

李锦添：参考自徐老师和正诞师兄的QMMM笔记

总体步骤：

1. 获取pdb晶体结构

小分子：

2. 从pdb晶体结构中见感兴趣的小分子（amber内置力场无法描述）加氢后存为pdb（*ligand.pdb*）；

（注：pymol加氢有的时候会有问题，可与maestro的加氢互相切换使用，尤其注意芳环上的氢）

3. 将小分子pdb转为mol2，schrodinger预测结晶pH下小分子的质子化状态（*ligprep*）；

4. 存出*ligprep*后的小分子（一个就可以，*ligand_ligprep.sdf*），在gaussian中用HF/6-31G(d)优化（因为gaussian内置amber力场比较古老，用更高级的方法基组意义不大），用pop=mk求电荷，加上iop(6/33=2,6/42=6)，具体关键词详见徐老师教程；

5. 用antechamber和parmchk提取上一步中正常计算的log文件中的原子类型、电荷和力场参数。（有时antechamber会提取不到电荷，可以直接从log文件读取）

蛋白质：

6. 将蛋白质结构中所有的配体、cofactor、水（重要水除外，可另存为新的文件，方便脚本操作）、离子等删去（*protein_del_lig.pdb*）；

（注：若是多聚体，可只保留一个；注意是否有alternative confirmation）

7. 用pdb2pqr预测蛋白质在结晶pH下的质子化状态，生成pqr文件（*protein_del_lig_addH.pqr*）；

8. 将pqr文件用openbabel转为pdb文件（*protein_del_lig_addH.pdb*）；

复合物：

9. 将（*ligand.pdb*）和（*protein_del_lig_addH.pdb*）及重要结晶水（如果有的话）用openbabel合并为1个pdb文件（*complex.pdb*）；

10. 将此pdb文件用GaussView打开，确定高低场（Edit中的PDB residues和Atom groups功能）；

11. 在Calculate中设置计算参数（见徐老师教程），存为gjf文件（*complex.gjf*）；

12. 在gjf文件中找到小分子的位置，补全小分子的原子类型和电荷；

13. 在gjf文件的末尾加上小分子的力场参数；

14. 如有可预判的力场参数缺失（例如截断面上的角度参数），可提前补全（*gaff.dat*）；

15. 提交*complex.gjf*，解决报错；

延伸：

TAO软件包固定外层原子？

1. 计算前小分子准备

1.1 小分子加氢

注意检查加氢后的小分子是否正确，pymol的加氢有时候会出问题（可以用maestro加氢），保存小分子pdb文件

从5TI2 pdb文件中提取的小分子自带的氢是正确的，可以直接用了。

1.2 小分子格式转换（pdb→mol2）

在pymol中直接保存小分子的mol2格式（在chem3D中检查没有问题）

1.3 预测小分子质子化状态（205）

```
chmod +x 01_ligprep.sh
```

```
./01_ligprep.sh -l 5Tl2_ligand.mol2 -o 5Tl2_ligand_ligprep.sdf
```

1.4 GV生成gjf文件

将上一步生成的 `5Tl2_ligand_ligprep.sdf` 文件导入GV, 保存出gjf文件

进一步为gjf文件添加、修改内容:

```
%chk=3b2i_ligand
%mem=2000Mb
%nproc=4
# opt hf/6-31g(d) geom=connectivity pop=mk iop(6/33=2,6/42=6, 6/50=1)
```

1.5 gaussian 优化小分子

```
g16 Tl2_ligand_ligprep.gjf
```

2. 计算前蛋白准备

2.1 保存蛋白polymer结构

2.2 pdb2pqr对蛋白进行质子化处理

202有pdb2pqr

```
pdb2pqr --ff=amber --ffout=amber --chain --with-ph=7.5 5Tl2_protein.pdb 5Tl2_protein.pqr
```

2.3 利用obabel将pqr格式转换为pdb格式

```
obabel -ipqr 5Tl2_protein.pqr -opdb -O 5Tl2_protein_pqr.pdb
```

3. 计算前复合物准备

3.1 obabel将小分子pdb (1.1) 和蛋白pdb (2.3) 连接起来 (205服务器)

```
obabel -ipdb 5Tl2_protein_pqr.pdb ../ligand/5Tl2_ligand.pdb -opdb -j -O
../complex/5Tl2_comp.pdb
```

3.2 gaussview确定复合物高低场 (onion layer)

