
Cif2x Documentation

リリース 1.1.0

ISSP, University of Tokyo

2024 年 09 月 10 日

目次

第 1 章	概要	1
1.1	cif2x とは?	1
1.2	ライセンス	1
1.3	開発貢献者	1
1.4	バージョン履歴	2
1.5	コピーライト	2
1.6	動作環境	3
第 2 章	インストールと基本的な使い方	4
第 3 章	チュートリアル	8
3.1	入力パラメータファイルを作成する	8
3.2	第一原理計算入力ファイルを生成する	10
3.3	パラメータセットを指定する	11
第 4 章	コマンドリファレンス	12
4.1	cif2x	12
第 5 章	ファイルフォーマット	14
5.1	入力パラメータファイル	14
5.2	Quantum ESPRESSO 向けパラメータ	16
5.3	VASP 向けパラメータ	18
5.4	OpenMX 向けパラメータ	20
5.5	AkaiKKR 向けパラメータ	20
第 6 章	拡張ガイド	22
6.1	Quantum ESPRESSO の mode を追加する	22
第 7 章	CIF データ取得ツール (getcif)	24
7.1	概要	24
7.2	チュートリアル	24
7.3	コマンドリファレンス	28
7.4	ファイルフォーマット	29
7.5	パラメータリスト	32

第 1 章

概要

1.1 cif2x とは?

近年、機械学習を活用した物性予測や物質設計 (マテリアルズインフォマティクス) が注目されています。機械学習の精度は、適切な教師データの準備に大きく依存しています。そのため、迅速に教師データを生成するためのツールや環境の整備は、マテリアルズインフォマティクスの研究進展に大きく貢献すると期待されます。

cif2x は、cif ファイルから第一原理計算用の入力ファイルを生成するツールです。入力パラメータを雛形として、物質の種類や計算条件によって変わる箇所を結晶構造データなどから構成します。特定の計算条件に応じた複数の入力ファイルを生成することが可能です。現在は、[VASP](#), [Quantum ESPRESSO](#), [OpenMX](#), [AkaiKKR](#) に対応しています。

付属プログラムとして、物質材料データベースから結晶構造データ等を取得するツール `getcif` を用意しています。現在は Materials Project からのデータ取得に対応しています。物質の組成や対称性、バンドギャップなどの物性値をもとにデータベースを検索し、データを取得することができます。

1.2 ライセンス

本ソフトウェアのプログラムパッケージおよびソースコード一式は GNU General Public License version 3 (GPL v3) に準じて配布されています。

1.3 開発貢献者

本ソフトウェアは以下の開発貢献者により開発されています。

- 開発者
 - 吉見 一慶 (東京大学 物性研究所)
 - 青山 龍美 (東京大学 物性研究所)

- 本山 裕一 (東京大学 物性研究所)
- 福田 将大 (東京大学 物性研究所)
- 井戸 康太 (東京大学 物性研究所)
- 福島 鉄也 (産業技術総合研究所)
- 笠松 秀輔 (山形大学 学術研究院 (理学部主担当))
- 是常 隆 (東北大学大学院理学研究科)
- プロジェクトコーディネーター
 - 尾崎 泰助 (東京大学 物性研究所)

1.4 バージョン履歴

ver.1.1.0
2024/09/09 リリース

ver.1.0.1
2024/03/31 リリース

ver.1.0.0
2024/03/19 リリース

ver.1.0-alpha
2023/12/28 リリース

1.5 コピーライト

© 2023- The University of Tokyo. All rights reserved.

本ソフトウェアは 2023 年度 東京大学物性研究所 ソフトウェア高度化プロジェクトの支援を受け開発されており、その著作権は東京大学が所持しています。

1.6 動作環境

以下の環境で動作することを確認しています。

- Ubuntu Linux + python3

第 2 章

インストールと基本的な使い方

必要なライブラリ・環境

HTP-tools に含まれる第一原理計算入力ファイル生成ツール cif2x を利用するには、以下のプログラムとライブラリが必要です。

- python 3.x
- pymatgen モジュール
- ruamel.yaml モジュール
- f90nml モジュール
- qe-tools モジュール
- numpy モジュール
- pandas モジュール
- monty モジュール
- OpenBabel モジュール (オプション)
- AkaiKKRPythonUtil モジュール

CIF データ取得ツール getcif を利用するには、上記に加えて以下のライブラリが必要です。

- mp-api モジュール

ソースコード配布サイト

- [GitHub](#) リポジトリ

ダウンロード方法

git を利用できる場合は、以下のコマンドで cif2x をダウンロードできます。

```
$ git clone https://github.com/issp-center-dev/cif2x.git
```

インストール方法

cif2x をダウンロード後、以下のコマンドを実行してインストールします。cif2x が利用するライブラリも必要に応じてインストールされます。

```
$ cd ./cif2x
$ python3 -m pip install .
```

実行プログラム cif2x, getcif がインストールされます。

なお、AkaiKKRPythonUtil モジュールは別途インストールが必要です。以下の手順で [配布サイト](#) からソースコードを取得しインストールします。また、必要な seaborn モジュールもインストールしておきます。

```
$ git clone https://github.com/AkaiKKRteam/AkaiKKRPythonUtil.git
$ cd ./AkaiKKRPythonUtil/library/PyAkaiKKR
$ python3 -m pip install .
$ python3 -m pip install seaborn
```

ディレクトリ構成

```
.
|-- LICENSE
|-- README.md
|-- pyproject.toml
|-- docs/
|   |-- ja/
|   |-- tutorial/
|-- src/
|   |-- cif2x/
|       |-- __init__.py
|       |-- main.py
|       |-- cif2struct.py
|       |-- struct2qe.py
|       |-- qe/
|           |-- __init__.py
|           |-- calc_mode.py
|           |-- cards.py
|           |-- content.py
|           |-- qeutils.py
```

(次のページに続く)

(前のページからの続き)

```

|         |-- tools.py
|         |-- struct2vasp.py
|         |-- struct2openmx.py
|         |-- openmx/
|         |   |-- __init__.py
|         |   |-- vps_table.py
|         |-- struct2akaikkr.py
|         |-- akaikkr/
|         |   |-- make_input.py
|         |   |-- read_input.py
|         |   |-- run_cif2kkkr.py
|         |-- utils.py
| |-- getcif/
|     |-- __init__.py
|     |-- main.py
|-- sample/

