#Docking manual

#Target: 11-beta-hydroxysteroid dehydrogenase 1

#Ligand: (2R)-N-(adamantan-2-yl)-1-(cyclopentylmethyl)pyrrolidine-2-carboxamide

* **建立Ligand檔案**

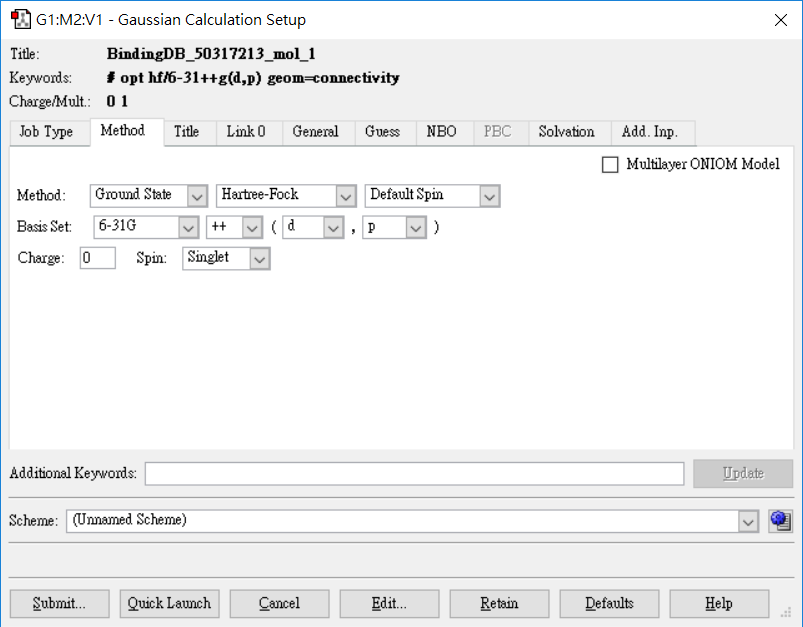
首先開啟GaussView5.0，讀取ligand的結構檔案(.sdf)。

點選右鍵→ Calculate→ Gaussian Calculation Setup進行設定。

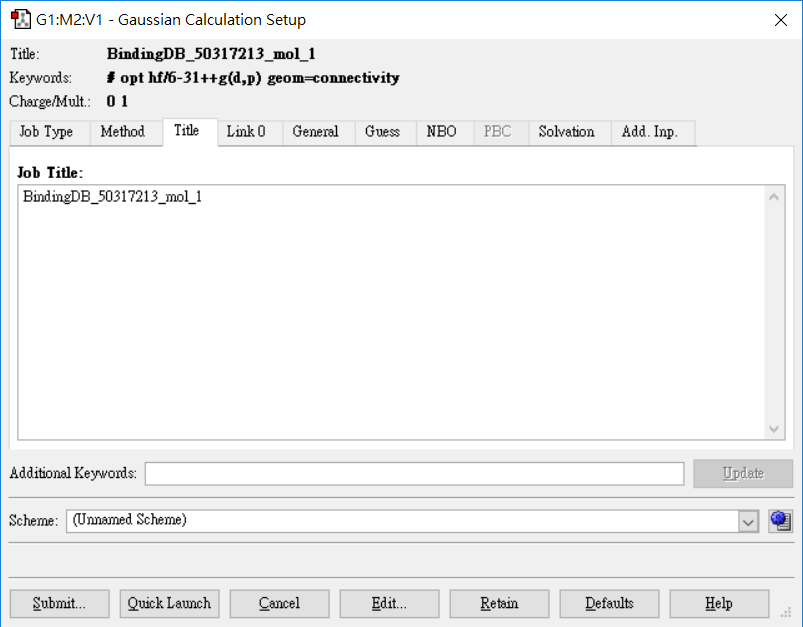
Job Type選單下拉選擇Optimization



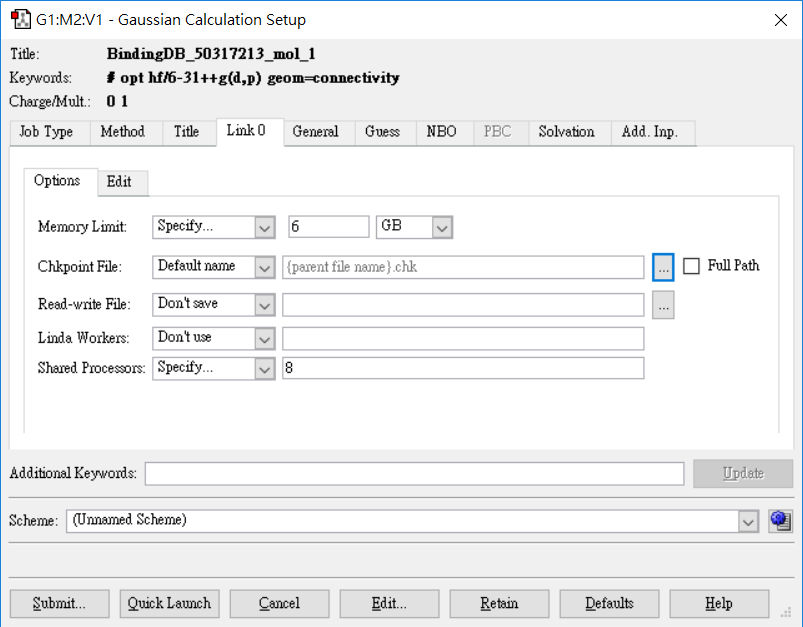
點選Method進行設定:



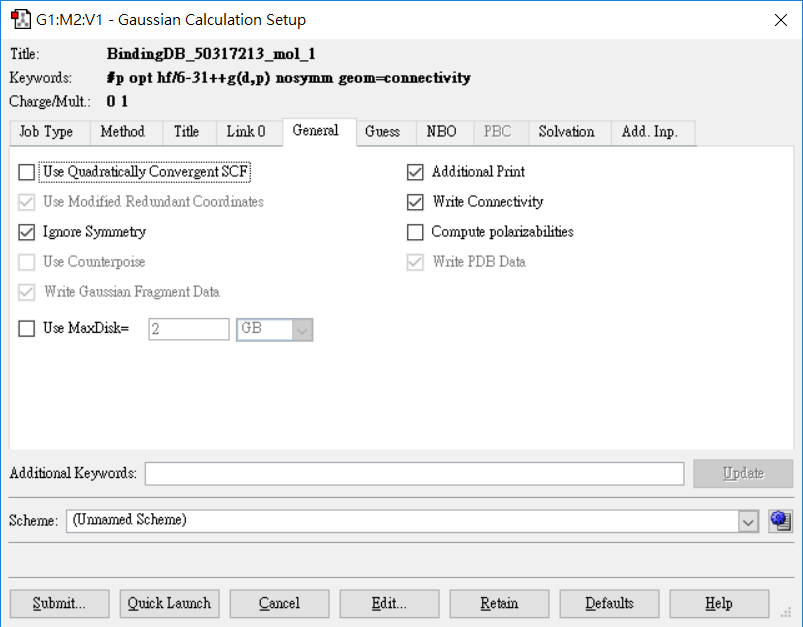
點選Title輸入Job Title



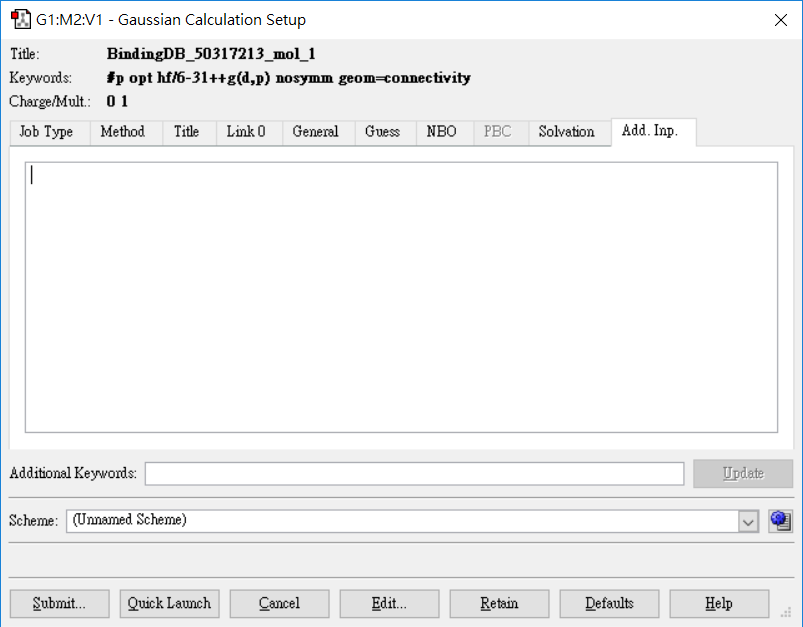
點選Link 0，取消勾選Full Path；Memory Limit預設為256MB可更改；Shared Processors選擇程式使用的CPU數



