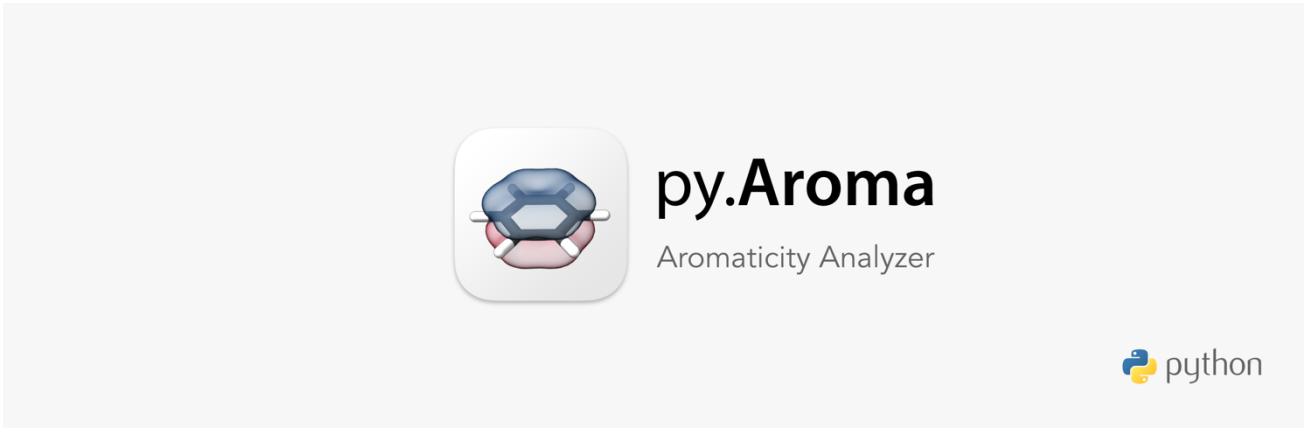


– An Aromaticity Analyzer –

Version 1.0.0

Homepage https://wongzit.github.io/program/pyaroma/

Developed by

Zhe Wang (Zit Wong)

E-mail wongzit@yahoo.co.jp

Homepage https://wongzit.github.io/

Graduate School of Science, Hiroshima University, Kagamiyama 1-3-1,

Higashi-hiroshima, Hiroshima 739-8526, Japan

Contents

1. Overview

1.1 About py.Aroma

py.Aroma is an open-source Python software for generating input files of NICS and ICSS calculations, and processing output files of HOMA, NICS, ICSS calculations. It combined the basic functions of **ICSSgen**, **ICSScsv**, **NICSgen**, **HOMAcalc**, **ICSSgen3D** and **ICSScub3D**.

py.Aroma is developed with Python 3.9.6, thus, users can run **py.Aroma** through Python IDE.

py.Aroma can be freely download from author's GitHub page (https://github.com/wongzit/pyAroma). The source code, executable files for macOS, Linux and Microsoft Windows could be found for the latest release.

1.2 How it Works

py.Aroma reads a *Gaussian* (Gaussian, Inc.) input or output file, users can choose which function they want to use: (1) creating input files for NICS, 2D-ICSS and 3D-ICSS calculations, or (2) extracting magnetic shielding tensors from 2D-ICSS and 3D-ICSS calculation outputs, or (3) calculating HOMA value of an optimized geometry.

1.3 Testing Platform

py.Aroma has been test on following platform:

1) MacBook Air (M1, 2020), macOS 11.5.2

2) Mac mini (i5-8500B, 2018), macOS 11.5.2

3) Intel Core™ i7-9700KF PC, Microsoft Windows 10 Education

4) Intel Core™ i7-10700 PC, Microsoft Windows 10 Education

5) Intel Core™ i7-10700 PC, Red Hat Enterprise Linux 8

6) MacBook Air (M1, 2020) virtual machine, Fedora 33 ARM64

7) MacBook Air (M1, 2020) virtual machine, Microsoft Windows 11 Pro (Insider Preview, build 22000.160)

2. Install

2.1 Run with Source Code

If Python IDE is already installed, **py.Aroma** could be run with the source code. Python 3.9+ is recommended. You can download the latest version of Python from homepage (https://www.python.org). **py.Aroma** is running with external library *matplotlib* and *numpy*, please make sure *matplotlib* and *numpy* are already installed on your computer before running **py.Aroma**.

