A Shell Script, cifconv.command for macOS, to Convert CIFs into a Variety of Files

Graduate School of Engineering, Kyoto University

Fujio Izumi*

December 16, 2021

1 What is cifconv.command?

A shell script, cifconv.command, for conversion of crystallographic information files (CIFs) [1] was developed for macOS and Microsoft Windows [2]. Classical text processing under the UNIX programming environment, *i.e.*,

(1) strong combination of a shell and powerful UNIX commands (grep, cat, sed, tr, cut, awk, tee, *etc.*) running on multiple platforms and consuming relatively less memory,

(2) parallel processing through exhaustive use of pipes,

(3) the ease with which source code can be modified and added by the user

enables speedy and elegant text manipulation taking full advantage of concurrent processing using multi-core processors. In particular, the stream editor, sed, is fully utilized mainly for string substitution.

Written in bash, the source code of cifconv.command can be easily edited by a text editor such as Jedit Ω pro[1] to add new features or give its functionality a brushup. Needless to say, porting cifconv.command to other platforms including Linux and Windows plus BusyBox is quite easy [3]. Though the source code of cifconv.command is as long as about 1500 lines, its modular structure consisting of a fairly good number of functions and many comments facilitate the adequate understanding of its contents and processing flow.

In what follows, details in the macOS version of cifconv.command will be described. Two different modes are available in cifconv.command. In a batch mode, cifconv.command is run in Terminal by double-clicking its icon in Finder as described in Procedure 2 of Sect. 3. Cifconv.command sequentially converts all CIFs, ∗.cif, in the current directory into a variety of text and binary files directly or indirectly after executing it. Another uniprocessing mode to deal with a single CIF and copy cifconv.command to the current directory where needed will be described in Sect. 9.

Table 1 lists all the kinds of files input and output by

*E-mail: fizumi3776@gmail.com

[1] http://www.artman21.com/jp/jedit0mega/

Table 1: A list of files output and input by cifconv.command and other programs. Inside = '+': The file is output by another program run in cifconv.command. Inside = '-': The file is output by another program run after executing cifconv.command.

File	Output by	Inside	Input by
hoge.pme[1)]	cifconv.command		MADEL
hoge.mad	MADEL	+	Text editor
hoge.ped[1)]	MADEL	−	VESTA
hoge.dab	bond_str	+	Text editor
hoge.bvs	bond_str	+	Text editor
hoge.pcb[1)]	MADEL, bond_str	+	Text editor
hoge.cco	cryscalc	+	Text editor
INCAR.ins[2)]	RIETAN-FP	+	cifconv.command
INCAR	cif2cell	+	VASP
POSCAR	cif2cell	+	VASP, VESTA
POTCAR[3)]	cif2cell	+	VASP
KPOINTS	cif2cell	+	VASP
hoge.in[4)]	C-Tools	+	Quantum ESPRESSO
hoge.dat[4)]	C-Tools	+	OpenMX
hoge.cg[4)]	C-Tools	+	xTAPP
fort.1[4)]	C-Tools	+	RSDFT
hoge.ins[5)]	cif2ins	+	RIETAN-FP, VESTA
multi_phase.ins[5)]	cifconv.command	+	RIETAN-FP
hoge.lst	RIETAN-FP	−	lst2cif.command, VESTA
hoge-report.pdf	cif2pdf.command	−	PDF browser
hoge-report-j.pdf	E2J.command	−	PDF browser
hoge.plt[6)]	RIETAN-FP	−	Plot.command
hoge.gpd[6)]	RIETAN-FP	−	Plot.command
hoge.pdf[6)]	gnuplot	−	PDF browser
hoge.xyz	RIETAN-FP	−	ORFFE
hoge.ffe	ORFFE	−	RIETAN-FP, VESTA, ffe2ins.command
BVS.inp	RIETAN-FP	−	PyAbstantia
BVS.pgrid	PyAbstantia	−	VESTA
BVEL.inp	RIETAN-FP	−	PyAbstantia
BVEL.pgrid	PyAbstantia	−	VESTA
cifconv.out	cifconv.commad		Text editor

1) See Sect. 8.
2) See 4.4.
3) POTCAR is output only when directories of pseudo-potentials have been copied in directory /Applications/cif2cell/potpaw_PBE (see Sect. 2).
4) See Sect. 6.
5) The CIF, hoge.cif, must be exported by VESTA after standardization of cyrstal data. Further, a template file named template.ins has to share the current directory.
6) A pair of files, hoge.plt and hoge.gpd, are input by gnuplot to give hoge.pdf with a pdfcairo terminal.

cifconv.command. BVS.pgrid and BVEL.pgrid output by PyAbstantia[2] are binary files whereas all the other files are text ones. As the end-of-line (EOL) code, only LF has to be used. If the EOL code is CR/LF, it is automatically changed into LF by a text converter nkf.[3] In Table 1, 'hoge' means the metasyntactic variable, whose extension is 'command' are bash scripts used in the integrated assistance environment for the RIETAN–VENUS system. To typeset hoge-report.pdf (English documents) and hoge-report-j.pdf (Japanese documents), MacTeX[4] (TeX Live for macOS) must have been installed.

As is clear from Table 1, all input files for applications of first-principles calculations are created with cif2cell[5] [4]. Many options given as arguments of cif2cell provide the basis for final input files (see Sect. 4).

A great advantage of cifconv.command over existing utilities is that it can safely input CIFs output by various applications including the Inorganic Crystal Structure Database[6] (ICSD), VESTA, lst2cif, and supercell (see 7.10) if appropriate Hermann–Mauguin space-group symbols are provided at any rate (see 7.1).

2 Installation

The RIETAN-FP–VENUS system[7] for macOS can be installed after mounting a disk image file, Mac_exercise.dmg, by double-clicking it. Next, go down to directory MacOS_versions, and execute Uninstall_RIETAN_VENUS and Install_RIETAN_VENU in this order. Then, run Jedit Ω pro to input MacOS_versions/JeditOmega_Preferences.plist in "Environmental Setting > General > Details."

A python library, PyCifRW 4.4,[8] which is used to read and write CIFs, must be installed to use cif2cell. Note that its installation requires Xcode. PyCifRW 4.4 is much more robust than PyCifRW 3.3 bundled in cif2cell 1.2.10 particularly when inputting Hermann–Mauguin space-group symbols from CIFs exported by ICSD. PyCifRW 4.4 is located under directory /Applications/cif2cell/PyCifRW. To install it, go down to the above directory in Terminal, and input the following command:

```
sudo python setup.py install
```

Then, enter your password. If the following line is finally output, the installation has been finished successfully:

```
Finished processing dependencies for PyCifRW==4.4
```

Registered users of VASP ought to copy all directories (Ac, Ag, Ag_new, Ag_pv, *etc.*) storing pseudo-potential files named POTCAR in directory /Applications/cif2cell/potpaw_PBE

[2] https://shinichinishimura.github.io/pyabst/
[3] https://ja.osdn.net/projects/nkf/
[4] http://tug.org/mactex/
[5] https://sourceforge.net/projects/cif2cell/
[6] http://www2.fiz-karlsruhe.de/icsd_home.html
[7] http://fujioizumi.verse.jp/download/download.html
[8] https://pypi.org/project/PyCifRW/4.4/

containing only PAW_potentials.txt, which is necessary to obtain POTCAR corresponding to a cutoff energy of 550 eV in addition to INCAR, POSCAR, and KPOINTS after running cifconv.command.

