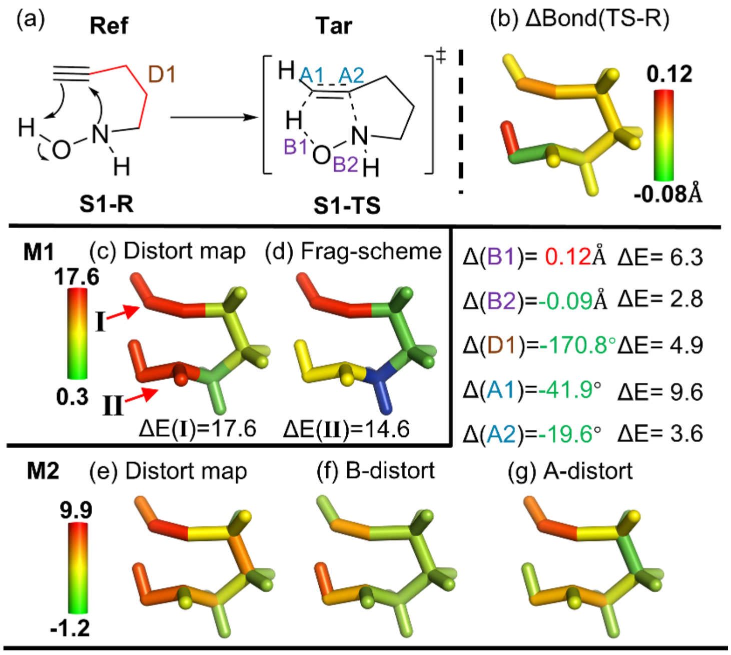
Tutorial for D2AF



Using the reverse Cope elimination as example

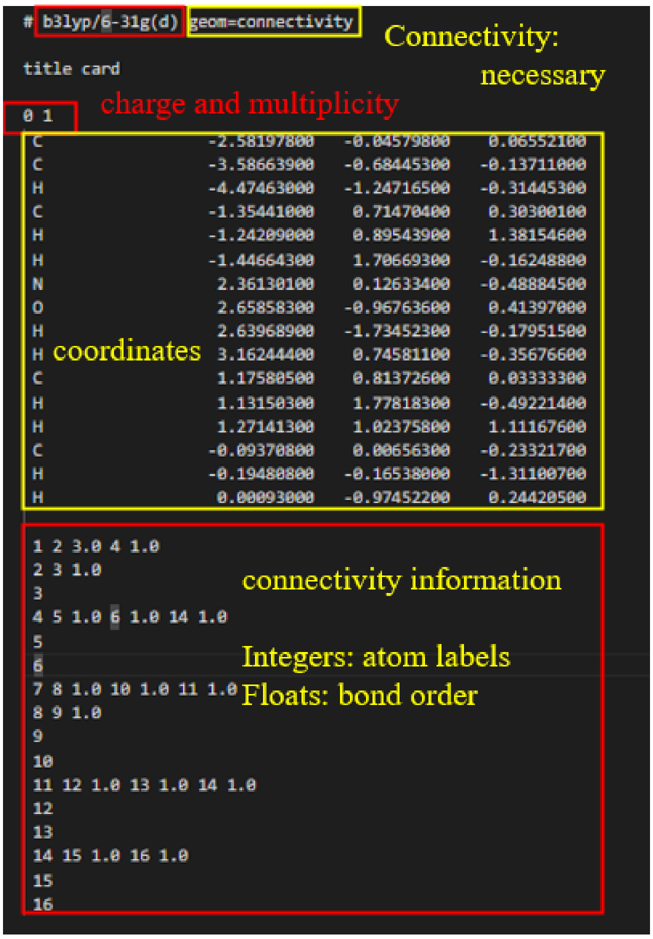
1. Preparations

Input files:

Ref: 4F-reactant.gjf & **Tar**: TS-4F.gjf

Ref: 4F-reactant.gjf

Now only support the gjf type file (see right)