For macOS and Linux users who want to run with source code, please run following command in terminal:

```
python3 /path_to_pyAroma/pyAroma_src_v_1_0_0.py
```

For Windows users, please execute following command in PowerShell or Command Prompt (*cmd.exe*):

```
py3 /path_to_pyAroma/pyAroma_src_v_1_0_0.py
```

2.2 Run with Executable File

2.2.1 Use Pre-packaged Executable File

Executable files are pre-packaged in *execufiles* folder.

For macOS/Linux users, you may need to add permission to the executable file before running for the first time. Assume the executable file is located at "`/home/user/pyAroma/execufiles/pyAroma_mac_v_1_0_0`", run below command to add executable permission to it.

```
chmod +x /home/user/pyAroma/execufiles/pyAroma_mac_v_1_0_0
```

For macOS/Linux users, you can add following contents to the *my_profile* (for macOS) or *.bashrc* (for Linux) file. Then, you can run **py.Aroma** with simple command `pyaroma`.

```
alias pyaroma=/home/user/pyAroma/execufiles/pyAroma_mac_v_1_0_0
```

For Microsoft Windows users, you can just run **py.Aroma** by double clicking the *pyAroma_win_v_1_0_0.exe* file.

2.2.2 Package from Source Code

If the pre-packaged executable files do not work normally, please try to run with source code, or package from source code with packaging tools like *pyinstaller*. The *pyinstaller* could be installed from `pip`:

For macOS/Linux: `pip3 install pyinstaller numpy matplotlib`

For Microsoft Windows: `pip install pyinstaller numpy matplotlib`

And then, run following command in command window:

```
pyinstaller /path_to_pyAroma/pyAroma_src_v_1_0_0.py --onefile
```

The packaged executable file will be generated in the *dist* folder name as *pyAroma_ src_ v_ 1_ 0_ 0.exe*. Only *.exe* file is needed, you can delete other files.

2.2.3 Common Issues

If the packaged programs cannot work due to system security problem, please refer to the following methods.

For masOS users:

If warning window says "Cannot open an app from an unidentified developer" showed up when you run packaged executable file, please go to "System setting" -> "Security & Privacy" and click "Open Anyway", click "Open". Then, you could run the program by double click.

For Microsoft Windows users:

If you are using Windows Defender (Windows Security) on your computer, the program may be stopped by Windows Defender. To avoid this, you need set a safe path for the program.

1) Turn off the real-time protection of Windows Defender (Windows Security) temporary.
2) Download the program. Unzip the *.zip* file and move it to the dictionary as you like.
3) Add the program file to the safe file list.
4) Turn on the real-time protection of Windows Defender.
5) Run the program, click "Show more information" and "Run/Open" in the warning windows (if there is).

3. Usage

3.1 Before Running

3.1.1 File Type

py.Aroma could recognize *.gjf*/*.com* as *Gaussian* input files and *.log*/*.out* as *Gaussian* output files. If your files have other file extension type, please rename it or tell the file type to py.Aroma.

```
Gaussian input/output file: support .gjf/.com/.log/.out
(e.g.: C:\pyAroma\examples\benzene.gjf)
/Users/wangzhe/Desktop/benzene.abc

Could not determine the file type, please specify:
   1 - Gaussian input file
   2 - Gaussian output file.
Please input 1 or 2 and press ENTER: 1

Gaussian input file detected.
```

3.1.2 For Creating Input Files for NICS and ICSS Calculations

You need prepare a *Gaussian* input file (*.gjf* or *.com*) including route section, and molecular coordinates. Please notice that only Cartesian coordinates is allowed. An example input file of benzene is attached:

```
%nprocshared=8
%mem=10GB
#p nmr=giao rb3lyp/6-31g(d)

Benzene_opt

0 1
 C              -1.33923600   -0.39585300    0.00000500
 ...
 H              -1.80027600    1.71035300   -0.00006300
```

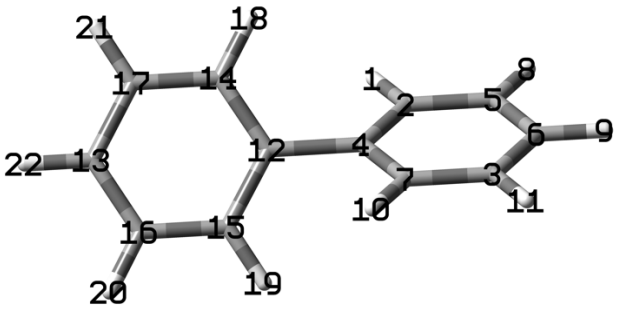
3.2 Create Input Files for NICS Calculations

