FMODD: フラグメント分子軌道法手引書　ver. 1

2020 年 4 月 1 日

古明地勇人、福澤薫、望月祐志、中野達也、沖山佳生、渡邊千鶴、加藤昭史、加藤幸一郎

謝辞:

本資料は、以下の講習会での講義資料を、自習用に作り直したものです。

「ABINIT-MP 講習会−TSUBAME3.0 を用いたハンズオン− 手引書 2020 年 1 月 28 日　東京工業大学」 後援　文部科学省　ポスト「京」重点課題(6)「革新的クリーンエネルギーシステムの実用化」。この講義資料の作成にご協力をいただいた、みずほ情報総研・塚本貴志博士と神戸大学・渡邉博文博士に感謝します。

fmodd.consortium@gmail.com

目次

0　序 ……2
1　課題 1：TrpCage の IFIE 解析と CHPI 解析 ……4
1.1　入力ファイル作成（AJF）……4
1.2　FMO 計算の実行 ……12
1.3　CHPI 解析 ……15
1.4　IFIE・PIEDA 解析……18
2　課題 2：エストロゲン受容体－リガンド複合体の IFIE 解析 ……27
2.1　入力データ作成 ……28
2.2　FMO 計算 ……28
2.3　IFIE 解析 ……29
3 課題３：Formamide と水の複合系の PIEDA 解析 ……33
3.1　入力ファイル作成 ……33
3.2　FMO 計算 ……33
3.3　IFIE・PIEDA 解析……34
4　課題 4：軌道レベルでの電荷移動相互作用解析（CAFI 解析）……37
4.1　入力データの作成 ……37
4.2　FMO 計算 ……41
4.3　CAFI 解析……42
5　課題 5：軌道レベルでの分極相互作用解析（FILM）……43
5.1　入力データの作成 ……43
5.2　FMO 計算 ……43
5.3　FILM 解析 ……44
まとめ ……48

0 序

この資料では、FMODD の会員向けに、Biostation Viewer（以下、Viewer）と ABINIT-MP の二つのプログラムを使って、FMO 計算を実習いたします。あらかじめ、二つともインストールしておいてください。参考までに、二つのプログラムの役割や、今回の実行環境などを表にまとめておきます。この表のとおり、今回は、Viewer は Windows PC で、ABINIT-MP は Linux の並列計算機で行います。そのため、両方の計算機の間でファイル転送できるようにしてください。たとえば、WinSCP や FFTP などの無料ソフトが利用できます。

プログラム名	バージョン	役割	実行環境	計算機
Biostation Viewer	Open 1.0 rev. 25 Binds 003	FMO 計算のプリポスト（入力ファイル作成と出力ファイルの解析）	Windows (Windows 10 Pro)	Panasonic Let's note CF-LX6 / CF-SZ26
ABINIT-MP	Open Ver. 1 Rev. 15	FMO 計算の実行	Linux/Unix (CentOS Linux 7.4., Intel compiler ver 18, Intel MPI Ver 2018)	HPC5000-XCl216-TS-D8 (Intel Xeon Gold 6252 x2, 24 cores)

計算環境が整ったら、実習用ファイル類 FMO_tutorial_files_ver1.zip を解凍します。以下のようなディレクトリー（フォルダー）構造です。ディレクトリーは名前の末尾に/をつけてあります。また、ディレクトリー名の 01_, 02_、等は次章以降の課題番号に対応します。work/ディレクトリー下の TrpCage.pdb, ER_50res.pdb などは、それぞれの課題で入力する分子構造ファイル（PDB 形式）で、run.sh は、FMO 計算実行用のシェスクリプトです。reference/ディレクトリーには、これから作成するファイルがすでに作ってありますので（ファイル名は以下では省略）、適宜、自分の結果と比較してください。

本資料内の実習用ファイルは、このディレクトリーを用いて示します。例えば、課題 1 のジョブ実行用ファイル run.sh は次のように表現します。

```
01_TrpCaga/work/run.sh
```

```
FMO_tutorial_files_ver1/
    01_TrpCage/
        reference/
        work/
                run.sh
                TrpCage.pdb
    02_ER/
        reference/
        work/
                run.sh
                ER_50res.pdb
    03_Formamide/
        reference/
        work/
                run.sh
                Formamide.pdb
    04_CAFI/
        reference/
        work/
                run.sh
                w3.pdb
    05_FILM/
        reference/
        work/
                run.sh
                met-ben.pdb
    Utility/
                mkinp_openver1rev15.py
```

ABINIT-MP プログラムは何種類ものファイルを入出力しますが、それらはファイル名の拡張子で区別します。この資料では、拡張子を大文字にしてファイルの種類（型）を表します。たとえば、入力パラメーターファイルの拡張子は".ajf"なので、"AJF ファイル"と呼びます。 以下は、この資料に登場するファイルの種類のリストです。

拡張子	ファイルの種類	内容
.ajf	AJF	ABINIT-MP を実行するための入力パラメータ。

.pdb	PDB	Protein Data Bank (PDB) 形式の分子構造の座標データ。
.log	LOG	ABINIT-MP の計算結果を人間が読みやすい形式で出力したもの。エエネルギーや原子の電荷などのデータが書かれている。
.cpf	CPF	Check Point Restart (CPF) 用の出力データ。計算機用の形式で書かれている。IFIE や PIEDA にも用いる。
.pno	PNO	Pair Natural Orbital (PNO)のグリッド形式のデータ。CAFI に用いる。
.lmp2	LMP2	Local MP2 (LMP2)のグリッド形式のデータ。FILM に用いる。
.prof	PROF	Viewer の状態(profile)をそのまま保存するファイル。あとで、Viewer 作業を再開したいときに用いる。
.pref	PREF	PROF と一緒に入出力される、Viewer の設定情報（preference）が記述されているテキストファイル。
.err	ERR	ABINIT-MP の実行時に、ABINIT-MP プルグラムではなく、OS からのメッセージが出力されるファイル。ジョブが正常に終了した場合は、ただの空ファイルになる。

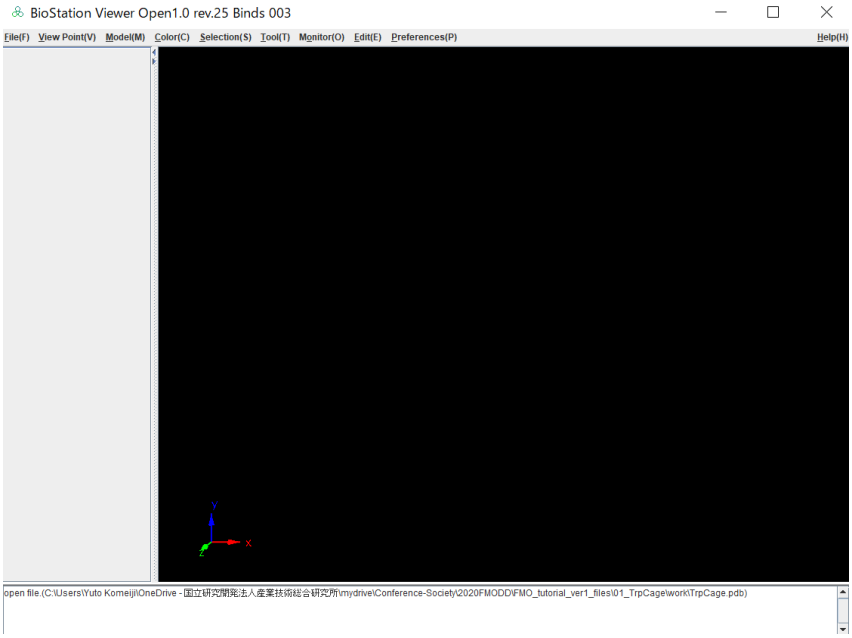
1　課題 1：TrpCage の IFIE 解析と CHPI 解析

本課題ではミニタンパク質の TrpCage の FMO-MP2 と FMO-HF 計算の入力データの作成、計算の実行を行い、フラグメント間相互作用エネルギー（IFIE）及び CH/π（CHPI）相互作用解析を行います。

1.1　入力ファイル作成 (AJF)

デスクトップ上のアイコンをクリックして、Viewer を起動します。

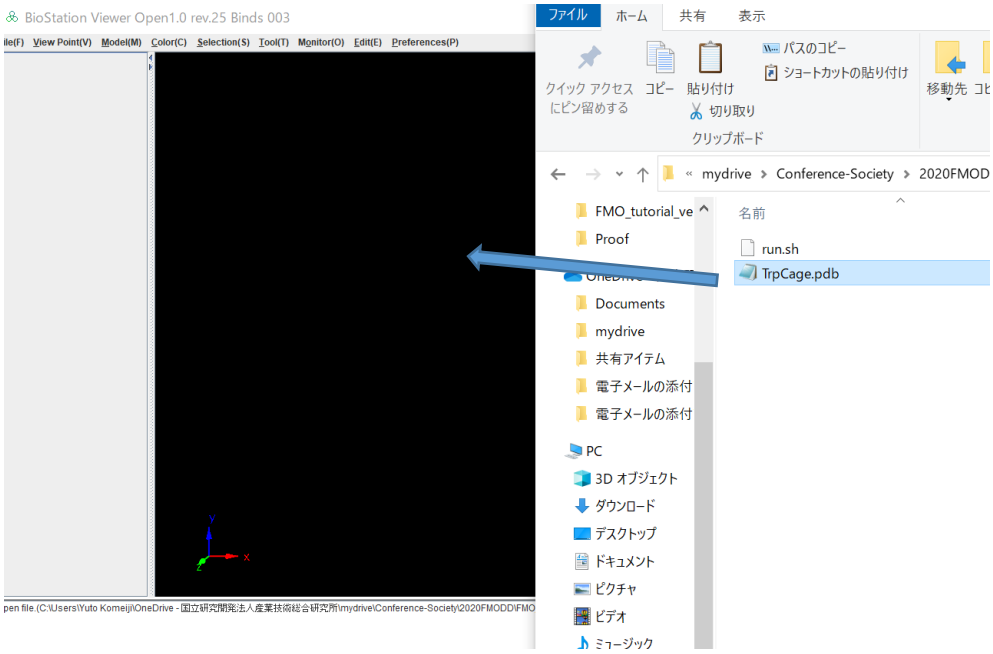
Viwer が起動します。



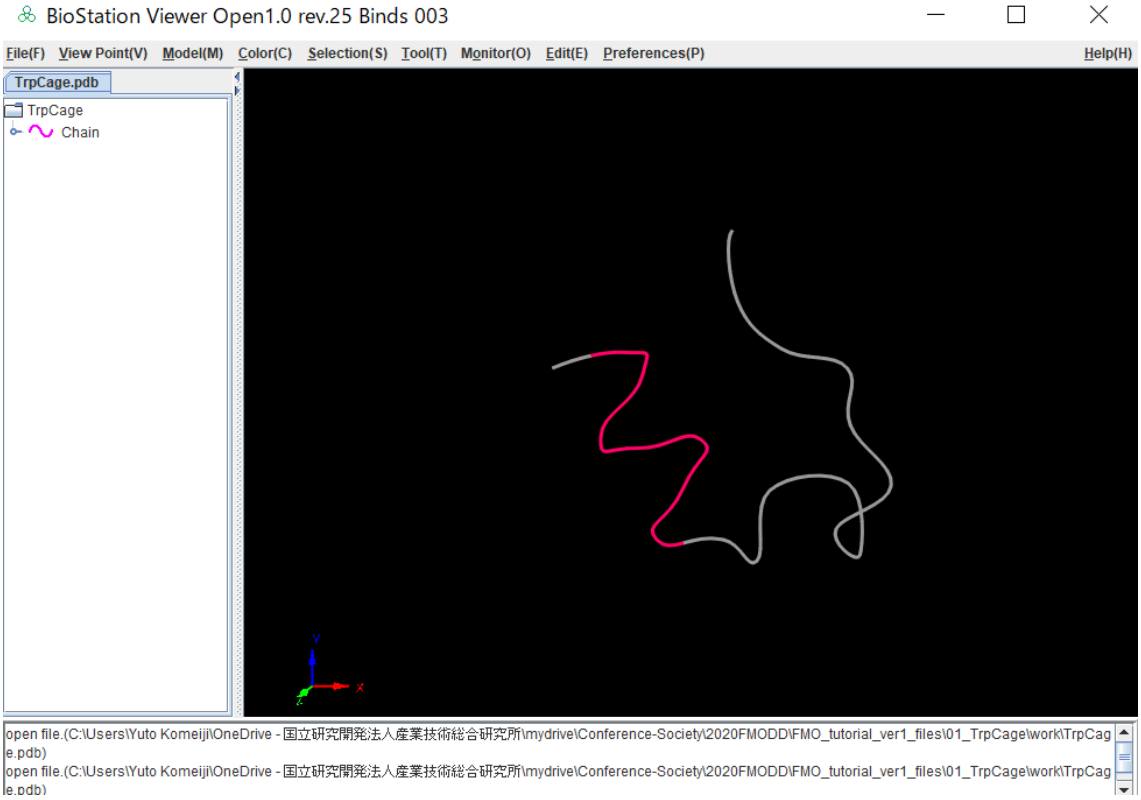
Viewer で、TrpCage.pdb の構造を読み込みます。0.序で述べたように、TrpCag.pdb は以下のファイルです。

```
01_TrpCage/work/TrpCage.pdb
```

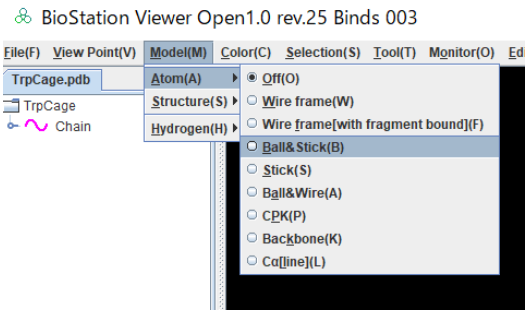
この分子構造ファイル（PDB ファイル）を読み込むには、”File>Open File”を実行するか、または、Viewer の画面に PDB ファイルをドラッグ＆ドロップします。これ以降はドラッグ＆ドロップの方法でのみ説明します。

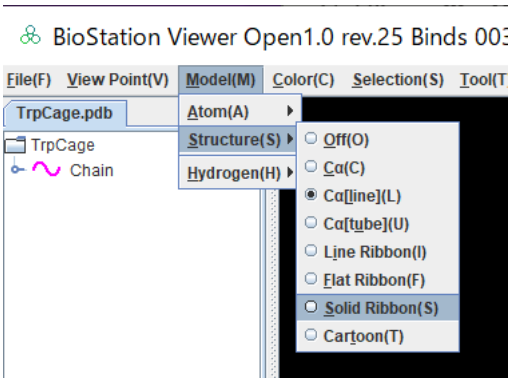


TrpCage の構造が表示されます。デフォルトの表示ではタンパク質の骨格のみが表示されます（主鎖の Cα 原子をラインでつないだ表示)。

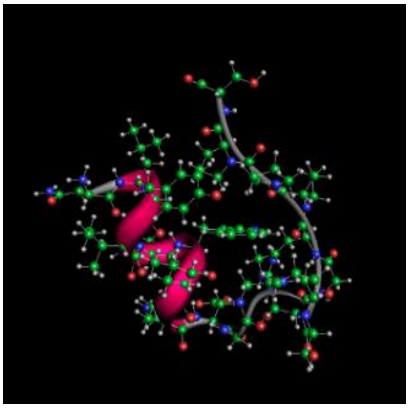


Model>Atom>Ball&Stick と Model>Structure>Solid Ribbon を実行して表示方法を変更しましょう。

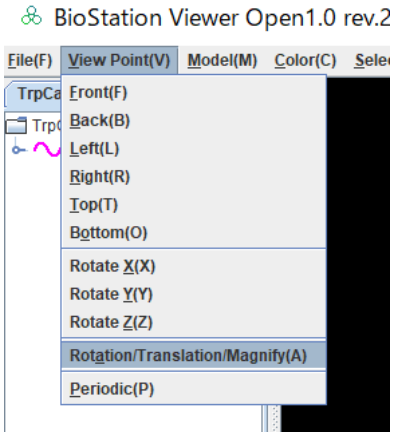


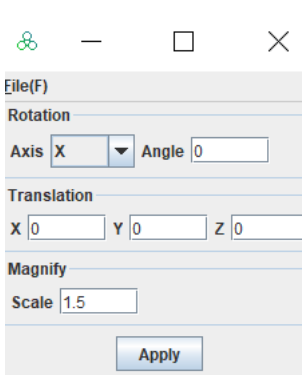


すると、TrpCage タンパク質の分子構造が以下のように表示されます。

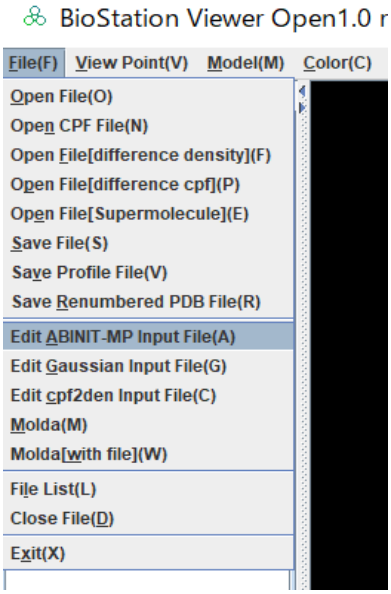


図を拡大または縮小したい場合があります。そのときは、View Point > Rotation/Translation/Magnify 選択し、ポップアップウィンドウで Scale を変更して Apply をクリックします。 この例では、Scale を 1.5 にリセットしたので、それに応じて図が 1.5 倍に拡大されています。

