GAML

Introduction

This is a document and a conclusion of its supporting information for the Genetic Algorithm Machine Learning program.

In the following, some key words and concepts are explained.

Char: any single character in a single key-stroke.

White-space: it is a collection of any unseeable characters except the paragraph separators. It includes whitespace and tablets.

Type: a collection of different things, which in some way have the same attributes. In here, it’s mainly used to distinguish the data-type, like the float number, integer and the string.

List: a python reserved word, anything bracked with the ‘[ ]’. In here, we also need to consider the list level. For smiliarity ‘[ ]’ is called the the list with one dimension, **1D** list; similarly like ‘[[ ]]’ is called the list with two dimensions, **2D** list and so on.

*__comments__: For bash, anything starts with ‘#’ in the same line
For Python, anything starts with ‘#’ in the same line; anythings between **‘’’ ‘’’** in the multiple lines.
For GROMACS, anything starts with ‘**;**’ in the same line.

For GAML, anything starts with ‘#’ in the same line.

File types and terminologies

top file & itp file

The GROMACS top file is a by rule collection of itp files. The itp file is the smallest integration of topology files. In the same line, any things that starts with the char “ **;** “ (comments) will be ignored.

For itp file:

```
[  atomtypes  ]
;  type    mass     charge  ptype sigma(nm)  epsilon(kJ/mol)
    HW     1.008      0.41    A    0.000000    0.000000
    OW    15.999     -0.82    A    0.316557    0.650194
```

```
 [ moleculetype ]
; molname      nrexcl
    SOL           2


 [ atoms ]
;   nr   type  resnr residue  atom    cgnr     charge     mass
     1    OW     1     SOL     OW       1      -0.82    15.999
     2    HW     1     SOL     HW1      1       0.41     1.008
     3    HW     1     SOL     HW2      1       0.41     1.008

 [ bonds ]
 [ angles ]
 [ dihedrals ]
 [constrains ]
```

This is a sample of full itp file. However, usually the itp file should split into two files. The border line is intentionally used to identify it. Here is some related terminologies, anything using “[ ]” bracketed are the reserved key words called the **directives**, the single molecule in the system is called **residue**. For GROMACS, it requests all the **[atomtypes]** directives input first before any residue’s bonded parameters, so it is why that the itp file needs to be split.

For the separation in the single line of those different parameters, the white-space can be used. However, it has an attention. By testing, I happened to find that GROMACS doesn’t have the ability to control and count the number of inputs. For example, look at the top itp file, we know that under the **[moleculetype]** directive, it needs two parameters, the first one is used to tell the molecule’s name, the other one is used to set the number-of-exclusions. In this line, if we add other parameters behind, it doesn’t have any influences for the compiling, which means the GROMACS only takes the number of parameters it needs from the beginning, as long as the input **type** is correct, for all other following parameters will be ignored.

Another attention is for the **[atom]** directive, because of the implementation of constrain algorithm, the maximum of the same charge group (cgnr) is the number 32, which means for the same charge group, only 32 atoms are allowed.

For top file

In principle, it has two types of top files. One is the full topology file including all the parameters which can be directly used for input. The other one is the pre-defined file, which needs to source the itp files. In here, we talk about the second one.

```
#define _FF_OPLS
#define _FF_OPLSAA

[defaults]
;nbfunc comb-rule gen-pairs fudgeLJ fudgeQQ
  1         3        yes      0.5     0.5

;;;LOAD ATOM TYPES

#include "/Path-To-AtomTypes/spc.atomtype.itp"

#include "/Path-To-Atoms/spc.itp"

[system]
; Name
Neat SOL

[molecules]
 SOL   10
```

The command starts with `#define` are used for the explanation of force field, this is intent for program future update. On the other hand, it also can be used as a share interaction with different force fields. For example, one can construct his own parameters, using the label, `#define _FF_force_filed_name`, and put this file under the folder `GROMACS_Folder/share/top/force_filed_name.ff/*itp`, then compile it, the GROMACS will automatically search the named share-folder to construct the full topology file.

The `[ defaults ]` tells the combination-rules, types of functions and ratios.

As a mention, in this pre-defined file, we can clearly see that the **atomtypes** directives are always loaded first.

gro file

The official link for GROMACS input gro file is: http://manual.gromacs.org/current/online/gro.html

Based on this document, again in the following we will only talk the file format that we will use.

We use the water gro file for example:

```
123456789012345678901234567890123456789012345678901234567890

MD of 2 waters, t= 0.0
    6
    1WATER  OW1    1   0.126   1.624   1.679  0.1227 -0.0580  0.0434
    1WATER  HW2    2   0.190   1.661   1.747  0.8085  0.3191 -0.7791
    1WATER  HW3    3   0.177   1.568   1.613 -0.9045 -2.6469  1.3180
    2WATER  OW1    4   1.275   0.053   0.622  0.2519  0.3140 -0.1734
    2WATER  HW2    5   1.337   0.002   0.680 -1.0641 -1.1349  0.0257
    2WATER  HW3    6   1.326   0.120   0.568  1.9427 -0.8216 -0.0244
   1.82060   1.82060   1.82060
```

The first line whose fonts are in red color is used for the char-position-reference.

The line `MD of 2 waters, t= 0.0` is the title line, which is used for the illustration and explanation for the input file. It will be ignored in the compiling.

The line starts with a number `6` , it is used to indicate for how many line entries will be read, and for those line entries, the common name is called the atoms.

For those entries, the char's position is very important.

First 5 positions occupied have to be integers, which is left-adjusted, this is used to label the residues, apparently, the same number means the same residue.

Following it, the five positions from 6~10 is used for residue name. This name is not much important and will be ignored in the compilation, the purpose in here is just for the human readable.

The positions in range 10~15 are used for the atom name, right-adjusted, which is very important, because this name is correspondent with all the Lennard-Jones and Coulombic parameters which are defined in the topology file.

The positions in range 16~20, right-adjusted, are used to count numbers.

The positions start in range 21~28, 29~36, 37~44, right-adjusted, are sequenced used for the residue's x-y-z Cartesian coordinates, the unit is **nanometer**. All these three number have to be the float number with three decimals.

The positions start in range 45~52, 53~60, 61~68, right-adjusted, are sequentially used for the residue's in x-y-z Cartesian projected velocities, the unit is **nm/ps** (or **km/s**). All these three number have to be the float number with four decimals.

The last line, only the simple case is introduced, three numbers are spaced by **white-space**, the position doesn't matter, which indicate the cubic box, and those numbers sequentially used for the box's length-width-height, the unit is **nanometer**.

