Materials Informatics: Binary and ternary composition and structure featurizer for ML models

Sangjoon Lee[1] <sl5400@columbia.edu>, Emil I. Jaffal[2,3], Anirudh Machathi[4], Danila Shiryaev[3], Alex Vtorov[3], Nikhil J. Barua [5], Holger Kleinke[5], Anton O. Oliynyk[2,3] <anton.oliynyk@hunter.cuny.edu>

in collaboration with Nishant Yadav, Siddha Sankalpa Sethi, Arnab Dutta, Partha Pratim Jana <ppj@chem.iitkgp.ac.in> *from* Indian Institute of Technology, Kharagpur, India

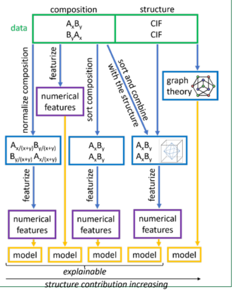
1. Department of Applied Physics and Applied Mathematics, Columbia University, NY, New York 10027
2. Ph.D. Program in Chemistry, The Graduate Center of the City University of New York, New York, NY 10016
3. Department of Chemistry, Hunter College, City University of New York , NY, New York 10065
4. School of Physics, Indian Institute of Science Education and Research Thiruvananthapuram, Vithura, Thiruvananthapuram, Kerala 695551, India
5. Department of Chemistry, University of Waterloo, 200 University Ave W, Waterloo, ON, Canada

Motivation

Abstract: Compositional features are commonly used in traditional ML models for solid-state materials informatics. We combine both compositional and structural features with minimal programming expertise required. Our approach utilizes open-source, interactive Python programs named Composition Analyzer Featurizer (CAF) and Structure Analyzer Featurizer (SAF). CAF generates numerical compositional features from a list of formulas provided in an Excel file, while SAF extracts numerical structural features from a .cif file.

134 features from CAF and 64 features from SAF were used either individually or in combination to cluster ten crystal structure in equiatomic AB intermetallics. The result was compared against that of JARVIS, MAGPIE, mat2vec, and OLED.

We combined compositional (formula) and structural (.cif) features for clustering crystal structures of equiatomic AB intermetallics

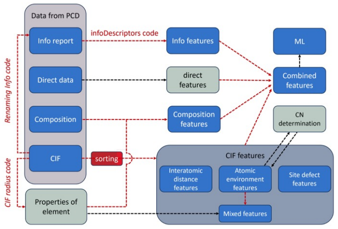


Structure/Composition Featurizers

We developed Python tools to extract features from .xlsx and .cif files

1. Structure Analyzer Featurizer (SAF)

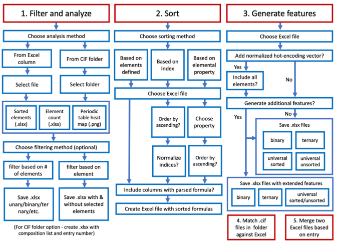
SAF parses .cif, generates supercell, extracts geometric features of interatomic, atomic environment, and coordination numbers - 94 features
for binary, 135 for ternary



GitHub
Available here:
github.com/bobleesj/
structure-analyzer-featurizer

2. Composition Analyzer Featurizer (CAF)

CAF reads .xlsx file with a column containing a formula for each row, generates features using the OLED property list[1]


1. Filter and analyze
Choose analysis method
From Excel column
From CIF folder
Select file
Select folder
Sorted elements (.xlsx)
Element count (.xlsx)
Periodic table heat map (.png)
Choose filtering method (optional)
filter based on # of elements
filter based on element
Save .xlsx unary/binary/ternary/etc.
Save .xlsx with & without selected elements
(For CIF folder option - create .xlsx with composition list and entry number)
2. Sort
Choose sorting method
Based on elements defined
Based on Index
Based on elemental property
Choose Excel file
Order by ascending?
Choose property
Normalize indices?
Order by ascending?
Include columns with parsed formula?
Create Excel file with sorted formulas
3. Generate features
Choose Excel file
Add normalized hot-encoding vector?
Yes
No
Include all elements?
Generate additional features?
Yes
No
Save .xlsx files
binary
ternary
universal sorted
universal unsorted
Save .xlsx files with extended features
binary
ternary
universal sorted/unsorted
4. Match .cif files in folder against Excel
5. Merge two Excel files based on entry

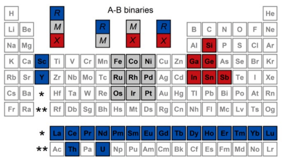
[1] *Data in Brief,* **53** (2024) 110178

GitHub
Available here:
github.com/bobleesj/
composition-analyzer-featurizer

Crystal Structure Prediction

We used 134 features from CAF and 64 features from SAF to classify 10 crystal structures in equiatomic *AB* intermetallics.