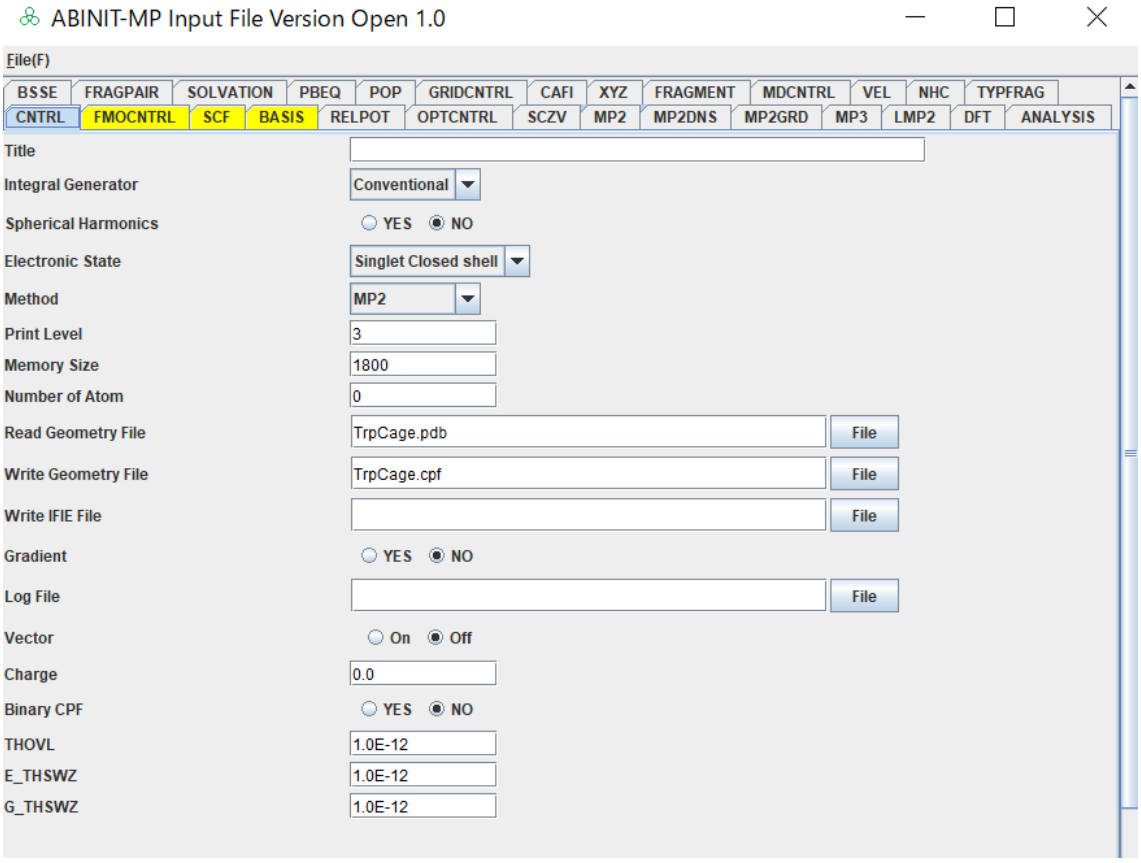




FMO 計算の入力ファイル（AJF ファイル）を作成するには、File>Edit ABINIT-MP Input File を実行します。



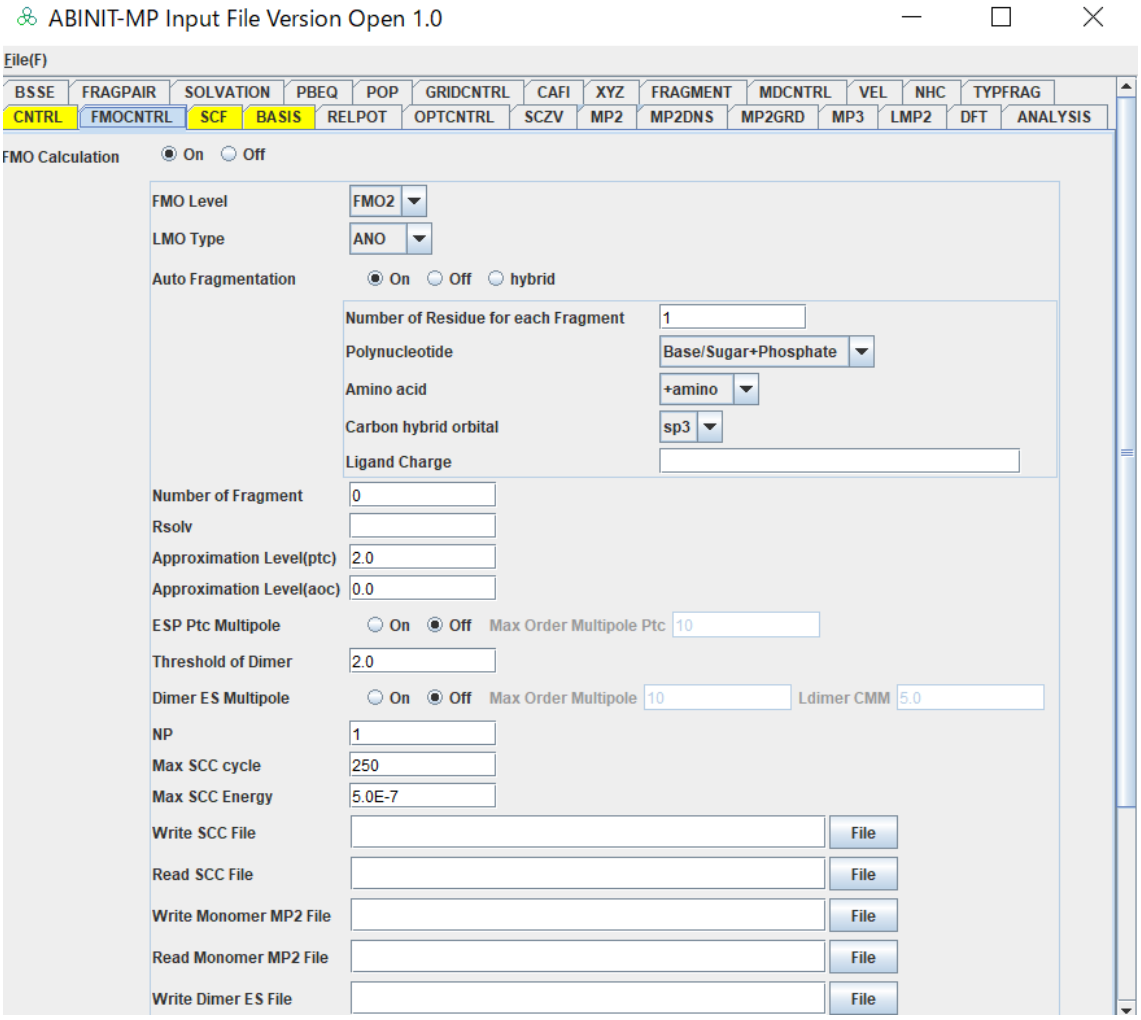
以下のウィンドウが起動します。CNTRL、FMOCNTRL、SCF など複数のタブがあり、それらの中で各種のパラメータが指定できます。



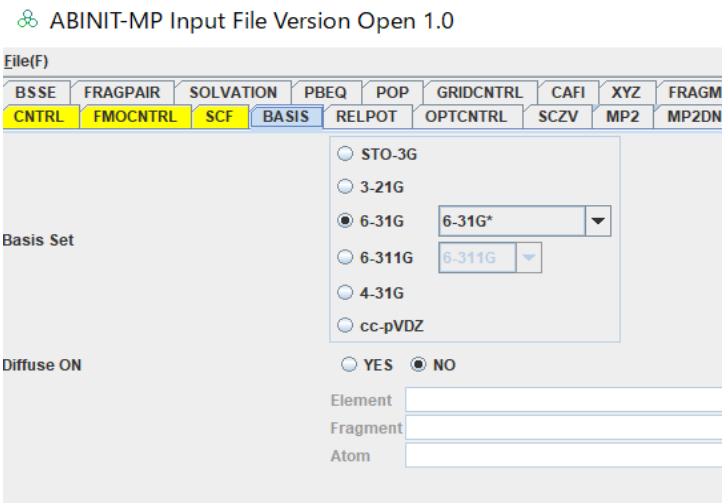
CNTRL タブの Method で計算レベルを制御できます。MP2 になっていることを確認してください。次に、Read Geometry File を TrpCage.pdb とします。Write Geometry File には TrpCage-FMO-MP2.cpf

と入力します。

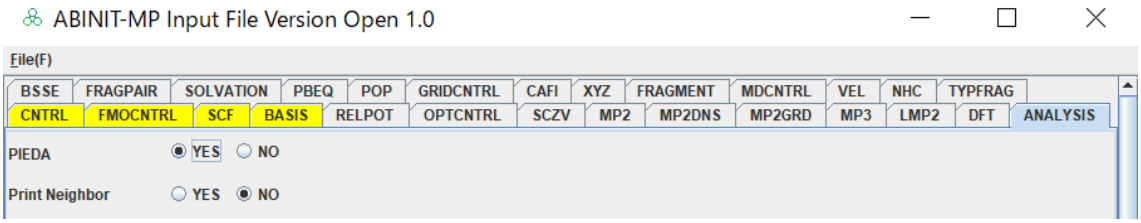
次に、FMOCNTRL タブにて FMO 関連のパラメータを設定します。ただし、本課題では特に変更はありません。



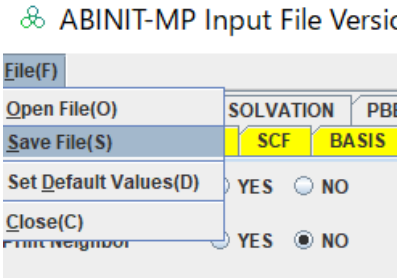
BASIS タブにて基底関数関連のパラメータを指定します。6-31G*になっていることを確認してください（通常は 6-31G*が推奨されます）。



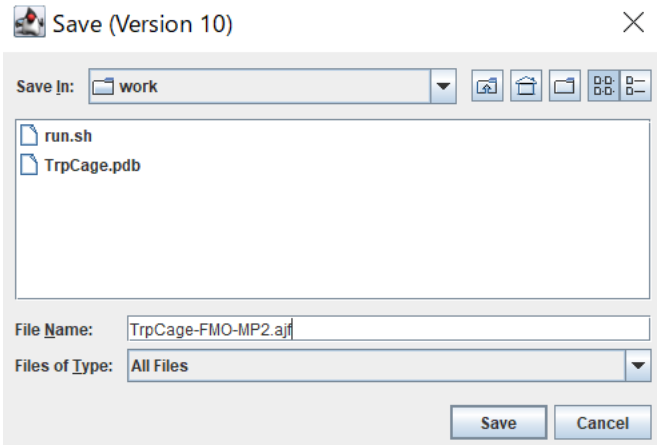
また、Analysis タブにて PIEDA を“YES”とすることで PIEDA 計算も実行できます。



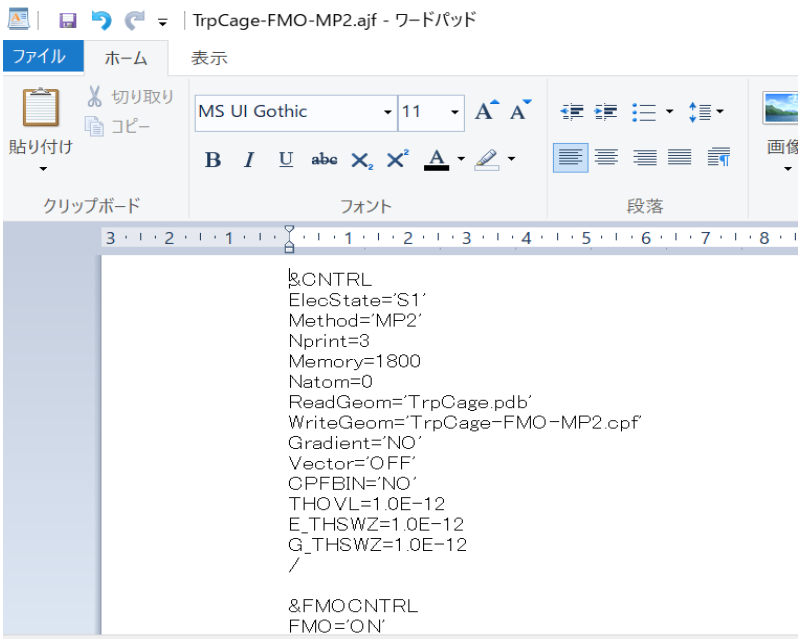
パラメータの設定が完了したら File>Save File を実行して入力ファイルを保存します。



ここでは TrpCage-FMO-MP2.ajf という名前で保存します。



PDB ファイルが置いてあるディレクトリーに TrpCage-FMO-MP2.ajf が出力されますので、テキストエディター（メモ帳、ワードパッド等）で開いて中身を確認しましょう。



手動で"LRD"および"/"行を AJF ファイルに追加します（理由は、この節の最後の「%注意」を参照してください)。.

```
&LMP2
/
&LRD
/
&DFT
/
```

さらに、HF / STO-3G 計算用に別の AJF ファイルを作成します。というのは、MP2 / 6-31G *の実行には 1 時間程度掛かるため、もっと短時間で終わる低レベルの試験計算も必要だからです。HF 用の AJF は、Viewer を実行する代わりに、TrpCage-FMO-MP2.ajf を TrpCage-FMO-HF.ajf にコピーし、テキストエディターで後者を変更します。

```
&CNTRL
    ...
    Method="HF"
    WriteGeom="TrpCage-FMO-HF.cpf"
    ...
&BASIS
    ...
    BasisSet="STO-3G"
```

%注意: BioStationViewer と ABINIT-MP の不整合への対処

AJF に"LRD"と"/"を手動で追加してもらいました。その理由は、Viewer で作成された AJF ファイルと ABINIT-MP の間には、通常、不整合があるからです。具体的には、Viewer で作った AJF ファイルに、ABINIT-MP に最近追加された新しい namelist がない、という事態がしばしば起るからです。今回、もし、"LRD"と"/"を AJF に追加せずに run.sh を実行すると、TrpCage-FMO-HF.log に以下のようなエラーメッセージが書き込まれて、異常終了してしまいます。

```
    ## READ NAMELIST "MP3" : IOSTAT =    0

    ## READ NAMELIST "LMP2" : IOSTAT =    0

    ## READ NAMELIST "LRD" : IOSTAT =   -1

    FAIL TO READ NAMELIST "LRD" , IOSTAT =   -1
INPUT ERROR!
```

このメッセージは、namelist"LRD"が AJF ファイルにないことを意味しています。 ABINIT-MP は、新しいアルゴリズムと機能を取り入れつつ急速に更新されているため、Viewer が追い付かず、結果として、こんなふうに、ABINIT-MP と Viewer の間に齟齬が生まれる、ことが多いのです。

この問題には、二つ回避策があります。 一つは、すでに行ったように、"LRD"namelist を手動で追加することです。 もう一つは、"mkinp_openver1rev15.py"スクリプトの使用です。これは、AJF ファイルを ABINIT-MP Open 1 rev に適したものに再フォーマットする Python スクリプトでです。

```
Utility/mkinp_openver1rev15.py
```

このスクリプトを Linux マシンに転送して、次のように実行します。

```
mkinp_openver1rev15.py <TrpCage-FMO-HF.ajf> TrpCage-FMO-HF.reform.ajf
```

すると"TrpCage-FMO-HF.reform.ajf"という再フォーマットされた出力ファイルが作成されて、これは、ABINIT-MP Open 1 rev. 15 でそのまま入力ファイルとして利用できます。お試しください。

1.2 FMO 計算の実行

いよいよ、Linux の計算サーバに入力データ（PDB ファイルと AJF ファイル）を WinSPC や FFTP などのソフトを使ってアップロードし、FMO 計算を実行します。ターミナル画面を開き、アップロードした入力データがあるディレクトリーに移動します。

```
[komeiji@pc074083 work]$ pwd
/home/komeiji/FMO_tutorial_work/01_TrpCage/work
[komeiji@pc074083 work]$ ls
run.sh*  TrpCage-FMO-HF.ajf  TrpCage-FMO-MP2.ajf  TrpCage.pdb
[komeiji@pc074083 work]$
```

vi などで AJF ファイルを開き、パラメータをもう一度確認します。

```
vi TrpCage-FMO-HF.ajf (or TrpCage-FMO-MP2.ajf)
```

