DigiFAB/FoNS datathon 2021

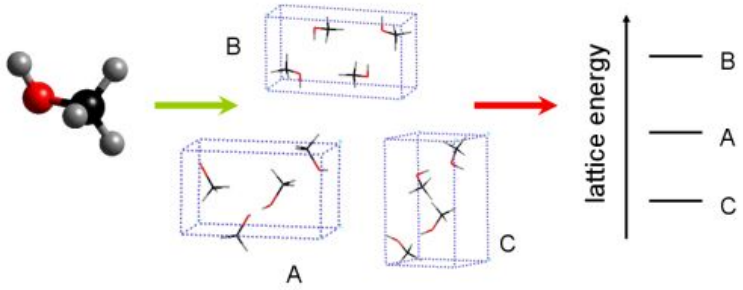
AI to speed up crystal structure prediction

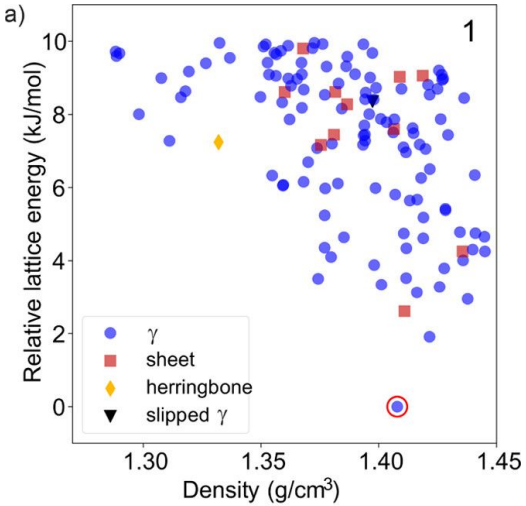
Motivation

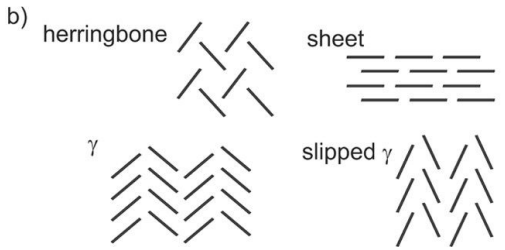
From knowledge of a molecule's structure alone, can we predict the solid-state packing of that molecule?

- Crystal structure prediction (CSP) is the first principles calculation of the packing of molecules
- From materials to drug discovery, performance is dependent on crystal packing (1)
- A recent paper on this problem: paper
- CSP is computationally expensive
- The Cambridge structural database (CSD) contains experimental crystal structures

Aim to use this data to improve CSP methods



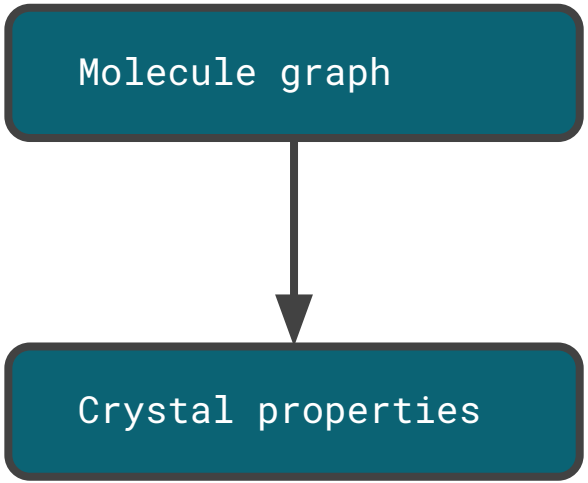




Goals of the Datathon

From descriptors of molecules to properties of crystal structures

- Data provided:
  - Properties of the crystal structure and contacts within it
    - Density/packing efficiency
    - What contacts a molecule forms in the solid-state
    - Packing shell
  - Descriptors and fingerprints of the molecule



Dataset size: 16810 entries; 80:20 split into training and test set

Pre-reading

Tools:

