Groupy 3.0 Manual

Groupy 3.0 Manual

1 Overview

1.1 Background

1.1.1 Group contribution method

1.1.2 SMILES

1.2 Dependency

1.3 Architecture of Groupy code

1.4 Install

2 Quick start

2.1 Input and input file

2.1.1 Single molecule

2.1.2 Molecule file

2.1.2.1 txt

2.1.2.2 csv

2.1.2.3 xlsx

2.2 Run Groupy

2.2.1 Groupy as a standalone program

2.2.1.1 Exit (q)

2.2.1.2 Visualizing molecules (0)

2.2.1.2.1 Visualizing a molecule based on its SMILES

2.2.1.2.2 Visualizing molecules based on a file

2.2.1.3 Calculating properties of a molecule (1)

2.2.1.4 Counting group numbers of a molecule (2)

2.2.1.5 Calculating properties of a batch of molecules (3)

2.2.1.6 Calculating properties of a batch of molecules with MPI acceleration (-3)

2.2.1.7 Counting number of different groups of a batch of molecules (4)

2.2.1.8 Counting number of different groups of a batch of molecules with MPI acceleration (-4)

2.2.1.9 File-related operations (5)

2.2.1.9.1 Converting SMILES to xyz file (5_1)

2.2.1.9.2 Converting a batch of SMILES to xyz files (5_2)

2.2.1.9.3 Converting a batch of SMILES to xyz files with MPI acceleration (5_-2)

2.2.1.9.4 Converting file format (5_3)

2.2.1.9.5 Converting format of a batch of files (5_4)

2.2.1.9.6 Converting format of a batch of files with MPI acceleration (5_-4)

2.2.1.9.7 Converting a file to SMILES (5_5)

2.2.1.9.8 Converting a batch of files to SMILES (5_6)

2.2.1.9.9 Converting a file to SMILES with MPI acceleration (5_-6)

2.2.1.9.10 Generating .gjf (input file of Gaussian) file by input SMILES of a molecule (5_7)

2.2.1.9.11 Generating a batch of .gjf files (5_8)

2.2.1.8.12 Generating a batch of .gjf files with MPI acceleration (5_-8)

2.2.2 Groupy as an external Python library

Calculating properties of a molecule

1 Overview

Groupy is a program for calculating various molecular properties and preparing input files of molecular simulation software such as Gaussian. This program requires only SMILES as input, but can output many new useful data and files in multiple formats. The output information is clear and easy to read. The tips to the users are very detailed and easy to follow when using. Message passing interface (MPI) parallelization is supported to reduce computing time when the properties of a large number of molecules are calculated. Groupy not only supports the calculation of molecular properties using the traditional group contribution method, but also directly outputs the group-contribution-style molecular fingerprints for machine learning. The code has strong extensibility, which can be used as an external library to build other programs. We hope that Groupy brings great convenience to both computational and experimental chemists in their daily research. The code of Groupy can be freely obtained at https://github.com/47-5/Groupy

1.1 Background

This section introduces the physical and chemical background and computer background for the development of this program.

1.1.1 Group contribution method

Since Macleod introduced the group contribution method for calculating the molar volume of liquids in 1923, the group contribution method has undergone rapid development over the past six decades. This method is a highly accurate approach for estimating the physicochemical properties of compounds. It offers advantages such as simplicity in prediction processes, wide applicability, and good generality. Since its proposal in the mid-20th century, this method has been widely utilized for estimating the physical properties of pure substances. It is also employed in equilibrium calculations between different phases, particularly in estimating equilibria between gas and liquid phases. The group contribution method is a means for predicting phase equilibrium, enabling not only the estimation and prediction of the physical and thermodynamic properties of pure substances but also the prediction of the thermodynamic properties of mixtures. Presently, the group contribution method can be used to predict various physical properties of pure substances (such as critical parameters, molar volumes, refractive indices, thermal conductivities, viscosities, molar volumes) as well as various thermodynamic properties of pure substances (including heat capacities, heat of vaporization, saturated vapor pressures, standard enthalpies of formation), and even the thermodynamic properties of mixtures (such as activity coefficients).

The fundamental assumption of the group contribution method is that the physical properties of a pure compound or mixture are equivalent to the sum of the contributions of various groups that constitute the compound or mixture. In other words, it assumes that the contribution of the same group to physical properties is consistent across different systems. The key advantage of the group contribution method lies in its high level of generality. While the number of molecules in the material world is vast and challenging to count, the number of common organic groups that constitute organic compounds is relatively small, typically in the hundreds. Therefore, by leveraging existing experimental values to estimate the contributions of different groups to various physical properties, it is possible to predict the properties of other organic compounds.

The Groupy implements the third-order group contribution method proposed in "Group-contribution based estimation of pure component properties," incorporating additional groups and parameters introduced in "Group-contribution+ (GC+) based estimation of properties of pure components: Improved property estimation and uncertainty analysis."

1.1.2 SMILES

SMILES (Simplified Molecular Input Line Entry System) is a line notation (a typographical method using printable characters) for entering and representing molecules and reactions, which was designed by Daylight. SMILES can succinctly represent the topology of a given molecule, while the hydrogen atoms in the molecule are omitted. SMILES, as a system that can fully describe molecular topology, is widely used in the establishment of databases and the training of generative deep learning models.

Below are some rules for SMILES notation:

1. Atoms are represented by their atomic symbols: C for carbon, O for oxygen, N for nitrogen, etc.
2. Hydrogen atoms are often omitted, and implicit hydrogen atoms are assumed based on the valency of the atom.
3. Single bonds are assumed between atoms unless specified otherwise.
4. Rings are indicated by adding a number after an atom to show closure in a cyclic structure.
5. Branches are enclosed in parentheses.
6. Aromatic rings are represented by lowercase letters (e.g., c for benzene).
7. Double and triple bonds are specified by using '=' and '#' symbols, respectively.
8. Chirality can be denoted using '@' symbols.
9. Isotopes can be indicated by adding a mass number before the atomic symbol (e.g., [13C] for carbon-13).
10. Aromaticity in a ring can be represented using lowercase letters or by explicitly using the aromatic bond symbol ':'.

These rules provide a basic framework for encoding molecular structures using SMILES notation.

1.2 Dependency

Groupy relies on the following python packages：

- python >= 3.6
- tqdm
- numpy
- pandas
- ase
- rdkit
- openbabel
- joblib

The older or newer versions of the aforementioned dependent packages are unlikely to pose significant issues; however, further extensive testing has not been conducted. Testing has been performed on Windows 11, Ubuntu 18.04, and CentOS 7.8, where no anomalies were detected.

1.3 Architecture of Groupy code

Groupy consists of 6 modules, namely Loader, Counter, Calculator, Convertor, Viewer and Generator. The architecture is illustrated in Figure 1, which overviews the modules in Groupy. Loader is responsible for loading internal data of Groupy, i.e. model parameters and other hyperparameters of the group contribution method. Counter is responsible for counting the number of different types of groups in a given molecule, and can also output a group-contribution style molecular fingerprint that can be used for machine learning. Calculator receives the results of Loader and Counter to calculate different properties of the molecule. Convertor implements the conversion of SMILES to common structural files, such as gro, xyz, POSCAR, as well as the conversion of one format to another among the common chemical files. Viewer provides an interface with ASE to visualize three-dimensional molecular structures. Generator can generate input files of commonly used computational chemistry software according to the molecular structure, such as single point energy calculation, geometric optimization, and frequency analysis of Gaussian. The computational chemistry software supported by Generator will be constantly updated.

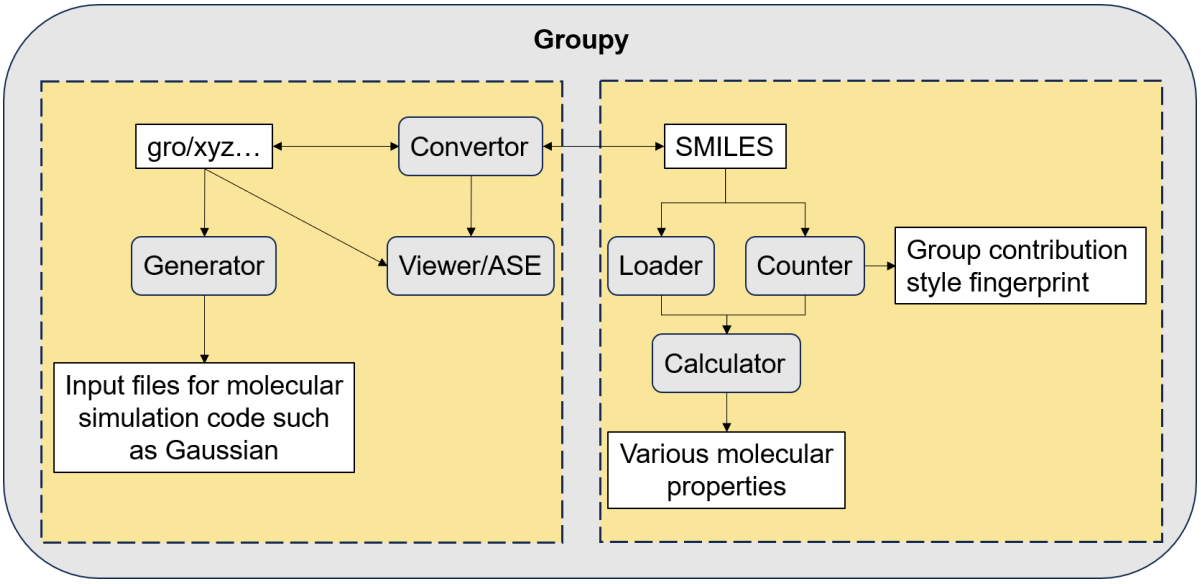

Figure1. Architecture and the modular map of Groupy code. The arrows indicate the direction of the data, the rounded gray boxes represent the module, and the white boxes represent the input or output data.