Computational method

needed for gaussian and ORCA calculator

Connectivity: necessary

charge and multiplicity

coordinates

connectivity information

Integers: atom labels

Floats: bond order

```
# b3lyp/6-31g(d) geom=connectivity

title card

0 1
C                  -2.58197800   -0.04579800    0.06552100
C                  -3.58663900   -0.68445300   -0.13711000
H                  -4.47463000   -1.24716500   -0.31445300
C                  -1.35441000    0.71470400    0.30300100
H                  -1.24209000    0.89543900    1.38154600
H                  -1.44664300    1.70669300   -0.16248800
N                   2.36130100    0.12633400   -0.48884500
O                   2.65858300   -0.96763600    0.41397000
H                   2.63968900   -1.73452300   -0.17951500
H                   3.16244400    0.74581100   -0.35676600
C                   1.17580500    0.81372600    0.03333300
H                   1.13150300    1.77818300   -0.49221400
H                   1.27141300    1.02375800    1.11167600
C                  -0.09370800    0.00656300   -0.23321700
H                  -0.19480800   -0.16538000   -1.31100700
H                   0.00093000   -0.97452200    0.24420500

1 2 3.0 4 1.0
2 3 1.0
3
4 5 1.0 6 1.0 14 1.0
5
6
7 8 1.0 10 1.0 11 1.0
8 9 1.0
9
10
11 12 1.0 13 1.0 14 1.0
12
13
14 15 1.0 16 1.0
15
16
```

Bond order:

- the connectivity (bond order) values should be **1.0, 2.0, 3.0** only!
- **1.5 is forbidden**, it should be modified before running D2AF

1.1 connectivity and auto-fragments

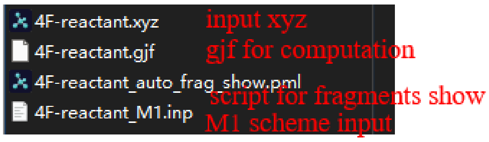
For only xyz case (or gjf without connectivity):

autofragment 4F-reactant.xyz

the connectivity can be generated using *openbabel*,

and fragments list can be auto generated

Outputs as:



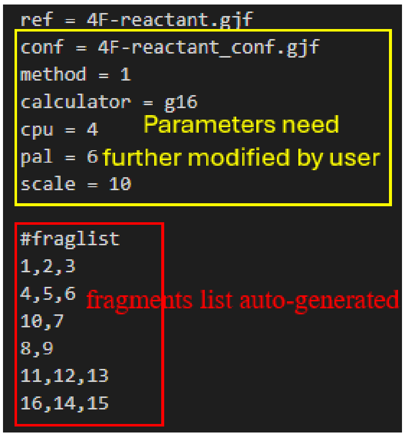
```
4F-reactant.xyz
4F-reactant.gjf
4F-reactant_auto_frag_show.pml
4F-reactant_M1.inp
```

input xyz

gjf for computation

script for fragments show

M1 scheme input



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

#fraglist
1,2,3
4,5,6
10,7
8,9
11,12,13
16,14,15
```

Parameters need further modified by user

fragments list auto-generated

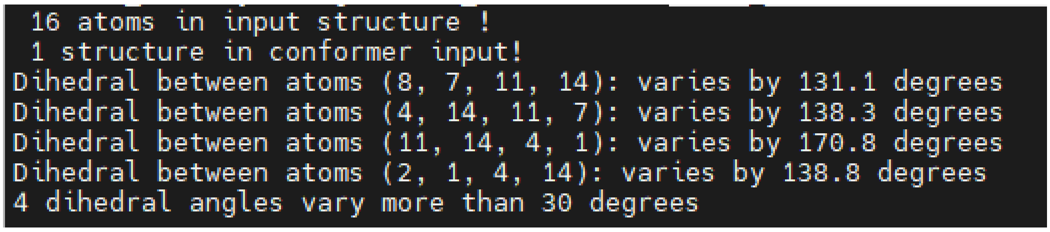
1.2 Check dihedrals

Check the large dihedral difference between reference and target structures:

Inputs: ref conf threshold (optional, default 30°)

check_dihedral 4F-reactant.gjf TS-4F.gjf

Output as:



```
 16 atoms in input structure !
 1 structure in conformer input!
Dihedral between atoms (8, 7, 11, 14): varies by 131.1 degrees
Dihedral between atoms (4, 14, 11, 7): varies by 138.3 degrees
Dihedral between atoms (11, 14, 4, 1): varies by 170.8 degrees
Dihedral between atoms (2, 1, 4, 14): varies by 138.8 degrees
4 dihedral angles vary more than 30 degrees
```

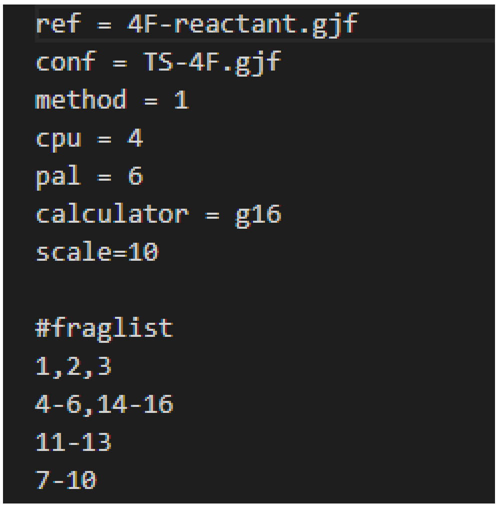
Dihedrals with a larger difference should be noted:

considered as a complete fragment in the **M1** scheme

included in the **M2/M3** scheme.

2. M1 scheme:

Input file: 4F_M1.inp

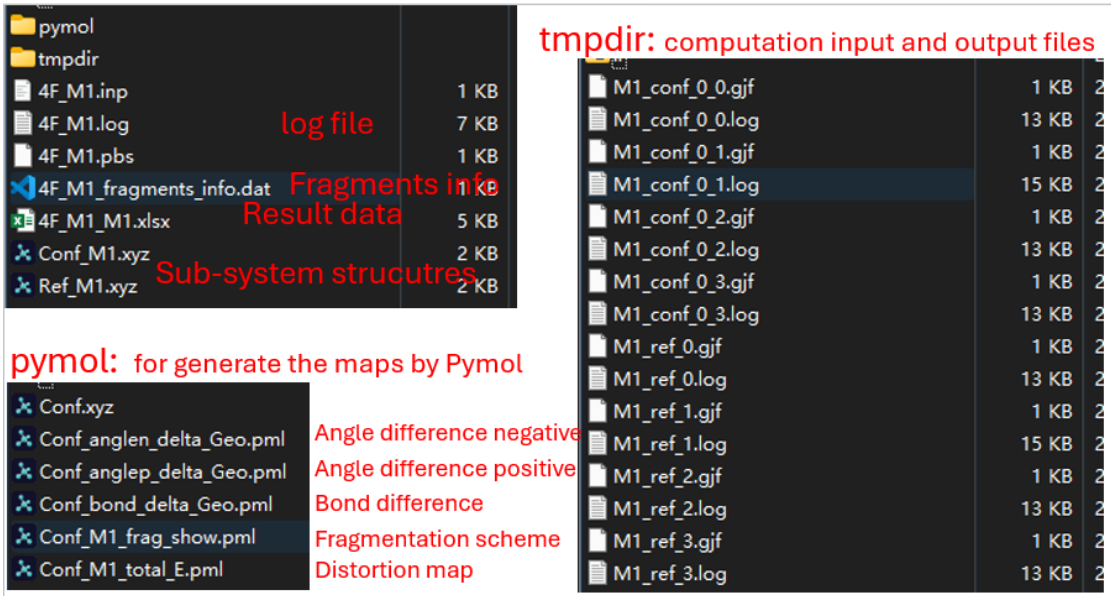


```
ref = 4F-reactant.gjf
conf = TS-4F.gjf
method = 1
cpu = 4
pal = 6
calculator = g16
scale=10

