* 小分子处理

位置：207 服务器 /home/g16/zlp/hb\_xb\_env\_QM\_MM/

* 质子化

先将小分子用pymol存成mol2格式

预测小分子质子化状态

预测完之后注意检查，与PDB网页上的结构核对一下，有时候双键会识别错误。这时下载网页上的mol2格式，再次加H即可，或者遇到卤素原子不识别的问题，用pymol将网页上的mol2转存一次，再加H即可

3kku

chmod +x 01\_keep3D.sh

../../01\_keep3D.sh -l lig.mol2

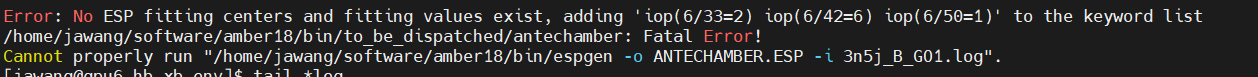
sdconvert -imae lig\_epikout.mae -osdf lig\_epikout.sdf

obabel -isdf lig\_epikout.sdf -opdb -O lig\_epikout.pdb

* 高斯优化小分子

sdf用antechamber或者Gaussianview存成gjf，修改gjf为，iop那一项一定要加6/33=2，不然后面就会报错

antechamber -i lig\_epikout.sdf -fi sdf -o lig\_epikout.gjf -fo gcrt -nc 0



----------------------------------------------------------------

(base) [zjxu@DDB hb\_xb\_env\_QM\_MM]$ head \*gjf

%chk=ligand

%mem=12000Mb

%nprocshared=12

# opt hf/6-31g(d) geom=connectivity pop=mk iop(6/33=2,6/42=6)

obj01

-2 1

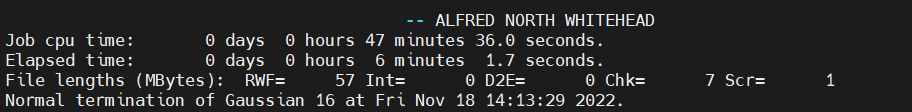
-----------------------------------------------------------------

这一步用HF的原因是：低层的原子电荷拟合的时候用的是HF，为了保持一致，小分子算resp电荷的时候也用HF

[科学网—Gaussian高斯里计算resp电荷方法 - 徐扬的博文 (sciencenet.cn)](https://blog.sciencenet.cn/home.php?mod=space&uid=3366368&do=blog&id=1079996)

nohup g16 lig\_epikout.gjf &

查看log文件，正常结束



* Antechamber提取小分子电荷

antechamber -i lig\_epikout.log -fi gout -o ligand.prep -fo prepi -c resp -nc 0

输出小分子电荷信息

awk '$1=="ATOM" {print $3, toupper($10) "-" $9}' ANTECHAMBER\_PREP.AC

* 生成小分子力场信息

parmchk -i ligand.prep -f prepi -o ligand.frcmod -a y

parmchk2 -i ligand.prep -f prepi -o ligand.frcmod -a y

* 蛋白处理
* 为了避免后面出现结构错误，我觉得应该补全结构（swiss-model），注意保留对应的alt
* 加氢

Pdq2pqr30生成的pqr文件没有总的电荷信息，所以用pdb2pqr

conda activate pdb2pqr

pdb2pqr\_cli --ff=AMBER --ffout=AMBER --chain --with-ph=7.35 model\_01.pdb protein.pqr

pdb2pqr --ff=AMBER --ffout=AMBER --chain --with-ph=7.35 model\_01.pdb protein.pqr

head protein.pqr

#记住总的电荷，后面要用

obabel -ipqr protein.pqr -opdb -O pro\_H.pdb

* 复合物

合并加H之后的小分子和蛋白质

obabel -ipdb ../protein/pro\_H.pdb ../ligand/lig\_epikout.pdb -opdb -j -O complex.pdb

