Molecular Point Group prediction of coordination compounds using ML

7th School of Computational Chemistry, AI & ML

J. A. Alvarado-Huayhuaz, M. A. Santos Silva, M. Balboni, K. dos Santos Machado, A. C. Valderrama Negrón

Universidad Nacional de Ingeniería
Lima, Perú

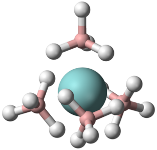
Outline

1. Introduction
2. Methodology
3. Results and discussion
4. Conclusions
5. Acknowledgements

Metals in biological systems

1. Our interest in the prediction of groups of points is based on the study of iron-siderophores stereoisomerism, therefore, we focus on metals in biological systems.
2. In biological systems, metal ions are always coordinated by water, biomolecules, or others called ligands.
3. The metal-ligand interaction depends on the chemical nature of both.
4. Pearson explained this association in terms of “Hard and Soft Acids and Bases”.[1]

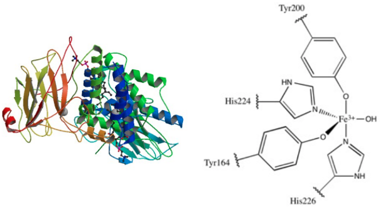
$$\boxed{\text{(n-1)d}^{x} ns^{2} \rightarrow (n-1)d^{x-i}}$$



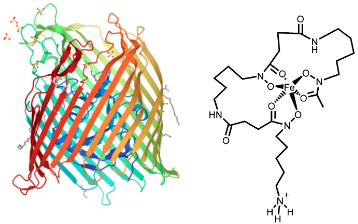
[1] J. Am. Chem. Soc. 1963, 85, 22, 3533–3539

Coordinated Number

1. The coordination number of a central atom in a molecule or crystal is the number of atoms, molecules or ions bonded to it.
2. Some examples of geometry in proteins with iron: tetrahedron (rubredoxin), trigonal bipyramide (catecholate dioxygenase), pyramid (tyrosine hydroxylase), octahedron (lipo-oxygenase)[2]:



Catecholate dioxygenase
(trigonal bipyramid)



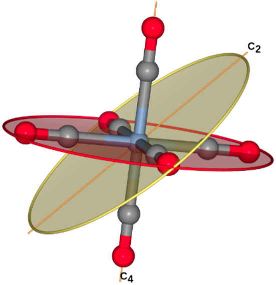
FhuA - Ferrioxamine
(octahedral)

[2] Toma HE. 2015. Química bioinorgánica e ambiental. Blucher Ltda, São Paulo, p. 268.

Group of Point

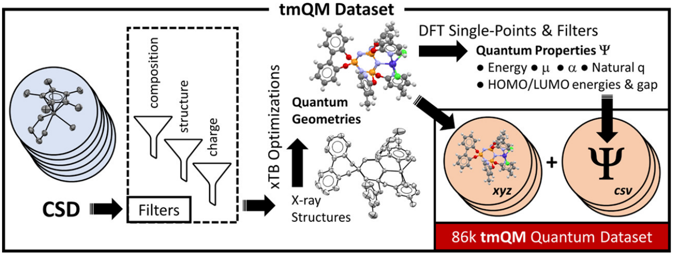
1. The structural organization around the metal can be classified according to the Theory of Groups, by a "Group of point".
2. These can be determined by symmetry operations: identity, proper reflection, improper reflection, rotation, and inversion[3].
3. Applications: chirality, hybridization, analysis of molecular vibrations, activity in infrared and Raman spectroscopy, etc.

O_h reflection (red)
O_d reflection (yellow)
C_2, C_4 rotations
$[Cr(CO)_6]$



[3] https://symotter.org/gallery

Datasets

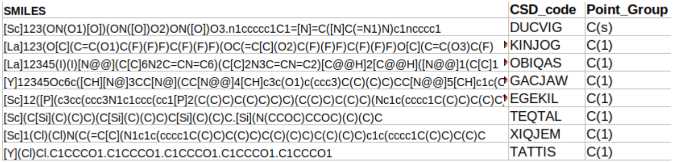


tmQM: 7 filters

- All structures contain a single TM[a]
- Minimum of one C and one H atoms. Allowed: B, Si, N, P, etc.
- Excluded: counterions, polymeric structures, all structures without three-dimensional coordinates, disordered atoms, charge higher than 1 and lower than 1.