1.4 Install

Download the source code:

```
git clone https://github.com/47-5/Groupy.git
```

One may create an environment using Anaconda:

```
conda create -n groupy_env python=3.10
```

```
conda activate groupy_env
```

Install:

```
pip install .\Groupy\dist\groupy-3.0.0.tar.gz
```

`conda install -c conda-forge openbabel` (**Do not** use `pip install openbabel`)

Then one can enter `Groupy` in terminal to start the program.

2 Quick start

This section describes how to use the program to calculate the physical and chemical properties of a molecule and to count the number of groups of a given molecule, as well as some other main functions.

2.1 Input and input file

Groupy supports two main modes of operation, one computes a single molecule and the other one computes all molecules recorded in a given file in batches.

2.1.1 Single molecule

When computing properties of an individual molecule, one can directly provide its SMILES string. For instance, the SMILES of cyclohexane is `C1CCCCC1`.

2.1.2 Molecule file

A file recording the SMILES strings of some molecules should be provided when one wants to calculate properties of a batch of molecules.

Its format can be written as follows:

2.1.2.1 txt

```
CCCCCCCC
CCCC(C)C
CCC(C)(C)C
C=CCCCC
C/C=C/CCC
C=C(C)CC
CC=C(C)C
CC(C)=C(C)C
C=C=CC
C=C=C(C)C
```

There should be only one molecule's SMILES per line (no space), and no blank line.

2.1.2.2 csv

```
index,smiles,molar_mass,
0,CC,30.069999999999993,
1,C1CC1,42.08100000000002,
2,CCC,44.09700000000002,
3,C1CCC1,56.10800000000002,
```

The format is free, as long as there is a column named `smiles`, the rest of the data will be ignored, but there should be no blank line.

2.1.2.3 xlsx

index	smiles	molar_mass
0	CC	30.07

index	smiles	molar_mass
1	C1CC1	42.081
2	CCC	44.097
3	C1CCC1	56.108
4	CC1CC1	56.108

The format is free, as long as there is a column named `smiles`, the rest of the data will be ignored, but there should be no blank line.

2.2 Run Groupy

When utilizing this program, users can employ Groupy as a standalone application. Additionally, to retain the maximum extensibility of Python itself, users can import this program as an external library into Python scripts those they create by themselves.

The two distinct ways of utilizing Groupy are outlined below.

2.2.1 Groupy as a standalone program

Note: The system we used for writing this manual is Windows 11, and the terminal is Anaconda Powershell Prompt provided by Anaconda, which supports some common commands in Linux and supports both Linux and Windows path formats. Please distinguish the file path formats of different systems when using it!

After the installation is complete according to the instructions in Section 1.4, enter `Groupy` in the terminal to start Groupy. The user will see the main interface as shown in Figure 2.1.

```
(groupy_env) PS C:\Users\tjulrc\Desktop\my_test> Groupy

--------------------------------------------------------------------------------
Groupy -- A Useful Tool for Molecular Analysis
Developer: Ruichen Liu
Hint: Please feel easy to contact the developer if you have any problems in use.
E-mail1: liuruichen@tju.edu.cn
E-mail2: 1197748182@qq.com (may reply more quickly than E-mail1)
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------
You are in main interface
what to do?
 q. exit
 0. show molecular structure by SMILES or file.
 1. calculate properties of a molecule.
 2. count group number of a molecule.
 3. calculate properties of a batch of molecules.    -3. use mpi to accelerate.
 4. count group number of a batch of molecules.      -4. use mpi to accelerate.
 5. generate files or covert file format for MD, DFT, Visualization...
--------------------------------------------------------------------------------
```

Figure 2.1 Main interface of Groupy.

The program first displays a basic information, including the program name, developer, and developer contact information (users can contact the developer if they encounter any problems during use, and the developer will provide as much help as possible within their capabilities). Then, the user's location (main interface) is displayed, and the program asks the user what

operation to perform. The user only needs to enter the corresponding serial number to command the program to perform the corresponding task.

2.2.1.1 Exit (q)

To exit the program gracefully, just enter `q` in the main interface and press `Enter` on the keyboard, as shown in Figure 2.2.

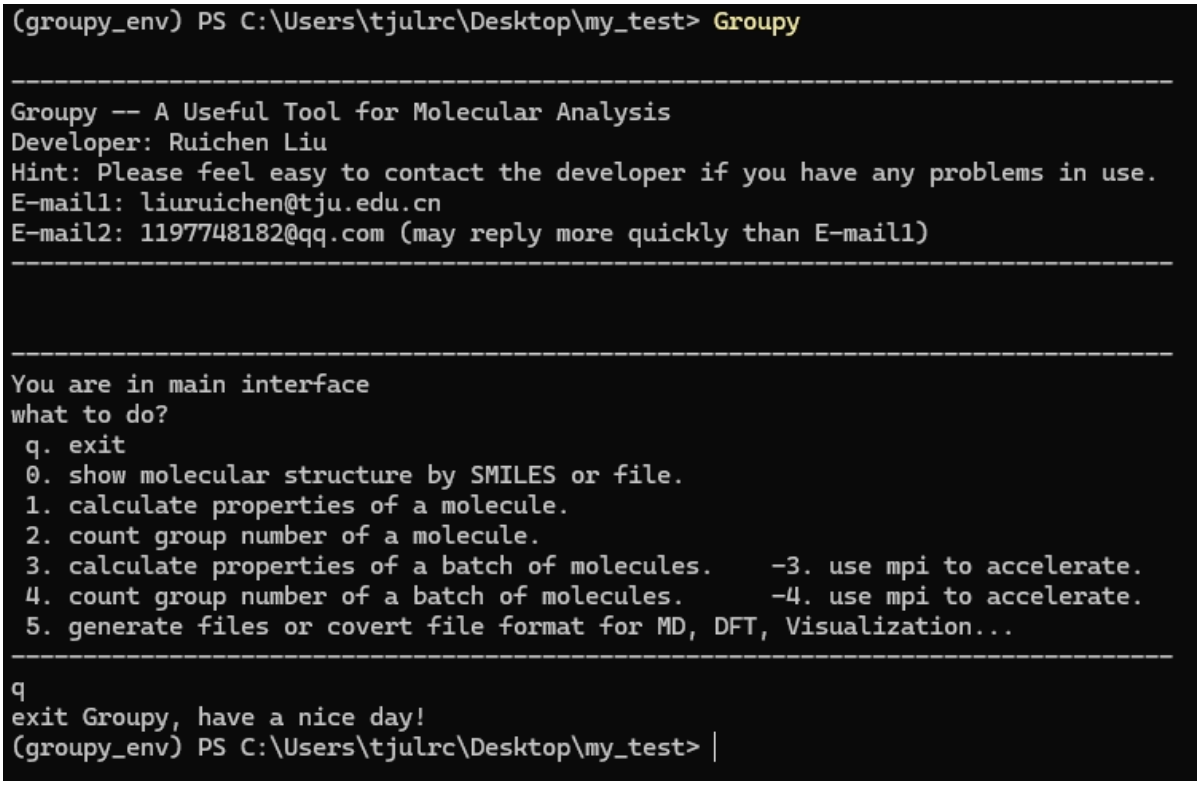

Figure 2.2 Exit of Groupy.

2.2.1.2 Visualizing molecules (0)

2.2.1.2.1 Visualizing a molecule based on its SMILES

After starting Groupy, enter `0`-`1`-`SMILES you want to Visualizing` in sequence.

```
(groupy_env) PS C:\Users\tjulrc\Desktop\my_test> Groupy

--------------------------------------------------------------------------------
Groupy -- A Useful Tool for Molecular Analysis
Developer: Ruichen Liu
Hint: Please feel easy to contact the developer if you have any problems in use.
E-mail1: liuruichen@tju.edu.cn
E-mail2: 1197748182@qq.com (may reply more quickly than E-mail1)
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------
You are in main interface
what to do?
 q. exit
 0. show molecular structure by SMILES or file.
 1. calculate properties of a molecule.
 2. count group number of a molecule.
 3. calculate properties of a batch of molecules.    -3. use mpi to accelerate.
 4. count group number of a batch of molecules.      -4. use mpi to accelerate.
 5. generate files or covert file format for MD, DFT, Visualization...
--------------------------------------------------------------------------------
0
show a SMILES (enter 1) or file (enter 2).
(enter help to show supported file formats)
1
input the SMILES of a molecule.
C1CCCC1
```

Figure 2.3 Visualizing a molecule based on its SMILES.