```

基本的な使用方法

cif2x は第一原理計算プログラムのための入力ファイルを生成するツールです。入力パラメータを雛形として、物質の種類や計算条件によって変わる箇所を結晶構造データなどから構成します。現在は Quantum ESPRESSO, VASP, OpenMX, および AkaiKKR の入力ファイル形式に対応しています。

1. 入力パラメータファイルの作成

cif2x を使用するには、まず、生成する入力ファイルの内容を記述したパラメータファイルを YAML 形式で作成します。詳細についてはファイルフォーマットの章を参照してください。

2. 結晶構造ファイルと擬ポテンシャルファイルの配置

対象となる物質の結晶構造を記述したファイルを用意します。ファイル形式は CIF または pymatgen が扱える POSCAR や xfs 形式に対応しています。

Quantum ESPRESSO の場合、利用する擬ポテンシャルファイルと、CSV 形式のインデックスファイルを配置します。擬ポテンシャルファイルの配置先などは入力パラメータファイル内に指定します。

VASP の場合、擬ポテンシャルファイルの格納場所を `~/.config/.pmgrc.yaml` ファイルに記述するか環境変数にセットします。入力パラメータファイル内で指定することもできます。

3. コマンドの実行

作成した入力パラメータファイルおよび結晶構造データファイルを入力として cif2x プログラムを実行します。Quantum ESPRESSO 用の入力ファイルを生成する場合はターゲットオプションに

-t QE を指定します。VASP の場合は -t VASP, OpenMX の場合は -t OpenMX, AkaiKKR の場合は -t akaikkr を指定します。

```
$ cif2x -t QE input.yaml material.cif
```

第 3 章

チュートリアル

第一原理計算入力ファイル生成ツール `cif2x` を使うには、入力パラメータファイルと結晶構造データおよび擬ポテンシャルファイルを用意した後、プログラム `cif2x` を実行します。現在は Quantum ESPRESSO, VASP, OpenMX, AkaiKKR の入力ファイル生成に対応しています。以下では、`docs/tutorial/cif2x` ディレクトリにある Quantum ESPRESSO 向けサンプルを例にチュートリアルを実施します。

3.1 入力パラメータファイルを作成する

入力パラメータファイルには、第一原理計算プログラムに与える入力ファイルの内容を記述します。

以下に入力パラメータファイルのサンプルを記載します。このファイルは YAML 形式のテキストファイルで、結晶構造データに対するオプションの指定や、出力する第一原理計算入力ファイルの内容を記述します。仕様の詳細については [ファイルフォーマット](#) の章を参照してください。

YAML フォーマットでは、`keyword: value` の辞書形式でパラメータを記述します。value には数値や文字列などのスカラー値や、複数の値を `[]` または箇条書きの形式で列挙するリスト型、または辞書型を入れ子にすることができます。

```
structure:
  use_ibrav: false
  tolerance: 0.05

optional:
  pseudo_dir: ./pseudo
  pp_file: ./pseudo/pp_psl_pbe_rrkjus.csv

tasks:
  - mode: scf
    output_file: scf.in
```

(次のページに続く)

(前のページからの続き)

```

output_dir: scf
template: scf.in_tmpl
content:
  namelist:
    control:
      prefix: pwscf
      pseudo_dir:
      outdir: ./work
    system:
      ecutwfc:
      ecutrho:
  CELL_PARAMETERS:
  ATOMIC_SPECIES:
  ATOMIC_POSITIONS:
    option: crystal
  K_POINTS:
    option: automatic
    grid: [8,8,8]

```

入力パラメータファイルは structure, optional, tasks のブロックから構成されます。structure は結晶構造データに関するオプションを指定します。optional は擬ポテンシャルに関する global な設定などを行います。

tasks は出力する第一原理計算入力ファイルの内容を指定します。一連の計算に対応して複数のファイルを生成できるよう、tasks は配列の値を取ります。各出力について、計算内容は mode パラメータで指定します。SCF 計算の scf や NSCF 計算の nscf に対応するほか、一般的な出力を行う任意の出力モードを指定できます。ファイルは output_dir および output_file で指定するファイルに書き出されます。

出力内容は content に記載します。Quantum ESPRESSO の入力ファイルは、&keyword で始まる Fortran90 の namelist 形式と、K_POINTS などのキーワードで始まり空行で分割される cards と呼ばれるブロックからなります。content には namelist と cards を入れ子の辞書形式で指定します。いくつかの例外を除いて、指定された内容が基本的にはそのまま入力ファイルに書き出されます。値が空欄のキーワードは、結晶構造データなどから求めた値が代入されます。

また、template として入力ファイルの雛形を指定することもできます。template に指定したファイルの内容と content のデータを合わせたものを入力データとして扱います。同じキーワードのデータが存在する場合は content の指定が優先されます。従って、既存の入力ファイルを元に必要な箇所を入力パラメータファイルで上書きする使い方が可能です。上記の例では次のファイル (scf.in_tmpl) を template として取り込み、カットオフと CELL_PARAMETER, ATOMIC_SPECIES, ATOMIC_POSITIONS, K_POINT の箇所を結晶構造等から決めます。ecutwfc と ecutrho が空欄で上書きされていることに留意してください。

```

&control
  calculation = 'scf'
  prefix = 'pwscf'
  pseudo_dir = './pseudo'
  outdir = './work'
  tstress = .true.
  tprnfor = .true.
/

&system
 ibrav = 0
  nat = 7
  ntyp = 3
  ecutwfc = 36.0
  ecutrho = 180.0
  occupations = 'smearing'
  smearing = 'm-p'
  degauss = 0.01
  noncolin = .true.
  nspin = 2
/

&electrons
  missing_beta = 0.1
  conv_thr = 1e-08
/

```

3.2 第一原理計算入力ファイルを生成する

入力パラメータファイル (input.yaml) と結晶構造データ (Co3SnS2_nosym.cif) を入力として cif2x を実行します。

```
$ cif2x -t QE input.yaml Co3SnS2_nosym.cif
```

予め必要な擬ポテンシャルのファイルを ./pseudo ディレクトリに配置し、擬ポテンシャルのインデックスファイルを ./pseudo/pp_psl_pbe_rrkjus.csv に作成しておきます。

cif2x を実行すると Quantum ESPRESSO 用の入力ファイルが生成され出力されます。出力先は入力パラメータファイル内のパラメータで指定するディレクトリ (output_dir) およびファイル (output_file) です。この例では ./scf/scf.in に SCF 計算用の入力ファイルが書き出されます。