3 Procedures of running cifconv.command

1. If necessary, CIFs are pretreated by a shell script called change_hoge.command (see Sect. 10) to change their names so as to show chemical compositions straightforwardly on the basis of `_chemical_formula_sum`.

2. Before using cifconv.command, change shell variables, *i.e.*, `SYMMETRY`, `MADEL`, `CRYSCALC`, `OX_NUM`, `CIF2CELL`, `QE`, `OpenMX`, `xTAPP`, `RSDFT`, `SUPERCELL`, `MAKE_INS`, `VESTA`, `dmax_dis`, and `REGION`, in its main program (see Sect. 7) to optimize its behavior during execution.

3. A macro, cifconv, for Jedit Ω pro can be selected under ContextMenu in the Macro menu or in the context menu that appears by clicking the mouse button on a window of the editor while pressing the control key. After opening a CIF by Jedit Ω pro and select the cifconv macro, it is treated in the uniprocessing mode (see Sect. 9), in which cifconv.command is copied from the /Applications/RIETAN_VENUS/commands_common directory to the current directory if it is absent there.

4. Prior to the execution of cifconv.command obtained in the above step, space-group symbols of Hermann–Mauguin should be checked by consulting /Applications/RIETAN_VENUS/-sg_name.txt.[9] Characters (except for 1, 2, R, H) attached to Hermann–Mauguin symbols with preceding ' ' (space) or ':' (colon) may cause troubles. It is a good idea to enter 'grep y_s *.cif' in Terminal and check all space-group symbols on use of ICSD. A line to give a space-group symbol of Hall [5] is automatically inserted into hoge.cif by cifconv.command in the absence of Hall's symbol, which considerably reduces failure in inputting CIFs by cif2cell.

5. Double-clicking on cifconv.command starts converting all the CIFs sequentially. Standard outputs are saved in cifconv.out in the current directory whereas error outputs can be browsed only in the Terminal window. A crystal structure is automatically drawn by VESTA[10] if the shell variable, `VESTA`, is 1.

6. An input file, hoge.pme, for MADEL is created from a CIF, hoge.cif, which is either exported by ICSD or output by supercell (see Sect. 5); both of these two give CIFs containing `_atom_type_oxidation_number`. An input data, `REGION` (see 7.14), is tentatively set at 9.99 in hoge.pme.

7. Interatomic distances, bond angles, coordination numbers, and bond valence sums (BVSs) [6] are calculated by bond_str included in the FullProf Suite[11] [7] from hoge.cif and saved in

[9] http://cci.lbl.gov/sginfo/hall_symbols.html
[10] Techniques to control VESTA by external programs are descrited in an Evernote note.
[11] https://www.ill.eu/sites/fullprof/

CFL_file.bvs (renamed hoge.dab) and CFL_file_sum.bvs (renamed hoge.bvs) storing only a summary of BVSs. To obtain them, bond_str need to be run twice; at first, bond_str outputs CFL_file.cfl, which is then input by bond_str to give hoge.dab and hoge.bvs. In the absence of information about oxidation numbers of ions in hoge.cif, default values for most elements are substituted for second '0.00' in an input file, CFL_file.cfl, to be corrected by the user, if needed. When processing a CIF output by supercell (7.10), no oxidation numbers of all sites need to be input because those included in the CIf are substituted for '0.00' in CFL_file.cfl.

8. The minimum interatomic distance minus 0.01 is substituted for a dummy string, '9.99', in hoge.pme.

9. Site potentials and an electrostatic (Madelung) energy are computed by MADEL version 2.1 or later by a Fourier method [8, 9] and saved as hoge.mad. The main part of hoge.mad is combined with coordination numbers and BVSs recorded in hoge.bvs to be output to hoge.pcb (see Sect. 8).

10. Cryscalc[12] converts hoge.cif, through hoge_cc.cfl, into hoge.cco containing a variety of crystallographic information such as Wyckoff positions, reflection conditions, constraints imposed on anisotropic atomic displacement parameters, geometrical parameters, and a reflection list containg hkl, m (multiplicity), 2θ, d, $|F_c|$, *etc.*

11. The python script, cif2cell, inputs hoge.cif to create three input files, *i.e.*, POSCAR, POTCAR, and KPOINTS, for VASP[13] [10] in a subdirectory named hoge_VASP provided that occupancies of all the sites are unity (see 4.6). If a template file of INCAR (the central input file of VASP) exists in the current directory, it is copied to the hoge_VASP directory with no change. The specification of calculation types (see (c) in Sect. 4) in hoge.cif makes it possible to modify INCAR. If not INCAR but INCAR.ins is located in the current directory, INCAR.ins is processed by RIETAN-FP v3.0X twice to give INCAR where only data input by VASP remain (see 4.4). A somewhat redundant template file of INCAR.ins written in conformity with the grammar of Tink (see 4.4) is located under each subdirectory of ƒRIETAN_VENUS_examples/CIF_ICSD. In function make_INCAR, new data can be substituted for parts of INCAR as needed with a stream editor: sed. Needless to say, INCAR or INCAR.ins written by the user for special purposes may be placed in the current directory.

12. Input files for Quantum ESPRESSO (PWscf), OpenMX, xTAPP, and RSDFT can also be output by C-Tools from INCAR; See Sect. 6 for details.

13. If $*$.cif exported by VESTA [11] exist in the current directory, cif2ins converts $*$.cif into an input file, hoge.ins, for RIETAN-FP [12], which makes it possible to obtain a list of possible reflections calculate interatomic distances and bond angles without any standard

[12] http://www.cdifx.univ-rennes1.fr/progs/cryscalc/cryscalc_main.htm
[13] https://cms.mpi.univie.ac.at/wiki/index.php/The_VASP_Manual

uncertainties by ORFFE, and plot a powder diffraction pattern in the simulation mode of RIETAN-FP (`NMODE = 1`). Each CIF has to record crystal data with a standardized lattice setting [13]. Bear in mind that template.ins must exist in the current directory to run cif2ins.

14. If two or more CIFs, *@*.cif, exist in the current directory, they are combined into a single template file, multi_phase.ins, for multi-phase Rietveld refinement, which requires considerable modification of multi_phase.ins.

15. After modifying hoge.ins slightly, 3D distributions of BVSs [14, 15] (`NPYABST =1`) and Bond-Valence Energy Landscapes [16] (`NPYABST = 2`) can be visualized in three dimensions in collaboration with RIETAN-FP, PyAbstantia, and VESTA.

The puzzling and troublesome thing is that the order of crystallographic sites in hoge.dab and hoge.bvs output by bond_str often differs from that in hoge.cif, which is considered on assignment of charges to the sites.