Then, a window showing the 3D structure of the molecule will pop up, as shown in Figure 2.4.

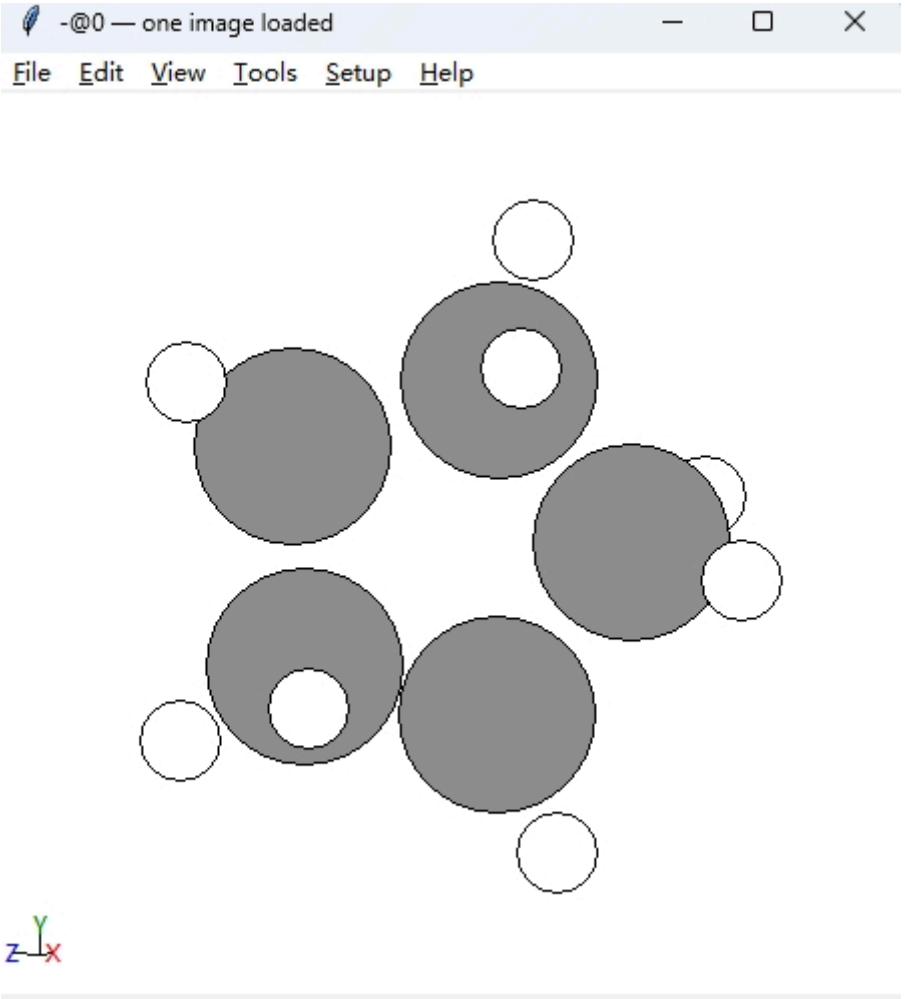

Figure 2.4 3D structure of a molecule.

2.2.1.2.2 Visualizing molecules based on a file

After starting Groupy, enter `0`-`2`-`file path you want to Visualizing`-`file format you used` in sequence, as shown in Figure 2.5.

```
--------------------------------------------------------------------------------
You are in main interface
what to do?
 q. exit
 0. show molecular structure by SMILES or file.
 1. calculate properties of a molecule.
 2. count group number of a molecule.
 3. calculate properties of a batch of molecules.    -3. use mpi to accelerate.
 4. count group number of a batch of molecules.      -4. use mpi to accelerate.
 5. generate files or covert file format for MD, DFT, Visualization...
--------------------------------------------------------------------------------
0
show a SMILES (enter 1) or file (enter 2).
(enter help to show supported file formats)
2
input the file path you want to show. e.g. ./temporary.xyz
./manual/example/main_0/C1CCCC1.xyz
input file format. e.g. xyz
xyz

--------------------------------------------------------------------------------
```

Figure 2.5 Visualizing a molecule based on a file.

2.2.1.3 Calculating properties of a molecule (1)

To calculate the physical and chemical properties of a single molecule, simply enter `1` in the main interface, then press `Enter` on the keyboard, and enter the required instructions according to the subsequent prompts, as shown in Figure 2.6.

```
--------------------------------------------------------------------------------
You are in main interface
what to do?
 q. exit
 0. show molecular structure by SMILES or file.
 1. calculate properties of a molecule.
 2. count group number of a molecule.
 3. calculate properties of a batch of molecules.    -3. use mpi to accelerate.
 4. count group number of a batch of molecules.      -4. use mpi to accelerate.
 5. generate files or covert file format for MD, DFT, Visualization...
--------------------------------------------------------------------------------
1
input the SMILES of a molecule.
C1CCCC1
{'smiles': 'C1CCCC1', 'molar_mass': 70.13499999999998, 'flash_point/K': 246.185, 'Tm/K': 133.463, 'Tb/K': 308.65, 'Tc/K'
: 526.651, 'Pc/bar': 42.659, 'Vc/(cm3/mol)': 260.02, 'density/(g/cm3)': 0.731, 'delta_G/(KJ/mol)': 44.883, 'delta_Hf/(KJ
/mol)': -78.023, 'delta_Hvap/(KJ/mol)': 28.841, 'delta_Hfus/(KJ/mol)': 3.232, 'molar_volume/(cm3/mol)(default298K)': 0.0
96, 'delta_Hc/(KJ/mol)': 3108.677, 'mass_calorific_value_h/(MJ/kg)': 44.324, 'ISP': 340.611, 'note': 'C1CCCC1 at 298K'}
Do you want to export results to a csv file? (y/n)
y
the results have been export to C1CCCC1_calculate.csv!
```

Figure 2.6. Main function 1 of Groupy.

2.2.1.4 Counting group numbers of a molecule (2)

To count the number of groups in a single molecule, simply enter `2` in the main interface, then press `Enter` on the keyboard, and enter the required instructions according to the subsequent prompts, as shown in Figure 2.7.

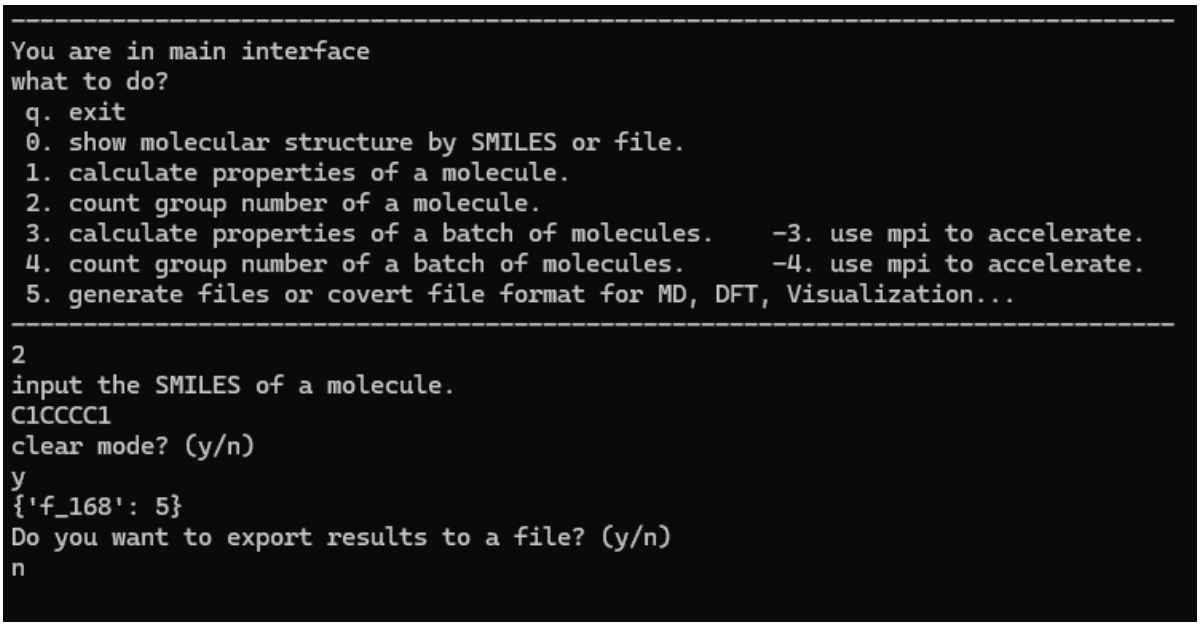

Figure 2.7. Main function 2 of Groupy, output results using the clear mode.

