

実習の概要

1. はじめに
2. 使用するプログラムとデータ
 - 1) 配布プログラムとデータ
 - 2) データの説明
3. モデリングを始める前のお膳立て
 - 1) PDB サイトから X 線構造データを入手する。
 - 2) PDB ファイル(3arc.pdb)に書かれている情報を調べる
 - 3) 欠落している原子を調べる
4. ps2 モノマー” 安直モデル” の作成
 - 1) モノマーの PDB ファイルを作る
 - 2) モノマーをポリペプチド(pro)、非ペプチド(chm)と結晶水(wat)部分に分割する
 - 3) pro に水素原子を付加する
 - 4) chm に水素原子を付加する
 - 5) wat に水素原子を付加する
 - 6) pro、chm、wat を統合してモノマーを再構成する
 - 7) ヒスチジンの δ または ϵ 位のプロトンを削除して、” 安直モデル” を完成する
 - 8) 溶媒和クラスターモデルを作成する(おまけ)
 - 9) 作成した構造データの品質検査を行う
5. ps2 モノマー” 修復モデル” の作成
 - 1) ポリペプチド鎖の軽い構造最適化 (TINKER を利用)
 - 2) 欠落原子がある非ペプチド分子の修復と構造最適化 (GAMESS を利用)
6. ps2 モノマー” 最適化モデル” の作成
 - 1) 環境中のポリペプチド分子の構造最適化
 - 2) 環境中の非ペプチド分子の構造最適化
7. 縮小モデルの作成
8. OEC クラスターの置換
9. おわりに

附録 A PDB ファイルの命名ルール

附録 B 残基の記述式

附録 C FU で特別な意味を持つファイル識別子

附録 D 中間データを保存するディレクトリの構造

1. はじめに

巨大（10 万原子規模）で複雑な分子系（ポリペプチド以外に多様な化合物が多数含まれる）の QM/MM 計算のための構造モデルを、1）効率的、かつ、2）再現性を担保して作成する手法を実習する。

2. 使用するプログラムとデータ

1) 配布プログラムとデータ

①fu-28Apr2014 ... 実習で用いる FU 開発版 (Windows7 32bit mode 版)

このフォルダーを C:¥ドライブ直下にコピーする。doc フォルダーに、説明書 1 式があるので、「fu-UsersGuide-ver0.1.1.pdf」で本講習会で使用する fumodel、また、「futools 使用説明書-28Apr2014.pdf」で futools の使い方をご覧ください。

②ps2 ... このフォルダーを C:¥ドライブ直下にコピーする。ここに実習の中間データや成果物を格納する。実習に便利のように、配布データの一部をここにコピーしてある。実習のすべての項目を行えば、中身が ps2-example と同じになる。

③ps2-material ... 実習で参考にする ps2 関連の資料

④ps2-example ... 実習の結果の例

⑤FU による PS II 構造モデリング実習テキスト.pdf (本テキスト)

2) データの説明

①ps2-material

./pdb-data ... PDB サイトで ps2 で検索してヒットした PDB データの一部(ファイル名が pdbxxx.ent である。)

./oec-models

batista-oec-2011 ... Batista らの OEC モデル

neese-oec-2012 ... Neese らの OEC のモデル

siegbahn-oec-2013 ... Siegbahn らの OEC モデル

./mhtfiles ... 非ペプチド分子のフレームデータファイル

②ps2-example

3arc.pdb ... 実習を行う ps2 の pdb データ (./pdb-data/pdb3arc.ent のコピー)
上記を含めて実習で作成するデータの見本。

このフォルダーを C:ドライブ直下にコピーして、c:¥ps2 と rename すると、本実習を一から全てやった結果と同じ状態になる。この状態にしておくと、実習の途中のどの操作をも再度試みることができる。ただし、操作で生成されるファイルは元のを上書きする。

3. モデリングを始める前のお膳立て

- 1) ps2 の構造データを PDB サイト (<http://pdj.org/>) から収集する
 - ① モデリングの元にするデータを決める（一般的には、最も解像度が高いデータを選ぶ。今回は 3arc.pdb を選択）。この時、複合体に含まれている非ペプチド分子の種類やその構造などを、「構造情報」を見て調べる。
 - ② モデリングの中間データを格納するフォルダー、c:/ps2/3arc と c:/ps2/3arca を作成し、前者に ps2-material/pdb-data/pdb3arc.ent を 3arc.pdb 名でコピーする。
（補足）本講習会の配布データの ps2 フォルダを c:/¥にコピーするとよい。
 - ③ まず、fumodel(32 ビット版ではメモリ不足になる) を起動して、” 3arc.pdb ” を眺める（PyMol などの高速ビューアソフトを使うほうがよい）。ここで、ps2 が巨大・複雑であることを実感してほしい。
- 2) エディタで 3arc.pdb ファイルの情報を読む。
 - ① 欠落情報をチェックする (Missing chains, residues and atoms)

リスト1 PDB MISSING RESIDUES

REMARK 465 MISSING RESIDUES

REMARK 465 THE FOLLOWING RESIDUES WERE NOT LOCATED IN THE

(省略)

REMARK 465 M RES C SSSEQI

REMARK 465 MET A 1

(省略)

REMARK 610 MISSING HETEROATOM

(省略)

REMARK 610 LMG A 751

・不明分子 (3arc.pdb では、残基名が ” UNL ” のもの) があることを確認する

HET OEX A 601 10

(省略)

HET UNL A 763 28

HET UNL A 748 16

(省略)

- ② 不明分子に関する情報をみる

リスト2 PDB HETEROGEN

REMARK 600 HETEROGEN

REMARK 600 ABOUT UNL A 763, C 745, D 723, a 763, d 723, k 745, THEY HAVE

REMARK 600 CHARACTERISTIC THE ELECTRON DENSITY MAP AS SOME KIND OF

REMARK 600 DIGLYCERIDES.

③HETNAM で非ペプチド分子の名称、FORMUL で化学構造式を調べる

リスト3 PDB HETNAM

HETNAM OEX CA-MN4-O5 CLUSTER

HETNAM FE2 FE (II) ION

HETNAM CL CHLORIDE ION

HETNAM BCT BICARBONATE ION

(省略)

④ps2 はホモ 2 量体であること、両モノマーともに 20 番目のペプチド鎖が欠落していることなどを認識しておく (Ca の Sr 置換体である 4il6.pdb には 20 番目のペプチド鎖がある)。4il6.pdb を見て、欠落ペプチド鎖は、OEC から遠方で複合体の表面部分にあることを確認しておく)。

この段階で、欠損ペプチド鎖を補修するか？または、OEC から遠方なので、欠損のままモデリングするか？などモデリングの方針をたてる (本講習会では、修復を練習をする)。

3) 欠落している原子を調べる

①futools を起動し、report_unique_res ツールで、どんな種類の残基が含まれているかを調べる。

リスト4 report_unique_res

```
executing program report_unique_res
+++ enter file name of input PDB data
c:\ps2\3arc\3arc.pdb
+++ enter 1:AA, 2:Non-AA, or 3:all kind residues
3
report residues: all kind residues
--- report-unique-res job starts at 2014/04/25 21:13:12
running ...
number of total and unique residues 8368 40
residue list [[' CA', 7], [' CL', 6], [' MG', 2], ['ALA', 464], ['ARG', 226],
(省略)
```

UNL が 38 個あるが、これらは不明分子なので同種分子ではなく別物の可能性があることに留意。

②分子フレームデータを収集する

PDB の ftp サイトから、非ペプチド分子のフレームデータ (.mht) を収集する。

FTP site of PDB: <ftp://ftp.wwpdb.org/pub/pdb/data/monomers/>

本講習会で用いるデータは事前に収集して、ps2-material¥mhtfiles フォルダに収めてある。

- ③後に fumodel で使うのに便利のように、make_filelst_file ツール (futools の tools メニューで misc を選ぶと、このツールの実行ボタンが表示される) で、これらのファイル名を記述したファイル mhtfiles.fuf を作っておく。

リスト5 make_filelst_file

```
executing program make_filelst_file
+++ enter directory name
c:\ps2\ps2-material\mhtfiles
+++ input file extension, like .pdb,...
.mht
+++ input special character for priority, '', '@', '!', '$', '%'
    ← Enter キーを入力する
+++ enter file name of filelst-file
c:\ps2\mhtfiles.fuf
--- make_filelst_file job starts at 2014/04/16 10:48:09
running ...
    (省略)
+++ enter command for make_filelst_file, 1:quit, 2:again, 3:imode
1
quit make_filelst_file
```

④欠落原子の詳細を調べる

report_missing_atoms (non-aa residue option)ツールを実行する。

リスト6 report missing atoms

```
executing program report_missing_atoms
+++ select job option. 1:non-aa residues, 2:aa-residues.
1
+++ enter file name of pdb file(.pdb)
c:\ps2\3arc\3arc.pdb
+++ input mht file(s). i.e. "file1,.." or "< file-1st file" (do not type
quotes)
< c:\ps2\mhtfiles.fuf
+++ do you want to output in log file? 1:yes, 2:no
2
missing atoms: resnam, number of missing atoms and atmnam list (heavy atom
only)
(省略)
BCR:649:B 0
SQD:768:D, 11, [' C21', ' C20', ' C22', ' C18', ' C19', ' C14', ' C15',
' C16', ' C17', ' C12', ' C13']
(省略)
```

SQD:768:D(残基名：残基番号：鎖名)に11個の原子の欠落がある。ここでは、SDQ:667:aのように、欠損がない残基もプリントされる。これは、欠損がある残基を修復する際の参考にするためである（後の修復作業では、完全なものから欠損原子をコピーする）。

⑤再度、report_missing_atoms(aa-residues option)ツールを実行する。

幸い、ポリペプチド分子には欠損原子がない（残基全体の欠損はあるが、全てN末端とC末端近辺である）

4. ps2 モノマー” 安直モデル” の作成

以降では、モノマーの構造モデル (chain 名が大文字のものの集合、モノマー a とする) を作成する。モノマー b も同様にして作成し、最後に、統合すれば 2 量体のモデルが完成する。

