ciftostr

Release 0.2.0

Matthew Rowles

Nov 12, 2022

CONTENTS

CHAPTER

ONE

OVERVIEW

docs	
tests	
package	

This program converts an arbitrary number of CIF (Crystallographic Information Format) files, each containing an arbitrary number of crystal structures, into a number of individual STR files that are compatible with the Rietveld analysis software, TOPAS.

If the program is run with zero command line arguments, then a GUI is launched, otherwise it is command-line driven.

- Free software: Apache Software License 2.0

1.1 Pre-installation

If you are on Windows, you must read this step. If you are on Linuz, you can continue.

`ciftostr` requires `PyCifRW >= 4.4.3`. If you install `PyCifRW` from PyPI via `pip`, then you will also need to compile the included C modules. To do so requires Microsoft Visual C++ 14.0 or greater. If you don't have this installed, or do not wish to install it, precompiled wheel files are available. You must download the wheel file corresponding to your Python installation.

To obtain information about your Python installation, run the command:

```
python -VV
```

An example output is *Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit (AMD64)]*, showing that this is 64 bit Python 3.9.

Using `pip` version 19.2 or newer, install your downloaded wheel file as:

```
pip install c:\path\to\file\name_of_file.whl
```

This should install `PyCifRW`, and you can move on to the next step. If you encounter any issues in the installation, please lodge an issue.

1.2 Installation

To install the release version of `ciftostr` from PyPI:

```
pip install ciftostr
```

You can also install the in-development version of `ciftostr` with:

```
pip install https://github.com/rowlesmr/ciftostr/archive/master.zip
```

1.3 How to run

If you would like to run it in the command line, you need to provide some command-line arguments:

```
ciftostr *.cif
ciftostr C:\user\username\some\directory\name.cif D:\data\file.cif
ciftostr /nfs/an/disks/jj/home/dir/file.cif home/folder/nest/deeper/important.cif
```

will convert all CIF files in the current directory or the two CIF files specifically - both Windows and Linux filepaths are accepted. The resulting STR files will be saved in the same path as their parent CIF file. Relative or absolute file paths can be used.

To launch the GUI:

```
ciftostr
```

Bonus Windows executable behaviour: You can drag a CIF file onto the program icon, and it will convert the CIF for you.

To see some other helpful information, try:

```
ciftostr --help
ciftostr --info
```

1.4 jEdit integration

Link text

John Evans has made available a number of macros for the text editor jEdit that make it integrate very well with TOPAS when operating in launch mode.

As a part of these macros, there is the ability to insert CIF files in STR format. If you wish to use `ciftostr` to generate the STR information, then you need to make the following changes:

0. `pip install ciftostr` (or use `ciftostr.exe`)
1. Replace your copy of `TAInsertCIF.bsh` with this one. (Your file is probably found in C:\Users\?????\AppData\Roaming\jEdit\macros)

Now, when you choose "`Insert CIFs in INP format`" from the plugin in jEdit, `ciftostr` will be used in the background to generate that format.

1.5 Documentation

https://ciftostr.readthedocs.io/

This program is designed to reformat data from a CIF format into the STR format suitable for use by the Rietveld analysis software, TOPAS. It doesn't carry out any validation checks to ensure the data is consistent, but simply transcribes the data as given in the CIF.

Where similar or identical data could be given in several places, the values are taken in a specific order of precedence, as detailed in the sections below. In general, if a value exists in an earlier place, the later places are not looked at.

This program uses the PyCifRW library, written by James Hester, to parse the CIF files into a format easily used to remix the underlying data. The GUI was created using PySimpleGUI.

The STR's phase_name is taken from `_chemical_name_mineral`, `_chemical_name_common`, `_chemical_name_systematic`, or `_chemical_name_structure_type`, in that order, appended with the value of the `data` block. If none of these keys are available, then the name of the `data_` block is used. This is also used as the filename of the STR file.

The unit cell parameters are taken from `_cell_length_a`, `_cell_length_b`, `_cell_length_c`, `_cell_angle_alpha`, `_cell_angle_beta`, and `_cell_angle_gamma`. Some comparisons are made to enable some standard macros to be used (eg Cubic, Tetragonal…). In any of these fail, then all parameters are given as a fail safe.

