AI in Chem, molecules properties UH 2025

Outline

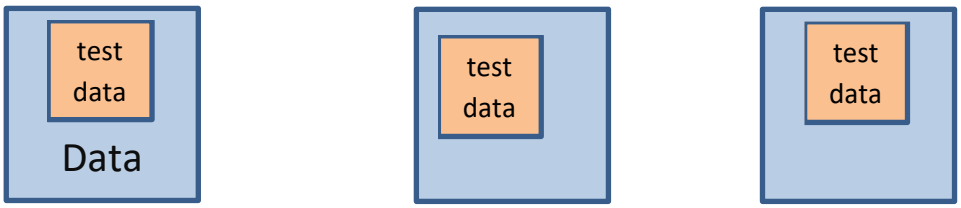
- Descriptors of molecules
- Solubility project
- Synthetic data, or DFT data, validatation
- redox potentials and pKa's
- Chemical reactions
- SHAP analysis

Machine learning classes

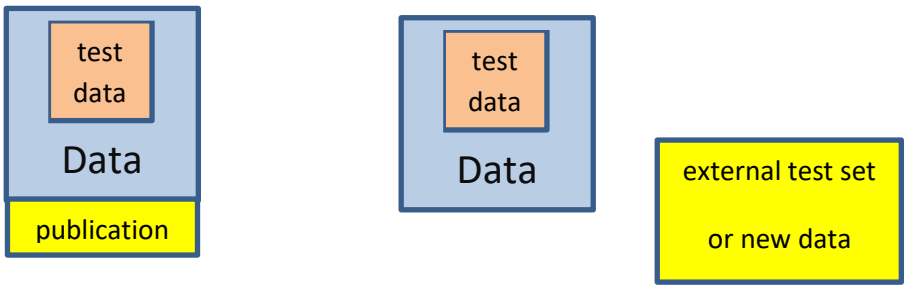
Supervised learning (SL): The aim is to learn known outputs and find good descriptors for the system. This is the most relevant ML for materials science. This is also a relatively easy ML problem.

How large the data set should be? As large as possible, but in chemistry/materials science the data sets are usually not very large. Data size of below 100 is useless since all ML methods rely on statistic. 1000 is OK and larger sets are even better.

Validation: One can make the training/test data partitioning in several times. This approach produces several ML models and test it. In this way, the quality of the ML models can be tested better than on single data partitioning.



One can also leave some data out of the cross-validation data and use that as second level test set or **publication set**. The publication set is never used in training.



Even better, some new and quite different data can be used to test the ML method. In real applications, the ML method needs to work on **new data**. If the external data is "**similar**" the ML method should work, but if the external data is very different from the training data the predictions are probably bad.

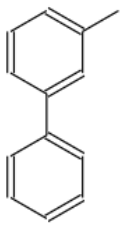
Descriptors of molecules

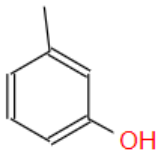
In any system that uses ML, we need to decide how data is presented. The molecules are non-trivial since they are not "binary" like pictures.

Smiles

Smiles is a very useful representation of molecules. It contains only letters and most chemical codes understand and can make SMILES.

Eg. benzene c1ccccc1

Cc2cccc(c1ccccc1)c2 = 

Cc1cccc(O)c1 = 

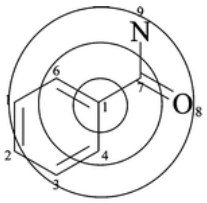
https://www.cheminfo.org/flavor/malaria/Utilities/SMILES_generator___checker/index.html

Try this tool for a few molecules.

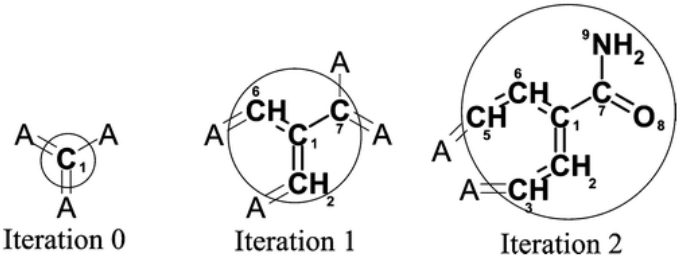
Smiles are not very easy for humans to read but from them one can compute several properties (The RDKit tool is very useful). The plain SMILES are not very useful for clustering or ML.

In chemical problems, we need some molecular descriptors. There are many possibilities, atomic numbers, atom-atom distances (SOAP), Coulomb matrix, HOMO, LUMO, dipole moments, atomic charges, etc. Some of these are structure or atomic **position based** and some require **information of the electrons** (needs quantum chemical calculations).

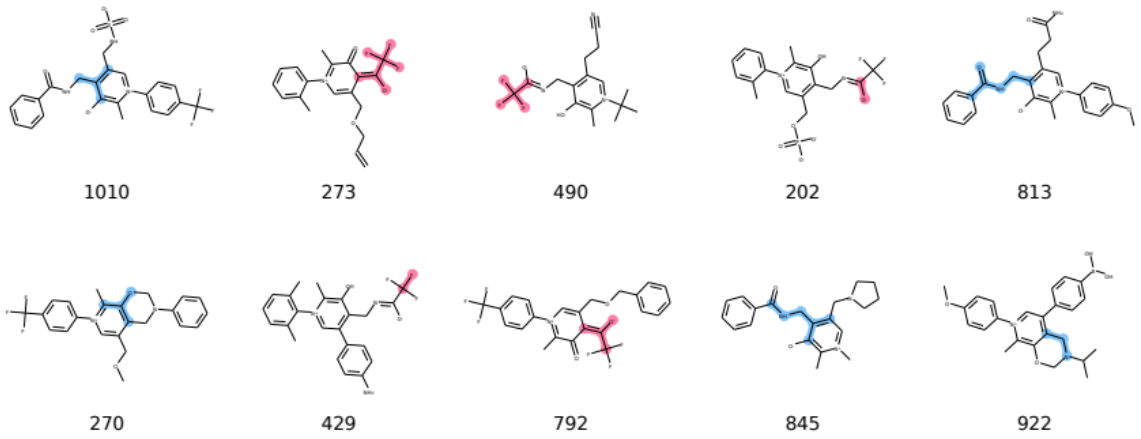
One chemically intuitive descriptor is **fingerprint**. One of them is the Extended-connectivity fingerprint (ECFP). It is a systematic tool that list atoms environment in molecules. (Ref: Rogers and Hahn, *J. Chem. Inf. Model.* 2010, 50, 5, 742-754). The 0 level is the atom itself, the level 1 contains the atoms neighbors and so on. The fingerprints are related to chemical groups, like OH, CH3, COOH, C-C, C=C, etc. Fingerprints are binary and this is a bit odd. If the same chemical group is several times in a molecule, it is counted only once. We tried weighted FP's where the number of the chemical groups is stored to the FP vectors. It seems to work (a bit) better.



