

Data extraction & analysis example

To demonstrate the usage of the Data extraction & Data analysis tools, we performed an example analysis of 3 aspirin structures included in the CSD:

- ACSALA02 (aspirin I),
- ACSALA13 (aspirin II),
- ACSALA24 (aspirin IV).

Directory structure

To start with, we create the necessary directories within the main working directory (e.g. Documents):

```
Crystal Math
├── CSD_DB_Analysis
│   ├── DB_Data
│   ├── Plots
│   │   ├── Contacts
│   │   └── H-Bonds
│   ├── Source_Code
│   │   ├── input_files
│   │   │   ├── input_data_extraction.txt
│   │   │   └── input_data_analysis.txt
│   └── Source_Data
│       └── fragment_list.json
```

All the code files *.py provided should be placed in the Source.Code directory. The input files input_data_extraction.txt, input_data_analysis.txt are placed in the input_files directory and the user generated fragment_list.json is placed in the Source.Data directory.

Fragments preparation

The first step is to run the crystal_math_reference_fragments.py that will convert the user generated fragment list into the standardized fragments format that will be written in the file reference_fragment_list.json within the Source.Data directory. The user generated fragment_list.json must contain all the fragments of interest. For aspirin, the fragments of interest are: acetic_acid, benzene and methyl_acetate.L.

The data extraction input file

Once the reference_fragment_list.txt is generated we can create the input_data_extraction.txt file to extract data from the CSD. In this example the input file is as follows:

```
1 {"save_directory": "../CSD_DB_Analysis/DB_Data/",
2  "fragments_input_file": "reference_fragment_list.json",
3  "structure_data_file": "ACSALA_structure_data.txt",
4  "contacts_data_file": "ACSALA_contacts_data.txt",
5  "h-bonds_data_file": "ACSALA_h-bonds_data.txt",
```

```

6 "plane_intersection_data_file": "ACSALA_plane_intersection_data.txt",
7 "fragments_geometry_data_file": "ACSALA_fragments_geometry_data.txt",
8 "structures_list": ["CSD", ["ACSALA02", "ACSALA13", "ACSALA24"]],
9 "target_species": ["C", "H", "N", "O", "S", "F", "Cl", "Br"],
10 "target_space_groups": ["P1", "P-1", "P21", "C2", "Pc", "Cc", "P21/m", "C2/m", "P2/c", "P21/c", "P21/n", "C2/c", "P21212", "P212121", "Pca21", "Pna21", "Pbcn", "Pbca", "Pnma"],
11 "target_z_prime_values": [1, 2, 3, 4, 5],
12 "molecule_weight_limit": 500.0,
13 "add_full_component": False,
14 "center_molecule": False,
15 "fragments_to_check_alignment": [],
16 "alignment_tolerance": 0.20,
17 "visualize_eigenvectors": True,
18 "proposed_vectors_n_max": 5}

```

Below is the explanation of the entries in the input file.

- **save_directory**: The directory in which the extracted data will be saved.
- **structure_data_file**: The file to write structure data.
- **contacts_data_file**: The file to write contacts data.
- **h-bonds_data_file**: The file to write H-bonds data.
- **plane_intersection_data_file**: The file to write the data for the intersection of the fragment principal planes to the edges of the unit cell.
- **fragments_geometry_data_file**: The file to write the fragments geometry data.
- **structures_list**: The first entry is set to CSD since we read structures from the CSD database, while the second entry is a list of the structures that will be examined.
- **target_species**: The allowed atomic species. In case a structure has an atom not included in the list, the structure will not be examined.
- **target_space_groups**: The allowed space groups. In case a crystal is in a space group not included in the list, the structure will not be examined.
- **target_z_prime_values**: The allowed Z' values. In case a structure has a Z' value not included in the list, the structure will not be examined.
- **molecule_weight_limit**: The maximum allowed molecular weight.
- **add_full_component**: Set to True if we need to get the properties of the complete molecule as a single fragment.
- **center_molecule**: Set to True if we want the reference molecule to be translated within the reference unit cell.
- **fragments_to_check_alignment**: A list of the specific fragments we need to analyze. If set to [], the algorithm will search for all the fragments in the **fragment_list.json** file,

- **alignment_tolerance**: The maximum allowed RMSD value to align fragments when estimating the principal axes of inertia.
- **visualize_eigenvectors**: Set to `True` if we need to visualize the principal axes/planes of inertia.
- **proposed_vectors_n_max**: The maximum value of n for the equivalent vectors set \mathbf{n}_c .

Extract data

The next step is to extract the data for the target structures by running `crystal_math_csd_data_extraction.py`. Once the extraction is completed, it will generate 5 files in the `save_directory`.

Data extraction output files

Each entry in the `structure_data_file` has 16 columns:

1. Structure name.
2. Space group.
3. Z value.
4. Z' value.
5. Composition (atomic species only - not formula).
6. Scaled a value (always 1.0).
7. Scaled b value.
8. Scaled c value.
9. Actual cell a value (\AA).
10. Actual cell b value (\AA).
11. Actual cell c value (\AA).
12. Cell α value.
13. Cell β value.
14. Cell γ value.
15. Unit cell volume (\AA^3).
16. vdW free volume (%).
17. Solvent accessible surface (%).

Each entry in the `contacts_data_file` has 9 columns:

1. Structure name.
2. Atom 1 label.
3. Atom 2 label.
4. Atom 1 species.
5. Atom 2 species.
6. Contact type.
7. Is in line of site.
8. Contact length.
9. Contract strength.