The gro file is a collection of all the residues' parameters, and the sequences are corresponded with the top file **[molecules]** directives. For example, if we have:

```
[molecules]
 Residue_1     number_1
 Residue_2     number_2
 Residue_3     number_3
```

Then what the gro file does is sequentially collecting all those residues one-by-one.

pdb file

The official link for the format of protein data bank file is:
ftp://ftp.wwpdb.org/pub/pdb/doc/format_descriptions/Format_v33_A4.pdf

Still, we only talk the format that will be used.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890

TITLE     [ACE] -- Acetate
REMARK   7
REMARK   7 Ionic liquid OPLS-AA force field
REMARK   8
REMARK   8 Version 3/2017
REMARK   9
REMARK   9 Orlando Acevedo
REMARK  10
REMARK  10 Email: orlando.acevedo@miami.edu
REMARK  11
REMARK  11 Contributors: S.V. Sambasivarao, B. Doherty, X. Zhong,
REMARK  11 S. Gathiaka, B. Li
REMARK  12
REMARK  12 S.V. Sambasivarao, O. Acevedo, Development of OPLS-AA
REMARK  12 Force Field Parameters for 68 Unique Ionic Liquids,
REMARK  12 J. Chem. Theory Comput., 2009, 5, 1038-1050.
REMARK  13
REMARK  13 B. Doherty, X. Zhong, S. Gathiaka, B. Li, O. Acevedo,
REMARK  13 Revisiting OPLS Force Field Parameters for Ionic
REMARK  13 Liquid Simulations, 2017, (submitted).
ATOM      1  CZ  ACE     1      -0.162   0.384   0.226                       C
ATOM      2  C   ACE     1       0.450   1.325   1.247                       C
ATOM      3  H   ACE     1       0.066   2.331   1.089                       H
ATOM      4  H   ACE     1       0.190   0.991   2.251                       H
ATOM      5  H   ACE     1       1.533   1.328   1.135                       H
ATOM      6  O   ACE     1      -1.308   0.717  -0.142                       O
ATOM      7  O   ACE     1       0.530  -0.595  -0.125                       O
END

12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

Again the red-color-number lines are used as the **char-position-reference**.

All the inputs are **case-insensitive**.

Position 1~6, it is used for the reserved-words. Anything labeled with **TITLE** and **REMARK** will be ignored, the **TITLE** is used to identify the name of residues, and the **REMARK** is used for the emphases and marks, as an attention, the count starts at **7**.

The **ATOM** or **HETATM** (heteroatoms, in here cause our residue is nothing about the protein, it is in the same meaning and function of the word **ATOM**) is used for parameters. And the following char-positions are specified:

Position 7~11, right-adjusted, integer number, is used for counting atoms.

Position 13~16, used for atom names. It has two criterions. For real atom types which have two chars, like Ca, Cl, Br, Fe and so on, they take the **positions 13~14**. For others, starts at **position 14**.

Position 18~20, right-adjusted, is used for the residue name.

Position 23~26, right-adjusted, integer, is used for the counting residues.

Position 31~38, 39~46, 47~54, float number with three decimals, right-adjusted, are used for the atom's x-y-z Cartesian Coordinates, the unit is in angstrom.

Position 77~78, right-adjusted, is used for element symbol. It has the higher priority and takes the precedence. For example, in **position 13~14**, we have an atom named **Fe**, but if we label it in this position as **H**, then when the PDB viewers (like the **VMD** or **GaussView**) process it, they will take it as **H** but not **Fe**.

mdp file

The official link for the format of GROMACS mdp file is:
http://manual.gromacs.org/online/mdp_opt.html

For the simulation, in mdp file except for the run-and-out controls are modified, all other settings are kept constant. The changings are labeled in the following, for the liquid simulation:

```
; RUN CONTROL PARAMETERS =

integrator = md        ; md integrator
tinit = 0              ; [ps] starting time for run
dt = 0.001             ; [ps] time step for integration
nsteps = 1000000       ; maximum number of steps to integrate
comm-mode = Linear     ; Remove center of mass translation


; OUTPUT CONTROL OPTIONS =
nstxout = 100000        ; [steps] freq to write coordinates to trajectory
nstlog = 100000         ; [steps] freq to write coordinates to log file
nstenergy = 5000        ; group(s) to write to energy file
```

At the output control, we have;

```
nstxout = 100000        ; [steps] freq to write coordinates to trajectory
```

Which means the steps or frequency to write coordinates to trajectory file. For every output a common and wildly used name is **frame**.

The purpose of those modifications is to use the least disc memory with least frames stored but still reasonable and enough for the study and analyses.

Notation: for the energy frequency output we have:

```
nstenergy = 5000       ; group(s) to write to energy file
```

In MD simulation, GROMACS will calculation system's energy **at every step**, the output control only simply controls the frequency for the data written in the hard discs, but calculation itself stores every step's results, thus it is why the energy binary file is more accurate and precise than the **g_dump** files or **log** files.

For the proper simulation time determinations, please refer the another file named:

`Proof_for_Simulation_time.docx`

It has a detailed discussion.

■ com file

This type file is the Gaussian standard input file. The official link for this file's format is: http://gaussian.com/input/

```
%chk=GaussianInput_ALL_random.chk
# HF/6-31G(d) Pop=CHelpG
empty-line
GaussianInput_ALL_random Charge Analysis
empty-line
0 1
O        4.384   4.838  18.369
H        3.840   4.897  19.291
H        4.620   6.018  17.734
empty-line-1
empty-line-2
```

Only the useful format is focused.

In the line starts with **%** means the **check point file** output controls. And this check point file will contain all the structures information and calculation results, either for the orbital visible analyses or for the sake of program crash or the simplified inputs.

The line starts with **#** is the basis_sets.

For the file terminations. It at least should have two consecutive empty lines.

GAML for Charge Refining

All the systems' starting points are from the already equilibrated boxes, but only with charges changed. For the liquid, the simulation time is 1ns, while for gas is 2ns.