點選General設定如下

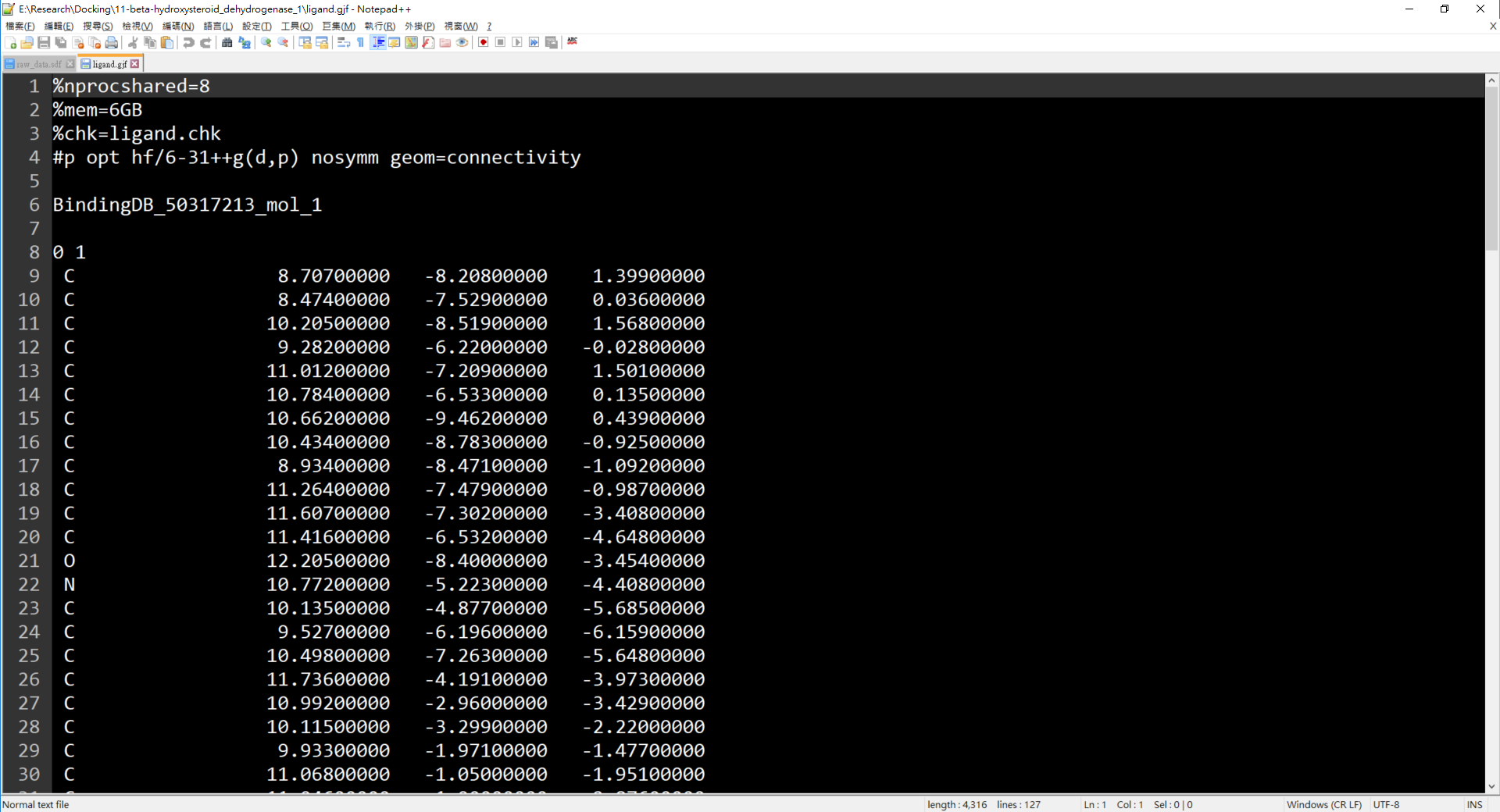


點選Add. Inp，scf=tight可設可不設



點選下方Edit→ Save ligand.gjf

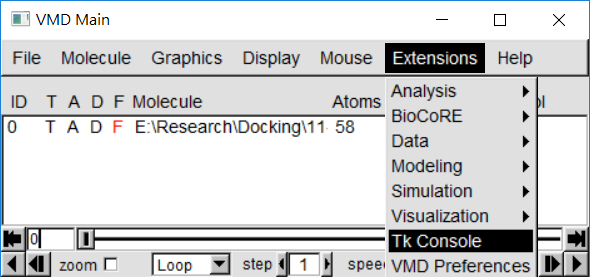
可以得到ligand.gjf檔



將ligand.gjf檔案送入機房電腦計算，指令為g09sub ligand，會跳出提示詢問運算CPU數目。

計算結果會輸出為ligand.log和ligand.chk檔，將檔案複製到使用電腦中，以GaussView開啟，按右鍵選擇File→ save，檔名存為ligand.pdb，save as: 選擇PDB File。

用VMD 1.9.2開啟ligand.pdb，選擇Extensions→ Tk Console。



輸入下列指令修改resname、resid和segname以進行後續分析。

set aaa [atomselect top “all”] aaa為變數名稱

$aaa set resname AAA 因為格式限制，atom name最多為三個字元

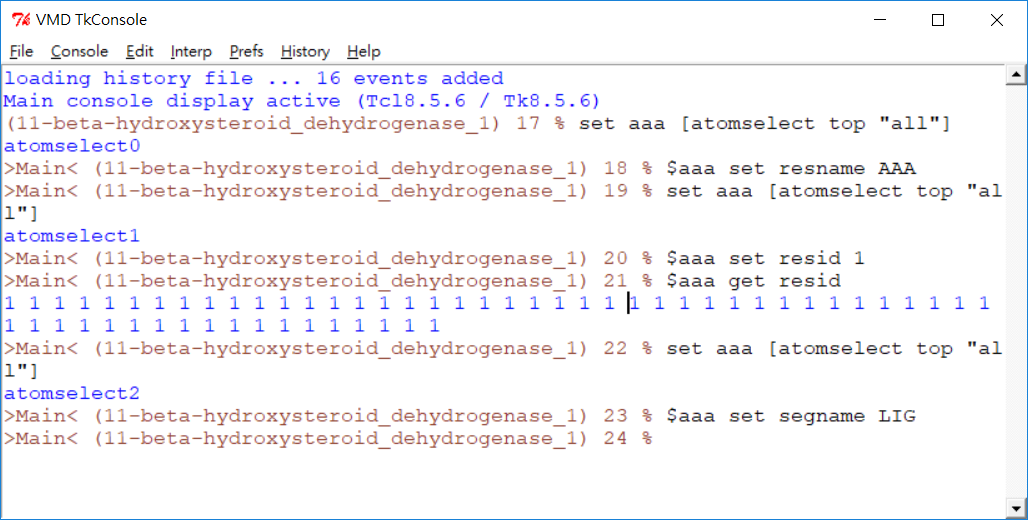
set aaa [atomselect top “all”]