1) Choose function `1 – NICS(n)` to load NICS module.

```
Please select calculation type:
```

```
  1 - NICS(n)
  2 - 2D-ICSS map
  3 - 3D-ICSS map
Input menu number and press ENTER: 1
```

2) Specify which cyclic ring would be investigated. Please input 3 or more atoms number (as defined in input file). For example, ghost atoms would be added in the phenyl ring, C12-C14-C17-C13-C16-C15, thus, you can input 13 14 15 or 12 16 17 or 12 13 14 15 16 17 (non-order sensitive). 13 16 15 is not a good choice because the ghost atoms would not locate at the center of the ring.




```
Please specify the target atoms number:
12 16 17
```

3) Specify the altitude over the ring plane. The altitude is defined in Å.

```
Please specify the altitude n of NICS(n):
1
```

4) If you want to add Bq atoms for other rings, like C2-C4-C7-C3-C6-C5, please input "y" and press ENTER. To quit the program, please input "*n*".

```
Continue to add other Bq atoms? (y/n):
y
```

5) A new input file for NICS calculation would be generated in the same dictionary as original input file, named with "*xxx_NICS.gjf*".

3.3 Create Input Files for 2D-ICSS Calculations

1) Choose function `2 – 2D-ICSS map` to load 2D-ICSS module.

```
Please select calculation type:
  1 - NICS(n)
  2 - 2D-ICSS map
  3 - 3D-ICSS map
Input menu number and press ENTER: 2
```

2) Specify the plane which 2D-ICSS map would be plotted on. The plane is defined in XY, XZ and YZ with no case-sensitivity. So, for user inputting, "xy", "XY" or "xY" are same.

```
Please specify the plane for ICSS map (XY, XZ, YZ):
xy
```

3) Specify the altitude over the molecular plane. The altitude is defined in angstrom (Å).

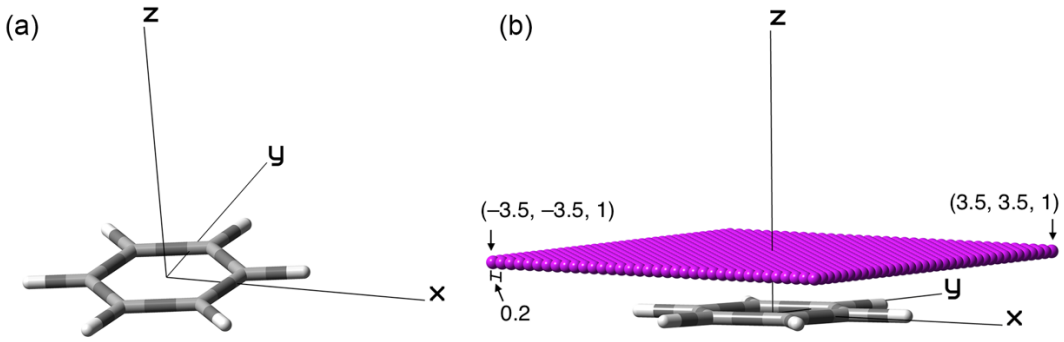
```
Please input the altitude over the plane (in angstrom):
1
```

4) Specify the 2D-ICSS map range. For each direction, minimum and maximum values are needed. After finish, the program will print out the user-determined parameters.

```
Please specify the range of X axis (in angstrom, e.g., -10 10):
-3.5 3.5

Please specify the range of Y axis (in angstrom, e.g., -8 8):
-3.5 3.5

2D-ICSS(XY,1.0) map in [X: -3.5 to 3.5, Y: -3.5 to 3.5].
```

5) Specify the grid quality. The grid quality is defined by the distance between two neighboring ghost atoms. Smaller grid value will give a smoother 2D-ICSS map, but more calculation cost is necessary. In my experience, a grid value of 0.25 is enough to produce a perfect 2D-ICSS map. The default value in **py.Aroma** is 0.2, you can press enter directly to use 0.2. Negative grid value is allowed but its absolute value would be used. 0 is not allowed, if user inputting is 0, the default value 0.2 would be used.

```
Please specify the grid quality (value smaller than 0.25 is recommended):
(press Enter to use default value 0.2)
(ENTER)
A grid quality of 0.2 will be used.
```



6) A new input file for ICSS calculation would be generated in the same dictionary as original input file, named with "*xxx_ICSS_plane_altitude.gjf*". In this example, the 2D-ICSS input file would be looks like figure (b).