#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:



pymol	
tmpdir	
4F_M1.inp	1 KB
4F_M1.log	7 KB
4F_M1.pbs	1 KB
4F_M1_fragments_info.dat	1 KB
4F_M1_M1.xlsx	5 KB
Conf_M1.xyz	2 KB
Ref_M1.xyz	2 KB

tmpdir: computation input and output files

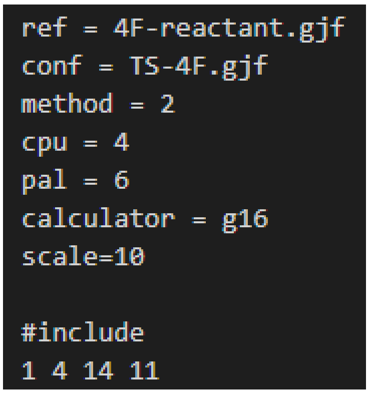
M1_conf_0_0.gjf	1 KB
M1_conf_0_0.log	13 KB
M1_conf_0_1.gjf	1 KB
M1_conf_0_1.log	15 KB
M1_conf_0_2.gjf	1 KB
M1_conf_0_2.log	13 KB
M1_conf_0_3.gjf	1 KB
M1_conf_0_3.log	13 KB
M1_ref_0.gjf	1 KB
M1_ref_0.log	13 KB
M1_ref_1.gjf	1 KB
M1_ref_1.log	15 KB
M1_ref_2.gjf	1 KB
M1_ref_2.log	13 KB
M1_ref_3.gjf	1 KB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

pymol: for generate the maps by Pymol

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

3. M2 scheme:

Input file: 4F_M2.inp

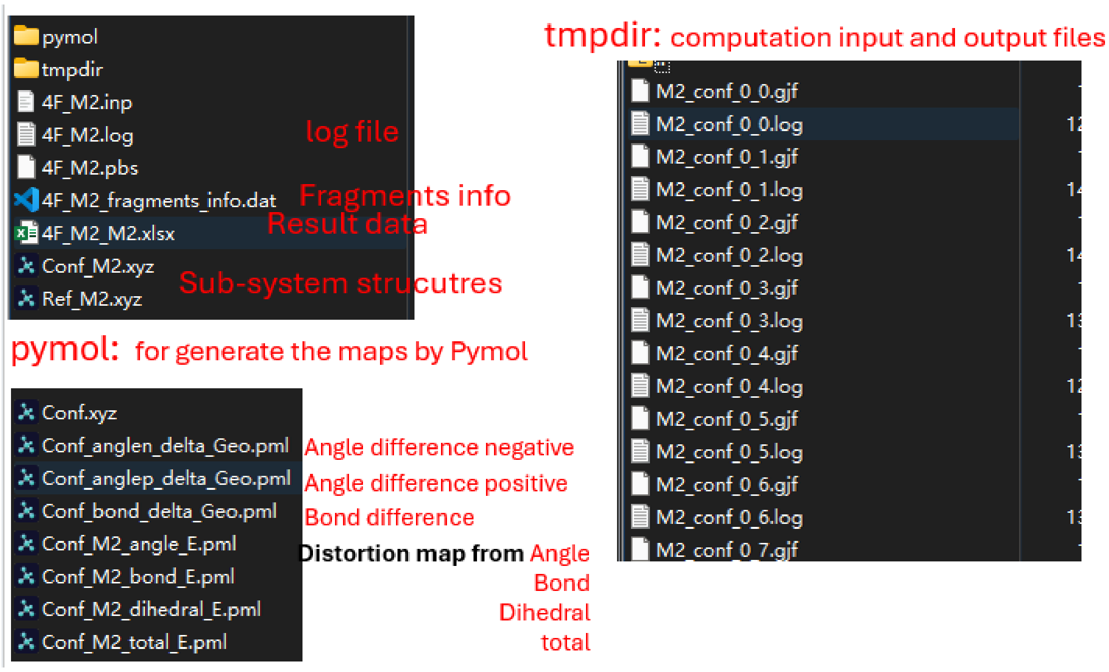


```
ref = 4F-reactant.gjf
conf = TS-4F.gjf
method = 2
cpu = 4
pal = 6
calculator = g16
scale=10

#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:



```
pymol
tmpdir
4F_M2.inp
4F_M2.log
4F_M2.pbs
4F_M2_fragments_info.dat
4F_M2_M2.xlsx
Conf_M2.xyz
Ref_M2.xyz
```

log file

Fragments info

Result data

Sub-system strucutres

pymol: for generate the maps by Pymol

```
Conf.xyz
Conf_anglen_delta_Geo.pml
Conf_anglep_delta_Geo.pml
Conf_bond_delta_Geo.pml
Conf_M2_angle_E.pml
Conf_M2_bond_E.pml
Conf_M2_dihedral_E.pml
Conf_M2_total_E.pml
```

Angle difference negative

Angle difference positive

Bond difference

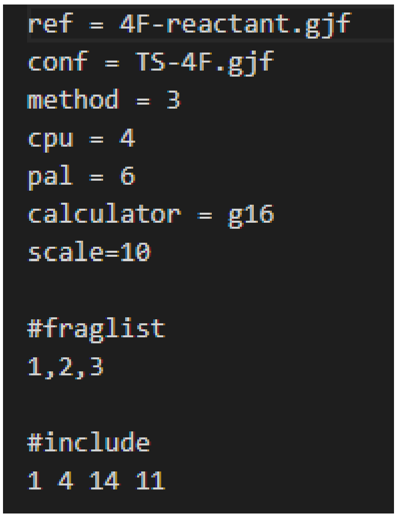
Distortion map from Angle
Bond
Dihedral
total

tmpdir: computation input and output files

```
M2_conf_0_0.gjf
M2_conf_0_0.log
M2_conf_0_1.gjf
M2_conf_0_1.log
M2_conf_0_2.gjf
M2_conf_0_2.log
M2_conf_0_3.gjf
M2_conf_0_3.log
M2_conf_0_4.gjf
M2_conf_0_4.log
M2_conf_0_5.gjf
M2_conf_0_5.log
M2_conf_0_6.gjf
M2_conf_0_6.log
M2_conf_0_7.gjf
```

4. M3 scheme:

Input file: 4F_M3.inp

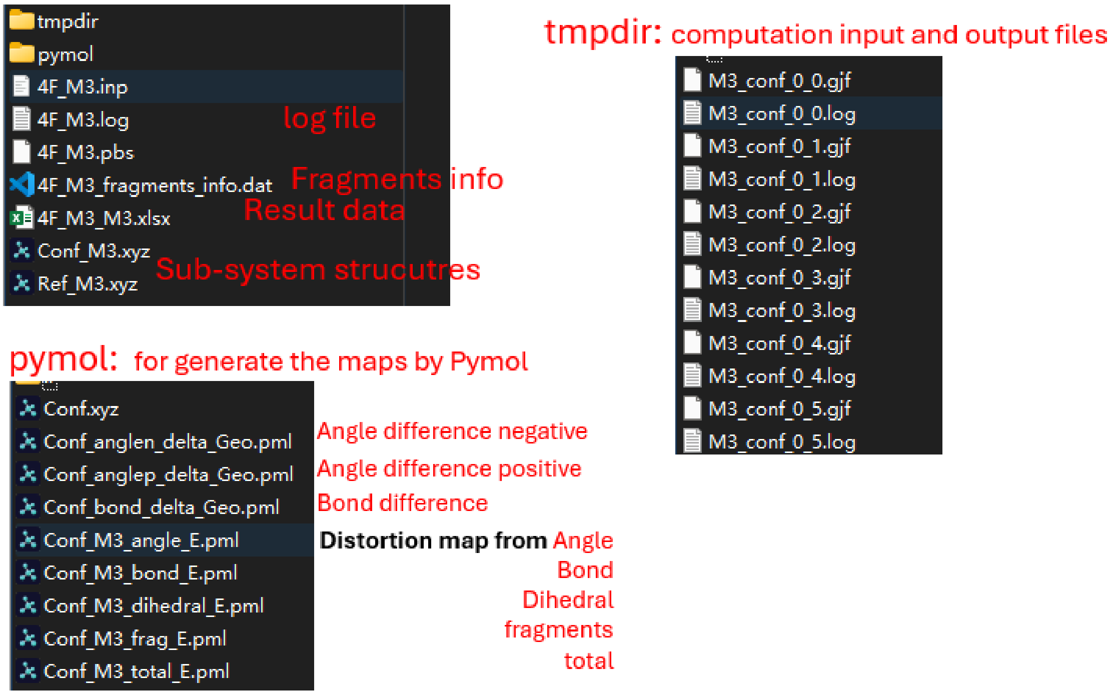