4 Information about cif2cell

As decribed in Sect. 3, cif2cell enables us to convert hoge.cif into input files for first-plinciples calculations programs, *i.e.*, VASP [10], CASTEP, Quantum ESPRESSO[14] [17], SIESTA, ABINIT, CPMD, SPR-KKR, xband, Elk, CP2K, Exciting, ASE, FLEUR, FHI-AIMS, CRYSTAL09, HUTSEPOT as well as hoge.XYZ with the XYZ format. I am confident that the feature of creating four input files of VASP from each CIF in the present version will contribute greatly to materials informatics studies.

4.1 Modification of cif2cell and related parts in cifconv.command

The source code of cif2cell can be easily modified by a text editor such as Jedit Ω pro. Therefore, contents of files output by cif2cell can be freely changed. Flags after cif2cell commands in this script may also be added. Unnecessary echo and cif2cell commands may be commented out by adding '`#`' as first characters in the following way:

```
#echo " $hoge.sys for xband" | tee -a cifconv.out
#cif2cell $hoge.cif -p xband -o $hoge.sys
```

Function VASP_files contains all commands related to VASP. At present, lines for all the applications other than VASP are commented out in cifconv.command; part of them will be supported in future.

4.2 Options of cif2cell

All the options of cif2cell are described in /Applications/cif2cell/docs/Options_cif2cell.txt. Further reading the cif2cell manual, cif2cell.pdf, in the same directory is strongly recommended. Keep in mind that

[14] Procedures of installing Quantum ESPRESSO for macOS are described in a blog entry on February 23, 2019.

```
VASP_PAWLIB="/Applications/cif2cell/potpaw_PBE"
```

is declared in the main program. Note that the content of the potpaw_PBE directory must be copied from your VASP application. Further, the potpaw_PBE directory need to contain PAW_potentials.txt (see 4.3).

4.3 Comment lines for VASP in CIFs

Two kinds of special comment lines for VASP may be included in hoge.cif:

(1) `#VASP --...`
Example: `#VASP --print-symmetry-operations --vca --supercell=[2,2,2]`
where additional options for cif2cell are given after '`#VASP`'. In cif2cell, the default resolution, in k-space, adopted to output KPOINTS is 0.2 $Å^{-1}$. To specify a higher resolution of 0.18 $Å^{-1}$, the following line has to be input:
`#VASP --k-resolution=0.18`
Note that '`--setup-all`' accompanied with '`--k-resolution=.....`' is included in the cif2cell command.

(2) `#VASP pbe ....` (`pbe`: case-insensitive)
Example: `#VASP pbe Ba_sv O_sv Ti_sv`
where directory names of pseudo-potentials [18] are specified for three elements in the same order as with POSCAR. They are used to combine POTCAR files of constituent elements to obtain a new POTCAR file.

In the absence of line (2) in a CIF, POTCAR files storing recommended PAW potentials for DFT calculations are selected by referring a text file: /Applications/cif2cell/potpaw_PBE/-AW_potentials.txt. The resulting POTAR file is suitable for calculations with a standard cutoff energy of 550 eV corresponding to '`ENCUT = 550`' set in INCAR.

4.4 Preprocessing of INCAR.ins by RIETAN-FP

If a template file named INCAR.ins exists in the current directory, it is handled twice by a preprocessor named Tink[15] embedded in RIETAN-FP v3.0 or later to produce INCAR [2]. Of course, 'ins' is the extension of standard input files for RIETAN-FP. It is interesting to note that Tink is a nickname of a fairy, Tinker Bell, who is short-lived enough to be completely forgotten by Peter Pan a short time later.

Integer variables are used to control inputting and skipping of lines in INCAR.ins. The name of an integer variable consists of alphabetical capital letters, numbers (0–9), and '@' with the first character being an alphabet in exactly the same manner as hoge.ins for Rietveld analysis by RIETAN-FP. Integer variables are given values as follows:

```
Variable name = Integer: Comment
Variable name = Integer! Comment
```

[15]For details in Tink, refer to 17.3 in RIETAN-FP_manual.pdf.

Even if no '@' is attached to a variable name, the above two kinds of lines are deleted during the processing of INCAR.ins by RIETAN-FP. In the former, both the variable name and integers are input to control the subsequent preprocessing. For example, '`MODE = 1: Optimize`' means that the value of `MODE` is 1 and that the calculation mode is optimization of a crystal structure. On the other hand, the latter, where an integer is followed by '`!`', is regarded as a comment line. Comments may be written in two-byte characters including kanji, hiragana, and katakana, which must be welcomed by Japanese. Saving such files in UTF-8 encoding is highly recommended on macOS.

Unlike in the case of hoge.ins for Rietveld analysis, a simple expression

```
Variable name = Integer
```

that is not followed by '`: Comment`' or '`! Comment`' in the same line should never be described in INCAR.ins. This specification was adopted because it is difficult to differentiate the above description from:

```
Keyword = Integer
```

to be input by VASP. All keywords (tags) of VASP are listed in a Web page of VASP.[16]

Let `MODE` be an integer variable. Then, two kinds of blocks can be included in INCAR.ins: `If` and `Select` blocks. An If block containing `MODE` is exemplified by

```
If MODE = 0 then
  .....
else if MODE = 1 then
  .....
else
  .....
end if
```

The corresponding Select block is

```
Select case MODE
case 0
  .....
case 1
  .....
case default
  .....
end select
```

Be aware that the first characters in '`If`' and '`Select`' must be capital letters; '`If`' and '`Select`' are, respectively, substituted for '`if`' and '`select`' if any. Although such syntax is very similar

[16] https://cms.mpi.univie.ac.at/wiki/index.php/Category:INCAR

to that of Fortran 90/95, Tink requires no pairs of parentheses: '`(`' and '`)`', which makes both of the blocks clean.

Up to two logical expressions may be placed either (a) between '`If`' and '`then`' or (b) between '`else if`' and '`then`'. Each logical expression contains (a) the name of an integer variable, (b) a logical operator, '`=`', '`>`', '`>=`', '`<`', '`<=`', or '`<>`' ($\neq$), and (c) an integer. Two logical expressions are related to each other with a relational operator, '`and`' (logical multiplication) or '`or`' (logical sum), *e.g.*,

```
If MODE = 0 and ILDAU = 1 then
```

and

```
If MODE = 1 or ILDAU = 0 then
```

Unlike Fortran, neither '.and.' nor '.or.' should not be used.

A nest of up to two If and Select blocks is allowed provided that inner blocks are indented at least 1 column, with two or three columns preferred to 1 column. In what follows, an If block includes a Select block:

```
If MODE = 3 then
  .....
  Select case NCHARGE
  case 1
    .....
  case 2
    .....
  case default
    .....
  end select
end if
```

In this case, all the data in the current Select block are input only when `MODE = 3`, which is followed by checking the value of NCHARGE.

Only lines input by VASP are left in INCAR resulting from the second execution of RIETAN-FP. Further, the preprocessing of INCAR.ins by RIETAN-FP makes it possible to use the only one template file for several purposes. Thus, cifconv.command presents a very convenient way for routine use of VASP.