Considering atom 1 in benzoic acid amide



Typically one has to limit the number of the different ECFP's of all the studied molecules. There are quite a few of them but surprisingly few. We did a project in which there were 7000 different molecules and we found 1024 ECFP of level 4 (same as iteration 2 above)



Unfortunately, the fingerprints are indexed with numbers, which is not intuitive, but they can be used as descriptors. We have studied several projects and used fingerprints as molecule descriptors.

They are easy to make, for example with RDKit program.

```
mol = Chem.MolFromSmiles(smiles[i])

fp = AllChem.GetMorganFingerprintAsBitVect(mol, int(0.5*diameter),
nBits=1024, bitInfo=bitinfo)

fingerprints[i] = np.array(fp)
```

Other common structural descriptors:

Coulomb matrix:

Is a descriptor that takes the positions and charges into account.

$$C_{ij} = \begin{cases} 0.5Z_i^{2.4} & i = j \\ \frac{Z_i Z_j}{|R_i - R_j|} & i \neq j \end{cases}$$

Where Z is the atomic number and R is atomic position. This is invariant to translation and rotation BUT it depends on atomic order. Same molecule have several C-matrixes. There is a python program to do the C-matrix (DScribe) (pip install dscribe). Coulomb matrix is not made for periodic systems. One can use Ewald sum matrix (in the DScribe).

```
from dscribe.descriptors import CoulombMatrix
cmw = CoulombMatrix(n_atoms_max=len(water))
```

```
cm_w = cmw.create(water)
cm_w = cm_w.reshape(len(water), len(water))

print(cm_w)
[[73.51669472  8.25964169  8.25964169]
 [ 8.25964169  0.5         0.65510279]
 [ 8.25964169  0.65510279  0.5 ]]
```

The C-matrix is quite large, its size is N^2, and for 36 atoms molecule it has 666 components (C-matric is symmetric).

SOAP

Smooth Overlap of Atomic Positions (SOAP) is a descriptor that encodes regions of atomic geometries by using a local expansion of a gaussian smeared atomic density with orthonormal functions based on spherical harmonics and radial basis functions. SOAP can be made for periodic systems.

Solubility project

We can next look at a solvation project. We have developed a ML project to predict molecules solubility. The experimental data have been taken from **SOMAS database**, which contains the molecules SMILES, solubility, temperature, and descriptor D1 of 11696 molecules

In this project, several very different descriptors were used. The descriptors go beyond molecule's structure.

10.26434/chemrxiv-2025-4111w

D1: include various chemical properties: molecular mass, solvation energy, dipole and quadrupole moments, molecular volume, surface area, highest occupied molecular orbital (HOMO) energy, lowest unoccupied molecular orbital (LUMO) energy, the HOMO–LUMO energy gap and temperature. (12 descriptors)

The electronic structure calculations were performed using DFT with COSMO solvation model. (NWChem software, same information can be obtained with any QC software, like Orca, Gaussian etc.)

D2: We would like to use SMILES as input for ML applications. We can use the RDKit for computing a list of numeric values representing **208 physicochemical properties**, such as the octanol-water partition

coefficient (log P), van der Waals surface area (LabuteASA), and number of rings. In additions the temperature is included. (205 descriptors)

D3: We can use python code Jazzy (input SMILES) to compute Gibbs free energy of hydration, (dgtot), and H-bond strength (sdc, sdx, and sa represent the strengths of the C–H donor, X–H donor (where X denotes non-carbon atoms), and acceptor, respectively.) From Jazzy we got also partial charge and polarizability, from them we can compute a potential and polarization variable V_q and V_a. The temperature is also included. (9 descriptors)

D4: Weighted Fingerprint (ECFP, 1024 bits) + temperature. We assign a weight to each bit corresponding to the number of occurrences. (1025 descriptors)