3.3 パラメータセットを指定する

入力パラメータ内の値をいくつか変えながら一連の入力ファイルを生成したいことがあります。例えばカットオフの値や k 点の数を変えて収束性を評価するなどの場合です。入力パラメータには値のリストや範囲を指定することができ、値の組み合わせごとに個別のディレクトリを作成して入力ファイルを生成します。パラメータセットの指定は特別な構文 `${...}` を用います。

```
content:
  K_POINTS:
    option: automatic
    grid:   ${ [ [4,4,4], [8,8,8], [12,12,12] ] }
```

例えば上記のように K_POINTS を指定すると、grid の値が [4,4,4], [8,8,8], [12,12,12] の入力ファイルがそれぞれ 4x4x4/, 8x8x8/, 12x12x12/ サブディレクトリ内に作成されます。

第 4 章

コマンドリファレンス

4.1 cif2x

第一原理計算のための入力ファイルを生成する

書式:

```
cif2x [-v][-q] -t target input_yaml material.cif
cif2x -h
cif2x --version
```

説明:

input_yaml に指定した入力パラメータファイルと material.cif に指定した結晶構造データを読み込み、第一原理計算プログラム用の入力ファイルを生成します。現在は Quantum ESPRESSO, VASP, OpenMX に対応しています。以下のオプションを受け付けます。

- -v

実行時に表示されるメッセージを冗長にします。複数回指定すると冗長度が上がります。

- -q

実行時に表示されるメッセージの冗長度を下げます。-v の効果を打ち消します。複数回の指定が可能です。

- -t *target*

対象となる第一原理計算プログラムを指定します。 *target* として指定可能な値は以下のとおりです。大文字小文字は区別しません。

- QE, espresso, quantum_espresso: Quantum ESPRESSO 向け入力ファイルを生成します。
- VASP: VASP 向け入力ファイルを生成します。

- OpenMX: OpenMX 向け入力ファイルを生成します。
 - AkaiKKR: AkaiKKR 向け入力ファイルを生成します。
- input.yaml
入力パラメータファイルを指定します。形式は YAML format です。
- material.cif
結晶構造データファイルを指定します。形式は CIF の他、pymatgen で扱える形式のファイルを指定できます。
- -h
ヘルプを表示します。
- --version
バージョン情報を表示します。

第 5 章

ファイルフォーマット

5.1 入力パラメータファイル

入力パラメータファイルでは、cif2x で第一原理計算入力ファイルを生成するための設定情報を YAML 形式で記述します。本ファイルは以下の部分から構成されます。

1. structure セクション: 結晶構造データの扱いについてのオプションを記述します。
2. optional セクション: 擬ポテンシャルファイルの指定や、YAML の参照機能を利用する場合のシンボル定義を行います。
3. tasks セクション: 入力ファイルの内容を記述します。

5.1.1 structure

use_ibrav (デフォルト値: false)

結晶構造の入力に Quantum ESPRESSO の ibrav パラメータを利用します。true の場合、格子のとり方を Quantum ESPRESSO の convention に合うように変換します。入力ファイルにはあわせて格子に関するパラメータ a, b, c, cosab, cosac, cosbc が (必要に応じて) 書き出されます。

tolerance (デフォルト値: 0.01)

use_ibrav = true の場合に、再構成した Structure データと元データとの一致を評価する際の許容度を指定します。

supercell (デフォルト値: なし)

supercell を設定する場合に supercell のサイズを $[n_x, n_y, n_z]$ で指定します。

5.1.2 optional

第一原理計算プログラムごとに必要な global な設定を行います。記述する内容は以下の各節に記述します。

5.1.3 tasks

入力ファイルの内容を記述します。複数の入力ファイルに対応するため、tasks には各入力ファイルごとのブロックからなるリスト形式をとります。各ブロックに記述する項目は以下のとおりです。

mode (Quantum ESPRESSO)

入力ファイルの計算内容を指定します。現時点では Quantum ESPRESSO の pw.x 向けに scf と nscf に対応しています。対応していない mode については、content の内容をそのまま出力します。

output_file (Quantum ESPRESSO)

出力ファイル名を指定します。

output_dir

出力先のディレクトリ名を指定します。デフォルト値はカレントディレクトリです。

content

出力内容を指定します。Quantum ESPRESSO の場合は namelist ブロックに namelist データ (&system や &control など) を記述し、K_POINTS 等の card データを個別のブロックとして記述します。card データはパラメータをとるものがあります。

template (Quantum ESPRESSO)

template_dir (VASP)

出力内容のテンプレートファイルを指定します。指定がない場合はテンプレートを利用しません。このファイルの内容を content に追加します。同じデータがある場合は後者を優先します。

5.1.4 パラメータセット指定

入力パラメータには値のリストや範囲を指定することができ、値の組み合わせごとに個別のディレクトリを作成して入力ファイルを生成します。パラメータセットの指定には特別な構文 `${...}` を用います。指定方法は次の通りです:

- リスト `${[A, B, ...]}`

パラメータセットを Python のリストとして列挙します。各項目はスカラー値やリストを指定できます。

- 範囲指定 `${range(N)}`, `${range(start, end, step)}`

パラメータの範囲を指定します。それぞれ 0 ~ N-1, start ~ end を step 刻み (step を省略した場合は 1) です。N, start, end, step は int または float です。

5.2 Quantum ESPRESSO 向けパラメータ

optional セクションおよび tasks セクションの content に記載する内容について、Quantum ESPRESSO 固有の内容を記述します。現時点では pw.x の scf および nscf に対応しています。

5.2.1 optional セクション

pp_file

元素種と擬ポテンシャルを対応付ける CSV 形式のインデックスファイルを指定します。ファイルの書式は、元素種、擬ポテンシャルファイルのタイプ、nexclude、orbitals です。例:

```
Fe,pbe-spn-rrkjus_psl.0.2.1,4,spd
```

擬ポテンシャルファイルのファイル名は Fe.pbe-spn-rrkjus_psl.0.2.1.UPF に対応します。

cutoff_file

擬ポテンシャルファイルとカットオフを対応付ける CSV 形式のインデックスファイルを指定します。ファイルの書式は、擬ポテンシャルファイル、ecutwfc の値、ecutrho の値 です。

pseudo_dir

擬ポテンシャルファイルを格納するディレクトリ名を指定します。カットオフの値を擬ポテンシャルファイルから取得する場合に使用します。Quantum ESPRESSO の pseudo_dir パラメータとは独立に指定します。