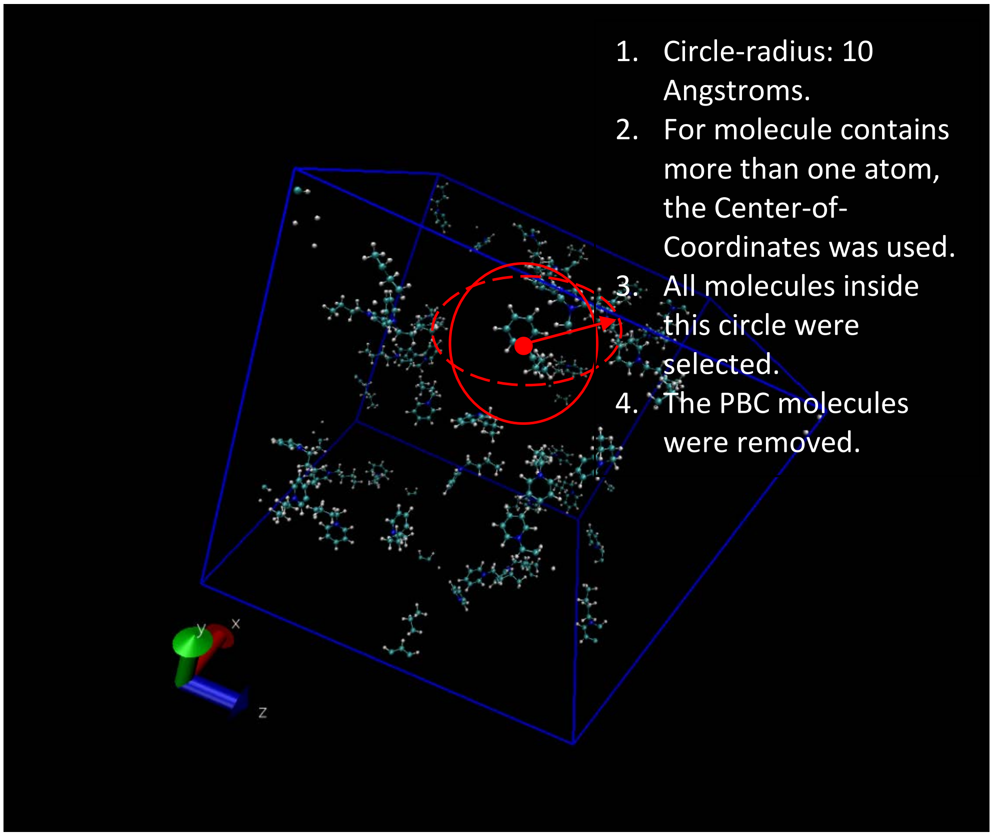
Charge Scheme

The first step, it's to find an appropriate way to generate different sets of charges. And this type charge generation for each atom should be consistent or reasonable with the purely structure analyses or common sense.

Since those charges will be finally used for the coulombic interactions, the electrostatic potential, so the quantum calculation in `# m062x/6-311+G(d,p) EmpiricalDispersion=GD3` level of theory is used.

So the next step is to find different residues to generate different `com` files. In order to cover a variety of combinations and **configurations**, we started with the already equilibrated box.

The following figure shows this concept, in a cubic box.



1. Circle-radius: 10 Angstroms.
2. For molecule contains more than one atom, the Center-of-Coordinates was used.
3. All molecules inside this circle were selected.
4. The PBC molecules were removed.

For example, if we want to find the two different combinations in two different molecules.

We firstly label those molecules 1 to *n*, then we randomly choose one of those molecules, and we randomly choose the second one from the remaining molecules to make the combination.

By randomly choosing those molecules, additionally if we use the different equilibrated boxes, along with the enough numbers of frame chosen, we will finally cover almost all the probabilities.

➢ Charge Range

For different frames, we will have different Gaussian charge calculation results. However, since it covers almost every combination/configuration, even good or bad results, we do not want to use all of them, so we have to find the proper charge ranges, which will finally decide the most favorable charge values for the real molecule states.

By collecting all the Gaussian output calculations in the log file. We can get the result like:

		Result_1			**Result_2**			**Result_3**
1	C	0.204316	1	C	-0.480945	1	C	0.134814
2	H	0.140943	2	H	0.202661	2	H	0.136437
3	C	-0.308636	3	C	0.614507	3	C	-0.232902
4	H	0.198395	4	H	0.104666	4	H	0.166667
5	C	0.253728	5	C	-0.538627	5	C	0.143675
6	H	0.114765	6	H	0.38117	6	H	0.110585
7	C	-0.288747	7	C	-0.327594	7	C	-0.232181
8	H	0.192429	8	H	0.435618	8	H	0.150664
9	C	0.15602	9	C	0.062087	9	C	0.106455
10	H	0.160123	10	H	0.191108	10	H	0.183518
11	N	-0.121176	11	N	0.356898	11	N	-0.085755
12	C	0.064	12	C	-0.371408	12	C	0.507346
13	H	0.085024	13	H	0.012592	13	H	-0.079473
14	H	0.047837	14	H	0.178622	14	H	-0.014491
15	C	-0.053523	15	C	0.380031	15	C	-0.997291
16	H	0.050279	16	H	-0.03545	16	H	0.232475
17	H	-0.000865	17	H	-0.134276	17	H	0.202642
18	C	0.178785	18	C	0.008379	18	C	0.30709
19	H	-0.001436	19	H	-0.030328	19	H	0.071597
20	H	-0.036362	20	H	0.057498	20	H	0.064163
21	C	-0.245494	21	C	-0.267729	21	C	-0.183945
22	H	0.06761	22	H	0.067014	22	H	0.148773
23	H	0.078487	23	H	0.063562	23	H	0.089672
24	H	0.064959	24	H	0.074317	24	H	0.071908

So what we do is collecting each atom's charge results, only for 1-Butyl-Pyridium, to form 24 different charge-sets. Then calculate those charge-sets one-by-one to yield good atom's charge ranges.

For example, for atom-1 carbon atom, if we have the charge value list is;

```
atom_1 = [0.18, 0.69, 0.77, 0.87, 0.03, 0.67, 0.53, 0.79, 0.58, 0.19, 0.03, 0.37,
0.0, 0.05, 0.76, 0.39, 0.36, 0.46, 0.54, 0.68]
```

We then sort this list from small to large;

```
atom_1_sorted = [0.0, 0.03, 0.03, 0.05, 0.18, 0.19, 0.36, 0.37, 0.39, 0.46, 0.53,
0.54, 0.58, 0.67, 0.68, 0.69, 0.76, 0.77, 0.79, 0.87]
```

Starting with the smallest number, we use the `stepsize = 0.1` to find the numbers of value falling within every evolution. Then we can get the results like;

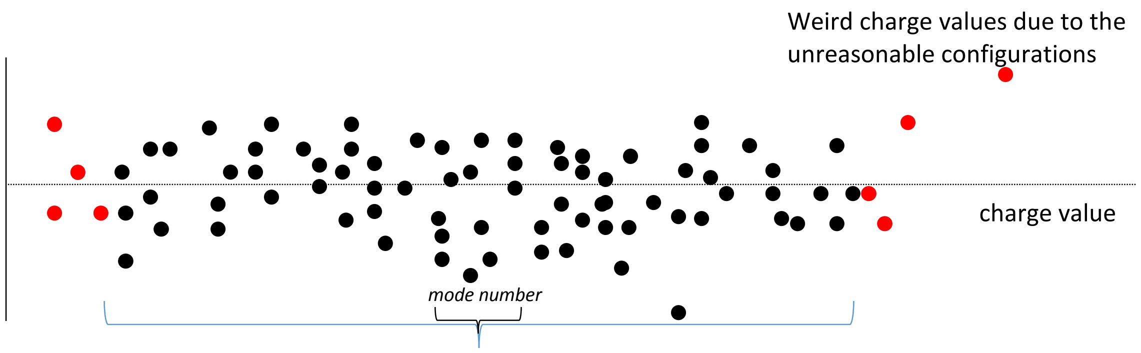
```
0.0~0.1  0.1~0.2  0.2~0.3  0.3~0.4  0.4~0.5  0.5~0.6  0.6~0.7   0.7~0.8  0.8~0.9
   4        2        0        3        1        3        3         3        1
```

Note, for range `a~b`, which means any value `x` fulfills `a <= x < y` will be counted.