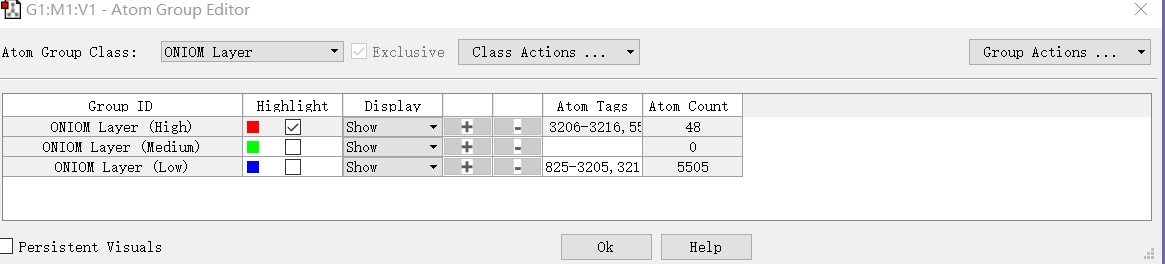
* GV设置高低层

将小分子和有互作的氨基酸设置为高层，其余为低层

高低层的连接处会自动显示连接

在tools– PDB Residues中将不敢兴趣的氨基酸残基都隐藏起来，只show感兴趣的残基，方便进行鼠标操作。

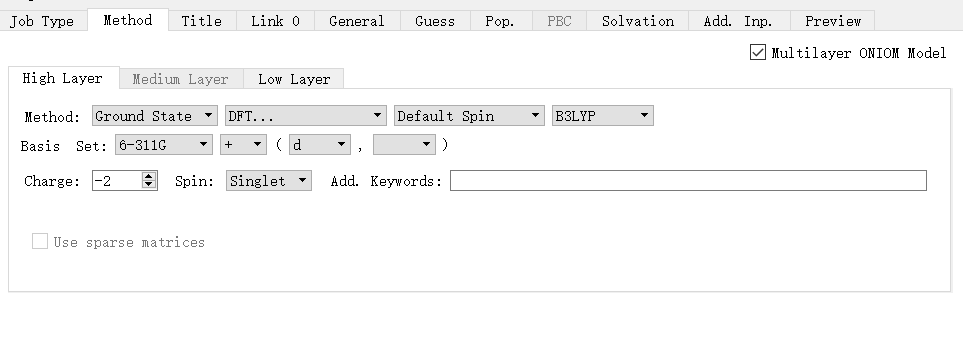
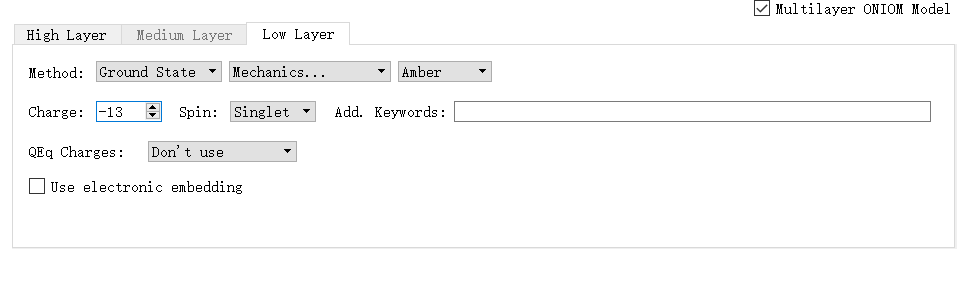
Edit-Atom groups-ONIOM Layer，用鼠标选中要增加到高层的原子，点击 ONIOM Layer （high）行的“+”即可。

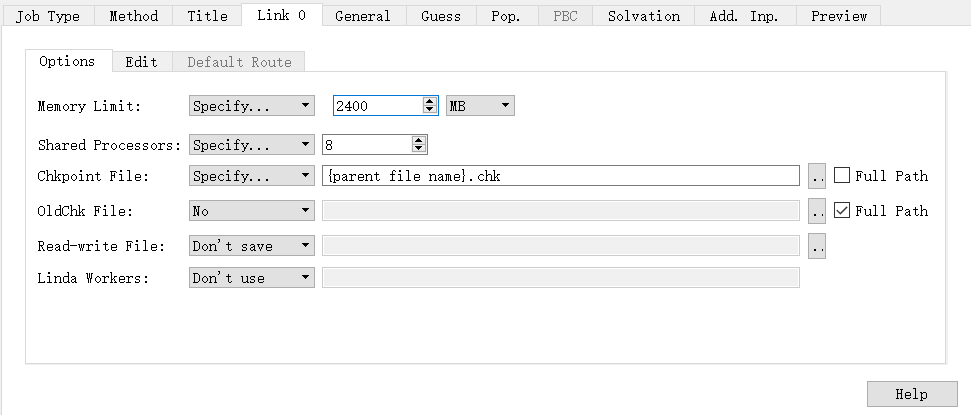


设置GJF参数：

1. Calculate > Gaussian Calculation Setup…
2. Job type: Optimization
3. Method: Multilayer ONIOM Model;
4. High layer: DFT > B3LYP > 6-311G+(d), charge根据具体体系定，保证spin（多重度）为1
5. Low layer: Mechanics > Amber, 其实后面在文本里还要修改(打开文件后，calculate里gaussian calculation setup中的Method里勾选右上角的Multilayer ONIOM Model，Low Layer里的method第二行选择Mechanics，第三行选Amber，这样才能在文本文件中出现原子电荷和原子类型的信息。)
6. Title自定
7. Link: Memory Limit(2400MB); Chk file(specify, 去掉full path); Shared Processors(4)

