Code availability

We list the original computer procedure codes about 'Dooerafa' and the software to install tools. This process will help other reviewers be able to run them.

1. PBS Script

```
#!/bin/bash
#PBS -N py_A
#PBS -l nodes=1:ppn=12
#PBS -l walltime=1440:00:00
#PBS -q q_share
#PBS -o jobID.$PBS_JOBID
#PBS -V
#PBS -S /bin/bash
cd /home/lipl/shixing/pro/pro_new/A/
touch jobID.$PBS_JOBID
chmod +x A.py
python A.py > A.txt
```

Description: The script file used to submit the task.

2. Code 1

```
# -*- coding: UTF-8 -*-
from collections import Counter
import copy
class Norm(object):
    """Create an atomic entity class as the parent class"""
    list = []
    def __init__ (self):
        """Initialize properties"""
        pass
        Norm.list.append(self)
class W1(Norm):
    """Create a subclass that the locus is one, inheriting the parent class named Norm"""
    def __init__ (self):
        """Initialize the attributes of the parent class"""
        super(W1,self).__init__()
        self.J1 = 0
        self.J2 = 1
        self.J3 = 1
        self.J4 = 1
class W2(Norm):
    """Create a subclass that the locus is two, inheriting the parent class named Norm"""
    def __init__ (self):
        """Initialize the attributes of the parent class"""
```

```
            super(W2,self).__init__()
            self.J1 = 0
            self.J2 = 0
            self.J3 = 1
            self.J4 = 1
class W3(Norm):
    """Create a subclass that the locus is three, inheriting the parent class named Norm"""
    def __init__ (self):
            """Initialize the attributes of the parent class"""
            super(W3,self).__init__()
            self.J1 = 0
            self.J2 = 0
            self.J3 = 0
            self.J4 = 1
class W4(Norm):
    """Create a subclass that the locus is four, inheriting the parent class named Norm"""
    def __init__ (self):
            """Initialize the attributes of the parent class"""
            super(W4,self).__init__()
            self.J1 = 0
            self.J2 = 0
            self.J3 = 0
            self.J4 = 0
c1 = W1()
c2 = W2()
c3 = W3()
c4 = W3()
c5 = W1()
c6 = W1()
c7 = W2()
c8 = W1()
#The selected classe is determined according to the number of bonds available at undetermined sites in the
compound structure.
e = Norm.list
a = []
for i in range(0,len(e)):
    if e[i].J3 is 0:
            a.append(i+1)
            a.append(i+1)
            a.append(i+1)
    elif e[i].J3 is 1 and e[i].J2 is 0:
            a.append(i+1)
            a.append(i+1)
    elif e[i].J3 is 1 and e[i].J2 is 1 and e[i].J1 is 0:
            a.append(i+1)
print(a)
class Solution:
```

```
    def permute(self,nums):
        visited = [False for _ in range(len(nums))]
        path = []
        depth = 0
        nums.sort()
        format = []
        def dfs(nums,visited,path,depth,format):
            if depth == len(nums):
                res = copy.deepcopy(path)
                step = 2
                x = sorted([res[it:it+step] for it in range(0,len(res),step)])
                b = sorted([list(t) for t in set(tuple(_) for _ in x)])
                re = []
                if b == x:
                    for lm in x:
                        l = sorted(lm)
                        if len(l) == len(set(l)):
                            re.append(l)
                if re == x:
                    format.append(re)
                return
            for i in range(len(nums)):
                if i >0 and nums[i] == nums[i-1] and not visited[i-1]:
                    continue
                if not visited[i]:
                    visited[i] = True
                    path.append(nums[i])
                    dfs(nums,visited,path,depth+1,format)
                    visited[i] = False
                    path.pop()
        dfs(nums,visited,path,depth,format)
        return format
#Perform random permutations and screen out the results meeting the conditions.
s = Solution()
n = s.permute(a)
new_z = []
for z in n:
    if z not in new_z:
        new_z.append(z)
print(new_z)
```

Description: Centered on self-created random permutations and combinations, the group data representing all possibilities for assembling the meta groups based on meta structures are obtained.

