Crystalline Sites Finder (`CrySF`)

User Manual

Henry A. Cortés[1], Mauricio R. Bonilla[1], and Elena Akhmatskaya[1,2]

[1]BCAM - Basque Center for Applied Mathematics, Alameda de Mazarredo 14, E-48009 Bilbao, Spain
[2]IKERBASQUE, Basque Foundation for Science, Plaza Euskadi 5, 48009 Bilbao, Spain

Introduction

CrySF is a specialized Python software designed to analyze atomic trajectories in simulations of crystallography solid-state ionic conductors. This tool not only identifies and characterizes the crystallographic sites occupied by diffusive atoms but also examines their transitions between sites. Such analyses provide crucial insights for researchers studying advanced materials.

The software operates through two distinct commands. The first command, `DensityMap`, processes the atomic trajectories into data that the second command, `CrySF`, can further analyze. For a comprehensive understanding of the theoretical foundations and functionalities of **CrySF**, readers are encouraged to consult the detailed manuscript "Name of the Manuscript" authored by Henry A. Cortés, Mauricio R. Bonilla, and Elena Akhmatskaya. This document includes numerous examples that demonstrate the practical applications of the software.

Installation

CrySF is a tool that can be used with Python versions 3.8 and above. It is essential to install the Python Packages, `scikit-learn` and `scipy` since **CrySF** uses various machine learning and data analysis functionalities from these libraries. Additionally, `Mdanalysis` must be installed since it is used to read the atomic trajectories.

Before proceeding with the installation, ensure you have installed the supported version of Python and fulfill all other requirements.

To install and use the **CrySF** software in a Linux-like system. The user can obtain the code by cloning the following repository:

```
git clone https://gitlab.bcamath.org/hacortes/crysf
```

The command will create the `crysf/` folder with the source files. For the **CrySF** package installation, execute the following command (in a UNIX-like terminal):

```
cd crysf/ python setup.py install −−record files.txt
```

Alternatively, you can download **CrySF** from its repository at the following URL:

```
https://gitlab.bcamath.org/hacortes/crysf
```

After downloading, unzip the file to find the `crysf/` folder as described earlier. Continue with the installation using the same procedure outlined above.

How to use the code

After performing an atomistic simulation, **Crysf** can be used for analysis via the following steps:

1. Prepare the Simulation Data:
   (a) Store the atomistic trajectory in a dedicated folder. Include the following files based on the simulation software used:
      - XDATCAR from VASP.
      - from LAMMPS or GROMACS topology and trajectory file must be included.
2. Set Up the Analysis Scripts:

- Copy the files `densitymap.py` and `crysf.py` into the same folder as your simulation data.

3. Execute `DensityMap` which creates a density map:

   - Run the following command to execute `densitymap.py`. `python`

     ```
     densitymap.py -to 'str' -tr 'str' -f 'str' -ts float -v
     float -a int or 'str' -tts int -verb int -clus int
     ```

4. Execute `CrySF` to analyze the density map generated in the previous step:

   - After generating the density map, run `crysf.py` with the command:

     ```
     python crysf.py -nts float -minv float -maxv float -clus
     int -dop int -verb int -deltf int -scaler 'str'
     ```

The settings to use `DensityMap` and `CrySF` commands are detailed in the **Settings** section.

The time it takes to analyze a trajectory depends on the simulation's number of atoms and the frequency with which the trajectory is printed. Usually, the analysis could take a few minutes.

Settings

Table 1: Command-line Settings for `DensityMap`.

Parameter	**Description**	**Default**
`-to`	Topology file name (.gro or .data formats). XDATCAR does not require to include `-to`	topo.gro
`-tr`	Trajectory file name.	trajectory.xtc
`-f`	Format options: XTC, TRR, LAMMPSDUMP and XDATCAR.	XDATCAR
`-ts`	Time interval between frames in the trajectory.	0.1
`-v`	Voxel side size.	0.2
`-a`	Atom selection: integer (index) or string (name) based on trajectory format.	Li
`-tts`	Frames interval to print the `VoxelIndices.dat`.[1]	1
`-clus`	Set whether clustering $<d_i>$ of the trajectory is applied (0 = NO or 1 = YES).	0
`-verb`	Set verbosity level (0 = L and HM or 1 = L, M, H and HM).[2]	0

[1] The frequency at which the position of each diffusive atom is recorded in terms of voxel indices can be adjusted. A value equivalent to 1 ps is suggested (i. e. `-ts` * `-tts` $\sim$ 1 ps) since it is in the range of a typical phonon frequency[1, 2].