4.5 Insertion of space-group symbols of Hall into hoge.cif

To apply cif2cell to conversion of hoge.cif most surely, either of `_symmetry_space_group_name_Hall .....` and `_space_group_name_Hall.....` [5] is inserted into hoge.cif in the absence of a space-group symbol of Hall, which is indispensable for the use of PyCifRW 3.3 in combination with cif2cell.

4.6 Announcement of deficient sites

VASP cannot process nonstoichiometric compounds containing deficient sites whose occupancies, g, are less than unity. If such vacancies exist, cifconv.command outputs no input files for VASP. For example, in the case of a ferrite, $ZnFe_2O_4$ [19], site labels, chemical-species symbols, multiplicities, Wyckoff letters, structure parameters, *etc.* are recorded in its CIF exported from ICSD as follows:

```
loop_
_atom_site_label
_atom_site_type_symbol
_atom_site_symmetry_multiplicity
_atom_site_Wyckoff_symbol
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_atom_site_B_iso_or_equiv
_atom_site_occupancy
_atom_site_attached_hydrogens
Zn1 Zn2+ 8 a 0.125 0.125 0.125 . 0.79(7) 0
Fe1 Fe3+ 8 a 0.125 0.125 0.125 . 0.21(7) 0
Fe2 Fe3+ 16 d 0.5 0.5 0.5 . 0.895(35) 0
Zn2 Zn2+ 16 d 0.5 0.5 0.5 . 0.105(35) 0
O1 O2- 32 e 0.258(2) 0.258(2) 0.258(2) . 1. 0
```

That is, the ninth data in each line corresponds to an occupancy. On treatment of the CIF with cifconv.command, the following message appears in a Terminal window, and calculation by VASP is skipped:

```
Site Zn1 is deficient: g = 0.79
Site Fe1 is deficient: g = 0.21
Site Fe2 is deficient: g = 0.895
Site Zn2 is deficient: g = 0.105
VASP cannot be run in the presence of sites with g < 1
Construction of derivative structures by supercell is highly recommended
```

5 Modeling of atomic substitutions and partial occupancies by supercell

5.1 Supercell program

When dealing with nonstoichiometric compounds, derivative structures (triclinic space group: $P1$) consisting of only fully occupied sites with $g = 1$ should be constructed by some utilities

such as supercell [20] and enumlib[17] wrapped in the pymatgen library.[18] The supercell program has general versatility and a particularly strong affinity with cifconv.command because it utilizes CIFs as input and output files. It has another great advantage that electrostatic (Coulomb) energies of derivative structures can be calculated with the Ewald summation algorithm, which is particularly effective in the efficient screening of possible derivative structures in nonstoichiometric compounds where two or more ions with different oxidation numbers share the same sites. A compiled binary of supercell for macOS can be downloaded from the Web site of supercell;[19] Note that the latest version 2.0 of supercell can be run only on M1 Mac. Other options of supercell can be changed by editing supercell.command.

5.2 Size of the cell

Special comment lines for supercell may be placed at the top of hoge.cif:

```
#supercell AxBxC
```

where `A`, `B`, and `C` are integer multipliers in primitive lattice (translation) vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$. For example, two lines

```
#supercell 2x1x1
#supercell 2x2x1
```

represent translation vectors, $2\boldsymbol{a}+\boldsymbol{b}+\boldsymbol{c}$ and $2\boldsymbol{a}+2\boldsymbol{b}+\boldsymbol{c}$.

On execution of cifconv.command after setting a variable SUPERCELL (see 7.10) at 1, a subdirectory, hoge_sc, containing hoge.cif suitable for running supercell is created.

5.3 supercell.command

When running a supercell macro under Others in the Macro menu while displaying hoge.cif in directory hoge_sc, CIFs of derivative structures are output in directory AxBxC, which includes a text file named hoge_coulomb_energy_l.txt recording names of CIFs and electrostatic energies for 20 structures[20] with relatively low energies. The standard output of supercell.command is stored in AxBxC/supercell.out. If two or more `#supercell .....` lines are placed at the top of hoge.cif, directory hoge_cif will contain subdirectories corresponding to them.

If a shell variable, PATTERN, is set at 1 in supercell.command, PDF files of powder-diffraction patterns are further output to the hoge_sc directory by executing cif2ins, RIETAN-FP, and gnuplot sequentially. The original CIF, hoge.cif, is copied in the hoge_sc directory to be processed in the same way provided that crystal data in hoge.cif are based on standardized crystal lattices in conformity with Structure Tidy. To run supercell.command to plot powder-diffraction patterns, a template file named template.ins has to be located in directory hoge_sc. Margins of the resulting PDF files are automatically cut thanks to the use of the pdfcairo terminal. This feature will serve for screening of candidates for derivative structures by comparing their powder-diffraction

[17] https://github.com/msg-byu/enumlib
[18] http://pymatgen.org/
[19] https://orex.github.io/supercell/
[20] This number can be varied by changing '`-n 120`' in supercell.command.

patterns with a pattern plotted from the original CIF. Friedel pairs, *i.e.*, hkl and $\overline{hkl}$ reflections, should strictly be generated by setting a variable `LPAIR` at 1 in template.ins owing to adoption of the noncentrosymmetric space group, $P1$, for derivative structures.

5.4 Calculation of electrostatic energies

To evaluate electrostatic energies, oxidation numbers must be given for all the sites in hoge.cif, *e.g.*, in the case of Al2O5Ti.cif as follows:

```
loop_
_atom_type_symbol
_atom_type_oxidation_number
O1 -2
Al1 3
Ti2 4
Al2 3
Ti1 4
O2 -2
O3 -2
#Ti4+ 4
#Al3+ 3
#O2- -2
loop_
_atom_site_label
_atom_site_type_symbol
_atom_site_symmetry_multiplicity
_atom_site_Wyckoff_symbol
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_atom_site_B_iso_or_equiv
_atom_site_occupancy
_atom_site_attached_hydrogens
Ti1 Ti4+ 4 c 0.1854(1) 0.25 0 . 0.333 0
Al1 Al3+ 4 c 0.1854(1) 0.25 0 . 0.667 0
Ti2 Ti4+ 8 f 0.13478(8) 0.56150(8) 0 . 0.333 0
Al2 Al3+ 8 f 0.13478(8) 0.56150(8) 0 . 0.667 0
O1 O2- 4 c 0.7577(3) 0.25 0 . 1. 0
O2 O2- 8 f 0.0485(2) 0.1167(2) 0 . 1. 0
O3 O2- 8 f 0.3125(2) 0.0721(2) 0 . 1. 0
```

When dealing with CIFs exported from ICSD by cifconv.command, such lines are automatically inserted provided that a variable, SUPERCELL, is set at 1 (see 7.10) in the main program. On the

other hand, lines containing data corresponding to `_atom_site_type_symbol` are commented out. Although the part corresponding to `_atom_type_symbol` and `_atom_type_-oxida- tion_number` is not strictly correct but odd, the above CIF is useful to obtain CIFs of derivative structures by supercell and BVSs by bond_str.