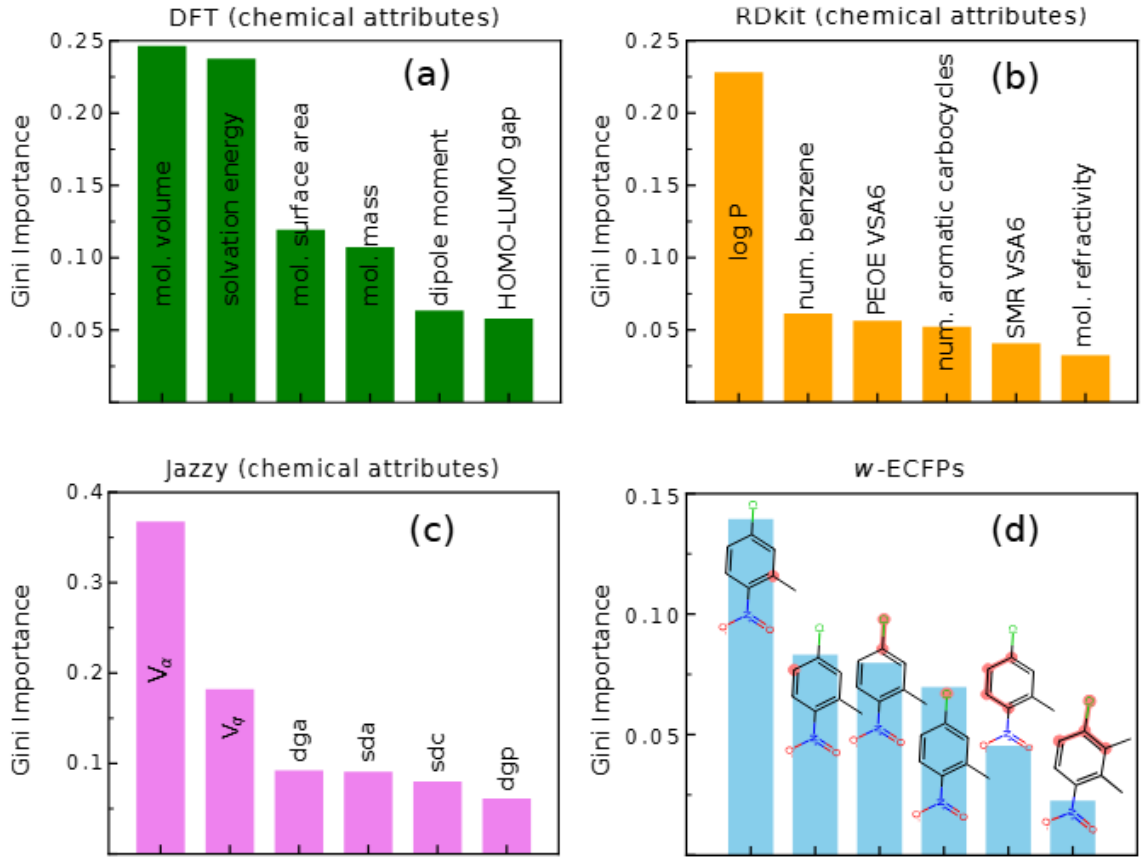
D5: Normal **Fingerprint** + temperature. (1025 descriptors)

Then we did Random Forest ML prediction for 4635 aromatic molecules from the SOMAS DB. Interestingly, descriptors D1-D4 all gave similar accuracy. They use very different descriptions of the molecules and the descriptor size varies a lot, from 9 to 1025. Normal fingerprint D5 is not that good.

In the table below N^{est} is the number of Random Forest trees. The more trees there are, the more difficult the learning is. Here it looks like with a few descriptors the learning is harder. (Note: these methods are very likely faster.)

MAE (log)	RMSE (log)	R^2	N^{est}	f^{max}	NOF	Descriptor	Package
1.77 ± 0.06	2.44 ± 0.09	0.72 ± 0.02	1880	0.42	12	D_1: chemical attributes	NWChem (DFT)
1.50 ± 0.05	2.17 ± 0.10	0.78 ± 0.02	600	0.23	205	D_2: chemical attributes	RDkit (SMILES)
1.78 ± 0.06	2.53 ± 0.12	0.72 ± 0.03	2250	0.45	9	D_3: chemical attributes	Jazzy (SMILES)
1.73 ± 0.07	2.41 ± 0.10	0.73 ± 0.02	950	0.25	1025	D_4: $w-$ECFP	RDkit (SMILES)
2.10 ± 0.07	2.83 ± 0.11	0.62 ± 0.02	425	0.40	1025	D_5: ECFP	RDkit (SMILES)
1.55 ± 0.06	2.24 ± 0.10	0.77 ± 0.03	1500	0.31	24	Six Key Features (6KFs)	

One important feature of ML methods is the descriptors importance analysis. Most of the ML methods can find the most important descriptors. This is useful if we want to reduce the number of descriptors. Typically, the most important descriptors works best. One has to try how many descriptors are needed. Usually the results with fewer descriptors are not as good as the full set, but the difference is not large. This can be useful for very large dataset.



The Six Key Features are these 24 descriptors.

In this study, we also did some outlier removal. The results improved but all descriptors D1-D4 worked similarly. The N^{est} also reduced, so the learning become easier.

MAE (log)	RMSE (log)	R^2	N^{est}	f^{max}	Descriptor
1.36 ± 0.03	1.76 ± 0.04	0.81 ± 0.01	850	0.52	D_1
1.24 ± 0.03	1.62 ± 0.04	0.85 ± 0.02	950	0.49	D_2
1.25 ± 0.04	1.63 ± 0.05	0.84 ± 0.01	1500	0.46	D_3
1.29 ± 0.03	1.64 ± 0.04	0.83 ± 0.02	950	0.49	D_4
1.07 ± 0.03	1.38 ± 0.05	0.90 ± 0.01	800	0.32	D

The main point here is that very different descriptors can be used and many of them are good. This is an important observation. The molecules can be described in many ways.

I do not have much experience with materials ML, but materials can also be described successfully with very different descriptors. There are more tools, like RDKit that has been developed for molecules, but several

structural descriptors, like Coulomb matrix and SOAP, work for periodic systems. The DFT computations are easy for solid systems. They typically take more time than molecular computations.

I prefer to divide the descriptors into two main classes:

1) **Structure only-based** descriptors. Which typically can be computed from SMILES using programs like RDKit (or Jazzy). This is not very complex (even RDKit is not very intuitive) and rather fast. The descriptors D2-D5 are in this category. Typically, these descriptors can contain electron-like descriptors like partial charges (see Jazzy). These are based on some models, not direct quantum chemical computations.