Now, we have a list containing the counted results. (In code implementation, we don't really need to sort the list, we can directly get the counted results.) For this list, we start with the biggest number, it is also called the **mode number in mathematic,** to find a surrounding range, and the summary of this surrounding range in the ratio on the total number no less than certain percent.

For example, if we set `percent = 0.8`, which means the summary of the result range should be no less than $\mathrm{percent} \times \mathrm{total\_number} = 0.8 \times 20 = 16$. The biggest number is 4 on the most left, then the search will forward to right. Finally, we can get the last number at range `0.6~0.7`, then we can determine that the charge range from `0.0~0.7` is the final range we want.

If we plot this selection in a graph, it can be shown like;

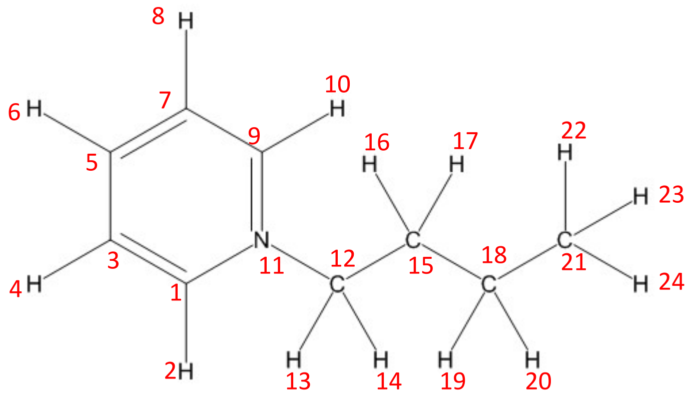


In this range, the percent of number of charges in total number of charges is no less than the `percent` parameter.

➢ Symmetry Consideration

The symmetry means one has to consider the chemical equivalent of atoms in the molecule.

Still for the 1-Butyl-Pyridium.



We labeled each atom's number. We can know that, the atom 1 and 9, 2 and 10, 3 and 7, 4 and 8, 13 and 14, 16, 17, 19 and 20, 15 and 18, 22, 23 and 24 are chemical equivalent. Easily, we can use the **2D list** to represent it;

```
[[1,9],[2,10],[3,7],[4,8],[13,14],[16,17,19,20],[22,23,24],[15,18],5,6,11,12,21]
```

Considering the atom's chemical symmetry property, in other words means the final atom's charge used for the MD simulation should be the same.

➢ Charge Constrain

Inherently it has at least one constrain for the total charges, either is neutral or equal to some scale numbers. If we use the **full charge systems**, which means the generated total charge for the 1-Butyl-Pyridium is **+1.0**.

Beside it, it also has another four common constrains. Frist, for some special atoms, we always would like to keep their changes either positive or negative. For example, hydrogen atom, it is nonsense to assign negative charges on it in the organic compounds. Thus another parameter should be defined;

```
bool_limit = [2p, 4p, 6p, 13p, 16p, 22p]   # 'p' means positive
```

Second, for atom's partial charges, usually its absolute value should not be bigger than number 1, so we have to have a parameter to define that `threshold=1`. Third, for most atoms in MD simulation, zero charges should be avoided, `bool_nozero=True`. Last, while this one might be complicate, for chain alkane molecules for instance, we want to make charge parameters to be extensible, so the total charges of the terminal function groups like **CH_3** has to be considered, which means the charge summary of these four atoms should be zero, so the parameter can be set `counter_list=[21,22]`.

However, to make constrain fulfilled, the offsets have to be defined. In order to minimize the offset influences, in total two different offsets can be used, defining in the parameter `offset_list`;

```
offset_list = [11,1]
```

The reason for choosing these two different atoms (in the chemical equivalent, yet the total atom number is three, one nitrogen and two carbons) is that, we already know the full charges in the pyridine-ring have the stronger influence on the overall properties, especially for the charges on the nitrogen atom. Therefore, this nitrogen atom is used as the strong offset in the first place, and the other one type atom two carbons are used as the weak offset.

We have the full symmetry-list. First, we randomly generate charges for other atom-types except these two offsets based on the given charge ranges. The next step is that we will try certain number to randomly generate the weak offset charges based on the charge ranges. To be clear, if we set this certain number to 5, `offset_nm=5`, then at every trial, we can use the charge constrain, `total_charge=1.0`, to calculate the strong offset charge. If this calculated charge fall within the given atom's charge range, then the charge generation is done. If not, repeat it until the set offset trial number. If the trial reach the threshold but still none of them fulfill the charge range demands, then we will use the average of these number charge trails as the final strong offset atom charges, then use the charge constrain to calculate the strong offset atom charges.

MD&GAML Simulations

MD simulation is evolution of Newton second equation, so which means every step can be reproduced if the specified settings are given. So all the original files have to be attached for check and filter out. And also the output files should not override any files by default.

Example 1: Genetic Algorithm

Genetic algorithm (GA) is a metaheuristic inspired by the process of natural selection. Typically, it has two prerequisites;

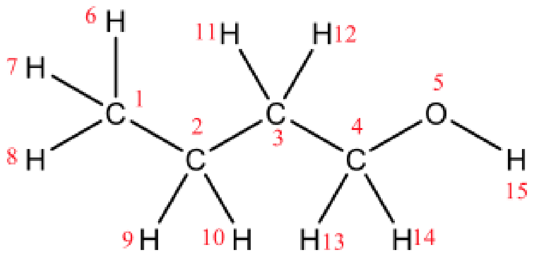
T1: A genetic representation of the solution domain.
T2: A fitness function to evaluate the solution domain.

This algorithm is designed for the parameterization of the charges for the small solvent molecules based on the existing OPLS-AA force fields. So the charge parameters and its corresponded MD results like Heat-of-Vaporization or Free Energy of Solvation will be used as the genetic representation (**T1**). Then the result of average-of-percent-error is used for the evaluation, after that, the best-fit charge genes are selected for the next generation, for which, three main types of rules are created;

(1) selection rules select the “parents” that contribute towards the next generation;
(2) crossover rules combine two parents to form “children” for the next generation;

(3) mutation rules apply random changes to individual parents to form children.

If we use the 1-butanol as an example,



To clarify, each atom is sequentially labelled a number. In the real-world, due to the single bond rotation, atoms in number **6**, **7**, **8** or **9**, **10** or **11**, **12** or **13**, **14** will be chemical equivalent. For four carbon atoms, we know that the farther distance with the Oxygen atom, the minor influence of charge induction effect. Presumably, compared with the alkane carbons, the charge on carbon atom **4** will be greater difference than any other three carbon atoms.