5.2.2 content

namelist

- structure セクションの use_ibrav パラメータに応じて、&system の格子情報の指定が上書きされます。
 - use_ibrav = false: ibrav は 0 にセットされます。また、格子パラメータに関する a, b, c, cosab, cosac, cosbc, celldm は削除されます。
 - use_ibrav = true: ibrav は結晶構造データから取得された Bravais 格子のインデックスがセットされます。また、Structure データは基本格子のとり方など Quantum ESPRESSO の convention に合わせて再構成されます。

- nat (原子数) および ntyp (元素種の数) は結晶構造データから取得される値で上書きされます。
- カットオフ ecutwfc および ecutrho の情報は、パラメータの値が空欄の場合は擬ポテンシャルファイルから取得します。

CELL_PARAMETERS

- structure セクションの use_ibrav パラメータが true の場合は出力されません。false の場合は格子ベクトルが出力されます。単位は angstrom です。
- 格子ベクトルの情報は結晶構造データから取得されます。data フィールドに 3x3 の行列を直接指定した場合はその値が用いられます。

ATOMIC_SPECIES

- 原子種・原子量・擬ポテンシャルファイル名のリストを出力します。
- 原子種の情報は結晶構造データから取得されます。擬ポテンシャルのファイル名は pp_list で指定する CSV 形式のインデックスファイルを参照します。
- data フィールドに必要なデータを指定した場合はその値が用いられます。

ATOMIC_POSITIONS

- 原子種と原子座標 (fractional coordinate) のリストを出力します。
- ignore_species に原子種または原子種のリストを指定した場合、その原子種については if_pos の値が 0 にセットされます。MD や構造最適化の際に使われます。
- data フィールドに必要なデータを指定した場合はその値が用いられます。

K_POINTS

- k 点の情報を出力します。option に出力タイプを指定します。
 - gamma: Γ 点を用います。
 - crystal: メッシュ状の k 点のリストを出力します。メッシュの指定は grid パラメータまたは vol_density や k_resolution から導出される値が用いられます。
 - automatic: k 点のメッシュを指定します。メッシュの指定は grid パラメータまたは vol_density や k_resolution から導出される値が用いられます。シフトの指定は kshifts パラメータを参照します。
- メッシュの指定は以下の順序で決定されます。
 - grid パラメータの指定。grid の値は n_x, n_y, n_z の配列またはスカラー値 n です。後者の場合は $n_x = n_y = n_z = n$ と仮定します。
 - vol_density パラメータから自動導出。

- `k_resolution` パラメータから自動導出。`k_resolution` のデフォルトは 0.15 です。
- `data` フィールドに必要なデータを指定した場合はその値が用いられます。

5.3 VASP 向けパラメータ

`optional` セクションおよび `tasks` セクションの `content` に記載する内容について、VASP 固有の内容を記述します。

5.3.1 optional

擬ポテンシャルのタイプや格納場所を指定します。

`pymatgen` では、擬ポテンシャルファイルを `PMG_VASP_PSP_DIR/functional/POTCAR.element (.gz)` または `PMG_VASP_PSP_DIR/functional/element/POTCAR` から取得します。`PMG_VASP_PSP_DIR` はディレクトリの指定で、設定ファイル `~/.config/.pmgrc.yaml` に記載するか、同名の環境変数に指定します。また、`functional` は擬ポテンシャルのタイプで、`POT_GGA_PAW_PBE` や `POT_LDA_PAW` などが決められています。

`pseudo_functional`

擬ポテンシャルのタイプを指定します。タイプの指定と上記の `functional` の対応は `pymatgen` 内のテーブルに定義され、`PBE → POT_GGA_PAW_PBE`, `LDA → POT_LDA_PAW` などのようになっています。

以下の `pseudo_dir` を指定した場合は `pymatgen` の流儀を無視して擬ポテンシャルの格納ディレクトリを探します。

`psuedo_dir`

擬ポテンシャルの格納ディレクトリを指定します。擬ポテンシャルファイルのファイル名は `pseudo_dir/POTCAR.element (.gz)` または `pseudo_dir/element/POTCAR` です。

5.3.2 tasks

テンプレートファイルは、`template_dir` で指定するディレクトリ内に `INCAR`, `KPOINTS`, `POSCAR`, `POTCAR` ファイルを配置します。ファイルがない項目は無視されます。

5.3.3 content

incar

- INCAR ファイルに記述するパラメータを列挙します。

kpoints

- type

KPOINTS の指定方法を記述します。以下の値に対応しています。タイプによりパラメータが指定可能なものがあります。詳細は `pymatgen.io.vasp` のマニュアルを参照してください。

- `automatic`
parameter: `grid`
- `gamma_automatic`
parameter: `grid`, `shift`
- `monkhorst_automatic`
parameter: `grid`, `shift`
- `automatic_density`
parameter: `kppa`, `force_gamma`
- `automatic_gamma_density`
parameter: `grid_density`
- `automatic_density_by_vol`
parameter: `grid_density`, `force_gamma`
- `automatic_density_by_lengths`
parameter: `length_density`, `force_gamma`
- `automatic_linemode`
parameter: `division`, `path_type` (`HighSymmKpath` の `path_type` に対応)

5.4 OpenMX 向けパラメータ

optional セクションおよび tasks セクションの content に記載する内容について、OpenMX 固有の内容を記述します。

5.4.1 optional

data_path

擬原子軌道および擬ポテンシャルのファイルを格納するディレクトリを指定します。入力ファイルの DATA.PATH パラメータに対応します。

5.4.2 content

precision

擬原子軌道を OpenMX マニュアル 10.6 章の Table 1, 2 にしたがって選択します。quick, standard, precise のいずれかの値を取ります。デフォルト値は quick です。

5.5 AkaiKKR 向けパラメータ

optional セクションおよび tasks セクションの content に記載する内容について、AkaiKKR 固有の内容を記述します。

5.5.1 optional

workdir

一時ファイルの出力先を指定します。指定しない場合は /tmp または TMPDIR 環境変数の値を利用します。

5.5.2 content

content には AkaiKKR の入力パラメータの内容を記述します。指定のない項目は空欄が出力され、AkaiKKR 内部のデフォルト値が使われます。以下のパラメータは結晶構造データから決まる値で置き換えられます。