[2] L, M and H represent the $<d_i>$ clusters low, medium, and high, respectively[3].

Parameter	Description	Default
`-nts`	New time interval between frames in `VoxelIndices.dat` (i. e., `-ts` * `-tts`).[1]	0.1
`-minv`	Minimum volume condition for the site.	0.28
`-maxv`	Minimum volume condition for the site.	3.05
`-clus`	Set whether clustering $<d_i>$ of the trajectory is used (0 = NO or 1 = YES).[2]	0
`-dop`	Sets whether to use a doped density map, 0 = NO or 1 = YES.[3]	0
`-verb`	Set verbosity level, 0 = L and HM(together) or 1 = L, M, H and HM(together).	0
`-deltf`	Frames intervals to calculate the simultaneous jumps.[4]	1
`-scaler`	Scaler method used for site type identification. MinMaxScaler or StandardScaler.	MinMaxScaler

[1] This value is an output of the `DensityMap` routine in the Voxel indices information (see `Time interval between frames`).

[2] Further details about this topic are discussed in the $<\mathbf{d}_i>$ **Clustering** section.

[3] Additional information on this topic can be found in the section **Doped systems**.

[4] Given that these jumps are rare events, it is often advantageous to segment the trajectory into chunks for counting simultaneous jumps. This setting is an alternative to `-tts` > 1 in `DensisyMap`. Here, it is possible to use `-tts` $= 1$ (all frames) and then split the trajectory into chunks where, in each of them, it is established whether or not there was a jump.

The units of measurement used within the software are angstroms (`Å`) for distance and picoseconds (`ps`) for time.

Output-files

1. The `DensityMap` command outputs trajectory information directly to the screen. Below is an example of the output format:

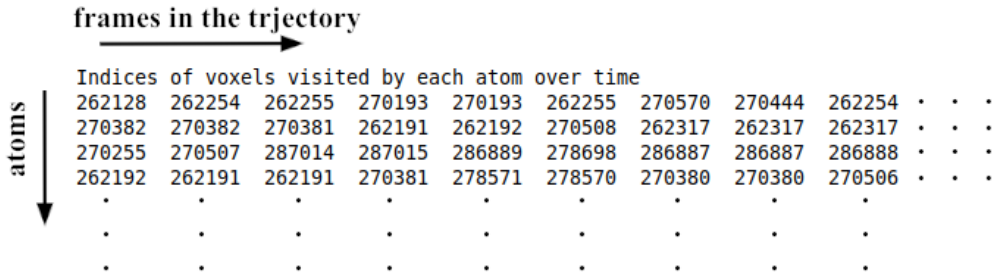
```
>> Reading the trajectory <<
>> Calculating density map <<
Format: XTC
Number frames DensityMap: 80001
Diffusive atoms: 448
Voxel size: 0.2
Number voxels: 2197000

Writing file: DensityMap.dat
Time interval between frames DensityMap: 0.0500

Writing file: VoxelIndices.dat
Time interval between frames VoxelIndices: 0.0500
Number of frames VoxelIndices: 80001

Writing file: CHGCAR_VESTA
>> Done <<
```

   Furthermore, this command generates the files:

   (a) `DensityMap.dat`: Details the density and coordinates (x, y, z) of the voxels visited by the mobile atoms. Each voxel is assigned an index, and the size of the voxels is specified in the header. Example shown below:

```
Density          x          y          z        index       voxel_ds: 0.2029 0.2029 0.2009
1.685288e+01     1          1          1          1
5.563482e+00     1          1          2          2
8.121871e-02     1          1          3          3
1.624374e+00     1          1         31         31
1.344170e+01     1          1         32         32
     .           .          .          .          .
     .           .          .          .          .
     .           .          .          .          .
```

   (b) `VoxelIndices.dat`: Records the indices of voxels visited by each atom over time. Each line corresponds to the trajectory of an atom in terms of voxel indices. Example shown below:



(c) `CHGCAR_VESTA`: this text file can be opened with the free visualization software `VESTA` (`https://jp-minerals.org/vesta/en/download.html`). It allows for the straightforward visualization of density maps.