Retain, save gjf, 勾选Write Cartesians



* 存成gjf文件并修改文件开头

%nprocshared=64

%mem=2GB

%chk=complex.chk

# opt oniom(b3lyp/6-31g(d):amber=hardfirst)

* Gjf文件处理：
* **查找缺失的原子、电荷**

cp complex.gjf complex-bak.gjf

缺失的原子类型：

grep -n '\-(' complex.gjf

缺失的电荷：

grep -n '[^0-9](' complex.gjf

grep -n '[^0-9][0-9](' complex.gjf

* 补全参数
* **补齐标准氨基酸 缺失的电荷、原子**

打开标准氨基酸电荷信息库：（202）

/home/zjxu/prog/amber16/dat/leap/lib/all\_amino94.lib （标准氨基酸）

/home/zjxu/prog/amber16/dat/leap/lib/all\_aminoct94.lib （位于C末端的标准氨基酸）

/home/zjxu/prog/amber16/dat/leap/lib/all\_aminont94.lib （位于N末端的标准氨基酸）

* **补齐小分子缺失的电荷、原子信息**

grep -n '[^0-9](' complex.gjf

awk '$1=="ATOM" {print $3, toupper($10) "-" $9}' ../ligand/ANTECHAMBER\_PREP.AC|cut -f2 -d " " > lig.info

将ANTECHAMBER\_PREP.AC中小分子的电荷和原子类型替换到gjf文件中相应的位置

小分子原子类型和电荷信息全部缺失的情况下，可以用以下命令进行替换，先读取对应行进行替换，再对整行进行替换：

先确认小分子的信息是从哪一行开始的(把3185这个数字改成对应的行数见1)

奇奇怪怪，shell变量居然会把行首的空格去掉

grep -n 'ResName=UNL' complex.gjf

num=`cat lig.info|wc -l`;for i in `seq 1 $num`;do info=`sed -n ${i}p lig.info`;tar\_num=`expr $i + 3185`;sub=`sed -n ${tar\_num}p complex.gjf|sed "s/-.\*(PDBName.\*)/-${info}/g"|sed 's/ /@/g'`;sed -i "${tar\_num}c $sub" complex.gjf;sed -i 's/@/ /g' complex.gjf;done

补齐小分子的原子半径、键链信息，加载gjf文件最后。

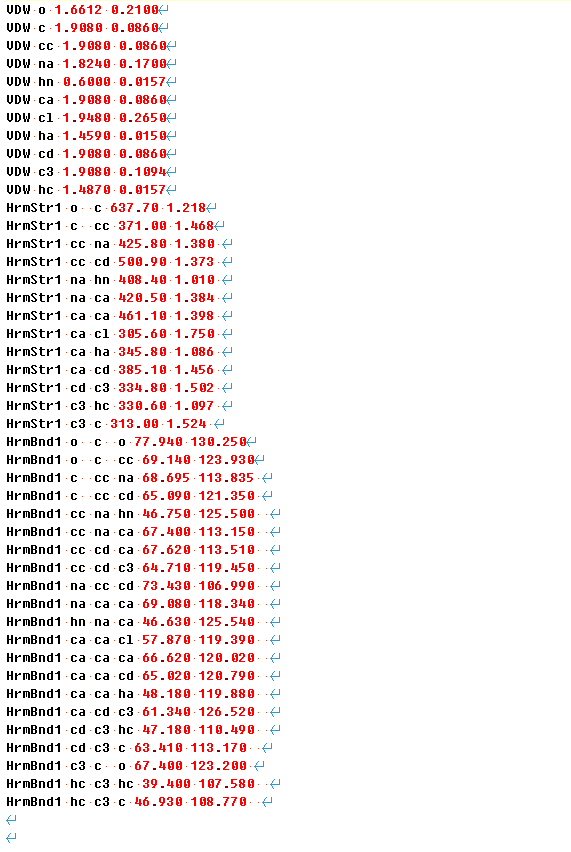
小分子的力场信息输入完之后需要加上一两行回车，否则报错。

awk '/NONBON/, /NR/' ../ligand/\*mod | awk '{print "VDW", $1, $2, $3}' >> complex.gjf

awk '/^BOND/, /ANGLE/' ../ligand/\*mod | awk '{print "HrmStr1", $1, $2, $3, $4}' | tr '-' ' '>> complex.gjf

awk '/^ANGLE/, /DIHE/' ../ligand/\*mod | awk '{print "HrmBnd1", $1, $2, $3, $4, $5}' | tr '-' ' ' >> complex.gjf