However, it will be trivial if we differentia the charges on carbon atom **2**, **3**, besides, we also want the charge parameters to be extendable, which means the charge on the same atom type, like in the OPLS-AA, will always have the same value, no matter how long the alkane chain is. This need is only designed for the same functional group molecule parameterizations, if we only focus on one molecule, this restriction should be released. Therefore, the charges on methyl group {**1**, **6**, **7**, **8**} should be neutralized. So the charges can be represented;

```
symmetry_list = [[2,3],[6,7,8],[9,10,11,12],[13,14],1,4,5,15]
counter_list  = [1,6]
```

Inside the `symmetry_list`, chemical equivalent atoms are indicated by the square-brackets. The `counter_list` will search back the `symmetry_list` to find the corresponded number of atoms, which means in here, the charges on atom type **1** (one atom) will be countered with the atom type **6** (three atoms). In case the `symmetry_list` is omitted, no chemical equivalent atoms at all, the other format `counter_list` can be used, however, if the `ratio` of the countered atom is not the 1 to 1, the number of atoms have to be clearly declared. For example, `counter_list=[1,2,5,3]`, which means the charges on two atom-type **1** are neutralized with the charges on three atom-type **5**.

Based on the `symmetry_list`, the charge pairs are sequentially constructed, for example;

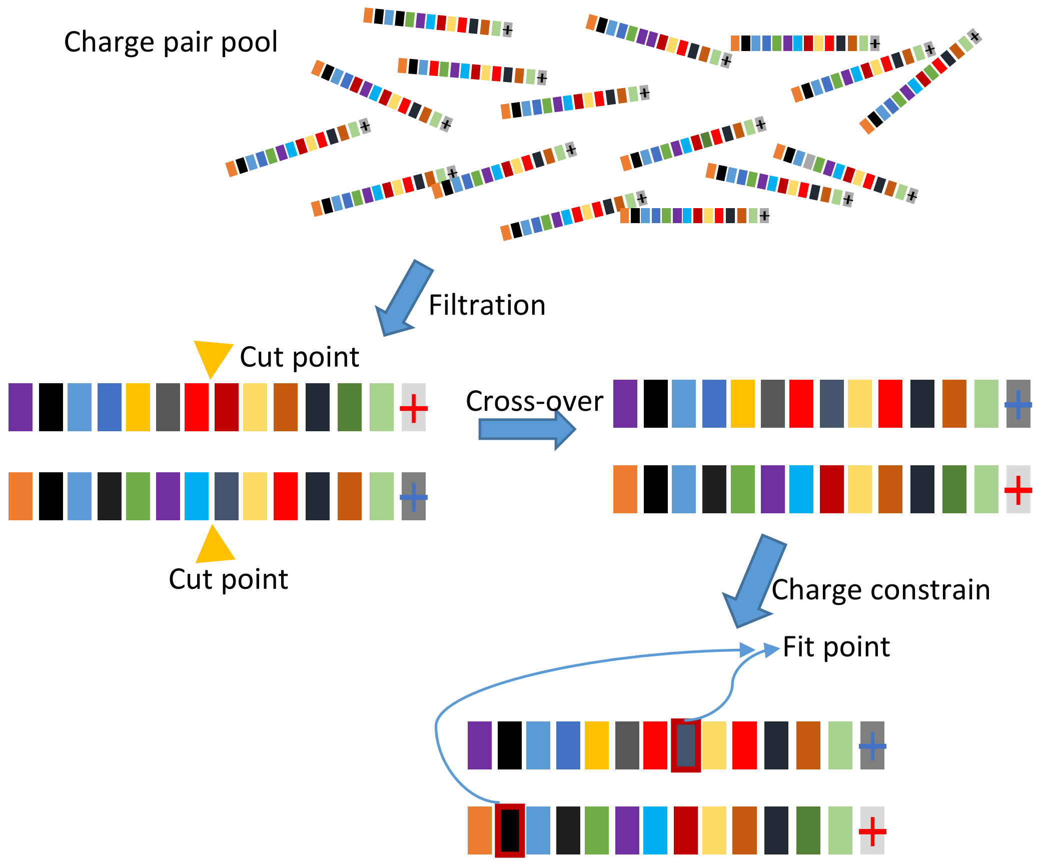
The rectangles with different color are represented different charge values, the one with the “plus” sign are the calculated averaged MD properties, which is used for the filtration.

At the starting point, many charge pairs are built for the MD simulations. Because we know that for the given molecule, the total charge is fixed, so inherently it will have the charge constrain. To apply it, the offset-point is defined. However, because for all other charge genes are generated from the calculated atom charge ranges but not the offset-point, so its value will be overflow or underflow the range bounds, thus in order to minimize this effect, a two offset points are defined, like `offset_list = [5,15]`. By defining in this way, we know that the atom type **5** will be used for the **strong** offset point yet the atom type **15** as the **weak** offset point. In this definition, first certain number trails will be used to fit the weak offset point based on the charge ranges, for each try, the strong offset point is calculated by the difference between `total_charge` and all other charge summations. If this calculated charge fall within the given strong offset point atom's charge range, then the charge generation is done. If not, repeat it until the set offset trial number, `offset_nm`. If the trial reach this number but still none of them fulfill the charge range demands, then we will use the average of those charge trails as the final weak offset atom charges, then use the charge constrain to calculate the strong offset atom charges.

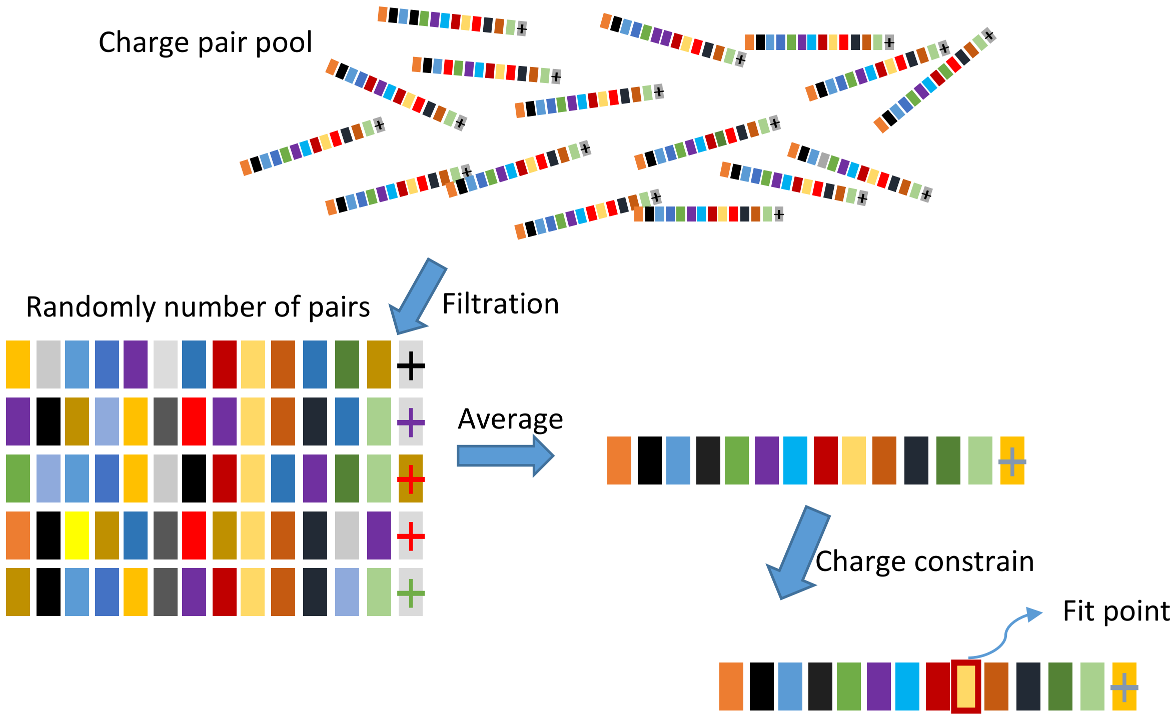
```
strong-offset + weak-offset = WS = total_charge – remaining_charge_summation

Charge Range
strong-offset: [s-low, s-high]
weak-offset  : [w-low, w-high]

Use weak-offset charge range generate charge, W_1. Then calculate strong-offset;
S_1 = WS – W_1

If S_1 belongs to strong-offset range, [s-low, s-high], done.
If not, try W_2, calculate S_2 … until offset_nm

If none of them, then weak-offset = average(W_1, W_2, W_3 … W_offset_nm), then
strong-offset = WS – weak-offset
```

