Documentation for the coordinate transformation script

Author: Bruno von Brüning (`bruno.vonbruning@student.kuleuven.be`)

Version from August 26, 2022

1 Informations

1.1 Problem

If a property of a molecule depends on the conformation of this molecule, multiple minima have to be considered. To find these minima multiple starting geometries have to be created to find all the possible minima. Depending on the molecule and the geometries to be considered , i.e. different dihedrals angles a multitude of structures may have to be genereated. Such displacement like change of dihedral will be called *operations* in the following. This can be a tedious and error-prone labor. The author provides a script that mostly automizes this process to enable you an easier and more efficient work flow. At the moment the script is limited to rotation around dihedrals but is designed to feature more operations. Please contact the author if you want to propose extension of the current script. Except the manual notation of the operations and values to be considered the user of this script will face the minimal work of executing this script and submitting calculations to his or her cluster.

1.2 Structure of the script

The main purpose of the provided code is to rotate a molecule around dihedral angles in order to obtain input structures for subsequent quantum-mechanical calculations. More transformations like changing the conformation could be implemented in the future for interest in this please contact the author (`bruno.vonbruening@student.kuleuven.be`). Next to this main feature a upstream script to permute through possible combinations of angle values is provided, the resulting combination of multiple operations is called transformation job. The script generates job file where the operations for the multiple displacement to be executed are saved (`transformation file`). Find more information about this script in section **??**.

The subsequent main script performs the transformations provided in the `transformation file` on coordinates of a molecule coordinated provided in example in a `.xyz` file. The script outputs all the resulting structures as coordinate files or input files for computational chemsitry software. At the moment only `Gaussian` input format is supported. Please contact the author if other input formats may be implemented. Find more information about this scipt in section **??**

The resulting calculations can be automatically evaluated by the `evaluate_output_script`. Resulting angles, energy and dipolemoment are read out and either written in a `.csv` or a `.txt` format. Identical resulting can conformations are indentified by the same resulting angles. Currently only `Gausssian .log` files can be read. Please contact the author if you desire other formats to be read.

1.3 Technical information and tipps

The three provided script are saved in individual folders. Modules are shared between these scripts and saved in the folder `modules`. The scripts access these modules by a relative path, hence please do not change the level of the `module` folder or also change the relative path in the scripts. A help entry for all of the following python scripts can be called by calling the respective funtion with the option `-h`.

```
./script_[...] -h
```

2 Generate the operation file

2.1 Usage

The script `script_ang-to-op.py` takes a `input` file with the definition of the possible operations and values and writes all the possible combintations of these values in an `output` file. By default the name of the output file while be name of the inputfile truncated at te extension or the occurence of `_ang` and get the ending `_op.txt` appended.

```
./script_ang-to-op.py [INPUTFILE]
```

Following further options can be chosen

Options	Purpose	example
`-ofilename`	Name for the outputfile to be generated	`my_output_file.txt`

```
!! angle alpha
204-219
ROT 3,2,204,205
pm 15

!! angle beta
214-219
EXC

[...]
```

Listing 1: Format of inputfile for operation file generation

2.2 Inputfile format

The format of the `input`-file goes in the following order:

Entry	Description	Example
Comments	comments are indicated with ”!” or ”!!”	`[command] ! [comment]`
First line	a range of atoms for example `{1,...,5,10,..13,17,..20}` where `-` indicates ranges and `,` a list	`1-5,10-13,17-20`

Second line	First a operation type (`ROT` or `FIX` for fixing atoms for the optimization (givin the `-1` as parameter)). Then separated by a space the following option are given.
	`ROT` takes four numbers for the labels of the atoms in the dihedral of interest
	`FIX` takes a list of atoms in the same format as above.
		`ROT 2,3,4,5`
		`FIX 30-40,50-60`

(Third line)	`FIX` option: blank or omitted `ROT` option: values of desired angles of dihedral as a list, prefix of `pm` if plus and minus these values are desired	For `ROT`: `pm 15,75` or `180`
Separation line	Between different entries there has to be an empty line (empty before possibly comments)	`" \n"`, `" ![!]  [...]\n"`

An arbitrary number of these objects these objects can be included in the program in arbitrary order. An example can be found in 1

2.3 Format of operations file

The ordering of the operations in the operations file will impact the filename that will be produced.

Operations are

The entries are separated by blank lines. The first and last line do not have to be a blank line. Notably lines with only comments also account as blank lines.

3 Generate the `Gaussian` input files

3.1 Purpose

The script reads the operation from the the operations file (see 2.3) and performs these operations on the coordinates provided within a molecule coordinate file. Either input files or molecule coordinate files can chosen to be generated (first need a further file with the head of the calculation as furthe input, the body file). These files will be named according t the corresponding values given in the operations file and saved within an own folder. The folder will get

3.2 Usage

The Displacment of the coordinates requires a file with input coordinates (at the moment only `.pdb` format) and the file containing the operations (See subsection 2.3). Furthermore, a body file is required if `.com` files are to be generated (find exmpale in listing 2. The basic execution and further options can be found below.

```
./script_displacement [molecule.pdb] [operations.txt]
```

Options	Purpose	example
`-oformat`	outputformat of transfomred coordinates	`XYZ`, `PDB`, `COM`(Default)
`-oname`	stem for folder and filenames to be used	
`-bfile`	headerfile, required to generate `.com` files	`body_file.txt`

```
!chkname
%mem=20GB
%nprocshared=16
# opt B3LYP/6-31g(d,p) scrf=iefpcm

!jobname

0 1
```

Listing 2: Example of a body file

3.3 `-oformat`: Output format

The format of the coordinates to be printed can be chosen with the option `-oformat`. Current accepted values are `XYZ`, `PDB`, `COM` that correspond respectively to `.xyz`, `.pdb`, and `.com` format. If the `.com` format is chosen a body file has to be provided by envoking the option `-bfile`.

3.4 `-bfile`: body file

To obtain input files for `Gaussian` a body with informations for this calculation has to be provided to which the calculated coordinates are going to be appended. An example of such a headerfile is provided in listing 2. Lines starting with `!` will be recognized to be replaced with information corresponding to the current file. The flag `!chkname` will be replaced with the specific filename. The flag `!jobname` will be replaced with with the filename without the file extension. Following is an example for such a header file.

3.5 `-oname`: Outputfile name

The stem – the first part of the names of the following files – can be determined with the option `-oname`. The default is the filename of the input file cut after the first occurence of a `.` character. In example, an input filename like `[path]/molecule.pdb` would give the default stem `molecule`. A folder for the output files is created with the name of the `stem` and appended `_{oformat}`, where the outputfiles are saved. The outputfiles themselves will be named by the *stem* and the appendices `_{values}` for all the values given for this operation in ordering of the outputfile. The file extension is the same as `omode`. For example, with the *stem* molecule and provided values 90, 180 and the chosen outut format `.xyz` will lead to the filename `molecule_90_180.xyz`. Notably, these values are casted to integers if float are given inside the operations file.

4 Utilities

4.1 chk_copy.sh script

This script transfers checkpoint files from a provided folder to the present working directory where the .com files building up on these .chk files may be saved. The name of the .com files is truncated after given regular expression and matched with the .chk files. In example, giving the script B3 as second option will let the script search for the Mol_0_90_15_B3LYP.com for a file matching Mol_0_90_15*.chk* in the folder provided in the first argument. This file is then copied to the present working directory and given the name Mol_0_90_15_B3LYP.chk

```
./chk_copy.sh [PATH_to_.chk_files] [string_to_cut_before]
```

5 Evaluate the `Gaussian` output files