Each entry in the `h-bonds_data_file` has 12 columns:

1. Structure name.
2. Atom 1 label (Donor).
3. Atom 2 label (H atom).
4. Atom 3 label (Acceptor).
5. Atom 1 species.
6. Atom 2 species.
7. Atom 3 species.
8. Contact type.
9. Is in line of site.
10. Contact length.
11. Donor-acceptor distance.
12. Bond angle.

Each entry in the `plane_intersection_data_file` has $39N_f+2$ columns (in our case 119 columns), where N_f the number of fragments:

1. Structure name.
2. Fragment name.
- 3-5. Fragment 1 principal plane 1 normal vector.
- 6-41. Fragment 1 principal plane 1 intersections.

42-44. Fragment 2 principal plane 2 normal vector.

45-80. Fragment 2 principal plane 2 intersections.

81-83. Fragment 3 principal plane 3 normal vector.

84-119. Fragment 3 principal plane 3 intersections.

Each entry in the `fragments_geometry_data_file` has $51 + 3N_a$ columns, where N_a is the number of atoms in the fragment:

1. Structure name.
- 2-4 Scaled cell vectors.
- 5-7. Cell angles.
8. Fragment name.
9. N_a .
- 10-12. Components of the first inertia tensor eigenvector \hat{e}_1 (physical coordinates).
- 13-15. Components of the second inertia tensor eigenvector \hat{e}_2 (physical coordinates).
- 16-18. Components of the third inertia tensor eigenvector \hat{e}_3 (physical coordinates).
- 19-21. Components of the first inertia tensor eigenvector \hat{w}_1 (fractional coordinates).
- 22-24. Components of the second inertia tensor eigenvector \hat{w}_2 (fractional coordinates).
- 25-27. Components of the third inertia tensor eigenvector \hat{w}_3 (fractional coordinates).
- 28-30. Components of the nearest eigenvector in the set \mathbf{n}_c to \hat{e}_1 .
31. Respective angle.
- 32-34. Components of the nearest eigenvector in the set \mathbf{n}_c to \hat{e}_2 .
35. Respective angle.
- 36-38. Components of the nearest eigenvector in the set \mathbf{n}_c to \hat{e}_3 .
39. Respective angle.
- 40-42. Components of the nearest eigenvector in the set \mathbf{n}_c to \hat{w}_1 .
43. Respective angle.
- 44-46. Components of the nearest eigenvector in the set \mathbf{n}_c to \hat{w}_2 .
47. Respective angle.
- 48-50. Components of the nearest eigenvector in the set \mathbf{n}_c to \hat{w}_3 .
51. Respective angle.
- 52-. Coordinates of the atoms in the fragment $(x_1, y_1, z_1), (x_2, y_2, z_2), \dots$

Post extraction analysis

Post extraction analysis can be performed by running the `crystal_math_csd_data_analysis.py` tool.

The data analysis input file

The input file for the analysis is named `input_data_analysis.txt` and should be placed in the `input_files` folder. In this example the input file is as follows:

```
1 {"structure_data_file": "../CSD_DB_Analysis/DB_Data/ACSALA_structure_data.txt",
2  "contacts_data_file": "../CSD_DB_Analysis/DB_Data/ACSALA_contacts_data.txt",
3  "h_bonds_data_file": "../CSD_DB_Analysis/DB_Data/ACSALA_h-bonds_data.txt",
4  "contacts_plots_dir": "../CSD_DB_Analysis/Plots/ACSALA_Contacts/",
5  "h_bonds_plots_dir": "../CSD_DB_Analysis/Plots/ACSALA_H-Bonds/",
6  "contacts_bin_width": 0.025,
7  "contacts_max_length": 5.0,
8  "h_bonds_bin_width": 0.025,
9  "h_bonds_angle_bin_width": 1.0,
10 "h_bonds_max_length": 4.0,
11 "h_bonds_da_max_length": 4.0,
12 "h_bonds_min_angle": 120.0,
13 "h_bonds_max_angle": 180.0
14 }
```

Below is the explanation of the entries in the input file.

- `structure_data_file`: The file to read structure data (same as for data extraction).
- `contacts_data_file`: The file to read contacts data (same as for data extraction).
- `h-bonds_data_file`: The file to read H-bonds data (same as for data extraction).
- `contacts_plots_dir`: The directory to save contact plots.
- `h_bonds_plots_dir`: The directory to save H-bond plots.
- `contacts_bin_width`: The bin width for the contacts histograms.
- `contacts_max_length`: The maximum length for the contact histograms.
- `h_bonds_bin_width`: The bin width for the H-bond length histograms
- `h_bonds_angle_bin_width`: The bin width for the H-bond angle histograms
- `h_bonds_max_length`: The maximum length for the H-bond histograms.
- `h_bonds_da_max_length`: The maximum length for the donor-acceptor distance histograms
- `h_bonds_min_angle`: The minimum angle for the H-bonds.
- `h_bonds_max_angle`: The maximum angle for the H-bonds.

Create plots

By running `crystal_math.csd_data_analysis.py` the script will generate plots for the close contacts and hydrogen bonds for every atomic pair encountered in the data files. The plots will be generated for all space groups, as well as for each space group separately, to check if there is a dependency on the space group setting. In the case of aspirin structure analysis, the plots can be found in the following directories

```
Crystal_Math
├── CSD_DB_Analysis
│   ├── DB_Data
│   └── Plots
│       ├── Contacts
│       │   ├── All_Space_Groups
│       │   └── P21/c
│       └── H-Bonds
│           ├── All_Space_Groups
│           └── P21/c
```

In this example, all structures are found in the $P2_1/c$ space group, so the algorithm generated a single space group folder.