Processing Al2O5Ti.cif by the supercell macro gives Al2O5Ti∗.cif and Al2O5Ti_coulomb_-energy_l.txt in directory AxBxC. In the latter file, names of twenty CIFs and Coulomb energies calculated by the Ewald method for them are recorded in descending order, *e.g.*, in the case of a translation vector, $\boldsymbol{a} + \boldsymbol{b} + 2\boldsymbol{c}$,

```
1x1x2/Al2O5Ti_il002012_w32.cif  -2646.189 eV
1x1x2/Al2O5Ti_il002106_w32.cif  -2646.171 eV
1x1x2/Al2O5Ti_il004104_w32.cif  -2645.985 eV
1x1x2/Al2O5Ti_il001122_w32.cif  -2645.959 eV
1x1x2/Al2O5Ti_il002095_w32.cif  -2645.898 eV
1x1x2/Al2O5Ti_il001133_w32.cif  -2645.846 eV
1x1x2/Al2O5Ti_il001173_w32.cif  -2645.799 eV
1x1x2/Al2O5Ti_il002146_w32.cif  -2645.739 eV
1x1x2/Al2O5Ti_il001857_w32.cif  -2645.651 eV
1x1x2/Al2O5Ti_il002158_w32.cif  -2645.610 eV
1x1x2/Al2O5Ti_il000919_w32.cif  -2645.584 eV
1x1x2/Al2O5Ti_il004849_w32.cif  -2645.535 eV
1x1x2/Al2O5Ti_il007456_w32.cif  -2645.529 eV
1x1x2/Al2O5Ti_il001008_w32.cif  -2645.489 eV
1x1x2/Al2O5Ti_il000806_w32.cif  -2645.489 eV
1x1x2/Al2O5Ti_il003751_w32.cif  -2645.483 eV
1x1x2/Al2O5Ti_il002174_w32.cif  -2645.462 eV
1x1x2/Al2O5Ti_il001737_w32.cif  -2645.460 eV
1x1x2/Al2O5Ti_il002007_w32.cif  -2645.436 eV
1x1x2/Al2O5Ti_il001095_w32.cif  -2645.424 eV
```

In the resulting CIFs of derivative structures, an at sign, '@', plus a serial number is attached to each site label (`_atom_site_label`) while `_atom_site_type_symbol` is placed at the second field. A Coulomb energy appears as the second line in each CIF.

The top part of sites for the derivative structure with the smallest Coulomb energy of $-2646.189\,\mathrm{eV}$ is as follows:

```
loop_
    _atom_site_label
    _atom_site_type_symbol
    _atom_site_fract_x
    _atom_site_fract_y
    _atom_site_fract_z
    _atom_site_occupancy
```

```
Ti1@1     Ti1    0.18540   0.25000   0.00000   1.000
Ti1@2     Ti1    0.81460   0.75000   0.00000   1.000
Ti1@3     Ti1    0.68540   0.25000   0.25000   1.000
Al1@4     Al1    0.31460   0.75000   0.25000   1.000
Al1@5     Al1    0.18540   0.25000   0.50000   1.000
.....
```

Oxidation numbers of site symbols (second fields) are obtainable from `_atom_type_symbol` in the upper part of this CIF.

Of course, the crystal structure recorded in each CIF can be displayed with VESTA directly or after opening it by Jedit Ω pro and selecting VESTA in the Macro menu. VESTA is further capable of simulating a powder diffraction pattern in collaboration with RIETAN-FP, which makes it possible to compare the resulting pattern with an observed one for the screening of candidates for structural models. Be aware that voxel numbers along $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ directions in a template file of hoge.ins, whcih is specified in the Preferences dialog box of VESTA, must be set at 0.

The supercell macro can deal with any CIFs (*e.g.*, those output by VESTA and lst2cif) where site names, oxidation numbers, fractional coordinates, and occupancies are recorded; for details, see an Evernote note.[21] Reference [2] describes applications of cifconv.command and the supercell macro to representative cathode-active materials: $LiCoO_2$ and $Li(Ni_{1/3}Mn_{1/3}Co_{1/3})O_2$.

5.5 Single-step generation of derivative structures

If a comment line, '`#supercell AxBxC`', specifying the size of a derivative structure is inserted at the top of hoge.cif which is located not in the hoge_sc directory but in the current directory on use of cofconv.command, hoge.cif slightly modified as described above is output in the hoge_sc directory. Next, twenty CIFs of derived structures are output by supercell in the hoge_sc/AxBxC directory. That is, the supercell macro need not be run on the assistance environment after executing cifconv.command. This alternative procedure is not recommended when AxBxC is relatively large or when two or more '`#supercell AxBxC`' lines are given because of a long computation time.

6 File conversion by C-Tools

C-Tools has a feature of read an INCAR file of VASP [10] to input a variety of data *via* its user-friendly graphical user interface (GUI), where three tabs can be switched: Unit Lattice, Reciprocal Lattice, and UDF setup. The Reciprocal Lattice dialog is useful for viewing reciprocal-lattice shape including K points. When a suprecell macro under Others in the Macro menu is selected while displaying INCAR, input files for VASP can be read in by C-Tools. **Figure 1** shows a dialog to input data for creating hoge.udf with a common UDF format. The manual of C-Tools is obtainable from here.

Though C-Tools supports the direct import of CIFs, it often fails in reading them without

[21] https://bit.ly/2VGUDLI.

