The developments can be tested in the Web demo under version “Dev with Enhanced Stereochemistry”**

Polymers

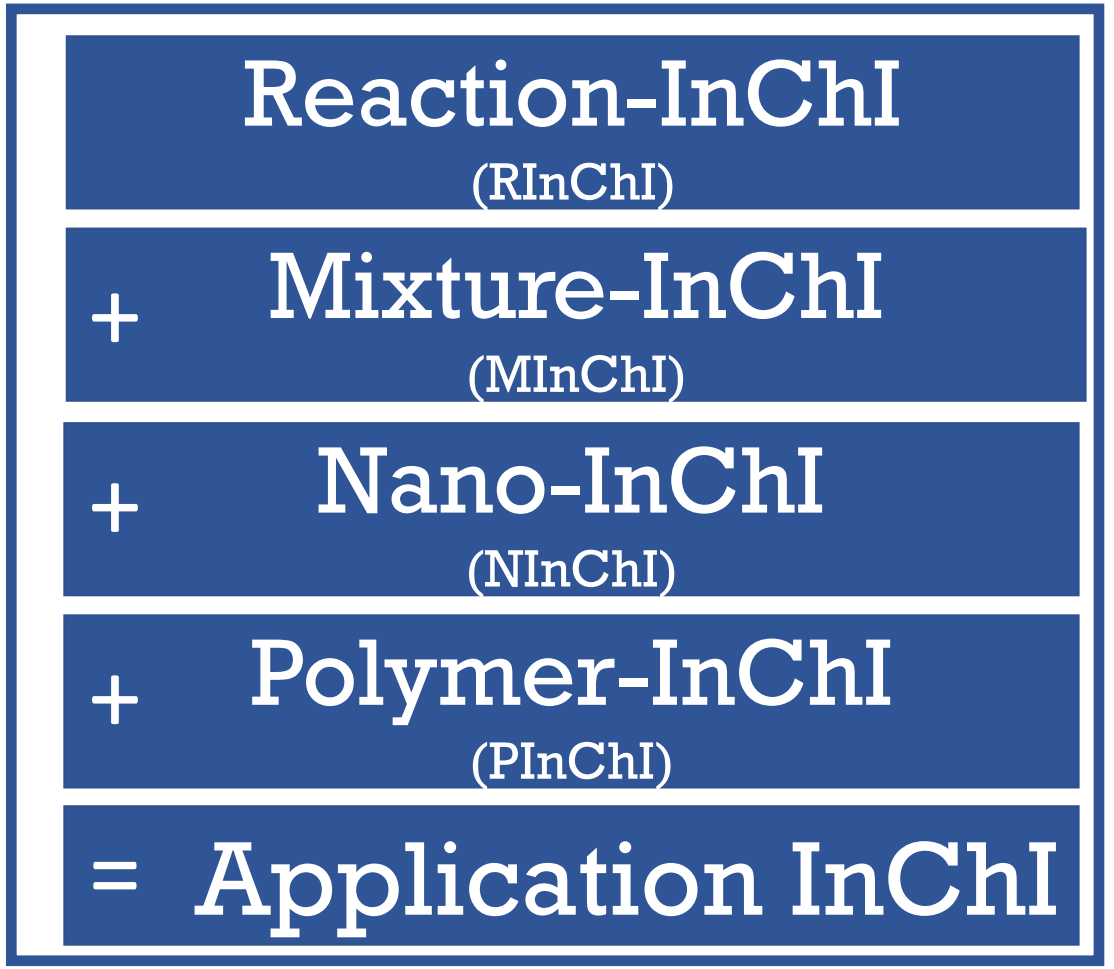
Polymers

- **Completion of the full Sgroup functionality**
  - Adding missing bracket types like “anyp” and “generic”
  - Component and mixture/formulation brackets
- **Introduction of Sgroup data**
  - Polymers
  - Molecular inorganics
  - Isotopologues
- **Structure representation will be the base for the Polymer InChI (PInChI, application InChI)**

Application InChIs

Application InChIs – Basics

- **Application InChIs**
  - Based on standard InChIs, Application InChIs use rule sets to combine multiple InChIs for the identification of chemical identities.
  - May have to use additional alphanumerical data for the identification