- What is Z'?
- RDKit: Getting Started In Python
- https://scikit-learn.org/stable/
- Github guides
- Blind tests: CSP Blind Tests

Research articles:

- Space group selection for crystal structure prediction of solvates
- Which conformations make stable crystal structures?
- Data-efficient machine learning for molecular crystal structure prediction
- Large-Scale Computational Screening of Molecular Organic Semiconductors Using Crystal Structure Prediction

Your data *snowpeople*:

Spot them with “(helper)” in the name

Steven, Harry, Florian, Andrew

Some answers to FAQS:

- We provide a list of main targets and side targets of different weightings
- Ask for datathon help by:
  - Raise hand (other emojis may work too)
  - Message a helper privately using the gather town chat function
  - Coming up to and chatting with a helper - be patient if they are in another room

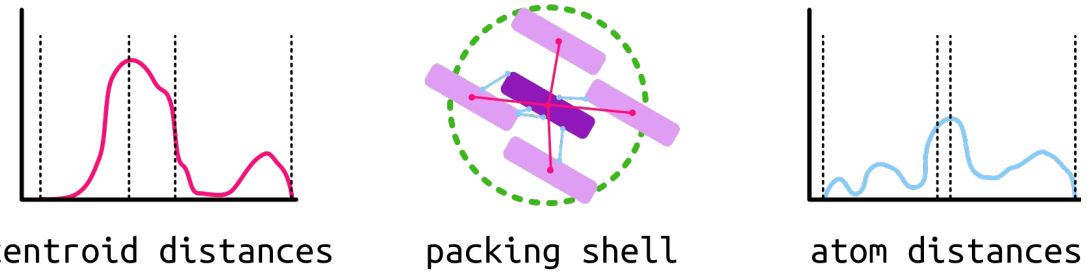
Useful things:

Packages:

- Mordred (2D and 3D descriptors): mordred-descriptor/mordred: a molecular descriptor calculator
- RDKit (2D descriptors): rdkit.Chem.Descriptors module — The RDKit 2020.09.1 documentation
- DScribe (3D descriptors): DScribe — DScribe 0.4.x documentation

Describing crystals:

- We extracted numerical and classification properties of crystal structures from the CSD
- We simplified the packing geometry in the crystal to a twelve molecule packing shell:
  - Atom distances from all atoms in central molecule to all surrounding molecules
  - Centroid distances from central molecule



Notes on collaborative coding

We need the code at the end of the day and will aim to rerun!

Split up specific tasks and steps, then merge code or functions through **Teams** channel, **Google Colab** or **Github** into one `main` code.

- person A writes the code to read in, split and prep data

- person B writes the code to build model 1 for task 1
- person C writes the code to build model 2 for task 1

Merge, run and have better chance at optimising models for one task

- person B writes the code to build all models for task 1
- person C writes the code to build all models for task 2

Merge, run and have consistent progress on two tasks

Notes about the Data

- The data is split into a train and test set for evaluation - **split further for validation**
- We have **precalculated descriptors and fingerprints**, such that tasks can be run “out-of-the-box”
- We provide molecule structures for more advanced uses
  - I.e. to generate graphs from molecule structures
  - Or to recalculate descriptors/fingerprints
- To solve these tasks, **you can not use data external to that provided by us**; the data provided can be manipulated in any way you see fit
- Any programming language or packages or models can be used.

Files in shared box folder (each has a test equivalent for final scoring):

OUTPUT

- **train_crystals.csv : data to predict about crystal structures**
- **train_distances.csv : atom distance data to predict about packing in crystal structures**
- **train_centroid_distances.csv : centroid distance data to predict about packing in crystal structures**

INPUT

- train_descriptors.csv : 2D descriptors of all molecules in crystal structures
- train_mols.sdf : Structure of lowest energy conformer used in 3D descriptors of molecules in crystal structures
- train_mord3d.csv : 3D Mordred descriptors of all molecules in crystal structures
- train_morgan.csv : Morgan fingerprints of all molecules in crystal structures
- train_rdk.csv : RDKit fingerprints of all molecules in crystal structures