Groupy will ask the user whether to use clear mode. The clear mode means that only the results of groups whose number is not zero will be output, while those groups whose number is zero will not be output. When the user wants to output the statistical results in clear mode, type `y` and press `Enter`, the program will start to count and then output the statistical results to the screen. After the statistics are completed, the program will ask the user whether to export the results as a `.csv` file. If the user does want to export, type `y` and press `Enter`, and a file named `{SMILES of the currently calculated molecule}_count.csv` will be generated in the main directory of the program. If the user does not need to output the results as a file, type `n` and press `Enter`.

If the user wishes to output all statistical results, the clean mode should not be used (this is common when one wishes to obtain a group contribution style molecular fingerprint). Then, the user's input and the program's output are shown in Figure 2.8.

```
--------------------------------------------------------------------------------
You are in main interface
what to do?
 q. exit
 0. show molecular structure by SMILES or file.
 1. calculate properties of a molecule.
 2. count group number of a molecule.
 3. calculate properties of a batch of molecules.    -3. use mpi to accelerate.
 4. count group number of a batch of molecules.      -4. use mpi to accelerate.
 5. generate files or covert file format for MD, DFT, Visualization...
--------------------------------------------------------------------------------
2
input the SMILES of a molecule.
C1CCCC1
clear mode? (y/n)
n
{'f_001': 0, 'f_002': 0, 'f_003': 0, 'f_004': 0, 'f_005': 0, 'f_006': 0, 'f_007': 0, 'f_008': 0, 'f_009': 0, 'f_010': 0,
 'f_011': 0, 'f_012': 0, 'f_013': 0, 'f_014': 0, 'f_015': 0, 'f_016': 0, 'f_017': 0, 'f_018': 0, 'f_019': 0, 'f_020': 0,
 'f_021': 0, 'f_022': 0, 'f_023': 0, 'f_024': 0, 'f_025': 0, 'f_026': 0, 'f_027': 0, 'f_028': 0, 'f_029': 0, 'f_030': 0,
 'f_031': 0, 'f_032': 0, 'f_033': 0, 'f_034': 0, 'f_035': 0, 'f_036': 0, 'f_037': 0, 'f_038': 0, 'f_039': 0, 'f_040': 0,
 'f_041': 0, 'f_042': 0, 'f_043': 0, 'f_044': 0, 'f_045': 0, 'f_046': 0, 'f_047': 0, 'f_048': 0, 'f_049': 0, 'f_050': 0,
 'f_051': 0, 'f_052': 0, 'f_053': 0, 'f_054': 0, 'f_055': 0, 'f_056': 0, 'f_057': 0, 'f_058': 0, 'f_059': 0, 'f_060': 0,
 'f_061': 0, 'f_062': 0, 'f_063': 0, 'f_064': 0, 'f_065': 0, 'f_066': 0, 'f_067': 0, 'f_068': 0, 'f_069': 0, 'f_070': 0,
 'f_071': 0, 'f_072': 0, 'f_073': 0, 'f_074': 0, 'f_075': 0, 'f_076': 0, 'f_077': 0, 'f_078': 0, 'f_079': 0, 'f_080': 0,
 'f_081': 0, 'f_082': 0, 'f_083': 0, 'f_084': 0, 'f_085': 0, 'f_086': 0, 'f_087': 0, 'f_088': 0, 'f_089': 0, 'f_090': 0,
 'f_091': 0, 'f_092': 0, 'f_093': 0, 'f_094': 0, 'f_095': 0, 'f_096': 0, 'f_097': 0, 'f_098': 0, 'f_099': 0, 'f_100': 0,
 'f_101': 0, 'f_102': 0, 'f_103': 0, 'f_104': 0, 'f_105': 0, 'f_106': 0, 'f_107': 0, 'f_108': 0, 'f_109': 0, 'f_110': 0,
 'f_111': 0, 'f_112': 0, 'f_113': 0, 'f_114': 0, 'f_115': 0, 'f_116': 0, 'f_117': 0, 'f_118': 0, 'f_119': 0, 'f_120': 0,
 'f_121': 0, 'f_122': 0, 'f_123': 0, 'f_124': 0, 'f_125': 0, 'f_126': 0, 'f_127': 0, 'f_128': 0, 'f_129': 0, 'f_130': 0,
 'f_131': 0, 'f_132': 0, 'f_133': 0, 'f_134': 0, 'f_135': 0, 'f_136': 0, 'f_137': 0, 'f_138': 0, 'f_139': 0, 'f_140': 0,
 'f_141': 0, 'f_142': 0, 'f_143': 0, 'f_144': 0, 'f_145': 0, 'f_146': 0, 'f_147': 0, 'f_148': 0, 'f_149': 0, 'f_150': 0,
 'f_151': 0, 'f_152': 0, 'f_153': 0, 'f_154': 0, 'f_155': 0, 'f_156': 0, 'f_157': 0, 'f_158': 0, 'f_159': 0, 'f_160': 0,
 'f_161': 0, 'f_162': 0, 'f_163': 0, 'f_164': 0, 'f_165': 0, 'f_166': 0, 'f_167': 0, 'f_168': 5, 'f_169': 0, 'f_170': 0,
 'f_171': 0, 'f_172': 0, 'f_173': 0, 'f_174': 0, 'f_175': 0, 'f_176': 0, 'f_177': 0, 'f_178': 0, 'f_179': 0, 'f_180': 0,
 'f_181': 0, 'f_182': 0, 'f_183': 0, 'f_184': 0, 'f_185': 0, 'f_186': 0, 'f_187': 0, 'f_188': 0, 'f_189': 0, 'f_190': 0,
 'f_191': 0, 'f_192': 0, 'f_193': 0, 'f_194': 0, 'f_195': 0, 'f_196': 0, 'f_197': 0, 'f_198': 0, 'f_199': 0, 'f_200': 0,
 'f_201': 0, 'f_202': 0, 'f_203': 0, 'f_204': 0, 'f_205': 0, 'f_206': 0, 'f_207': 0, 'f_208': 0, 'f_209': 0, 'f_210': 0,
 'f_211': 0, 'f_212': 0, 'f_213': 0, 'f_214': 0, 'f_215': 0, 'f_216': 0, 'f_217': 0, 'f_218': 0, 'f_219': 0, 'f_220': 0,
 's_001': 0, 's_002': 0, 's_003': 0, 's_004': 0, 's_005': 0, 's_006': 0, 's_007': 0, 's_008': 0, 's_009': 0, 's_010': 0,
 's_011': 0, 's_012': 0, 's_013': 0, 's_014': 0, 's_015': 0, 's_016': 0, 's_017': 0, 's_018': 0, 's_019': 0, 's_020': 0,
 's_021': 0, 's_022': 0, 's_023': 0, 's_024': 0, 's_025': 0, 's_026': 0, 's_027': 0, 's_028': 0, 's_029': 0, 's_030': 0,
 's_031': 0, 's_032': 0, 's_033': 0, 's_034': 0, 's_035': 0, 's_036': 0, 's_037': 0, 's_038': 0, 's_039': 0, 's_040': 0,
 's_041': 0, 's_042': 0, 's_043': 0, 's_044': 0, 's_045': 0, 's_046': 0, 's_047': 0, 's_048': 0, 's_049': 0, 's_050': 0,
 's_051': 0, 's_052': 0, 's_053': 0, 's_054': 0, 's_055': 0, 's_056': 0, 's_057': 0, 's_058': 0, 's_059': 0, 's_060': 0,
 's_061': 0, 's_062': 0, 's_063': 0, 's_064': 0, 's_065': 0, 's_066': 0, 's_067': 0, 's_068': 0, 's_069': 0, 's_070': 0,
 's_071': 0, 's_072': 0, 's_073': 0, 's_074': 0, 's_075': 0, 's_076': 0, 's_077': 0, 's_078': 0, 's_079': 0, 's_080': 0,
 's_081': 0, 's_082': 0, 's_083': 0, 's_084': 0, 's_085': 0, 's_086': 0, 's_087': 0, 's_088': 0, 's_089': 0, 's_090': 0,
 's_091': 0, 's_092': 0, 's_093': 0, 's_094': 0, 's_095': 0, 's_096': 0, 's_097': 0, 's_098': 0, 's_099': 0, 's_100': 0,
 's_101': 0, 's_102': 0, 's_103': 0, 's_104': 0, 's_105': 0, 's_106': 0, 's_107': 0, 's_108': 0, 's_109': 0, 's_110': 0,
 's_111': 0, 's_112': 0, 's_113': 0, 's_114': 0, 's_115': 0, 's_116': 0, 's_117': 0, 's_118': 0, 's_119': 0, 's_120': 0,
 's_121': 0, 's_122': 0, 's_123': 0, 's_124': 0, 's_125': 0, 's_126': 0, 's_127': 0, 's_128': 0, 's_129': 0, 's_130': 0,
 't_001': 0, 't_002': 0, 't_003': 0, 't_004': 0, 't_005': 0, 't_006': 0, 't_007': 0, 't_008': 0, 't_009': 0, 't_010': 0,
 't_011': 0, 't_012': 0, 't_013': 0, 't_014': 0, 't_015': 0, 't_016': 0, 't_017': 0, 't_018': 0, 't_019': 0, 't_020': 0,
 't_021': 0, 't_022': 0, 't_023': 0, 't_024': 0, 't_025': 0, 't_026': 0, 't_027': 0, 't_028': 0, 't_029': 0, 't_030': 0,
 't_031': 0, 't_032': 0, 't_033': 0, 't_034': 0, 't_035': 0, 't_036': 0, 't_037': 0, 't_038': 0, 't_039': 0, 't_040': 0,
 't_041': 0, 't_042': 0, 't_043': 0, 't_044': 0, 't_045': 0, 't_046': 0, 't_047': 0, 't_048': 0, 't_049': 0, 't_050': 0,
 't_051': 0, 't_052': 0, 't_053': 0, 't_054': 0, 't_055': 0, 't_056': 0, 't_057': 0, 't_058': 0, 't_059': 0, 't_060': 0,
 't_061': 0, 't_062': 0, 't_063': 0, 't_064': 0, 't_065': 0, 't_066': 0, 't_067': 0, 't_068': 0, 't_069': 0, 't_070': 0,
 't_071': 0, 't_072': 0, 't_073': 0, 't_074': 0}
Do you want to export results to a file? (y/n)
y
the results have been export to C1CCCC1_count.csv!
```