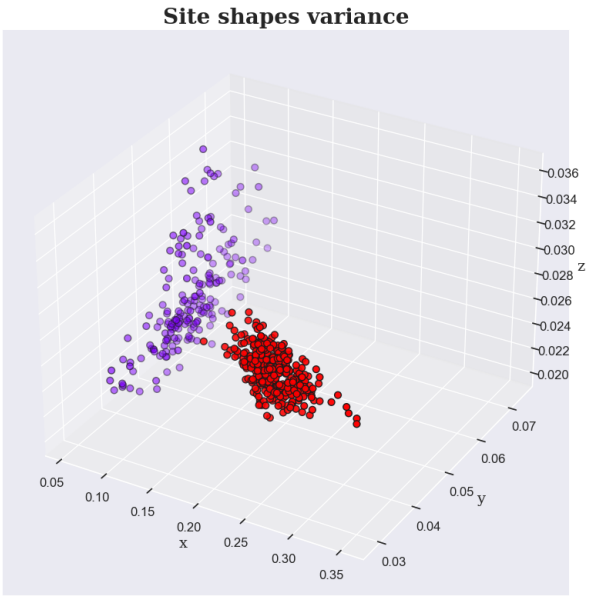
2. The `CrySF` command processes the `DensityMap.dat` and `VoxelIndices.dat` files, displaying a variety of analysis results and the settings used. Below is an example of the output shown on the screen:

```
>> Finding the density cutoff for clustering <<
Final Density cutoff: 0.0873

Amplitude types: 2
Site amplitudes type 0: 0.6344
Site amplitudes type 1: 1.0356
Distance used as cutoff in NN type 0: 2.0246
Distance used as cutoff in NN type 1: 2.0452

>> Sites map of the structure <<
Writing file: SitesMap.dat

Information for site type
Label site: 0, Number of Sites: 192
Label site: 1, Number of Sites: 384

Reading file: VoxelIndices.dat
>> Trajectory data <<
Total time: 4000.0000
Time step VoxelIndices: 0.05

Writing file: jumps_info.dat
Writing file: simultaneous_jumps.dat
Writing file: string_frequency.dat
```

```
Frames interval: 20
Time interval between frame intervals: 1.0000
Updated total time: 4000.0000
```

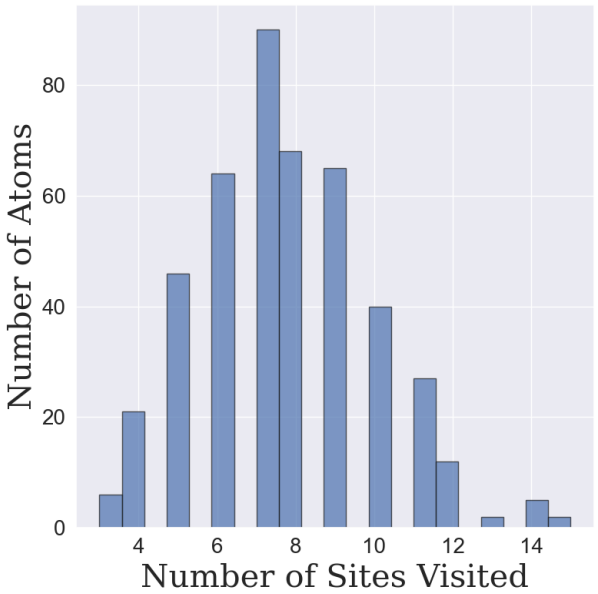
Additionally, this routine generates the following files:

(a) `site_shapes_variance.png`: displays the final clusters in the λ-space (spatial variances) of crystal sites. A detailed description of this file can be found in the manuscript titled "Name of the Manuscript" by Henry A. Cortés, Mauricio R. Bonilla, and Elena Akhmatskaya.



(b) `site_amplitudes.png`: illustrates the amplitudes as a function of the site index, with varying colors indicating different site types.

(c) `site_visited.png`: shows a histogram of the distribution of the number of sites visited by diffusive atoms.

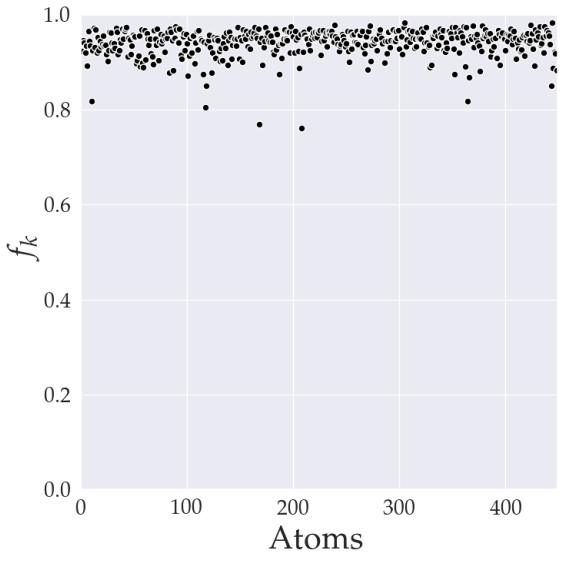


(d) `SitesMap.dat`: provides the coordinates (x, y, z) of the centers of crystal sites, along with additional site-specific details such as the site type, number of nearest neighbors (NN), occupancy, and amplitude.