[a] J. Chem. Inf. Model. 2020, 60, 12, 6135–6146

Datasets



SMILES	CSD_code	Point_Group
[Sc]123(ON(O1)[O])(ON([O])O2)ON([O])O3.n1ccccc1C1=[N]=C([N]C(=N1)N)c1ncccc1	DUCVIG	C(s)
[La]123(O[C](C=C(O1)C(F)(F)F)C(F)(F)F)(OC(=C[C](O2)C(F)(F)F)C(F)(F)F)O[C](C=C(O3)C(F)	KINJOG	C(1)
[La]12345(I)(I)[N@@](C[C]6N2C=CN=C6)(C[C]2N3C=CN=C2)[C@@H]2[C@@H]([N@@]1(C[C]1	OBIQAS	C(1)
[Y]12345Oc6c([CH][N@]3CC[N@](CC[N@@]4[CH]c3c(O1)c(ccc3)C(C)(C)C)CC[N@@]5[CH]c1c(C	GACJAW	C(1)
[Sc]12([P](c3cc(ccc3N1c1ccc(cc1[P]2(C(C)C)C(C)C)C)(C(C)C)C(C)C)(Nc1c(cccc1C(C)C)C(C)C	EGEKIL	C(1)
[Sc](C[Si](C)(C)C)(C[Si](C)(C)C)C[Si](C)(C)C.[Si](N(CCOC)CCOC)(C)(C)C	TEQTAL	C(1)
[Sc]1(Cl)(Cl)N(C(=C[C](N1c1c(cccc1C(C)C)C(C)C)C(C)(C)C)C(C)(C)C)c1c(cccc1C(C)C)C(C)C	XIQJEM	C(1)
[Y](Cl)Cl.C1CCCO1.C1CCCO1.C1CCCO1.C1CCCO1.C1CCCO1	TATTIS	C(1)

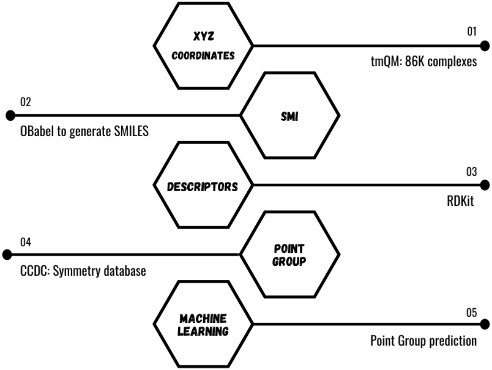
CSDSymmetry

- Provides an extremely flexible source of symmetry related information: molecular point group, space group, Z, Z' and the symmetry of the occupied Wyckoff position for molecules in the CSD.
- Intersected with chemical or substructural searches performed in ConQuest.
- CSDSymmetry is a relational database built using Microsoft Access (2007) and is available as a free download (terms and conditions apply)[a].

[a] https://www.ccdc.cam.ac.uk/community/csd-community/csdsymmetry/

Summary

Available from Google Colab[4].



XYZ COORDINATES

01

tmQM: 86K complexes

SMI

02

OBabel to generate SMILES

DESCRIPTORS

03

RDKit

POINT GROUP

04

CCDC: Symmetry database

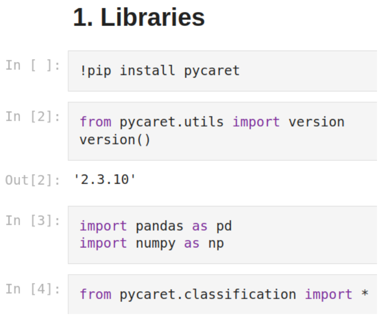
MACHINE LEARNING

05

Point Group prediction

[4] https://github.com/inefable12/OhTd_PointGroup_qmTM_CCDC

Libraries and Data



1. Libraries

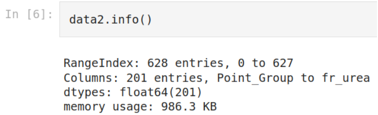
```
In [ ]:  !pip install pycaret

In [2]:  from pycaret.utils import version
         version()

Out[2]:  '2.3.10'

In [3]:  import pandas as pd
         import numpy as np

In [4]:  from pycaret.classification import *
```



```
In [6]:  data2.info()

RangeIndex: 628 entries, 0 to 627
Columns: 201 entries, Point_Group to fr_urea
dtypes: float64(201)
memory usage: 986.3 KB
```

Pre-processing

Delete columns with entropy 0

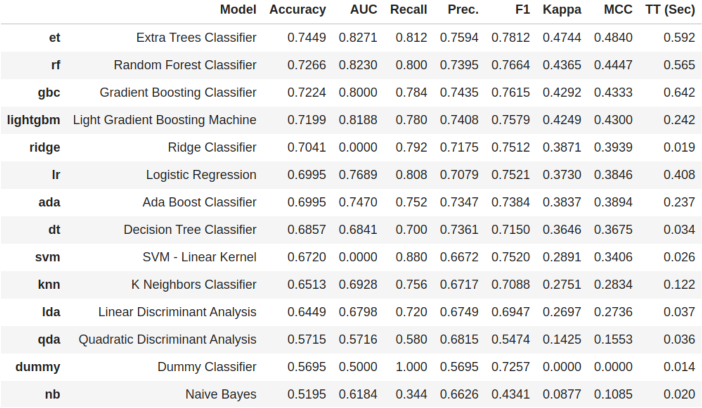
```
for column in df:
  min = df[column].min()
  max = df[column].max()
  if max == min:
    del(df[column])
    continue
```