After the MD simulation is done, the corresponded properties are collected as the leader genes. For the pairs with both charge genes and leader gene, we use pre-setup value filter out all the "bad" gene pairs whose leader genes are bigger than this value. Then randomly choose two pairs from the remaining pairs as the parent genes to make the **cross-over** generation. Again, the charge constrain should be considered. Different with the offset, the fit-point is randomly chosen.

Because of the charge on the strong offset is a calculation from the total charge and all other remaining points, its value under some circumstance will be too positive (bigger than +1) or too negative (less than -1), which is not reasonable in the fractional charge parameters, thus should be removed. So the setting in the absolute value `threshold = 1` is used to filter out all those pairs. Besides, for most atoms except free hydrogen in the oxhydryl group, the zero charge is nonsense, thus should be removed, the parameters `bool_nozero = yes` should be used.



Beside the cross over method, the average method is also used to generate average charge pairs. **The reason is on the assumption (but in fact it is not) that the system's energy is continuing and integrable on the charge changes.** It is like an approach method, to find the best average value between best genes. The pros are that this method is very reasonable and efficient, we can image that as long as the selected charge pairs have good genes, the averaged charge pair will always keep those good properties but continuously focus on the local refinements. However, the con is that it possibly will be not accurate, since it doesn't have the ability to cover all the minimum points.



The third method is the mutations, the parameter `charge_extend_by=0.3` is added. It is used to either extend or shrink the calculated charge in low or high range bound, which can maximumly increase the charge flexibilities. The purpose is to find the global minimum point.

To better parameterize charge pairs, the ratio of three methods is set to `mutation : average : cross-over = 1:2:7`. Besides, this parameter can also be used to change the program focus. For example, if we increase the ratio of the average method, then which means the program will spend much time on the local refinements.

➢ Example 2: Average Algorithm

In this example, four systems are chosen, they are **BPYR_Br**, **BPYR_Cl**, **BPYR_BF4** and **BPYR_NTF2**. The starting point for the equilibrated boxes, for the first two is the 20ns npt simulation, and for the last two is 40ns production run.

If we have got equilibrated boxes (the gro files), mdp files, and different charge pairs:

```
PAIR   0.080  0.110 -0.310  0.180  0.040  0.030  0.030  0.020  0.080  0.120  0.850 -0.180 -0.320
PAIR   0.220  0.170 -0.200  0.150  0.130  0.010  0.080  0.050  0.140  0.110  0.110 -0.310 -0.370
PAIR   0.040  0.130 -0.230  0.160  0.100 -0.020  0.030  0.160  0.160  0.120  0.310  0.000 -0.320
PAIR   0.090  0.170 -0.330  0.170  0.030 -0.050  0.050  0.060  0.170  0.130  0.620  0.060 -0.310
PAIR   0.050  0.170 -0.260  0.150  0.080  0.000  0.070 -0.010  0.140  0.110  0.570 -0.100 -0.290
PAIR   0.110  0.110 -0.220  0.170  0.100 -0.060  0.040  0.010  0.210  0.110  0.500  0.050 -0.310
PAIR   0.110  0.120 -0.180  0.150  0.050 -0.010  0.090 -0.170  0.160  0.130  0.760 -0.250 -0.190
PAIR   0.070  0.140 -0.210  0.160  0.060  0.000  0.090  0.130  0.140  0.110  0.510 -0.340 -0.390
PAIR   0.130  0.170 -0.330  0.160  0.070 -0.030  0.030  0.070  0.100  0.120  0.540 -0.040 -0.230
PAIR   0.110  0.120 -0.270  0.150  0.040 -0.030  0.070  0.020  0.170  0.130  0.740 -0.140 -0.330
```

Those charges start with the keyword **PAIR**, and each line is corresponded to the input symmetry_list. Then those charge pairs can be used to generate different top files.

To generate GROMACS topology files, specially choose the reside 1-Butyl-Pyridium, `reschoose='bpy'`, the parameters are case-insensitive. Then use the mdp file setting to get the GROMACS binary input tpr files, separately both in gas phase and in liquid phase. After the simulation is done, we can get each system's Heat-of-Vaporizations. And for those calculation results we already know the literature values. What we do is comparing these simulation results with the experiment values to find the better ones. When simulation is done, choosing any favorable charge pairs, whose error is less than the setting `error_tolerance`, as the next MD run.

For the second train, after the first simulations are finished, again we choose the first **10** most favorable charge pairs of each system based on their experimental values. Then the next step is doing the random combinations, like;

```
    Name:         System_1    System_2    System_3    System_4
Charge_pair_nm      nm_1        nm_2        nm_3        nm_4
  Choose_nm          a            b           c           d    --->  average
```

The randomly choose charge pair number is from zero to ten.

The cation in here is still on the charge constrains. We know that because of the round error, the total charge may not be exactly equal to the total charge we set. So the solution that I took is randomly choose one atom type from the symmetry list as the offset, then use it generate the averaged charge pairs.

