Software status
and InChI version 2
InChI Software releases

- **1.00**
  Apr 2005
  The beginning

- **1.01**
  Aug 2006
  InChI2Struct and many other things appear

- **1.02 beta**
  Sep 2007
  Introduced InChIKey (experimental), API changes

- **1.02 final**
  Jan 2009
  Introduced Standard InChI
  (only Standard supported; InChIKey layout changed)

- **1.03**
  Jun 2010
  Both Standard and Non-std InChI/Key now supported
InChI Software releases

- **1.04**
  Maintenance release;
  more permissive license
  
  Sep 2011

- **1.05**
  Introduced (experimental) support of polymers,
  large molecules, V3000 Molfiles;
  novel API section; multi-threading
  
  pre-release  Oct 2016
  update       Jan 2017
  final        Feb 2017

- InChI for Reactions
  Mar 2017
InChI Software v. 1.05 release

- Maintenance release with a number of significant new features
- All things not included there will most likely be postponed to InChI version 2
Added more elements

- Updated software to current IUPAC confirmed elements list
  - (up to 118 oganesson which closes the Period 7)

- not too much work
- but has to be done in IUPAC-endorsed software
Support of Molfile V3000 format

- Allows one to deal with
  - large (bio)molecules beyond 1000 atoms limit
  - enhanced stereochemistry (e.g., mix of Rel/Abs)
  - extended support of organometallics (haptic bonds)
- The last two features are implemented in reader but are awaiting a future use in InChI 2
- Large molecules ( > 1000 atoms) reading
Support of large molecules

- Limit of number of atoms increased (technically) from 1023 to 32767
- Some other internal limits relaxed
- May be extended further
- Tests on PDB
  100,000+ molecules;
  PDB -(OpenBabel) MOL V3000 → mol2inchi
25272 atoms
25836 bonds
2.84 sec (i5 2.5 GHz)
12757 atoms
13104 bonds
49.95 sec (i5 2.5 GHz)

PDB
5a21
viral protein, largest chain

InChI=1S/C8080H12360N210502526S46/c1-852-952-1654-4476-2684-2735-8046(4476)7957(12634)9769...
InChIKey=SBVFJWLGWCUFFW-BDSVIIHDASA-N
Cautionary notes

- Is speed a concern?

- Benchmarks: i5 2.5 GHz CPU (single-core) SSD

- 99% of longest chains of ~100,000 proteins of PDB (up to ~16,000 atoms) converted to InChI for <=180 sec

- Average processing time ~ 3.8 sec (average size 2400 atoms)

- Still, there are molecules not converted to InChI for reasonable time...
Cautionary notes

- InChI was not designed with $\gg 1000$ atoms in mind

- Though canonicalization and normalizations algorithms principally should work...

- and no problems were reported yet...

- several issues were already found by internal tests
Cautionary notes

- Renumbering tests
- ~70,000 max-length protein chains from PDB were tested, with 100 random atomic renumberings for each
- 14 failures detected so far
  - that is, 14 molecules from PDB give different InChI/Key’s on re-numberings
- No final clarity yet
  - problem may lie in normalization (mobile H) rather than in canonicalization
Cautionary notes

- InChI’s are getting very long
- InChIKey in its current form may be too short to serve for all the large molecules people may start to play with
- Experimental (beta) large-mol InChI/Keys are isolated from others by using ‘B’
Support of polymers

- Only simple polymers (no cross-linked, etc.)
- Source-based representation
- Structure-based representation
Known issues with polymers

- Issues on elucidation of canonical SRU
  - reported by Roger Sayle and John Mayfield, re-iterated today
  - BTW: explicitly stated in documentation (in part)

- Issue #1, simplified:
  - $[-\text{CH}_2\text{CH}_2-]_n$ $\text{NE}$ $[-\text{CH}_2-]_n$
  - But should it? Odd/even, etc., repeatability

- Issue #2:
  - no polymer SRU “frame shift” analyzed when explicit end groups specified
    - $\text{H}_2\text{N}-[-\text{CH}_2-\text{C(O)}-\text{NH}-]_n-\text{CH}_2-\text{C(O)}\text{OH}$ $\text{NE}$ $\text{H}_2\text{N}-\text{CH}_2-[-\text{C(O)}-\text{NH}-\text{CH}_2-]_n-\text{C(O)}\text{OH}$ $\text{NE}$ $\text{H}_2\text{N}-\text{CH}_2-\text{C(O)}-[-\text{NH}-\text{CH}_2-\text{C(O)}-]_n-\text{OH}$
  - BTW: frame shift is of course supported when star atoms (*) are shown instead
    - $*-[-\text{CH}_2-\text{C(O)}-\text{NH}-]_n-*$ $\text{EQ}$ $*-[-\text{C(O)}-\text{NH}-\text{CH}_2-]_n-*$ $\text{EQ}$ $*-[-\text{NH}-\text{CH}_2-\text{C(O)}-]_n-*$

- In principle, solvable
  - But solution seems to be far from nice
  - Further feedback desired (this meeting, discussions, opinions of polymer chemists?)
New "extensible" (IXA) API

- IXA stands for “InChI Extensible API”
- Adds new API procedures including low-level functions to deal with atoms, bonds, etc.
- Code supplied by Digital Chemistry
  John Barnard with co-workers
- Ported to Linux
  with help of Pubchem team
  Evan Bolton, Paul Thiessen
- No problems reported (yet)
Support of safe multi-thread execution

- Allows one to significantly increase speed of InChI/Key generation while calling InChI Library on multi-CPU hardware (most of modern systems)

- Code changes supplied by Bio-Rad
  - Karl Nedwed

- Porting/testing on Linux with help of Pubchem team
  - Evan Bolton, Paul Thiessen

- No problems reported (yet)
Current status

- To early to remove “experimental” label from both large molecules and polymers

- 1.051 intermediate release
  - to include fixes for several already found minor bugs & ”features”
  - may be launched on Fall 2017
Suggested near future updates

- 1.051 intermediate release
  - to include fixes for several minor bugs & ”features”
  - tentatively planned for Fall 2017
InChI version 2

- Working groups

...
InChI version 2

- Very rough estimate (0-5) of implementation effort
  - Tautomerism 3.5
    - moderate to significant
  - Organometallics 4.5
    - significant to monstrous
  - Advanced large molecules 4.5
    - significant to monstrous
      (depend on canonicalization issues, HELM integration, ...)
InChI version 2

- Very rough estimate of implementation effort
  - QR-codes: 2
    - minor
  - Mixtures: 2.5
    - minor to moderate
InChI version 2

- Other (no working-groups)

- Enhanced stereo (following V3000)
  - Collections, ABS/AND/OR
    - Relatively straightforward
InChI version 2

- Other (no working-groups)

- Longer InChIKey
  - “codebreaking” sport
  - anyway, 1\textsuperscript{st} block is not a real issue
    - Tolerate $\sim 1 \times 10^9$ entries
    - (Andrey Erin: 12 collisions per $27 \times 10^9$, theor. estimate is $\sim 10$)
    - may be slightly increased in length

- 2\textsuperscript{nd} block is what really counts!
InChI version 2

- Longer InChIKey
  
  2\textsuperscript{nd} block may really have problems

  There are much things there already (think of carbohydrates!) ...
  people are trying to squeeze everything in there (polymers...mixtures...)
  and this likely will continue)

  - Make 2\textsuperscript{nd} block significantly longer
  - or just add 3\textsuperscript{rd} car to the train?