Normalization

```
for column in df:
  if column != "Value":
    min = df[column].min()
    max = df[column].max()
    for i in range(len(df[column])):
      df[column][i] = (df[column][i]-min)/(max-min)
```

PyCaret

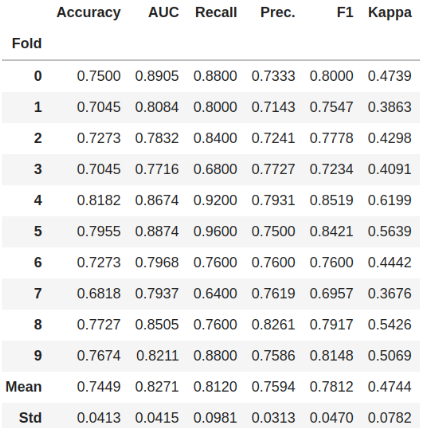
- Setup(df, target = ’Point_Group’, session_id=123)
- Compare_models()



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.7449	0.8271	0.812	0.7594	0.7812	0.4744	0.4840	0.592
rf	Random Forest Classifier	0.7266	0.8230	0.800	0.7395	0.7664	0.4365	0.4447	0.565
gbc	Gradient Boosting Classifier	0.7224	0.8000	0.784	0.7435	0.7615	0.4292	0.4333	0.642
lightgbm	Light Gradient Boosting Machine	0.7199	0.8188	0.780	0.7408	0.7579	0.4249	0.4300	0.242
ridge	Ridge Classifier	0.7041	0.0000	0.792	0.7175	0.7512	0.3871	0.3939	0.019
lr	Logistic Regression	0.6995	0.7689	0.808	0.7079	0.7521	0.3730	0.3846	0.408
ada	Ada Boost Classifier	0.6995	0.7470	0.752	0.7347	0.7384	0.3837	0.3894	0.237
dt	Decision Tree Classifier	0.6857	0.6841	0.700	0.7361	0.7150	0.3646	0.3675	0.034
svm	SVM - Linear Kernel	0.6720	0.0000	0.880	0.6672	0.7520	0.2891	0.3406	0.026
knn	K Neighbors Classifier	0.6513	0.6928	0.756	0.6717	0.7088	0.2751	0.2834	0.122
lda	Linear Discriminant Analysis	0.6449	0.6798	0.720	0.6749	0.6947	0.2697	0.2736	0.037
qda	Quadratic Discriminant Analysis	0.5715	0.5716	0.580	0.6815	0.5474	0.1425	0.1553	0.036
dummy	Dummy Classifier	0.5695	0.5000	1.000	0.5695	0.7257	0.0000	0.0000	0.014
nb	Naive Bayes	0.5195	0.6184	0.344	0.6626	0.4341	0.0877	0.1085	0.020

Model

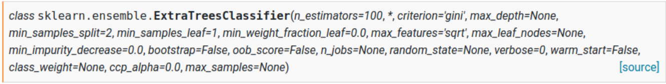
- et = create_model('et')



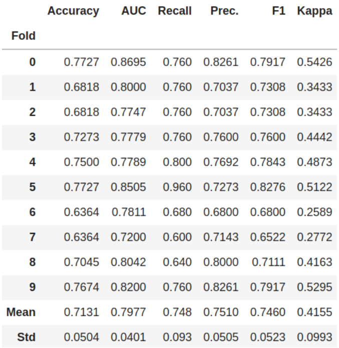
	Accuracy	AUC	Recall	Prec.	F1	Kappa
Fold						
0	0.7500	0.8905	0.8800	0.7333	0.8000	0.4739
1	0.7045	0.8084	0.8000	0.7143	0.7547	0.3863
2	0.7273	0.7832	0.8400	0.7241	0.7778	0.4298
3	0.7045	0.7716	0.6800	0.7727	0.7234	0.4091
4	0.8182	0.8674	0.9200	0.7931	0.8519	0.6199
5	0.7955	0.8874	0.9600	0.7500	0.8421	0.5639
6	0.7273	0.7968	0.7600	0.7600	0.7600	0.4442
7	0.6818	0.7937	0.6400	0.7619	0.6957	0.3676
8	0.7727	0.8505	0.7600	0.8261	0.7917	0.5426
9	0.7674	0.8211	0.8800	0.7586	0.8148	0.5069
Mean	0.7449	0.8271	0.8120	0.7594	0.7812	0.4744
Std	0.0413	0.0415	0.0981	0.0313	0.0470	0.0782