2) **Electron-based** descriptors. These require some quantum chemistry (QC) computations (usually DFT). With proper codes, like Orca, this is not complex, but it requires the QC code (and some knowledge of its usage and output). With normal size molecules the QC calculations are not time consuming. (One can even use the xTB model which is very fast. See the timings at next Chapter)

Synthetic data, or DFT data

In addition to the experimental databases, the data can be generated with DFT computations (or other computations). In ML/AI this is called **Synthetic Data**. The DFT computations can be done systematically and the data is good quality. The DFT is not perfect and there is bias in it (hopefully not a large one). With modern computers large data sets can be generated (100 000 molecules or more).

The computations have also a lot of information of the systems. In the case of molecules, we have all the atomic positions, but we can also get information on dipole moments, atomic charges, HOMO, LUMO electronic states, information of orbitals, etc.

The synthetic data has several advantages:

- It is systematic. All values have been computed in similar ways.
- We have a lot of information on the molecules
- We can get large data sets

The disadvantages are:

- DFT can differ from real results. The results are biased.
- Molecules are static, no temperature, Free energy is possible to compute (harmonic approximation).
- Information on single molecules or small solid/surface systems. Soft material systems are more difficult (but not impossible).

Automatized DFT computations

To create a 1 milj. (or 100 000) molecule DFT database is quite easy. Of course it takes CPU time and one need to get reasonable atomic positions and manage the computations. One very useful approach is to use the SMILES to create the 3D coordinates. This is easy (if the SMILES exist) with RDKit. The identical DFT computations are easy to automatize.

Example: orca computations (no optimization) for mol3457 (or any other number)

```
!PBE def2-SVP
*xyzfile 0 1 mol3457.xyz
```

The automated submission is easy and the automated data collection is a bit more challenging. The output of any DFT program is standard and any data from the output can be collected systematically. There are some tools, like FireWorks, for doing this more reliably. It is also important to recognize the failed computations.

As an example 3000 molecules (average 28 atoms) TZVP type basis DFT computations took 100 CPUh. With 128 cores, around 1 h wall time. Then 3 milj molecules would take ca. 1 month. The 3000 molecules with SVP basis took 26 h (1 core) and with xTB model 2.2 min. (The structural optimization will increase the total time to 40 min.) With xTB 3 milj molecules (no optimization) would take 1.5 days !

Because the DFT computations or most other computations are fast and systematic they are often used in chemistry and materials science ML projects. The DFT results need to be compared to experiments as well as possible. Typically, there is much less experimental data than DFT (or computational) data.

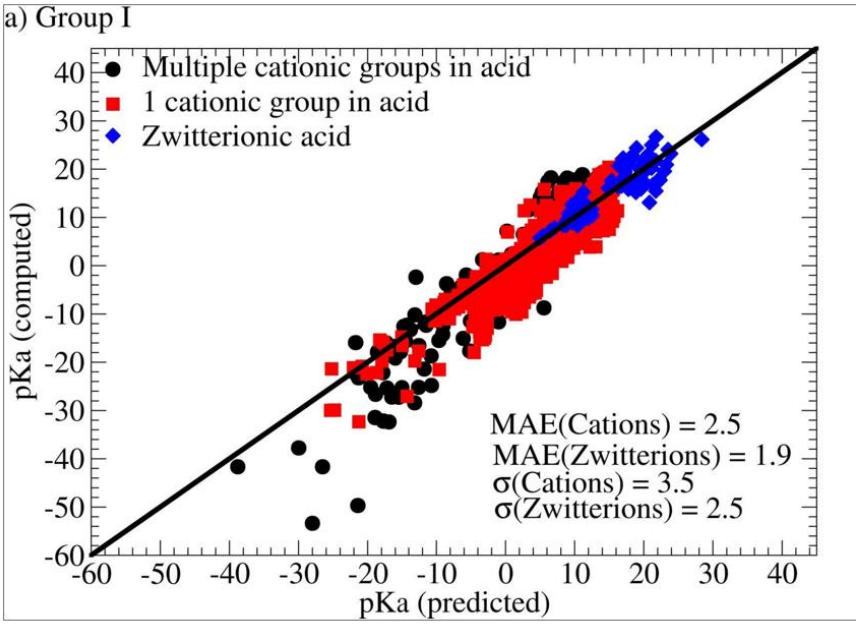
Validation

The validation of the DFT data is important. Here the data sets do not need to be very large. Of course, if there is a lot of

experimental data the better. Below is an example of pKa. The raw DFT data is not very reliable and we used scaling for protic solvents