以下、計算機ごとに変更する必要が多いパラメータを示します。これらは計算を流す前によく確認し、必要に応じて変更します。

```
&CNTRL
…
Memory=1800        1プロセスあたりの使用メモリ上限
…
ReadGeom='TrpCage.pdb'        PDBファイルへのパス
（計算機によっては絶対パスで指定しないとエラーになるかも知れません）
WriteGeom='TrpCage-FMO-HF.cpf'
…
/

&FMOCNTRL
…
NP=4  ← 1フラグメントを計算するのに使用するプロセス数
４を使うことが多いが、ジョブに合わせて最適な数を選ぶ必要があります
…
```

ABINIT-MP プログラムは、シェルスクリプト run.sh で実行します。run.sh のシンプルな例を下記に上げておきます。まず、変数"ABINIT"に代入している ABINIT-MP の実行ファイルを、ご自分の計算環境に合わせて書き直してください。計算機によってはもっと複雑に書き直す必要があるかも知れません。特に、計算機センターの超並列計算機などは、面倒かも。幸運を祈ります。

```
#!/bin/sh

NPALL=24      # total number of processors
ABINIT=/home/komeiji/ABINIT-MP_rev15/AIST-KMJ/abinitmp
              # the executable program of ABINIT-MP
# HF
JOB=TrpCage-FMO-HF                # define & describe JOB
rm -f $JOB.log $JOB.err $JOB.cpf  # remove old output files
mpirun -np $NPALL $ABINIT< $JOB.ajf 1> $JOB.log 2>$JOB.err
                                  # execute FMO with ABINIT-MP
# MP2
# NB: comment out the following lines if you can't wait for long
JOB=TrpCage-FMO-MP2               # define & describe JOB
rm -f $JOB.log $JOB.err $JOB.cpf  # remove old output files
```

```
mpirun -np $NPALL $ABINIT< $JOB.ajf 1> $JOB.log 2>$JOB.err
                                    # execute FMO with ABINIT-MP
```

MP2 のジョブは、終了まで 1 時間近くかかります。それで、もしこれをスキップしたければ、MP2 ジョブは以下のようにコメントアウトしてください。

```
# MP2
# NB: comment out the following lines if you can't wait for long
# JOB=TrpCage-FMO-MP2             # define & describe JOB
# rm -f $JOB.log $JOB.err $JOB.cpf      # remove old output files
# mpirun -np $NPALL $ABINIT< $JOB.ajf 1> $JOB.log 2>$JOB.err
                                    # execute FMO with ABINIT-MP
```

コメントアウトした run.sh は、HF/STO-3G を実行するだけですので、通常数分で終了します。

さて、run.sh を以下のように実行してください。

run.sh & （または./run.sh &）

ps コマンドか、あるいはもし qsub で実行した場合は qstat コマンドで、"abinitmp"という名のジョブが走っていることを確認してください。NPALL で指定した数だけの abinitmp が走っているはずです。

```
[komeiji@pc074083 01_TrpCage]$ run.sh &
[1] 39133
[komeiji@pc074083 01_TrpCage]$ ps
  PID TTY          TIME CMD
34617 pts/1    00:00:00 bash
39133 pts/1    00:00:00 run.sh
39135 pts/1    00:00:00 mpirun
39140 pts/1    00:00:00 mpiexec.hydra
39141 pts/1    00:00:00 pmi_proxy
39145 pts/1    00:00:03 abinitmp
39146 pts/1    00:00:03 abinitmp
39147 pts/1    00:00:03 abinitmp
39148 pts/1    00:00:03 abinitmp
39149 pts/1    00:00:03 abinitmp
39150 pts/1    00:00:03 abinitmp
```

計算が完了すると TrpCage-FMO-HF.log（人間が読める形式のデータ）と TrpCage-FMO-HF.cpf（計算機が読むためのデータ）が出力されます。TrpCage-FMO-HF.log のファイル末尾に以下が表示されて入れば、計算は問題なく終了しています。

```
-----------------
```

```
  ABINIT-MP DONE
-----------------
  2020/03/03  18:41:36
```

もし reference/ディレクトリーも Linux 上にコピーしておいた場合は、次のコマンドで、比較してみてください。

```
diff TrpCage-FMO-HF.log ../reference | more

diff TrpCage-FMO-MP2.log ../reference | more
```

計算環境により、エネルギーの値に微妙な違いが出ますが、この程度の違いなら、許容範囲でしょう。もちろん、計算時間は、計算機によってかなり変わります。

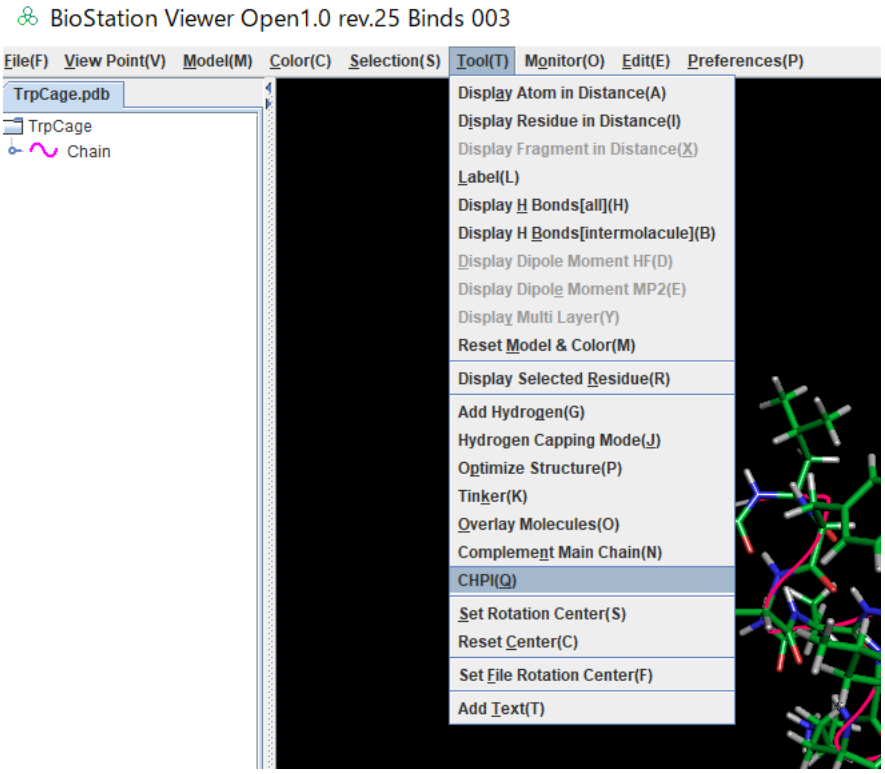
```
<         Electronic energy =        -54623.0450313802
<         Total      energy =         -7344.8962247025
<
---
>         Electronic energy =        -54623.0450313776
>         Total      energy =         -7344.8962246999
>
2589,2591c2622,2624
<         Elapsed time: Monomer SCF     =           18.9 seconds
<         Elapsed time: Dimer ES        =            1.1 seconds
<         Elapsed time: Dimer SCF       =           21.2 seconds
---
>         Elapsed time: Monomer SCF     =           21.2 seconds
>         Elapsed time: Dimer ES        =            1.9 seconds
>         Elapsed time: Dimer SCF       =           32.2 seconds
2594c2627
<         Elapsed time: FMO (Total)     =           40.1 seconds
---
>         Elapsed time: FMO (Total)     =           53.4 seconds
```

MP2 ジョブまで行った人は、TrpCage-FMO-MP2.log と TrpCage-FMO-mp2.cpf ファイルを、もとの、Viewer が載っている、Windows PC に送ってください。FMO 出力の解析作業に用います。

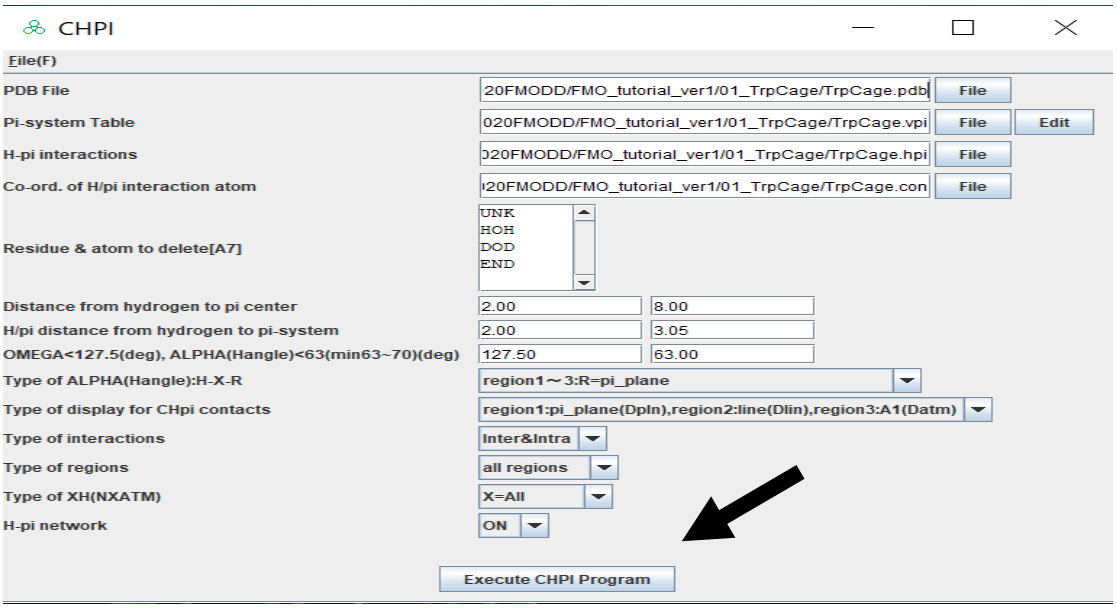
1.3 CHPI 解析

FMO の出力データ解析を行う前に、CHPI 解析を行います。CHPI プログラムは西尾先生と梅沢先生が開発されたプログラムで、PDB ファイルから分子構造を入力し、その構造だけを見て、CH/π 相互作用を探し出します。

まず、Viewer で TrpCage.pdf を開き、Model > Atom > Sticks” で、分子構造を表示させます。さらに、Tool>CHPI を実行してください。なお、CHPI 解析で何か失敗したときは、この節の終わりの「%注意」を読んでください。



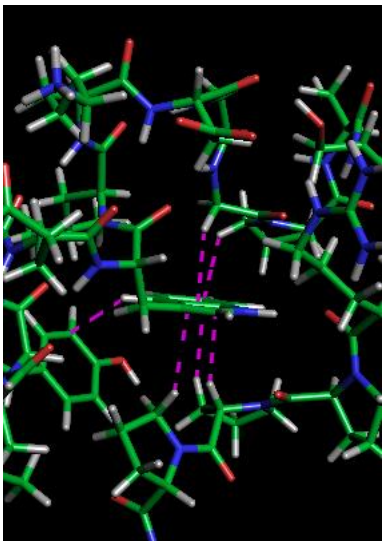
CHPI 解析のパラメータ設定を行うウィンドウが起動します。通常は変更する必要がないためここではデフォルト設定のまま Execute CHPI Program ボタンをクリックします。各パラメータの詳細については書籍「有機化学のための分子間力入門、西尾元宏、講談社サイエンティフィック」をご参照ください。



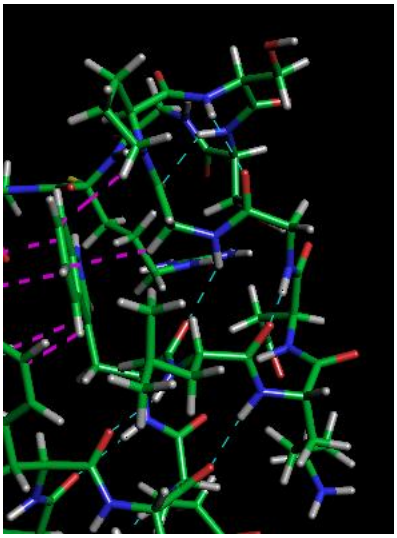
DOS プロンプトが起動し、CHPI プログラムの出力が表示されます。Please input exit と表示されれば完了です。exit コマンドを実行するか、ウィンドウの×ボタンをクリックして DOS プロンプトを閉じます。

```
C:¥WINDOWS¥system32¥cmd.exe
range      2.00  < DHPI  <    3.05
range   -127.50  < OMEGA <  127.50
range      0.00  < Hangl <   63.00
Type of H_angle(H-X-R)?
  [R=pi_plane,line(C1-C2),C1(closest pi_atom)]
 0:R=pi_plane(all regions)
 1:R=pi_plane(region-1),R=line(region-2),R=C1(region-3)
 2:R=C1(all regions)
:0
Type of display?  0:(pi_plane,line,C1)  1:C1, 2:pi_center
:0
Type of interactions?  0:intra&inter  1:inter  2:intra
:0
Type of regions?
 0:all regions  1:region-1  2:region-2  3:region-3
:0
Type of XH?  0:ALL  1:C, 2:N, 3:O, 4:S, 5:except C
:0
H-pi network?   0:off  1:on
:1

**Inter&Intramolecular H/pi interactions**


Num of H/pi interactions :   6 (region_1:  3, region_2:  2, region_3:  1)


Please type 'exit', and press 'Enter.'

C:¥Program Files¥ABINIT-MP Open Consortium¥BioStationViewer>_
```

以下のように CH/π相互作用がある箇所に紫の破線が表示されます。



さらに、水素結合を表示してみます。Tool → Display H Bonds [all] を選択すると、水素結合の位置に水色の破線が表示されます。



以上の CHPI 表示や水素結合表示は、PDB 立体構造から相互作用の候補位置を検索するツールです。実際に相互作用があるかどうかは、IFIE/PIEDA 解析（1.4 節）をすることでエネルギーを定量的に計算することで、解析できます。

%注意：CHPI プログラムで扱う PDB ファイル

PDB ファイルの書式が整っていない場合には CHPI プログラムはエラーとなります。その場合には、File > Save Renumbered PDB File を選択し、書式の整った PDB ファイルを出力し、その新しく作成した PDB ファイルを用いて再度実行してください。

1.4 IFIE･PIEDA 解析

本節では、FMO 計算の出力に対して、まず IFIE、次いで PIEDA 解析を行います。

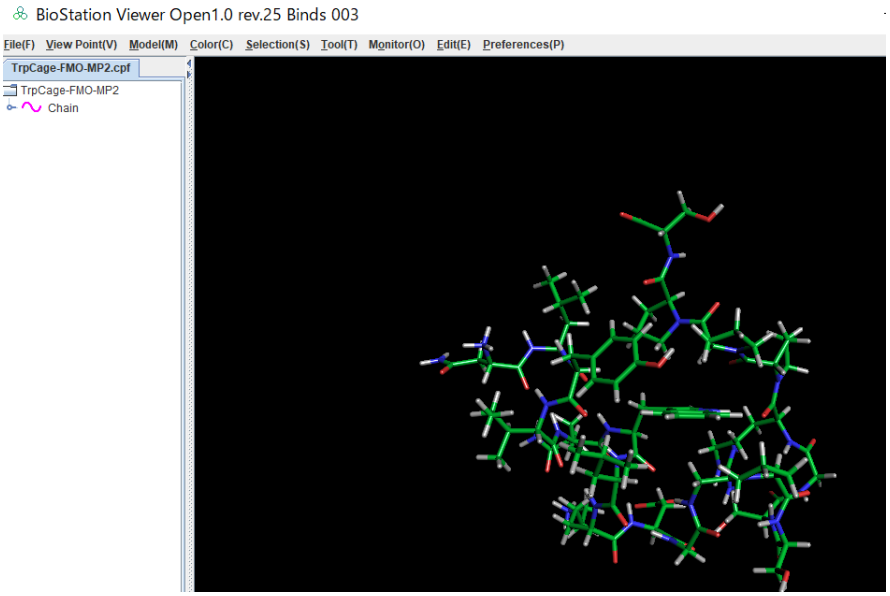
CPF ファイルを BioStation Viewer の画面にドラッグ&ドロップして読み込みます。FMO 計算は HF/STO-3G と MP2/6-31G*の両方で行いましたが、ここでは MP2/6-31G*計算のデータのみを解析します。1.2 節で、TrpCage-FMO-MP2 ジョブを実行した場合は、そこで作成された TrpCage-FMO-MP2.cpf ファイルを解析してください。

```
01_TrpCage/work/TrpCage-FMO-MP2.cpf
```