- brvtyp: ただし、brvtyp に aux (を含む) 値が指定されている場合は上書きされません。
- 格子パラメータ: a, c/a, b/a, alpha, beta, gamma, r1, r2, r3

- タイプ情報: ntyp, type, ncmp, rmt, field, mxl, anclr, conc
- 元素種の情報: natm, atmicx, atmtyp

なお、rmt と field の値は、入力パラメータファイル内で ntyp 個の要素からなるリストとして指定されている場合のみ、入力パラメータファイルの値が使われます。

第 6 章

拡張ガイド

6.1 Quantum ESPRESSO の mode を追加する

Quantum ESPRESSO の計算モードへの対応を追加するには、`src/cif2x/qe/calc_mode.py` の `create_modeproc()` 関数に mode と変換クラスの対応付けを記述します。

```
def create_modeproc(mode, qe):
    if mode in ["scf", "nscf"]:
        modeproc = QEmode_pw(qe)
    else:
        modeproc = QEmode_generic(qe)
    return modeproc
```

mode ごとの変換機能は `QEmode_base` の派生クラスとしてまとめられています。このクラスは `update_namelist()` で `namelist` ブロックの更新と、`update_cards()` で `cards` ブロックのデータ生成を行います。現在は `pw.x` の `scf` および `nscf` に対応する `QEmode_pw` クラスと、変換せずそのまま出力する `QEmode_generic` クラスが用意されています。

```
class QEmode_base:
    def __init__(self, qe):
    def update_namelist(self, content):
    def update_cards(self, content):
```

`namelist` については、空欄の値を結晶構造データ等から生成して代入するほか、格子パラメータなど `Structure` から決まる値や、他のパラメータとの整合性を取る必要のある値を強制的にセットする場合があります。処理内容はモードごとに個別に対応します。

`cards` ブロックについては、`card` の種類ごとに関数を用意し、`card` 名と関数の対応付けを `card_table` 変数に列挙します。基底クラスの `update_cards()` では、`card` 名から対応する関数を取得して実行し、`card` の情報を更新します。もちろん、全く独自に `update_cards()` 関数を作成することもできます。


```
self.card_table = {  
    'CELL_PARAMETERS': generate_cell_parameters,  
    'ATOMIC_SPECIES': generate_atomic_species,  
    'ATOMIC_POSITIONS': generate_atomic_positions,  
    'K_POINTS': generate_k_points,  
}
```

card ごとの関数は `scr/cif2x/qe/cards.py` にまとめられており、関数名は `generate_{card 名}` としています。この関数は card ブロックのパラメータを引数に取り、card 名、option、data フィールドからなる辞書データを返します。

第 7 章

CIF データ取得ツール (getcif)

7.1 概要

getcif は物質材料データベースから結晶構造データ等を取得するツールです。現在は Materials Project からのデータ取得に対応しています。物質の組成や対称性、バンドギャップなどの物性値をもとにデータベースを検索し、データを取得することができます。

7.2 チュートリアル

結晶構造などのデータを物質材料データベースから取得するツール getcif を使うには、検索条件と取得するデータを記述した入力パラメータファイルを作成し、プログラム getcif を実行します。現在は Materials Project が公開しているデータベースに対応しています。以下では docs/tutorial/getcif ディレクトリにある ABO₃ 系の物質を検索・取得するサンプルを例にチュートリアルを実施します。

7.2.1 API キーを取得する

Materials Project のデータベースをプログラムから検索するには、あらかじめ Materials Project にユーザ登録し、API キーを取得する必要があります。Materials Project の公式サイト <https://next-gen.materialsproject.org> にアクセスし、Login します。API キーはユーザ登録時に自動的生成され、ユーザのダッシュボードから確認できます。取得した API キーは安全に保管し、他人に知られないようにしましょう。

API キーを getcif にセットするには、以下のいずれかを実行します。

- (a) pymatgen の設定ファイルに登録する

```
$ pmg config --add PMG_MAPI_KEY <API_KEY>
```

を実行するか、設定ファイル ~/.config/.pmgrc に

```
PMG_MAPI_KEY: <API_KEY>
```

を書き込みます。

(b) 環境変数にセットする

```
$ MP_API_KEY="<API_KEY>"
$ export MP_API_KEY
```

を実行します。

(c) ファイルに格納する

API キーをファイルに書き込み、getcif を実行するディレクトリに配置します。ファイル名のデフォルトは materials_project.key です。異なるファイル名を使う場合は、入力パラメータファイル (input.yaml) の api_key_file にファイル名を指定します。ファイル名は .key の拡張子が必要です。

```
database:
  api_key_file: materials_project.key
```

註: バージョン管理ツールを使っている場合は、.key の拡張子を持つファイルを管理から除外するとよいでしょう。(Git の場合は .gitignore ファイルに *.key を追加します。)

7.2.2 入力パラメータファイルを作成する

入力パラメータファイルにはデータベース検索および出力についての設定を記述します。

以下に入力パラメータファイルのサンプルを記載します。このファイルは YAML 形式のテキストファイルで、データベースへの接続に必要な情報や、検索条件、取得するデータの種類などの内容を記述します。仕様の詳細については [ファイルフォーマット](#) の章を参照してください。

YAML フォーマットでは、keyword: value の辞書形式でパラメータを記述します。value には数値や文字列などのスカラー値や、複数の値を [] または箇条書きの形式で列挙するリスト型、または辞書型を入れ子にすることができます。検索条件と出力項目については、特別な記法として、リスト型を括弧を使わず空白区切りで列挙する形式でも書くことができます。

```
database:
  target: materials project

option:
  output_dir: result
  # dry_run: false
```

(次のページに続く)

```
properties:
  band_gap: < 1.0
  is_stable: true
  is_metal: false
  formula: "***O3"
  spacegroup_symbol: Pm-3m

fields: |
  structure
  band_gap
  symmetry
```