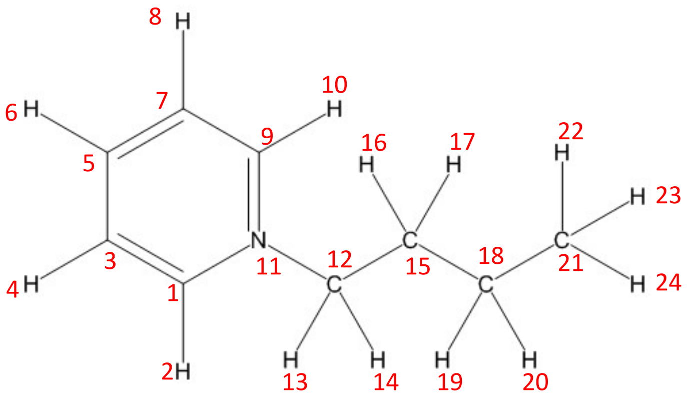
The reason for the average-charge pairs is based on the “fake assumption” that the system's energy is continuing and integrable on the charge changes.

In order to cover all the charge possibilities and search all the energy local minima. A charge extension is defined, `extend_charge_range_by = 0.3`, which means the charge range's low bound and high bound either shrinks by this number or extends by this number. Then we use this method to generate another 10 charge pairs. Combining with the minima 10 charge pairs, we again get 20 charge pairs for the next MD run. And from here, the simulation repeats.

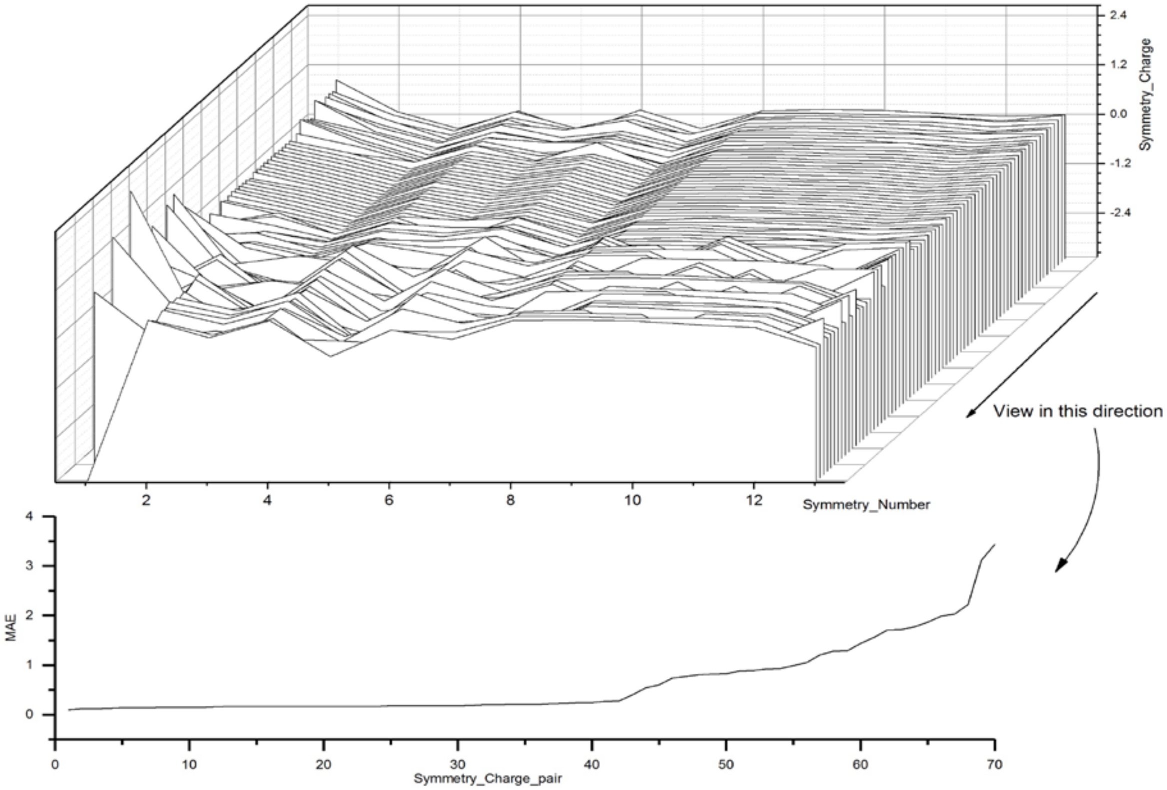
▪ Results analyses

We still use these four systems as an example.

The molecule's label is;



If we plot 70 different charge pairs along with their Mean Absolute Error;



NA	CY	CW	CR	CA	CS	CT	HY	HW	HR	HA	HS	CT
[11]	[1,9]	[3,7]	[5]	[12]	[15,18]	[21]	[2,10]	[4,8]	[6]	[13,14]	[16,17,19,20]	[22,23,24]

For the 3D picture, the vertical axis is the index of the charge values. The plane Z-axis means the charge pair numbers. In the table, they are atom types and atom numbers, respectively. Inside each list different numbers means the chemical equivalent.

If we only consider the first twenty charge pairs with the minimum MAE.

NA	CY	CW	CR	CA	CS	CT	HY	HW	HR	HA	HS	CT	MAE
0.87	0.1	-0.31	0.12	-0.38	0.14	-0.31	0.12	0.16	0.13	0.05	-0.01	0.03	0.1
0.67	0.1	-0.29	0.18	-0.22	0.11	-0.37	0.14	0.14	0.11	0.06	-0.04	0.09	0.12
0.7	0.02	-0.31	0.11	-0.17	0.07	-0.34	0.11	0.16	0.12	0.13	0.01	0.06	0.12
0.7	0.15	-0.33	0.14	-0.28	0.03	-0.21	0.15	0.17	0.12	0.08	-0.03	0.05	0.13
0.58	0.09	-0.32	0.13	-0.03	0.2	-0.32	0.13	0.15	0.13	0.08	-0.06	0.03	0.14
0.74	0.11	-0.27	0.17	-0.14	0.02	-0.33	0.12	0.15	0.13	0.04	-0.03	0.07	0.14
0.85	0.08	-0.31	0.08	-0.18	0.02	-0.32	0.11	0.18	0.12	0.04	0.03	0.03	0.14
0.48	0.13	-0.29	0.15	-0.09	0.05	-0.35	0.17	0.16	0.12	0.1	-0.04	0.07	0.15
0.54	0.13	-0.33	0.1	-0.04	0.07	-0.23	0.17	0.16	0.12	0.07	-0.03	0.03	0.15
0.62	0.09	-0.33	0.17	0.06	0.06	-0.31	0.17	0.17	0.13	0.03	-0.05	0.05	0.15
0.7	0.12	-0.3	0.16	-0.12	0	-0.39	0.17	0.15	0.13	0.12	-0.06	0.08	0.15
0.43	0.12	-0.28	0.14	-0.11	0.09	-0.31	0.14	0.16	0.11	0.09	-0.02	0.06	0.16
-0.13	0.02	-0.18	0.22	-0.15	0.3	-0.31	0.12	0.15	0.12	0.08	0	0.09	0.17
0.41	0.11	-0.27	0.15	-0.09	0.09	-0.31	0.15	0.16	0.1	0.08	-0.02	0.06	0.17
0.42	0.1	-0.27	0.14	-0.09	0.1	-0.31	0.14	0.16	0.12	0.08	-0.02	0.06	0.17
0.42	0.1	-0.27	0.15	-0.08	0.11	-0.3	0.14	0.16	0.1	0.08	-0.02	0.05	0.17
0.42	0.1	-0.27	0.16	-0.09	0.1	-0.31	0.15	0.16	0.12	0.08	-0.03	0.06	0.17
0.43	0.09	-0.25	0.16	-0.1	0.08	-0.32	0.14	0.16	0.11	0.09	-0.02	0.06	0.17
0.43	0.1	-0.27	0.15	-0.09	0.1	-0.33	0.14	0.16	0.12	0.08	-0.02	0.06	0.17
0.43	0.1	-0.27	0.15	-0.07	0.1	-0.31	0.14	0.16	0.12	0.08	-0.03	0.06	0.17
0.46	**0.13**	**0.08**	**0.1**	**0.44**	**0.2**	**0.18**	**0.06**	**0.04**	**0.03**	**0.1**	**0.09**	**0.06**	