モデリングに入る前に、各部の呼称およびデータ保存のディレクトリ名とファイル名などいくつかのルールを、附録(A, B, C, D)のように決めておく。

1) モノマーの PDB ファイルを作る

①split_pdb ツールの chain option で、3arc.pdb を chain ごとの PDB ファイルに分割する。

リスト7 split_pdbの実行

```
executing program split_pdb
+++ select split option. 1: at TER, 2:chains, 3: pro,chm,wat
2
split option: chains
+++ enter file name of input pdb file
c:\ps2\3arc\3arc.pdb
+++ enter directory name of output PDB files
c:\ps2\3arc\chain
directory not found. directory name= c:\ps2\3arc\chain
+++ do you want to re-enter or create? 1:re-enter, 2:create, 3:quit the job
2
+++ do you want to output in log file? 1:yes, 2:no
2
base name of created pdb files= 3arc
(省略)
--- split_pdb job ends at 2014/04/15 10:01:50, elapsed time: 1 sec.
```

②3arc-chain.sp1 ファイルをエディタで編集して、小文字の chain 名のものを削除して、c:\ps2\3arca\3arca.mrg 名で保存する。

リスト8 3arc-chain.spl ファイルの編集

```
# created by split_pdb at 2014/04/15 10:01:50
# parent pdb file:c:\ps2\3arca\3arca.pdb
c:\ps2\3arca\chain\3arc-uA.pdb #A -3467
c:\ps2\3arca\chain\3arc-uB.pdb #B -5786
省略
c:\ps2\3arca\chain\3arc-uZ.pdb #Z -534
c:\ps2\3arca\chain\3arc-la.pdb #a -3409
c:\ps2\3arca\chain\3arc-lb.pdb #b -5666
省略
c:\ps2\3arca\chain\3arc-lz.pdb #z -519
```

小文字の chain 名の行をすべて削除。
ファイル名
'c:\ps2\3arca\3arca.mrg' 名で保存する。

③merge_pdb ツールで、3arca.mrg ファイルを入力して、モノマー a の PDB ファイル c:\ps2\3arca\3arca.pdb を作成する。

リスト9 merge_pdbの実行

```
executing program merge_pdb
+++ input pdb files to be merged. i.e. "file1,.." or "< file-1st file" (do not type quotes)
< c:\ps2\3arca\3arca.mrg
+++ enter file name of output pdb file(full path)
c:\ps2\3arca\3arca.pdb
--- merge_pdb job starts at 2014/04/25 21:26:13
running ...
total number of atoms 27140
--- merge_pdb job ends at 2014/04/25 21:26:14, elapsed time: 1 sec.
```

ここで、構造を見たければ、fumodel で見ればよい(fumodel 32 ビット版では、表示できる最大原子数は5万程度である。ただし、描画が大変遅くなるので、この状態でモデリング作業を行うのは実用的ではない。

2) モノマーをポリペプチド、非ペプチドと結晶水に分割する

split_pdb ツール(pro, chm, wat option, no environment)で、3arca.pdb を、ポリペプチド (3arca-pro.pdb)、非ペプチド (3arca-chm.pdb) と結晶水部分 (3arca-wat.pdb) に分割する。分割ファイルを格納するフォルダ名は、c:\ps2\3arca\pro-chm-wat とする。

リスト10 split_pdb

```
executing program split_pdb
+++ select split option. 1: at TER, 2:chains, 3: pro, chm, wat
3
split option: into pro, chm, wat
+++ enter file name of input pdb file
c:\ps2\3arca\3arca.pdb
+++ enter directory name of output PDB files
c:\ps2\3arca\pro-chm-wat
directory not found. directory name= c:\ps2\3arca\pro-chm-wat
+++ do you want to re-enter or create? 1:re-enter, 2:create, 3:quit the job
2
+++ do you want to output in log file? 1:yes, 2:no
2
(省略)
split-files1st          file                      is          created
c:\ps1\3arca\pro\chm\wat\3arca-pro-chm-wat.spl
number of created split files= 3
```

3) pro に水素原子を付加する

- ①3arca-pro.pdb を split-pdb(chain オプション) ツールで chain に分割する。分割ファイルを格納するフォルダ名は、c:/ps2/3arca/pro-chm-wat/pro-chain とする。

リスト11 split_pdb

```
executing program split_pdb
+++ select split option. 1: at TER, 2:chains, 3: pro, chm, wat
2
split option: chains
+++ enter file name of input pdb file
c:\ps2\3arca\pro-chm-wat\3arca-pro.pdb
+++ enter directory name of output PDB files
c:\ps2\3arca\pro-chm-wat\pro-chain
(省略)
split-fileslst          file                      is          created
c:\ps2\3arca\pro-chm-wat\pro-chain\3arca-pro-chain.spl
number of created split files= 19
--- split_pdb job ends at 2014/04/26 17:04:01, elapsed time: 3 sec.
```

+++ enter command... で '3' を入力、続いて、'5' を入力する。

リスト12 futoolsからfumodelを起動する

```
+++ enter command for split_pdb, 1:quit, 2:again, 3:imode
3
+++ >>> 1:quit, 2:again, 3:view input pdb, 4:view output pdb, 5:view pdb(&frg)
in directory.
5
opening fumodel ...
(省略)
```

fumodel が起動される。起動時に、分割された全ての PDB ファイルが読み込まれている (chain は全部で 19 本)。右上にある Molecule 窓で分子を選択またはその右にある ' > ' ボタンで順次分子を選択して、Add-H and bond to AA residue メニュー (パネルのオプションは、全て Default にする) で水素をつけ、それぞれを元の PDB ファイルのファイル名の最後に @ をつけて保存する (たとえば、3arca-pro-uA.pdb → 3arca-pro-uA@.pdb)。fumodel を close しないと futools に制御が戻らないので注意すること。

(補足) ファイルのベース名の最後に @ を付けるのは、後で make_files_file ツールを使って、ディレクトリ内の特定のファイル群を取り出すためである。

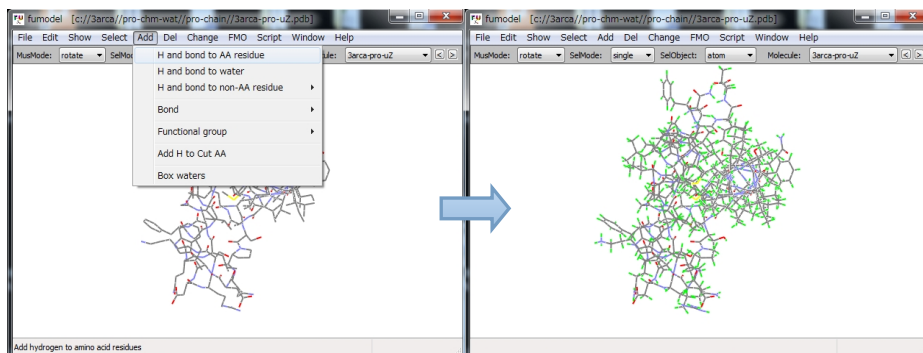


図1 ポリペプチドへの水素原子付加

②全てのポリペプチド鎖に水素を付け終われば、fumodel を close する。

(補足) 全部で 19 本のポリペプチドに手動で水素をつけるのは面倒である。fumodel には、script 実行機能が備わっているのをこれを活用する。c:\ps2\fu-script に収めた script は本講習会のために作った。ここで、PyCrust コンソールから、

```
>>> fum.ExecuteScript1('c://ps2//fu-script//add_h_to_pro.py')
```

と入力して、add_h_to_pro.py スクリプトを実行すると、19 本のポリペプチドに次々と水素をつけて、その PDB ファイルが保存される。

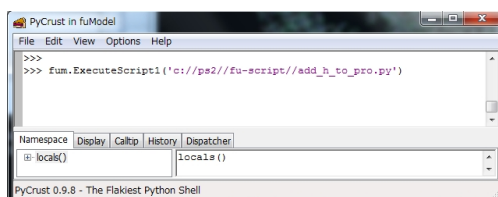


図2 PyCrust コンソールの入力

(注意1) ファイル名のディレクトリの区切りは” // ”を使う（¥は駄目）。

(注意2) ここで提供する script は全て、本講習会で一時的に使うものであるため、プログラム中でファイル名を明示的に書いてある。従って、本講習会のデータ保存のルール（ディレクトリ名やファイル名のルール）に従っていないとファイルが無いというエラーが起きるので注意（その場合は、先頭部分に書いてあるファイル名を変更すればよい）。

(注意3) fumodel（このインスタンスが fum）は、内部で script を実行する method を2つ持っている（ExecuteScript と ExecuteScript1）。前者は汎用な script を Script メニューに登録して使うと便利で、後者は一時的な script を実行するのに便利である。

③split_pdb ツールを終了し、make_filelst_file を実行して、次に merge_pdb ツールで、ベース名の最後尾が@のファイルのファイルリスト ファイル c:\ps2\3arca\pro-chm-wat\3arca-pro@.mrg を作る。

リスト13 make_filelst_file

```
executing program make_filelst_file
+++ enter Directory name
c:\ps2\3arca\pro-chm-wat\pro-chain
+++ input file extension, i.e. .pdb,...
.pdb
+++ input priority character at the end of base name, '', '@', '!', '$', '%'
@
+++ enter file name of filelst-file
c:\ps2\3arca\pro-chm-wat\3arca-pro@mrg
--- make_filelst_file job starts at 2014/04/26 17:14:06
running ...
searched files with extension, .pdb, in directory
c:\ps2\3arca\pro-chm-wat\pro-chain
number files found: 38
number of picked up files with special character, @ = 19
c:\ps2\3arca\pro-chm-wat\3arca-pro@mrg file is created.
number of files: 19
--- make_filelst_file job ends at 2014/04/26 17:14:06, elapsed time: 0 sec.
```

④merge_pdb ツールを実行して、pro を再構成する (3arca-pro@.pdb を作成する)。

リスト14 merge_pdb

```
executing program merge_pdb
+++ input pdb files to be merged. i.e. "file1,.." or "< file-lst file" (do
not type quotes)
< c:\ps2\3arca\pro-chm-wat\3arca-pro@mrg
+++ enter file name of output pdb file(full path)
c:\ps2\3arca\pro-chm-wat\3arca-pro@.pdb
--- merge_pdb job starts at 2014/04/26 17:15:49
running ...
total number of atoms 41116
--- merge_pdb job ends at 2014/04/26 17:15:51, elapsed time: 2 sec.
```