入力パラメータファイルは database, option, properties, fields のブロックからなります。database にはデータベース接続に関する設定を記述します。例では target に Materials Project を指定していますが、現時点ではこの項目は無視されます。その他、api_key に API キーを指定できます。API キーは pymatgen の設定ファイルや環境変数にセットすることもできます。例では後者の方式を仮定しています。

option には実行時のオプションを記述します。output_dir は取得したデータの格納先ディレクトリを指定します。省略時にはカレントディレクトリに書き出されます。dry_run を true にセットすると、データベースへの接続はせず、検索条件を出力して終了します。dry_run はコマンドラインオプションでも指定できます。

properties は検索条件の指定を行います。検索条件を「項目: 値」の書式で列挙し、これらの条件は AND で扱われます。例では、バンドギャップが 1.0 以下、安定な絶縁体で、組成式は ABO₃ (A, B は任意の元素種)、空間群は Pm-3m という条件を指定しています。band_gap には値の範囲を上限・下限の組で指定するほか、< 1.0 のような記法も使用できます。検索条件にどのような項目が指定できるかは Appendix を参照してください。

fields には出力項目を列挙します。YAML のリスト形式のほか、空白区切りで項目を列挙する書き方もできます。また、例に示したとおり、YAML の記法を使って複数行で書くこともできます。structure は結晶構造データで、取得したデータは CIF 形式で出力されます。band_gap はバンドギャップの数値、symmetry は対称性の情報です。この他に、material_id で Materials Project 内の物質データのインデックスと、formula_pretty で組成式が暗黙的に取得されます。出力項目の一覧は Appendix を参照してください。また、getcif コマンドのヘルプメッセージにも一覧が出力されます。

7.2.3 データを取得する

入力パラメータファイル (input.yaml) を引数として getcif を実行します。

```
$ getcif input.yaml
```

getcif を実行すると Materials Project のデータベースに接続し、検索条件に合致するデータを取得します。標準出力には、以下のように、物質の material ID と組成式、データ項目のサマリーが出力されます。

```
material_id formula band_gap symmetry formula_pretty
mp-861502 AcFeO3 0.9887999999999995 crystal_system=<CrystalSystem.cubic: 'Cubic'>
↳symbol='Pm-3m' number=221 point_group='m-3m' symprec=0.1 version='2.0.2' AcFeO3
mp-977455 PaAgO3 0.915 crystal_system=<CrystalSystem.cubic: 'Cubic'> symbol='Pm-3m'
↳number=221 point_group='m-3m' symprec=0.1 version='2.0.2' PaAgO3
mp-11775 RbUO3 0.454200000000000016 crystal_system=<CrystalSystem.cubic: 'Cubic'>
↳symbol='Pm-3m' number=221 point_group='m-3m' symprec=0.1 version='2.0.2' RbUO3
mp-3163 BaSnO3 0.37239999999999984 crystal_system=<CrystalSystem.cubic: 'Cubic'>
↳symbol='Pm-3m' number=221 point_group='m-3m' symprec=0.1 version='2.0.2' BaSnO3
mp-4126 KUO3 0.445400000000000024 crystal_system=<CrystalSystem.cubic: 'Cubic'> symbol=
↳'Pm-3m' number=221 point_group='m-3m' symprec=0.1 version='2.0.2' KUO3
mp-865322 UTlO3 0.273600000000000007 crystal_system=<CrystalSystem.cubic: 'Cubic'>
↳symbol='Pm-3m' number=221 point_group='m-3m' symprec=0.1 version='2.0.2' UTlO3
mp-753781 EuHfO3 0.4795999999999996 crystal_system=<CrystalSystem.cubic: 'Cubic'>
↳symbol='Pm-3m' number=221 point_group='m-3m' symprec=0.1 version='2.0.2' EuHfO3
```

取得したデータは、output_dir で指定した result ディレクトリ内に物質ごとに格納されます。この例では、result 以下に mp-3163 から mp-977455 までの 7 つのディレクトリが作成され、各ディレクトリには次のファイルが書き込まれます。

- band_gap

バンドギャップの値

- formula

組成式 (formula_pretty に対応します)

- structure.cif

CIF 形式の結晶構造データ

- symmetry

対称性に関するデータ

getcif の実行に `--dry-run` オプションを付けると、以下のように検索条件を出力して終了します。データベースに実際に接続する前に検索項目を確認できます。

```
$ getcif --dry-run input.yaml
{'band_gap': (None, 1.0), 'is_stable': True, 'is_metal': False, 'formula': '**O3',
 'spacegroup_symbol': 'Pm-3m', 'fields': ['structure', 'band_gap', 'symmetry', 'material_
↳ id', 'formula_pretty']}
```

7.3 コマンドリファレンス

7.3.1 getcif

結晶構造データをデータベースから取得する

書式:

```
getcif [-v][-q] [--dry-run] input_yaml
getcif -h
getcif --version
```

説明:

`input_yaml` に指定した入力パラメータファイルを読み込み、データベースを検索して結晶構造等のデータを取得します。以下のオプションを受け付けます。

- `-v`

実行時に表示されるメッセージを冗長にします。複数回指定すると冗長度が上がります。

- `-q`

実行時に表示されるメッセージの冗長度を下げます。 `-v` の効果を打ち消します。複数回の指定が可能です。

- `--dry-run`

検索パラメータを出力し、データベースへの接続をせずに終了します。検索条件を確認することができます。入力パラメータファイルの `dry_run` の指定より優先します。

- `input_yaml`

入力パラメータファイルを指定します。ファイルフォーマットは YAML 形式です。

- `-h, --help`

ヘルプを表示します。指定可能な検索条件 (properties) と取得データ (fields) の一覧も表示されます。

- `--version`

バージョン情報を表示します。

7.4 ファイルフォーマット

7.4.1 入力パラメータファイル

入力パラメータファイルでは、`getcif` で Materials Project の物質材料データベースから結晶構造等のデータを取得するための設定情報を YAML 形式で記述します。本ファイルは以下の部分から構成されます。

1. `database` セクション: 接続するデータベースについての情報を記述します。
2. `option` セクション: 出力先のディレクトリや実行条件などを記述します。
3. `properties` セクション: 検索条件を記述します。
4. `fields` セクション: 取得データの種別を記述します。

database

`target`

接続先のデータベースを指定します。現在はこの項目は無視されます。

`api_key_file` (デフォルト値: `materials_project.key`)

データベースに接続する際の API キーを格納したファイルのファイル名を指定します。ファイル名の拡張子は `.key` とします。ファイルが存在しない、または有効な API キーが見つからない場合は、環境変数 `MP_API_KEY` または `pymatgen` の設定ファイル `~/.config/.pmgrc` の `PMG_MAPI_KEY` から API キーを取得します。