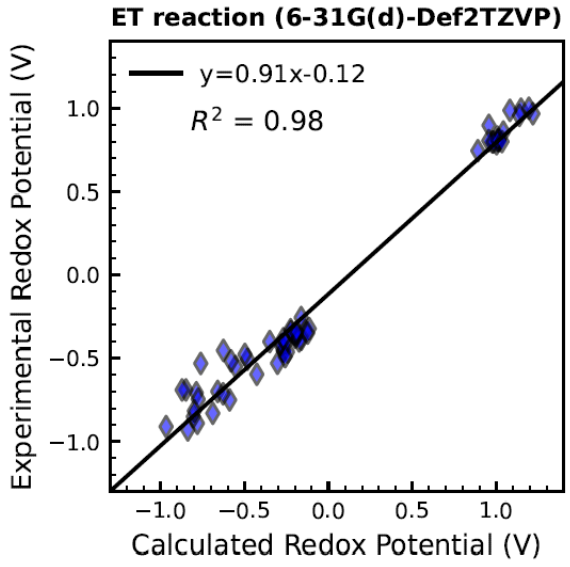
$$pKa(exp) = 0.49\ pKa(DFT) + 3.2$$

This gives a good correlation to experiments



https://doi.org/10.1002/chem.202201667

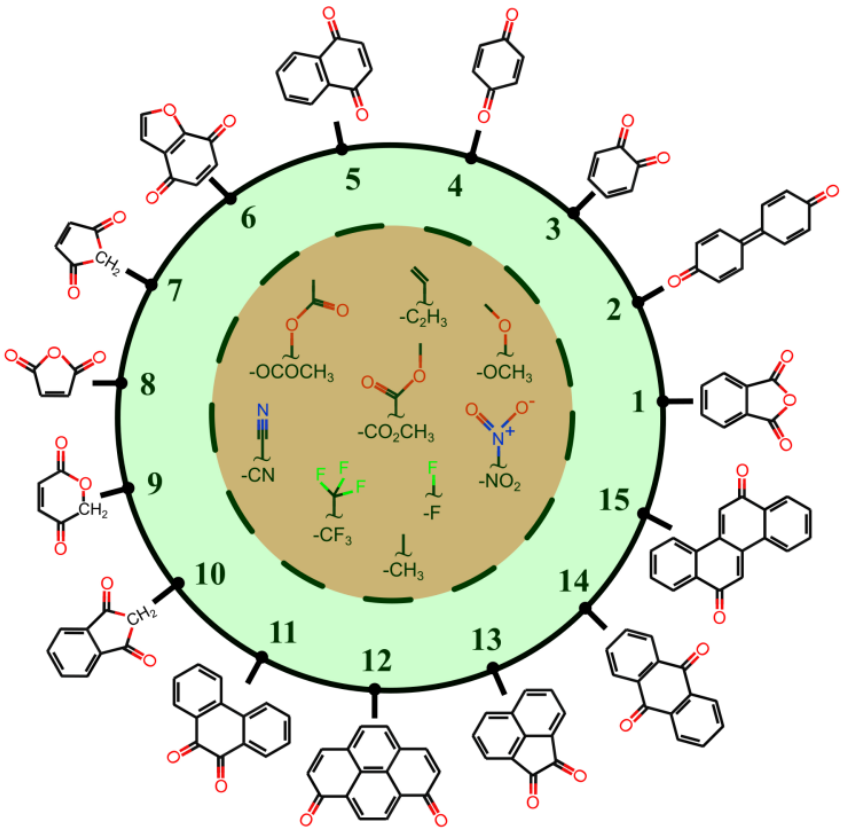
Often the DFT results are very good. Below is an example of redox potential (basically the energy difference of E(A) and E(A-))



https://doi.org/10.26434/chemrxiv-2024-r9qks

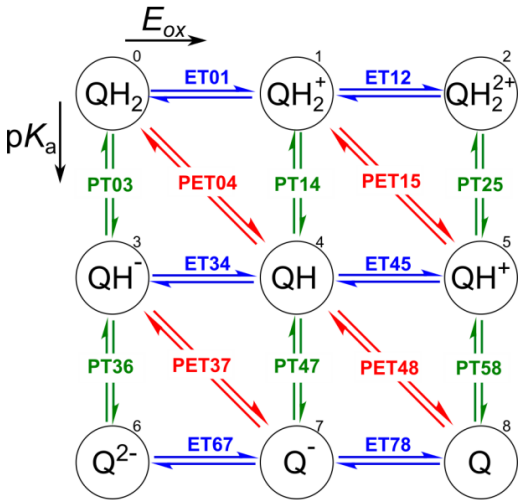
CompBat database for ring molecules (8240 molecules)

We studied a large set of quinone-type molecules and computed their redox potentials and pKa's



Hashemi, A.; Khakpour, R.; Mahdian, A.; Busch, M.; Peljo, P.; Laasonen, K. Density Functional Theory and Machine Learning for Electrochemical Square-Scheme Prediction: An Application to Quinone-type Molecules Relevant to Redox Flow Batteries, Digital Discovery (2023), 5, 1565-1576.

We are interested in the redox potentials and pKa's for 2 H+ and 2 e- reactions. This will lead to the Square-Scheme.



As above we used several descriptors

I: HOMO, LUMO, HOMO-1, LUMO+2, total charge, no. of atoms, weight and volume (solvation vol)

II: HOMO, LUMO, HOMO-1, LUMO+2

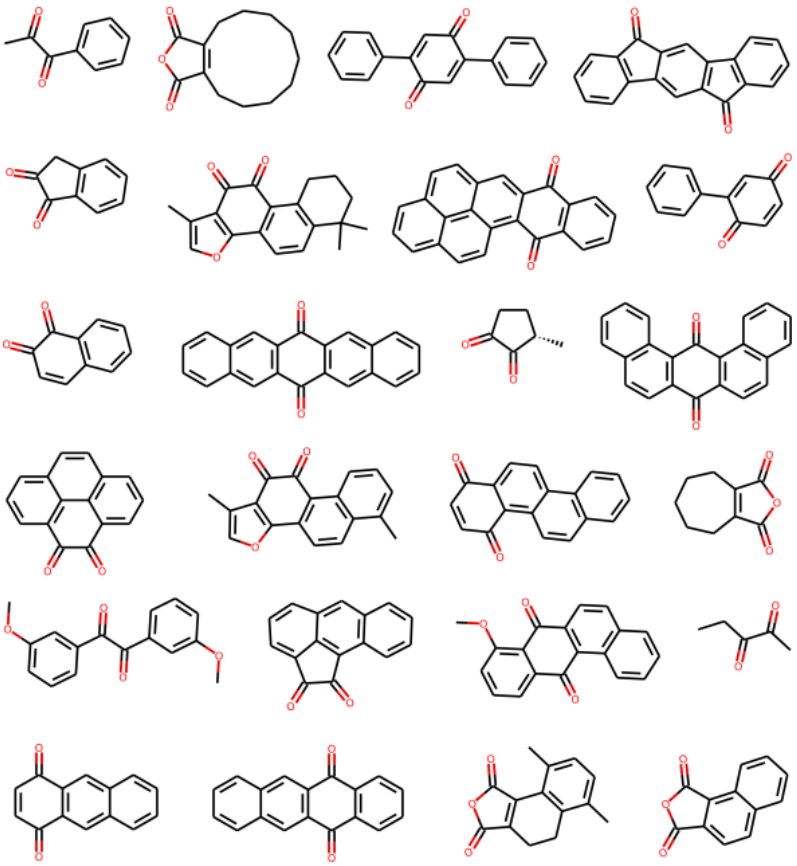
III: HOMO

IV: Fingerprints, total charge (only structure)