$aaa set resid 1

$aaa get resid

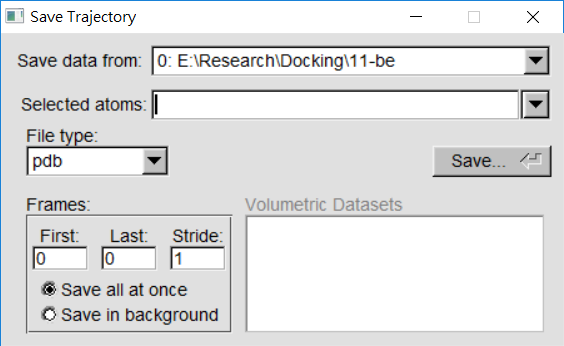
set aaa [atomselect top “all”]

$aaa set segname LIG



之後在VMD Main點選File→ Save Coordinates會跳出視窗，

File type選擇pdb後save，檔案名稱為drug.pdb。



接著要為drug.pdb裡的atom命名，建立文件”reorder\_drug\_pdb.pl”，code如下:

use strict;

($#ARGV != 0)?

die "Please give your pdb filename.\n":

open(INF, "$ARGV[0]") || die "Can't open $ARGV[0].\n";

my $order = 1;

my %num\_class;

while(my $line=<INF>){

if($line =~ /^ATOM/ || $line =~ /^HETATM/){

my @word=split(//, $line);

my $name;

for(my $i=13; $i<=15; $i++){

if($word[$i] ne ' '){

$name.=$word[$i];

}

else{

last;

}

}

if(exists $num\_class{$name}){

$num\_class{$name}++;

}

else{

$num\_class{$name} = 1;

}

my @reorder = split(//, $num\_class{$name});

for(my $i=13; $i<=15; $i++){

if($word[$i] eq ' '){

if($#reorder >=0){

$word[$i] = shift @reorder;

}

}

}

print @word;

}

else{

print $line;

}

}

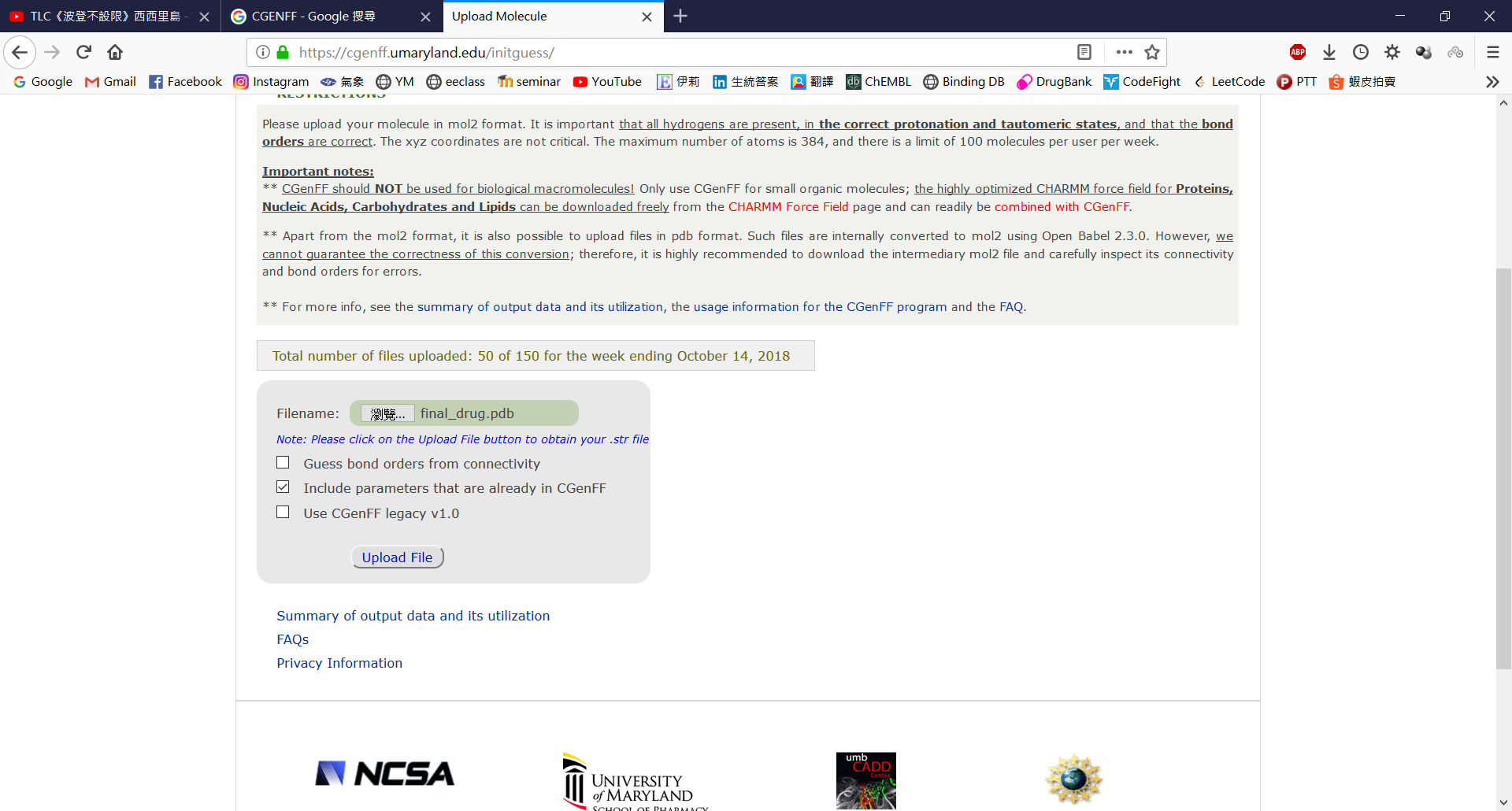
在工作資料夾開啟cmd輸入:

perl reorder\_drug\_pdb.pl input檔名 > output檔名

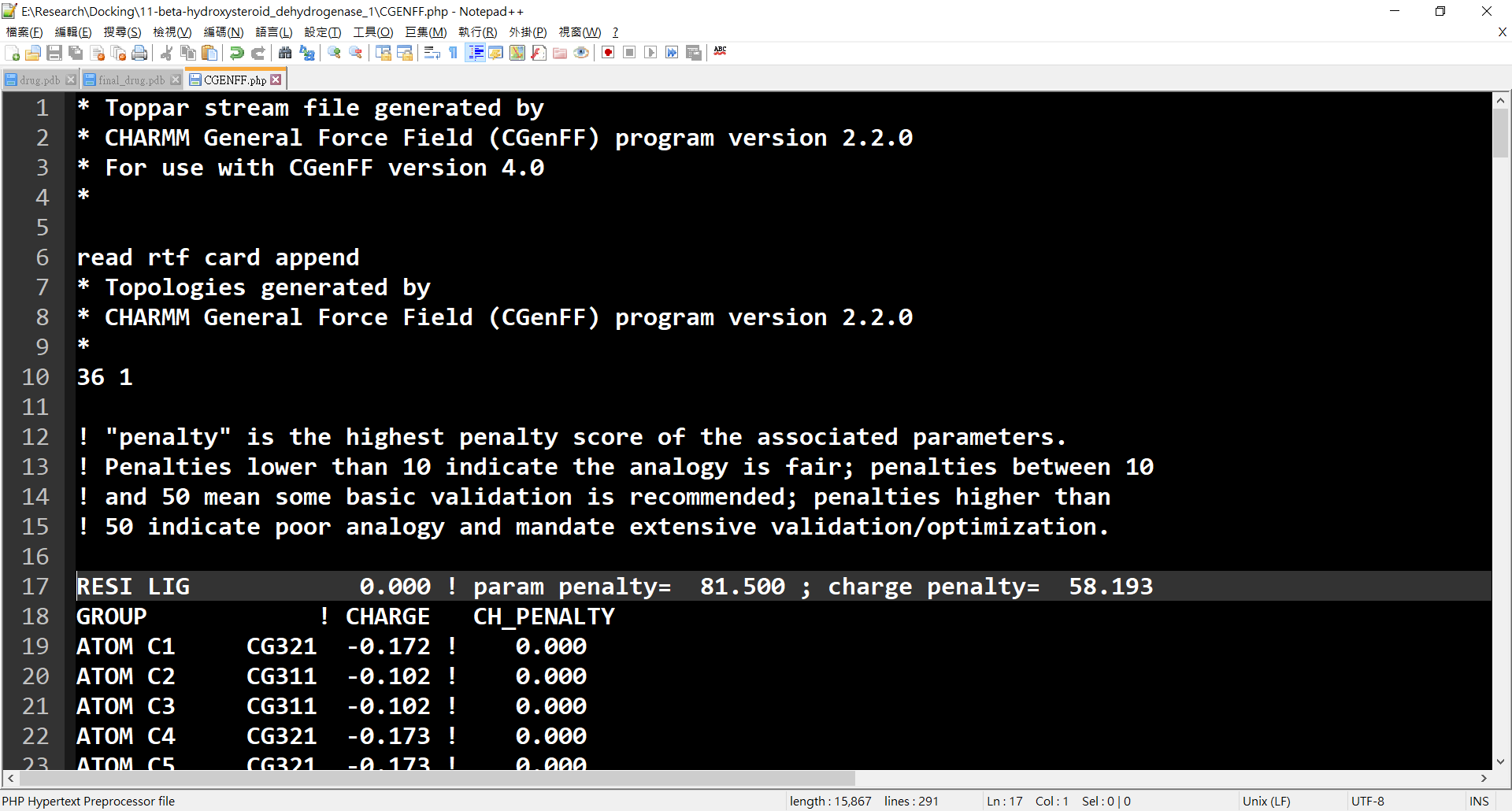
這邊輸出的pdb檔案會是最終藥物的pdb檔。

* **建立Topology和Parameter**
* 方法一：利用CGenFF建立

前往CGenFF網頁: <https://cgenff.umaryland.edu/>，點選上方選單中的Upload molecule登入後，上傳藥物pdb檔案，勾選Include parameters that are already in CGenFF後點選Upload File，會產生新的.str檔。

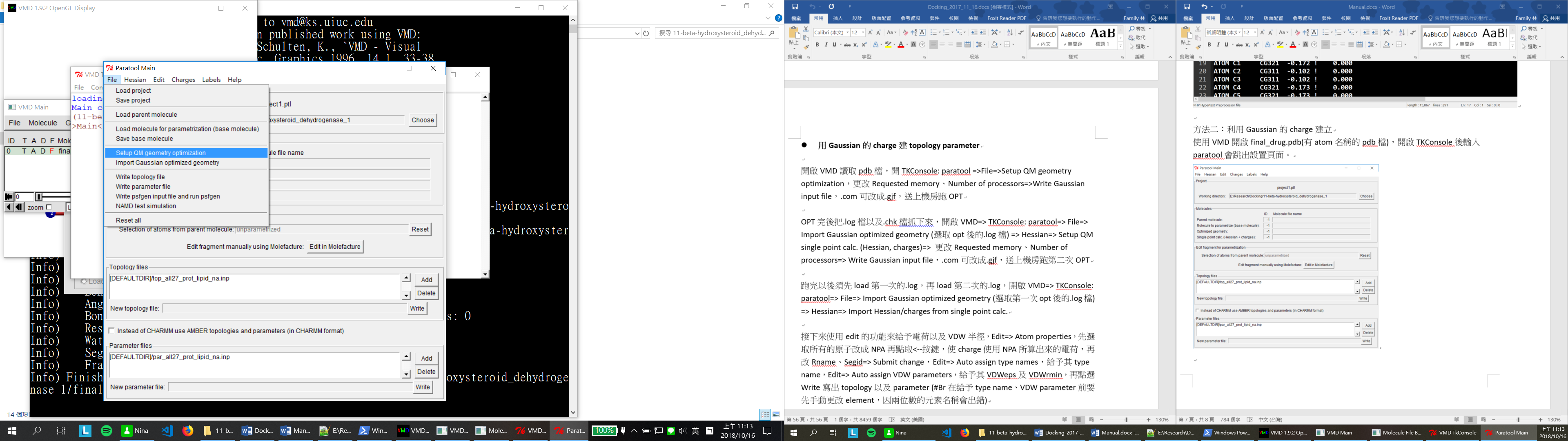


將str檔案下載後，前半部資訊為Topology，後半部資訊為Parameter，並將紅框處更改藥物名稱。

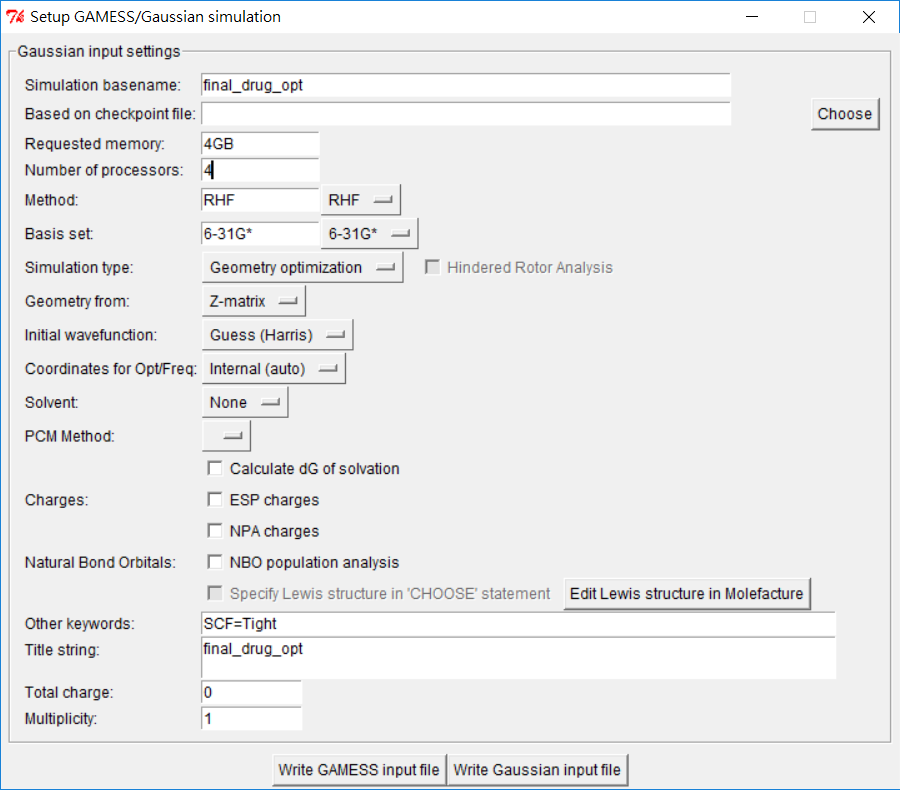


* 方法二：利用Gaussian的charge建立

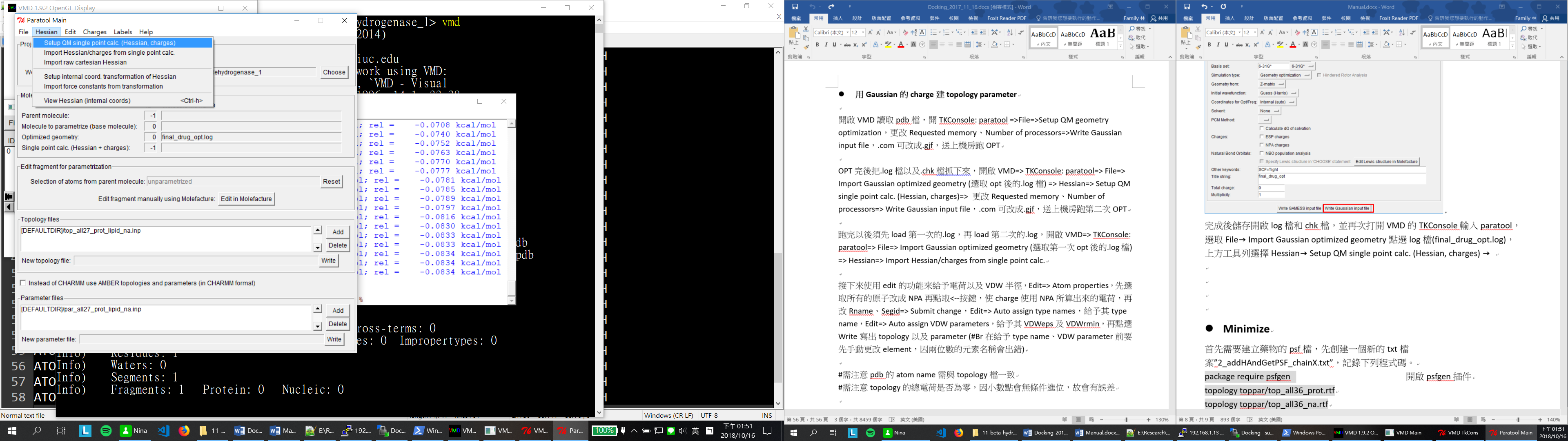
使用VMD開啟final\_drug.pdb(有atom名稱的pdb檔)，開啟TKConsole後輸入paratool會跳出設置頁面。