⑤fumodel で 3arca-pro@.pdb を読み込んで、Select-Non-bonded atom メニューで、結合
ができていない原子をチェックする(異常構造の簡単なチェックになる)。

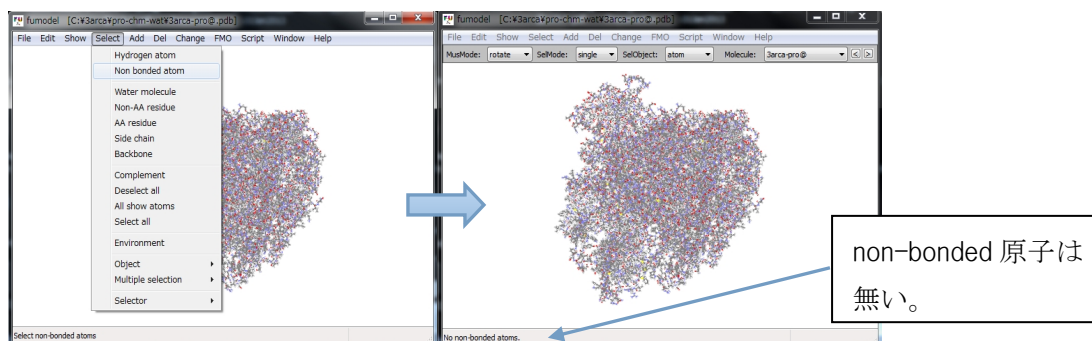


図3 非結合原子をチェック

(補足) 3arca-pro.pdb はポリペプチドのみからなるので、全体に対して水素原子を付加すれば、一度ですむので効率的である。しかし、この系では一部の残基 (LEU:40:J, LEU46:Y, ALG:45:F, LEU:352:D, LYS:84:E) で水素付加に失敗するので、後で手動での手直しが必要になる。それでも、全部を一度に処理したければ、fumodel で、3arca-pro.pdb (c:/ps2/3arca/pro-chm-wat フォルダ内にできている) を読み込んで、Add-H and bond to AA residue メニューを実行する。

(注意) fumodel 起動時の設定ファイル fumodelset.py

で, fum.usefmoutil=True(default) とすると Fortran モジュールが使われるが、False の場合は python のコードを使うので、とてもとても遅くなる (30 分かかる?)

4) chm に水素原子を付加する

①fumodel の File-Open メニューで、3arca-chm.pdb と mhtfiles.fuf を読む

(注意) fu-23Dec2013 版では、この機能は使えない。mht file をひとつずつ読まないといけない。

②Add-H and bond to non-AA residue-Use frame data メニューで水素をつける

(少し時間がかかるので待つ)

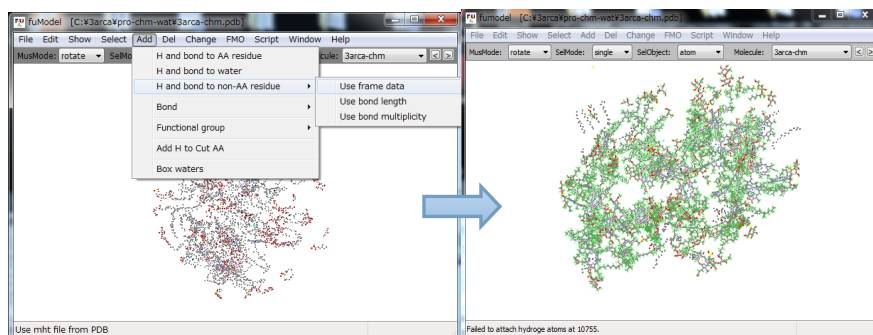


図4 非ペプチド分子への水素付加

③正しく付いたかどうかを目視で確認する (SelObject-residue にして L-Click、または tree selector で対象 residue を選択状態にして、Show-Selected only, ‘f’ キーの押下を繰り返す)。

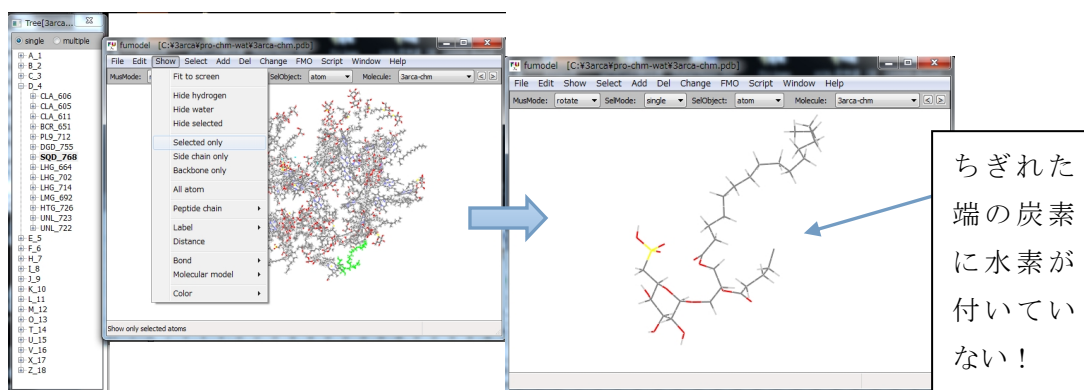


図5 水素が正しく付加されているかチェック

欠落原子がある残基には正しく付かないことが多いので、特に注意深く目視検査する (ちぎれた末端の原子に注目)。先に調べたように、欠落原子がある残基は以下のものである。HTG:726:D, DGD:660:C, DGD:657:C, LMG:751:A, SQD:768:D, LMG:784:Z, DGD:755:D, LHG:772:E, LMG:729:C, LMG:776:C, LMG:669:B, DGD:661:C, LMG:692:D, DGD:663:H

水素の未付加や過付加などがあれば、その原子を L-Click して選択状態にして、Add-functional group や Del-Selected (または Delete キーを押下) で水素を付加/削除する。DGD と LMG は水素未付加の末端炭素が 2 個ある。

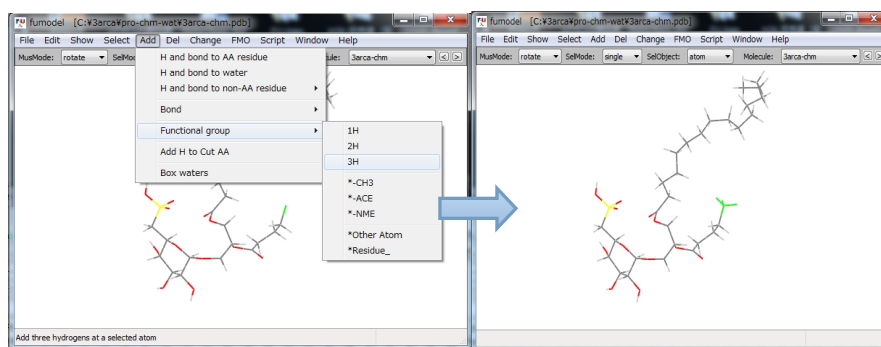


図6 原子を選択状態にして水素を追加する

④不明リガンド (残基名 UNL) は、分子フレームデータ (.mht ファイル) がないので水素が付かないので、手動で付ける。Select-Selector-Name/Number パネルの Residue 窓で UNL を選択する。

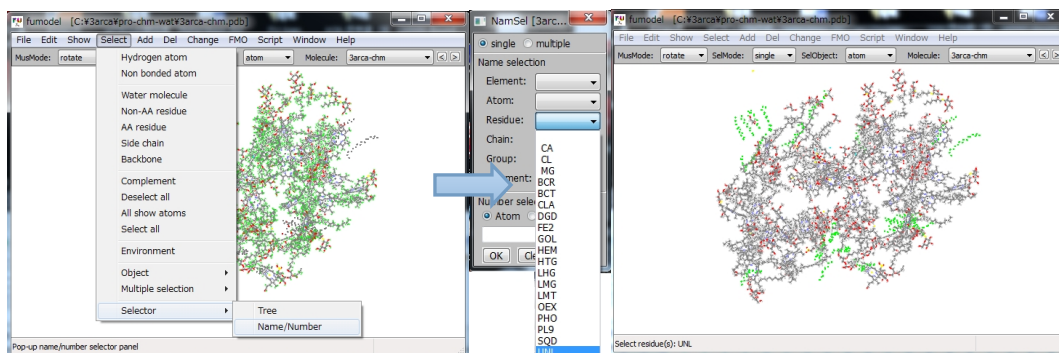


図7 不明リガンド (残基名 : UNL)を選択状態にする

⑤Add-H and bond to non-aa residue-Use bond lengthメニューで水素をつける。

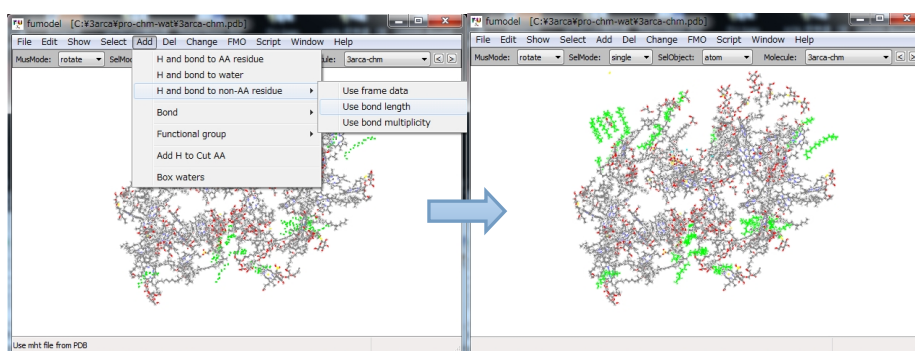


図8 不明リガンドに水素を付加する

正しく付いたかどうかを目視で確認する。水素が正しく付いていない場合、③同様に手動で付加/削除する。

⑥全ての処理を終えたら、3arca-chm@.pdb で保存する。

(補足) 実は、この系では、分子フレームデータを使って水素をつけた後の付けの残し(原子欠落残基の末端原子と未知リガンドを含めて)は全て自動的に正しく水素付加ができるので、上記の面倒な操作は不要である(本講習会では安心な操作を説明している)。すなわち、Add-H and bond to non-AA residue-Use frame dataメニューの実行後に、Add-H and bond to non-AA residue-Use bond lengthメニューを実行すればよい。一般的なことであるが、自動処理を行った場合、正しく処理されたかどうかを(目視で?)確認すべきである。