Notes about the Data: crystal structure properties

Crystal structure properties: (+ the packing shell) - from the CSD (The CSD)

Data from the CSD is licensed to Imperial. Therefore, we are only providing manipulated data today. We can also provide Python scripts to extract and manipulate the data used today, but you will require access to the CSD to run them and we will not share those today.

identifier	n_heavy_atoms	a	b	c	alpha	beta	gamma	z_value	z_prime	spacegroup_symbol	contacts	cell_volume	calculated_density	packing_coefficient	void_volume	is_centrosymmetric	is_sohncke
AAPYPE	14	7.024	12.42	12.28	90	108.4	90	4	1	P21/c	(5, 2)	1016.515331	1.255993926	0.607740227	0	TRUE	FALSE
ABACEN	13	6.1409	5.1952	13.1844	90	95.118	90	2	1	P21	(14, 5, 3, 0)	418.9476059	1.507357638	0.712741559	0	FALSE	TRUE
ABADOX	14	10.0469	9.5169	10.5877	90	110.317	90	4	1	P21/a	(7, 2, 1)	949.3643888	1.477905517	0.692363423	0	TRUE	FALSE

Notes about the Data: molecule properties

- We provide **.sdf** files of molecular structures that are not necessary in general. However, using RDKit, the molecular graphs can be used as input to models. These structures are not the conformer in the crystal structure!
- The remaining descriptors and fingerprints are provided as simple **.csv** files for each entry to use as numerical input into models - these can be used however you want and in whatever combination!
- All descriptors and fingerprints can also be recalculated using the structures or smiles strings available in the **INPUT** data
- Many descriptors are cheminformatic measures that are *difficult* to grasp from a chemical perspective.
- Careful of “na” values in descriptors!

Columns in data:

- train_descriptors.csv : has 1617 (988) descriptors (columns)
- train_mols.sdf : has 13449 molecular structures
- train_mord3d.csv : has 176 descriptors (columns)
- train_morgan.csv : 1024 bit vectors (columns)
- train_rdk.csv : 1024 bit vectors (columns)

Prediction Targets

Main tasks:

1. Density of crystal structure (column: ‘calculated_density’) - **regression**
2. Is centrosymmetric (column: ‘is_centrosymmetric’) - **categorisation**
3. Mean of **centroid** distances in packing shells (column: ‘mean’ in **train_centroid_distances.csv**) - **regression**
4. Number of van der Waals contacts from **atom** distances (column:’n_vdw_contacts’ in **train_distances.csv**) - **regression**

Side tasks (worth slightly less):

- Packing coefficient of crystal structure (column: ‘packing_coefficient’) - **regression**
- Cell volume (column: ‘cell_volume’) - **regression**
- Space group symbol (column: ‘spacegroup_symbol’) - **categorisation**

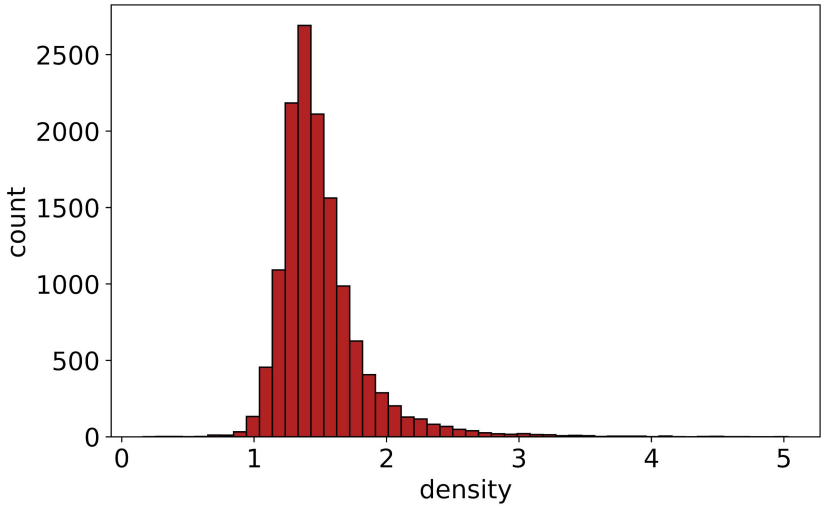
Side tasks (for fun...):