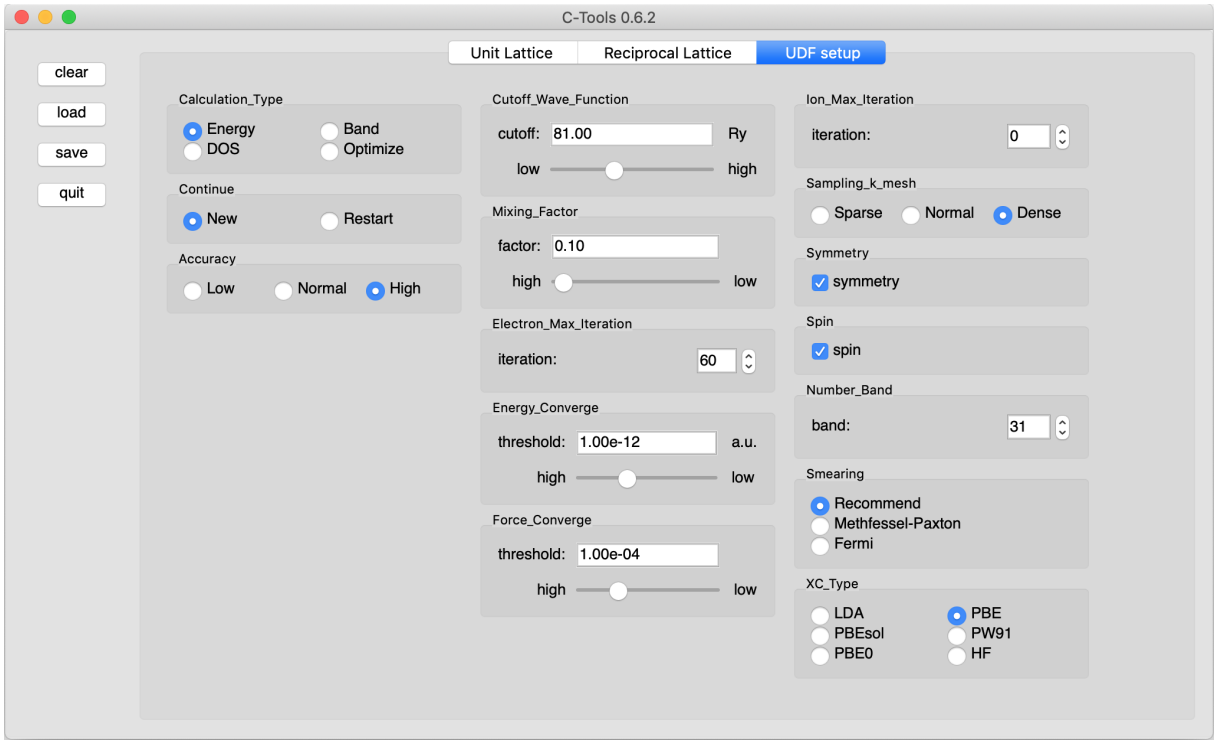

Figure 1: A UDF setup dialog in C-Tools after loading a CIF of La_2CuO_4. A calculation type, Energy, is checked with an accuracy mode of High.

any error message. Non-standard axes settings particularly prevent the normal input of CIFs. In contrast, cifconv.command is capable of converting CIFs most surely into the four input files (INCAR, POSCAR, POTCAR, and KPOINTS) of VASP thanks to the Python library PyCifRW. By clicking a [save] button, C-Tools can export input files of VASP [10], Quantum ESPRESSO [17], OpenMX [21], xTAPP [22], and RSDFT [23] for density functional theory (DFT) calculations, which makes it possible to convert INCAR *etc.* safely into input files for the other four DFT programs. The preprocessing of CIFs by cifconv.command, therefore, provides us with convenient and flexible means of converting INCAR *etc.* output by cif2cell into input files for Quantum ESPRESSO, OpenMX, xTAPP, and RSDFT.

With C-Tools, conversion of file formats is also possible by command-line specification of input and output files. For example, INCAR is converted into hoge.in for Quantum ESPRESSO with the following command:

```
c-tools -vasp INCAR -qe hoge.in
```

Written in C++, C-Tools can convert files for the above applications fairly fast, which makes conversion of INCAR practicable.

Unfortunately, INCAR output by C-Tools is too simple to use it without adding many tags. Hence, overwriting INCAR resulting from cifconv.command is never recommended. On the other hand, input files for Quantum ESPRESSO, OpenMX, xTAPP, and RSDFT exported by C-Tools contain fairly rich contents. On execution of the cifconv macros, INCAR is therefore automatically converted into hoge.in for Quantum ESPRESSO, hoge.dat for OpenMX, hoge.cg

for xTAPP, and fort.1[22] for RSDFT by C-Tools if four variables, *i.e.*, `QE`, `OpenMX`, `xTAPP`, and `RSDFT`, are, respectively, set at `1` (see Sect. 7).

7 Shell variables

The shell variables, *i.e.*, `SYMMETRY`, `MADEL`, `OX_NUM`, `CIF2CELL`, `QE`, `OpenMX`, `xTAPP`, `RSDFT`, `SUPERCELL`, `MAKE_INS`, `dmax_dis`, and `REGION`, are defined in the main program (final part) of cifconv.command. Needless to say, their values may be freely changed to control the behavior of cifconv.command during its execution.

7.1 SYMMETRY

Our three-dimensional visualization program, VESTA, as well as a utility, lst2cif, output CIFs where formal definitions `_space_group_crystal_system`, `_space_group_IT_number`, `_space_group_name_H-M_alt`, `_space_group_name_Hall`, `_space_group_symop_operation_xyz`, and `_space_group_symop_id` [24] are included. Unfortunately, many programs can input only superseded definitions[23] `_symmetry_...`, which have been still included in most CIFs [2]. For example, CIFs exported by four databases, ICSD, PDF-4+, the Materials Project,[24] and MatNavi,[25] contain `_symmetry_...` definitions. Furthermore, MedeA[26] and C-Tools[27] cannot input any CIFs including `_space_group_...` definitions, which are recorded in CIFs output by lst2cif[28] and exported by VESTA.

To overcome such serious problems, cifconv.command makes it possible to substitute the following superseded definitions for formal ones, `_space_group_...`, in CIFs for backward compatibility:

- `_symmetry_cell_setting` ← `_space_group_crystal_system`
- `_symmetry_Int_Tables_number` ← `_space_group_IT_number`
- `_symmetry_space_group_name_H-M` ← `_space_group_name_H-M_alt`
- `_symmetry_space_group_name_Hall` ← `_space_group_name_Hall`
- `_symmetry_equiv_pos_site_id` ← `_space_group_symop_id`
- `_symmetry_equiv_pos_as_xyz` ← `_space_group_symop_operation_xyz`

To utilize this feature, a variable, `SYMMETRY`, should be defined not as `0` but as `1` in the main program. Such backward substitutions provide us with CIFs that can be directly input by MedeA, C-Tools, *etc.*

[22] In addition to fort.1, fort.970 and symdat are also output.
[23] https://www.iucr.org/__data/iucr/cifdic_html/1/cif_core.dic/index.html
[24] https://materialsproject.org/
[25] http://mits.nims.go.jp/
[26] https://www.materialsdesign.com/medea-software
[27] https://sourceforge.net/projects/c-tools/
[28] A macro contained in the integrated assistance environment for the RIETAN–VENUS system.

7.2 MADEL

With cifconv.command, input files for a Fortran program, MADEL, can be obtained to calculate site potentials and a Madelung energy by the Fourier method [8, 9]. This feature is made effective by setting a variable, `MADEL`, not at `0` but at `1`.

7.3 CRYSCALC

If `CRYSCALC` is defined as `1`, CRYSCALC converts a CIF into hoge.cco containing a variety of crystallographic information.

7.4 OX_NUM

`OX_NUM` need to be input if `MADEL=0` (7.2) while `OX_NUM` is automatically set at 1 if `SUPERCELL=1` (7.10). Unless oxidation numbers of chemical species are included in CIFs, bond_str cannot calculate any bond valence sums. If `OX_NUM` is set not at `0` but at `1`, cifconv.command asks whether or not parts of oxidation numbers are changed by the user after opening CFL_file.cfl containing default values of oxidation numbers by Jedit Ω pro. After editing it if necessary, press the macOS key to start calculating BVSs.

7.5 CIF2CELL

If `CIF2CELL` is defined as `1`, cif2cell [4] is run to convert a CIF, hoge.cif, into INCAR, POSCAR, POTCAR, and KPOINTS for VASP to move them into directory hoge_VASP [10].

7.6 QE

If `QE` is defined as `1`, C-Tools is run to convert INCAR and POSCAR into hoge.in for Quantum ESPRESSO [17]. `QE` should be set at 1 to use this feature.

7.7 OpenMX

If `OpenMX` is defined as `1`, C-Tools is run to convert INCAR and POSCAR into hoge.dat for OpenMX [21]. `OpenMX` should be set at 1 to use this feature.