Figure 2.8. Main function 2 of Groupy.

2.2.1.5 Calculating properties of a batch of molecules (3)

If the user needs to calculate the physical and chemical properties of a batch of molecules, he/she needs to first prepare a molecular file introduced in Section 2.1.2, enter `3` in the main interface, and then press `Enter` on the keyboard, and enter the required instructions according to the subsequent prompts, as shown in Figure 2.9.

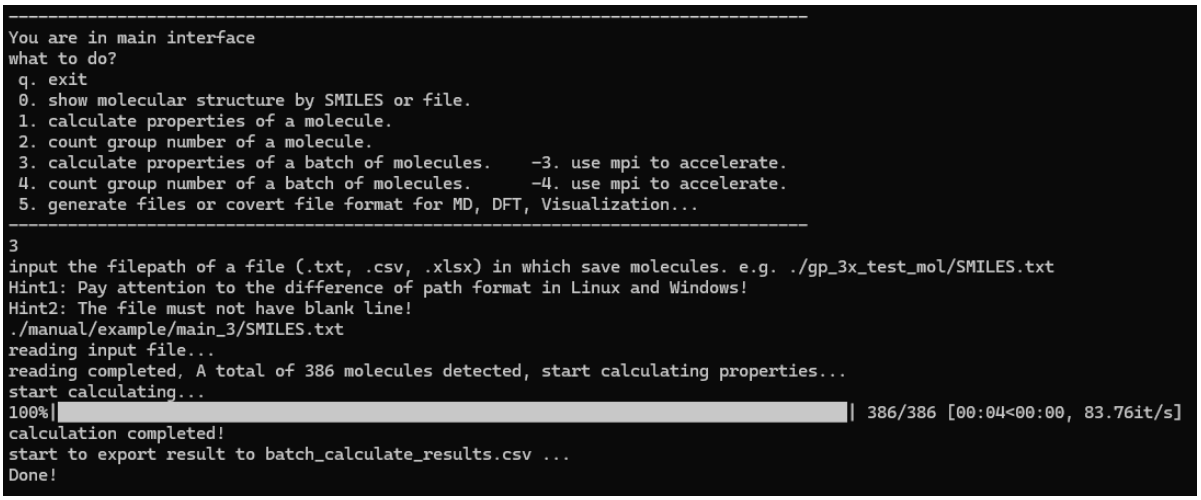

Figure 2.9. Main function 3 of Groupy.

When you enter main function 3 by typing `3` in the main interface, the program will prompt the user to enter the file path where the SMILES strings of the molecules are saved. After the user enters the file path and hits `Enter`, the program will automatically read the SMILES recorded in the molecule file and then perform the calculation. When the calculation is completed, the results will be written to `batch_results.csv` in the current working directory.

2.2.1.6 Calculating properties of a batch of molecules with MPI acceleration (-3)

In modern computers, the Central Processing Unit (CPU) typically consists of multiple cores. To fully leverage the multi-core capabilities of the CPU when computing properties for a batch of molecules, one can utilize Message Passing Interface (MPI) for parallel computing.

Similar to Section 2.2.1.5, one needs to prepare a file as introduced in Section 2.1.2. In the main interface, input `-3`, then press `Enter` on the keyboard. Then, the user follows subsequent prompts to input the desired command, as illustrated in Figure 2.10.

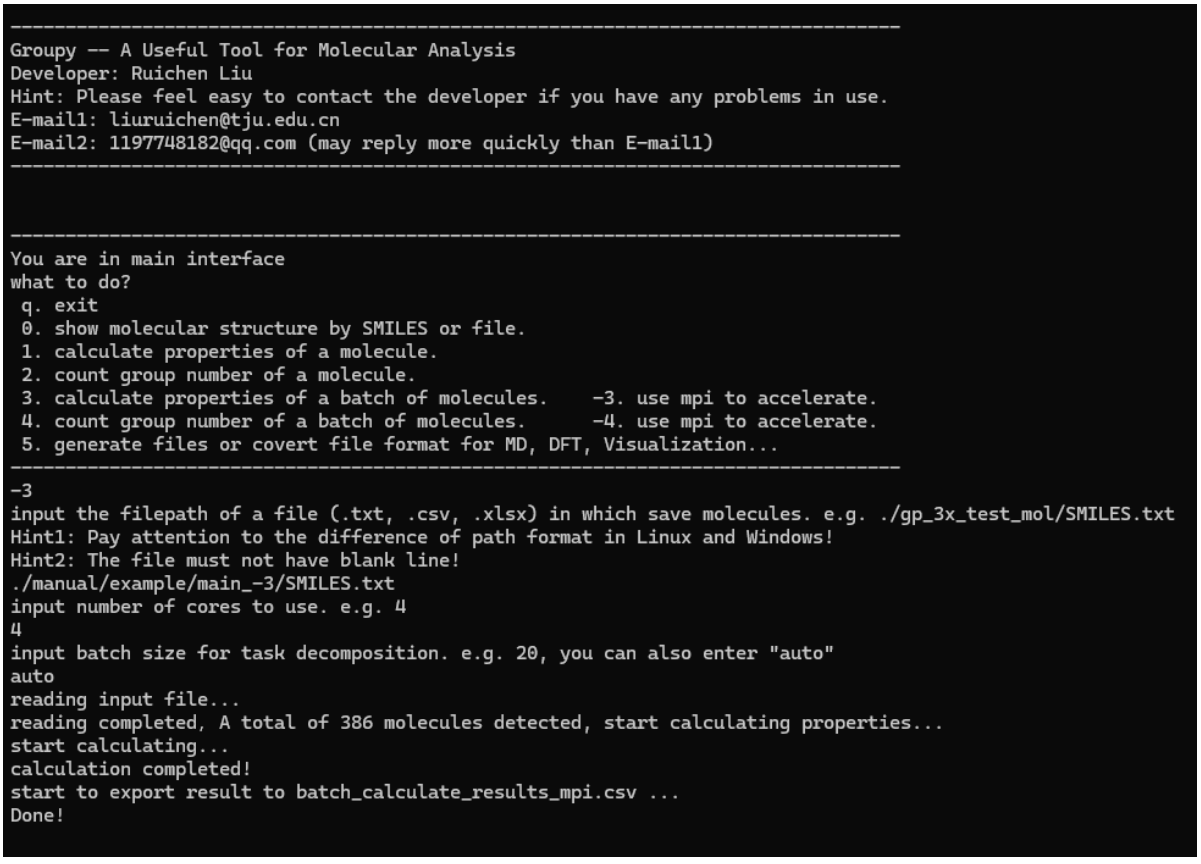

Figure 2.10. Main function -3 of Groupy.

2.2.1.7 Counting number of different groups of a batch of molecules (4)

There is almost no difference between the usage of main function 4 introduced in this section and main function 3 introduced in 2.2.1.5, as shown in Figure 2.11.

```
--------------------------------------------------------------------------------
You are in main interface
what to do?
 q. exit
 0. show molecular structure by SMILES or file.
 1. calculate properties of a molecule.
 2. count group number of a molecule.
 3. calculate properties of a batch of molecules.    -3. use mpi to accelerate.
 4. count group number of a batch of molecules.      -4. use mpi to accelerate.
 5. generate files or covert file format for MD, DFT, Visualization...
--------------------------------------------------------------------------------
4
input the filepath of a file in which save molecules. e.g. ./gp_3x_test_mol/SMILES.txt
Hint1: Pay attention to the difference of path format in Linux and Windows!
Hint2: The file must not have blank line!
./manual/example/main_4/SMILES.txt
reading the input file...
Done, totally detected 386 molecules, start counting...
100%|████████████████████████████████████████████████████████████| 386/386 [00:04<00:00, 84.44it/s]
Done!
writing to csv...
Done!
```

Figure 2.11. Main function 4 of Groupy.