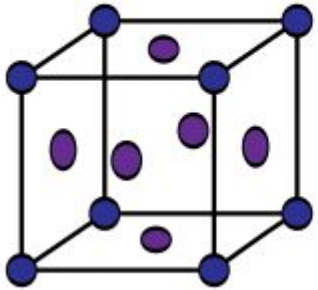
- Z prime (column: ‘z_prime’) - **regression/categorisation**
- Number of inter/intra-molecular vdw interactions (columns: ‘vdw_inter’, ‘vdw_intra’) - **regression**
- Number of inter/intra-molecular H bonds (columns: ‘hbond_inter’, ‘hbond_intra’) - **regression**
- Distribution of **atom** distances (columns: ‘min’, ‘max’, ‘mode’, ... in **train_distances.csv**) - **regression**
- Distribution of **centroid** distances (columns: ‘min’, ‘max’, ‘mode’, ... in **train_centroid_distances.csv**) - **regression**

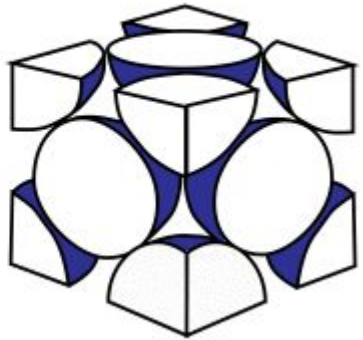
Task 1 - Crystal density

Use a regression model to predict the density of the crystal structure

- Notes:
  - CSP requires generation of structures - this could help narrow this down
  - The error on prediction (i.e. range of valid densities) is useful too!
- Column: **'calculated_density'**
- File: **train_crystals.csv**



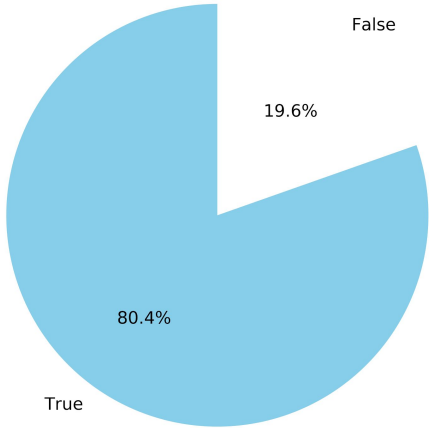


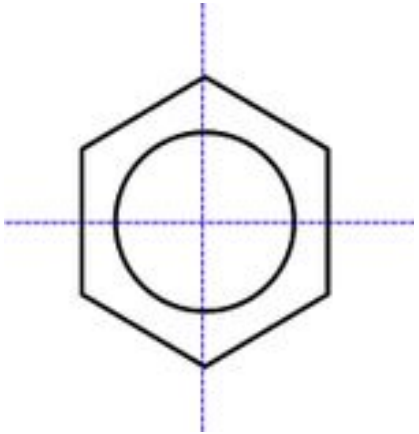


Task 2 - Crystal symmetries

Use a classification model to predict if a crystal structure will pack in a centrosymmetric way

- Notes:
  - In crystallography, a centrosymmetric point group contains an inversion center as one of its symmetry elements.
  - Narrows down the number of space groups that need to be tested in a crystal structure prediction workflow
- Column: **‘is_centrosymmetric’**
- File: **train_crystals.csv**