Selection of A-B elements



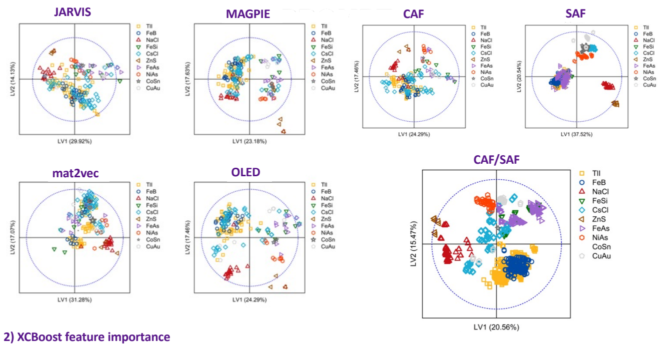
Number of files per structure from PCD

Structure type	Label	Search result	Ambient conditions
TlI	1	411	401
FeB	2	279	197
NaCl	3	243	236
FeSi	4	190	164
CsCl	5	188	138
ZnS	6	89	89
FeAs	7	86	79
NiAs	8	85	83
CoSn	9	51	41
CuAu	10	47	41
Cu	N/A	141	104
Mg	N/A	32	29

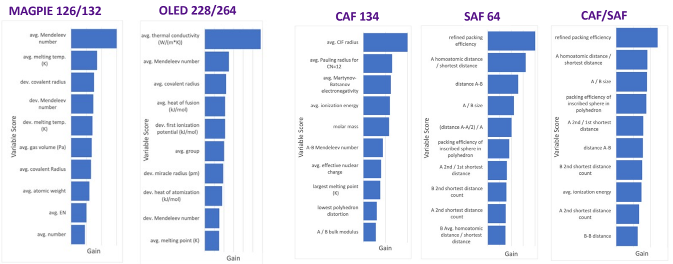
Result

	features		PLS-DA			SVM		
	used	generated	sensitivity	specificity	error	sensitivity	specificity	error
JARVIS	1758	2625	0.920	0.790	0.145	0.995	0.996	0.004
MAGPIE	126	132	0.898	0.799	0.151	0.995	0.996	0.004
mat2vec	492	1194	0.900	0.991	0.055	0.900	1.000	0.050
OLED	228	264	0.898	0.812	0.145	0.998	0.998	0.002
CAF	134	134	0.948	0.667	0.193	0.995	0.996	0.004
SAF	64	64	0.970	0.814	0.108	0.994	0.998	0.004
CAF+SAF	198	198	0.961	0.815	0.112	0.999	1.000	0.001

1) PLS-DA



2) XCBoost feature importance



Conclusion: *best performance achieved by combining SAF and CAF, with structural features from both appearing in top ten features*

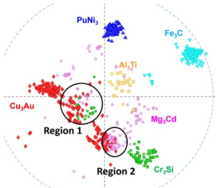
Experimental Validation with Unsupervised ML

Contributors: Nishant Yadav, Siddha Sankalpa Sethi, Arnab Dutta, Partha Pratim Jana

We used unsupervised ML for structure type prediction for 1:3 intermetallic structure types

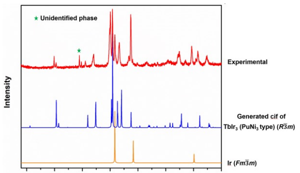
Motivation

- Supervised learning maps compositional and geometric features to structure type
- Unsupervised learning remains challenging to differentiate between similar structures
- Extracted 2366 reports (PCD) with 1:3 stoichiometry. 97 features generated with CAF+SAF were used with K-means method to find 10 clusters of the most common 1:3 intermetallic structure types



Experimental validation

- Synthesized novel intermetallic $TbIr_3$ suggested to be $PuNi_3$-type
- X-ray powder diffraction confirmed the predicted structure type, with only few problematic regions.



Interactive Code for Unsupervised ML

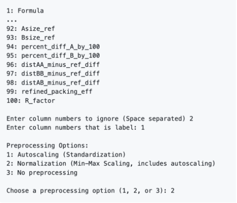
Contributors : Anirudh Machathi

We are developing an interactive code for non-programmers to apply unsupervised learning and visualize trends

Step 1. Prepare Excel containing features per formula (Use CAF/SAF)

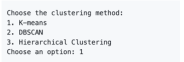
Formula	Molar mass	AB distance	...	A CIF radius	B CIF radius
Nd0.25Sn0.75	262.952	3.326	...	1.657	1.489
Ce0.25Pd0.75	246.535	2.907	...	1.723	1.376
Y0.25Pb0.75	296.105	3.406	...	1.681	1.725

Step 2. Read headers and preprocess



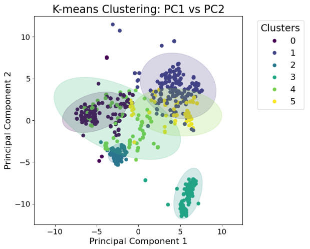
```
1: Formula
...
92: Asize_ref
93: Bsize_ref
94: percent_diff_A_by_100
95: percent_diff_B_by_100
96: distAA_minus_ref_diff
97: distBB_minus_ref_diff
98: distAB_minus_ref_diff
99: refined_packing_eff
100: R_factor

Enter column numbers to ignore (Space separated) 2
Enter column numbers that is label: 1

Preprocessing Options:
1: Autoscaling (Standardization)
2: Normalization (Min-Max Scaling, includes autoscaling)
3: No preprocessing

Choose a preprocessing option (1, 2, or 3): 2
```

Step 3. Choose clustering method



```
Choose the clustering method:
1. K-means
2. DBSCAN
3. Hierarchical Clustering
Choose an option: 1
```

Example output



K-means method (Elbow method clustering)

Available here (beta):
https://github.com/AnirudhM2110/pythonProject