3arca-chm.pdb に水素をつけて 3arca-chm@.pdb で保存する script を作成してある。PyCrust コンソールから、次のコマンドを入力すればよい。

```
>>> fum.ExecuteScript1( 'c://fu-script//add_h_to_chm.py' )
```

5) wat に水素原子を付加する

①fumodel に 3arca-wat.pdb を読み込む。

②Add-H and bond to water で水素を付加する (H の方向はランダムにつけられる)。

③3arca-wat@.pdb で保存する。

(補足) ここで、H の位置を最適化すると、水分子間のみに水素結合が作られるのでよくない。

6) pro、chm、wat を統合してモノマーを再構成する (3arca@.pdb を作成する)

①3arca@.mrg ファイルをつくる (エディタまたは make_files_file ツール)。

②merge_pdb ツールで、統合して、c:/ps2/3arca/3arca@.pdb で保存する。

(補足) fumodel の File-Merge メニューで順次これらの PDB ファイルを読み込んでもよい。ただし、表示に時間がかかるのでいらいらする (GUI が邪魔になる例である)。

7) ヒスチジン (HIS) の δ または ϵ 位のプロトンを削除して、”安直モデル” が完成

ポリペプチドに水素を付加した際に、HIS 残基は全て (54 個ある) プロトン化した。ここで、全部の HIS 残基を目視して、 δ または ϵ 位のプロトンのどちらかを削除する (プロトン化されたままでよいものはそのままにする)。この判断は、水素結合が形成できる方を残す方針で行う (金属に配位している HIS の場合は削除すべきプロトンは自明)。

① pick_up_obj ツールで、3arca@.pdb 中の HIS を環境付きでとり出す。

リスト15 pick up obj

```
eexecuting program pick_up_obj
+++ input object name, i.e. "ALA", "chm", "pro"(do not type quotes).
HIS
+++ with environment? 1:no, 2: yes, 3:quit
2
+++ input cut-off distance(in A) of environment.
4
+++ enter file name of input pdb file
c:\ps2\3arca\3arca@.pdb
+++ enter directory name of output PDB files
c:\ps2\3arca\3arca@\his-e40
directory not found. directory name= c:\ps2\3arca\3arca@\his-e40
+++ do you want to re-enter or create? 1:re-enter, 2:create, 3:quit the job
2
--- pick_up_obj job starts at 2014/04/26 17:47:58
    (省略)
number of atoms in picked-up residue= 259
number of created PDB files= 54
split          fileslst          file          is          created
c:\ps2\3arca\3arca@\his-e40\3arca@-pick-up-HIS-e40.spl
--- pick_up_obj job ends at 2014/04/26 17:48:17, elapsed time: 18 sec.
+++ enter command for pick_up_obj, 1:quit, 2:again, 3:imode
3
+++ >>> 1:quit, 2:again, 3:view input pdb, 4:view output pdb, 5:view pdb(&frg)
in directory.
5
    (省略)
```

fumodel が起動され、54 個の HIS 残基（4Å以内の環境残基を含む）の PDB ファイルが読み込まれる。

② **Molecule** 窓またはその右の'<'または'>'で、カレント分子を順次選んで、**HIS** の不要なプロトンを削除する（不要なプロトンを選択して”**Delete**”キーを押す、または、**Del-Selected** メニューを使う）。修正した構造を **Save as** メニューで元の PDB ファイルのベース名の最後に@を付けた名前で保存する。

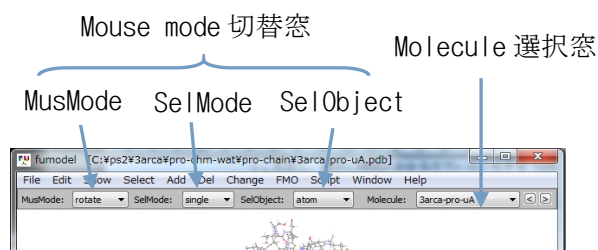


図9 分子の選択とマウスモードの切替窓

例として、**3arca@-HIS9-uB-e40.pdb** の処理を以下に示す。

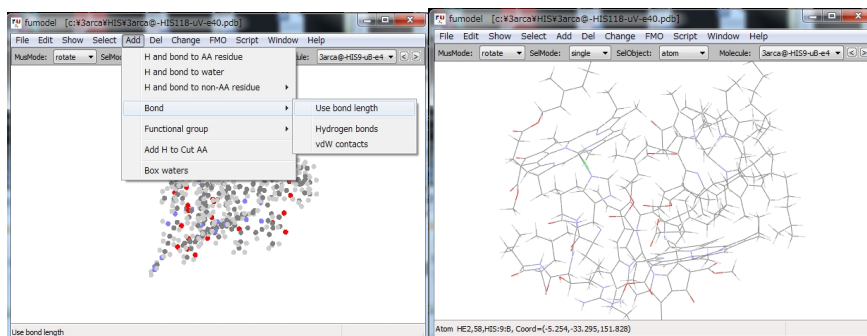


図10 Molecule 窓で分子を選んで、Add-Bond メニューで結合を付ける

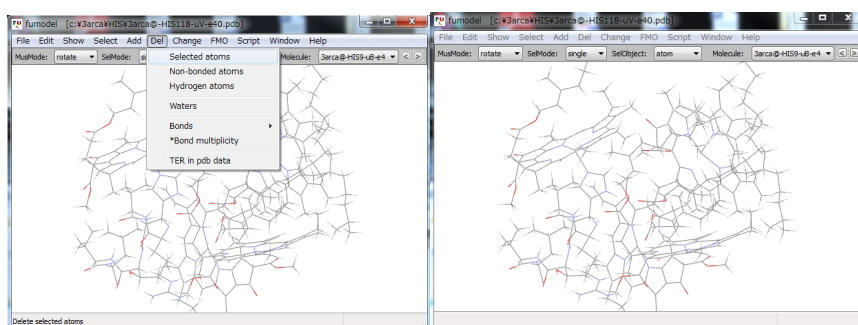


図11 MG に近い（結合が作られているものもある）H を選択して、削除する。

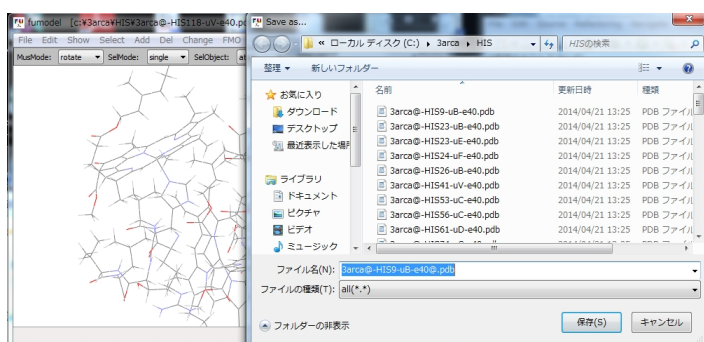


図 12 PDB ファイルを保存する

- ③全ての HIS について処理が終われば、make_filelst_file ツールで
 c:/ps2/3arca/3arca@/his-e40 フォルダ内の PDB ファイル (base 名の最後が@のもの) のファイルリストファイル
 (c:/ps2/3arca/3arca@/his-e40/3arca@-replace-his.rpl 名) を作成する。
- ④ replace_res ツールで、c:/ps2/3arca/3arca@.pdb 中の HIS を置き換えて、
 c:/ps2/3arca/3arca@@.pdb を作る。

リスト16 replace_res

```
executing program replace_res
+++ enter file name of input target pdb file
c:\ps2\3arca\3arca@.pdb
+++ input residue pdb file(s)
< c:\ps2\3arca\3arca@\his-e40\3arca@-replace-his.rpl
+++ enter file name of output pdb files
c:\ps2\3arca\3arca@@.pdb
--- replace_res job starts at 2014/04/26 18:52:55
running ...
resnam,resnmb,chain in ReplaceRes HIS 100 B
(省略)
note: connect data were recreated based on distance.
--- replace_res job ends at 2014/04/26 18:53:04, elapsed time: 9 sec.
```

- ⑤fumodel で 3arca@@.pdb を読み込んで、Select-Selectore-Name/Number メニューを実行して、residue 窓で HIS を選ぶ。Show-Selected only で HIS だけ表示させて、すべての HIS の処理を終えたかどうかを目視で確認する。

これで、モノマー a の” 安直モデル” (c:/ps2/3arca/3arca@@.pdb) が完成した。

(補足) 54 個もの HIS の構造をいちいち上記のやり方で操作するのは苦痛である。ある程度、目視した後で、削除するプロトンについての規則性が分かれば script を作って一括処理をすればよい。はっきりしているのは、金属イオン (CLA の MG など) に配位している HIS である。これを一括で処理すると残る手作業は 18 個になる (これでも多すぎてうんざりする。うんざりするとミスをする)。本講習会用に、金属に配位した HIS を処理するスクリプト `c:\¥3arca¥fu-script¥his_delete_h.py` を用意した。使い方は、fumodel を立ち上げて (何も読み込まなくてよい)、PyCrust コンソールから、

```
>>> fum.ExecuteScript1( 'c://ps2//fu-script//his_delete_h.py' )
```

と入力する。これを実行した後、残る 18 個 (この script が 3arca@-not-done-his.spl ファイルを作るので、これを fumodel で読み込んで) を手動でプロトンを削除し、ベース名の最後の@をつけて保存する。そして③、④を実行する。

8) 溶媒和クラスターモデルの作成(おまけ)

- ①fumodel に 3arca@@.pdb を読み込む。
- ②Add-box water で溶媒水分子を付加する (32bit 版の FU では、メモリ不足になるので、ここでは薄い shell (3.5 Å) をつける。
- ③ 3arca@@-w35.pdb で保存する