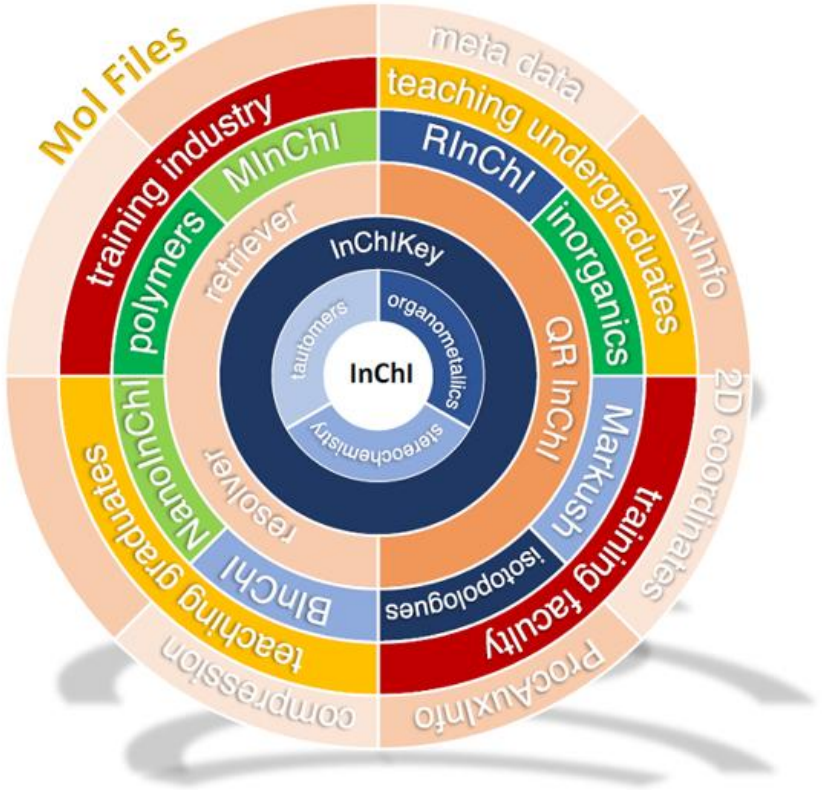


Application InChIs - Status

- **Status**
  - Reaction-InChI (**RInChI**): identification of reactions representing reactants, products, and agents
    - Version 1.0, 1.1 for testing
    - Planned additional auxiliary layers: atom mapping
  - Mixture-InChI (**MInChI**): Identification of mixture components including the mixture order (formulation) and concentration
    - Prototype developed by Leah McEwan and Alex Clark supported by CDD Vault
  - Nano-InChI (**NInChI**): Identification of nano materials by InChIs for the structural representations and alphanumerical data like crystal classes, composition, etc.
    - Definition phase
  - Polymer-InChI (**PInChI**): Identification of complex polymers based on the polymer capabilities of the InChI
    - Definition phase

Application InChIs

- **Exchangeability: InChIs and application InChI must be usable in (other) application InChIs**
  - Share syntax rules
  - Avoid overlaps of naming conventions
    - Layer definitions and concatenators must be unique for InChIs and all application InChIs
- **Status: Requirements phase has started to develop unique syntax and provide common reader and writer**



New InChI versioning schema

New InChI versioning schema

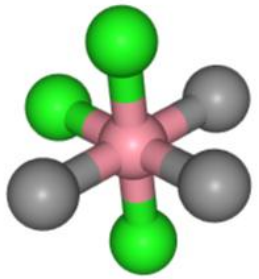
- **The InChI software versioning schema is changed to**
  - year.*N*-release tag
    - N = 1, 2, …: beginning with 1 in each year
    - Release tag: name of the main feature implemented
      - Without tag: technical bug fixes without changes in the InChI string (e.g. fixes of memory leaks)
  - E.g. 2026.1-Molecular_Inorganics_Bonds
  - Notes:
    - Version schema reflects major and minor releases
    - Version system should have a generally accepted format
      - Allow the use of standard tools (e.g. for automated compilation processes)
      - This schema is quite frequently used by software companies, e.g. Biovia, Oracle, Revvity
    - Current schema: 1.07.x

New InChI versioning schema – InChI (string)

- **InChI string and InChI key identifier**
  - The molecular inorganics release will change about 3% of all InChIs in general (About 3% of the PubChem substance are molecular inorganics)
    - The InChI of organic compounds does not change
  - Changes in the InChI string can be easily identified because they are related to modifications in the InChI layers
  - Changes in the InChIKey cannot be recognized because the InChIKey is a hashed representation of the InChI string with fixed lengths for each layer.
    - The InChIKey changes with each change in the InChI string
    - The hash process does not allow the identification of the modified InChI string section