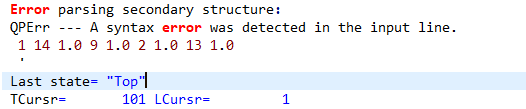
并删除不必要的部分



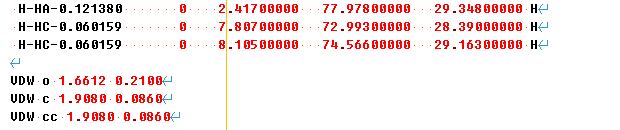
* 删除不必要的信息

sed "s/(PDBName.\*)//g" complex.gjf > complex\_del.gjf

syntax error报错：

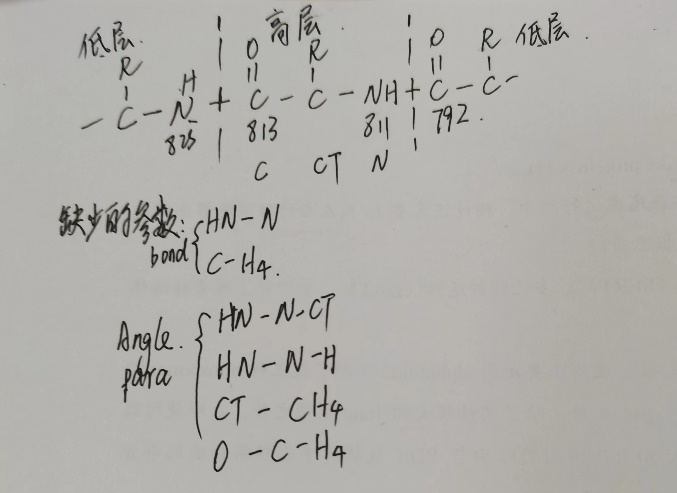


把原子信息后面的那些连接信息删了，再加上一行空行。



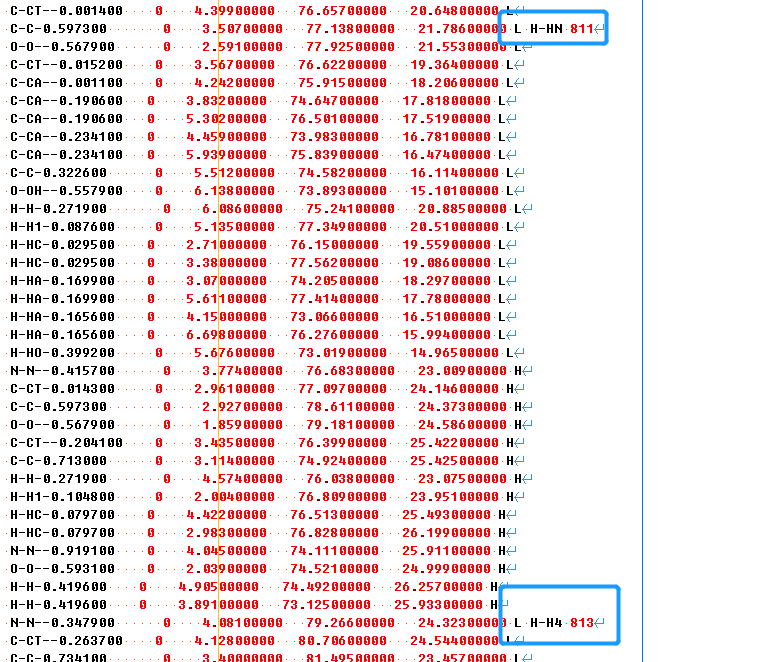
* 高低层连接处修改

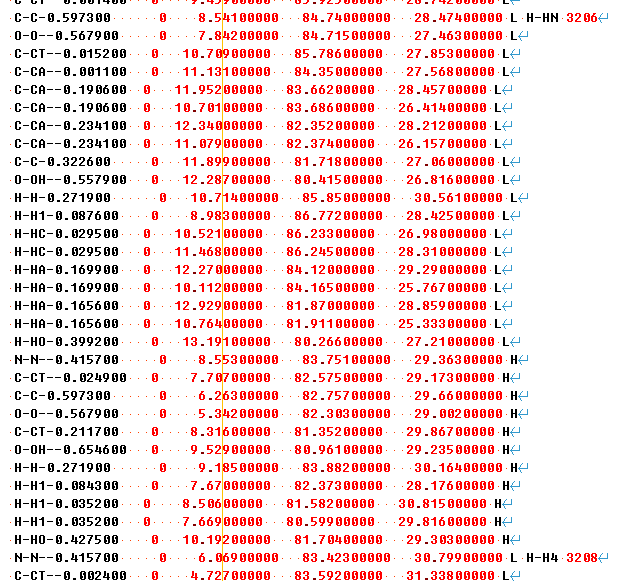
高低层断开的地方需要一个H原子来封闭高层，所以825、792处的原子应该变成一个H来封闭高层。