API キーファイルはテキスト形式です。 `#` から始まる行はコメントとして扱われます。前後の空白は無視されます。複数行からなる場合は最初の有効な行からキーを取得します。

option

`output_dir` (デフォルト値: `""`)

取得データを格納するディレクトリを指定します。データは `output_dir` 以下に、material ID をディレクトリ名としたディレクトリに出力されます。指定がない場合はカレントディレクトリです。

`dry_run` (デフォルト値: `False`)

データベースへの接続は行わず、検索条件を出力して終了します。検索内容の確認を行うことができます。

symprec (デフォルト値: 0.1)

結晶構造データを CIF ファイルに出力する際の対称性を判定する許容精度を指定します。デフォルトは 0.1 です。symprec に 0.0 を指定した場合は symprec を指定しないものとして扱い、対称性を考慮しない CIF ファイルが生成されます。

symprec は、結晶構造における対称性を判定する際の許容精度 (tolerance) を指定するパラメータです。対称性の計算においては、原子位置の微細なずれや数値計算の精度の影響を考慮する必要があります。symprec はこのずれの許容範囲を制御し、対称操作が適用されるかどうかを決定する際の閾値として機能します。

symprec を小さく設定する (例: 0.01) と、対称性の判定がより厳密になり、結晶構造のわずかなずれでも対称操作が適用されない可能性が高まります。その結果、より低い対称性の空間群が得られることがあります。逆に、symprec を大きく設定する (例: 1.0) と、対称性の判定が緩やかになり、わずかなずれが無視され、より高い対称性が認められることがあります。

なお、fields セクションで symmetry を指定すると、Materials Project でデフォルトの symprec=0.1 で判定された対称性の情報を取得し、テキストファイル (symmetry) に出力します。

properties

検索条件を記述します。元素組成や結晶の対称性、物性値の範囲などの項目を、「項目名: 値」の形式で指定します。これらの条件は AND で扱われます。指定できる項目は Materials Project の API に定義されていますが、指定方法は mp-api ライブラリの materials.summary.search のパラメータに準拠します。項目のリストは Appendix を参照してください。また、getcif --help で一覧を見ることができます。

値の指定方法は次のとおりです。YAML 形式に準拠しますが、一部に簡便な記法を用意しています。

- 数値、文字列

そのまま記述します。

- 真偽値

true または false を記述します

- 数値や文字列のリスト

YAML 形式の簡条書きおよび [...] にカンマ区切りで記述するほか、空白区切りで列挙する記法も可能です。例:

```
element: Sr Ti
```

- 数値の範囲

上限・下限のリストとして [min, max] のように記述するほか、空白区切りで min max のように記述することもできます。また、以下の記法も可能です。

`<= max`

max 以下

`< max`

max より小さい (実数の場合は `<=` と同等。整数の場合は `<= max-1` として扱われる)

`>= min`

min 以上

`> min`

min より大きい (実数の場合は `>=` と同等。整数の場合は `>= min+1` として扱われる)

`min ~ max`

min 以上 max 以下

注記:

- 記号と数値の間は空白を置きます。
- YAML 記法では `>` は特殊文字として扱われるため、`>= min`, `> min` はそれぞれ `">= min"`, `"> min"` のように `" "` で囲む必要があります。
- リストで記述する場合、`<= max`, `>= min` はそれぞれ `[None, max]`, `[min, None]` のように表記します。

• ワイルドカード

`formula` には元素種にワイルドカード `*` を指定できます。その場合は値を `" "` で囲みます。例:

```
formula: "**03"
```

ABO_3 系の物質を指定します。

fields

取得するデータの種別を記述します。項目のリストを YAML 形式で列挙するほか、空白区切りの文字列として記述することもできます。文字列は YAML 記法 `|` を用いて複数行で書くこともできます。指定できる項目は Materials Project の API の `fields` パラメータに準拠します。項目のリストは Appendix を参照してください。また、`getcif --help` で一覧を見ることができます。

`material_id` と `formula_pretty` は暗黙的に取得します。

取得したデータは、`option` セクションの `output_dir` で指定したディレクトリ内に、物質ごとに `material_id` をディレクトリ名とするディレクトリを作成し、その中に格納されます。項目ごとに、項目名をファイル名としたファイルに保存されます。但し、結晶構造データ (structure) は `structure.cif` というファイル名で CIF 形式で書き出されます。

7.5 パラメータリスト

7.5.1 検索条件 (properties)

properties に指定できる項目と、その項目がどのような値を取るかを以下にまとめます。

Materials Project API のクライアントアプリケーションの一つとして mp-api パッケージが Materials Project から公開されており、getcif はこのライブラリを利用してデータベースへの接続を行います。以下は MPRester クラスの materials.summary.search メソッドのパラメータに対応します。(以下の表は materials.summary.search のソースコードのコメントから転記しました。)

値の型の表記は次のとおりです。

- str: 文字列型
- List[str]: 文字列型のリスト
- str | List[str]: 単一の文字列、または、文字列型のリスト
- int: 整数型
- bool: 真偽値 (true または false)
- Tuple[float, float]: 実数値 2 つからなる組 (リスト)
- Tuple[int, int]: 整数値 2 つからなる組 (リスト)
- CrystalSystem: 結晶のタイプを表す文字列。Triclinic, Monoclinic, Orthorhombic, Tetragonal, Trigonal, Hexagonal, Cubic のいずれか。
- List[HasProps]: 特性値のタイプを表す文字列のリスト。特性値は emmet.core.summary に定義されている。以下のいずれかの値を取る。

absorption, bandstructure, charge_density, chemenv, dielectric, dos, elasticity, electronic_structure, eos, grain_boundaries, insertion_electrodes, magnetism, materials, oxi_states, phonon, piezoelectric, provenance, substrates, surface_properties, thermo, xas
- Ordering: 磁気秩序を表す文字列。FM, AFM, FiM, NM のいずれか。

値のリストは、YAML 形式の箇条書きおよび [...] にカンマ区切りで記述するほか、空白区切りで列挙する記法も可能です。

Tuple で表される型は値の範囲 (min, max) の指定に使われます。値のリストとして記述するほか、空白区切りで min max のように記述することもできます。また、以下の表記も可能です。