選擇File→ Setup QM geometry optimization，設定Requested memory、Number of processors後，點選Write Gaussian input file，存檔時預設副檔名為.com，需改為.gjf，接著使用機房電腦進行OPT。

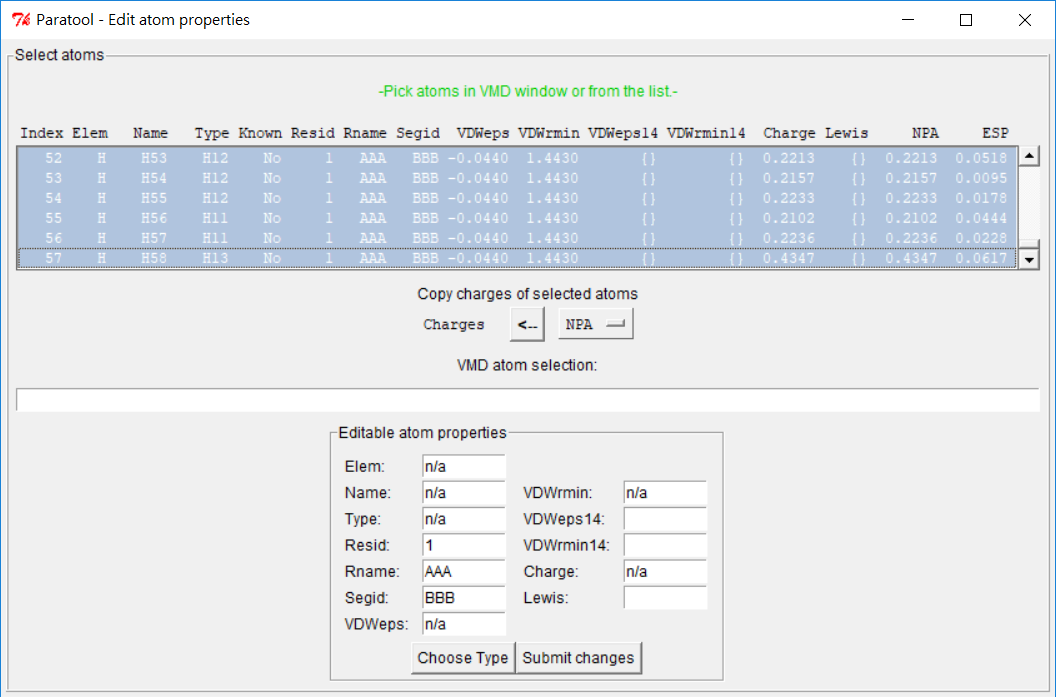


完成後儲存log檔和chk檔，並再次打開VMD的TKConsole輸入paratool，選取File→ Import Gaussian optimized geometry點選log檔(final\_drug\_opt.log)，上方工具列選擇Hessian→ Setup QM single point calc. (Hessian, charges)，設定使用記憶體大小以及運算CPU數後點選Write Gaussian input file存成final\_drug\_sp，副檔名.com改為.gjf，將final\_drug\_sp.chk和final\_drug\_sp.gjf送入機房電腦進行OPT。



開啟VMD，打開TKConsole後輸入paratool開啟設定視窗，點選File→ Import Gaussian optimized geometry，選擇第一次OPT的log檔(final\_drug\_opt.log)，接著點選Hessian→ Import Hessian/charges from single point calc.，選擇第二次OPT的log檔(final\_drug\_sp.log)。

下一步點選Edit→ Atom properties來設定電荷和VDW半徑，反白選取所有原子，下方Charges選單改為NPA，再點選左邊🡨按鈕，並在Rname和Segid欄位更改名稱，最後Submit changes。



點選Edit🡪 Auto assign type names給予type name，再點選Edit🡪 Auto assign VDW parameters給予VDWeps及VDWrmin，接著Write出topology和parameter檔案，輸出AAA.top和AAA.par檔案。

! 在給予type name和VDW parameter時遇到兩位數的元素名稱會出錯，所以要先手動更改element。

! pdb檔案的atom name要和topology檔案一致

! 要驗算topology的電荷總和是否為0。

* **Minimize**

首先需要建立藥物的psf檔，先創建一個新的txt檔案”2\_addHAndGetPSF\_chainX.txt”，記錄下列程式碼。

package require psfgen 開啟psfgen插件

topology toppar/top\_all36\_prot.rtf

topology toppar/top\_all36\_na.rtf

topology toppar/top\_all36\_lipid.rtf

topology toppar/top\_all36\_carb.rtf

#topology toppar/top\_all36\_cgenff.rtf 如果使用Gaussian自建topology則不需要

topology AAA.top 讀取Topology檔案

topology AAA.par 讀取parameter檔案

pdbalias residue HIS HSE

pdbalias residue CSW CYS

pdbalias residue CSO CYS

pdbalias atom ILE CD1 CD 修正結構中不同的標示名稱

segment X {pdb drug.pdb;first none;last none} 建立segment X並讀取pdb

coordpdb drug.pdb X 讀取pdb中的座標

guesscoord 根據top檔案補齊缺少座標

writepdb drug\_H.pdb

writepsf drug\_H.psf write寫出檔案

exit

開啟cmd，輸入vmd -e 02\_addHAndGetPSF\_chainX.txt，會得到一個pdb和一個psf檔(這裡是drug\_H.pdb和drug\_H.psf)。這兩個檔案用機房電腦進行minimize，使用下面的code(01\_mini.namd)進行。