Illustration: #Code 1 and PBS Script are performed in Supercomputing Center in Pilot National Laboratory for Marine Science and Technology.

3. Code 2 and Code 3

Code 2

```
import time
import pyautogui
number = 1
yx = 329
y1 = 289
y2 =313
numbermax = 5
while True:
    if number == num:
#the num is the last number computing
        break
    else:
        if number<= numbermax:
            print(pyautogui.moveTo(x=1197, y=yx,duration=0.5))
            print(pyautogui.doubleClick())
            print(pyautogui.doubleClick())
            print(time.sleep(7))
            print(pyautogui.moveTo(x=1466, y=925,duration=0.5))
            print(pyautogui.click(button='left'))
            print(time.sleep(7))
            print(pyautogui.moveTo(x=1211, y=238,duration=0.5))
            print(pyautogui.click(button='left'))
            print(time.sleep(50))
            print(pyautogui.moveTo(x=1364, y=1052,duration=0.5))
            print(pyautogui.click(button='left'))
            print(pyautogui.moveTo(x=1052, y=y1,duration=0.5))
            print(pyautogui.click(button='left'))
            print(pyautogui.hotkey('ctrl','c'))
            print(pyautogui.moveTo(x=414, y=543,duration=0.5))
            print(pyautogui.click(button='left'))
            print(pyautogui.moveTo(x=308, y=207,duration=0.5))
            print(pyautogui.click(button='left'))
            print(pyautogui.moveTo(x=336, y=398,duration=0.5))
            print(pyautogui.click(button='left'))
            print(pyautogui.hotkey('ctrl','v'))
            print(pyautogui.moveTo(x=1151, y=814,duration=0.5))
            print(pyautogui.click(button='left'))
            print(pyautogui.moveTo(x=466, y=895,duration=0.5))
            print(pyautogui.dragTo(x=518, y=895,duration=0.5))
            print(pyautogui.moveTo(x=495, y=895,duration=0.5))
            print(pyautogui.click(button='right'))
            print(pyautogui.moveTo(x=581, y=749,duration=0.5))
            print(pyautogui.click(button='left'))
```

```
            print(time.sleep(2))
            print(pyautogui.moveTo(x=1425, y=1059,duration=0.5))
            print(pyautogui.click(button='left'))
            print(pyautogui.moveTo(x=1842, y=180,duration=0.5))
            print(pyautogui.click(button='left'))
            print(time.sleep(4))
            print(pyautogui.moveTo(x=1379, y=y2,duration=0.5))
            print(pyautogui.click(button='left'))
            print(pyautogui.hotkey('ctrl','v'))
            number += 1
            yx += 180
            y1 += 180
            y2 += 180
        else:
            print(pyautogui.scroll(-1100))
            print(pyautogui.moveTo(x=1374, y=1059,duration=0.5))
            print(pyautogui.click(button='left'))
            print(pyautogui.moveTo(x=1662, y=753,duration=0.5))
            print(pyautogui.click(button='left'))
            print(pyautogui.scroll(-1600))
            print(pyautogui.moveTo(x=1426, y=1054,duration=0.5))
            print(pyautogui.click(button='left'))
            numbermax += 30
            yx = 329
            y1 = 289
            y2 =313
```

Code 3

```
import pyautogui
print(pyautogui.position())
```

Description: The five molecular energy calculation in mechanic force field is automatically performed by Python controlling the Microsoft Excel containing all the structures and Chem3D.

Illustration: After further Data processing and filtering, Code 2 and Code 3 run based on the structures converted from the data sets obtained by Code 1 and PBS Script by drawing manually with the ChemDraw software.

The related softwares

The Python program package was installed from https://www.python.org/downloads/
The Sublime text 3 editor was installed from https://www.sublimetext.com/
The ChmeBioOiffice was installed from the file of ChemBioOffice
The EasyConnect was installed from https://144.123.46.68:4668
The MobaXterm was installed from https://mobaxterm.mobatek.net/download.html

How to install the Python program package was followed by http://c.biancheng.net/view/4161.html

How to install the Sublime text 3 editor was followed by
https://blog.csdn.net/wjlvxuewei/article/details/80165545