Current version of the InChI identifier (string)

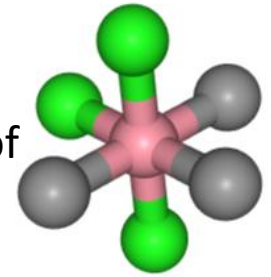
- **InChI-Identifier**
  - InChI
    - Currently “1”
      - With “S” for Standard release
      - With “B” for Beta release
      - Without additional identifier: InChI calculated using special parameters
      - Example: InChI=1S/3CH3.3ClH.Co/h3*1H3;3*1H;/q;;;;;;+3/p-3
  - InChIKey-Identifier
    - Currently “A” (referring to InChI “1”)
      - With “S” for Standard release as “SA”
      - With “B” for Beta release as “BA”
      - With “N” for keys based on InChI calculation using special parameters as “NA”
      - Example: YPJLJEZTPSNLJF-UHFFFAOYSA-K
  - AuxInfo
    - Currently “1” (not changed)



Example: $CoCl_3(CH_3)_3$

New version of the InChI identifier (string)

- **Introduction of a subversion indicator based on characters B, C, D, …**
  - Schema for the InChI string: InChI=1S*/….. with * = B, C, D, … in case new functionality changes the InChI string, else * is kept empty
    - The subversion is only indicated if the related chemical functionality occurs in the molecule
      - E.g. the introduction of the new molecular inorganic rules does not change the InChI of organic compounds. These InChIs are not marked by a subversion letter.
    - InChIs of molecular inorganics are indicated by the subversion “B”.
    - Example: InChI=1SB/C3H9Cl3Co/c1-7(2,3,4,5)6/h1-3H3