The space_group is taken from `_symmetry_space_group_name_H-M`, `_space_group_name_H-M_alt`, `_symmetry_Int_Tables_number`, or `_space_group_IT_number`, in that order. If none of these keys are available, then an empty string is written as the space group. The value of the space group is as exactly given in the CIF. No validation or editing is done.

The atomic sites are constructed as follows: The atom labels are taken from `_atom_site_label`, with the fractional x, y, and z coordinates given by `_atom_site_fract_x`, `_y`, and `_z`. If the decimal values of the fractional coordinates are consistent with the fractions 1/6, 1/3, 2/3, or 5/6, then the decimal value is replaced by the fractions. The site occupancy is given by `_atom_site_occupancy`, or by `1`, if that key is not given. The atom type is given by `_atom_site_type_symbol`, where available, or by the first one or two characters of the site label. If these characters match an element symbol, then that is used, otherwise, the label is used in it's entirety, and the user must decide the correct atom type to use. If the site label starts with `Wat`, (and no atom type is given) it is assumed that oxygen (from water) is correct. An attempt is also made to reorder the charge given on an atom, to ensure it is compatible with TOPAS ordering, eg `Fe+2`, not `Fe2+`.

Isotropic Atomic Displacement Parameters (ADPs; Biso), are taken from `_atom_site_B_iso_or_equiv`, or from `_atom_site_U_iso_or_equiv`, where they are then multiplied by 8*Pi^2 to give B values. If anisotropic ADPs are given, then `_atom_site_aniso_B_11`, `_atom_site_aniso_B_22`, and `_atom_site_aniso_B_33` are averaged to give an equivalent Biso value. Alternatively, the equivalent U values are used to calculate Biso. As anisotropic values could be given for a subset of the atoms in the structure, the labels given by `_atom_site_label` and `_atom_site_aniso_label` are matched, and if an atom doesn``t have an anisotropic value, it takes its isotropic value, or is assigned a value of `1`.

The atomic site is also given a `num_posns 0` entry, which will update with the multiplicity of the site following a refinement. This will allow the user to compare this value with the CIF or Vol A to help ensure that the correct symmetry is being applied.

Finally, the STR is given a fixed Lorentzian crystallite size of 200 nm, and a refinable scale factor of 0.0001 to allow for an easy start to a refinement. All other values given in the STR are fixed, and require active intervention to name, refine, constrain, or restrain them.

If you have any feedback, please contact me. If you find any bugs, please provide the CIF which caused the error, a description of the error, and a description of how you believe the program should work in that instance.

1.6 Development

Come and talk to me!

CHAPTER

TWO

INSTALLATION

At the command line:

```
pip install ciftostr
```

CHAPTER

THREE

USAGE

To use ciftostr in a project:

```
import ciftostr
```

CHAPTER

FOUR

REFERENCE

4.1 ciftostr

CHAPTER

FIVE

CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

5.1 Bug reports

When reporting a bug please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.2 Documentation improvements

ciftostr could always use more documentation, whether as part of the official ciftostr docs, in docstrings, or even on the web in blog posts, articles, and such.

5.3 Feature requests and feedback

The best way to send feedback is to file an issue at https://github.com/rowlesmr/ciftostr/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome :)

5.4 Development

To set up *ciftostr* for local development:

1. Fork ciftostr (look for the "Fork" button).
2. Clone your fork locally:

```
git clone git@github.com:YOURGITHUBNAME/ciftostr.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

4. When you're done making changes run all the checks and docs builder with tox one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .
git commit -m "Your detailed description of your changes."
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

5.4.1 Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`).
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- pytest -k test_myfeature
```

To run all the test environments in *parallel*:

```
tox -p auto
```

CHAPTER

SIX

AUTHORS

- Matthew Rowles - .

CHAPTER

SEVEN

CHANGELOG

7.1 0.1.2 (2022-02-01)

- Given zero values for B_iso are now ignored.
- Added an option to add some work-related options.

7.2 0.1.1 (2022-01-05)

- Now checks for negative fractional coordinates
- Adds MVW(0,0,0) to strs
- Adds a comment containing the original unit cell prms so you can easily undo them

7.3 0.1.0 (2021-11-07)

- It looks like everything is working!

7.4 0.0.0 (2021-11-07)

- First release on PyPI.

CHAPTER

EIGHT

INDICES AND TABLES

- genindex
- modindex
- search