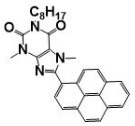
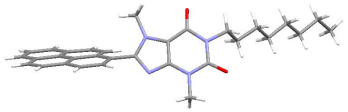
Yunping, Weishi, Vivian, Minh, Guanning
2/20/2020

Background



Materials syntheses

Interpreting K-space diffraction data



Fluorescence and electron transport

Proposed **2D** chemical structure → Experiment #1 → .... → Experiment #N E(N) >10 → Optimized **3D** molecular conformation → Property explanation or prediction

What we do via ML

Less lab work but accelerated science discovery

Why not DFT?

Function and packages

Format to feed into ML: Simplified molecular-input line-entry system

SMILES

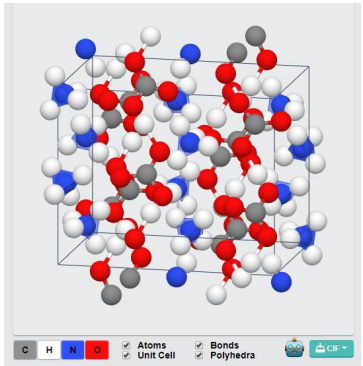
OpenEye Python

2D dataframe

ML processing

3D dataframe

N≡N	N#N
$CH_3-N=C=O$	CN=C=O
$Cu^{2+}SO_4^{2-}$	[Cu+2].[O-]S(=O)(=O)[O-]
	O=Cc1ccc(O)c(OC)c1 COCc1cc(C=O)ccc1O



Visualization

.CIF files

Storage

Function and packages

Building data corpus: **OpenEye**, **CIF parser** and **pandas**

What is in a Crystallographic Information File (**CIF**):

a standard text file format for representing crystallographic information, promulgated by the International Union of Crystallography

Accessing with **CIF parser**

Reading, Modification, Writing CIF files

```
col_x = block.find_values('_atom_site_fract_x')
for n, x in enumerate(col_x):
    print (x[0:7])
#     col_x[n] = float(x[:7])
```

```
0.36787
0.42713
0.28794
0.27273
0.44785
0.46355
0.33048
0.40473
0.34603
0.38912
0.46319
0.49136
0.30783
0.27107
```

SMILES from CIF with **CCDC**

CIF contains SMILES, which will be extracted with CDCC

2D structure with **OpenEye**

SMILES will be converted to 2D structure with OpenEye

CCDC

The Cambridge Crystallographic Data Center

OpenEye SCIENTIFIC

Crystallography Open Database

Search results

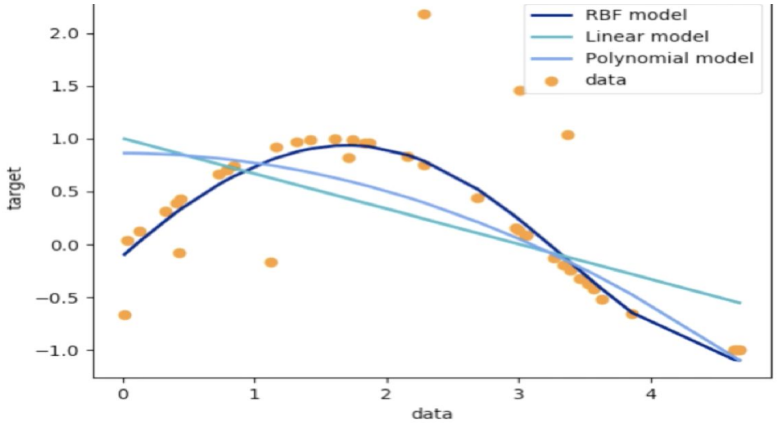
Result: there are 41351 entries in the selection

CIF files at an open database

```
loop_
 _atom_site_label
 _atom_site_type_symbol
 _atom_site_fract_x
 _atom_site_fract_y
 _atom_site_fract_z
 _atom_site_U_iso_or_equiv
 _atom_site_thermal_displace_type
 _atom_site_occupancy
 _atom_site_calc_flag
 _atom_site_refinement_flags
 _atom_site_disorder_group
S1 S 0.36787(3) 0.0394(3) 0.34408(9) 0.0131(3) Uani 1 d . .
S2 S 0.42713(3) 0.4577(3) 0.67768(10) 0.0153(4) Uani 1 d . .
C8 C 0.28794(12) 0.1237(10) 0.3999(4) 0.0116(8) Uani 1 d . .
H8 H 0.27273(12) 0.0311(10) 0.3261(4) 0.014 Uiso 1 calc R .
C6 C 0.44785(11) 0.1980(11) 0.4837(4) 0.0148(9) Uani 1 d . .
H6 H 0.46355(11) 0.1182(11) 0.4269(4) 0.018 Uiso 1 calc R .
C5 C 0.33048(12) 0.1520(11) 0.4251(4) 0.0119(8) Uani 1 d . .
C4 C 0.40473(13) 0.1861(11) 0.4647(4) 0.0141(9) Uani 1 d . .
C3 C 0.34603(12) 0.2968(11) 0.5391(4) 0.0115(8) Uani 1 d . .
C2 C 0.38912(12) 0.3143(11) 0.5615(4) 0.0118(8) Uani 1 d . .
C1 C 0.46319(13) 0.3396(12) 0.5945(4) 0.0176(9) Uani 1 d . .
H1 H 0.49136(13) 0.3709(12) 0.6238(4) 0.021 Uiso 1 calc R .
S3 S 0.30783(3) 0.4038(3) 0.61561(9) 0.0112(3) Uani 1 d . .
C7 C 0.27107(11) 0.2462(10) 0.4951(4) 0.0109(8) Uani 1 d . .
```

Machine Learning Method

Regression



Package: Scikit-learn

- Integrates well with Numpy
- provides useful utilities for splitting data, computing common statistics

Support Vector Machine, SVM

- more effective in high dimensional spaces
- effective in cases where number of dimensions is greater than the number of samples

Decision Tree/Regression Tree

- does not require normalization of data
- requires less effort for data preparation during pre-processing

Gradient Boost Tree (also available in sklearn)

Boosting is an ensemble technique where new models are added to correct the errors made by existing models.

1. A loss function to be optimized.
2. A weak learner to make predictions.
3. An additive model to add weak learners to minimize the loss function.

Package: XGboost

Advantages:

- Execution Speed
  XGBoost was almost always faster than the other benchmarked implementations from R, Python
- Model Performance
- Support Regression

Disadvantages:

- Difficult for people to interpret
- Require careful tuning of learning rate or other parameters
- Learning curve

Questions?