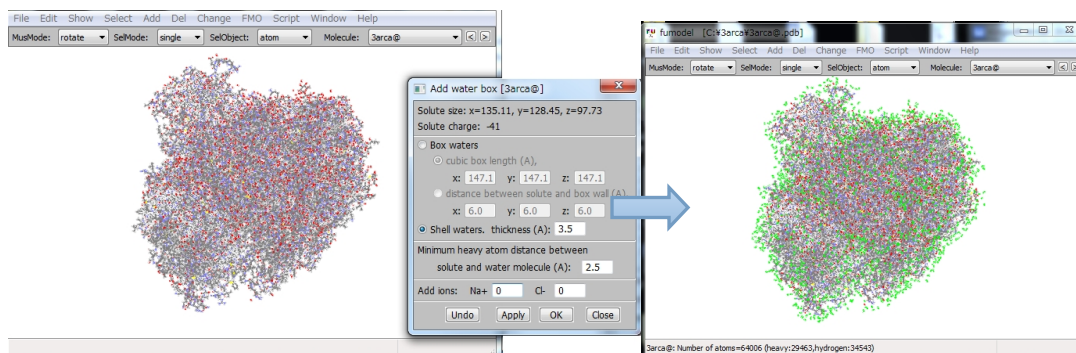


図 13 溶媒水を加えて水和クラスターモデルを作る

(補足 1) ここで配置した水分子はバルク水の 300K で MD を行った結果のトラジェジェクトリ (FU のデータファイル¥data¥water-box.pdb 参照) をつぎはぎして作られる。従って、Amber などの力場で軽く最適化または MD を行うなどして溶媒分子の配置を決めなければならない。力場計算を行うには、非ペプチド分子に力場パラメータをアサインしなければならないが、FU は力場パラメータのアサイン機能が未実装なので (TINKER を使用しているのでポリペプチドのみであれば容易にできるが、非ポリペプチド分子が含まれると削除される)、別途、AMBER や GROMACS などの力場プログラムで行って欲しい。

(補足 2) 3arc.pdb の構造を見ると OEC 近辺は結晶水に囲まれていて、溶媒の水分子は接近できないように見える。これが本当なら、溶媒は PCM など連続誘電体モデルで扱うのもよいだろう。

(注意) ps2 はチラコイド膜に埋まっているので、ここで作った水和クラスターはモデルとして適切ではない。チラコイド膜を付加したモデルの作成は、次回以降で挑戦したい。

9) 作成した構造データの品質検査

特に、missing atoms がある残基、不明残基は注意深く検査すること。

①大雑把な検査

- i) check_geom ツールによる、結合距離、結合角の検査
3arca-chm@@.pdb を検査する。

リスト17 check_geomの実行

```
executing program check_geom
! this program requires pdb file with connect data.
+++ select check option. 1:bond length, 2:angle, 3:vdW contact, 4:quit.
1
threshold length should be give by factors:fmin and fmax for
r < fmin*r0, r> fmax*r0 (r0:standard single bond length).
+++ input fmin and fmax, like "0.8 1.2" (" is not input data).

find bond whose distance is shorter than 0.8 or longer than 1.2 times of
standard value
+++ enter file name of pdb file(.pdb)
c:\ps2\3arca\3arca@@.pdb
+++ do you want to output in log file? 1:yes, 2:no
1
job may take several minutes for 10,000 atoms.
--- check_geom job starts at 2014/04/15 16:47:07
running ...
total number of bond length parameters= 11704
number of wrong length bonds= 55
  number rij  i-atom  j-atom  i-atmnam  j-atmnam  i-resnam  j-resnam
      1  1.205    725    743   C11   C12 BCR:A BCR:A
      2  1.199    748    750   C15   C16 BCR:A BCR:A
      (省略)
     55  1.202  10960  10962   C20   C21 BCR:T BCR:T
  (省略)
```

短いC-C結合が多数リストされるが、全てBCR（共役系）のものであり、2重結合にしては短すぎるが我慢できる範囲である（この判断はいい加減）。angle optionで90度以下の結合角を書き出すと、大部分はOECやCLAのもので問題はない。ただしBCR:651:DのC29-C30-C39の結合角が88.7度であるのは異常である。

ii) 原子欠落残基の目視検査

fumodelで実施。

iii) 不明リガンド (UNL)の目視検査

fumodel で実施。

iv) fumodel の Add-Bond-Hydrogen bond メニューを実行して、水素結合ネットワークの妥当性を見る（妥当性の判断は難しいが）。

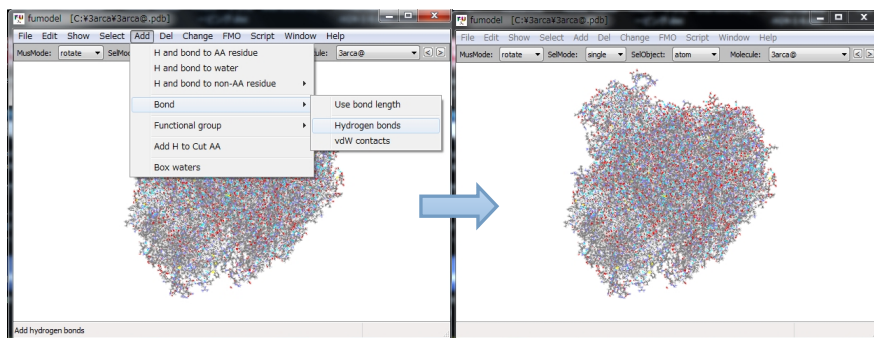


図 14 水素結合を見る

②丁寧な検査

各残基をひとつずつ MO (ab initio または semi-empirical または FMO) 計算して、収束を見る。FMO 法の PIE (ペア相互作用エネルギー) を見れば、重要な水素結合が形成されているかどうかチェックできる。FMO-RHF/STO-3G 計算が手軽なので、これを推奨する。

ここでは、例として、BCR:651:D で、AM1 計算によって検査を行う。

i) c:/ps2/3arca/pro-chm-wat フォルダ下で extract フォルダを作って、問題のある残基をここに抽出する。

ii) extract_res ツールを実行して、3arca@@.pdb から BCR:651:D を抽出する。

リスト18 `extract_res`の実行

```
executing program extract_res
+++ input residue data in full form, like "ALA:1:A"(do not type quotes)
BCR:651:D
+++ with environment? 1:no, 2:yes, 3:quit
1
+++ enter file name of input pdb file
c:\ps2\3arca\3arca@@.pdb
+++ enter file name of output pdb files
c:\ps2\3arca\3arca@\extract\BCR651-uD.pdb
--- extract_res job starts at 2014/04/26 19:03:07
running ...
extracted residues: ['BCR:651:D']
number of atoms in extracted residue= 96
--- extract_res job ends at 2014/04/26 19:03:08, elapsed time: 1 sec.
```

iii) 抽出した残基の PDB ファイルを `fumodel` で読み込む。抽出の際に結合情報が落ちてしまうので、`Add-Bond-Usebond length` でつける (MO 計算する場合、結合情報は不要であるがつけると構造が見やすい)。ついでに、結合角が異常であった **C29-C30-C39** を近辺の構造を確認する。

iv) Script メニューから `GAMESS assist.py` を実行する。

`GAMESS assistant` パネルで `Run type` を `energy` にして入力データを作成する。AM1 の入力データの作成はサポートされていないので、できた入力ファイルを編集して、AM1 計算用に修正する)。Make input ボタンを押したあとで、Edit/View メニューで `GAMESS input file` を編集する。input file の不要部分を削除し、`$basis GBASIS=AM1 $end` を加える。また、必要な場合、`$contrl` グループに電荷 (`ichrag=`) やスピン多重度 (`mult=`) を加える。編集後、上書き保存で保存する (ここでファイル名を変更すると `fumodel` が見失ってしまう)。

リスト 19 GAMESS の AM1 計算の入力データ

! Created by fu at 2014/04/17 07:55:07

\$contrl runtyp=energy ispher=1 nprint=-5 \$end

\$system mwords=115 \$end

\$scf dirsdf=.t. npunch=0 \$end

\$basis GBASIS=AM1 \$end

\$data

SQD768uD-e0-o

C1

1	6	9.663000	-80.917000	178.790000
2	1	10.094000	-79.890000	178.929000

(省略)

v) RunGAMESS ボタンを押して AM1 計算を実行する。

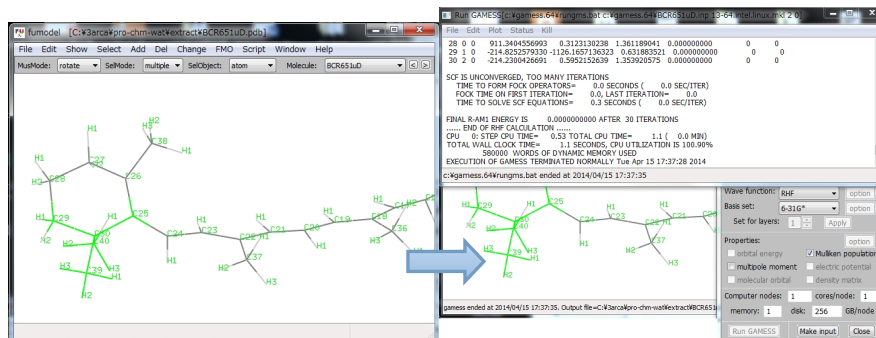


図 15 AM1 の 1 点計算

この計算は 30 回の iteration では収束しなかった。この構造のままで、電子状態計算を行うのは危ない。

(補足 1) 多数の残基の検査を行う場合、extract_res でいちいち残基の PDB ファイルを作るのは面倒である。先に split_pdb で分割しておいて、これらを、python script を作って一括処理する方法を 7-2) で説明する。

(補足 2) イオンの電荷 (正負の整数) を設定するには、fumodel で原子を選択しておいて、Change-Charge メニューを実行する。アミノ酸残基の電荷 (非整数電荷) の設定は、FMO-Fragment tools-Charge assign メニューで行う。後者は PDB ファイルには保存できない。非整数電荷データを保存するには、'.xyz' ファイルで保存する。

構造の品質検査の結果から”安直モデル”は問題がありそうであるが、一応、これを本講習会のお土産とする。

5. ps2 モノマー “修復モデル” の作成

ps2 に含まれるポリペプチドの構造に問題は見つからなかったが、通常の手順として、力場で軽く構造最適化を行う。

また、原子が欠落している非ペプチド分子を修復して、部分/全体の構造最適化を行う。非ペプチド分子は、AM1、または ab initio、または FMO を用いる。本講習会では、計算時間の関係で、AM1 を用いる。

1) ポリペプチドを軽く構造最適化する

ポリペプチド鎖全体を一度に最適化するのが効率的であるが計算時間がかかるので、ここでは例として、小さなポリペプチド鎖（すでに、c:/ps2/3arca/pro-chm-wat/pro-chain フォルダにある）3arca-pro@-uF.pdb について行う。

