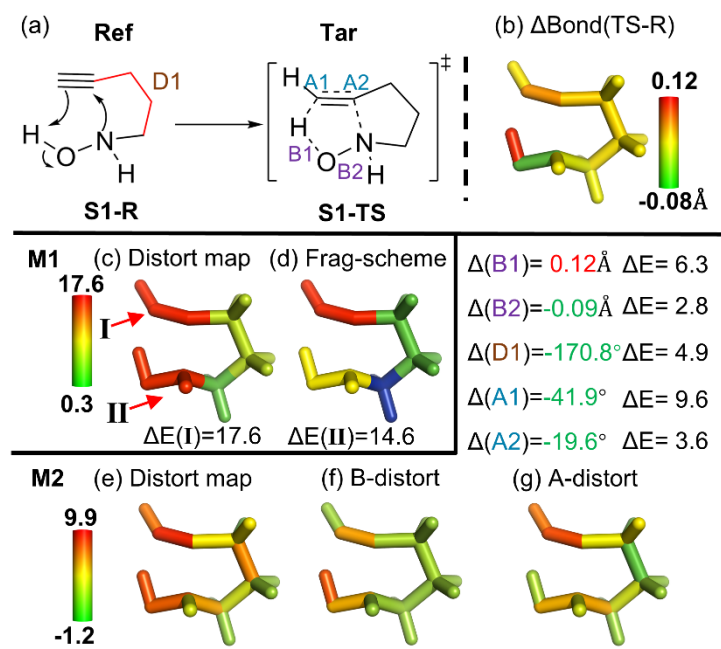


Tutorial for D2AF



Using the reverse Cope elimination as example

Structure file: Ref: 4F-reactant.gjf & Tar: TS-4F.gjf

4F-reactant.gjf contains the **connectivity** information

1. M1:

Input file: 4F_M1.inp

```
ref = 4F-reactant.gjf
conf = TS-4F.gjf
method = 1
cpu = 4
pal = 6
calculator = g16
scale=10

#fraglist
1,2,3
4-6,14-16
11-13
7-10
```

Pre-define the Gaussian 16 (or other calculators) environment then run:

D2AF -inp 4F_M1.inp > 4F_M1.log

Outputs as:

File	Size	Description
pymol		
tmpdir		
4F_M1.inp	1 KB	
4F_M1.log	7 KB	log file
4F_M1.pbs	1 KB	
4F_M1_fragments_info.dat	1 KB	Fragments info
4F_M1_M1.xlsx	5 KB	Result data
Conf_M1.xyz	2 KB	Sub-system structures
Ref_M1.xyz	2 KB	

pymol: for generate the maps by Pymol

File	Size	Description
Conf.xyz		
Conf_anglen_delta_Geo.pml		Angle difference negative
Conf_anglep_delta_Geo.pml		Angle difference positive
Conf_bond_delta_Geo.pml		Bond difference
Conf_M1_frag_show.pml		Fragmentation scheme
Conf_M1_total_E.pml		Distortion map

tmpdir: computation input and output files

File	Size	Description
M1_conf_0_0.gjf	1 KB	
M1_conf_0_0.log	13 KB	
M1_conf_0_1.gjf	1 KB	
M1_conf_0_1.log	15 KB	
M1_conf_0_2.gjf	1 KB	
M1_conf_0_2.log	13 KB	
M1_conf_0_3.gjf	1 KB	
M1_conf_0_3.log	13 KB	
M1_ref_0.gjf	1 KB	
M1_ref_0.log	13 KB	
M1_ref_1.gjf	1 KB	
M1_ref_1.log	15 KB	
M1_ref_2.gjf	1 KB	
M1_ref_2.log	13 KB	
M1_ref_3.gjf	1 KB	
M1_ref_3.log	13 KB	

2. M2:

Input file: 4F_M2.inp

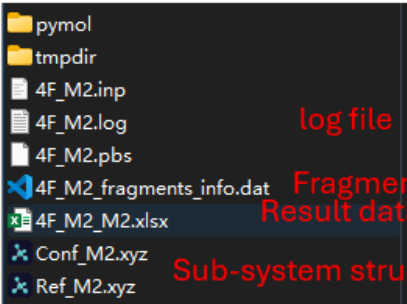
```
ref = 4F-reactant.gjf
conf = TS-4F.gjf
method = 2
cpu = 4
pal = 6
calculator = g16
scale=10

#include
1 4 14 11
```

Pre-define the Gaussian 16 (or other calculators) environment then run:

D2AF -inp 4F_M2.inp > 4F_M2.log

Outputs as:

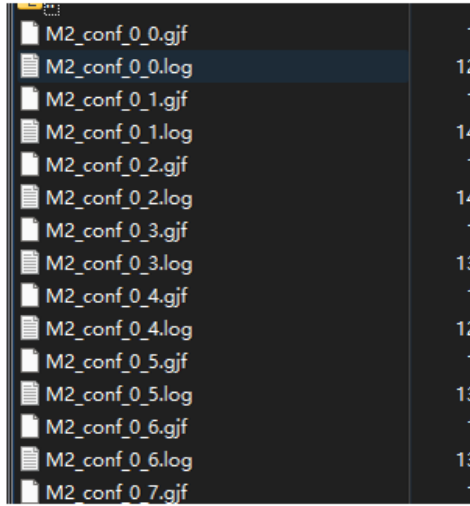


log file

Fragments info
Result data

Sub-system structures

pymol: for generate the maps by Pymol



Angle difference negative

Angle difference positive

Bond difference

Distortion map from Angle
Bond
Dihedral
total

3. M3:

Input file: 4F_M3.inp

```
ref = 4F-reactant.gjf
conf = TS-4F.gjf
method = 3
cpu = 4
pal = 6
calculator = g16
scale=10

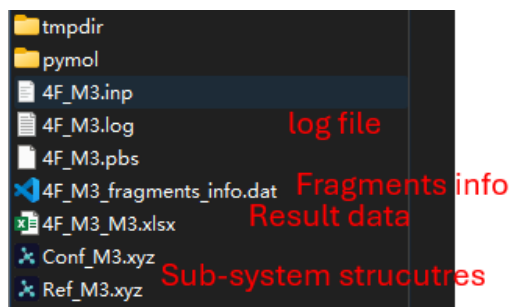
#fraglist
1,2,3

#include
1 4 14 11
```

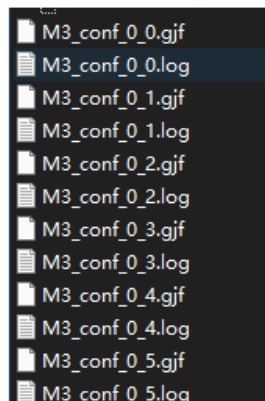
Pre-define the Gaussian 16 (or other calculators) environment then run:

D2AF -inp 4F_M3.inp > 4F_M3.log

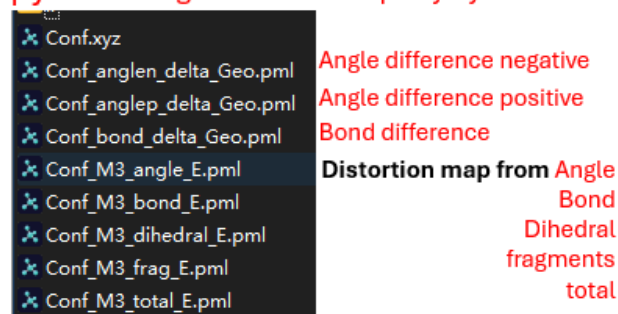
Outputs as:



tmpdir: computation input and output files



pymol: for generate the maps by Pymol



3. Energy computation by user (here use M2 as example)

Input file: 4F_M2.inp

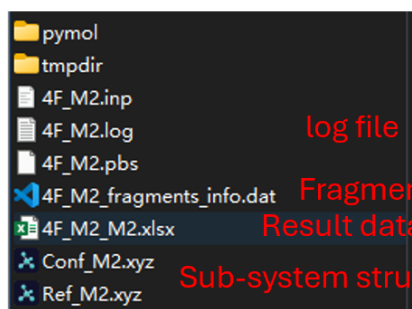
```
ref = 4F-reactant.gjf
conf = TS-4F.gjf
method = 2
cpu = 4
pal = 6
calculator = nocalc
scale=10

#include
1 4 14 11
```

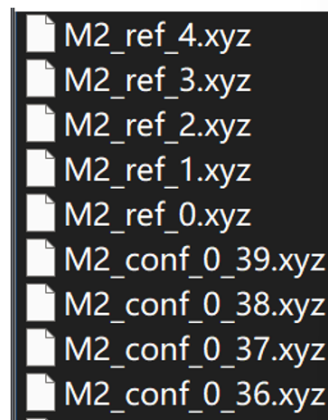
run:

D2AF -inp 4F_M2.inp > 4F_M2.log

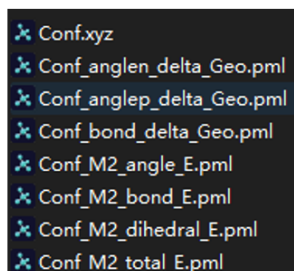
Outputs as:



tmpdir: fragments in xyz format



pymol: for generate the maps by Pymol



Angle difference negative

Angle difference positive

Bond difference

Distortion map from Angle

Bond

Dihedral

total

Distortion map wrong

Run QC calculations and extract the energies by users

Two possible ways to get the results (xlsx, pymol scripts for maps):

(a) Extract the energies to ****xxx_i.log**** (in a.u. unit; 1st line) for each structure in the ****\tmpdir**** folder*

e.g.: M2_ref_0.log for M2_ref_0.xyz

-77.32558773

Run the following cmd again:

D2AF -inp 4F_M2.inp > 4F_M2.log

(b) Extract the energies to a energy data file *M2_ene.dat*

Format of line(each line for one xyz file):

xyzname E(in a.u.)

M2_conf_0_0	-77.32041336
M2_conf_0_10	-172.06641049
M2_conf_0_11	-40.51684076
M2_conf_0_12	-79.82886505
M2_conf_0_13	-79.82882825
M2_conf_0_14	-131.69969592
M2_conf_0_15	-56.53531520
M2_conf_0_16	-95.83954060
M2_conf_0_17	-131.70256199
M2_conf_0_18	-171.01142018

Then run:

D2AF -d 4F_M2_M2.xlsx -m 2 -f pymol Conf.xyz -e M2_ene.dat

Strain energies in **4F_M2_M2.xlsx** and **pymol scripts** was updated based on **M2_ene.dat**

Output result is the same the **2. M2**