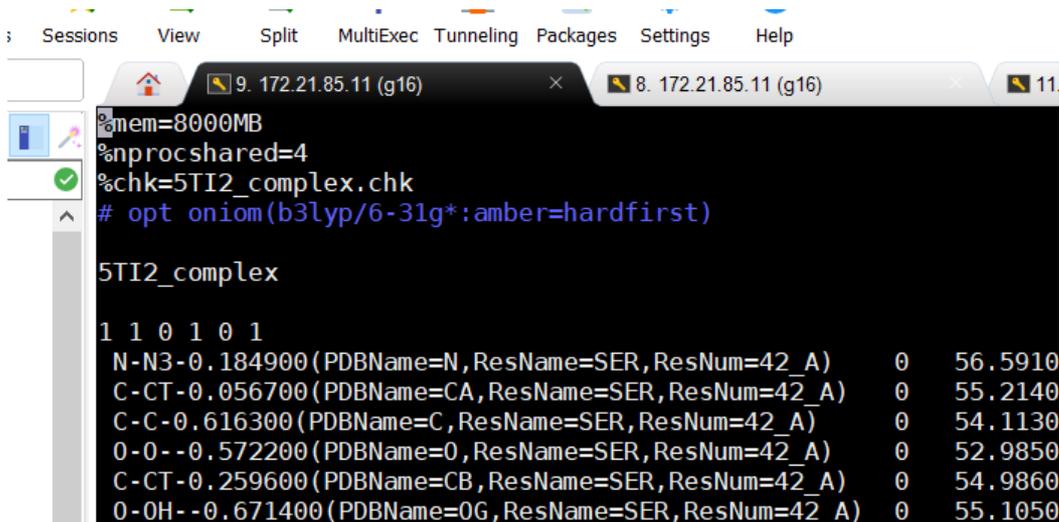
将5Tl2_comp.pdb导入gaussview。由于甲硫氨酸的侧链和小分子形成卤键, 所以把侧链和小分子设为高层, 其他设为底层。

通过查看目标的原子序号，直接在gaussview的atom tags中添加高低层原子序号，再通过选中局部微调。

3.3 GV优化结构

Calculate > Gaussian Calculation Setup...

3.4 修改gjf文件头



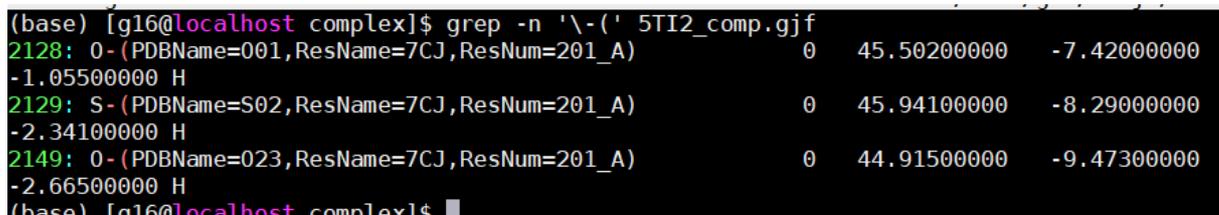
```
Sessions View Split MultiExec Tunneling Packages Settings Help
9. 172.21.85.11 (g16) 8. 172.21.85.11 (g16) 11.
%mem=8000MB
%nprocshared=4
%chk=5TI2_complex.chk
# opt oniom(b3lyp/6-31g*:amber=hardfirst)

5TI2_complex

1 1 0 1 0 1
N-N3-0.184900 (PDBName=N, ResName=SER, ResNum=42_A) 0 56.5910
C-CT-0.056700 (PDBName=CA, ResName=SER, ResNum=42_A) 0 55.2140
C-C-0.616300 (PDBName=C, ResName=SER, ResNum=42_A) 0 54.1130
O-O--0.572200 (PDBName=O, ResName=SER, ResNum=42_A) 0 52.9850
C-CT-0.259600 (PDBName=CB, ResName=SER, ResNum=42_A) 0 54.9860
O-OH--0.671400 (PDBName=OG, ResName=SER, ResNum=42_A) 0 55.1050
```

3.5 查找缺失的原子、电荷

缺失的原子: `grep -n '\-(' *.gjf`



```
(base) [g16@localhost complex]$ grep -n '\-(' 5TI2_comp.gjf
2128: O- (PDBName=001, ResName=7CJ, ResNum=201_A) 0 45.50200000 -7.42000000
-1.05500000 H
2129: S- (PDBName=S02, ResName=7CJ, ResNum=201_A) 0 45.94100000 -8.29000000
-2.34100000 H
2149: O- (PDBName=023, ResName=7CJ, ResNum=201_A) 0 44.91500000 -9.47300000
-2.66500000 H
(base) [g16@localhost complex]$
```