根据/home/g16/software/amber16/dat/leap/parm/gaff.dat对原子类型的定义以及与H成键的高层基团，对gjf文件L、H连接处做如下修改：

813上加个H，应该是醛基，H定义为H4，N的H叫HN





改了之后，先试试跑跑

nohup g16 complex\_del.gjf &

报错：Include all MM classes

Bondstretch undefined between atoms 792 811 HN-N [L,H]

Bondstretch undefined between atoms 813 825 C-H4 [H,L]

Bondstretch undefined between atoms 3187 3206 HN-N [L,H]

Bondstretch undefined between atoms 3208 3217 C-H4 [H,L]

Angle bend undefined between atoms 792 811 812 HN-N-CT [L,H,H]

Angle bend undefined between atoms 792 811 817 HN-N-H [L,H,H]

Angle bend undefined between atoms 812 813 825 CT-C-H4 [H,H,L]

Angle bend undefined between atoms 814 813 825 O-C-H4 [H,H,L]

Angle bend undefined between atoms 3187 3206 3207 HN-N-CT [L,H,H]

Angle bend undefined between atoms 3187 3206 3212 HN-N-H [L,H,H]

Angle bend undefined between atoms 3207 3208 3217 CT-C-H4 [H,H,L]

Angle bend undefined between atoms 3209 3208 3217 O-C-H4 [H,H,L]

MM function not complete

Error termination via Lnk1e in /home/g16/prog/g16/l120.exe at Sun Nov 20 20:31:32 2022.



按照以上办法封闭高层之后这些，会缺失相应的参数，在/home/g16/software/amber16/dat/leap/parm/gaff.dat中找到后，手动补在gjf文件后面

Gjf中的原子与gaff中的原子类型对应关系：

(肽键上的N是sp2杂化)n-na ct-c3

* 补L、H连接处的参数：

HN-N 408.4 1.0100 (对应gaff中的hn-na)

C-H4 310.7 1.1121 (对应gaff中的c-h4)

HN-N-CT 45.80 117.68 (对应gaff中的c3-n-hn, c3-na-hn的参数没找到)

HN-N-H 39.87 116.80 (对应gaff中的hn-na-hn)

CT-C-H4 45.64 114.64 (对应gaff中的c3-c-h4)

O-C-H4 54.22 120.70 (对应gaff中的h4-c-o)

加载gjf后面：

HrmStr1 hn n 408.4 1.0100

HrmStr1 c h4 310.7 1.1121

HrmBnd1 hn n ct 45.80 117.68

HrmBnd1 hn n h 39.87 116.80

HrmBnd1 ct c h4 45.64 114.64

HrmBnd1 o c h4 54.22 120.70

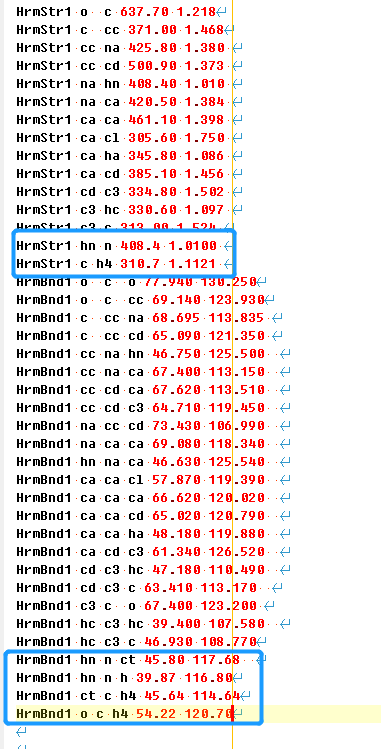
断侧链，补参数的原理同上

看报错信息，缺啥补啥



HrmBnd1 h1 ct h1 39.180 109.550

HrmBnd1 h1 ct sh 53.660 109.210



nohup g16 complex\_del.gjf &

* QM/MM优化结构报错

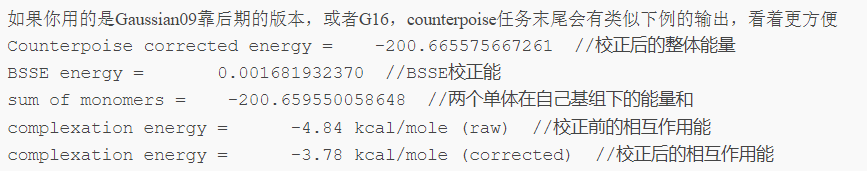
Linear angle in Tors.