7) *Gaussian* has a limitation of number of atoms (8000), if the 2D-ICSS input file includes more than 8000 atoms, a warning would be displayed before termination of **py.Aroma**.

```
                    ------ W A R N I N G ------
 Total number of atoms (1212213) in this input file exceed the limitation
of Gaussian (8000), the calculation could not be normal terminated. Please
try: (1) separate into more input files; (2) use larger grid quality
value; or (3) decrease the plotting region.
```

3.4 Create Input Files for 3D-ICSS Calculations

1) Choose function `3 – 3D-ICSS map` to load 3D-ICSS module.

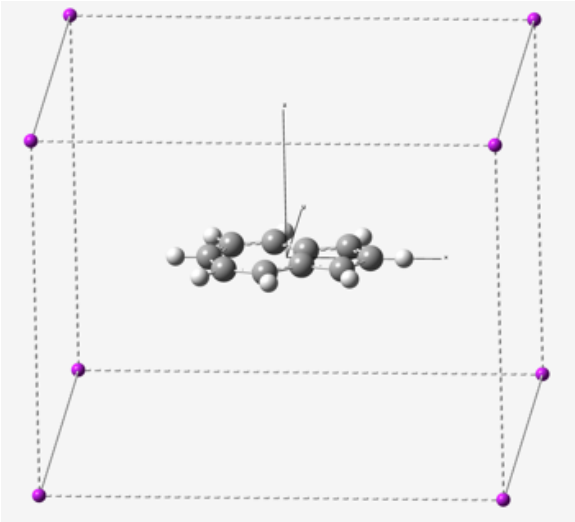
```
Please select calculation type:
  1 - NICS(n)
  2 - 2D-ICSS map
  3 - 3D-ICSS map
Input menu number and press ENTER: 3
```

2) Specify the calculation region. The region is defined in the range of X, Y, and Z axis.

```
Please specify the range of X axis (in angstrom, e.g., -10 10):
-7.5 7.5
```

```
Please specify the range of Y axis (in angstrom, e.g., -8 8):
-6.5 6.5

Please specify the range of Z axis (in angstrom, e.g., -8 8):
-6 6

3D-ICSS map in [X: -7.5 to 7.5, Y: -6.5 to 6.5, Z: -6.0 to 6.0].
```

In this example, the ghost atoms would be added in the cuboid, with X in [–7.5, 7.5], Y in [–6.5, 6.5] and Z in [–6, 6], as shown in the following figure.



3) Specify the grid quality of 3D-ICSS calculation. The default grid value is 0.2, you can press ENTER key to use the default value.

```
Please specify the grid quality (value smaller than 0.25 is recommended):
(press Enter to use default value 0.2)
0.25
ICSSgen3D will use grid quality of 0.25.
```

4) The **py.Aroma** will create several input files for ICSS calculations, name as "*xxx_3DICSS_0001.gjf*", "*xxx_3DICSS_0002.gjf*", "*xxx_3DICSS_0003.gjf*", ... Maximum Bq atom number is 7000 in one file.

```
/Users/wangzhe/Desktop/methylazulene_3DICSS_0001.gjf has been saved.
/Users/wangzhe/Desktop/methylazulene_3DICSS_0002.gjf has been saved.
/Users/wangzhe/Desktop/methylazulene_3DICSS_0003.gjf has been saved.
...
**************************************************************************
```

```
                    Input files are successfully generated.

                      Normal termination of py.Aroma.

*****************************************************************************
```

5) Please submit these input files to *Gaussian* calculation. If you are using Bash Shell on your *Gaussian* machine, you can try author's Bash Shell script **RunGJF** (https://github.com/wongzit/minorScripts), a script for submitting *Gaussian* jobs automatically.

3.5 Calculate HOMA Values

1) Choose function `1 – HOMA` to load HOMA module.

```
            Please select calculation type:
              1 - HOMA
              2 - 2D-ICSS map
              3 - 3D-ICSS map
            Input menu number and press ENTER: 1
```

2) Specify the atom number of cyclic structures. An example of [4]CPP is presented in following, the atom number (label) is marked on carbon atoms, hydrogen atoms are omitted.