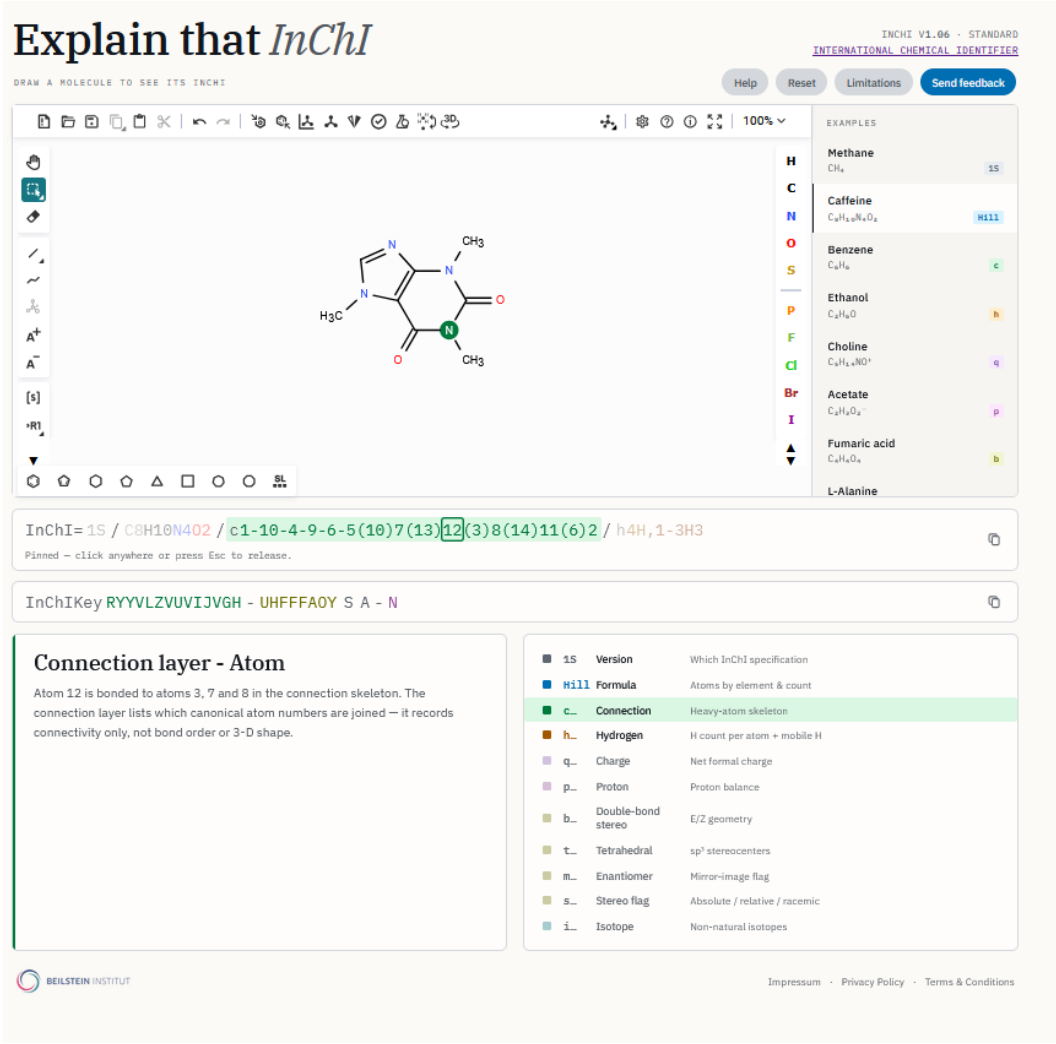
Example: $CoCl_3(CH_3)_3$

  - Schema for the InChIKey: The InChIKey version identifier “…SA-…” is modified to “…S*-…” with * = A, B, C, D, … where “A” indicates the current Standard InChI without any further modification
    - Example: AHAXEBASYQUWLI-UHFFFAOYSB-N

InChI – educational support

InChI – educational support

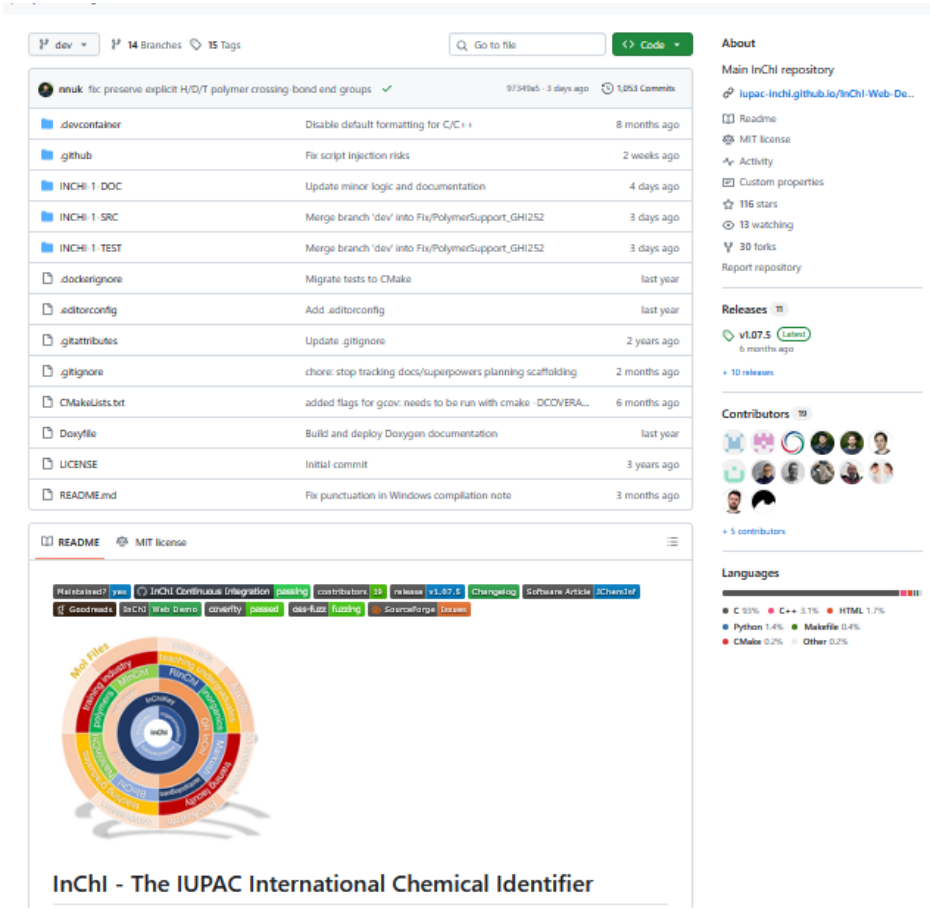
- **Explain that InChI**
  - InChI visualization tool to relate any layer, atom, bond, stereocenter etc. of the InChI string in the chemical structure
  - Contribution by Beilstein Institute, Frankfurt(Main), Germany
  - https://cheminfo.beilstein.org/explain-that-inchi/



