

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Wed Jun 26 14:10:39 2019

@author: eleonorashantsila
"""

import optparse

parser = optparse.OptionParser()

parser.add_option('--fasta_sequence', dest='fasta_sequence',
default='', help='fasta sequence for the .tertiary file
in .fasta.txt format')
parser.add_option('--tertiary_file', dest='tertiary_file', default =
'', help = '.tertiary file')
#parser.add_option('--xyz_filename', dest='xyz', default = '' ,help =
'.tertiary file')

(options,args) = parser.parse_args()

print('-----')
res=options.fasta_sequence
print('res=',res)
data= options.tertiary_file
print('data=',data)
protein_name=str(res[0:4])
print('protein_name=',protein_name)

res=open(res,"r")
data=open(data,"r")

residue=[]
for letter in res:
    residue.append(letter)

print('residue=',residue)
print('no of lines in res=', len(residue))
del residue[0]

print(residue)

residue_1=''
for i in range(len(residue)):
    print('i=',i)
    x=residue[i]
    x=x[0:(len(x)-1)]
```

```
residue_1=residue_1+x

#z=residue[2]
#z=z[0:(len(y)-1)]

#w=residue[3]
#w=w[0:(len(y)-1)]

#q=residue[4]
#q=q[0:(len(y)-1)]


print('residue_1=',residue_1)

residue=[]

for i in range(len(residue_1)):
    current=residue_1[i]
    if current=='A' :
        residue.append('ALA')
    if current=='R' :
        residue.append('ARG')
    if current=='N' :
        residue.append('ASN')
    if current=='D' :
        residue.append('ASP')
    if current=='C' :
        residue.append('CYS')
    if current=='Q' :
        residue.append('GLN')
    if current=='E' :
        residue.append('GLU')
    if current=='G' :
        residue.append('GLY')
    if current=='H' :
        residue.append('HIS')
    if current=='I' :
        residue.append('ILE')
    if current=='L' :
        residue.append('LEU')
    if current=='K' :
        residue.append('LYS')
    if current=='M' :
        residue.append('MET')
    if current=='F' :
        residue.append('PHE')
    if current=='P' :
        residue.append('PRO')
    if current=='S' :
        residue.append('SER')
    if current=='T' :
        residue.append('THR')
    if current=='W' :  
    if current=='W' :
```

```

        residue.append('TRP')
    if current=='Y' :
        residue.append('TYR')
    if current=='V' :
        residue.append('VAL')
    if current=='B' :
        residue.append('ASX')
    if current=='Z' :
        residue.append('GLX')
    if current=='X' :
        residue.append('XAA')

print(residue)

```

```

data_2=[]

for line in data:
    data_2.append(line)
del data_2[1]
del data_2[0]
imp_length=len(data_2)
print(imp_length)

#x
xx=data_2[0]
#y
yy=data_2[1]
#z
zz=data_2[2]

xx=xx.split()
yy=yy.split()
zz=zz.split()

#x
i_x=[]
for numb in range(len(xx)):
    x=float(xx[numb])/100
    i_x.append(x)
#y
i_y=[]
for numb in range(len(yy)):
    x=float(yy[numb])/100
    i_y.append(x)
#z

```

```

i_z=[]
for numb in range(len(zz)):
    x=float(zz[numb])/100
    i_z.append(x)

#coords=[]

#g=open('aa_2hh6.pdb','w')
#for i in xrange(len(i_x)):
#    co=[]
#    co.append(i_x[i])
#    co.append(i_y[i])
#    co.append(i_z[i])
#    coords.append(co)
#    g.write(co)

xyz_name=protein_name+'.xyz'

g=open(xyz_name,'w')

for i in range(int(len(i_x)/3)):

    ca='N'+' '+str(i_x[(i*3)])+' '+str(i_y[(i*3)])+'
'+str(i_z[(i*3)])+"\n"
    g.write(ca)

    n='CA'+' '+str(i_x[(i*3+1)])+' '+str(i_y[(i*3+1)])+'
'+str(i_z[(i*3+1)])+"\n"
    g.write(n)

    cp='CP'+' '+str(i_x[(i*3+2)])+' '+str(i_y[(i*3+2)])+'
'+str(i_z[(i*3+2)])+"\n"
    g.write(cp)

g.close()

h_ca=[]
h_n=[]
h_cp=[]

for i in range(int(len(i_x)/3)):

    ca=[]

```

```

ca.append(i_x[(i*3)])
ca.append(i_y[(i*3)])
ca.append(i_z[(i*3)])
h_ca.append(ca)

n=[]
n.append(i_x[(i*3+1)])
n.append(i_y[(i*3+1)])
n.append(i_z[(i*3+1)])
h_n.append(n)

cp=[]
cp.append(i_x[(i*3+2)])
cp.append(i_y[(i*3+2)])
cp.append(i_z[(i*3+2)])
h_cp.append(cp)

```

```

PDB=[]

pdb_name=protein_name+'_'+model.pdb'

f=open(pdb_name,'w')