Errors with RandomForest

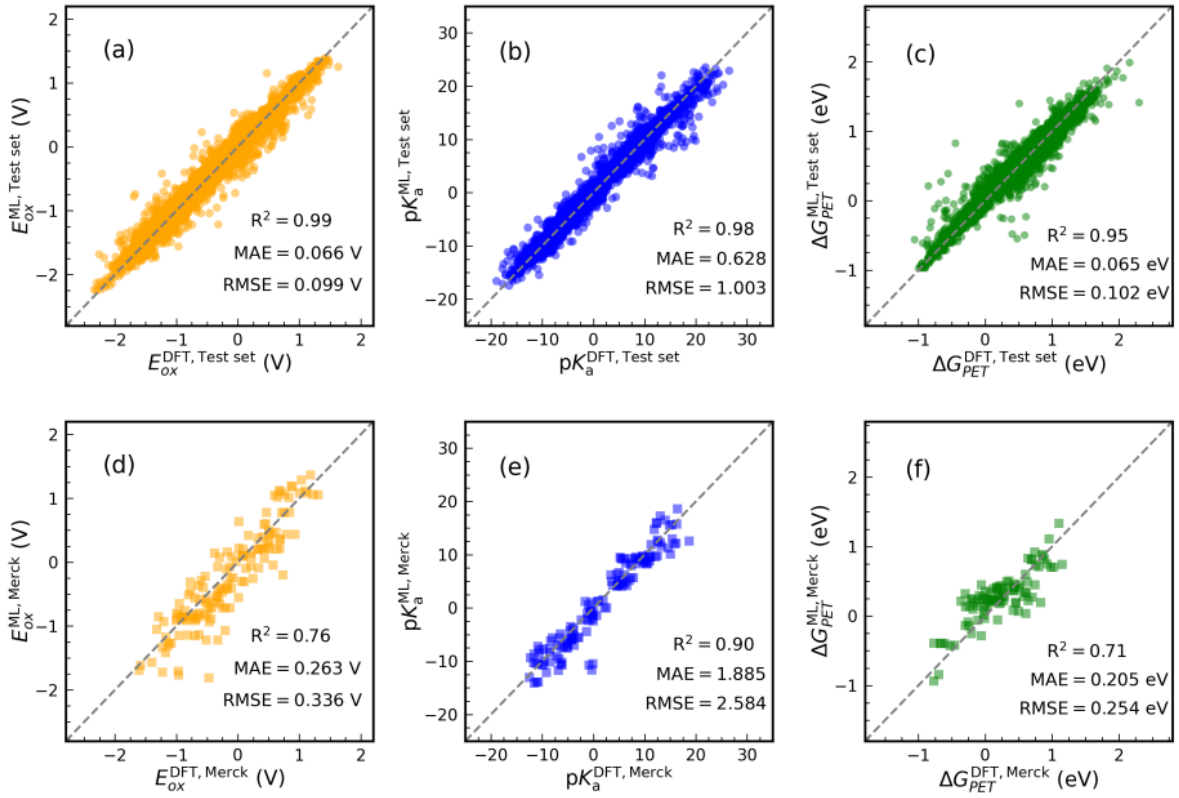
Model	Descriptor	Target	RMSE	MAE	R^2_{trn}	R^2_{tst}	R^2_{oob}
1	I	E_{ox}	0.093	0.062	1.00	0.98	0.98
2	II	E_{ox}	0.102	0.068	0.99	0.97	0.97
3	III	E_{ox}	0.229	0.177	0.92	0.87	0.87
4	I	$\mathrm{p}K_\mathrm{a}$	1.203	0.759	0.99	0.97	0.97
5	II	$\mathrm{p}K_\mathrm{a}$	1.477	0.929	0.99	0.96	0.96
6	I	ΔG_{PET}	0.106	0.073	0.99	0.94	0.94
7	II	ΔG_{PET}	0.127	0.084	0.98	0.92	0.92
8	III	ΔG_{PET}	0.397	0.305	0.53	0.22	0.22

An interesting question is, How different molecules the ML method can predict? We tested this with the following set of molecules:



The results are very good

Descriptor: IV (fingerprint) training a) E_{ox} b) pKa c) ΔG_{PET}
External molecules d) E_{ox} e) pKa f) ΔG_{PET}