7.8 xTAPP

If `xTAPP` is defined as `1`, C-Tools is run to convert INCAR and POSCAR into hoge.cg for xTAPP [22]. `xTAPP` should be set at 1 to use this feature.

7.9 RSDFT

If `RSDFT` is defined as `1`, C-Tools is run to convert INCAR and POSCAR into hoge_RSDFT/fort.1 for RSDFT [23]. `RSDFT` should be set at 1 to use this feature.

7.10 SUPERCELL

If `SUPERCELL` is defined as `1`, a CIF named hoge.cif, which is input by supercell [20] to generate potential candidates of derivative structures, is output in subdirectory hoge_sc. The subsequent execution of macro supercell while displaying the resulting hoge.cif in Jedit Ω outputs CIFs of derivative structures.

7.11 MAKE_INS

Setting `MAKE_INS` not at `0` but at `1` outputs ∗.ins for RIETAN-FP by changing parts of a template file: template.ins.

7.12 VESTA

Setting `VESTA` not at `0` but at `1` makes it possible to draw crystal structures from crystal data in CIFs by VESTA [11] in the batch mode.

7.13 dmax_dis

`dmax_dis` is the maximum interatomic distance (in Å) computed and output in hoge.dab by bond_str.

7.14 REGION

On use of MADEL, `REGION` should be set at a reciprocal-space range (in Å^{-1}) within which Fourier coefficients are summed up. MADEL sums up the Fourier coefficients with respect to all *hkl*'s within a sphere having a radius equal to another input data: RADIUS. Choose an appropriate value within the range 2.0–4.0 Å^{-1} according to the desired precision of calculation. In addition, check whether or not a curve for Madelung energy versus `REGION` is nearly flat around the selected value of `REGION`.

8 How to modify hoge.pme for MADEL

Kato *et al.* [25] has recently revealed that site potentials calculated by MADEL proved to be effective in capturing essential features of valence band structures, which is the reason why the feature of sequential calculations of site potentials by MADEL from CIFs exported from ICSD or output by supercell was implemented in cifconv.command.

The radius of the ionic sphere, RADIUS, is usually calculated at the minimum interatomic distance (in a unit of Å) minus 0.01. If the minimum distance calculated by bond_str is less than 1.0 Å for any reason, RADIUS is set at 1.0 Å for convenience sake.

When dealing with a nonstoichiometric compound where two or more elements occupy the same position with occupancies less than unity, it is strongly recommended to locate a virtual chemical species consisting of the elements at that position with $g = 1$. Then, the occupancy of the virtual chemical species is simply regarded as 1 on calculation of both Z(I) and W(I), which makes it secure to calculate site potentials. Unfortunately, we have devised no safe procedure to process compounds where parts of sites contain vacancies.

Append a line where four integers (NPOT, NX, NY, and NZ) are input in free format to output a voxel file, hoge.ped, which can be input by VESTA to visualize isosurfaces of potentials in three dimensions.

NPOT = 1: Output potentials (POBBA) without any contributions from neighboring atoms.
NPOT = 2: Output potentials (PMBBA) containing contributions from neighboring atoms.
NX, NY, and NZ: Division numbers along a, b, and c axes.

When a voxel is located within an ionic sphere, MADEL outputs a potential (POBBA) excluding contribution of the sphere and, in addition, a potential (PMBBA) calculated by substituting an original point charge for the sphere. If the site is not contained in any ionic sphere, a potential in which contribution of the nearest neighbor (either one if two or more nearest neighbors are present) is subtracted is output as POBBA. On use of this feature, RADIUS should be set in such a way as not to overlap with each other: less than half the smallest interatomic distance in the whole structure.

Execution of MADEL to process a CIF exported by ICSD produces hoge.pcb, which includes oxidation numbers (`V`) and site potentials (`phi`) in addition to coordination numbers (`CN`) and BVSs [6] output by bond_str in hoge.bvs. Let us suppose that a site is occupied by ions, A_i ($i = 1$–n), with occupancies of g_i while A_i is bonded to counter ions, X_j, with occupancies of G_j. Then, the BVS, ν_i, of the site is calculated by

$$\nu_i = \sum_i^n g_i \sum_j \nu_{ij} = \sum_i^n g_i \sum_j G_j \exp\left(\frac{R_{ij} - d_{ij}}{B}\right),$$

where ν_{ij} is the bond valence for an A_i and X_j pair, R_{ij} is the bond valence parameter for the combination of A_i and X_j, d_{ij} is the A_i–X_j bond length in Å, and $B = 0.37$ Å.[29] Both R_{ij} and B are empirical parameters compiled in bvparm∗.cif.[30] Oxidation numbers included in bvparm2016.cif must be given in CIFs processed by cifconv.command. BVSs are useful for confirming and considering the validity and stability of crystal structures [26].

In case of $BaTi_2O_5$, hoge.pcb includes the following table:

Site	V	Site	x	y	z	g	phi	CN	BVS
Ba1	2	4i	0.48223	0.00000	0.13107	1.000	-1.555852	9.00	2.250(9)
Ba2	2	2a	0.00000	0.00000	0.00000	1.000	-1.281211	12.00	2.446(8)
Ti1	4	4i	0.19441	0.00000	0.66610	1.000	-2.985741	6.00	4.052(20)
Ti2	4	4i	0.12731	0.00000	0.29292	1.000	-2.759538	6.00	3.968(21)
Ti3	4	4i	0.28946	0.00000	0.46098	1.000	-3.116296	6.00	3.806(15)
O1	-2	4i	0.16250	0.00000	0.17490	1.000	1.905389	4.00	1.827(8)
O2	-2	4i	0.10150	0.00000	0.76510	1.000	2.217431	4.00	2.062(10)
O3	-2	4i	0.29020	0.00000	0.96460	1.000	1.555682	5.00	1.817(5)
O4	-2	4i	0.31540	0.00000	0.34720	1.000	1.672874	4.00	2.324(17)
O5	-2	2c	0.50000	0.00000	0.50000	1.000	1.539246	6.00	2.026(7)
O6	-2	4i	0.21070	0.00000	0.55760	1.000	1.716391	4.00	2.048(13)
O7	-2	4i	0.07100	0.00000	0.39110	1.000	1.847775	5.00	1.914(15)
O8	-2	4i	0.37800	0.00000	0.71130	1.000	1.331919	4.00	2.293(18)

Needless to say, an electrostatic (Madelung) energy per asymmetric unit recorded in hoge.mad is also output along with some crystal data. Orders of sites in hoge.bvs are often changed from

[29] https://www.jstage.jst.go.jp/article/jcrsj/57/1/57_72/_pdf
[30] https://www.iucr.org/resources/data/datasets/bond-valence-parameters

those in CIFs for unknown reasons. Such integration of hoge.mad and hoge.bvs is, therefore, very convenient.

9 Uniprocessing mode

The batch mode of cifconv.command described above was designed to convert two or more CIFs sequentially. On the other hand, its uniprocessing mode deals with only one CIF. On demand from the Computational Materials Group of JFCC, this feature was implemented in the integrated assistance environment for RIETAN–VENUS to save budgets for expensive commercial software packages, *e.g.*, MedeA and Materials Studio.