```
ref = 4F-reactant.gjf
conf = TS-4F.gjf
method = 3
cpu = 4
pal = 6
calculator = g16
scale=10

#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
pymol
4F_M3.inp
4F_M3.log
4F_M3.pbs
4F_M3_fragments_info.dat
4F_M3_M3.xlsx
Conf_M3.xyz
Ref_M3.xyz
```

log file

Fragments info

Result data

Sub-system strucutres

tmpdir: computation input and output files

```
M3_conf_0_0.gjf
M3_conf_0_0.log
M3_conf_0_1.gjf
M3_conf_0_1.log
M3_conf_0_2.gjf
M3_conf_0_2.log
M3_conf_0_3.gjf
M3_conf_0_3.log
M3_conf_0_4.gjf
M3_conf_0_4.log
M3_conf_0_5.gjf
M3_conf_0_5.log
```

pymol: for generate the maps by Pymol

```
Conf.xyz
Conf_anglen_delta_Geo.pml
Conf_anglep_delta_Geo.pml
Conf_bond_delta_Geo.pml
Conf_M3_angle_E.pml
Conf_M3_bond_E.pml
Conf_M3_dihedral_E.pml
Conf_M3_frag_E.pml
Conf_M3_total_E.pml
```

Angle difference negative

Angle difference positive

Bond difference

Distortion map from Angle

Bond

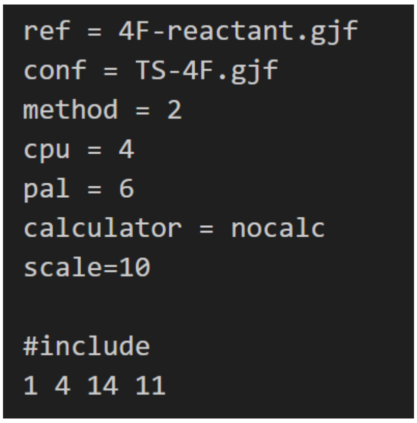
Dihedral

fragments

total

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

Input file: 4F_M2.inp

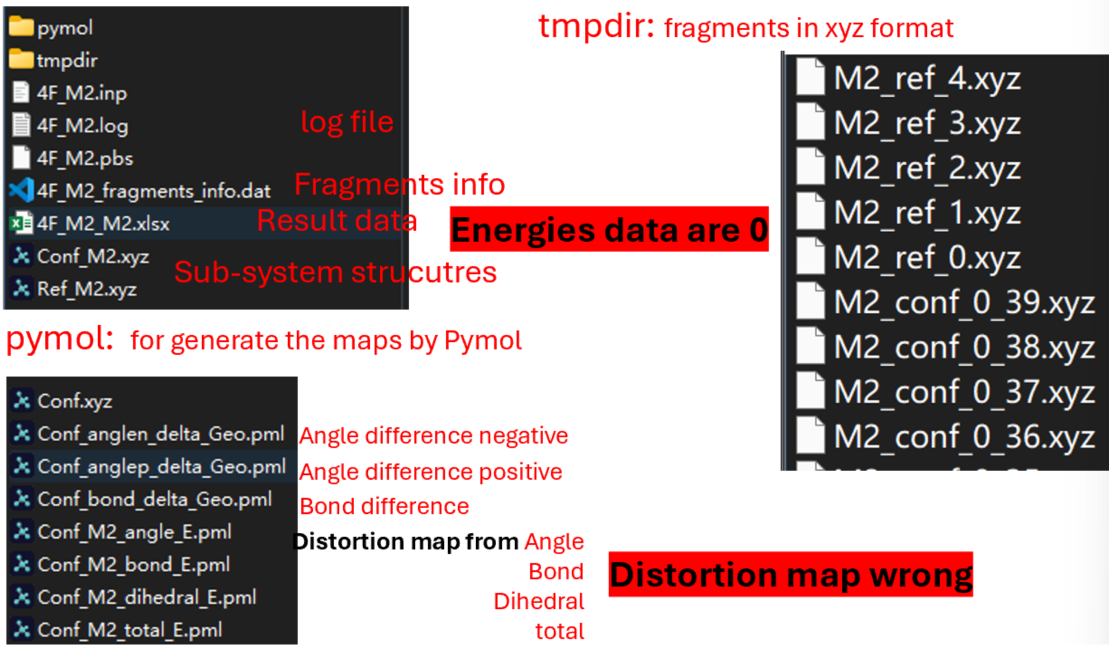