Site-label	Center(X)	Center(Y)	Center(Z)	Site-Type	N-Neighbos	Occupancy	Site-Amplitud
1	0.191628	0.199143	0.212061	0	4	0.932709	0.662871
2	0.192583	0.199461	6.524142	0	4	0.972005	0.572340
3	0.195903	0.206398	12.857600	0	4	0.969318	0.576409
4	0.192755	0.185992	19.189298	0	4	0.930022	0.577810
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

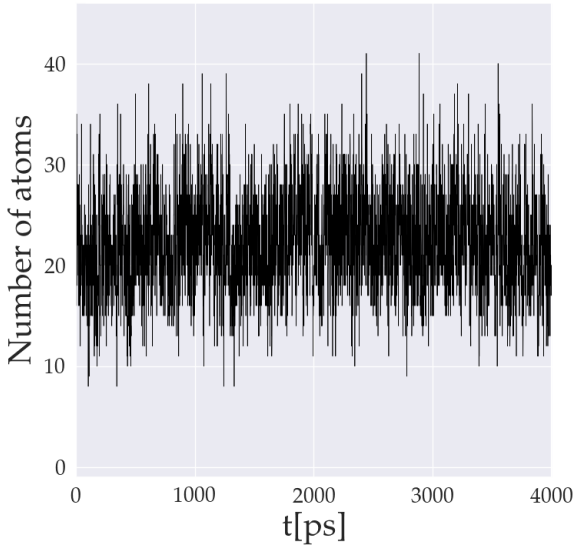
(e) `jumps_info.dat`: records the total number of jumps (NJumps), reverse jumps (RJumps), and the jump coefficient per atom (Jump-Coefficient). The jump coefficient quantifies the directional persistence of successive jumps, ranging from 0 to 1. A coefficient of 0 indicates no successive reverse jumps, while 1 implies an equal number of forward and successive reverse jumps. Atoms that remain stationary throughout the simulation are assigned a value of -1.

Atom	NJumps	RNJumps	JumpCoefficient
1	118	108	0.9153
2	491	481	0.9796
3	243	232	0.9547
4	222	201	0.9054
.	.	.	.
.	.	.	.
.	.	.	.



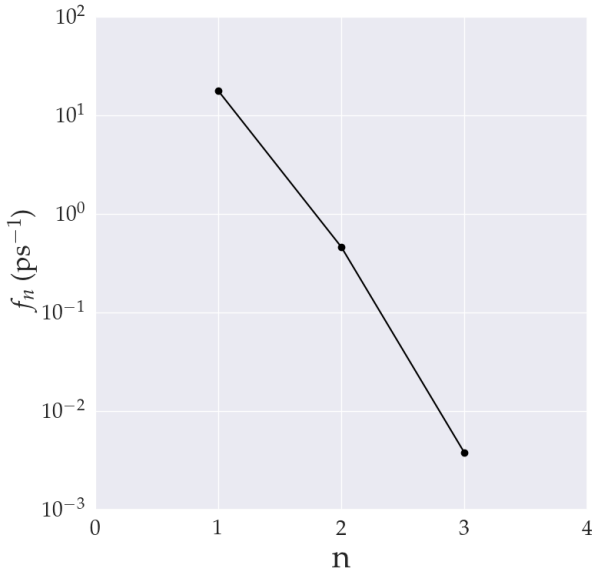
(f) `simultaneous_jumps.dat`: lists the number of simultaneous jumps (NSjumps) against time. Modifying this to time intervals with `-deltf` is also possible (refer to the **Settings** section). An example of the content is shown below (`-delt` 20):

Frame	NSjumps
1	0
2	47
3	38
4	24
.	.
.	.
.	.



(g) `string_frequency.dat`: details the probability of strings of atoms involved in cooperative diffusion. A comprehensive description of this file can be found in the manuscript titled "Name of the Manuscript" by Henry A. Cortés, Mauricio R. Bonilla, and Elena Akhmatskaya.

