The Good, the Bad and the Ugly: Elementary algorithmic robustness and consistency in chemical information software

Wolf-D. Ihlenfeldt, Xemistry GmbH, Königstein, Germany

The Premise

It is an undisputed fact that there are many ways to draw and encode one and the same chemical structure.

Canonicalizing storage formats such as Unique SMILES or InChI are not useful for many applications because they cannot store auxiliary structure information in a portable fashion.

When commingling structure information with data, formats such as 2D or 3D SD files, RD files, Sybyl2 Molfiles or PDB are standard, and in these files 2D structures can be drawn, and the atoms and bonds numbered, in a nearly arbitrary fashion, as long as overall connectivity and stereochemistry is correct.

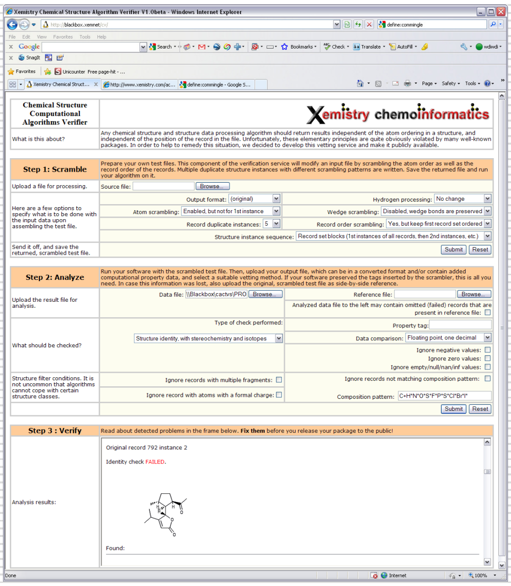
Software processing such files as input, or converting between structure file formats, is expected to yield results independent of atom and bond order as well as the exact placement of wedges and other stereochemistry indicators.

However, after being dissappointed more than once when confronted with actual results, we set out to quantify and study this condition in more detail.

Test Data Set

We used the first 1000 records from the VCH Natural Products Database for this study. Compared to typical drug design datasets this dataset is diverse, contains structures of above-average size and lots of stereochemistry, including complex cases. The the right are some examples of structures posing special problems for various software we tested.

We ran the specific software tests described below with a fully scrambled (atom, bonds, records) set of 5 structure instances in record set sequence arrangement (5000 records total).



Xemistry Chemical Structure Algorithm Verifier V1.0beta - Windows Internet Explorer

http://blackbox.xemnet/cv/

Chemical Structure Computational Algorithms Verifier

What is this about? Any chemical structure and structure data processing algorithm should return results independent of the atom ordering in a structure, and independent of the position of the record in the file. Unfortunately, these elementary principles are quite obviously violated by many well-known packages. In order to help to remedy this situation, we decided to develop this vetting service and make it publicly available.

Step 1: Scramble Prepare your own test files. This component of the verification service will modify an input file by scrambling the atom order as well as the record order of the records. Multiple duplicate structure instances with different scrambling patterns are written. Save the returned file and run your algorithm on it.

Upload a file for processing. Source file: Browse...

Here are a few options to specify what is to be done with the input data upon assembling the test file.

Output format: (original)
Hydrogen processing: No change
Atom scrambling: Enabled, but not for 1st instance
Wedge scrambling: Disabled, wedge bonds are preserved
Record duplicate instances: 5
Record order scrambling: Yes, but keep first record set ordered
Structure instance sequence: Record set blocks (1st instances of all records, then 2nd instances, etc.)

Send it off, and save the returned, scrambled test file. Submit Reset

Step 2: Analyze Run your software with the scrambled test file. Then, upload your output file, which can be in a converted format and/or contain added computational property data, and select a suitable vetting method. If your software preserved the tags inserted by the scrambler, this is all you need. In case this information was lost, also upload the original, scrambled test file as side-by-side reference.

Upload the result file for analysis.

Data file: \\Blackbox\cactvs\PRO Browse...
Reference file: Browse...
Analyzed data file to the left may contain omitted (failed) records that are present in reference file: □

What should be checked?

Type of check performed: Structure identity, with stereochemistry and isotopes
Property tag:
Data comparison: Floating point, one decimal
Ignore negative values: □
Ignore zero values: □
Ignore empty/null/nan/inf values: □

Structure filter conditions. It is not uncommon that algorithms cannot cope with certain structure classes.

Ignore records with multiple fragments: □
Ignore record with atoms with a formal charge: □
Ignore records not matching composition pattern: □
Composition pattern: C+H*N*O*S*F*P*S*Cl*Br*I*

Submit Reset

Step 3 : Verify Read about detected problems in the frame below. **Fix them** before you release your package to the public!

Analysis results:

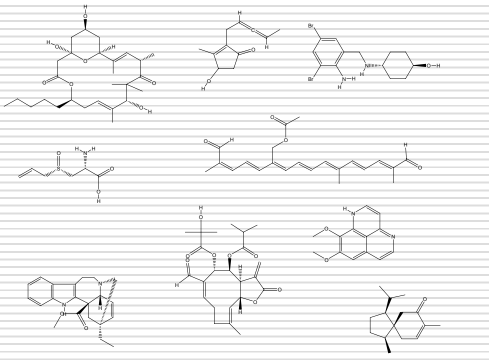
Original record 792 instance 2

Identity check FAILED.