structure drug\_H.psf

coordinates drug\_H.pdb

temperature 0

paraTypeCharmm on

parameters AAA.par

outputEnergies 1000

outputTiming 1000

xstFreq 1000

dcdFreq 1000

timestep 1

nonBondedFreq 2

fullElectFrequency 4

stepsPerCycle 20

switching on

switchDist 10

cutoff 12

pairlistdist 14

exclude scaled1-4

1-4scaling 1.0

binaryoutput off

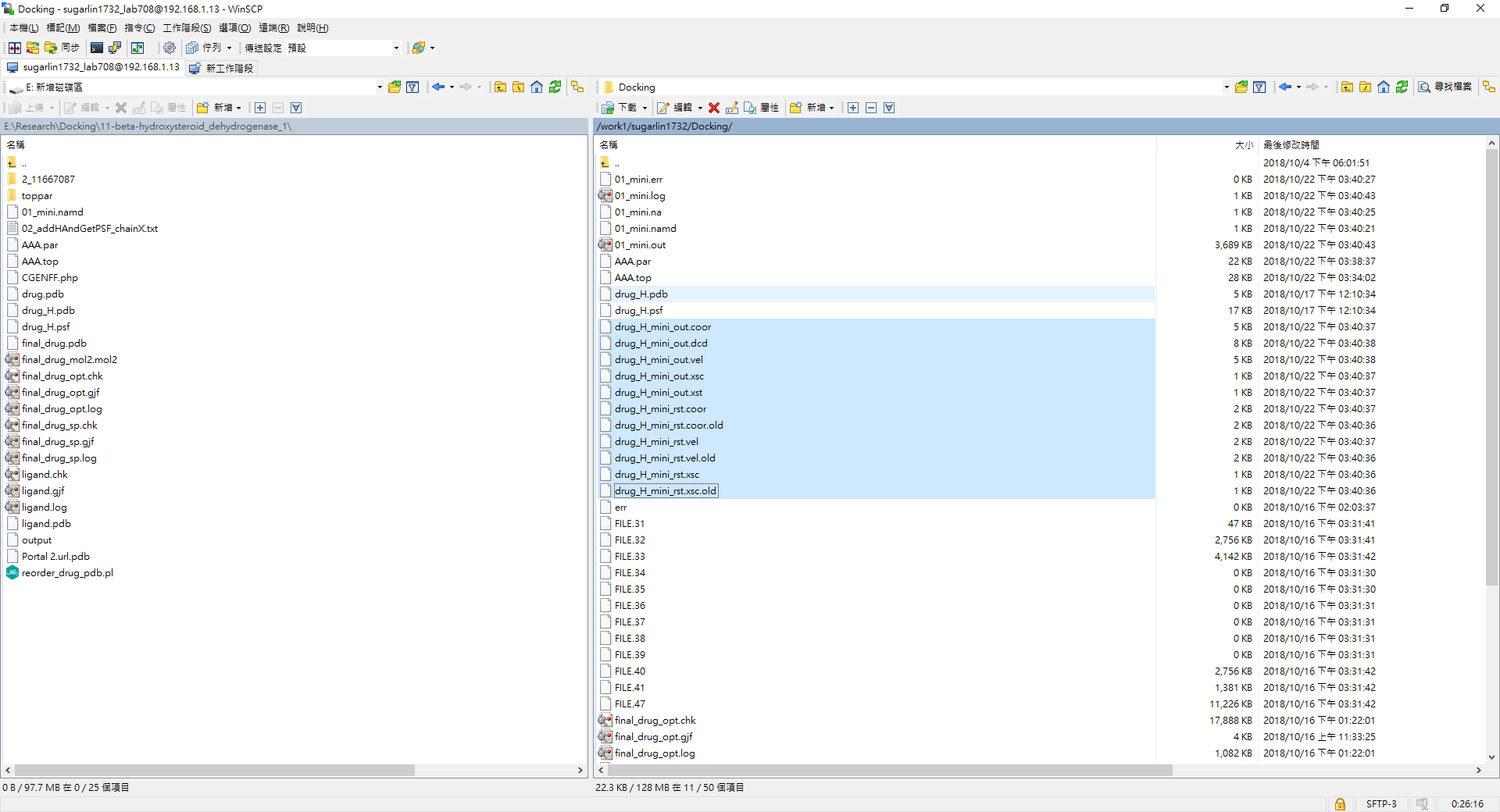
outputname drug\_H\_mini\_out

restartname drug\_H\_mini\_rst

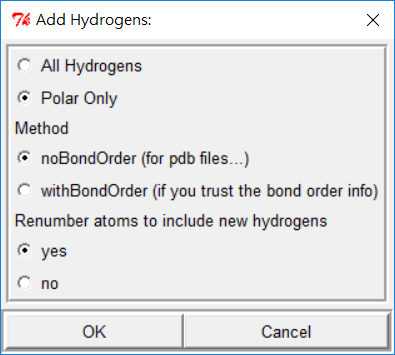
restartfreq 1000

minimize 10000

完成後會得到11個檔案，其中.dcd檔案為軌跡檔，而.coord檔則相當於minimize最後一步，最佳化的結構，將其更改為.pdb檔(🡪 drug\_H\_mini\_out.pdb)。



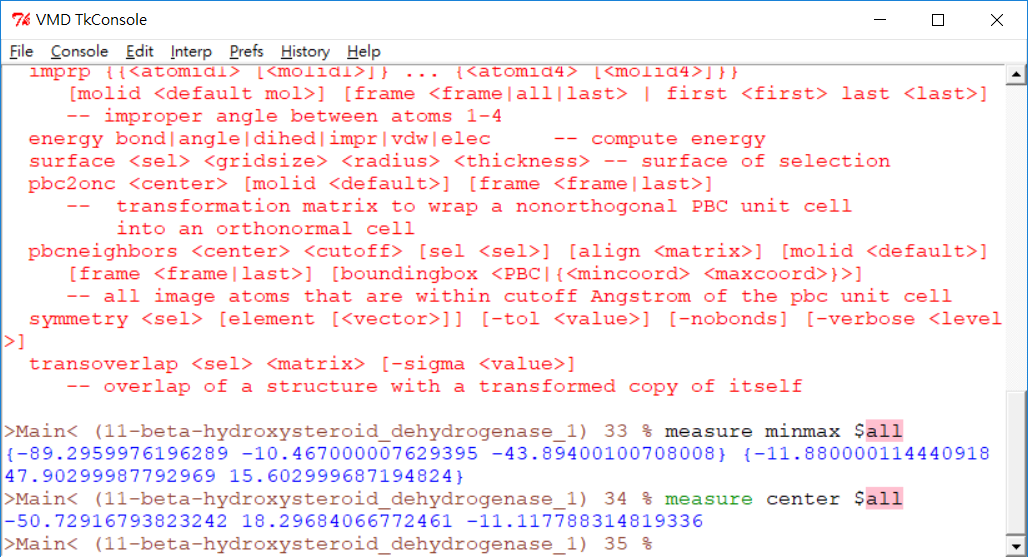
* **Docking**
* 需先安裝MGLTools 1.5.6和AutoDock Vina 1.1.2
* 使用MGLTools中的AutoDockTools準備AutoDock Vina需要的protein和ligand的.pdbqt檔
* 準備protein的.pdbqt檔

1. 開啟AutoDockTools，File🡪 Read Molecule🡪 選擇protein的pdb檔案
2. 點選Edit🡪 Delete Water以去除結晶水，再點選Edit🡪 Delete🡪 Delete Hydrogens去除Hydrogens
3. 因為X-ray解結構無法解出protein上的氫，因此要自行補回去，點選Edit🡪 Hydrogens🡪 Add🡪 Polar only以加上Polar(O,N)上的氫。
4. 計算protein的中心點以及大小範圍，開啟VMD後讀取protein的pdb檔，再開啟TkConsole輸入下列code。

set all [atomselect top "all"]

measure minmax $all 得到protein中原子最大及最小的座標

measure center $all 得到protein的中心座標



記錄下此座標以設定Grid Box，也就是之後docking所需要搜索的範圍。

1. 點選Grid🡪 Macromolecule🡪 Choose選擇protein pdb檔案的名稱後點選Select Molecule🡪 Save protein.pdbqt

* 準備ligand的.pdbqt檔

1. 開啟AutoDockTools，點選Ligand🡪 Input🡪 open，選擇要開啟的檔案(drug\_H.pdb)
2. 點選Ligand🡪 Torsion Tree🡪 Choose Torsion可以看到可以轉動和不能轉動的鍵結，點選Done
3. 點選Ligand🡪 Output🡪 Save as PDBQT，儲存為.pdbqt檔

* 使用Autodock Vina

首先建立文檔vina.conf.txt，內容如下。

receptor = protein.pdbqt 讀入protein的pdbqt

ligand =drug\_H\_mini\_out.pdbqt 讀入ligand的pdbqt

center\_x = 輸入中心點XYZ座標

center\_y =

center\_z =

size\_x = 輸入docking範圍的大小

size\_y =

size\_z =

out = vina.out.pdbqt 輸出的檔案名稱

log = vina.log.txt 記錄檔名稱

exhaustiveness = 8

num\_modes = 9 輸出結果的數量

energy\_range = 3 輸出結果的能量差距範圍

開啟VMD輸入指令”vina --config vina.conf.txt”後會產生vina.out.pdbqt和vina.log.txt兩個檔案，vina.out.pdbqt會有所有結果的ligand座標，而vina.log.txt是紀錄每個(9個)結果的affinity。

建立文檔” vinaPP.01.pl”，code如下，用來分割上面輸出檔vina.out.pdbqt的9個結果，並將他們存成9個pdb檔。

#!/bin/usr/perl

#vina post processing

#v.0.0.1-20100505

use strict;

if ($ARGV[0] eq "")

{

print "Please assign 1 parameters, when executing this script.\n";

print "\t1 the file name of input data.\m\n";

exit();

}

my $line;

my @words;

my @tmp;

open(INFILE, "$ARGV[0]");

while($line = <INFILE>)

{

chomp($line);