```
ref = 4F-reactant.gjf
conf = TS-4F.gjf
method = 2
cpu = 4
pal = 6
calculator = nocalc
scale=10

#include
1 4 14 11
```

run:

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

Outputs as:



```
pymol
tmpdir
4F_M2.inp
4F_M2.log
4F_M2.pbs
4F_M2_fragments_info.dat
4F_M2_M2.xlsx
Conf_M2.xyz
Ref_M2.xyz
```

```
M2_ref_4.xyz
M2_ref_3.xyz
M2_ref_2.xyz
M2_ref_1.xyz
M2_ref_0.xyz
M2_conf_0_39.xyz
M2_conf_0_38.xyz
M2_conf_0_37.xyz
M2_conf_0_36.xyz
```

```
Conf.xyz
Conf_anglen_delta_Geo.pml
Conf_anglep_delta_Geo.pml
Conf_bond_delta_Geo.pml
Conf_M2_angle_E.pml
Conf_M2_bond_E.pml
Conf_M2_dihedral_E.pml
Conf_M2_total_E.pml
```

tmpdir: fragments in xyz format
log file
Fragments info
Result data
Energies data are 0
Sub-system strucutres
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

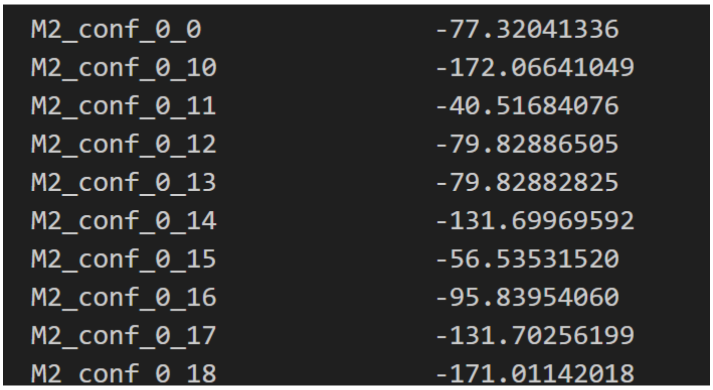
Run the following command again:

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

(b) Extract the energies to an 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

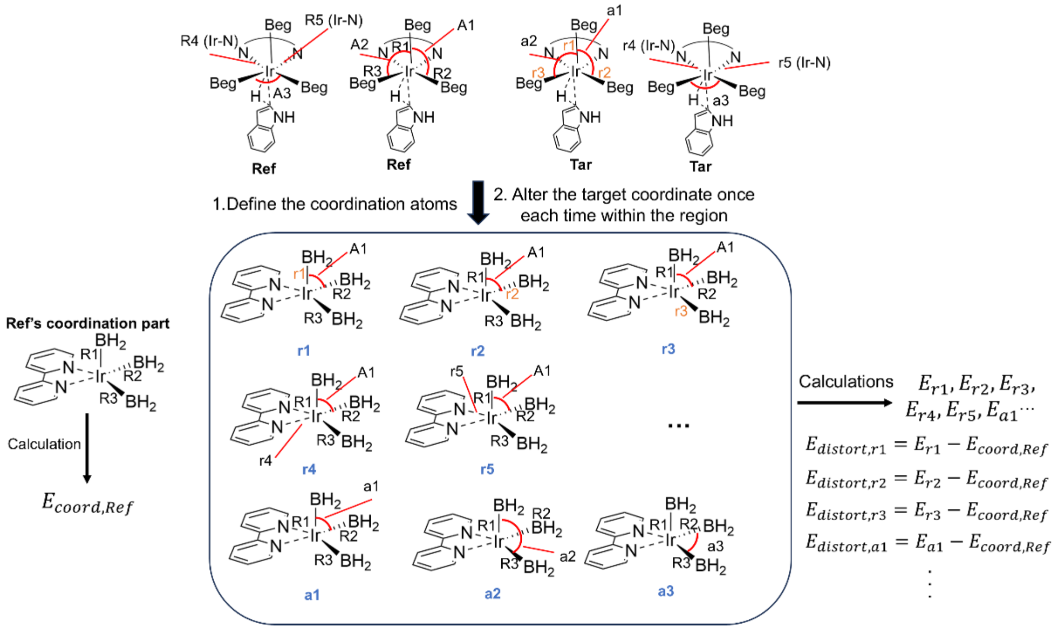
Strain energies in **4F_M2_M2.xlsx** and **pymol scripts** were updated based on **M2_ene.dat**

Output result is the same the **M2 scheme**

6. Coordination case in M3

According to ligand field theory, metal coordination compounds exhibit distinctive and complex metal-ligand interactions. The key orbitals for this system are generally influenced and sensitive by ligands (coordinate numbers/elements), directly reflecting the electronic structure around the transition metal. Therefore, maintaining consistent metal-ligand interactions becomes crucial when analyzing distortions related to the transition metal compounds. To tackle this challenge, we propose a novel feature in **M3**, in which users can define a minimum and reasonable metal coordination region as one special fragment to capture the key metal-ligand interactions using **M2**.