缺失的电荷: `grep -n '^[^0-9](' *.gjf`

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

```
(base) [g16@localhost complex]$ grep -n '[^0-9](' 5TI2_comp.gjf
4:# opt oniom(b3lyp/6-31g*:amber=hardfirst)
1703: O-0M(PDBName=0,ResName=PR0,ResNum=142_A)      0  55.62900000  -0.84500000  -
11.31300000 L
1714: C-CT(PDBName=CA,ResName=GLY,ResNum=143_A)    0  53.27700000   0.54900000  -
11.32300000 L H 2158
1715: C-C(PDBName=C,ResName=GLY,ResNum=143_A)     0  52.36700000  -0.59500000  -
10.94200000 L
1716: O-0(PDBName=0,ResName=GLY,ResNum=143_A)    0  51.33500000  -0.75900000  -
11.56100000 L
1717: H-HC(PDBName=HA2,ResName=GLY,ResNum=143_A)  0  52.73900000   1.35500000  -
11.38100000 L
1718: H-HC(PDBName=HA3,ResName=GLY,ResNum=143_A)  0  53.64300000   0.36500000  -
12.20200000 L
2128: O-(PDBName=001,ResName=7CJ,ResNum=201_A)   0  45.50200000  -7.42000000  -
-1.05500000 H
2129: S-(PDBName=S02,ResName=7CJ,ResNum=201_A)   0  45.94100000  -8.29000000  -
2.24100000 H
```

3.6 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_aminoc94.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)

3.7 补齐小分子缺失的电荷、原子

antechamber 提取小分子电荷 (210) (跑很快)

```
antechamber -i 5TI2_ligand_ligprep.log -fi gout -o 5TI2_ligand_ligprep.prep -fo prepi -c resp
-nc 0
```

输出原子类型和电荷信息：

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

3.8 添加小分子立场信息

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

VDW后加上frcmod文件中NONBON里的信息， Hrmstr1后加上BOND里的信息， HrmBnd1后加上ANGLE里的信息。都增加到复合物的gjf文件后面。

```
awk '/NONBON/, /NR/' *mod | awk '{print "VDW", $1, $2, $3}'
```

```
awk '/^BOND/, /ANGLE/' *mod | awk '{print "HrmStr1", $1, $2, $3, $4}' | tr '-' ' '
```

```
awk '/^ANGLE/, /DIHE/' *mod | awk '{print "HrmBnd1", $1, $2, $3, $4, $5}' | tr '-' ' '
```

小分子的立场信息输入完之后需要加上一两行回车，否则报错。

4. gauss软件计算

```
nohup g16 5TI2_comp_fill.gjf &
```

5.复合物QMMM优化后的分析

5.1 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）

5.2 保存优化的最后一帧

在GV最后一帧界面保存gjf文件即可

5.3 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）

5.4 在GV中将低层相连的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）

5.5 在5.4中修改好的hlayer_comp.gjf文件中提取AA（蛋白质）高层、lig（小分子）高层的gjf文件

5.6 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
```

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=20
%mem=2400MB
%chk=5TI2_complex_opt.chk
# m062x/6-311+g(d) maxdisk=200GB scf=tight

5TI2_opt_complex
```

- 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，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

0 1

- 检查gjf文件

注意最后一行为空行

- **nohup g16 hlayer_comp_del.gjf &**

5.7 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=20
%mem=2400MB
%chk=5TI2_aa_opt.chk
# m062x/6-311+g(d) maxdisk=200GB scf=tight

5TI2_opt_aa
```

- 添加电荷和自旋多重度:

0 1

- 检查gjf文件

- **nohup g16 hlayer_aa_del.gjf &**

5.8 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
```

- 在del文件中添加关键字:

```
%nprocshared=20
%mem=2400MB
%chk=5TI2_lig_opt.chk
# m062x/6-311+g(d) maxdisk=200GB scf=tight
```

- 添加电荷和自旋多重度:

0 1

- 检查gjf文件

- **nohup g16 hlayer_lig_del.gjf &**

5.9 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=20
```

```
%mem=2400MB
```

```
%chk=5TI2_bsse_opt.chk
```

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

```
5TI2_opt_bsse
```

- 添加电荷和自旋多重度:

0 1

- 检查gjf文件

- **nohup g16 hlayer_bsse_del.gjf &**

5.10 计算小分子与氨基酸作用能

$$E(\text{interaction}) = E(\text{AB}) + E(\text{BSSE}) - E(\text{A}) - E(\text{B})$$

高斯常见报错及解决方法

https://github.com/liyuanhe211/Solution_for_Every_Gaussian_Error_Message/blob/main/%E4%B8%AD%E6%96%87%E7%89%88.md

<http://bbs.keinsci.com/thread-4829-1-1.html>