Tuned model

- tuned_et = tune_model(et, optimize = 'AUC')

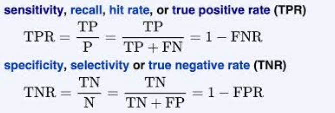


```
class sklearn.ensemble.ExtraTreesClassifier(n_estimators=100, *, criterion='gini', max_depth=None,
min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None,
min_impurity_decrease=0.0, bootstrap=False, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False,
class_weight=None, ccp_alpha=0.0, max_samples=None)                                                              [source]
```



Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa
0	0.7727	0.8695	0.760	0.8261	0.7917	0.5426
1	0.6818	0.8000	0.760	0.7037	0.7308	0.3433
2	0.6818	0.7747	0.760	0.7037	0.7308	0.3433
3	0.7273	0.7779	0.760	0.7600	0.7600	0.4442
4	0.7500	0.7789	0.800	0.7692	0.7843	0.4873
5	0.7727	0.8505	0.960	0.7273	0.8276	0.5122
6	0.6364	0.7811	0.680	0.6800	0.6800	0.2589
7	0.6364	0.7200	0.600	0.7143	0.6522	0.2772
8	0.7045	0.8042	0.640	0.8000	0.7111	0.4163
9	0.7674	0.8200	0.760	0.8261	0.7917	0.5295
Mean	0.7131	0.7977	0.748	0.7510	0.7460	0.4155
Std	0.0504	0.0401	0.093	0.0505	0.0523	0.0993

Receiver Operating Characteristic (ROC)



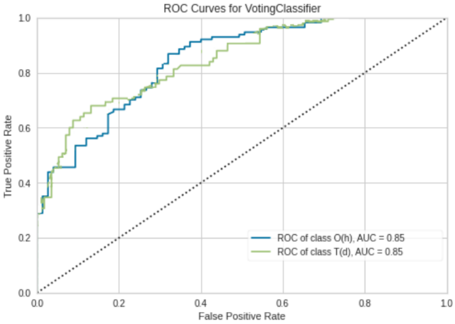
sensitivity, **recall**, **hit rate**, **or true positive rate (TPR)**

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} = 1 - \mathrm{FNR}$$

specificity, **selectivity** **or true negative rate (TNR)**

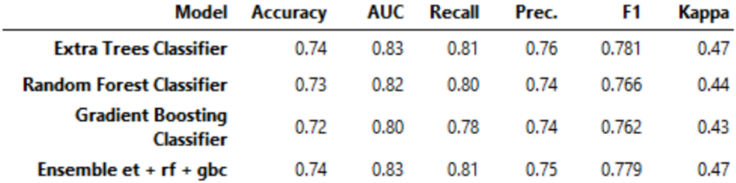
$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} = 1 - \mathrm{FPR}$$

- Area Under Curve (AUC)



et + rf + gbc ?

- PyCaret: Blend_models(estimator_list=[et,rf,gbc])

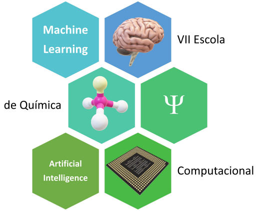


Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
Extra Trees Classifier	0.74	0.83	0.81	0.76	0.781	0.47
Random Forest Classifier	0.73	0.82	0.80	0.74	0.766	0.44
Gradient Boosting Classifier	0.72	0.80	0.78	0.74	0.762	0.43
Ensemble et + rf + gbc	0.74	0.83	0.81	0.75	0.779	0.47

Conclusions

1. The ensemble et, rf and gbc models fit best to predict the Oh and Td point groups of the coordination compounds
2. Perspectives: Using quantum descriptors (global and local reactivity) using PRIMoRDiA (Macromolecular Reactivity Descriptors Access) for a generalized point group classification model.

Acknowledgements



Machine Learning

VII Escola de Química Computacional

Ψ

Artificial Intelligence

Universidade de São Paulo
Instituto de Química

Centro de Ciencias
Computacionales
FURG, Brasil

LIBIPMET

Universidad Nacional de Ingeniería
Scientia et Labor
Lima 1876 Perú

Applied
Computational
Chemistry
Group
Prof. Ataualpa A. C. Braga, Ph.D.

C3 - CENTRO DE CIÊNCIAS COMPUTACIONAIS
COMBI-LAB
UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG

FONDECYT

jalvaradoh@uni.pe