2.2.1.8 Counting number of different groups of a batch of molecules with MPI acceleration (-4)

There is almost no difference between the usage of main function -4 introduced in this section and main function -3 introduced in 2.2.1.6, as shown in Figure 2.12

```
--------------------------------------------------------------------------------
You are in main interface
what to do?
 q. exit
 0. show molecular structure by SMILES or file.
 1. calculate properties of a molecule.
 2. count group number of a molecule.
 3. calculate properties of a batch of molecules.    -3. use mpi to accelerate.
 4. count group number of a batch of molecules.      -4. use mpi to accelerate.
 5. generate files or covert file format for MD, DFT, Visualization...
--------------------------------------------------------------------------------
-4
input the filepath of a file (.txt, .csv, .xlsx) in which save molecules. e.g. ./gp_3x_test_mol/SMILES.txt
Hint1: Pay attention to the difference of path format in Linux and Windows!
Hint2: The file must not have blank line!
./manual/example/main_-4/SMILES.txt
input number of cores to use. e.g. 4
4
input batch size for task decomposition. e.g. 20, you can also enter "auto"
auto
reading the input file...
Done, totally detected 386 molecules, start counting...
Done!
writing to csv...
Done!
```

Figure 2.12. Main function -4 of Groupy

2.2.1.9 File-related operations (5)

Although the group contribution method is versatile and computationally efficient, its accuracy is limited. Therefore, in addition to the group contribution method itself, this program also provides users with functionality for visualization and generating files required for molecular dynamics and quantum chemical calculations. This feature is particularly useful when initially screening with the group contribution method and subsequently refining with higher-precision methods.

2.2.1.9.1 Converting SMILES to xyz file (5_1)

The xyz file (http://sobereva.com/477) is almost the simplest file format for recording three-dimensional structure of a molecule. Almost all visualization programs can open it (such as Gaussview, VESTA, VMD, etc.).

If the users want to obtain the xyz file of a molecule by entering its SMILES string, they only need to enter `5` in the main interface, then type `Enter` on the keyboard, and further enter `1` and type `Enter`. Then, enter the required instructions according to the subsequent prompts, as shown in Figure 2.13.

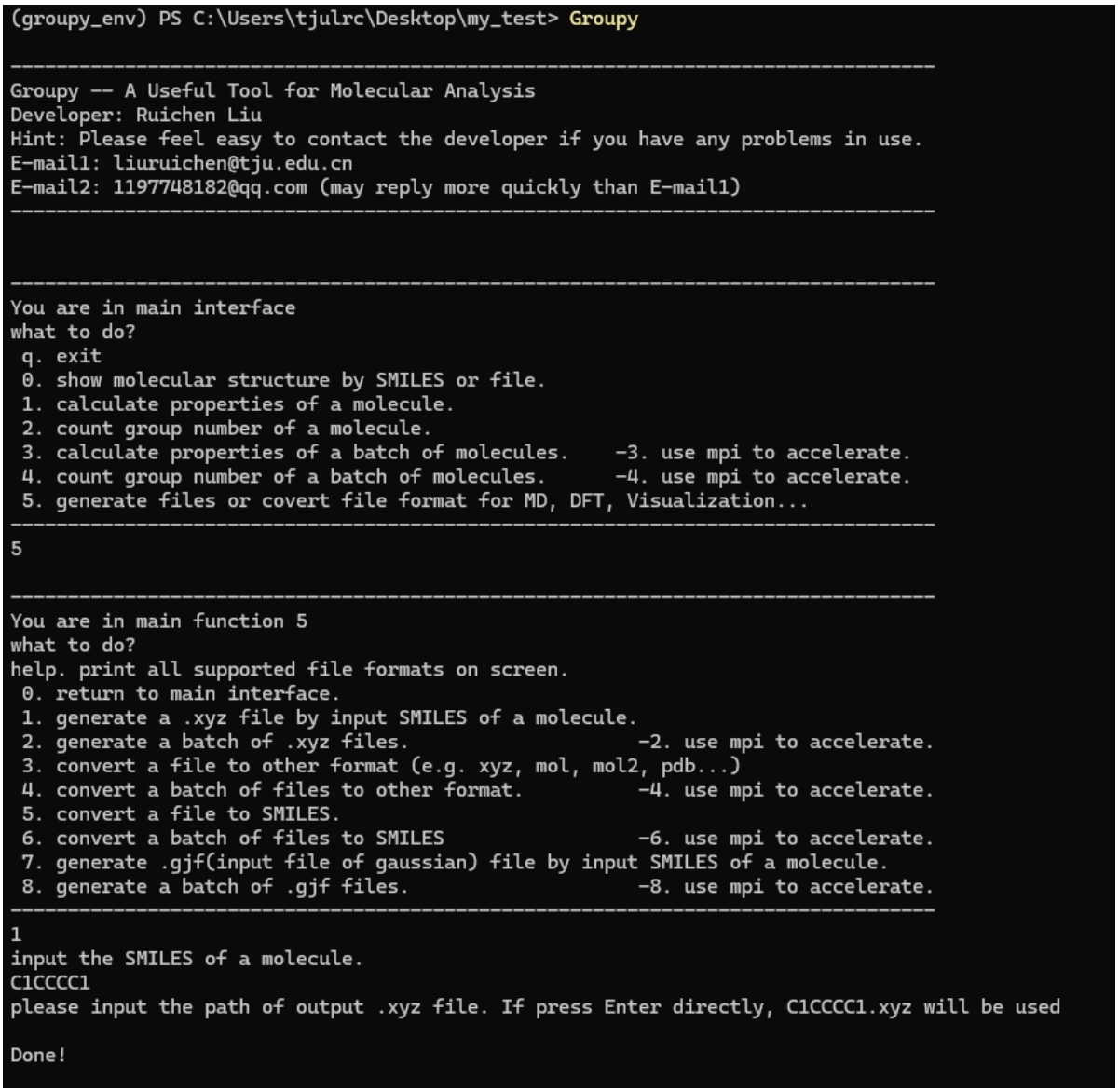

Figure 2.13 Sub-function 1 of main function 5

2.2.1.9.2 Converting a batch of SMILES to xyz files (5_2)

If the user needs to generate a batch of xyz files for molecules, he/she needs to first prepare a molecular file introduced in Section 2.1.2, enter `5` in the main interface, press `Enter` on the keyboard, then enter `2` and press `Enter`, and enter the required instructions according to the subsequent prompts, as shown in Figure 2.14.

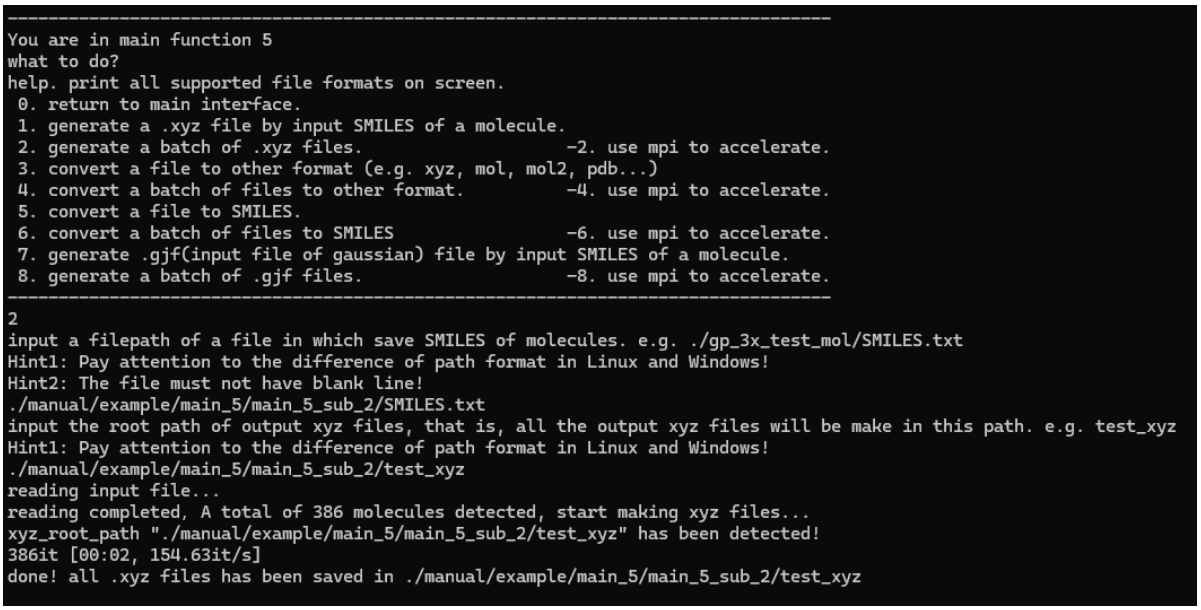

Figure 2.14 Sub-function 2 of main function 5.