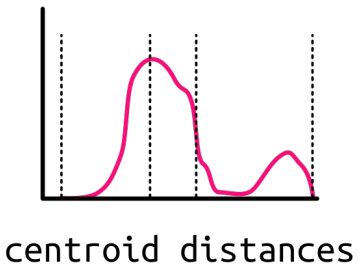


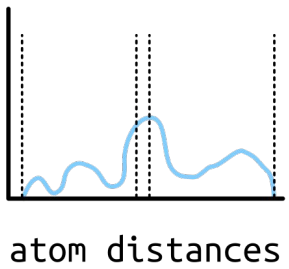


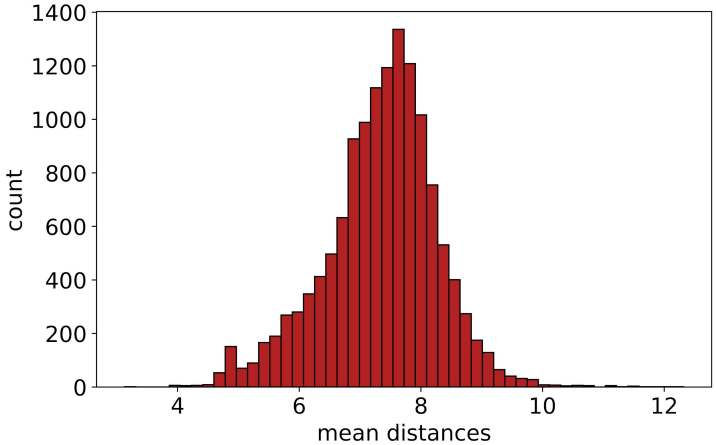
Task 3 - Centroid distance distribution

Use a regression model to predict the **mean** of these distributions:

- Notes:
  - CSP requires generation of structures - this could help narrow down where to place molecules
  - The error on prediction is useful for a CSP search
- Column: **'mean'**
- File: **train_centroid_distances.csv**



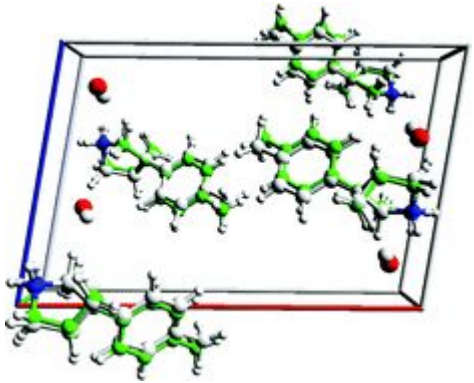


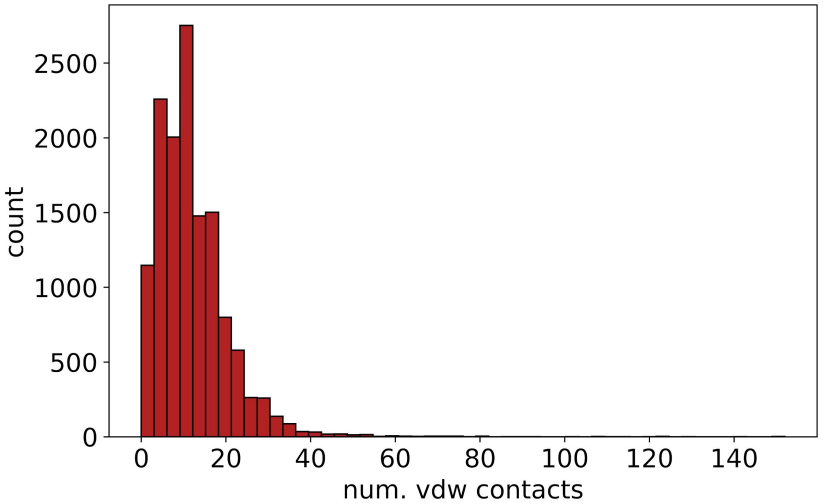


Task 4 - Molecular contacts

Use a regression model to predict the likely number of vdw contacts between molecules in a crystal structure

- Notes:
  - We have two measures for this!
  - CSP requires generation of structures - this could help narrow down where to place molecules
- Column: **‘n_vdw_contacts’**
- File: **train_distances.csv**