Found:

Done

File Edit View Favorites Tools Help
Google Search Bookmarks Check Translate AutoFill
Favorites Unicounter Free page-hit - ...
Xemistry Chemical Struct...
http://www.xemistry.com/ac...
define:commingle - Google S...
Page Safety Tools
Internet
100%



Software

For this project, we developed a generic chemical structure algorithm results consistency checker as a publicly accessible Web service. It is implemented as a simple CGI application scripted with the Cactvs Chemoinformatics Toolkit. The size of the CGI script is only about 550 lines of code, demonstrating the power of the toolkit at providing solutions for chemical information processing problems on a very high level.

The Web service consists of two components:

1. A data file record shuffler. Structures from an uploaded multi-record input file are returned as a file with multiple input structure instances with scrambled atom and bond numbering and wedge bond placements, possibly also in shuffled record order. The structures are tagged in multiple fields so that supposedly equivalent structures can still be recognized after 3rd party software has processed the file.

2. A result file verifier. After processing the shuffled test data file with software to be vetted, the result file, with augmented data and/or in a converted format, is uploaded to the second component. The original shuffled test file can be provided as additional side-by-side information source for cross-checking. The verifier reads all records and compares the actual structural identity of the supposedly equivalent records, or the consistency of data computed for those records. Various options allow the control of comparison precision, equivalence criteria, etc. The verifier delivers a detailed report about any inconsistencies found.

The public URL is www.xemistry.com/cv

Data submitted to this server is logged. Do not send confidential data. In-house versions of the software can be licensed from Xemistry.

The Good

• Cactvs file I/O and conversion. There is a 100% structure identity consistency rate between various file interconversions, for formats like SDF, SMILES, MOL2, and still 98% for bondless PDB to SDF.

• NIST InChI encoder. We did not find any flaws in the implementation. InChI strings are constant for all structures regardless of scrambling and record re-ordering.

• Cactvs InChI interface. It delivers the same InChI strings as the NIST implementation, even when computing them from Cactvs-converted file formats the NIST code does not read and then running a side-by side comparison with NIST results from a parallel SD file.

• Cactvs XLOGP2 re-implementation. The code was scripted from the description in the paper and, for finer points, by looking into the source code. This implemenation is stable.

• CORINA 3D generator, with some caveats. There are 13 cases (0.3%) which cannot be solved in some atom numbering – but all structures yield a reasonable 3D structure in at least one instance. Interestingly, there is an option to run internal canonicalization before coordinate computation. If set, one always gets the same results – but can miss solutions which are not found in the canonic numbering.

The Bad

• The original XLOGP2 implementation. 6 crashes for structures where hydrogen is not at the rear of the record, plus 116 value deviations >0.1 (2.3%), many of them significant. There is obviously a problem with detecting π systems linking donor/acceptor centers depending on the atom numbering. Example XlogP values for R985 #1 and #2: -0.12, #3: 0.72, #4: Crash, #5: 0.30

• MOLD2 descriptor calculation. 4 crashes plus 76044 value deviations >0.1, some of them several orders of magnitude. Still, since the descriptor set are 777 values per structure, this is a modest problem rate of about 2%. The deviating descriptors are heavily clustered. Many have no problems at all, others in 4 of the 5 instances, in some cases spanning the whole spectrum from reasonable value, large deviation, suspicious 999.0 values and/or crash. Additionally, there seems to be a record memory. Whether a given structure instance crashes is dependent on the previous records – it works with some leading records, or as first structure in file, but not in other contexts.

• NIST InChI resolver. 741 of 5000 records (14.8%) do not yield the original structure. However, this sounds worse than it is. These are generally tautomers of the original structure. Adding a few rules for tautomer normalization, even if just for simple cases such as amide groups, would greatly reduce this count.

The Downright Ugly

OpenBabel was tested in the most current version 2.2.3. It exhibits a disturbing lack of stereochemical understanding, for example concerning stereogenic double bonds in large rings or exo to rings. Even classes of stereochemistry it understands in principle are often and randomly mishandled and mixed up.

• InChI from 2D SDF: 524 of 5000 (10.5%) records not identical with NIST result. Generally missing or wrong stereochemistry.

• SMILES from 2D SDF: 271 of 5000 (5.4%) records do not resolve to original structure. Missing or wrong stereochemistry, plus broken aromaticity encoding and decoding.

The Maintainer Attitude

Start page: *Downloaded over 100.000 times!*

Bug tracker: *E/Z stereochemistry switched on conversion of trivial SMILES C/C=C/C=C/C(=O)OCC(=O)N*

Status: *Closed, nobody assigned, resolution maybe in a planned future full rewrite of stereochemistry handling without a schedule.*

Given the widespread use of Babel, relying on the stereochemistry in SMILES or InChI from any public source is not recommended. This is a very unfortunate situation.