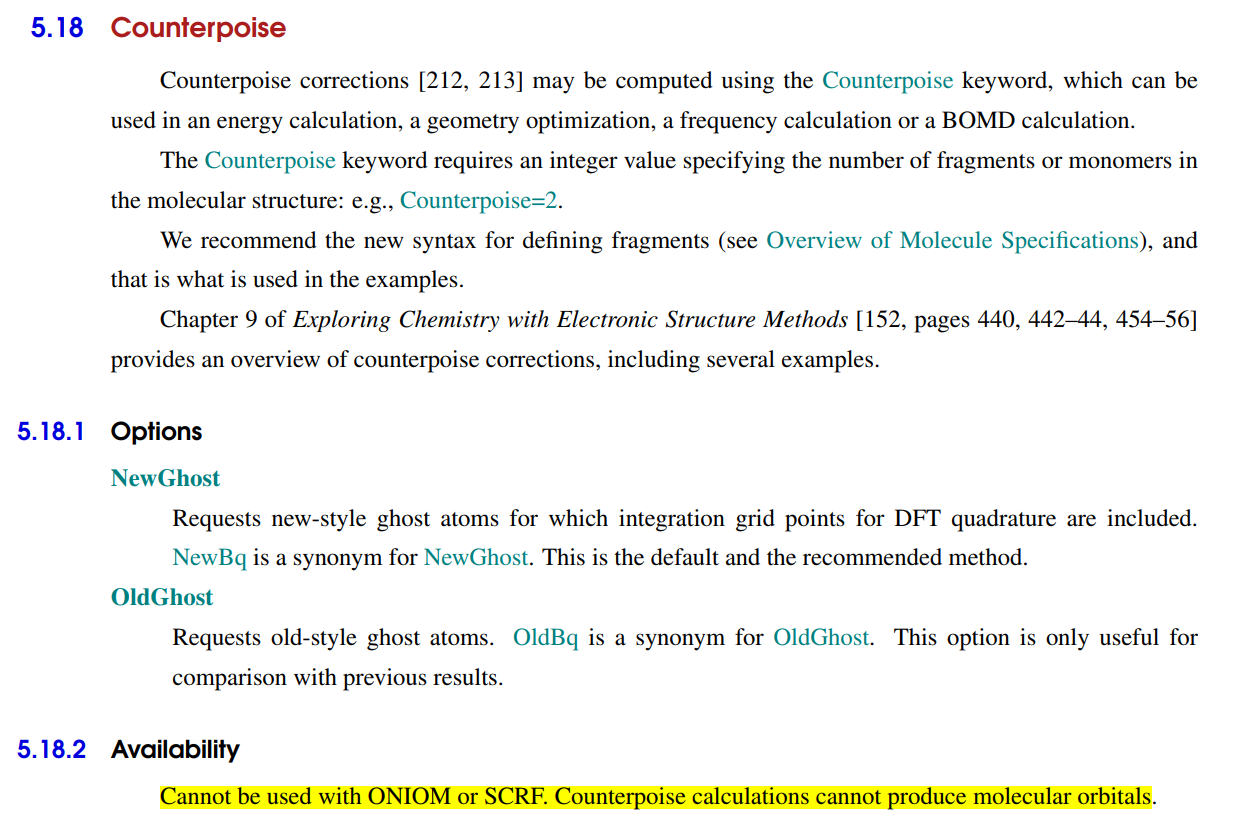
Error termination via Lnk1e in /home/chpeng/software/g16/g16/l103.exe at Thu Dec 1 17:08:39 2022.

多加了几个残基进高层也没有解决

* 复合物QMMM优化后的分析
* 参考资料

[谈谈BSSE校正与Gaussian对它的处理 - 思想家公社的门口：量子化学·分子模拟·二次元 (sobereva.com)](http://sobereva.com/46)