①fumodel でこの PDB ファイルを読む。

②Script メニューから tinker_optimize.py を実行する。

③TINKER Optimization パネルで、RMS gradient convergence を 5 kcal/mol/Å にして Exec ボタンを押す（部分構造最適化を行う場合は、最適化する原子を選択状態にする）。

（注意）fumodel の PyCrust ウィンドウを閉じてしまうと、TINKER が実行できない。

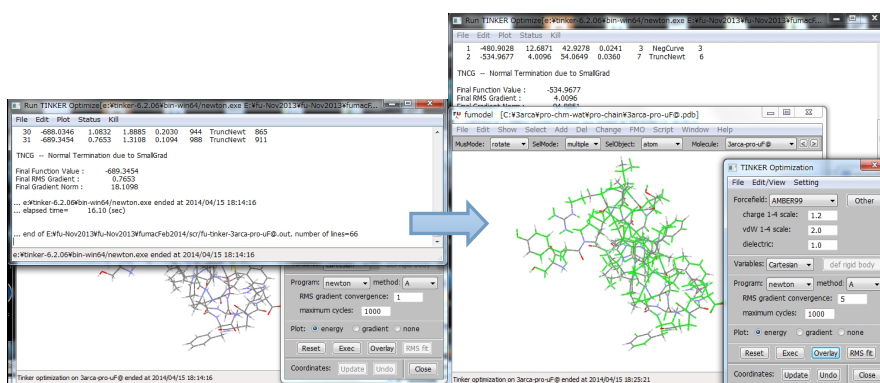


図 16 TINKER によるポリペプチド分子の構造最適化

④最適化計算が終了したら、Overlay ボタンを押して元の構造と比較し、良ければ Update ボタンを押す。これを、File-Save as メニューで c:\yps2\3arca\pro-chm-wat\extract\3arca-pro@-uF-o.pdb と名づけて保存する。

⑤軽く最適化した残基を、replace_res ツールで、3arca@@.pdb のものと置換する。

リスト20 replace_resの実行

```
executing program replace_res
+++ enter file name of input target pdb file
c:\ps2\3arca\3arca@@.pdb
+++ input residue pdb file(s)
c:\ps2\3arca\pro-chm-wat\extract\3arca-pro-uF@-o.pdb
+++ enter file name of output pdb files
c:\ps2\3arca\3arca-pro@@-o.pdb
--- replace_res job starts at 2014/04/15 18:30:15
(省略)
+++ enter command for replace_res, 1:quit, 2:again, 3:imode
1
quit replace_res
```

2) 欠落原子がある残基の修復と部分構造最適化

例として、欠落原子がある SQD:768:D に、完全な SQD:668:B から欠落原子を transfer する（このやり方は、UNL の分子名が分かった後の修復、欠落ポリペプチド鎖を他のタンパク質から transfer する、8. で述べる OEC クラスターの置換など、一部重なりがある reference 系があればまったく同じようにできる）。

- ①extract_res ツールで、3arca@@.pdb から SQD:768:D と SQD:668:B を抜き出して、extract フォルダに入れる（修復は水素をつける前の構造、または、読み込んだあと一旦水素を削除して行い、修復後に再度水素をつける）。
- ②fumodel を起動して、target である SQD768D.pdb を読み込む（ここで Add-Bond-Use bond length メニューで結合を作ると見やすい）。
- ③Script メニューから transfer_atoms.py を実行する。
- ④transfer_atoms.py パネルの、Read ボタンを押して、SQD668uB.pdb を読み込む。読み込んだあとで、'f' キーを押す。画面上で離れていて見難いので、MusMod を translate にして、CTRL キーを押しながらマウスを R-Drag して、target 分子の近くまで持ってくる。
(補足) fumodel では、CTRL+R-Drag すると座標パラメータそのものが変更される。
- ⑤target 分子の C7, C8, C9, C10, C11 をマウスで L-Click して選択状態にする（SelMod を multi、SelObject を atom にしておくこと）。reference 分子の方は、全原子を選択状態にする（selObject を group にしてどれか原子を L-Click する）。

- ⑥ここで、Transfer_atoms パネルの matching パラメータを atom name だけにして(resnam と resnmb のチェックを外す)、Apply ボタンを押す。
- ⑦ついで、RMS fit の Do ボタンを押す。
- ⑧reference から target に原子が移動したら、Select atom の Apply ボタンを押す。
- ⑨Reference molecule の Del ボタンを押して、reference を消去する。

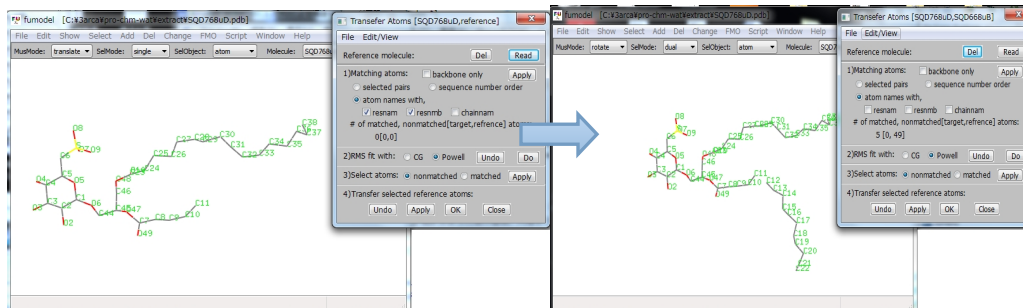


図 17 参照分子から原子を移入する

- ⑩C11 と C12 の距離が 2.21Å (Show-Distance メニューをオンにして、これらの原子を L-Click すると距離が表示される) と長くなってしまった。C12 から C22 を選択状態にして、CTRL を押しながらマウスを R-Drag して距離が 1.54Å 程度になるように選択原子集団を移動させる(MusMode を translate にしてから操作する)。

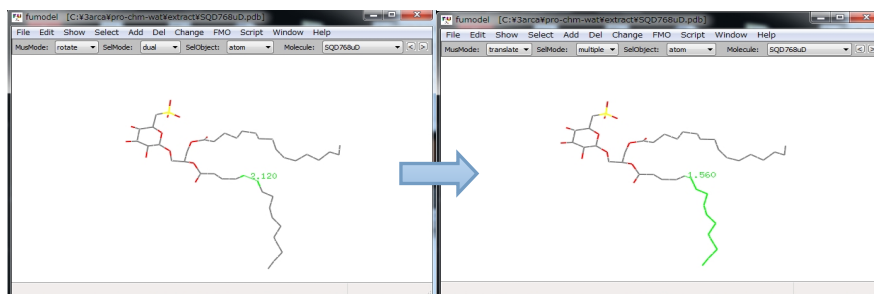


図 18 移入した原子集団を移動させて適切な位置に配置する

- ⑪SQD の mht file (SQD.mht) を読み込んで、水素をつける。
- ⑫Script メニューから、GAMESS-assist.py を起動し、Runtype を optimize にして、Make input ボタンを押す。
- ⑬Edit/View メニューで input data を AM1 用に編集する。\$statpt に opttol=0.01 を加え上書き保存する。
- ⑭RunGAMESS ボタンを押して実行する。

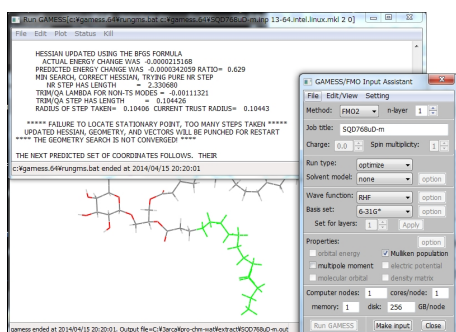


図 19 AM1 計算による構造最適化

⑮構造は収束していないが、軽い最適化を行ったと考えて、この構造を採用する。

Change-Update coordinate from file メニューを実行する。

⑯File-Save as で c:\yps2\3arca\pro-chm-wat\extract\YSQD768D-o.pdb 名で保存する

⑰replace_res ツールで、3arca@@.pdb 中の元の残基と置換する(6.④と同じ)

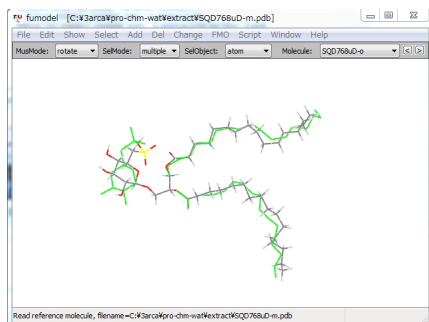


図 20 元の構造(緑)と最適化後の構造の重ね合わせ

(補足 1) 初期構造を丁寧に作る場合は、fumodel の Change-Conformation メニューの Z-Matrix editor を使う。これを用いると結合距離、結合角、2 面角を精密に設定できる。

(補足 2) 最適化構造と元の構造を比較するには、File-Merge で元の構造を読み込む(単に重ねて描画するだけで rms fit ではない)、または、Script メニューから rms_fit.py を実行する(使い方は transfer-atoms.py とほとんど同じ)。

本格的な修復を行う場合には、他の ps2 や関連の実験構造データを十分に調査して、最適なものを参照する。

6. ps2 モノマー “最適化モデル” の作成

残基を孤立状態で(部分)構造最適化するのはすでに述べた。ここでまでで、FU によるモデリングのほとんどの技術の紹介が終わっている。ここでは、部分の構造最適化の際に、周辺の残基を含めたモデル系を作り、構造最適化を行う技術を紹介する。系のサイズが大きくなるので、構造最適化に使う手法に限られる（あるいは計算時間がかかるのを我慢する）。ポリペプチドは、まず、力場を第一候補とし、非ペプチド分子は、AM1 (AM1+LJ 法が欲しいところである) や FMO 法を用いるのが実用的と考えられる。

1) 環境中のポリペプチドの構造最適化

先にやった小さなポリペプチド (3arca-pro-uF@.pdb) を再び例に説明する。

- ①extract_res (with environment option) ツールで、
residue 名を '*:*:F' (‘はいらない’) と入力する。
- ②fumodel File-Open で読み込んで、Add-Bond-Bondlength で結合をつける。
- ③Add-H to cut AA で、切断されてできた終端原子に水素を付ける。
- ④chain F を select して、最適化対象にする。
- ⑤Script メニューから、tinker_optimize.py を実行し、パネルでオプション (RMS gradient 1kcal/A? など) をセットして計算実行。