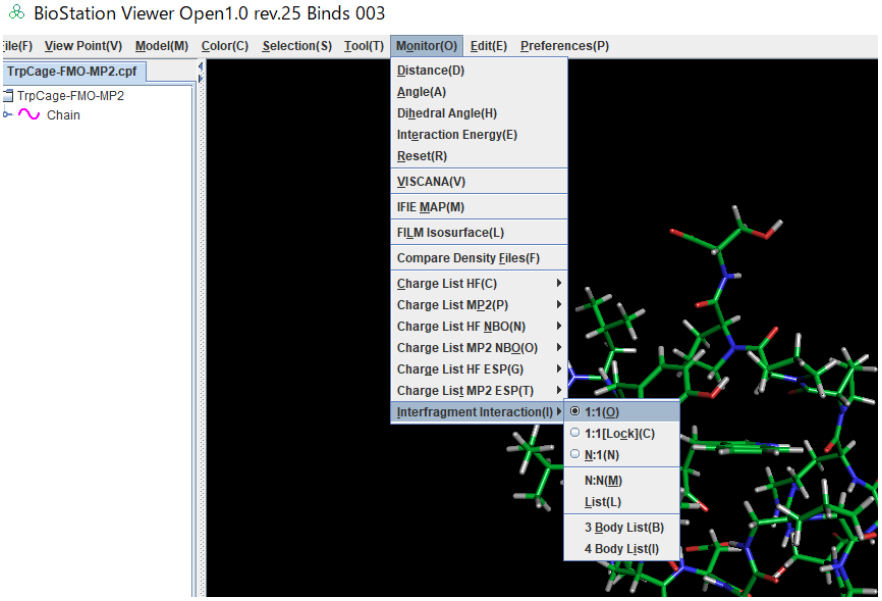
ただし、MP2 計算を実行していない場合は、以下の reference/ディレクトリー内の TrpCage-FMO-MP2.cpf ファイルを使ってください。。

```
01_TrpCage/reference/TrpCage-FMO-MP2.cpf
```

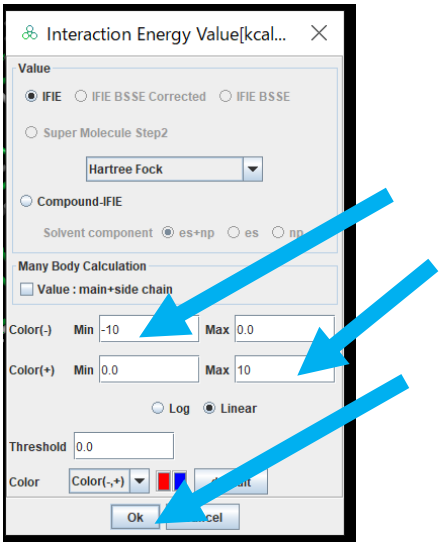
Viewer を再起動し、cpf ファイルをその上にドラッグし、「Model> Atom> Stick、Model> Structure（s）> Off」を実行して分子構造を視覚化します。



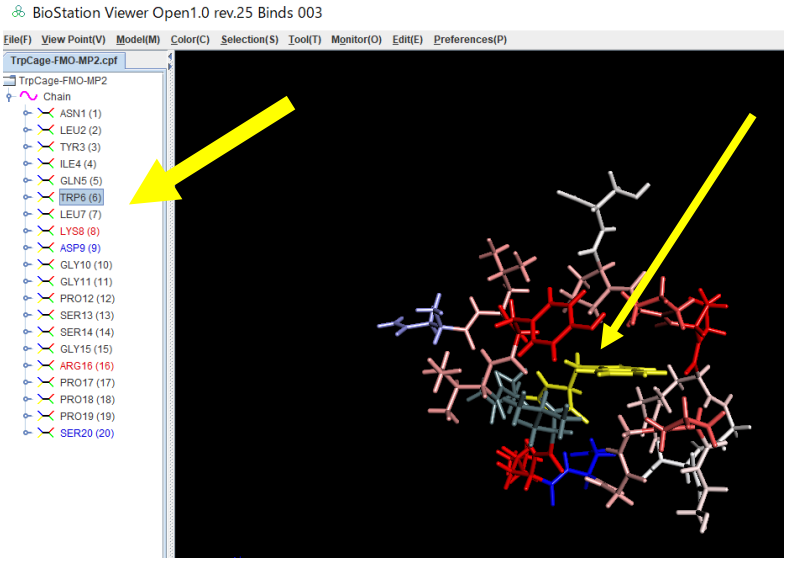
Monitor > Interfragment Interaction > 1:1 を実行します。ここで IFIE 解析には 1:1、N:1、N:N などの選択肢がありますが、1:1 は 1 フラグメント 1 対 1 フラグメント 1 の IFIE を解析します。注目する残基（リガンドなど）が複数のフラグメントに分割されている場合は N:1 などを使うことがあります。また、タンパク質対 DNA の IFIE を見る場合などに N:N を使うことがあります。



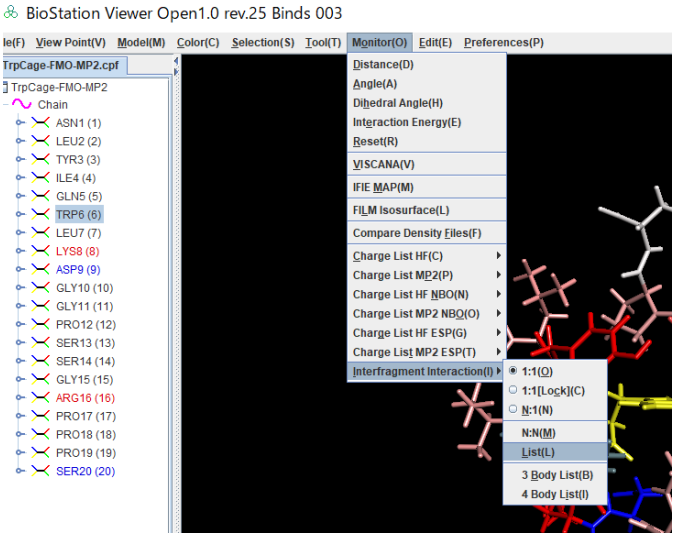
IFIE 解析のパラメータ設定ウィンドウが起動します。本課題では FMO-MP2 計算を行ったので Value のプルダウンは MP2 を選択します。Color のテキストボックスでは IFIE を青赤のグラデーションで表示する際の上限値と下限値を指定します。ここでは-10 から 10 とします。



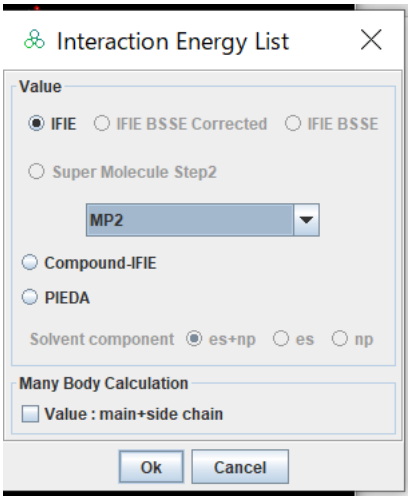
OKボタンをクリックし、IFIE解析したいフラグメントをクリックすると、そのフラグメントとのIFIEに基づいて周囲のフラグメントが色付けされます。下の例では、Trp6を選んでいます。黄色は現在注目しているフラグメント、赤は注目しているフラグメントと引力的な相互作用をしているフラグメント、青は注目しているフラグメントと斥力的な相互作用をしているフラグメント、灰色はそのフラグメントと共有結合しているためIFIEを評価できないフラグメントを意味します。



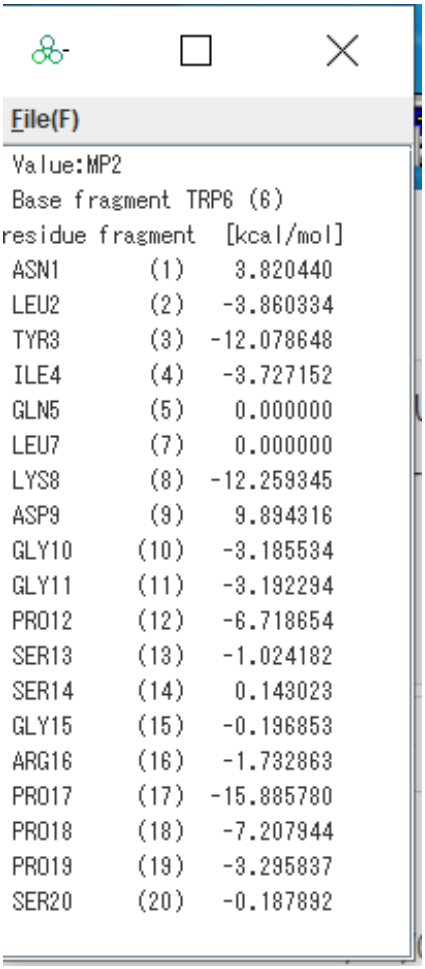
IFIEの数値を表示する場合にはMonitor＞Interfragment Interaction＞Listを実行します。



MP2 を選び、Ok をクリックしてください。



注目しているフラグメントとその周囲のフラグメント間の IFIE が kcal/mol 単位で表示されます。

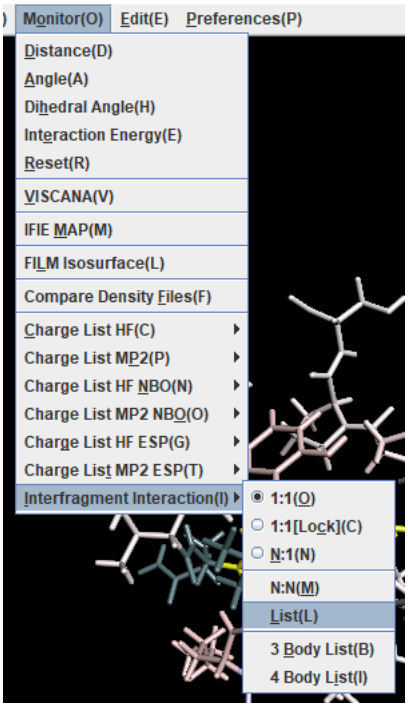


File(F)

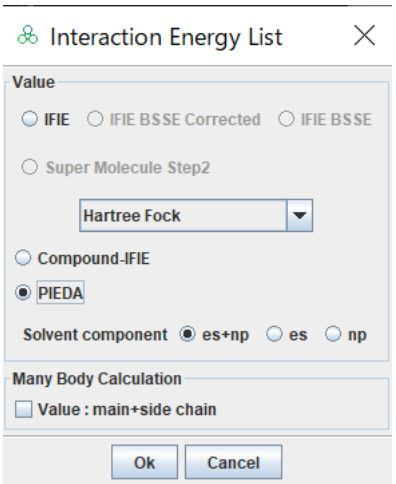
Value:MP2
Base fragment TRP6 (6)

residue	fragment	[kcal/mol]
ASN1	(1)	3.820440
LEU2	(2)	-3.860334
TYR3	(3)	-12.078648
ILE4	(4)	-3.727152
GLN5	(5)	0.000000
LEU7	(7)	0.000000
LYS8	(8)	-12.259345
ASP9	(9)	9.894316
GLY10	(10)	-3.185534
GLY11	(11)	-3.192294
PRO12	(12)	-6.718654
SER13	(13)	-1.024182
SER14	(14)	0.143023
GLY15	(15)	-0.196853
ARG16	(16)	-1.732863
PRO17	(17)	-15.885780
PRO18	(18)	-7.207944
PRO19	(19)	-3.295837
SER20	(20)	-0.187892

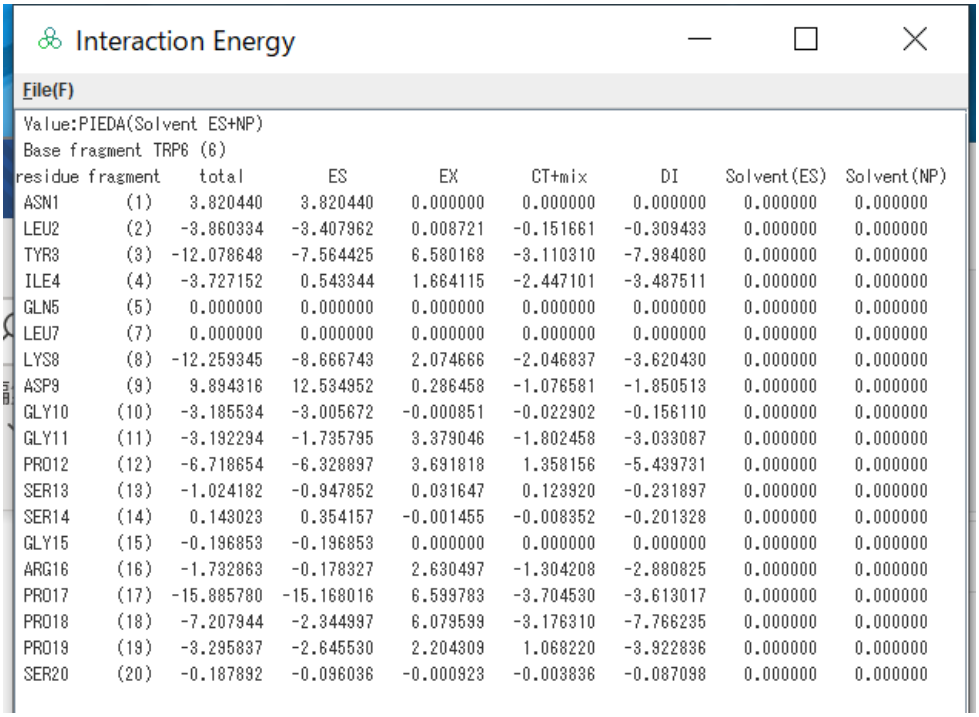
IFIE の中身をさらにエネルギーの種類ごとに分割したい、つまり PIEDA を行いたいときは、Monitor > Interfragment Interaction > List を実行してください。



以下のウィンドーが現れますので、PIEDA を選び、Ok を押してください。



PIEDA のリストが現れます。



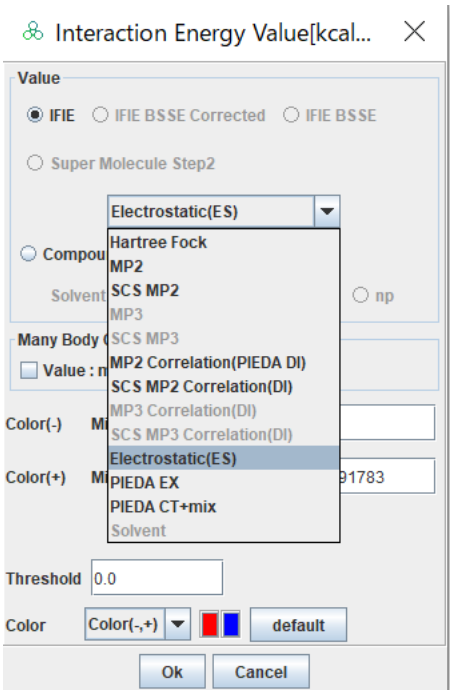
Interaction Energy

File(F)