Chemical reactions

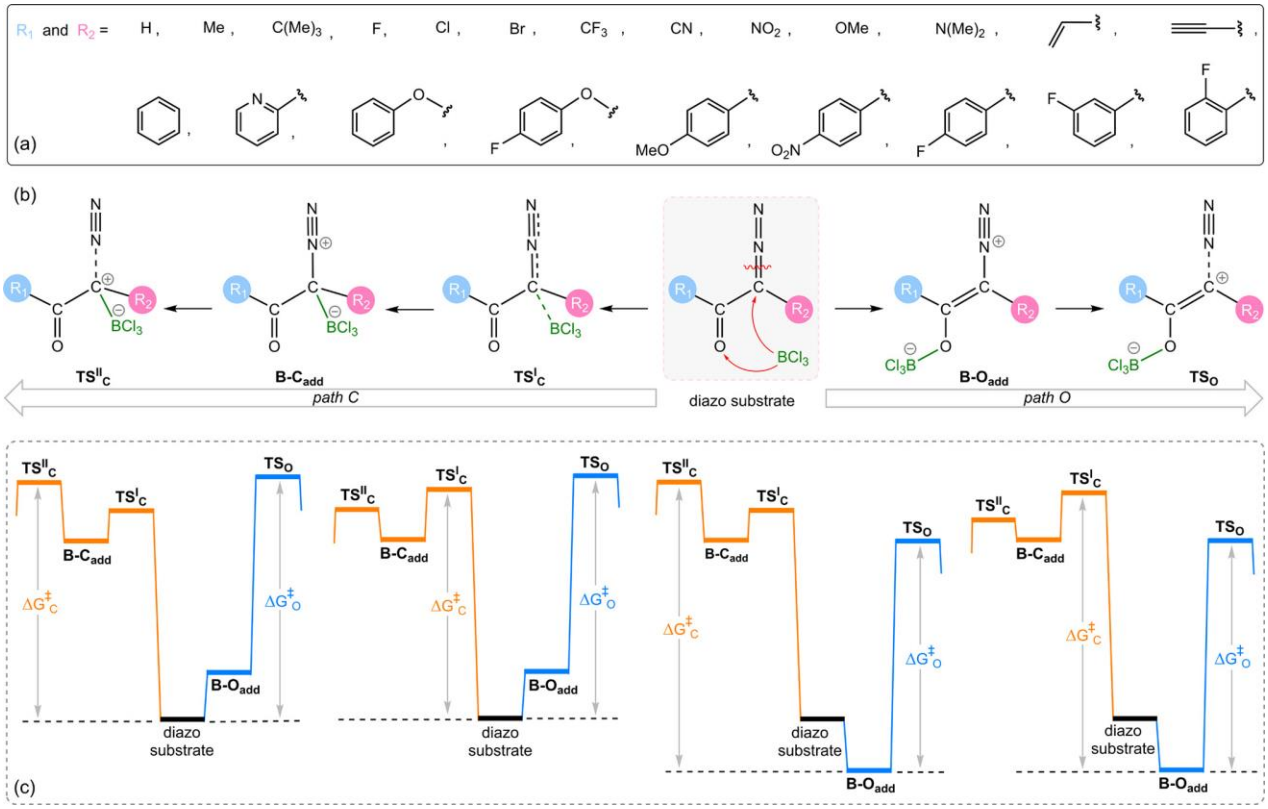
ML can also be used for chemical reactions. A single (or few step) reaction is quite easy. Here, some computations are needed and most of the studies are done with DFT.

Comparison of the Efficiency of B–O and B–C Bond Formation Pathways in Borane-Catalyzed Carbene Transfer Reactions Using α-Diazocarbonyl Precursors: A Combined Density Functional Theory and Machine Learning Study

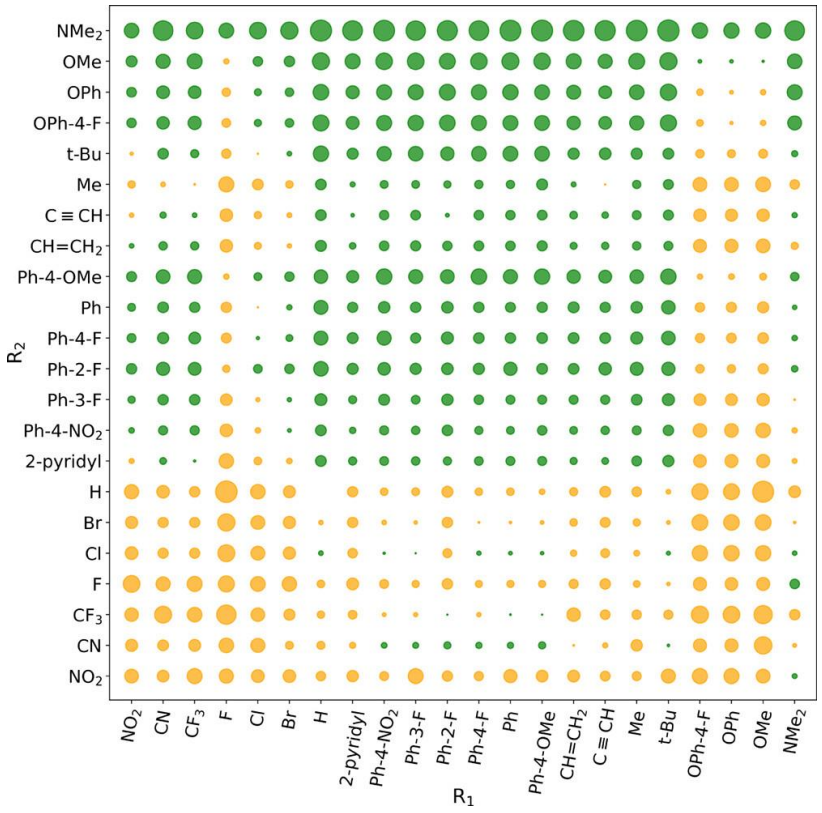
Kaveh Farshadfar and Kari Laasonen

ACS Catal. 2024, 14, 14486–14496

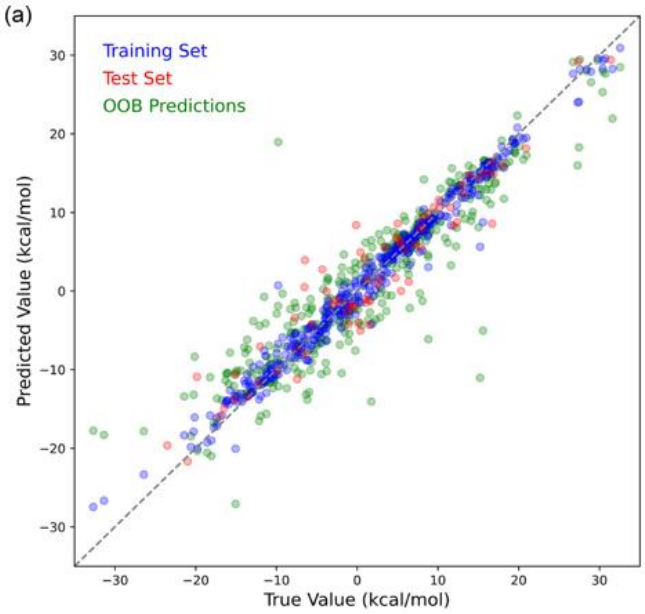
In this work, we studied borane catalysed N2 dissociation reactions. The main interest is which path O or path C are more likely. Before this work the path O is assumed to be the reaction path. We tried 22 different substituents and we can conform that both O- and C- reaction paths are possible.