リスト 21 TINKER による構造最適化計算

```
(省略)
22   -715.5719      1.6957      2.4148      0.1492      319  TruncNewt      306
23   -718.1711      0.9632      2.5992      0.0712      333  TruncNewt      323
TNCG -- Normal Termination due to SmallGrad
Final Function Value :           -718.1711
Final RMS Gradient :              0.9632
Final Gradient Norm :             22.7946
... e:Wtinker-6.2.06Wbin-win64/newton.exe ended at 2014/04/16 18:36:36
... elapsed time=           28.58 (sec)
```

- ⑥Overly ボタンを押して、元の構造と比較し、よければ Update ボタンを押す。
- ⑦3arca-pro@-uF-e5-o.pdb 名で保存する。

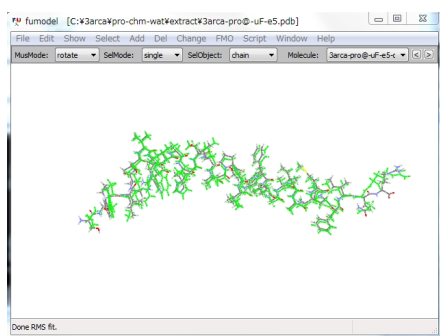


図 21 元の構造（緑）と最適化後の構造の rms fit(主鎖のみ)。RMSD=0.894Å。

⑫replace_res ツールで、3arca@@.pdb 中のもものと置き換える。(6-③ですすでにやった)

2) 環境中の非ペプチド分子の構造最適化

例として、HTG698B を取り上げる。

①から③は、7-1)と同じ。

②HTG を選択状態にする。

③Script GAMESS_assist.py を実行する。

④Runtype で optimize を選択し、Makeinput ボタンを押す

⑤Edit/View メニューでエディタを開き、input file に、\$contrl icharg=-1, \$statpt opttol=0.01, \$basis GBASIS=AM1 \$end を追加する。

⑥RunGAMESS ボタンを押して、GAMESS を実行する。

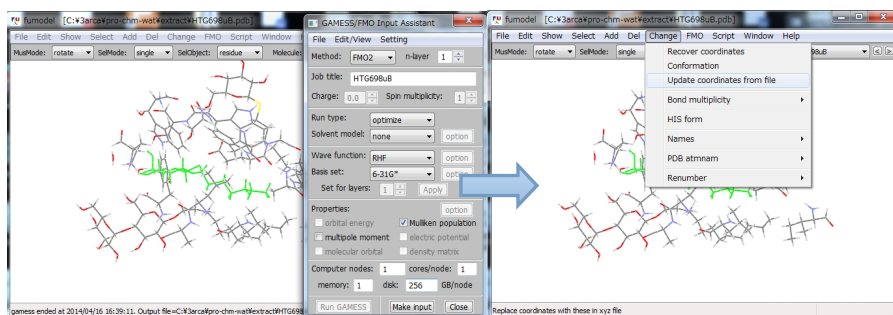


図 22 GAMESS による構造最適化

リスト 22 GAMESS による構造最適化計算

```

MAXIMUM GRADIENT = 0.0087992    RMS GRADIENT = 0.0011381
  NSERCH:   15    E=      -1434.1968347733    GRAD. MAX=   0.0087992    R.M.S.=
0.0011381

      ***** EQUILIBRIUM GEOMETRY LOCATED *****
      (省略)
.....END OF GEOMETRY SEARCH.....
CPU      0: STEP CPU TIME=      6.60 TOTAL CPU TIME=      344.0 (   5.7 MIN)
TOTAL WALL CLOCK TIME=      344.0 SECONDS, CPU UTILIZATION IS 100.00%
  
```

GAMESS の入力ファイルは 'HTG698uB-e5.inp'、出力ファイルは 'HTG698uB-e5.out' となる (PDB ファイルのベース名に .inp と .out 識別子が付けられる)。

⑦Change-Update coordinates from file メニューを実行する。

StatusBar に 'deviation between old and new coordinates=0.0053 Å' メッセージがでたので、異常な構造になっていないと判断できる。

(補足) 石橋をたたいて渡る人は、ここで新と旧構造を RMS fit して見ればよい。

⑧Save as メニューで、HTG698B-e5-o.pdb と名づけて保存する。

⑨replace_res ツールで、3arca@@.pdb 内の該当する残基を置き換える (6-③ですでにやった)。

(補足) 多数の入力データを手作業で作成するのはうんざりする。fumodel の script を作って、一括処理を行うのがよい。本講習会用に、make_aml_input.py スクリプト (c:/ps2/fu-script フォルダーにある) を作成したので使って欲しい。これを使うために、3arca@@.pdb を pick_up_obj ツールで (residues, with environment option) で抽出する (抽出ファイルの格納ディレクトリは、c:\ps2\3arca@@\chm-res-e50 とする。水分子全部もひとつの "residue" として分割されているので、この計算が不要の場合、c:\ps2\3arca@@\chm-res-e50\3arca@@-res.sp1 から削除しておく。これらの準備が終わったら、PyCrust コンソールから、

```
> fum.ExecuteScript1('c://ps2//fu-script//make_aml_input.py')
```

でスクリプトを実行する。同じディレクトリ内に、xxx.inp の名前で GAMESS の入力ファイルが作成される。また、これらを一括計算するための Windows のバッチファイル xxx.bat が作成される (GAMESS のバージョンなどが異なる場合は、この script の後尾部分にある dat='rungms '+infile+' 13-64 1 0 '+outfile+' %n' を修正すればよい)。

(注意) 分子の電荷やスピン多重度は正しくない恐れがあるのでチェックが必要。なお、イオンの電荷は、3arca@@.pdb を fumodel に読み込んで、イオンを選択して

Change-Charge メニューで事前に設定しておく。ポリペプチドの残基の電荷は、FMO のフラグメントに電荷を付ける機能を利用して script の中でつけている。

10 万原子超系の構造最適化を完全に QM レベルでやるのは、京コンピュータを使っても困難で、exa スケールコンピュータの時代を待たなければならない。ここで説明した MM と AM1 さらに OEC などに *ab initio* を用いていわゆる *microiteration optimization* なら実行可能である。FMO 法の *frozen domain* 近似 (FMO/FD, D.G.Fedorov et al., J. Phys. Chem. Lett., 2, 282 (2011)) で全系の環境下で *microiteration optimization* を行うことも可能であろう。

7. 縮小モデルの作成

全系から一部をとりだした縮小モデルは、それ自体をモデル系として研究対象にすることが多いばかりでなく、全系計算の準備として様々な予備計算を行う場合にも使う。従って、場合に応じて様々な縮小モデルが簡単に作れると便利である。

これは、すでにやったように `extract_res(with environment option)` ツールで容易に作成できるので、ここでは繰り返さない。futools には、切った結合を水素でキャップするツールがないので、作成後 `fumodel` で水素をつけた。1 万原子程度の系であれば描画の待ち時間も苦にならないので、`fumodel` で縮小モデルを作ると、水素キャップ、構造最適化が連続して行える。興味ある人は `fumodel` の使用説明書を参照されたい。

8. OEC クラスターの置換

ps2 の水分解反応機構の量子化学計算による研究で、OEC の様々な構造が提案されている。そのほとんどが OEC クラスターのみを対象としたものであるが、周辺の一部のアミノ酸残基 (側鎖のみ) を含めたモデルも作られている。これらを、全系の中に埋め込んで、新たなモデルを作成することが多いと推測される。`fumodel` の、欠落原子の修復で使った技術をこの目的にも使うことができる。

OEC だけを入れ替えるので、巨大なタンパク質全体を扱うのではなく、まず、OEC+環境 (5 Å) の縮小モデルを作っておいて、これに `transfer` する。ついで、この OEC を全系のものと置き換えるという 2 段階で行うことにする。

- 1) OEC+環境残基モデルを作成する。ファイル名は、
c:/ps2/neese-oec-model/OEX601A-e50.pdb とする (OEX-e50 と呼ぶ)。

リスト23 extract_res

```
e
executing program extract_res
+++ input residue data in full form, like "ALA:1:A"(do not type quotes)
OEX:601:A
+++ with environment? 1:no, 2:yes, 3:quit
2
+++ input cut-off distance(A) of environment.
5
+++ enter file name of input pdb file
c:\ps2\3arca\3arca@@.pdb
+++ enter file name of output pdb files
c:\ps2\neese-oec-model\OEXA-e50.pdb
--- extract_res job starts at 2014/04/26 19:11:22
running ...
extracted residues: ['OEX:601:A']
cut off distance of environmental residue= 5.0
environmental residues: ['ASP:61:A', 'ASN:87:A', 'TYR:161:A',
(省略)
```

2) Neese の OEC モデル

F. Neese らの OEC モデルは周辺のアミノ酸残基（の側鎖）を含むので、タンパク質への埋め込みが容易である。まず、座標データはデカルト座標なので、これを PDB ファイルに変換する（この際、座標パラメータの桁落ちが起こるので、あくまで初期構造の作成として使う）。

① 元のデカルト座標

c:/ps2/ps2-material/oec-models/neese-oec-2012/neese-structure-a.xyz

(Dimitrios A. et al., Angew. Chem. Int. Ed. 2012, 51, 9935, SI:Optimized Cartesian Coordinates Structure A) を fumodel に読み込む。

② Add-bond-Use bond length で結合を付ける。

③ File-Save as メニューで、c:\ps2\neese-oec-model\neese-structure-a.pdb に保存する (OEX-neese と呼ぶ)。

④ この PDB ファイルをエディタで編集して、OEC クラスターに、原子名を付ける。

リスト 24 OCEのデカルト座標データをPDBで保存

REMARK Created by fu at 2014/03/20 21:21:43

HETATM	1	MN1	-1	-24.871	-35.533	203.969	0.00	0.00	MN 0
HETATM	2	MN2	-1	-27.359	-35.264	205.245	0.00	0.00	MN 0
HETATM	3	MN3	-1	-27.348	-33.371	203.184	0.00	0.00	MN 0
HETATM	4	MN4	-1	-27.606	-33.205	200.442	0.00	0.00	MN 0
HETATM	5	CA1	-1	-27.875	-36.772	202.211	0.00	0.00	CA 0
HETATM	6	O1	-1	-26.431	-36.528	204.227	0.00	0.00	O 0
HETATM	7	O2	-1	-28.376	-34.764	203.816	0.00	0.00	O 0