2.2.1.9.3 Converting a batch of SMILES to xyz files with MPI acceleration (5_-2)

There is almost no difference between the usage of sub function -2 of main function 5 introduced in this section and main function -3 introduced in 2.2.1.6.

2.2.1.9.4 Converting file format (5_3)

This program also provides the function of converting file formats. Users only need to provide the original file format and its path to be converted, and then specify the required file format and path, as shown in Figure 2.15. **This function of this program is based on openbabel, so this function supports all file formats supported by openbabel, such as: xyz, mol, mol2, pdb...**

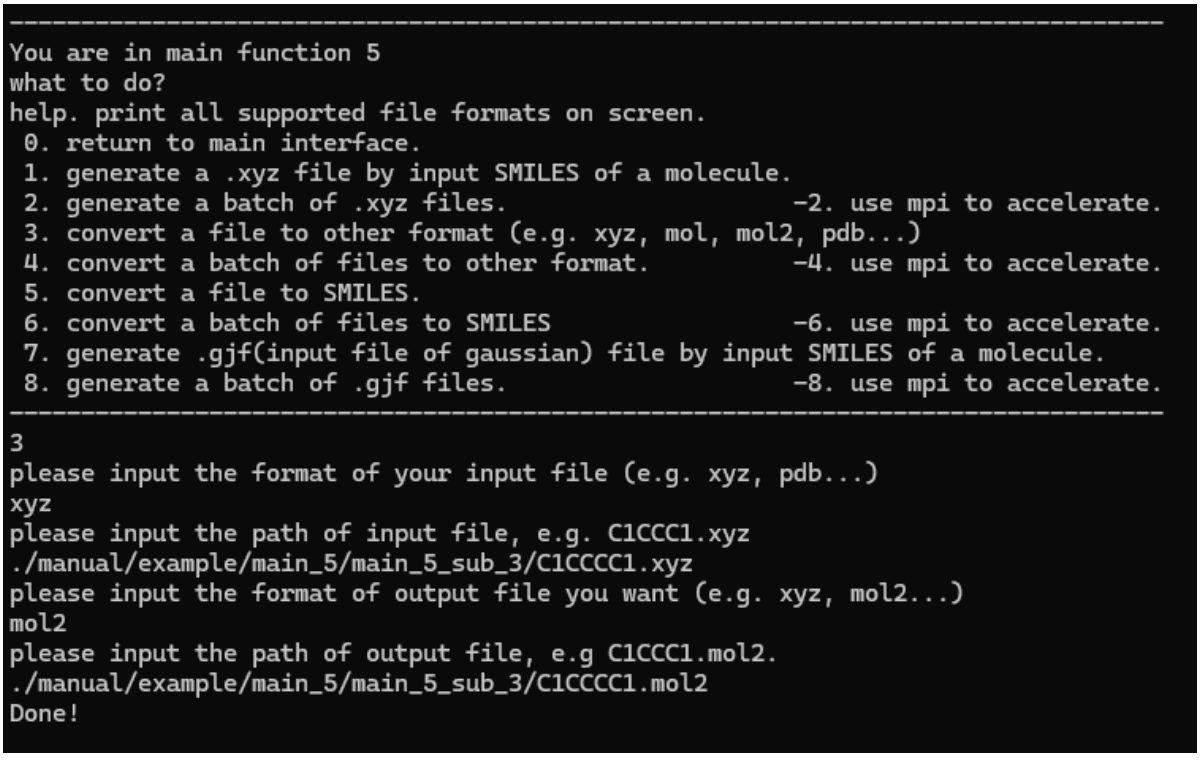

Figure 2.15 Sub-function 3 of main function 5.

2.2.1.9.5 Converting format of a batch of files (5_4)

When one needs to convert the file formats of a batch of files, he/she should first put the files to be converted into the same folder. Then, enter the sub-function `4` of the main function `5`. First, the program will ask the user what the file format is to be converted, and then one needs to enter the root directory of the file to be converted (that is, the directory where all the files you want to convert are saved). Then, the program asks the user what format he/she wants to convert the file to. After the user enters the command (such as xyz, mol, mol2, pdb, gro, etc.), the program requires the user to specify the root directory of the new file after conversion (that is, all newly generated files will be saved there). Figure 2.16 shows the specific operations.

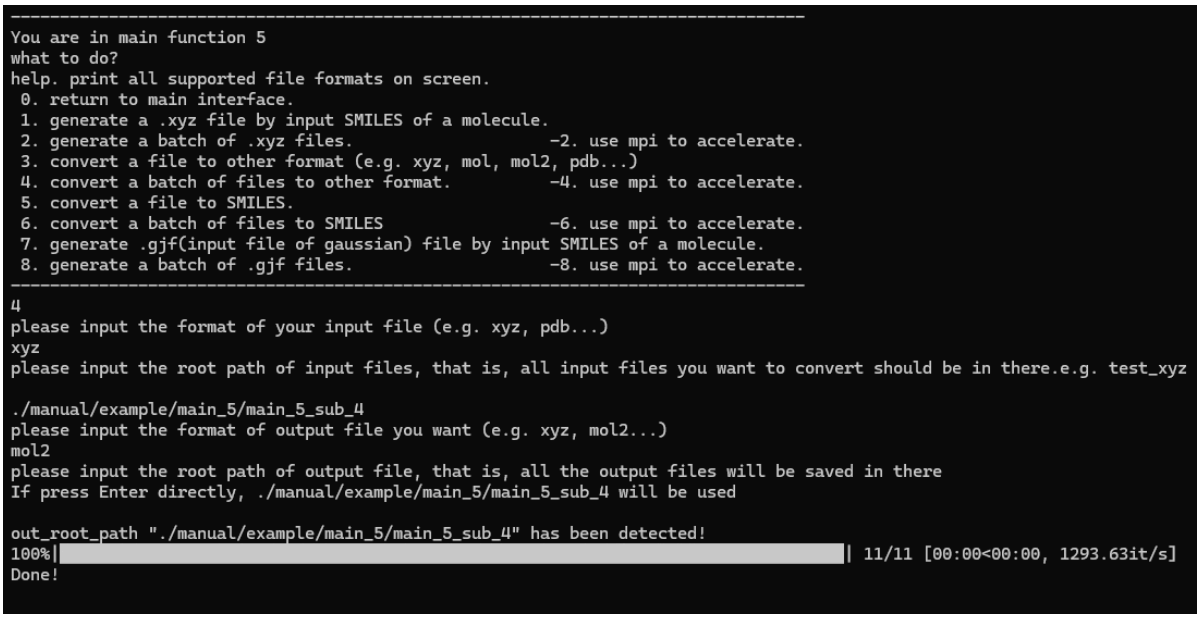

Figure 2.16 Sub-function 4 of main function 5.

2.2.1.9.6 Converting format of a batch of files with MPI acceleration (5_-4)

There is almost no difference between the usage of sub function -4 of main function 5 introduced in this section and main function -3 introduced in 2.2.1.6.

2.2.1.9.7 Converting a file to SMILES (5_5)

One can convert a file (xyz, mol2, ...) to a SMILES string, as shown in Figure 2.17. The result will be printed on the screen.

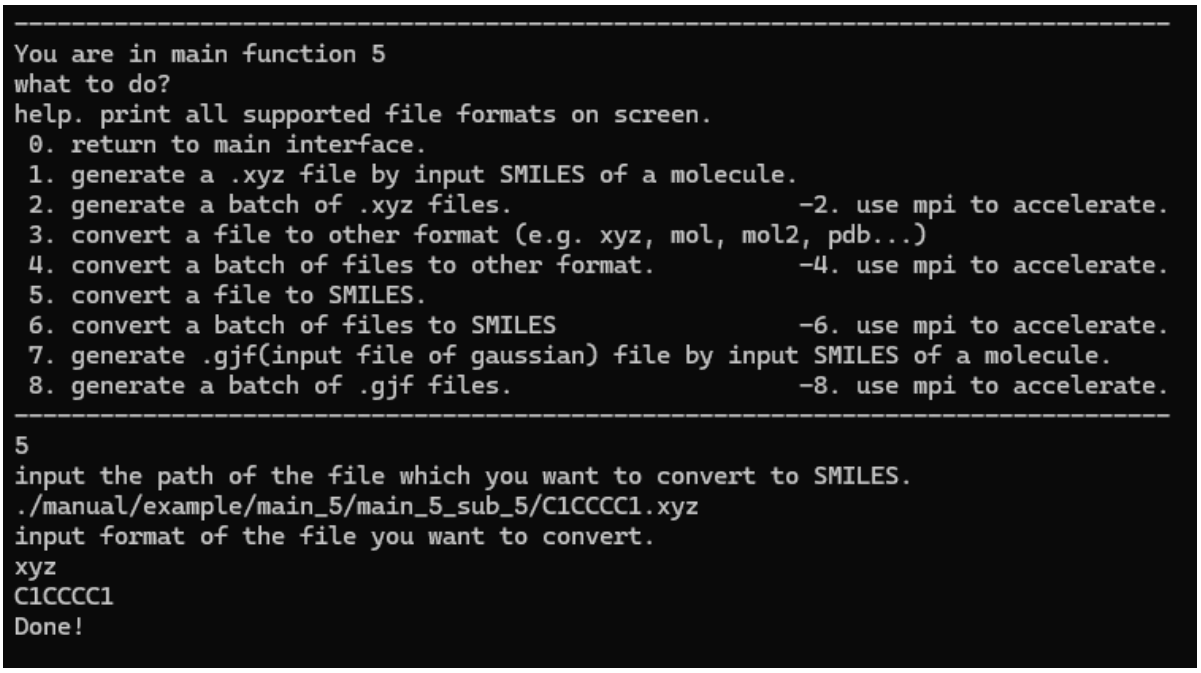

Figure 2.17 Sub-function 5 of main function 5.