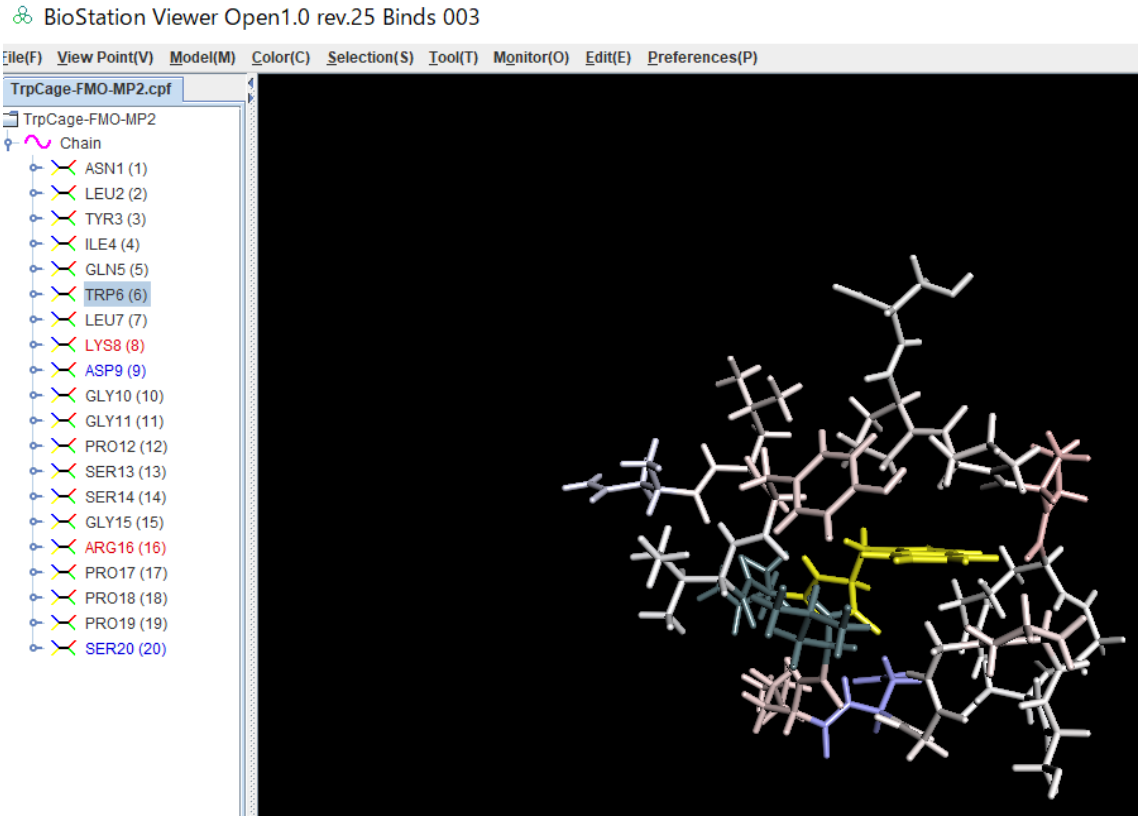
Value:PIEDA(Solvent ES+NP)
Base fragment TRP6 (6)

residue	fragment	total	ES	EX	CT+mix	DI	Solvent(ES)	Solvent(NP)
ASN1	(1)	3.820440	3.820440	0.000000	0.000000	0.000000	0.000000	0.000000
LEU2	(2)	-3.860334	-3.407962	0.008721	-0.151661	-0.309433	0.000000	0.000000
TYR3	(3)	-12.078648	-7.564425	6.580168	-3.110310	-7.984080	0.000000	0.000000
ILE4	(4)	-3.727152	0.543344	1.664115	-2.447101	-3.487511	0.000000	0.000000
GLN5	(5)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
LEU7	(7)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
LYS8	(8)	-12.259345	-8.666743	2.074666	-2.046837	-3.620430	0.000000	0.000000
ASP9	(9)	9.894316	12.534952	0.286458	-1.076581	-1.850513	0.000000	0.000000
GLY10	(10)	-3.185534	-3.005672	-0.000851	-0.022902	-0.156110	0.000000	0.000000
GLY11	(11)	-3.192294	-1.735795	3.379046	-1.802458	-3.033087	0.000000	0.000000
PRO12	(12)	-6.718654	-6.328897	3.691818	1.358156	-5.439731	0.000000	0.000000
SER13	(13)	-1.024182	-0.947852	0.031647	0.123920	-0.231897	0.000000	0.000000
SER14	(14)	0.143023	0.354157	-0.001455	-0.008352	-0.201328	0.000000	0.000000
GLY15	(15)	-0.196853	-0.196853	0.000000	0.000000	0.000000	0.000000	0.000000
ARG16	(16)	-1.732863	-0.178327	2.630497	-1.304208	-2.880825	0.000000	0.000000
PRO17	(17)	-15.885780	-15.168016	6.599783	-3.704530	-3.613017	0.000000	0.000000
PRO18	(18)	-7.207944	-2.344997	6.079599	-3.176310	-7.766235	0.000000	0.000000
PRO19	(19)	-3.295837	-2.645530	2.204309	1.068220	-3.922836	0.000000	0.000000
SER20	(20)	-0.187892	-0.096036	-0.000923	-0.003836	-0.087098	0.000000	0.000000

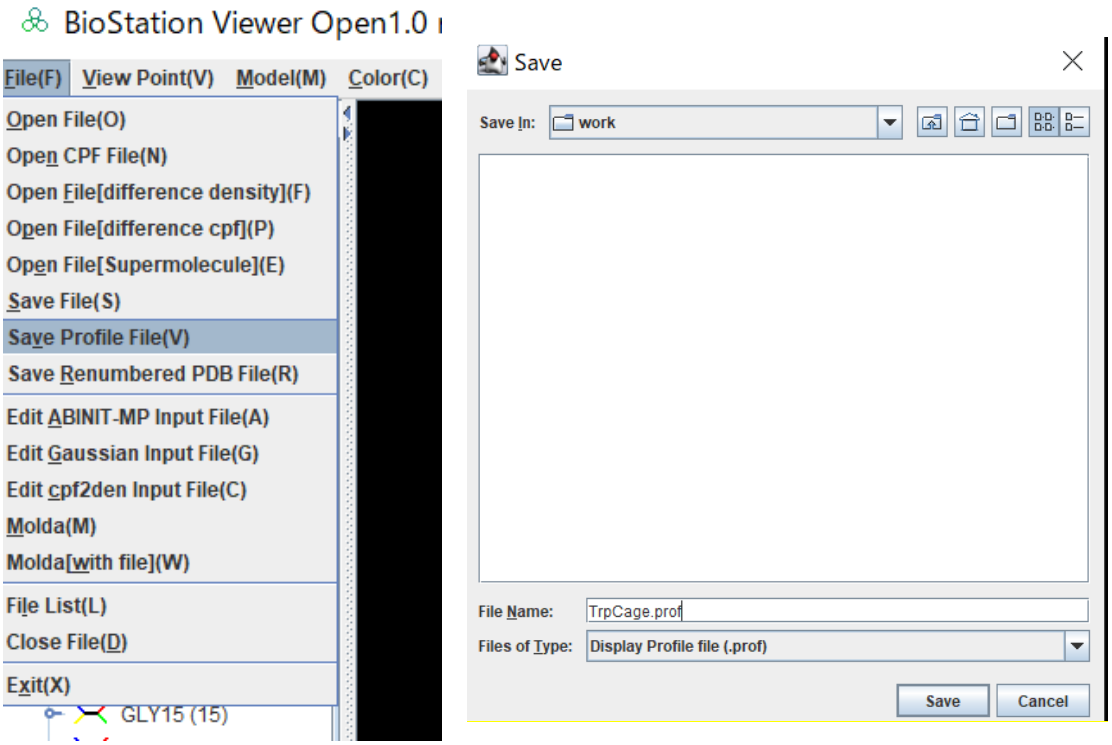
PIEDA の各成分をカラー表示して分子構造に投影したければ、Monitor> Interfragment> 1：1 を再度実行して、次のウィンドウを表示します



ここで、"Electrostatic (ES)"、"PIEDA EX"、"PIEDA CT + mix"、 "MP2 Correlation (DI)"の中から欲しい成分を選択して、Ok をクリックします。この例では Electrostatic を選択しました。すると、Trp 6 と静電相互作用、つまり、Electrostatic で強く相互作用しているフラグメントが色付きで表示されます。



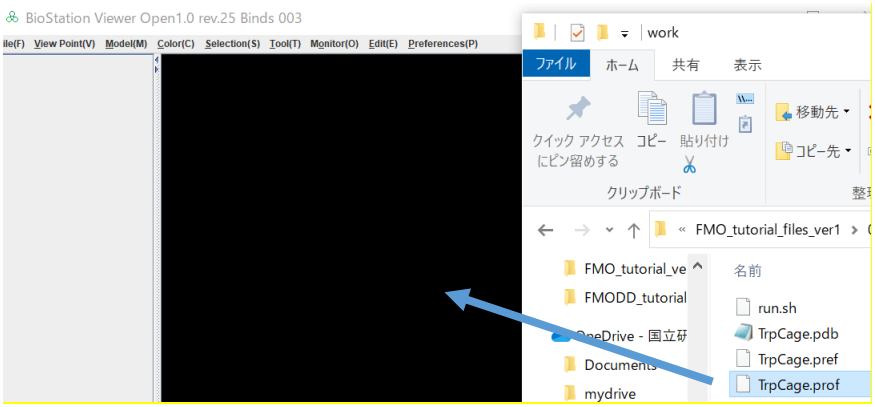
以上で、IFIE/PIEDA 解析は終了しました。これで、Viewer を閉じてしまっても構いません。ですが、この段階での Viewer の設定を保存して、後で、解析を再開したい方もいると思います。その場合は、以下のように PROF ファイルを作ります。File > Save Profile File (V)を実行し、開いた Save ウィンドウに、.prof という拡張子のファイル名を入れ（ここでは TrpCage.prof）、save を押します。

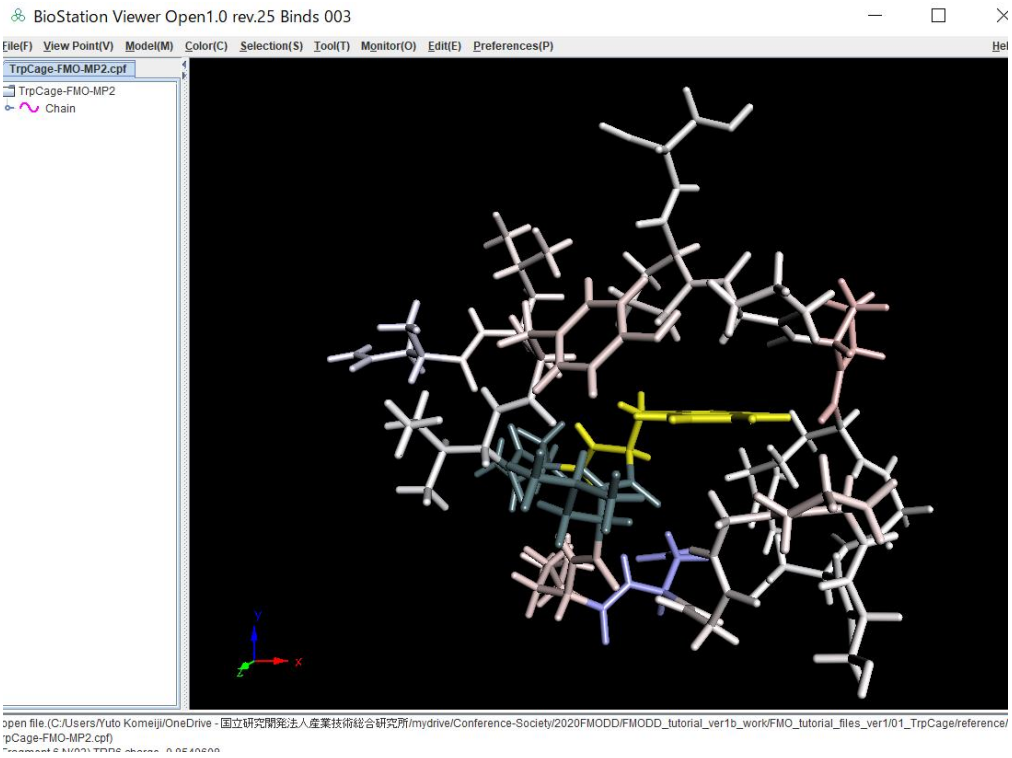


すると、TrpCage.prof と、もう一つ、TrpCage.pref というファイルが作成されます。後者をテキストエディターで開いてみると、Viewer の設定情報（preference）が記述されていることがわかります。

```
// Creation by BIO Station Viewer Version Open1.0 rev.25 Binds 003
// Preference file
// Date : Thu Mar 12 14:20:18 JST 2020

// rotaion speed
ROTATION_SPEED    = 20000

// connect atom
CONNECT_ATOM_SCALE   = 1.0

// resolution
LINE_RESOLUTION   = 1
CA_LINE_RESOLUTION= 3
BALL_RESOLUTION   = 32
STICK_RESOLUTION  = 32
CPK_RESOLUTION    = 64
CA_TUBE_RESOLUTION   = 64

// radius
BALL_RADIUS          = 0.36
BALL_AND_STICK_RADIUS = 0.1
...
```

Viewer をもう一度開いて、作成した TrpCage.prof を drag&drop すると、PROF ファイルを作った時と同じ状況が再現されます。





この画面で、中断した作業を再開することができます。なお、メッセージは出ませんが、PREF ファイルも同時に読み込まれています（もし、PREF を消してしまった場合は、Viewer が警告メッセージを出します）。

PROF ファイルは、Viewer の任意の作業段階で作成することができます。このチュートリアルではこの機能にはこれ以上触れませんが、必要に応じて、作成してください。

これで、課題 1 はおしまいです。課題 1 では FMO 計算の典型的な手順と結果の解析方法を学習しました。ここでの経験に基づいて、後続の課題に取り組んでください。

2　課題 2：エストロゲン受容体－リガンド複合体の IFIE 解析

本課題ではエストロゲン受容体タンパク質（Estrogen Receptor, ER）とそのリガンド（Estrogen, EST）複合体の FMO 計算を行い、IFIE 解析を行います。タンパク質全体は大きすぎるので、ここでは 50 残基だけ抜き出してモデル化した構造（以下、ER_50）を用います。FMO 法は薬剤設計において受容体とリガンド相互作用の計算に広く使われていますが、本課題の対象は、その典型例です。

2.1 入力データ作成

課題 2 では課題 1 で作成した HF 用の AJF ファイルを編集して入力データを作成します。まず、課題 1 で作成した AJF ファイルを別名でコピーします。もし、時間に余裕があれば、HF だけでなく、MP2 についても、同じようにコピーして AJF ファイルを作ってください。

```
cp TrpCage-FMO-HF.ajf ER_50res-FMO-HF.ajf
```

この ER_50res-FMO-HF.ajf を vi などのテキストエディターで開いてください。

```
vi ER_50res-FMO-HF.ajf
```

CNTRL ネームリストの ReadGeom キーワードと WriteGeom キーワードを変更します。

```
&CNTRL
…
ReadGeom='ER_50res.pdb'
WriteGeom='ER_50res-FMO-HF.cpf'
…
/
```

これで入力データの作成は完了です。

2.2 FMO 計算

FMO 計算の実行方法は課題 1 の時と同じです。まず、run.sh ファイルを編集してください。HF 計算だけ実行し、計算時間が長い MP2 ジョブをスキップする場合は、以下の MP2 部分はコメントアウトしてください。その場合、2.3 節以降の IFIE/PIEDA 解析では、reference として用意された出力ファイルを解析してもらうことになります。

```
#!/bin/sh

NPALL=24   # total number of processors
ABINIT=/home/komeiji/ABINIT-MP_rev15/AIST-KMJ/abinitmp

# HF
JOB=ER_50res-FMO-HF                # define & describe JOB
rm -f $JOB.log $JOB.err $JOB.cpf  # remove old output files
mpirun -np $NPALL $ABINIT< $JOB.ajf 1> $JOB.log 2>$JOB.err
                                   # execute FMO
# MP2
# NB: comment out the following lines if you can't wait for long
```

```
JOB=ER_50res-FMO-MP2                # define & describe JOB
rm -f $JOB.log $JOB.err $JOB.cpf  # remove old output files
mpirun -np $NPALL $ABINIT< $JOB.ajf 1> $JOB.log 2>$JOB.err
                                    # execute FMO
```

計算が完了すると ER_50res-FMO-HF.log と ER_50res-FMO-HF.cpf が出力されるのでこれらのファイルを Viewer がインストールしてある PC にコピーします。

2.3 IFIE 解析

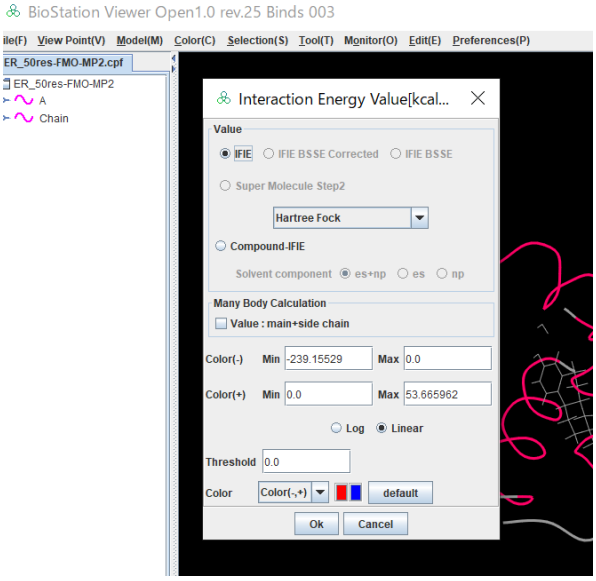
CPF ファイルを Viewer の画面にドラッグ&ドロップして読み込みます。前節で、MP2 計算まで実行した場合は、ご自分の CPF ファイルを使ってください。