Input the atom numbers in the bond-connecting order of cyclic structure, the numbers should be separated by space. For example, for the ring C1-C4-C5-C2-C6-C3, please input "1 4 5 2 6 3", or "4 5 2 6 3 1", or "6 2 5 4 1 3" ..., rather than "1 2 3 4 5 6". By press ENTER key, the HOMA value would be displayed on the command window. You can calculate HOMA for other cyclic ring by input the atom number, or input "q" to terminate the **py.Aroma**.

```
            Please input the atom numbers, separated by space:
```

```
(e.g.: 1 2 3 4 5 6)
1 4 5 2 6 3

HOMA value of ring [1 4 5 2 6 3] is 0.9842.

Input atom numbers to calculate for other ring, or input 'q' to quit.
35 32 36 33 31 34

HOMA value of ring [35 32 36 33 31 34] is 0.9842.

Input atom numbers to calculate for other ring, or input 'q' to quit.
q
```

3.6 Process 2D-ICSS Output

1) Choose function `2 – 2D-ICSS map` to load 2D-ICSS module.

```
Please select calculation type:
  1 - HOMA
  2 - 2D-ICSS map
  3 - 3D-ICSS map
Input menu number and press ENTER: 2
```

2) Choose which component will be used for 2D-ICSS map. Please input the number of the component, and press ENTER.

```
Choose shielding tensor for ICSS map:
     1 - Isoptropic      2 - Anisotropy
     3 - XX component    4 - YX component    5 - ZX component
     6 - XY component    7 - YY component    8 - ZY component
     9 - XZ component   10 - YZ component   11 - ZZ component
Please input the No.: 11
```

3) A *.csv* file including shielding tensor data would be generated in the same dictionary as the *Gaussian* output file, named with "*xxx_output_component.csv*", along with a *.png* image in same name.

3.7 Process 3D-ICSS Output

0) **py.Aroma** could detect the 3D-ICSS output if "*_3DICSS_*" is included in the file name. If you have several

output files, you can just specify any one of those files to **py.Aroma**. For example:

```
/Users/wangzhe/Desktop/methylazulene_3DICSS_0016.gjf
```

1) Choose function `3 – 3D-ICSS map` to load 3D-ICSS module.

```
Please select calculation type:
  1 - HOMA
  2 - 2D-ICSS map
  3 - 3D-ICSS map
Input menu number and press ENTER: 3
```

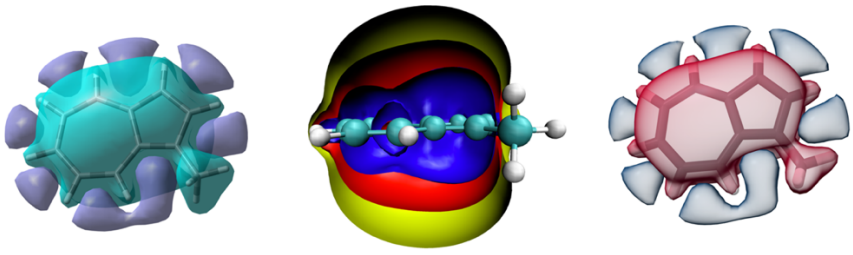
2) **py.Aroma** will load all of the output files in the inputted dictionary, it may take some time to load all files.

```
py.Aroma found 23 output files.

Please wait...
py.Aroma is extracting magnetic shielding tensor from the output files...

Processing /Users/tetsu/Desktop/example/methylazulene_3DICSS_0001.log...
Processing /Users/tetsu/Desktop/example/methylazulene_3DICSS_0002.log...
…
Processing finished!
```

3) Choose the component of 3D-ICSS plot. User can choose from isotropic, anisotropic, and 9 $X/Y/Z$ components of magnetic shielding tensors from the menu.

```
Choose shielding tensor for 3D-ICSS map:
     1 - Isoptropic      2 - Anisotropy
     3 - XX component    4 - YX component    5 - ZX component
     6 - XY component    7 - YY component    8 - ZY component
     9 - XZ component   10 - YZ component   11 - ZZ component
Please input the No.: 11
```

4) The *.cub* file including ICSS tensors would be generated in the same dictionary as the output files. You can visualize it with *GaussView*, *VMD*, *ChimeraX*, etc.



GaussView
iso = 6.56

VMD
iso = –20, –10, –5

ChimeraX
iso = 6.56

LICENSE

py.Aroma is following MIT license. The LICENSE file could be found along with py.Aroma source code.

Bug Report

I am willing to fix the bugs in **py.Aroma**. If you have found any bugs, it would be helpful if you can report the bug to me by sending an e-mail to wongzit@yahoo.co.jp or zit.wong@aol.com. When report a bug, please include following information as possible as you can.

1) Platform (OS, version).

2) Is the bug(s) reproducible?

3) (If possible) please attach the *Gaussian* file(s).

4) Detail description about the bug.

5) Any other information you want to let me know.