StringLength (#n)	f_n(ps^-1)
1.0000	0.0268
2.0000	0.7743
3.0000	0.1306
4.0000	0.0404
.	.
.	.
.	.



(h) `SitesMap_OVITO.dat`: this text file can be opened with the free visualization software `OVITO` (`https://www.ovito.org/`), enabling the easy visualization of the sites map.

<d_i> Clustering

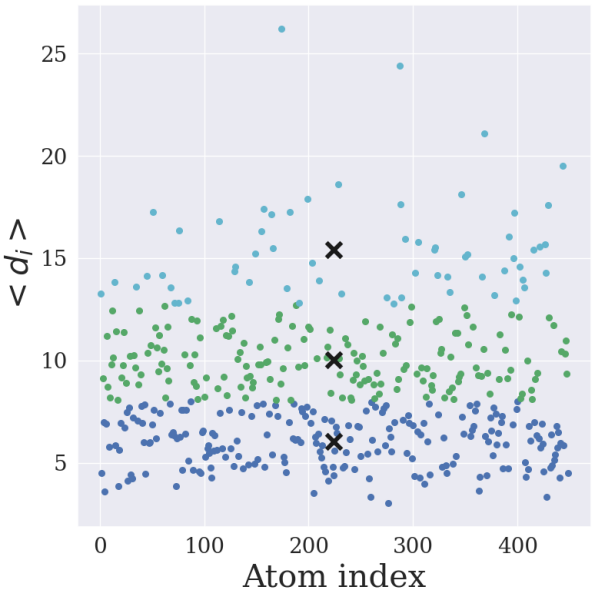
1. When the <d_i> clustering is enabled using the `-clus 1` in the `DensityMap` command (see the **Settings** section for more details), the format of the output is modified. The output includes information about the trajectory segment utilized for <d_i> clustering and the number of atoms in each cluster. For a detailed explanation of the <d_i> clustering methodology, please refer to the study by Cortes *et al.* [3]. See screen output example (`-verb or verbosity 0`):

```
>> Reading the trajectory <<
>> Calculating density map <<
Format: XTC
Number frames DensityMap: 80001
Diffusive atoms: 448
Voxel size: 0.2
Number voxels: 2197000

>> Computing <di> clustering <<
```

```
Initial frame: 8000
Final frame: 72000
Total Steps used: 64000
>> Done <<

Number of atoms per <di> cluster:
L: 212
M: 174
H: 62
HM: 236
>> Calculating density map for <di> clusters <<
Time interval between frames DensityMaps: 0.0500
Time interval between frames VoxelIndices: 0.0500
Number of frames VoxelIndices: 80001
Writing files: L-DensityMap.dat, L-VoxelIndices.dat,
   ↪ CHGCAR_VESTA-L
Writing files: HM-DensityMap.dat, HM-VoxelIndices.dat
   ↪ , CHGCAR_VESTA-HM
>> Done <<
```

Furthermore, the following files are generated:

(a) `di_avg.png`: the average displacement results for $<d_i>$ clustering, specifically for the L, M, and H clusters. For further details on this methodology, refer to the study by Cortes *et al.*[3].



(b) Extra cluster files: As outlined in the previous section, several files are generated based on the verbosity level set in the commands. These include:

`-verb 0`

- `L-DensityMap.dat`, `L-VoxelIndices.dat` and `CHGCAR_VESTA-L`
- `HM-DensityMap.dat`, `HM-VoxelIndices.dat`, `CHGCAR_VESTA-HM`

`-verb 1`

- `L-DensityMap.dat`, `L-VoxelIndices.dat` and `CHGCAR_VESTA-L`
- `M-DensityMap.dat`, `M-VoxelIndices.dat` and `CHGCAR_VESTA-M`
- `H-DensityMap.dat`, `H-VoxelIndices.dat` and `CHGCAR_VESTA-H`
- `HM-DensityMap.dat`, `HM-VoxelIndices.dat` and `CHGCAR_VESTA-HM`

2. To effectively use the `CrySF` routine on the $<d_i>$ cluster files, it is essential first to generate the density map of the entire trajectory. This involves running the `DensMap` command with the option `-clus` 0. After this step, ensure that the folder contains the necessary files (`-verb 0`):

   - `L-DensityMap.dat`
   - `HM-DensityMap.dat`
   - `DensityMap.dat`
   - `L-VoxelIndices.dat`
   - `HM-VoxelIndices.dat`
   - `VoxelIndices.dat`.