@words = split(/\s+/, $line);

if($words[0] eq "MODEL")

{

$tmp[0] = sprintf("%02d", $words[1]);

open(OUTFILE, "> $tmp[0].pdb");

while($line = <INFILE>)

{

@words = split(/\s+/, $line);

if($words[0] eq "ATOM")

{

print OUTFILE $line;

}

elsif($words[0] eq "ENDMDL")

{

print OUTFILE "END\n";

close OUTFILE;

last;

}

}

}

}

close INFILE;

開啟cmd輸入指令” perl vinaPP.01.pl vina.out.pdbqt”。

* **建立complex**

這步驟要將上述docking後得出的9個結果，將protein、protein上的金屬離子和ligand作為一組complex，分別建立出9個complex的psf和pdb檔案。

首先建立文檔” 02\_addHAndGetPSF\_AZX.txt”，code如下。

package require psfgen

topology toppar/top\_all36\_prot.rtf 讀取topology檔

topology toppar/top\_all36\_na.rtf

topology toppar/top\_all36\_lipid.rtf

topology toppar/top\_all36\_carb.rtf

topology toppar/toppar\_water\_ions.str

topology AAA.top

pdbalias residue HIS HSE

pdbalias atom ILE CD1 CD

segment A {pdb 3pdj\_A.pdb;first nter;last cter} 若protein有兩個chain

segment B {pdb 3pdj\_B.pdb;first nter;last cter} 則會有兩個segment

#segment Z {pdb zinc.pdb;first none;last none}

segment X {pdb 01.pdb;first none;last none} 進行9次記得更換檔名

#patch ZNSP Z:401 A:193 A:196 A:228 A:230

coordpdb 3pdj\_A.pdb A

coordpdb 3pdj\_B.pdb B

#coordpdb zinc.pdb Z

coordpdb 01.pdb X 進行9次記得更換檔名

guesscoord

writepdb protein\_ligand\_zn\_01.pdb

writepsf protein\_ligand\_zn\_01.psf

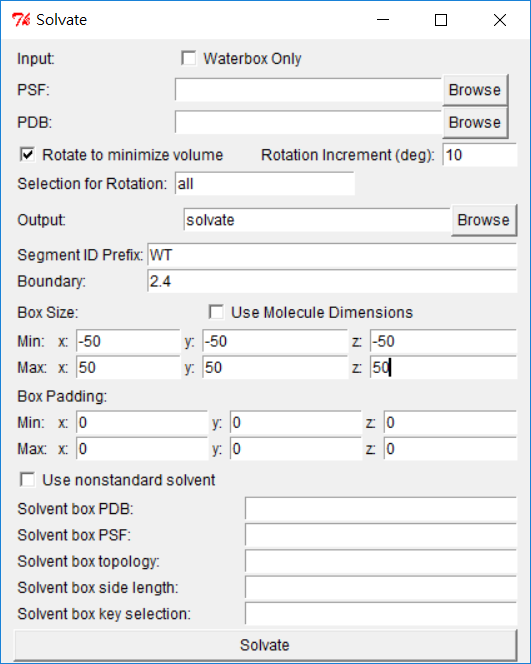
exit

需先確認protein有幾個chain並事先分割pdb檔案。

開啟cmd輸入指令” vmd -e 02\_addHAndGetPSF\_AZX.txt”，因為有九組complex需要建立，因此這步驟需要進行9次，每次記得更改讀入的檔名。

* 為complex建立水盒子以及離子

首先開啟VMD後讀取complex的psf和pdb檔(“protein\_ligand\_zn\_01.pdf/.pdb”)，建立水盒子Extensions🡪 Modeling🡪 Add Solvate Box，取消勾選Water Only，勾選Rotate to minimize volume，Box Size範圍最好比complex最長編再大20，點選Solvate，會輸出solvate.log、solvate.pdb以及solvate.psf。



接著點選Extensions🡪 Modeling🡪 Add Ions，input檔選擇solvate.pdb以及solvate.psf ，Choose salt可選擇添加的離子(常見為NaCl)，並選擇Only neutralize system with NaCl，點選Autoionize，會輸出ionized.pdb和ionized.psf。

* **分子動力學模擬**
* Minimize

首先建立minimize的script檔(“01\_complex\_mini.namd”)，code如下，。

structure ionized.psf

coordinates ionized.pdb

temperature 0

paraTypeCharmm on

parameters ../toppar\_charmm36/toppar/par\_all36\_prot.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_na.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_lipid.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_carb.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_cgenff.prm

parameters ../toppar\_charmm36/toppar/toppar\_water\_ions\_namd.str

#parameters zinc\_finger\_par.txt

parameters AAA.par

outputEnergies 1000

outputTiming 1000

xstFreq 1000

dcdFreq 1000

wrapAll on

wrapNearest on

timestep 1

nonBondedFreq 2

fullElectFrequency 4

stepsPerCycle 20

switching on

switchDist 10

cutoff 12

pairlistdist 14

cellOrigin 0 0 0

cellBasisVector1 100.00 00.00 00.00 設定box大小

cellBasisVector2 00.00 100.00 00.00

cellBasisVector3 00.00 00.00 100.00

Pme on

PmeGridsizeX 100 需大於或等於box大小

PmeGridsizeY 100 若為2、3、5的倍數運

PmeGridsizeZ 100 算速度會較快

exclude scaled1-4

1-4scaling 1.0

binaryoutput off

outputname complex\_mini\_out

restartname complex\_mini\_rst

##########

#constraints on 限制參數，依case而定

#consexp 10

#consref cons\_ref.pdb

#conskfile cons\_ref.pdb

#conskcol B

#constraintScaling 0.1

##########

restartfreq 1000

minimize 50000

將檔案送入機房，輸入指令namdsub 01\_complex\_mini計算，進行結構能量最佳化，會得到輸出complex\_mini\_out和complex\_mini\_rst檔案，副檔名為coor, dcd, vel, xsc, xst。

* Heating

Minimize完後要繼續進行加熱，建立02\_complex\_heat.namd，內容如下。

structure ionized.psf

coordinates ionized.pdb

bincoordinates complex\_mini\_rst.coor 為上一步minimize的輸出檔

binvelocities complex\_mini\_rst.vel 為上一步minimize的輸出檔

extendedSystem complex\_mini\_rst.xsc 為上一步minimize的輸出檔

paraTypeCharmm on

parameters ../toppar\_charmm36/toppar/par\_all36\_prot.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_na.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_lipid.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_carb.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_cgenff.prm

parameters ../toppar\_charmm36/toppar/toppar\_water\_ions\_namd.str

parameters AAA.par

outputEnergies 1000

outputTiming 1000

xstFreq 1000

dcdFreq 1000

wrapAll on

wrapNearest on

timestep 1

nonBondedFreq 2

fullElectFrequency 4

stepsPerCycle 20

switching on

switchDist 10

cutoff 12

pairlistdist 14

Pme on

PmeGridsizeX 100

PmeGridsizeY 100

PmeGridsizeZ 100

exclude scaled1-4

1-4scaling 1.0

langevin on

langevinDamping 1

langevinTemp 300

langevinHydrogen on

langevinPiston on

langevinPistonTarget 1.01325

langevinPistonPeriod 200

langevinPistonDecay 100

langevinPistonTemp 300

binaryoutput off

outputname complex\_heat\_out

restartname complex\_heat\_rst

##########

#constraints on

#consexp 10

#consref ../cons\_ref.pdb

#conskfile ../cons\_ref.pdb

#conskcol B

#constraintScaling 0.1

##########

restartfreq 1000

run 0

langevinTemp 100

langevinPistonTemp 100

run 1000

langevinTemp 120

langevinPistonTemp 120

run 1000

langevinTemp 140

langevinPistonTemp 140

run 1000

langevinTemp 160

langevinPistonTemp 160

run 1000

langevinTemp 180

langevinPistonTemp 180

run 1000

langevinTemp 200

langevinPistonTemp 200

run 1000

langevinTemp 220

langevinPistonTemp 220

run 1000

langevinTemp 240

langevinPistonTemp 240

run 1000

langevinTemp 260

langevinPistonTemp 260

run 1000

langevinTemp 280

langevinPistonTemp 280

run 1000

langevinTemp 300

langevinPistonTemp 300

run 1000

此步驟要進行加熱，從100K開始到300K，每1000步加熱20K。

* Equilibrium

加熱完要在300K進行平衡，進行兩次，每次各5000000步(5ns)