As described in Sect. 3, the cifconv macro can be selected under ContextMenu in the Macro menu or in the context menu. A template file of INCAR and hoge.cif have to share the same directory in the same way as the batch mode. After hoge.cif has been opened by Jedit Ω pro, selecting the cifconv macro starts to run cifconv.command in the uniprocessing mode to produce POSCAR, POTCAR, KPOINTS, and INCAR under the hoge_VASP subdirectory if the variable, CIF2CELL, is set at 1 in cifconv.command. The resulting four files under subdirectory hoge_VASP are automatically opened by Jedit Ω pro for browsing and editing them. Unless cifconv.command exists in the current directory, it is copied from directory /Applications/RIETAN_VENUS/commands_common to the current directory for later use of the batch mode as described in Sect. 3.

The assistance environment supports close cooperation with C-Tools. After selecting a C-Tools macro under Others in the Macro menu while INCAR is displayed, INCAR and the other three files are input by C-Tools. Then, the GUI of C-tools as illustrated in Fig. 1 can be utilized to visualize change and input various data.

Both the visualization technology and crystallographic features implemented in VESTA are believed to be much more excellent than any other free and commercial applications. The author strongly hopes that this new feature of the assistance environment will contribute in a significant way to research and development at many research organizations.

10 On a related utility change_hoge.command

When exporting CIFs from databases such as ICSD, tentative file names, ∗.cif, are usually given for labor-saving. A convenient shell script, change_hoge.command, which is placed in each subdirectory of RIETAN_VENUS_examples/CIF_ICSD, was developed to rename ∗.cif automatically. It deal with all CIFs in the following way:

1. a CIF in the current directory are input,
2. data after '`_chemical_formula_sum`' are read,
3. spaces in the data are deleted,
4. A redundant character, '1', corresponding to a subscript after each element name in the data are deleted if any,

5. '_n' ($n = 1, 2, 3, ...$) is attached to avoid the same file name in the current directory if necessary,

6. an extension, '.cif', is appended to give a new file name.

The above procedure is repeated for all the CIFs in the current directory. For example, a line, `_chemical_formula_sum 'K1 Na2 Ni1 O2'`, in a CIF changes its name to KNa2NiO2.cif. Consequently, a chemical composition can be immediately learned from the file name of each CIF, which is a great advantage when dealing with many CIFs.

Because a comment line

`#(C) 2018 by FIZ Karlsruhe - ..... Infrastructure.  All rights reserved.`

at the top part of a CIF exported by ICSD is longer than 80 characters, it is shortened into the following line:

`#(C) 2018 by FIZ Karlsruhe - Leibniz Institute for Information Infrastructure`

Unfortunately, names of CIFs exported by the Materials Project[31] often include a pair of parentheses: '(' and ')'. Because shell scripts cannot process such CIFs, it is highly recommended to run change_hoge.command prior to conversion of CIFs is highly recommended.

References

[1] H. J. Bernstein, J. C. Bollinger, I. D. Brown, S. Gražulis, J. R. Hester, B. McMahon, N. Spadaccini, J. D. Westbrookh, and S. P. Westripi, *J. Appl. Crystallogr.*, **49**, 277 (2011).

[2] F. Izumi and K. Miyazaki, *Bull. Ceram. Soc. Jpn.*, **54**, 473 (2019).

[3] T. Yamamori, "Shell Script Basic References," Revised 3rd ed., Gijutsu-Hyohron Co., Tokyo (2017).

[4] T. Björkman, *Comput. Phys. Commun.*, **182**, 1183 (2011).

[5] S. R. Hall, *Acta Crystallogr., Sect. A: Found. Adv.*, **37**, 517 (1981).

[6] I. D. Brown and D. Altermatt, *Acta Crystallogr., Sect. B: Struct. Sci., Cryst. Eng. Mater.*, **41**, 244 (1985).

[7] J. Rodríguez-Carvajal, *Phys. B (Amsterdam, Neth.)*, **192**, 55 (1993).

[8] E. F. Bertaut, *Phys. Rev.*, **91**, 415 (1953).

[9] J. Kanamori, T. Moriya, K. Motizuki, and T. Nagamiya, *J. Phys. Soc. Jpn.*, **10**, 93 (1955).

[10] J. Hafner, *Comput. Phys. Commun.*, **177**, 6 (2007).

[11] K. Momma and F. Izumi, *J. Appl. Crystallogr.*, **44**, 1272 (2011).

[12] F. Izumi and K. Momma, *Solid State Phenom.*, **130**, 15 (2007).

[31] https://materialsproject.org/

[13] E. Parthé and L. M. Gelato, *Acta Crystallogr., Sect. A: Found. Crystallogr.*, **40**, 169 (1984).

[14] S. Adams, *Solid State Ionics*, **136**–**137**, 1351 (2000).

[15] S. Adams, *Acta Crystallogr., Sect. B: Struct. Sci., Cryst. Eng. Mater.*, **57**, 278 (2001).

[16] S. Adams and R. P. Rao, *Phys. Status Solidi A*, **208**, 1746 (2011).

[17] P. Giannozzi *et al.*, *J. Phys.: Condens. Matter*, **21**, 395502 (2009); P. Giannozzi *et al.*, *ibid.*, **29**, 465901 (2017).

[18] G. Kresse and J. Joubert, *Phys. Rev. B: Condens. Matter Mater. Phys.*, **59**, 1758 (1999).

[19] J. A. Gomes, M. H. Sousa, F. A. Tourinho, J. Mestnik-Filho, R. Itri, and J. Depeyrot, *J. Magn. Magn. Mater.*, **289**, 184 (2005).

[20] K. Okhotnikov, T. Charpentier, and S. Cadars, *J. Cheminform.*, **8**, 17 (2016).

[21] T. Ozaki, *Phys. Rev. B: Condens. Matter Mater. Phys.*, **67**, 155108 (2003); T. Ozaki and H. Kino, *ibid.*, **69**, 195113 (2004); T. Ozaki and H. Kino, *ibid.*, **72**, 045121 (2005).

[22] J. Yamauchi, M. Tsukada, S. Watanabe, and O. Sugino, *Phys. Rev. B: Condens. Matter Mater. Phys.*, **54**, 5586 (1996); O. Sugino and A. Oshiyama, *Phys. Rev. Lett.*, **68**, 1858 (1992).

[23] J. Iwata, D. Takahashi, A. Oshiyama, T. Boku, K. Shiraishi, S. Okada, and K. Yabana, *J. Comput. Phys.*, **229**, 2339 (2010).

[24] S. R. Hall, F. H. Allen, and I. D. Brown, "International Tables for Crystallography," Vol. G: Definition and exchange of crystallographic data (2005), 4.1.

[25] D. Kato, K. Hongo, R. Maezono, M. Higashi, H. Kunioku, M. Yabuuchi, H. Suzuki, H. Okajima, C. Zhong, K. Nakano, R. Abe, and H. Kageyama, *J. Am. Chem. Soc.*, **139**, 18725 (2017).

[26] I. D. Brown, "The Chemical Bond in Inorganic Chemistry: The Bond Valence Model," Oxford University Press, Oxford (2002).