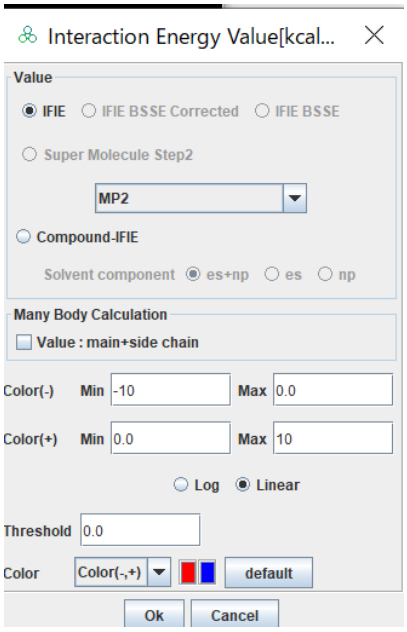
02_ER/**work**/ER_50res-FMO-MP2.cpf

そうでない場合は、reference 内の CPF ファイルを利用します。

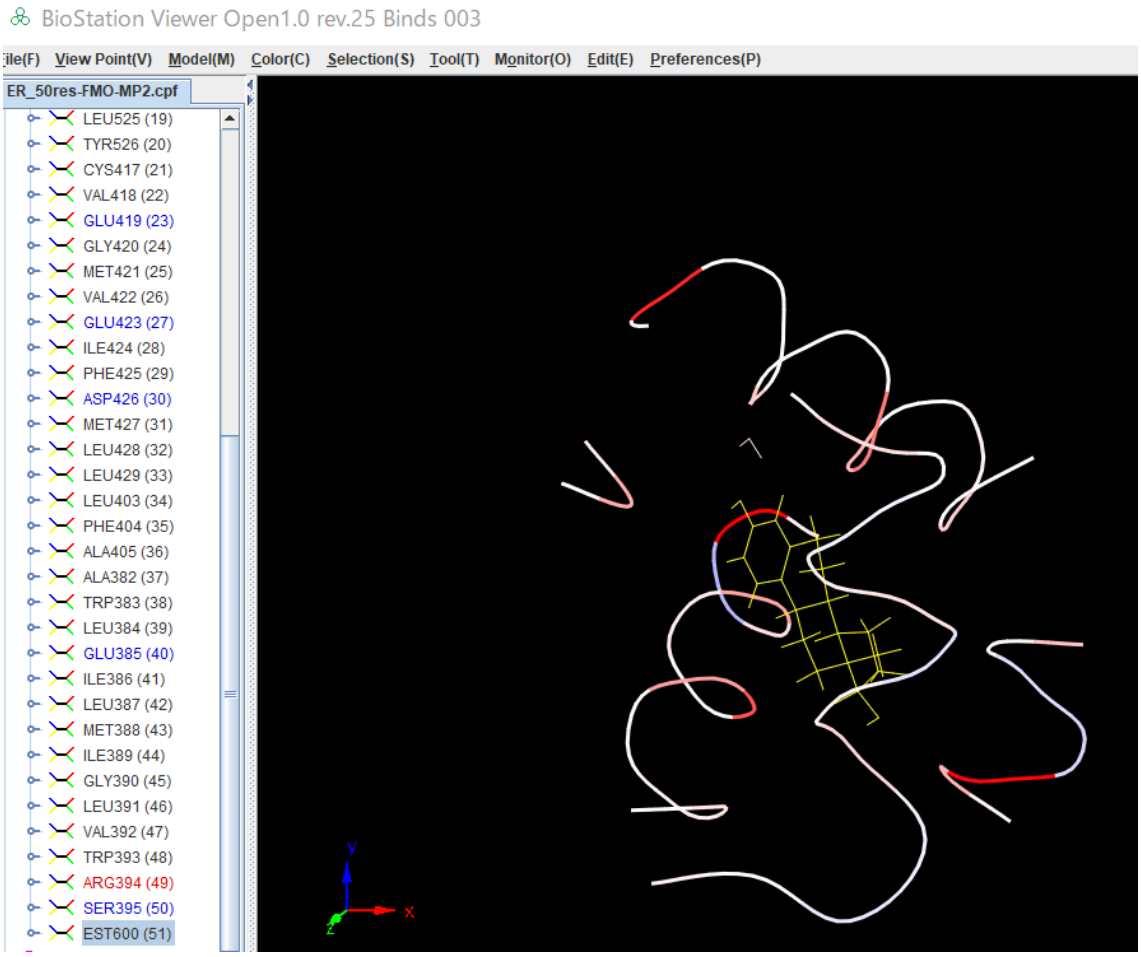
02_ER/**reference**/ER_50res-FMO-MP2.cpf

リガンドをクリックし、Monitor→Interfragment Interaction→1:1 を実行します。Value のプルダウンは MP2 を選択し、Color のテキストボックスでは-10 から 10 とします。

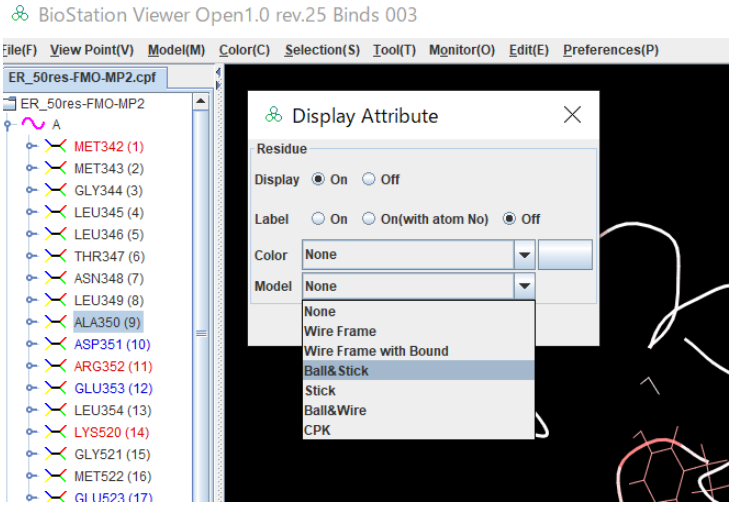


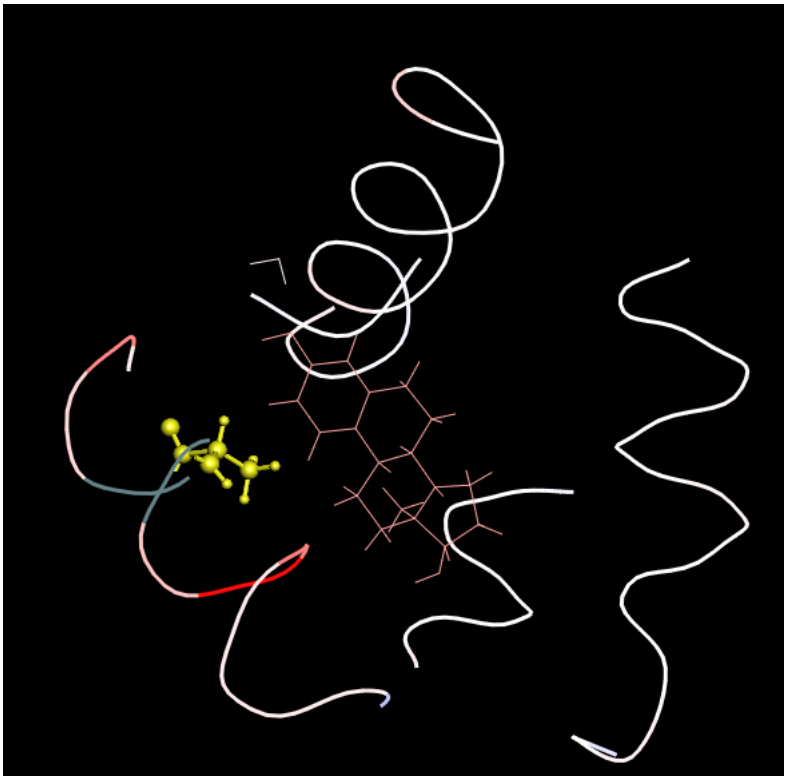


左のウィンドウから、フラグメント（アミノ酸残基またはリガンド）を選択します。 すると選択したフラグメントを囲むフラグメントが、選択したフラグメントとの IFIE 値に基づいて色付けされます。 次の例では、リガンド（EST600）が選択されています。

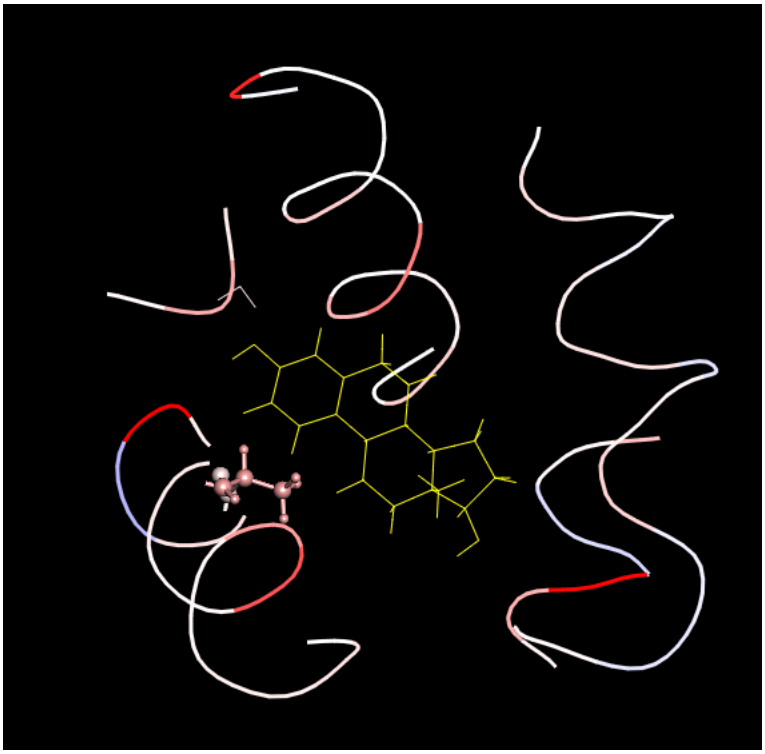


別のフラグメントをクリックすれば、そのフラグメントとの IFIE に基づいて周囲のフラグメントを色付けすることができます。ここでクリックした ALA350 を Ball&Stick 表示にしてみましょう。画面左のツリー構造にて ALA350 を右クリックし、Display Attribute ウィンドウで Model を Ball&Stick にします。





再びリガンドに焦点を戻す(リガンド EST600 をクリックする)と、リガンドと ALA350 が相互作用する様子を分子構造と IFIE を照らし合わせて見ることができます。



また、Monitor>Interfragment Interaction>List を実行すると注目しているフラグメントとその周囲のフラグメント間の IFIE が kcal/mol 単位で表示されます。

File(F)

Value:MP2
Base fragment EST600 (51)

residue	fragment	[kcal/mol]
MET342	(1)	0.017124
MET343	(2)	-1.069078
GLY344	(3)	-0.589014
LEU345	(4)	-0.269404
LEU346	(5)	-3.760819
THR347	(6)	-6.700130
ASN348	(7)	-0.278450
LEU349	(8)	-0.769325
ALA350	(9)	-3.594809
ASP351	(10)	-1.159857
ARG352	(11)	2.832267
GLU353	(12)	-41.071605
LEU354	(13)	-0.598211
LYS520	(14)	-2.445991

これで、課題２は終了です。FMO が、創薬などの対象になる「リガンドと受容体」間の相互作用解析に有効なことが実感いただけたと思います。

3 課題３：Formamide と水の複合系の PIEDA 解析

本課題では Formamide と水の複合系に対する FMO-MP2 計算を行い、PIEDA 解析を行います。この課題は、水溶液中の低分子化合物の解析例として作成しました。計算も解析も短時間で終わります。

3.1 入力ファイル作成

課題 3 では、課題 1 で作成した MP2 用の AJF ファイルを編集して入力データを作成します。まず、課題 1 で作成した MP2/6-31G*の AJF ファイルを別名でコピーします。

```
cp TrpCage-FMO-MP2.ajf Formamide-FMO-MP2.ajf
```

コピーした Formamide-FMO-MP2.ajf を編集します。CNTRL ネームリストの ReadGeom キーワードと WriteGeom キーワードを変更します。

```
&CNTRL
…
ReadGeom='Formamide.pdb'
WriteGeom='Formamide-FMO-MP2.cpf'
…
/
```

以上で入力データの作成は完了です。

3.2 FMO 計算

FMO 計算の実行方法は課題 1 の時と同じです。run.sh スクリプトを編集して AJF ファイル名を変更した上で計算を流します。なお、本課題の系は 3 フラグメントとなるため、run.sh 内で“NP=12”として計算を行ってください(1フラグメントあたりのコア数をNP=4に設定したので、それが3つで4x3=12)。また、HF 計算は行いません。run.sh は以下になります。

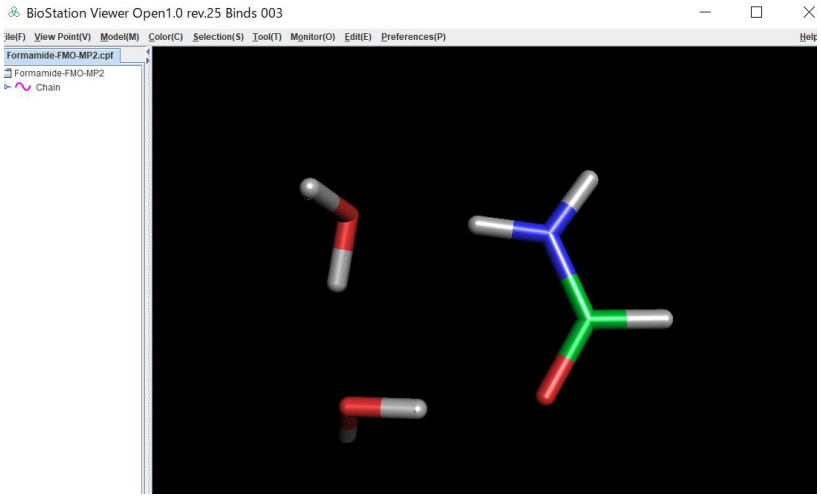
```
#!/bin/sh

NPALL=12   # total number of processors
ABINIT=/home/komeiji/ABINIT-MP_rev15/AIST-KMJ/abinitmp

# MP2
```

```
JOB=Formamide-FMO-MP2                # define & describe JOB
rm -f $JOB.log $JOB.err $JOB.cpf  # remove old output files
mpirun -np $NPALL $ABINIT< $JOB.ajf 1> $JOB.log 2>$JOB.err
                                     # execute FMO
```

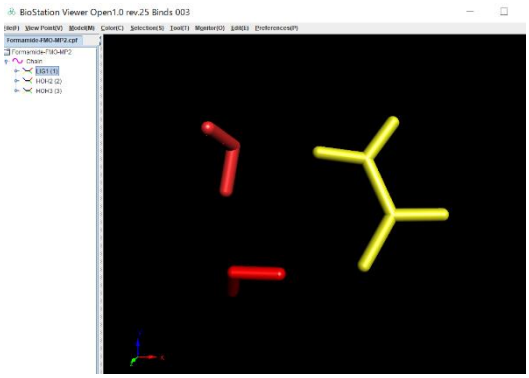
計算が完了すると Formamide-FMO-MP2.log と Formamide-FMO-MP2.cpf が出力されるのでこれらのファイルを PC にコピーします。

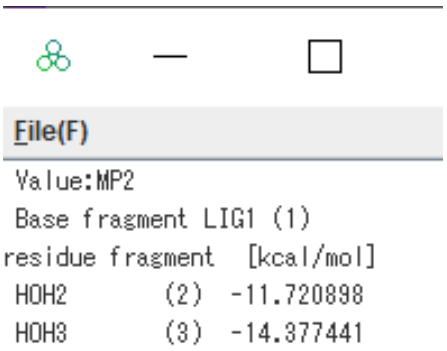
3.3 IFIE･PIEDA 解析

得られた CPF ファイルを Viewer の画面にドラッグ&ドロップして読み込みます。



IFIE の可視化には、Monitor > Interfragment Interaction > 1:1 を実行します。Value のプルダウンは MP2 を選択します。Formamide をクリックすると、Formamide に対する IFIE が可視化されます。さらに、Monitor > Interfragment Interaction > List から、Fromamide に対する MP2 IFIE のリストを表示すると、以下の様になります。

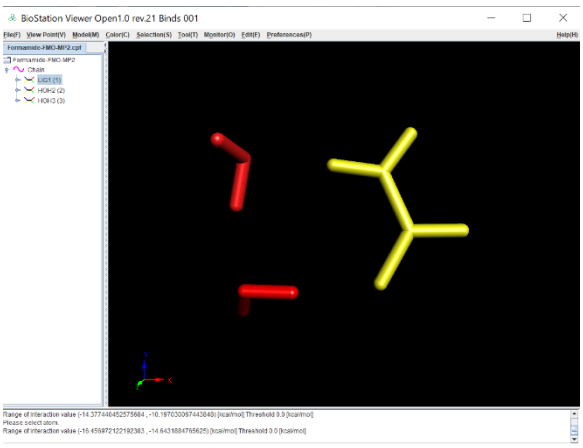




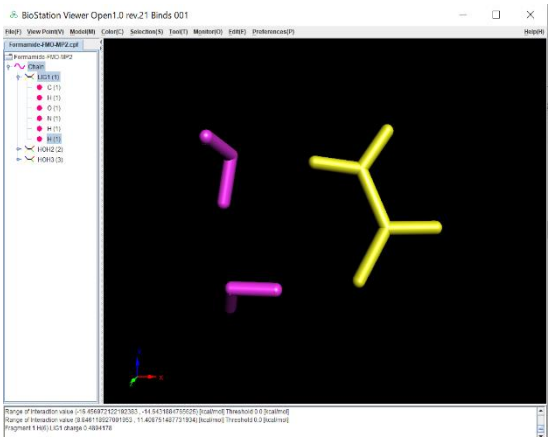
次に、PIEDA の各成分を可視化してみます。PIEDA の各成分を表示する場合には、Electrostatic（ES

項)、PIEDA EX（EX 項）, PIEDA CT+mix (CT 項)、, MP2 Correlation (DI)（DI 項）をそれぞれ選択することで、相互作用のカラー表示やエネルギーリストを得ることができます。