如果只是计算真空中的结合能的话，是可以直接用最后一步bsse的处理方式算complexation energy。

但由于counterpoise与溶剂不能共用，所以计算溶剂中的结合能的话，还是需要分开一步步走滴！！！

* 检查复合物能量收敛情况

GV打开gauss输出的log文件（注意勾选 Read Intermediate Geometries以查看不同帧）

保存优化的最后一帧

在GV最后一帧界面保存gjf文件即可

* 提取gjf文件中的高层原子（蛋白AA+小分子lig）

文本处理方法选取以H结尾的行（即高层原子的标记）

注意：不要漏选 L H连接的原子；把关键字添上（否则在gv中无法打开）

head -n 7 3n5j\_opt.gjf > high\_layer.gjf

grep H 3n5j\_opt.gjf |grep -v H- >> high\_layer.gjf

* 在上一步的基础上，用GV将与低层相连的C变为H，保存gjf文件

删除C：右键builder->delete atom ->选中C

添加H：右键builder->add valence ->选中连接H的C

(注意：添加的氢原子序号为最后一个，在分离AA、lig的时候要注意这个H被分到正确的文件)

在修改好的hlayer\_comp.gjf文件中提取AA（蛋白质）高层、lig（小分子）高层的gjf文件，分成com、lig、AA

* 修改comp gjf文件，计算能量

删除原子类型和电荷:

cat \*.gjf | tail -n +9 |awk '{print substr($1,1,1) "\t" $3 "\t" $4 "\t" $5}'>hlayer\_comp\_del.gjf

或者分开处理aa和lig之后再合并为com

cat ../aa/hlayer\_aa\_del.gjf | tail -n +9 > hlayer\_com\_del.gjf;cat ../lig/hlayer\_lig\_del.gjf | tail -n +9 >> hlayer\_com\_del.gjf

注意：只有原子为一位表示的才可以这么删除原子类型和电荷

在del文件中添加关键字：

%nprocshared=8

%mem=2400MB

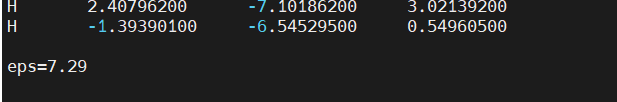
%chk=com.chk

# SCRF (pcm, solvent= generic, read) m062x/6-311+g(d) maxdisk=200GB scf=tight

com

添加电荷和自旋多重度

在原子信息的最后面，隔一行添加介电常数，如果是真空环境的话，关键字中黄色高亮部分和介电常数去除



检查gjf文件

注意最后一行为空行

nohup g16 hlayer\_comp\_del.gjf &

* 修改AA gjf文件，计算能量

删除原子类型和电荷:

cat \*.gjf | tail -n +9 |awk '{print substr($1,1,1) "\t" $3 "\t" $4 "\t" $5}'>hlayer\_aa\_del.gjf

注意：只有原子为一位表示的才可以这么删除原子类型和电荷

在del文件中添加关键字：

%nprocshared=4

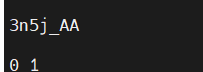
%mem=2400MB

%chk=aa.chk

# SCRF (pcm, solvent= generic, read) m062x/6-311+g(d) maxdisk=200GB scf=tight

aa

添加电荷和自旋多重度：



检查gjf文件

nohup g16 hlayer\_aa\_del.gjf &

* 修改lig gjf文件

删除原子类型和电荷:

cat \*.gjf | tail -n +9 |awk '{print substr($1,1,1) "\t" $3 "\t" $4 "\t" $5}'>hlayer\_lig\_del.gjf

注意：只有原子为一位表示的才可以这么删除原子类型和电荷，CL和BR需要手动改！！

在del文件中添加关键字：

%nprocshared=8

%mem=2400MB

%chk=lig.chk

# SCRF (pcm, solvent= generic, read) m062x/6-311+g(d) maxdisk=200GB scf=tight

lig

添加电荷和自旋多重度



检查gjf文件

nohup g16 hlayer\_lig\_del.gjf &

* 修改bsse gjf文件

删除原子类型和电荷:

cat ../aa/hlayer\_aa\_del.gjf |tail -n +9|awk '{print $1 "\t" $2 "\t" $3 "\t" $4 "\t" "1"}'>hlayer\_bsse\_del.gjf

cat ../lig/hlayer\_lig\_del.gjf |tail -n +9|awk '{print $1 "\t" $2 "\t" $3 "\t" $4 "\t" "2"}'>>hlayer\_bsse\_del.gjf

在del文件中添加关键字：

%nprocshared=8

%mem=2400MB

%chk=bsse.chk

# m062x/6-311+g(d) maxdisk=200GB scf=tight counterpoise=2

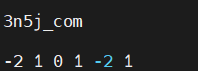
bsse

添加电荷和自旋多重度：

整体、片段1、片段2的电荷和自旋多重度都一样的情况下，只需写一次即可

若整体与片段的电荷和自旋多重度不一样，应该写：整体电荷，整体自旋多重度，片段

1电荷，片段1自旋多重度，片段2电荷，片段2自旋多重度



检查gjf文件

nohup g16 hlayer\_bsse\_del.gjf &

* 计算小分子与氨基酸作用能

E(interaction) = E(AB) + E(BSSE) – E(A) – E(B)