2.2.1.9.8 Converting a batch of files to SMILES (5_6)

When users want to convert a batch of files to SMILES, they can use sub-function 6 of main function 5, as shown in Figure 2.18.

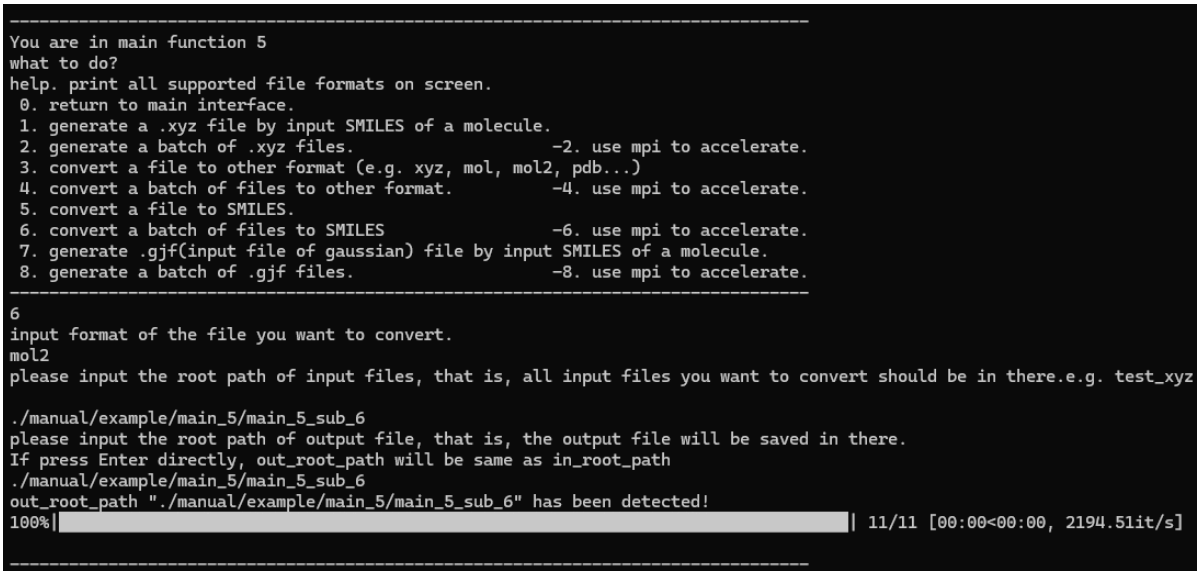

Figure 2.18 Sub-function 6 of main function 5.

2.2.1.9.9 Converting a file to SMILES with MPI acceleration (5_-6)

There is almost no difference between the usage of sub function -6 of main function 5 introduced in this section and main function -3 introduced in 2.2.1.6.

2.2.1.9.10 Generating .gjf (input file of Gaussian) file by input SMILES of a molecule (5_7)

The program also offers users the capability to generate Gaussian input files (gjf files) based on the SMILES of molecules. Users simply need to access the sub-function 7 of the main function 5 and follow the prompts accordingly, as depicted in Figure 2.19.

```
--------------------------------------------------------------------------------
You are in main function 5
what to do?
help. print all supported file formats on screen.
 0. return to main interface.
 1. generate a .xyz file by input SMILES of a molecule.
 2. generate a batch of .xyz files.                    -2. use mpi to accelerate.
 3. convert a file to other format (e.g. xyz, mol, mol2, pdb...)
 4. convert a batch of files to other format.          -4. use mpi to accelerate.
 5. convert a file to SMILES.
 6. convert a batch of files to SMILES                 -6. use mpi to accelerate.
 7. generate .gjf(input file of gaussian) file by input SMILES of a molecule.
 8. generate a batch of .gjf files.                    -8. use mpi to accelerate.
--------------------------------------------------------------------------------
7
input the SMILES of a molecule.
C1CCCC1
input the CPU cores you want to use. e.g. 12
12
input the memory you want to use. e.g. 12GB
12GB
input the path of chk file. e.g. Cc1ccccc1.chk
Hint1: If press Enter directly, C1CCCC1.chk will be used.
Hint2: Attention please! the symbol such as (, ), /, \ and # should not appear in a filepath!
C1CCCC1.chk
input the path of gjf file. e.g. Cc1ccccc1.gjf
Hint1: If press Enter directly, C1CCCC1.gjf will be used.
Hint2: Attention please! the symbol such as (, ), /, \ and # should not appear in a filepath!
./manual/example/main_5/main_5_sub_7/C1CCCC1.gjf
input the keywords of Gaussian to define task you want to run.e.g. #p opt freq b3lyp/6-31g*
Hint1: if press Enter directly, "#p opt freq b3lyp/6-31g*" will be used.
#p opt freq b3lyp/6-31g*
Input charge and multiplicity. e.g. 0 1
Hint: If press Enter directly, Groupy will automatically calculate them
Weather to add some other tasks in this .gjf (y/n).
y
Input keywords you want to add. If there are more than one other tasks, Please separate them with commas (,)
Hint: if press Enter directly, "#p m062x/def2tzvp geom=check,#p m062x/def2tzvp scrf=solvent=water geom=check" will be us
ed
#p m062x/def2tzvp geom=check
Done!
```

Figure 2.19 Sub-function 7 of main function 5.

2.2.1.9.11 Generating a batch of .gjf files (5_8)

The gjf file of molecules can also be generated in batches. Groupy asks the user to enter the path of the file that records the SMILES of a batch of molecules, and the program will automatically read the SMILES recorded in it. The program then asks the user to enter the root directory to save generated gjf file (that is, all the generated gjf files in this task will be saved there). Figure 2.20 shows detailed operations.

```
--------------------------------------------------------------------------------
You are in main function 5
what to do?
help. print all supported file formats on screen.
 0. return to main interface.
 1. generate a .xyz file by input SMILES of a molecule.
 2. generate a batch of .xyz files.                    -2. use mpi to accelerate.
 3. convert a file to other format (e.g. xyz, mol, mol2, pdb...)
 4. convert a batch of files to other format.          -4. use mpi to accelerate.
 5. convert a file to SMILES.
 6. convert a batch of files to SMILES                 -6. use mpi to accelerate.
 7. generate .gjf(input file of gaussian) file by input SMILES of a molecule.
 8. generate a batch of .gjf files.                    -8. use mpi to accelerate.
--------------------------------------------------------------------------------
8
input the filepath of a file in which save molecules. e.g. ./gp_3x_test_mol/SMILES.txt
Hint1: Pay attention to the difference of path format in Linux and Windows!
Hint2: The file must not have blank line!
./manual/example/main_5/main_5_sub_8/SMILES.txt
Input the root path of output gjf files, that is, all the output gjf files will be make in this path. e.g. test_gjf
Hint1: Pay attention to the difference of path format in Linux and Windows!
Hint2: if press Enter directly, test_gjf will be used.
./manual/example/main_5/main_5_sub_8/test_gjf
input the CPU cores you want to use. e.g. 12
12
input the memory you want to use. e.g. 12GB
12GB
input the keywords of Gaussian to define task you want to run.e.g. #p opt freq b3lyp/6-31g*
Hint1: if press Enter directly, "#p opt freq b3lyp/6-31g*" will be used.

Input charge and multiplicity. e.g. 0 1
Hint: If press Enter directly, Groupy will automatically calculate them
Weather to add some other tasks in this .gjf (y/n).
n
reading input file...
reading completed, A total of 386 molecules detected, start calculating properties...
gjf_root_path "./manual/example/main_5/main_5_sub_8/test_gjf" has not been detected, I will create it for you
386it [00:02, 137.09it/s]
done! all .gjf files has been saved in ./manual/example/main_5/main_5_sub_8/test_gjf
```

Figure 2.20 Sub-function 8 of main function 5.

2.2.1.8.12 Generating a batch of .gjf files with MPI acceleration (5_-8)

There is almost no difference between the usage of sub function -8 of main function 5 introduced in this section and main function -3 introduced in 2.2.1.6.

2.2.2 Groupy as an external Python library

The program can also be imported as an external library into user-written python scripts. We only give an example here, for more details, please refer to the API documentation in Groupy/doc

Calculating properties of a molecule

```
from groupy.gp_calculator import Calculator


c = Calculator()
result = c.calculate_a_mol('C1CCCC1')
print(result)
```