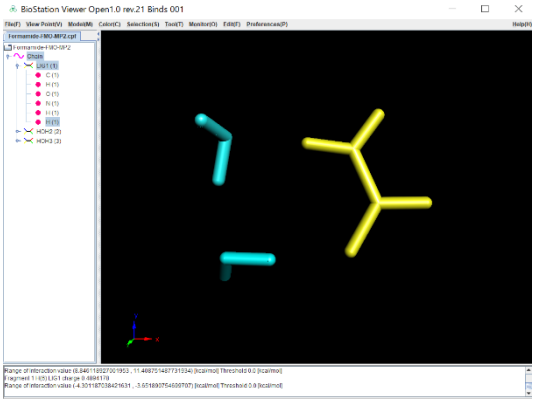
Electrostatic (ES)



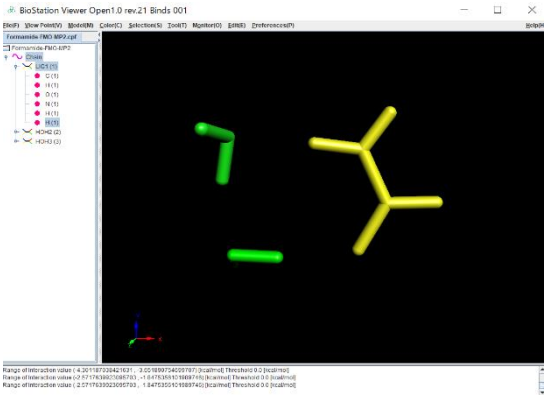
PIEDA EX



PIEDA CT+mix



MP2 Correlation (DI)



また、Formamide-FMO-MP2.log の中にも、以下の様に各フラグメント間の PIEDA の値が書き出されています。テキストエディターで開き“PIEDA”を検索すると、ファイルの後半に以下の箇所を見るけることができます。

```
## PIEDA

IJ-PAIR      ES           EX           CT+mix       DI(MP2)      q(I=>J)
/          kcal/mol    / kcal/mol    / kcal/mol    / kcal/mol    / e
      ---------------------------------------------------------------------------
2   1     -14.643187     8.846118     -3.651890     -2.271937     -0.043483
3   1     -16.456971     8.952482     -4.301187     -2.571764      0.040811
3   2     -15.618868    11.408751     -4.139376     -1.847535     -0.044469
```

課題３はこれで終了です。FMO は、巨大分子だけでなく、このように、溶媒中の低分子の解析にも利

用されています。

4　課題 4：軌道レベルでの電荷移動相互作用解析（CAFI 解析）

本課題では水の 3 量体を題材に、CAFI を用いて軌道レベルでの電荷移動相互作用解析の方法を説明します。課題 4 の CAFI と課題 5 の FILM は、軌道レベルのデータを直接可視化しますので、IFIE/PIEDA と比べて、一段と量子化学的な解析方法と言えるかも知れません。

4.1　入力データの作成

以下の pdb ファイルを BioStation Viewer で開きます。

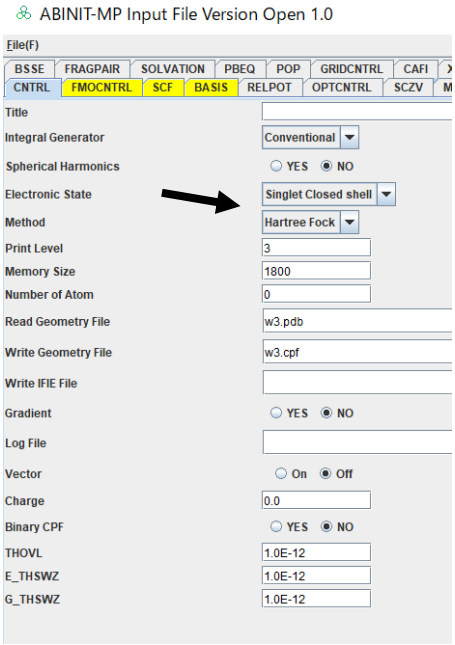
```
04_CAFI/work/w3.pdb
```

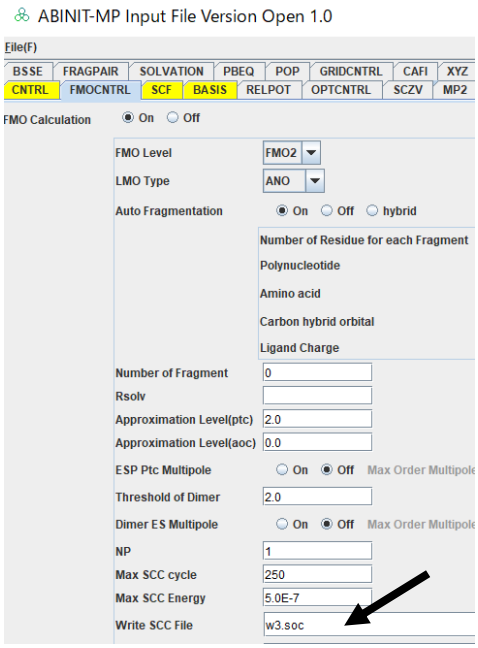
CAFI の計算は以下の 2 段階で実施します。

(1) 通常の FMO 計算を実施し、scc ファイル、cpf ファイルを作成する。(w3-1st.ajf)

(2) CAFI 計算を実施し、可視化用の pno ファイルを作成する。(w3-2nd.ajf)

そこで、File→Edit ABINIT-MP Input File から AJF ファイル作成画面を開き、2 つの ajf ファイルを作成します。まずは、w3-1st.ajf です。CNTRL タブで Method を"Hartree Fock"とし、FMOCNTRL タブの Write SCC File にて "w3.scc" を指定してください。その上で、File→Save File から "w3-1st.ajf" として保存してください。



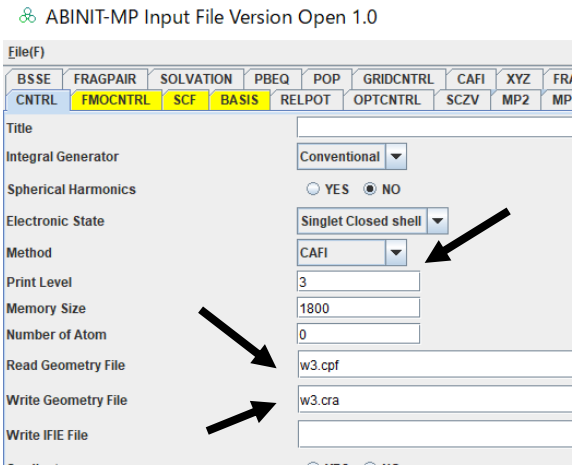


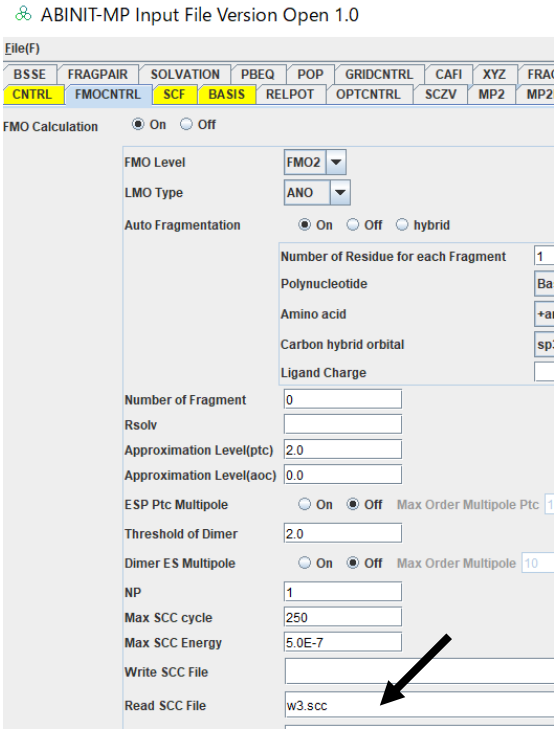
さらに、保存した "w3-1st.ajf" をテキストエディターで開き、&CNTRL に以下の様に CPFVER=4.2 を

追記してください。

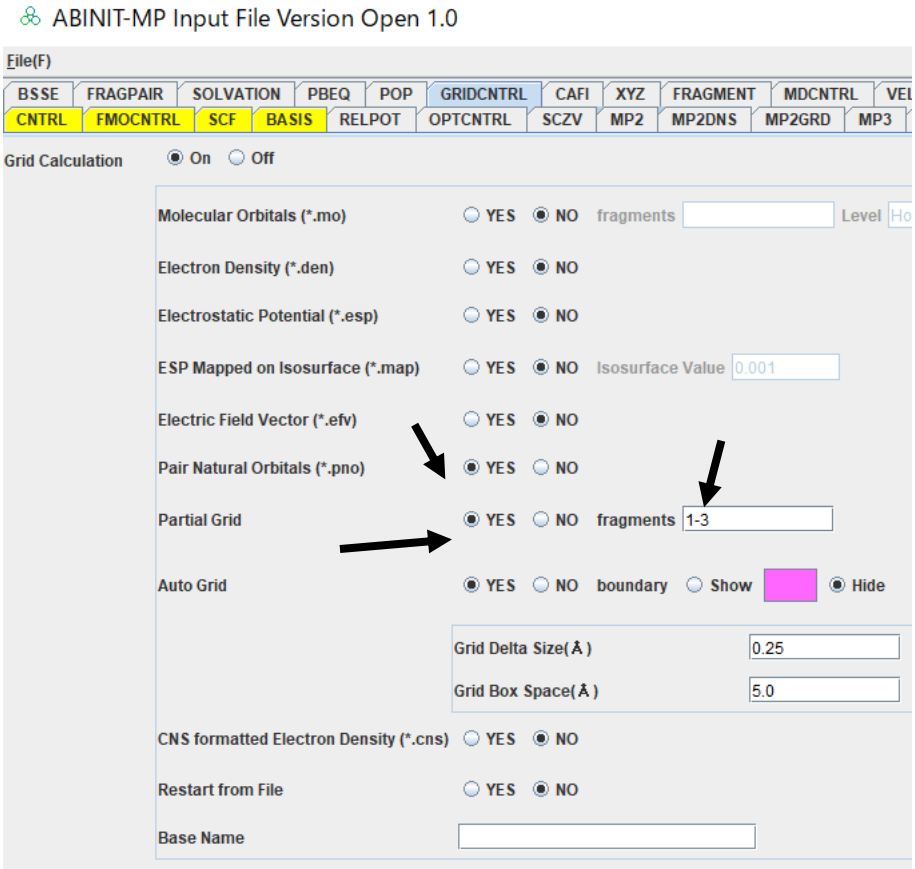
w3_1st.ajf
&CNTRL ElecState='S1' Method='HF' Nprint=3 Memory=1800 Natom=0 ReadGeom='w3.pdb' WriteGeom='w3.cpf' Gradient='NO' Vector='OFF' CPFBIN='NO' THOVL=1.0E-12 E_THSWZ=1.0E-12 G_THSWZ=1.0E-12 **CPFVER=4.2** /

続いて、2 つ目の入力ファイル“w3_2nd.ajf”を作成します。上記と同様に、ABINIT-MP 入力ファイル作成画面を開き、CNTRL タブにて Method を“CAFI”としてください。さらに、Read Geomrtry File を“w3.cpf”、Write Geometry File を“w3.cra”としてください。また、FMOCNTRL タブで Write SCC File の“w3.scc”を削除し、Read SCC File に“w3.scc”を記入してください。

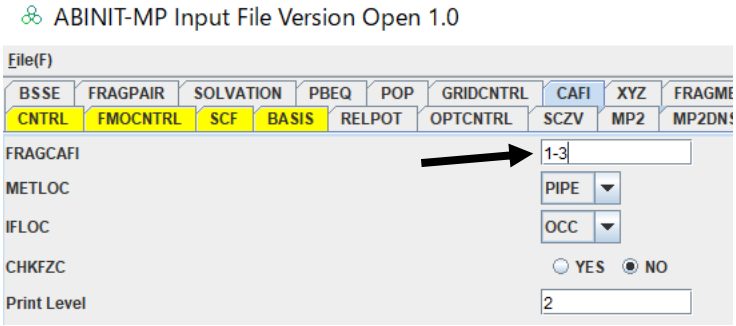




次に、GRIDCNTRL タブにて Grid Calculation を“On”とすると、次の画面になりますので、Pair Natural Orbitals(*.pno)を“YES”とし、Partial Grid も“YES”とした上で fragments を“1-3”と指定してください。



さらに、CAFI タブにて FRAGCAFI を“1-3”と指定してください。



作成した二つの AJF ファイルに namelist の LRD があるかどうか確認し、なければ、“&LRD”と“/”を加えてください（詳細は、1.1 節の　%注意　参照）。

```
&LMP2
/
```

```
&LRD
/
&DFT
/
```

作成した 2 つの AJF ファイルを Linux 計算サーバーに転送してください。

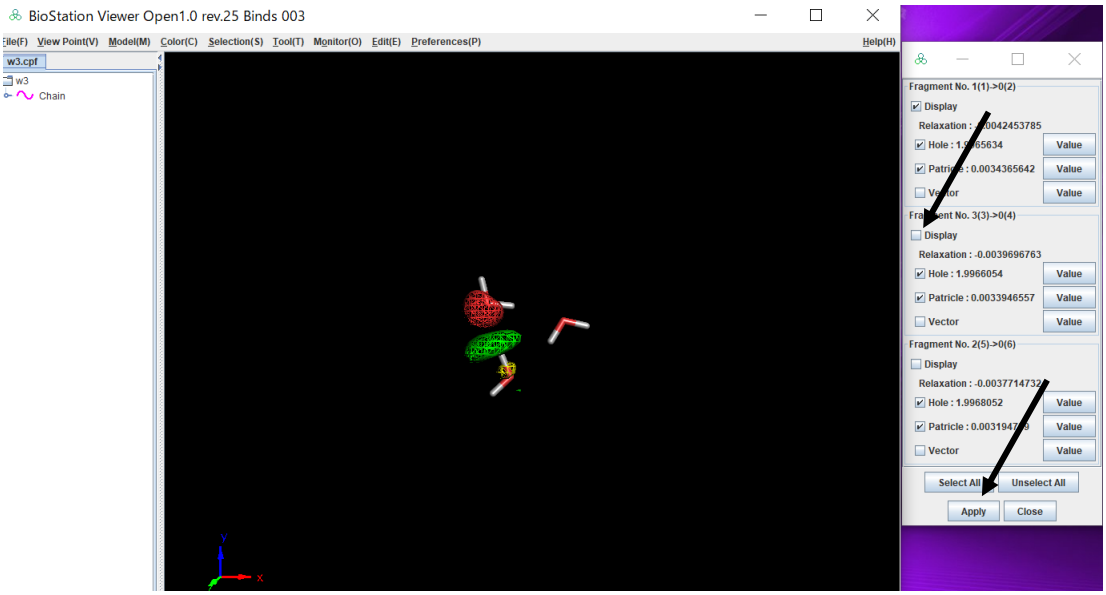
4.2 FMO 計算

FMO 計算の実行方法は課題 1 の時と同じです。本課題の系は 3 フラグメントとなるため、run.sh 内で“NPALL=3”とします。このスクリプトを実行すれば、1 回目のジョブ（w3-1st）の計算完了後、2 回目のジョブ（w3-2nd）が流れます。計算が完了すると生成されるファイルのうち、w3.cpf と w3.pno をダウンロードしてください。

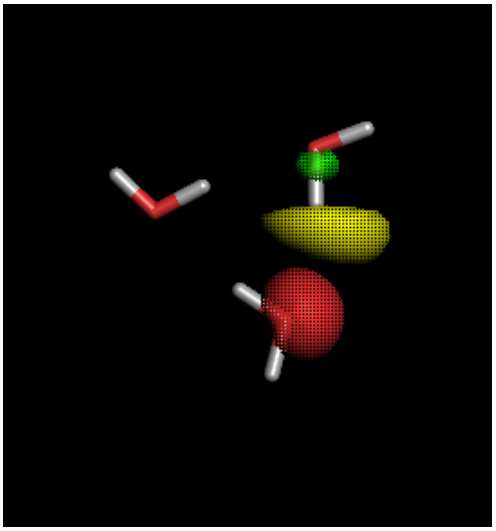
```
#!/bin/sh
NPALL=3   # total number of processors
ABINIT=/home/komeiji/ABINIT-MP_rev15/AIST-KMJ/abinitmp
# rm old files
rm -f w3.pno w3.cpf w3.cra w3.scc

# 1st job
JOB=w3-1st                        # define & describe JOB
rm -f $JOB.log $JOB.err           # remove old output files
mpirun -np $NPALL $ABINIT< $JOB.ajf 1> $JOB.log 2>$JOB.err
                                  # execute FMO
# 2nd job
JOB=w3-2nd                        # define & describe JOB
rm -f $JOB.log $JOB.err           # remove old output files
mpirun -np $NPALL $ABINIT< $JOB.ajf 1> $JOB.log 2>$JOB.err
                                  # execute FMO
```