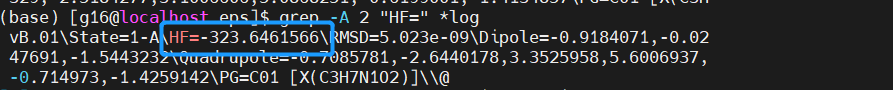
1 hartree = 2625.5 kJ mol⁻¹ = 27.21 eV/个 = 627.51kcal mol⁻¹

grep -A 2 "HF=" ./aa/hlayer\_aa\_del.log

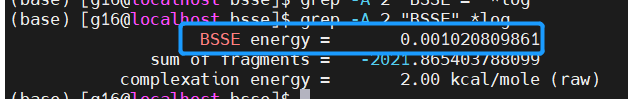
grep -A 2 "HF=" ./lig/hlayer\_lig\_del.log

grep -A 2 "HF=" ./com/hlayer\_com\_del.log

grep -A 3 "BSSE" ./bsse/hlayer\_bsse\_del.log

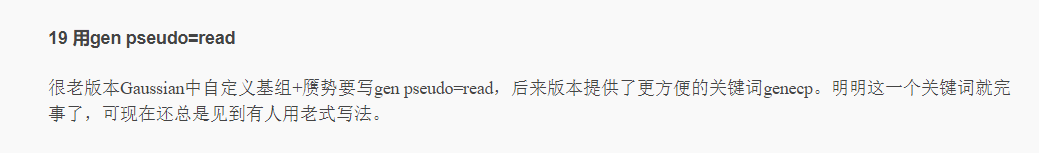


grep -A 2 "BSSE" ./bsse/hlayer\_bsse\_del.log



* 含I体系的处理

含碘的体系有些特别，6-31g\*对碘没有定义，因此需要用混合基组进行优化，gjf参数需要自己补，其他的跟普通体系是一样的。混合基组就是指不同的原子用不同的基组。写gen关键词代表从坐标部分后面读入基组定义。



[科学网—QM 计算小分子RESP电荷 补重元素半径 - 陈照强的博文 (sciencenet.cn)](https://blog.sciencenet.cn/home.php?mod=space&uid=950202&do=blog&id=941228)

* 小分子优化

其他的处理步骤跟普通的体系是一样的，但是gjf文件有些特别，需要手动修改

Gjf文件开头：

%chk=ligand

%mem=12000Mb

%nprocshared=12

#opt hf genecp geom=connectivity pop=(mk,readradii) iop(6/33=2,6/42=6)

3mbl

0 1

Gjf文件结尾空一行，加上自定义的混合基组、赝势以及范德华半径（元素要写全）：

C H O N 0

6-31G(d)

\*\*\*\*

I 0

SDD

\*\*\*\*

I 0

SDD

I,1.98

I,1.98

* QM/MM优化

Gjf文件开头：

-------------------------------------------------------------------------------

%chk=complex.chk

%mem=8000Mb

%nproc=16

# opt oniom(b3lyp/genecp:amber=hardfirst)

Title Card Required

-5 1 0 1 0 1

Gjf文件结尾空一行，加上自定义的混合基组、赝势以及范德华半径（元素要写全）：

-------------------------------------------------------------------------------

C H O N F S 0

6-31G(d)

\*\*\*\*

I 0

LANL2DZ

\*\*\*\*

I 0

LANL2DZ

-------------------------------------------------------------------------------

* 计算能量
* Lig

%nprocshared=8

%mem=2400MB

%chk=lig.chk

# SCRF (pcm, solvent= generic, read) m062x/genecp maxdisk=200GB scf=tight

lig

0 1

C H O N F 0

6-311+g(d)

\*\*\*\*

I 0

SDD

\*\*\*\*

I 0

SDD

eps=10.3

-------------------------------------------------------------------------------

* com

%nprocshared=8

%mem=2400MB

%chk=com.chk

# SCRF (pcm, solvent= generic, read) m062x/genecp maxdisk=200GB scf=tight

com

0 1

结尾

C H O N F 0

6-311+g(d)

\*\*\*\*

I 0

SDD

\*\*\*\*

I 0

SDD

eps=10.3

* bsse

%nprocshared=8

%mem=2400MB

%chk=bsse.chk

# m062x/genecp maxdisk=200GB scf=tight counterpoise=2

bsse

0 1 0 1 0 1

结尾

C H O N F 0

6-311+g(d)

\*\*\*\*

I 0

SDD

\*\*\*\*

I 0

SDD