The green circle indicated the path O and yellow path C. The size of the is related to the difference.



This data can be put to a ML procedure. Here the target value is ΔΔG. A positive value means path O.



As descriptors we used:

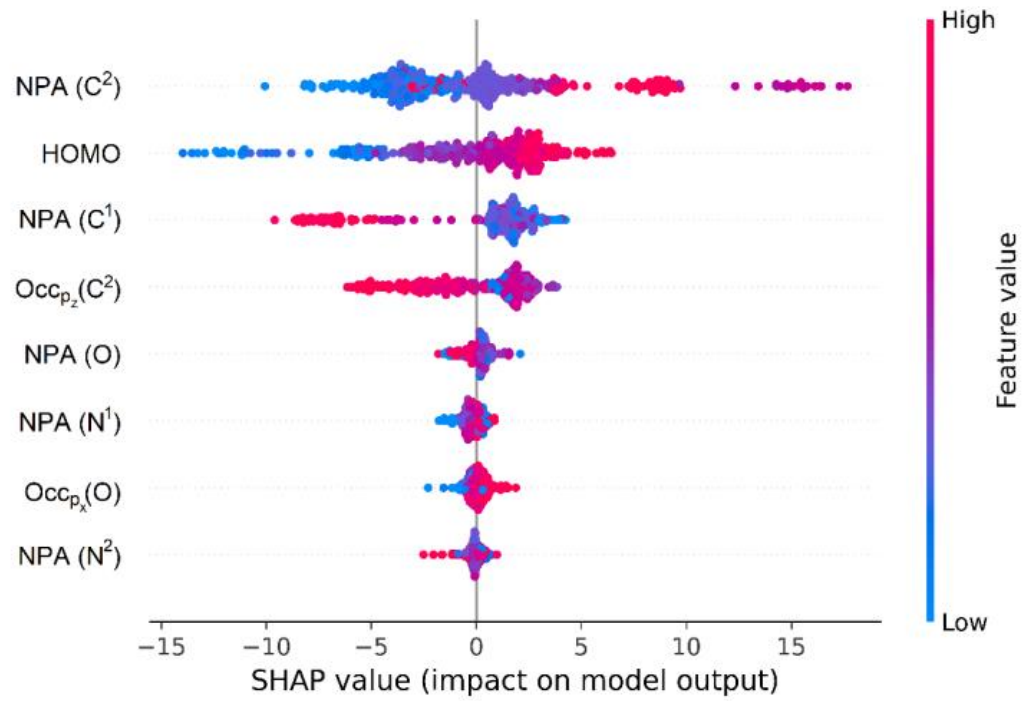
Natural Population Analysis (NPA) charge on the key atoms (C2,C1,O,N1,N2). HOMO energies and occupation of the pz orbital of (C2 and O).

SHAP analysis

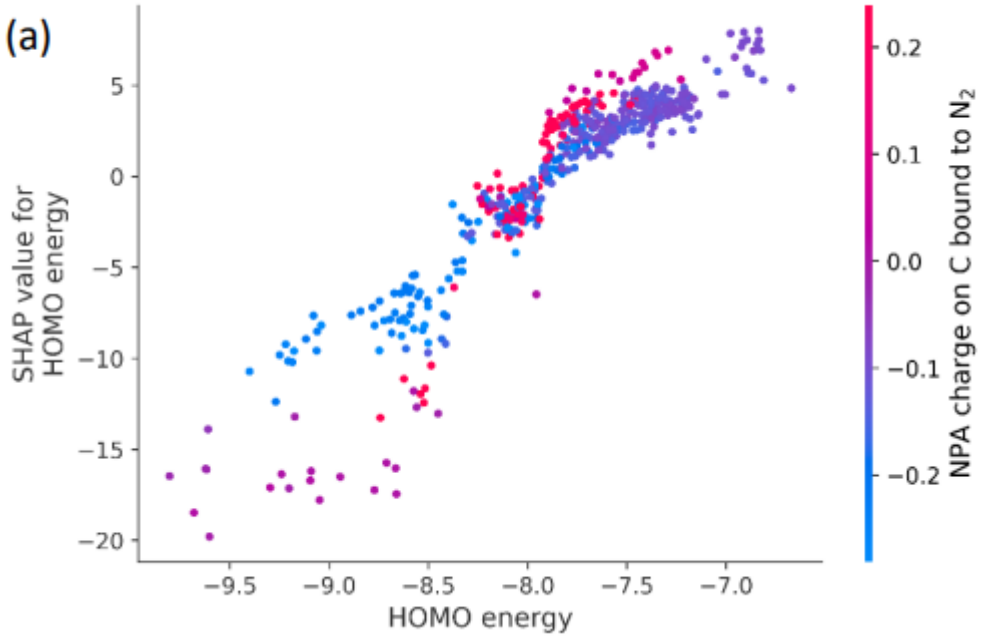
We have used eXplainable AI (SHAP) methods in several projects. This is a very deep way to analyse what the machine has learned. Without it we only get the predictions and often the importance analysis. The SHAP will tell in more detail how each descriptor will contribute to the prediction.

One next page, there is the SHAP analysis to the ΔΔG data. For example low value of the NPA(C2) will indicate negative ΔΔG (path C). Similarly the low HOMO value indicate negative ΔΔG. If these descriptors have high values the positive ΔΔG is more likely.

For example, the NPA(N) values have only a small contribution to the ΔΔG.



The SHAP analysis can also be done in 2D.



The SHAP analysis is particularly useful when we want to UNDERSTAND how the descriptors will affect the target of interest.