Purely based on the structure analyzation, it is impossible to get negative charges on the nitrogen atom. The line with the dark red color can't be neglected. The reason it gets lower MAE is just the exception, it is not in the common cases. So except this line, the bottom with the bold font is the data range in each column.

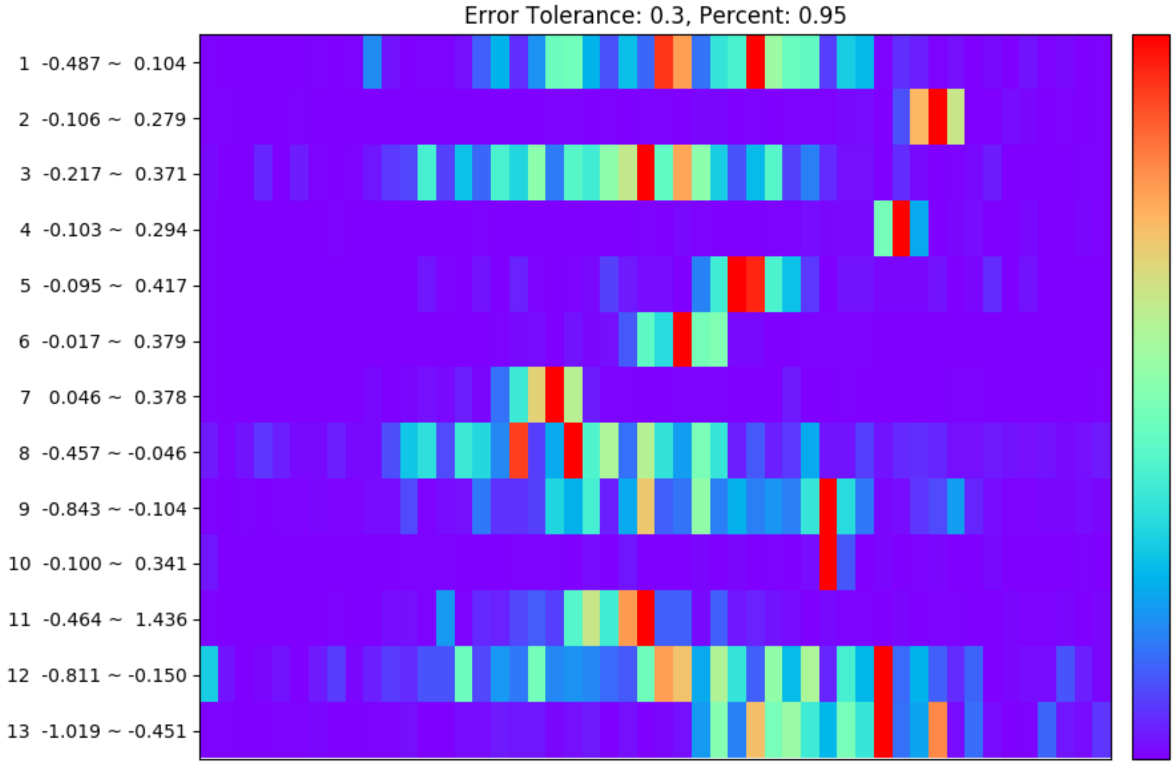
If we set some levels to reflect the influence index, which means the degrees of charge value changes cause the fluctuation of the Heat of Vaporizations, then it will be;

Levels 1:	0.00 ~ 0.10	almost no influences
Levels 2:	0.10 ~ 0.20	small in influences,
Levels 3:	0.20 ~ 0.30	medium
Levels 4:	> 0.30	dominated

Based on this distinction, we can clearly see that the charges on the nitrogen and on the first none pyridine carbon atom linked with the nitrogen (**CA**) have the most influences on the overall Heat of Vaporization properties.

And charges of the pyridine-ring carbons, **CW**, which directly linked with the nitrogen atom, also exert the great influences on the Heat of Vaporizations. Interestingly, the outmost terminal carbon atom, **CT**, also take the great responsibilities on the changes of Heat of Vaporization properties, which can be explained by the structure analyses. We know that overall charges used for the 1-Butyl-Pyridium cation is **+1**, the majority positive partial charges are spread on the pyridine-ring, especially on the nitrogen atom. So, in order to get the lowest intramolecular coulombic interactions, the most favorable choose is to put rest charges on the furthest atoms.

The Feature Statistical Selection will output the analysis result like;



If multiple runs are executed, combine them, we will know which atoms will have big influence on the overall properties changes.

```
# Strong to weak

# Error-tolerance = 0.20, percent-range = 0.80
CY     CS     CA     CW     CR     HS     HT     NA     HR     CT     HA     HW     HY

# Error-tolerance = 0.20, percent-range = 0.90
CY     CA     CS     CR     CW     HT     HS     HR     CT     NA     HA     HW     HY

# Error-tolerance = 0.25, percent-range = 0.80
CA     CY     CS     CR     CW     HS     HR     NA     HT     CT     HA     HW     HY

# Error-tolerance = 0.25, percent-range = 0.90
CY     CA     CS     CR     CW     CT     HR     NA     HS     HA     HW     HT     HY

# Error-tolerance = 0.28, percent-range = 0.80
CA     CY     CS     CR     CW     HR     CT     NA     HA     HS     HT     HY     HW
```

```
# Error-tolerance = 0.28, percent-range = 0.90
CA    CY    CS    CR    CW    HR    CT    NA    HS    HA    HY    HT    HW

# Error-tolerance = 0.30, percent-range = 0.80
CA    CY    CS    CR    CW    HR    NA    CT    HS    HA    HT    HW    HY

# Error-tolerance = 0.30, percent-range = 0.90
CA    CY    CR    CS    CW    HR    NA    HT    HS    CT    HA    HW    HY
```