   When `CrySF` is executed with `-clus` 1, the output includes the following information related to $<d_i>$ clusters, displayed directly on the screen (`-verb or verbosity 0`):

```
>> Sites map of <di> clusters <<

Reading file: HM-DensityMap.dat
Writing file: HM-SitesMap.dat

Information for site type
Label site: 0, Number of Sites: 184
Label site: 1, Number of Sites: 348

Reading file: HM-VoxelIndices.dat
Writing file: HM-jumps_info.dat
```

```
Writing file: HM-simultaneous_jumps.dat
Writing file: HM-string_probability.dat

Reading file: L-DensityMap.dat
Writing file: L-SitesMap.dat

Information for site type
Label site: 0, Number of Sites: 186
Label site: 1, Number of Sites: 358

Reading file: L-VoxelIndices.dat
Writing file: L-jumps_info.dat
Writing file: L-simultaneous_jumps.dat
Writing file: L-string_probability.dat
```

Additionally, the following files are generated:

`-verb or verbosity 0`

- `L-SitesMap.dat`, `L-jumps_info.dat`, `L-simultaneous_jumps.dat` and `L-string_probability.dat`
- `HM-SitesMap.dat`, `HM-jumps_info.dat`, `HM-simultaneous_jumps.dat` and `HM-string_probability.dat`

`-verb or verbosity 1`

- `L-SitesMap.dat`, `L-jumps_info.dat`, `L-simultaneous_jumps.dat` and `L-string_probability.dat`
- `M-SitesMap.dat`, `M-jumps_info.dat`, `M-simultaneous_jumps.dat` and `M-string_probability.dat`
- `H-SitesMap.dat`, `H-jumps_info.dat`, `H-simultaneous_jumps.dat` and `H-string_probability.dat`
- `HM-SitesMap.dat`, `HM-jumps_info.dat`, `HM-simultaneous_jumps.dat` and `HM-string_probability.dat`

Doped systems

1. To effectively use the `CrySF` routine on a doped structure, it is essential to use `DesnsityMap` command on both the undoped and doped structure separately. Subsequently, rename the output files `DensityMap.dat` and `VoxelIndices.dat` for the doped structure as:

- `Dope-DensityMap.dat`
- `Dope-VoxelIndices.dat`

2. Then, execute `CrySF` command with the option `-dop` 1. The following snippet illustrates the output displayed on the screen.

```
>> Sites map of the doped structure <<

Reading file: Dope-DensityMap.dat
Writing file: Dope-SitesMap.dat

Information for site type
Label site: 0, Number of Sites: 192
Label site: 1, Number of Sites: 384

Reading file: Dope-VoxelIndices.dat
Writing file: Dope-jumps_info.dat
Writing file: Dope-simultaneous_jumps.dat
Writing file: Dope-string_probability.dat
```

   As above, A set files with the `Dope-` prefix are created:

   - `Dope-SitesMap.dat`
   - `Dope-jumps_info.dat`
   - `Dope-simultaneous_jumps.dat`
   - `Dope-string_probability.dat`

3. In addition, it is possible to combine `-dop` 1 with `-clus` 1, the output files for the $<d_i>$ clusters of the doped structure would have to be renamed as:

   - `L-Dope-DensityMap.dat` and `L-Dope-VoxelIndices.dat`
   - `HM-Dope-DensityMap.dat` and `HM-Dope-DensityMap.dat`

   Then, execute the `CrySF` command with `-dop` 1 `-clus` 1.

References

[1] M.-S. Lim and S.-H. Jhi, "First-principles study of lithium-ion diffusion in β-Li_3PS_4 for solid-state electrolytes," *Current Applied Physics*, vol. 18, no. 5, pp. 541–545, 2018.

[2] R. L. Kam, K. Jun, L. Barroso-Luque, J. H. Yang, F. Xie, and G. Ceder, "Crystal structures and phase stability of the Li_2S–P_2S_5 system from first principles," *Chem. Mater.*, vol. 35, no. 21, pp. 9111–9126, 2023.

[3] H. A. Cortés, M. R. Bonilla, H. Früchtl, T. van Mourik, J. Carrasco, and E. Akhmatskaya, "A data-mining approach to understanding the impact of multi-doping on the ionic transport mechanism of solid electrolytes materials: the case of dual-doped $Ga_{0.15}/Sc_y$ $Li_7La_3Zr_2O_{12}$," *J. Mater. Chem. A*, vol. 12, no. 9, pp. 5181–5193, 2024.