The **Ref** fragment was generated including all atoms in this special fragment (coordination region with their link atom(s)) taken from the reference form. Then, only one target internal coordinate in the coordination region is altered to be identical as to that coordinate value in the target system to set up all combinations of the **Tar** fragments. Accordingly, in local distribution distortion of each bonding coordinate within the coordination region is evaluated individually. After calculations to obtain energy of these fragments, (relative) local distortion energy of each individual target M-L bond can be estimated. It should be noted that this coordinate treatment cannot work perfectly if any large coordinate change involving the metal-ligand chelation is involved between the **Ref** and **Tar** cooridinate fragment. This approach should help facilitate our analysis of local distortion distribution in metal coordination compounds.



R4 (Ir-N)
R5 (Ir-N)
Beg
A3
NH
Ref
A2
A1
R1
R2
R3
Ref
a1
a2
r1
r2
r3
Tar
r4 (Ir-N)
r5 (Ir-N)
a3
Tar
1.Define the coordination atoms
2. Alter the target coordinate once each time within the region
Ref's coordination part
BH_2
Calculation
$E_{coord,Ref}$
r1
r2
r3
r4
r5
...
a1
a2
a3
Calculations
$E_{r1}, E_{r2}, E_{r3}, E_{r4}, E_{r5}, E_{a1} \cdots$
$E_{distort,r1} = E_{r1} - E_{coord,Ref}$
$E_{distort,r2} = E_{r2} - E_{coord,Ref}$
$E_{distort,r3} = E_{r3} - E_{coord,Ref}$
$E_{distort,a1} = E_{a1} - E_{coord,Ref}$
⋮

7. Others

Manual update xlsx data and regenerate maps

e.g.: exclude selected terms

or manual calculates the difference between the i^{th} ref and conf fragments and update the xlsx data

run:

```
D2AF -d 4F_M2_M2.xlsx -m 2 -f pymol Conf.xyz
```

pymol scripts were generated.

Output result is the same the **M2 scheme**