(以下省略)

⑤File-Open メニューで OEX601A-e50. pdb を読み込む。

⑥Script メニューから、rms-fit.py を実行し、Reference molecule: の Read ボタンを押して、OEX-neese を読み込む。両方で OEC 周辺のアミノ酸残基（側鎖）はほとんど同じなので、次の rms fit をしなくても、OEX-neese の OEX 部分をそのまま OEX601A-e50. pdb あるいは全系の PDB ファイルにコピーするだけでよいことが分かるが、一応、手順として次に進む。

⑦Select-Select All メニューで全原子を選択状態にする。

⑧RMS Fitting パネルで、maiching parameter を atom names だけにして、Matting atoms: の Apply ボタンを押して、さらに RMS fit: の Fit ボタンを押す。

アミノ酸残基（側鎖）部分は、実験構造と完全に一致しているので、何もすることなく、replace_res ツールで入れ替えればよいことが分かる。

⑨ neese の OEX を transfer した構造データに、Add-Add H to cut AA メニューで終端に水素を付加し、（見やすくするために）さらに OEX クラスターに結合をつけて、c:\yps2\neese-oec-model\neese-structure-a-e50. pdb 名で保存する（これが OEC を含む縮小モデルである）。

（補足）結合距離に基づいて結合をつけても OEX クラスターの結合は正しく付かないので、手動で付ける。SelObject を atom、SelMode を dual にして、2 個の原子を選択状態にして” 1” キー（数字のいち）を押すと結合ができる。結合を削除する場合は” d” キーを押す。

（補足）Neese らのモデルは、周辺のアミノ酸残基があるので置換が容易である。これが無い裸の OEC クラスターの場合では、fumodel の機能だけで適切な構造を作るのは難しいので、取り込んだあとで、何らかのやり方でペプチド鎖との相対配置を決めなければならない（OEC クラスターを剛体として扱って、何らかの構造最適化を実行する

必要がある)。

9. おわりに

モデリングを簡単に、再現性を担保して行うためにFUを開発してきた。モデリングの技術は個人が習得するだけでなく広く共有すべきである。ノウハウをソフトウェアの機能として実装すれば、みんなが容易に共有できて、モデリング作業に費やす時間が省け、最も重要なサイエンスの研究により多くの時間を割くことができる。一方、モデル構造の再現性が担保されると、第3者が構造モデルを再現することができるだけでなく、本人も系統的にモデルの改良を進めることにも役立つ。ps2のような複雑な分子系の構造モデルの場合、ソフトウェアのバージョンアップのように、みんなで継続的に改良を続けるべきである。

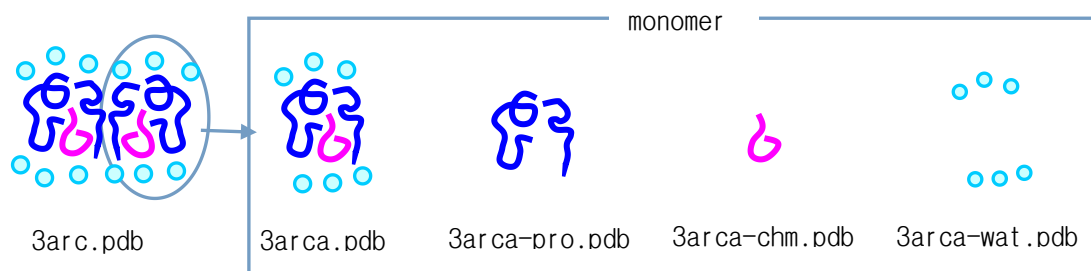
FUを開発を始めてまだ2年弱で、現時点で十分な機能を持つとはいいがたい。特に、“安直モデル”の作成は、終わって見れば、単に水素を付けたただけなのに、何で沢山の作業を行う必要があったのか疑問を感じる。たとえば、1) mht ファイルの収集は、事前に一つ一つダウンロードした。これは、当然、ソフトウェアで自動的に収集させるべきである、2) 水素を付けるのに、分割したり統合したりを行ったが、丸ごとを一回ボタンを押すだけでできるはずである（ここでやった作業は人間がやる価値があるものではない）、3) 構造の品質検査までを自働でやって、おかしい構造が含まれていれば自働で修正できるはずである。

本実習の結果をまとめると、pro への水素付加は fumodel の水素付加機能のバグのせいで分割して行った。これを修正すれば、直接 3arca.pdb に水素付加をして 3arca@-pro.pdb ファイルを作ることができる。また、最後に HIS のプロトンを削除して 3acra@.pdb 中のものと置き換えた操作は、his_delete_h.py スクリプトをもう少し賢くして、金属に配位している HIS だけでなく、近辺の水素結合性の官能基を調べて自動的に処理できるはずである。ここまでできれば、一発のコマンドで“安直モデル”(3arca@@.pdb) が作れるようになるので、モデリングソフトウェアとして合格「可」と評価していただけると思う（あと一步）。

今回は、電子状態計算ができるクオリティの構造モデルを作成することをテーマとした（“最適化モデル”を作ることであるが、目標を達成できなかった）。“安直モデル”には、結合距離が短かすぎるまたは長すぎるたり、結合角がおかしいものなどの部分が含まれていて、このままでは電子状態計算が収束しない。“最適化モデル”を自動的に作れるようにすべきである（あと二歩）。

京コンピュータのような巨大計算機が利用可能になったことから、今後、巨大・複雑・多数の分子系の計算が日常的に行われるはずである。そのような時代に、計算は1日で終わるのに、その構造モデルを作るのに1週間かけるのでは釣り合わない（人間の価値が相対的に低くなりすぎる）。この観点から、FUはまだまだ実用レベルに到達していない。今後も継続して改良していく予定である。皆様のご意見・ご協力を願うものである。

附録 A PDB ファイルの命名ルール



- 1) split_pdb ツールの pro-chm-wat option で分割したファイルには、元の PDB ファイルのベース名-pro.pdb, -chm.pdb, -wat.pdb となる。(例: 3arca-pro.pdb)
- 2) split_pdb ツールの chain option で分割したファイルには、元の PDB ファイルのベース名-chain 名.pdb となる (chain 名は大文字の場合 uA, 小文字の場合 la となる)。(例: 3arc-uA.pdb)
- 3) split_pdb ツールの resdiue option で分割したファイルのファイル名は、元の PDB ファイルのベース名-残基名+残基番号-chain 名となる。(例: 3arca-chm-SQD668-uB.pdb)
- 4) split_pdb ツールの unique residue option で分割したファイルには、元の PDB ファイルのベース名-残基名.pdb)となる。(例: 3arca-chm-SQD.pdb)

附録 B 残基の記述式

残基名: 残基番号: chain 名 (3 文字: 1 から 5 数字文字: 1 文字) と書く。

(例) SQD:768:D または SQD:*** (*は何がきてもよいことを示す。この場合、全ての SQD)

附録 C FU で特別な意味を持つファイル識別子

.pdb ... PDB file
.ent ... PDB file
.xyz ... FU の Cartesian 座標ファイル (座標データその他フラグメントデータを含む)。
GAMESS および GAMESS-FMO の Cartesian ファイルも読める。
.tin ... TINKER の xyz file
.mht ... 分子のフレームデータのファイル
.fuf ... fumodel で各種ファイル(.pdb, .mht, .frg, .bda など)を一括して読む
ときに使う。ファイル名を記載したファイルで format は.spl や.mrg と同じ。
.frg ... fragment data file
.bda ... bda data file

.inp ... GAMESS input file
.out ... GAMESS output file
.log ... futools の pdb log file. (session log ファイルの識別子は既定でない)
.fub ... futools のバッチファイル
.spl ... split_pdb ツールで作成した PDB file のファイル名を記載したファイル
.mrg ... merge_pdb ツールで読み込むファイル (.spl と同じ format)

附録 D 中間データを保存するディレクトリ構造

FU で決まっているわけではないが、多種類の多数のファイルを扱うので、本講習会では、以下のように、ディレクトリで整理する。

黒字は提供ファイル（またはディレクトリ）のコピーで、青字が実習で作成するファイル（およびディレクトリ）である。

```
c:/ps2 ← 配布データの ps2 フォルダをコピーする
|__./ps2-material
|    |__./pdb-data
|    |    |__ pdb3arc.ent
|    |    |__ ...
|    |__./mhtfiles
|    |    |__BCR.mht
|    |    |__ ...
|    |__./oec-models
|    |    |__./batista-oec-2011
|    |    |    |__ ...
|    |    |__./nesse-oec-2012
|    |    |    |__ ...
|    |    |__./seigbarn-oec-2013
|    |    |__ ...
|__./fu-script
|    |__his_delete_h.py
|    |__make_am1_input.py
|    |__add_h_to_pro.py
|    |__add_h_to_chm.py
|__mhtfiles.fuf
|__./3arc
|    |__3arc.pdb
|    |__./chain
|    |    |__3arc-chinxxx.pdb
|    |    |__ ...
|    |    |__3arc-chain.spl
|__./neese-oec-model ← おみやげのおまけ
|    |__neese-structure-a-e50.pdb
|    |__ ...
```



```

|__./3arca
|   |__3arca.pdb
|   |__3arca.mrg
|   |__./pro-chm-wat
|       |__./extract
|       |   |   |__ ...
|       |   |__./pro-chain
|       |   |   |__3arca-pro-xxx.pdb
|       |   |   |__ ...
|       |   |__3arca-pro.pdb
|       |   |__3arca-chm.pdb
|       |   |__3arca-wat.pdb
|   |__./3arca@
|       |__./his-e40
|       |   |__ ...
|   |__3arca@.pdb
|   |__3arca@.mrg
|   |__3arca@@.pdb ← “安直モデル” の完成品
|   |__./3arca@@
|       |__./chm-res-e50
|       |   |__xxx.pdb
|       |   |__xxx.inp
|       |   |__ ...
|   |__3arca@@-wat35.pdb

```