4.3 CAFI 解析

Viewer にまず CPF ファイル、ついで PNO ファイルをドラッグ＆ドロップすると、左のように、デフォルトでフラグメント 1 の軌道が描画され、また、右の CAFI の画面がポップアップで表示されます。



ポップアップされた画面には、"Relaxation energy"、"Hole（orbital）"の占有数、"Particle (orbital)"の占有数が表示されます。可視化したい軌道の"Display"にチェックを入れ、"Apply"を押すと、下図の様にチェックした軌道が描画されます。（下図は、Fragment No.3 を選び、Hole, Particle の Value を押すとポップアップで表示される設定画面にて、Draw Type を Surface、Transparency を 20 としたものです。）



5　課題 5：軌道レベルでの分極相互作用解析（FILM）

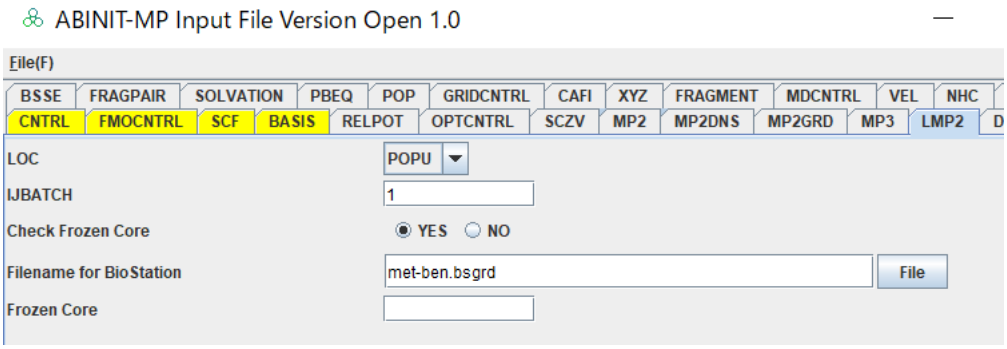
本課題では、メタン - ベンゼンのモデル系を題材に、fragment interaction analysis based on Local MP2 (FILM)を用いた、軌道レベルでの分極相互作用解析を行います。

5.1　入力データの作成

以下の pdb ファイルを BioStation Viewer で開きます。

```
05_FILM/work/met-ben.pdb
```

File>Edit ABINIT-MP Input File で AJF ファイル作成画面を開き、CNTRL タブにて Method を“LMP2”に設定します。次に、LMP2 タブにて Filename for BioStation に“met-ben.bsgird”を記入します。その上で File>Save File で、“met-ben.ajf”として保存してください。



さらに、"&LRD"と"/"を手で付けくわえてください（1.1 節 %注意　参照）。

5.2　FMO 計算

FMO 計算の実行方法は課題 1 の時と同じです。ただし、本課題では、run.sh 内で“NPALL=1”とした上で計算を行ってください。生成されたファイルの中で、"met-ben.cpf"と"met-ben.bsgrid_2_1.lmp2"を Windows PC に転送して、その後の FILM 分析を行います。

%注意：LMP2 ファイルを保存するディレクトリー（フォルダー）

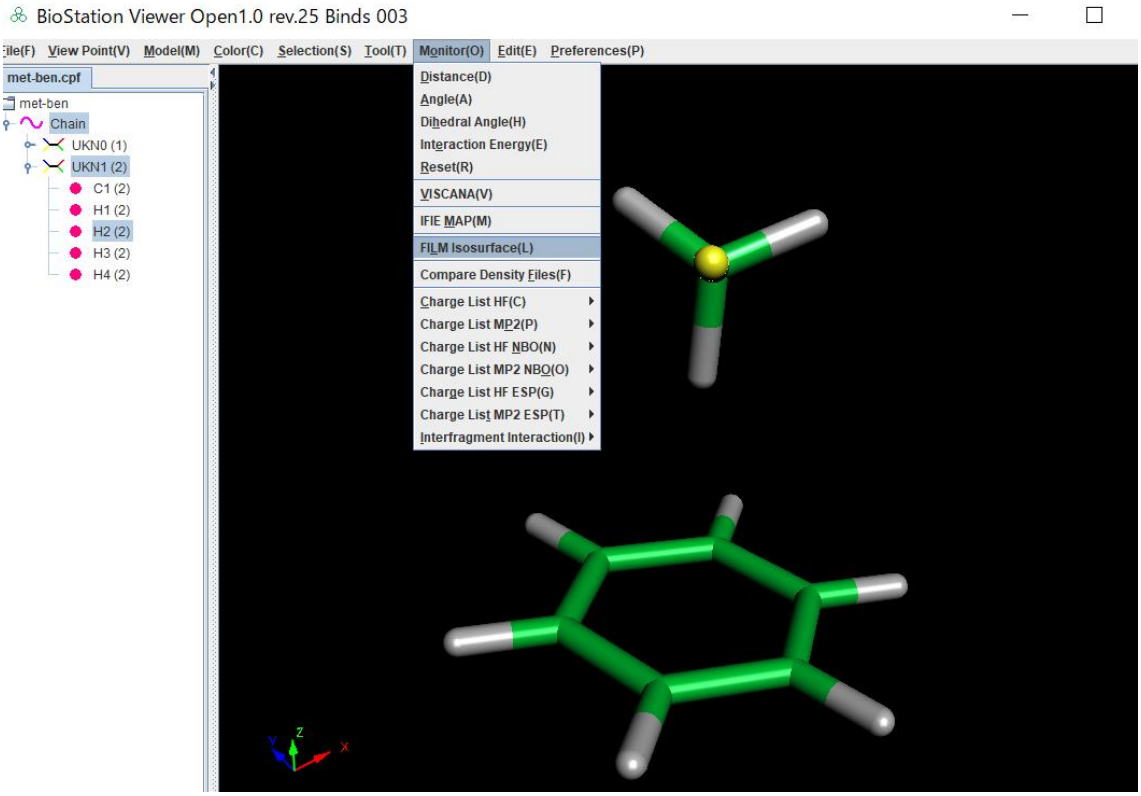
これは、BioStation Viewer Open 1 Rev. 21 002 以降のバージョンを使っている方は、読む必要はありません。それ以前のバージョンを使っている方は、LMP2 ファイル"met-ben.bsgrid_2_1.lmp2"は、中にサブディレクトリーを含まないディレクトリーに保存する必要があります。さもないと、後続の FILM 解析は描画の段階で停止することがあります (停止しないこともあります)。今回は、work/の中に LMP2

ファイルを保存しますが、work/内にはサブディレクトリーを作らないでください。

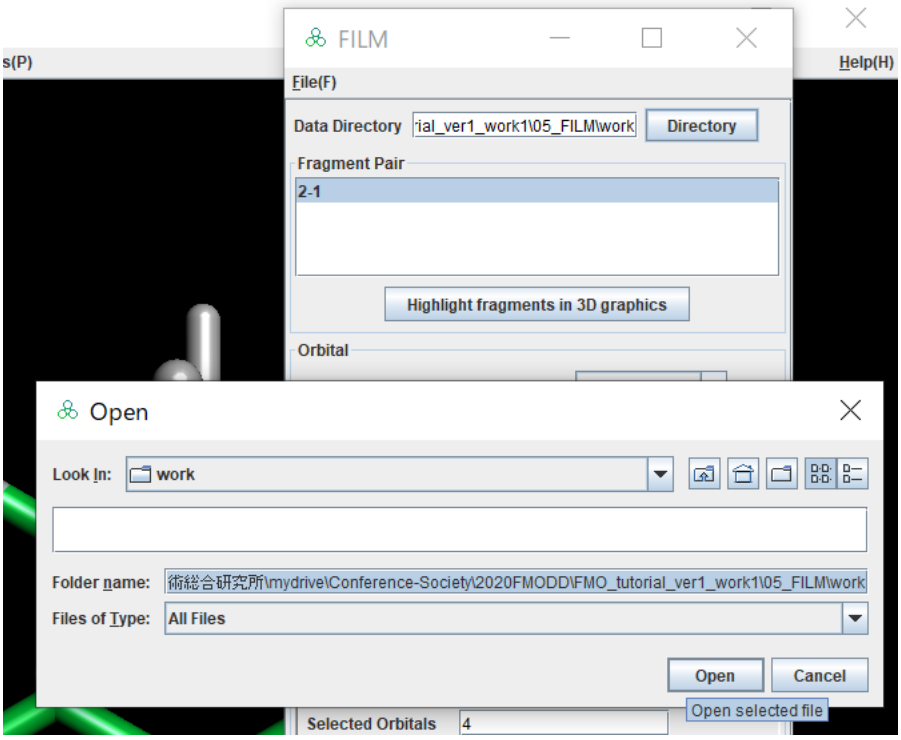
05_FILM/work/met-ben.bsgrid_2_1.lmp2

5.3 FILM 解析

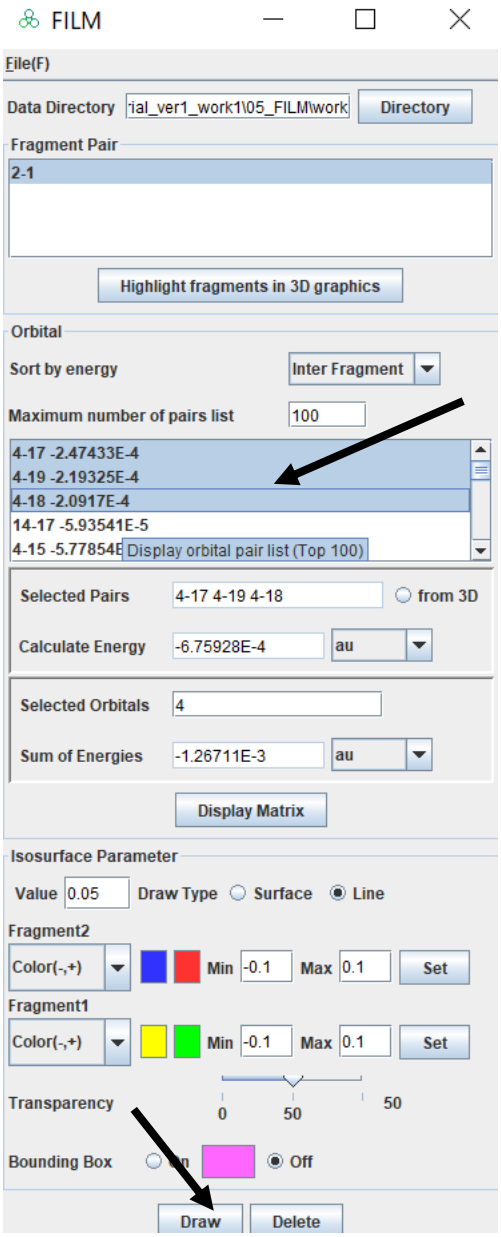
CPF ファイルを Viewer で読み込んだ後、Monitor>FILM Isosurface をクリックします。

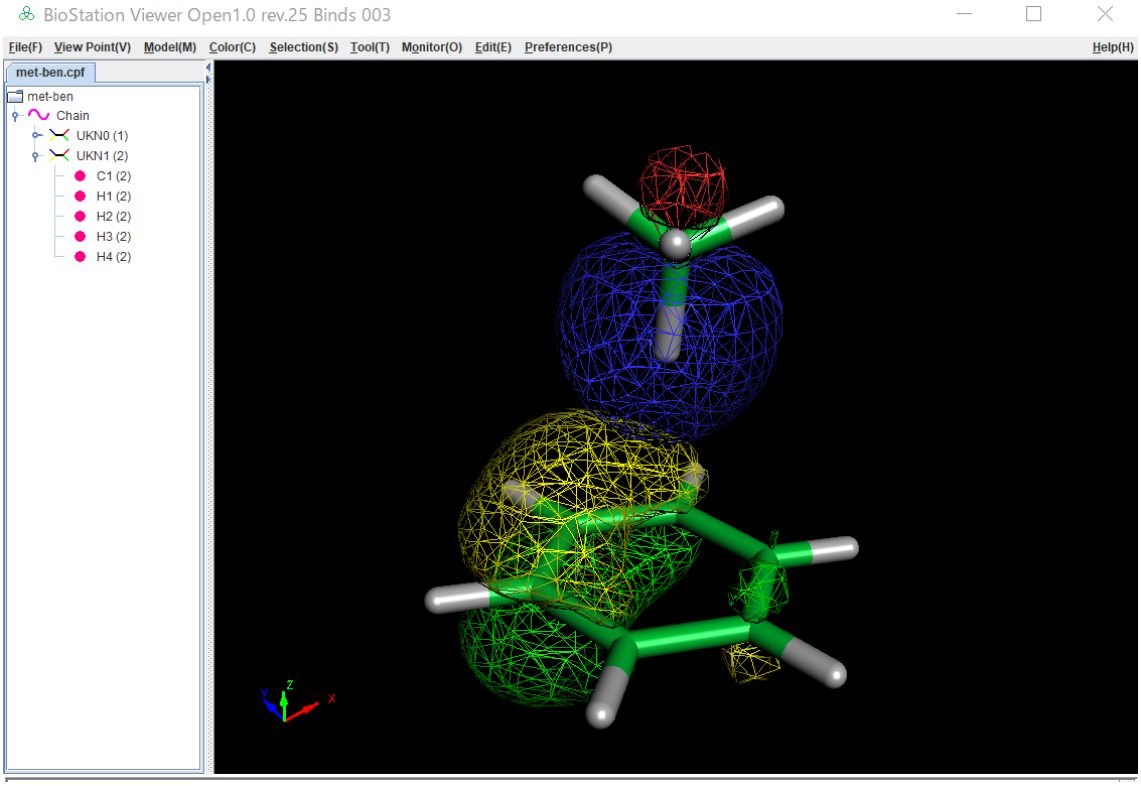


すると、FILM 設定画面が表示されますので、Data Directory として、“met-ben.bsgrid_2_1.lmp2”がおいてある"work/"を指定してください。

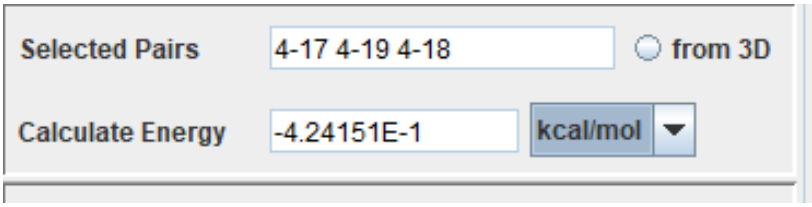


FILM 設定画面の Orbital 部は複数選択が可能（Shift を押しながらクリック）となっているため、4-17, 4-19, 4-18 を選択した状態で Draw を押すと、ベンゼンのπ軌道とメタンの CH 間にある CH-π相互作用の様子を可視化することが出来ます（もし、Viewer が考え込んでしまったときは、5.2 節の%注意を読んでやり直してください）。Isosurface Parameter の Value や Transparency を変更することで、可視化の調整が可能です。


FILM
File(F)
Data Directory
ial_ver1_work1\05_FILM\work
Directory
Fragment Pair
2-1
Highlight fragments in 3D graphics
Orbital
Sort by energy
Inter Fragment
Maximum number of pairs list
100
4-17 -2.47433E-4
4-19 -2.19325E-4
4-18 -2.0917E-4
14-17 -5.93541E-5
4-15 -5.77854E
Display orbital pair list (Top 100)
Selected Pairs
4-17 4-19 4-18
from 3D
Calculate Energy
-6.75928E-4
au
Selected Orbitals
4
Sum of Energies
-1.26711E-3
au
Display Matrix
Isosurface Parameter
Value
0.05
Draw Type
Surface
Line
Fragment2
Color(-,+)
Min
-0.1
Max
0.1
Set
Fragment1
Color(-,+)
Min
-0.1
Max
0.1
Set
Transparency
0
50
50
Bounding Box
Off
Draw
Delete



また、FILM 設定画面の中央部にある Calculate Energy から対応するエネルギーの確認ができます。その際、エネルギーの単位系として au 及び kcal/mol が選択可能です。



以上、課題 4 と 5 では、軌道に関するエネルギーと描画を行うことで、より量子化学的な解析法を試していただきました。

まとめ

以上、課題 1－5 を通じて、BioStation Viewer と ABINIT-MP プログラムを用いた FMO 計算の様々な側面に触れていただきました。ここで解説した機能を使っただけでも、実践的な FMO 計算が実行できると思います。

なお、今回の課題では、FMO 計算対象の分子構造は、すべてモデリング済みの構造を PDB 形式にして、配布してあります。しかし、ユーザーにとっては、分子構造をどうモデリングするかというのが最初の関門になります。PDB から取って来たファイルをそのまま計算できることは、通常ありません。水素など欠けている原子、溶媒分子、イオンなどの追加をはじめ、対象と目的によって、さまざまなモデリング作業が必要になります。

モデリングだけでなく、FMO に関する問題や情報を共有する場が FMODD ですので、気軽にご相談いただけると幸いです。

連絡先：fmodd.consortium@gmail.com