Quantifying Model Accuracy

Regression Problems

Normalised mean absolute error: $$nMAE = \frac{\frac{1}{n}\sum|f_t - a_t|}{\frac{1}{n}\sum a_t} = \frac{\sum|f_t - a_t|}{\sum a_t}$$

```
from sklearn.metrics import mean_absolute_error
```

Classification Problems

F1 Score: `F1 = 2 * (precision * recall) / (precision + recall)` [A blog post]

For multi-class problems: “macro-averaged” F1-score

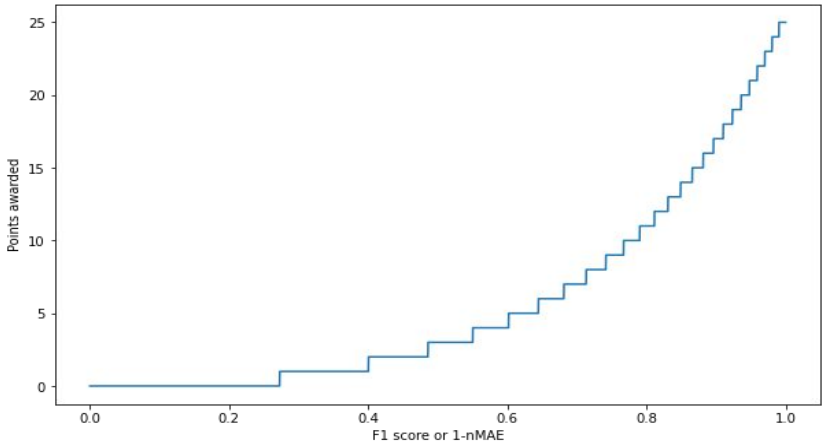
```
from sklearn.metrics import f1_score
```

Use the parameter: average=”macro”

Assessing Overall Performance

Each **main task** will be scored **out of 25 points** in the following way:

- For classification problems, we will compute the **macro F1 score (0=worst, 1=best)**
- For regression problems, we will compute **1-nMAE (<0=worst, 1=best)**
- We award points according to the following exponential distribution:



- We encourage you to spend time optimising models, as a lot of points become available for good predictions

Assessing Overall Performance

- **Side tasks** will be awarded bonus points **out of 10** on a similar exponential scale
- We encourage you to at least attempt the main tasks, before solving the bonus tasks
- Your overall team performance will be your **total points**
- In the case of **ties**, we will base the final verdict on the presentations given at the end of the day, which will be judged by a jury formed by the scientific committee

Submitting predictions:

- Throughout the day, you can submit predictions to see how well you're doing so far
- Please submit your predictions in the **csv** file format with a **single column of predictions**
    - Each task should come as a separate file

The automatic scoring checking system

- Throughout the day, you will be able to automatically check your predictions and obtain your projected total points using the following GitHub repository:

  https://github.com/stevenkbennett/fons_datathon_testing

- If you're unsure as to how to use this, please do not hesitate to contact any one of your friendly helpers!

- We will be using this system to obtain the final point scores at the end of the day. However, if you do not want to use this yourself, contact us and we will upload and score your predictions for you.

Timeline for the day

Find Feng Li for general questions

09:30-10:00 -- Welcome and introductions

10:00-12:30 -- Datathon

11:30 - submit models for progress testing **[see: a helper]** (we will highlight how you are doing compared to the rest)

12:30-13:00 -- Lunch

13:00-13:40 -- Keynote speaker: Dr Edward O. Pyzer-Knapp

13:45-16:00 -- Datathon

13:30 - submit models for progress testing **[see: a helper]** (we will highlight how you are doing compared to the rest)

15:30 - start submitting final models for testing

15:50 - all final models for evaluation must be submitted!

16:00-17:00 -- Presentations, prizes and closing remarks

16:00-16:15 - top 5-6 teams present 2 minute outcomes **[ensure you are preparing a presentation throughout]**

16:30-16:45 - final decisions