03\_complex\_equi.namd

structure ionized.psf

coordinates ionized.pdb

bincoordinates complex\_heat\_rst.coor

binvelocities complex\_heat\_rst.vel

extendedSystem complex\_heat\_rst.xsc

paraTypeCharmm on

parameters ../toppar\_charmm36/toppar/par\_all36\_prot.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_na.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_lipid.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_carb.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_cgenff.prm

parameters ../toppar\_charmm36/toppar/toppar\_water\_ions\_namd.str

parameters AAA.par

outputEnergies 1000

outputTiming 1000

xstFreq 1000

dcdFreq 1000

wrapAll on

wrapNearest on

timestep 1

nonBondedFreq 2

fullElectFrequency 4

stepsPerCycle 20

switching on

switchDist 10

cutoff 12

pairlistdist 14

Pme on

PmeGridsizeX 100

PmeGridsizeY 100

PmeGridsizeZ 100

exclude scaled1-4

1-4scaling 1.0

langevin on

langevinDamping 1

langevinTemp 300

langevinHydrogen on

langevinPiston on

langevinPistonTarget 1.01325

langevinPistonPeriod 200

langevinPistonDecay 100

langevinPistonTemp 300

binaryoutput off

outputname complex\_equi\_out

restartname complex\_equi\_rst

##########

#constraints on

#consexp 10

#consref ../../../cons\_ref.pdb

#conskfile ../../../cons\_ref.pdb

#conskcol B

#constraintScaling 0.1

##########

restartfreq 1000

run 5000000

04\_complex\_equi.namd

structure ionized.psf

coordinates ionized.pdb

bincoordinates complex\_equi\_rst.coor

binvelocities complex\_equi\_rst.vel

extendedSystem complex\_equi\_rst.xsc

paraTypeCharmm on

parameters ../toppar\_charmm36/toppar/par\_all36\_prot.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_na.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_lipid.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_carb.prm

parameters ../toppar\_charmm36/toppar/par\_all36\_cgenff.prm

parameters ../toppar\_charmm36/toppar/toppar\_water\_ions\_namd.str

parameters AAA.par

outputEnergies 1000

outputTiming 1000

xstFreq 1000

dcdFreq 1000

wrapAll on

wrapNearest on

timestep 1

nonBondedFreq 2

fullElectFrequency 4

stepsPerCycle 20

switching on

switchDist 10

cutoff 12

pairlistdist 14

Pme on

PmeGridsizeX 100

PmeGridsizeY 100

PmeGridsizeZ 100

exclude scaled1-4

1-4scaling 1.0

langevin on

langevinDamping 1

langevinTemp 300

langevinHydrogen on

langevinPiston on

langevinPistonTarget 1.01325

langevinPistonPeriod 200

langevinPistonDecay 100

langevinPistonTemp 300

binaryoutput off

outputname complex\_equi2\_out

restartname complex\_equi2\_rst

##########

#constraints on

#consexp 10

#consref ../../../cons\_ref.pdb

#conskfile ../../../cons\_ref.pdb

#conskcol B

#constraintScaling 0.1

##########

restartfreq 1000

run 5000000

利用產生的dcd檔案做後續分析。

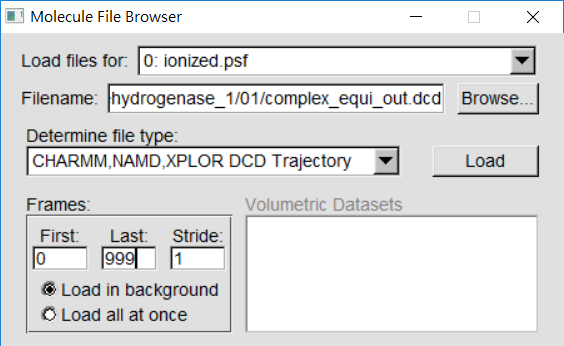
* **結果分析**

將dcd檔存回電腦，首先開啟VMD讀取ionized.psf，再讀取dcd。

也可以將dcd檔分割，將每1000個frame去水後再儲存為一個dcd檔，再分段分析。

* 分段步驟：

1. 點選File🡪 New Molecule🡪 Browse，選擇ionized.psf後load。
2. 再Browse選擇complex\_equi2\_out.dcd，Frames欄位First選擇0，Last選擇999，Stride為1，代表從地個Frame讀到第999個Frame，總共1000個，點選Load。



1. 接著將水去除後儲存，點選File🡪 Save Coordinates🡪 Selected atoms: chain A Z X🡪 File type: dcd🡪 Save，檔案類型選擇dcd。
2. 在Main視窗對Molecule點右鍵Delete Frames🡪 Delete
3. 重複步驟2~4並更改First和Last，每1000步存一次。

完成後重開VMD，先讀取已經去水的psf，再將分段的dcd依序讀取，最後另存為一個新的dcd。

* 可以利用RMSD來判斷結構是否已經達到平衡，點選Extensions🡪 Analysis🡪 RMSD Trajectory Tool，左上方輸入protein，勾選Backbone後點選ALIGN，會以protein做為中心固定Backbone，再將protein改為chain X (藥物)，將Backbone改為noh，點選RMSD，可以觀察藥物是否趨近平衡，若有則RMSD會穩定，勾選Plot可以作圖，勾選Save可以儲存RMSD結果。
* **自由能計算**

可以分為MM、NMA、APBS三部分

根據RMSD穩定度判斷，挑選dcd檔中較穩定的片段來計算，長度介於5ns~10ns，把這段dcd檔另外存出，frame調整為1001個frame。

自由能在計算時需要分別計算complex、ligand、protein的能量，因此需要將dcd以及psf檔分割成三部分，並且都要去水。

* 分割dcd檔

開啟VMD讀取ionized.psf後讀取complex\_equi\_out.dcd，在Main視窗對Molecule點選右鍵Save Coordinates。

存complex去水：Selected atoms輸入protein or chain X，File type選擇dcd後Save成complex\_no\_water.dcd。

存ligand去水：Selected atoms輸入chain X，File type選擇dcd後Save成drug\_no\_water.dcd。

存protein去水：Selected atoms輸入protein，File type選擇dcd後Save成protein\_no\_water.dcd。

* 分割psf檔

開啟VMD讀取ionized.psf和ionized.pdb，打開TkConsole輸入下列指令。

mol new ionized.psf type psf first 0 last -1 step 1 waitfor all 讀取檔案

mol addfile ionized.pdb type pdb first 0 last -1 step 1 waitfor all

set badwater [atomselect top "water and name OH2"]

set badion [atomselect top "ion"]

set badprotein [atomselect top "protein and name CA"]

#############################

package require psfgen

readpsf ionized.psf

coordpdb ionized.pdb

foreach segid [$badwater get segid] resid [$badwater get resid] {

delatom $segid $resid

}

foreach segid [$badion get segid] resid [$badion get resid] {

delatom $segid $resid

}

writepsf complex\_no\_water.psf 輸出去水complex的psf

############################

resetpsf

psfcontext reset

readpsf ionized.psf

coordpdb ionized.pdb

foreach segid [$badwater get segid] resid [$badwater get resid] {

delatom $segid $resid

}

foreach segid [$badion get segid] resid [$badion get resid] {

delatom $segid $resid

}

foreach segid [$badwater get segid] resid [$badwater get resid] {

delatom $segid $resid

}

writepsf drug\_no\_water.psf 輸出去水ligand的psf

############################

resetpsf

psfcontext reset

readpsf ionized.psf

coordpdb ionized.pdb

foreach segid [$badwater get segid] resid [$badwater get resid] {

delatom $segid $resid

}

foreach segid [$badion get segid] resid [$badion get resid] {

delatom $segid $resid

}

delatom X 1

writepsf protein\_no\_water.psf 輸出去水protein的psf

exit;

* MM (Molecular Mechanics)

需要將complex、protein、drug分開計算，