PDB.append('HEADER      Bla          XX-')
PDB.append('XXX-9-    xxxx')                         XX-
f.write('HEADER      Bla          XX-XXX-9-')
f.write('xxxx\n')
a='ATOM'
count=1
space=' '
chain='A'
for i in range(len(residue)):
    res=residue[i]

    if count<10:
        extra='      '
    elif count>9 and count<100:
        extra='     '
    elif count>99 and count<1000:
        extra='    '
    elif count>999 and count<10000:
        extra='   '

```

```

if (i+1)<10:
    add=' '
elif (i+1)>9 and (i+1)<100:
    add=' '
elif (i+1)>99 and (i+1)<1000:
    add=' '

n=h_n[i]
first=n[0]
second=n[1]
third=n[2]

if first>=0 and first<10:
    plus=' '
elif first>=10 and first<100:
    plus=' '
elif first<0 and first>-10:
    plus=' '
elif first<=-10 and first>-100:
    plus=' '
elif first<=-100 and first>-1000:
    plus=' '

if second>=0 and second<10:
    plus2=' '
elif second>=10 and second<100:
    plus2=' '
elif second<0 and second>-10:
    plus2=' '
elif second<=-10 and second>-100:
    plus2=' '
elif second<=-100 and second>-1000:
    plus2=''

if third>=0 and third<10:
    plus3=' '
elif third>=10 and third<100:
    plus3=' '
elif third<0 and third>-10:
    plus3=' '
elif third<=-10 and third>-100:
    plus3=' '
elif third<=-100 and third>-1000:
    plus3=''

y=a+extra+str(count)
+space+space+'N'+space+space+space+res+space+chain+add+str((i+1))
+plus+format(n[0], '.3f')+plus2+format(n[1], '.3f')
+plus3+format(n[2], '.3f')
count=count+1
PDB.append(y)

```

```
f.write(y)
f.write("\n")

if count<10:
    extra=' '
elif count>9 and count<100:
    extra=' '
elif count>99 and count<1000:
    extra=' '
elif count>999 and count<10000:
    extra=' '

if (i+1)<10:
    add=' '
elif (i+1)>9 and (i+1)<100:
    add=' '
elif (i+1)>99 and (i+1)<1000:
    add=' '

c_a=h_ca[i]
first=c_a[0]
second=c_a[1]
third=c_a[2]

if first>=0 and first<10:
    plus=' '
elif first>=10 and first<100:
    plus=' '
elif first<0 and first>-10:
    plus=' '
elif first<=-10 and first>-100:
    plus=' '
elif first<=-100 and first>-1000:
    plus=' '

if second>=0 and second<10:
    plus2=' '
elif second>=10 and second<100:
    plus2=' '
elif second<0 and second>-10:
    plus2=' '
elif second<=-10 and second>-100:
    plus2=' '
elif second<=-100 and second>-1000:
    plus2=' '

if third>=0 and third<10:
    plus3=' '
elif third>=10 and third<100:
    plus3=' '
```

```

    elif third<0 and third>-10:
        plus3=' '
    elif third<=-10 and third>-100:
        plus3=' '
    elif third<=-100 and third>-1000:
        plus3=''

    x=a+extra+str(count)
+space+space+'CA'+space+space+res+space+chain+add+str((i+1))
+plus+format(c_a[0],'.3f')+plus2+format(c_a[1],'.3f')
+plus3+format(c_a[2],'.3f')
    count=count+1
    PDB.append(x)
    f.write(x)
    f.write("\n")

if count<10:
    extra=' '
elif count>9 and count<100:
    extra=' '
elif count>99 and count<1000:
    extra=' '
elif count>999 and count<10000:
    extra=' '

if (i+1)<10:
    add=' '
elif (i+1)>9 and (i+1)<100:
    add=' '
elif (i+1)>99 and (i+1)<1000:
    add=' '

c_p=h_cp[i]
first=c_p[0]
second=c_p[1]
third=c_p[2]

if first>=0 and first<10:
    plus=' '
elif first>=10 and first<100:
    plus=' '
elif first<0 and first>-10:
    plus=' '
elif first<=-10 and first>-100:
    plus=' '
elif first<=-100 and first>-1000:
    plus=' '

if second>=0 and second<10:
    plus2=' '

```

```

        elif second>=10 and second<100:
            plus2=' '
        elif second<0 and second>-10:
            plus2=' '
        elif second<=-10 and second>-100:
            plus2=' '
        elif second<=-100 and second>-1000:
            plus2=''

        if third>=0 and third<10:
            plus3=' '
        elif third>=10 and third<100:
            plus3=' '
        elif third<0 and third>-10:
            plus3=' '
        elif third<=-10 and third>-100:
            plus3=' '
        elif third<=-100 and third>-1000:
            plus3=''

z=a+extra+str(count)
+space+space+'C'+space+space+space+res+space+chain+add+str((i+1))
+plus+format(c_p[0], '.3f')+plus2+format(c_p[1], '.3f')
+plus3+format(c_p[2], '.3f')
count=count+1
PDB.append(z)
f.write(z)
f.write("\n")

PDB.append('END')
f.write('END')
f.write("\n")

#with open("new_pdb.txt","w") as output:
#    output.write(str(PDB))

f.close()

```