<= max : max 以下

< max : max より小さい

`>= min` : min 以上

`> min` : min より大きい

`min ~ max` : min 以上 max 以下

表 7.1: 検索条件のキーワード

Keyword	Type	Description
band_gap	Tuple[float,float]	Minimum and maximum band gap in eV to consider.
chemsys	str List[str]	A chemical system, list of chemical systems (e.g., Li-Fe-O, Si-*, [Si-O, Li-Fe-P]), or single formula (e.g., Fe2O3, Si*).
crystal_system	CrystalSystem	Crystal system of material.
density	Tuple[float,float]	Minimum and maximum density to consider.
deprecated	bool	Whether the material is tagged as deprecated.
e_electronic	Tuple[float,float]	Minimum and maximum electronic dielectric constant to consider.
e_ionic	Tuple[float,float]	Minimum and maximum ionic dielectric constant to consider.
e_total	Tuple[float,float]	Minimum and maximum total dielectric constant to consider.
efermi	Tuple[float,float]	Minimum and maximum fermi energy in eV to consider.
elastic_anisotropy	Tuple[float,float]	Minimum and maximum value to consider for the elastic anisotropy.
elements	List[str]	A list of elements.
energy_above_hull	Tuple[int,int]	Minimum and maximum energy above the hull in eV/atom to consider.
equilibrium_reaction_energy	Tuple[float,float]	Minimum and maximum equilibrium reaction energy in eV/atom to consider.
exclude_elements	List[str]	List of elements to exclude.
formation_energy	Tuple[int,int]	Minimum and maximum formation energy in eV/atom to consider.
formula	str List[str]	A formula including anonymized formula or wild cards (e.g., Fe2O3, ABO3, Si*). A list of chemical formulas can also be passed (e.g., [Fe2O3, ABO3]).
g_reuss	Tuple[float,float]	Minimum and maximum value in GPa to consider for the Reuss average of the shear modulus.
g_voigt	Tuple[float,float]	Minimum and maximum value in GPa to consider for the Voigt average of the shear modulus.
g_vrh	Tuple[float,float]	Minimum and maximum value in GPa to consider for the Voigt-Reuss-Hill average of the shear modulus.
has_props	List[HasProps]	The calculated properties available for the material.
has_reconstructed	bool	Whether the entry has any reconstructed surfaces.

次のページに続く

表 7.1 – 前のページからの続き

Keyword	Type	Description
is_gap_direct	bool	Whether the material has a direct band gap.
is_metal	bool	Whether the material is considered a metal.
is_stable	bool	Whether the material lies on the convex energy hull.
k_reuss	Tuple[float,float]	Minimum and maximum value in GPa to consider for the Reuss average of the bulk modulus.
k_voigt	Tuple[float,float]	Minimum and maximum value in GPa to consider for the Voigt average of the bulk modulus.
k_vrh	Tuple[float,float]	Minimum and maximum value in GPa to consider for the Voigt-Reuss-Hill average of the bulk modulus.
magnetic_ordering	Ordering	Magnetic ordering of the material.
material_ids	List[str]	List of Materials Project IDs to return data for.
n	Tuple[float,float]	Minimum and maximum refractive index to consider.
num_elements	Tuple[int,int]	Minimum and maximum number of elements to consider.
num_sites	Tuple[int,int]	Minimum and maximum number of sites to consider.
num_magnetic_sites	Tuple[int,int]	Minimum and maximum number of magnetic sites to consider.
num_unique_magnetic_sites	Tuple[int,int]	Minimum and maximum number of unique magnetic sites to consider.
piezoelectric_modulus	Tuple[float,float]	Minimum and maximum piezoelectric modulus to consider.
poisson_ratio	Tuple[float,float]	Minimum and maximum value to consider for Poisson's ratio.
possible_species	List[str]	List of element symbols appended with oxidation states. (e.g. Cr2+, O2-)
shape_factor	Tuple[float,float]	Minimum and maximum shape factor values to consider.
spacegroup_number	int	Space group number of material.
spacegroup_symbol	str	Space group symbol of the material in international short symbol notation.
surface_energy_anisotropy	Tuple[float,float]	Minimum and maximum surface energy anisotropy values to consider.
theoretical	bool	Whether the material is theoretical.
total_energy	Tuple[int,int]	Minimum and maximum corrected total energy in eV/atom to consider.
total_magnetization	Tuple[float,float]	Minimum and maximum total magnetization values to consider.
total_magnetization_normalized	Tuple[float,float]	Minimum and maximum total magnetization values normalized by formula units to consider.
total_magnetization_normalized	Tuple[float,float]	Minimum and maximum total magnetization values normalized by volume to consider.

次のページに続く

表 7.1 – 前のページからの続き

Keyword	Type	Description
uncorrected_energy	Tuple[int,int]	Minimum and maximum uncorrected total energy in eV/atom to consider.
volume	Tuple[float,float]	Minimum and maximum volume to consider.
weighted_surface_energy	Tuple[float,float]	Minimum and maximum weighted surface energy in J/m^2 to consider.
weighted_work_function	Tuple[float,float]	Minimum and maximum weighted work function in eV to consider.

7.5.2 出力項目 (fields)

fields に指定できる項目を以下に列挙します。

```
band_gap
bandstructure
builder_meta
bulk_modulus
cbm
chemsys
composition
composition_reduced
database_IDs
decomposes_to
density
density_atomic
deprecated
deprecation_reasons
dos
dos_energy_down
dos_energy_up
e_electronic
e_ij_max
e_ionic
e_total
efermi
elements
energy_above_hull
energy_per_atom
```

(次のページに続く)

(前のページからの続き)

```
equilibrium_reaction_energy_per_atom
es_source_calc_id
formation_energy_per_atom
formula_anonymous
formula_pretty
grain_boundaries
has_props
has_reconstructed
homogeneous_poisson
is_gap_direct
is_magnetic
is_metal
is_stable
last_updated
material_id
n
nelements
nsites
num_magnetic_sites
num_unique_magnetic_sites
ordering
origins
possible_species
property_name
shape_factor
shear_modulus
structure
surface_anisotropy
symmetry
task_ids
theoretical
total_magnetization
total_magnetization_normalized_formula_units
total_magnetization_normalized_vol
types_of_magnetic_species
uncorrected_energy_per_atom
universal_anisotropy
vbm
volume
```

(次のページに続く)

(前のページからの続き)

```
warnings
weighted_surface_energy
weighted_surface_energy_EV_PER_ANG2
weighted_work_function
xas
```