InChI development and test environments

InChI development and test environments

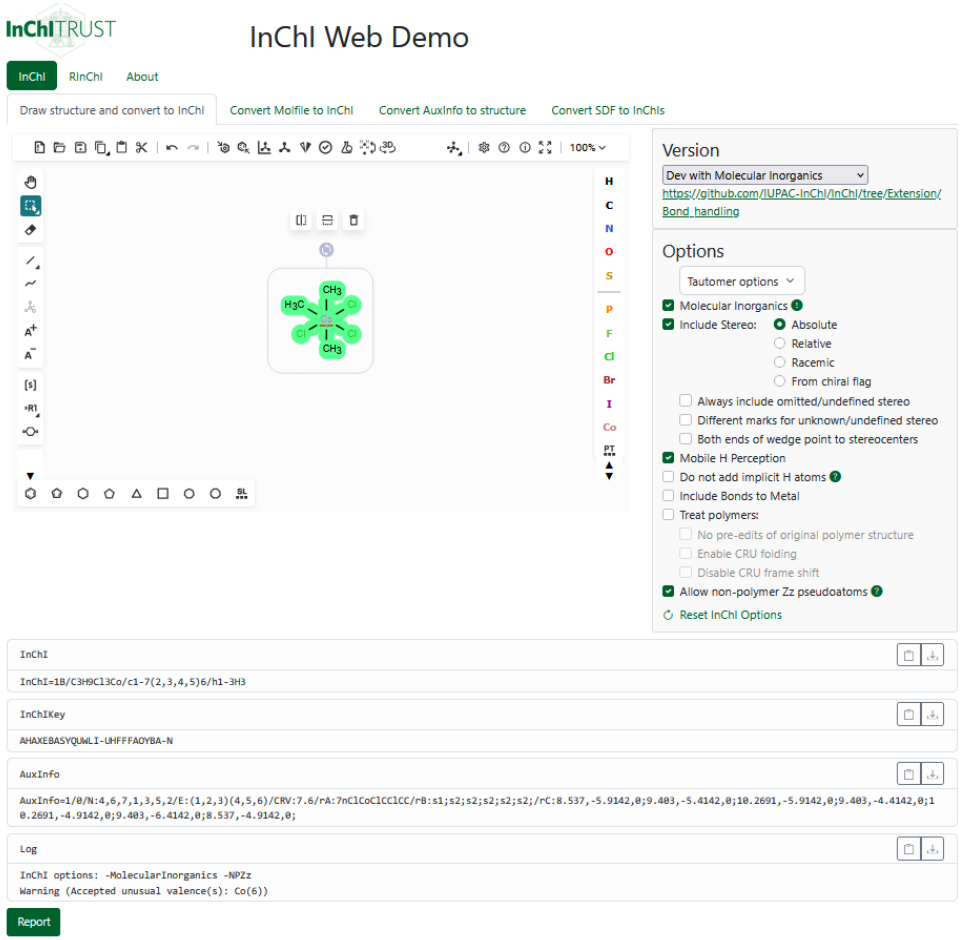
- **InChI moved to GitHub**
  - https://github.com/IUPAC-InChI/InChI
    and can be downloaded from there



  - Use the GitHub issue system to report any bugs, enhancement requests, or any other question

InChI development and test environments

- **InChI Tests**
  - Release tests are run against the full substance set of PubChem (about 350 million structures not normalized.)
  - All releases including actual developments are available within the Web-Demo
    - https://iupac-inchi.github.io/InChI-Web-Demo/
  - Use the report bottom in case of any issue detected



InChI development and test environments

- **Code refactoring**
  - Current status
    - About 220,000 lines of C code developed over more than 25 years.
    - Code is documented sparsely
    - Unit tests have been added to the new developments but do not cover the old code base.
    - Manual refactoring is estimated with 2 FTEs for at least 2 years.
  - Prototype: code refactoring supported by Claude
    - Results look very promising after multiple refactoring steps
      - InChI becomes faster (up to 10%)
      - About 50% code coverage by unit tests
      - Automated Code documentation
      - Consistency is checked by testing against PubChem
  - In case you embedded InChI functionality into your code, please contact us to ensure that your code will continue to work: test@inchi-trust.org

Thanks a lot for funding and support



Thank You