myPresto 5.0

- cosgene_pack -

TUTORIAL

2020/08/12

Copyright (C) 2006-2020 Next Generation Natural Product Chemistry (N^2PC)

本ドキュメントについて

本ドキュメントは、「***myPresto* 5.0** USER MANUAL」の別冊です。コピーライト、プログラム使用許諾条件、著者および引用文献については、「***myPresto* 5.0** USER MANUAL」の記述に準じます。

謝辞

本ソフトウェアの研究開発は、国立研究開発法人日本医療研究開発機構(AMED)の援助によって行われました。ここに感謝の意を記します。

本ソフトウェアは、故・京極好正博士の始められた研究の中で開発されました。

目次

1. 本チュートリアルについて 1
2. cosgene_pack のインストール方法 3
3. cosgene_pack のディレクトリ構成 6
4. テストプログラムの実行 7
5. MD 計算の手順 8
6. MD 計算実行までの手順の具体例 9
6.1. 準備と PDB ファイル(X 結晶構造情報)の確認 10
6.2. PDB ファイルの加工 12
6.3. pdbcheck の実行 13
6.4. Hgene の実行 16
6.5. tplgeneL の実行 18
6.6. tplgeneX の実行 21
6.7. タンパク質と低分子化合物の PDB ファイルのマージ 24
6.8. setwater の実行 25
6.9. add_ion の実行 28
6.10. tplgeneX の実行(2 回目) 30
6.11. tpl2capbc の実行 33
6.12. min.inp の作成 35
6.13. cosgene の実行(エネルギー極小化) 36
6.14. SHAKEinp の実行 37
6.15. md.inp の作成 39
6.16. cosgene の実行(MD) 42
6.17. 解析 43
7. 発展 44
8. ツールプログラムについての説明 45
9. 参照文献 48

1. 本チュートリアルについて

創薬計算用ソフトウェア myPresto は、MD シミュレーションプログラム(cosgene、cosgene_MPI, psygene-G、omegagene)、ドッキングシミュレーションプログラム(sievgene)、高分子用トポロジー情報ファイル作成プログラム(tplgeneX)、低分子トポロジー情報ファイル作成プログラム(tplgeneL)等、多くのプログラムから構成されています。

本チュートリアルでは、myPresto に含まれるプログラムのうち、基本的な MD 計算の実行に必要なプログラムを集めた cosgene_pack を使用します。cosgene_pack は、myPresto を使った MD 計算を初めて実行する方が、簡単に MD 計算を実行するための環境を提供しています。簡単な MD 計算を実際に実行することにより、myPresto での MD シミュレーションの準備作業プロセスについて理解を深めることができます。cosgene_pack は、Linux が動いている比較的多くの計算機環境で動作するように、設計されています。

より効率的に MD 計算を行うためには、複数の CPU コアを同時に使って MD 計算を同時に計算する cosgene_MPI を使います。cosgene_MPI を使うためには、Message-Passing Interface(MPI)が利用可能な環境をあらかじめ設定した後で、cosgene_MPI をコンパイルする必要があります。MPI やジョブスケジューリングプログラムを使った実行方法は、環境によって使用するコマンドが異なります。また、管理者権限が無いと設定できない場合があります。

また、myPresto には、GPU を使って高速に MD 計算を行う psygene-G が用意されています。Psygene-G の実行には、GPGPU が使える環境が必要です。本マニュアルでは、psygene-G を使った MD 計算実行手順については説明しません。

cosgene_pack には、以下のプログラムが含まれています。

- cosgene　（MD 計算エンジン）
- tplgeneX　（タンパク質、核酸等の高分子用のトポロジーファイル作成プログラム）
- tplgeneL　（低分子化合物用のトポロジーファイル作成プログラム）
- Hgene　（低分子化合物の部分電荷を計算するプログラム）
- add_ion　（水溶液中にイオンを配置するプログラム）
- setwater　（水分子を配置するプログラム）
- tpl2capbc　（球状に配置した水に対する束縛を設定するプオグラム）
- SHAKEinp　（SHAKE を設定するプログラム）
- RIGIDinp　(剛体モデルを設定するプログラム)
- pdbcheck　（PDB ファイルを修正するプログラム）
- ツールスクリプトプログラム
  - get_pdb_info.pl　(PDB ファイルの内容を確認するプログラム)
  - select_chain.pl　(PDB ファイルから特定の鎖を抜き出すプログラム)
  - select_res.pl　(PDB ファイルから特定の残基名の分子を抜き出すプログラム)
  - del_res.pl　(PDB ファイルから特定の残基名の行を削除するプログラム)
  - exec_tplgeneL.sh　(環境変数 TPLL_DB_PATH を設定せずに tplgeneL を実行するプログラム)
  - exec_tplgeneX.sh　(環境変数 TPL_DB_PATH を設定せずに tplgeneX を実行するプログラム)
- インストール関連プログラム
  - install.sh　(インストーラープログラム)
  - check_binary.sh　(バイナリプログラムをチェックするプロググラム)
  - clean_binary.sh　(バイナリプログラムを削除するプロググラム)
- テスト計算自動実行プログラム
  - test_analysis.sh　(解析のテスト実行プログラム)

  以下は、MD のテスト実行プログラムです。4HP0 もしくは 4lxz は初期構造に使用している立体構造の PDB ID を示し、CAP はタンパク質の周りに球状に水分子を配置した系、peri は直方体状で周期的境界の系、cosgene もしくは cosgene_MPI は計算に使うプログラムを示しています。
  - test_MD_4HP0_CAP_cosgene_MPI.sh
  - test_MD_4HP0_CAP_cosgene.sh
  - test_MD_4HP0_peri_cosgene_MPI.sh
  - test_MD_4HP0_peri_cosgene.sh
  - test_MD_4lxz_CAP_cosgene_MPI.sh
  - test_MD_4lxz_CAP_cosgene.sh
  - test_MD_4lxz_peri_cosgene_MPI.sh

cosgene_pack に含まれているプログラムは以上です。

2. cosgene_pack のインストール方法

cosgene_pack は、cosgene_packYYMMDD.tar.gz という圧縮ファイルで配布されています。(YYMMDD は、年月日を表す数字です。) 本パッケージは、Linux/Unix 環境用で使用することを想定しています。本パッケージは、ユーザーが書き込み可能なディレクトリ(例えば、ホームディレクトリ)にインストールして、自分のみで使用ことを想定しています。/usr/local/bin 等の共用ディレクトリにはインストールしませんので、管理者用アカウントは必要ありません。書き込み可能なディレクトリのどこかに、cosgene_packYYMMDD.tar.gz を配置して、次のコマンドで解凍してください。

```
% tar -xzvf cosgene_packYYMMDD.tar.gz
```

解凍後に cosgene_packYYMMDD という名前のディレクトリができます。次に、ソースコードからコンパイルするプログラムをインストールします。インストールするためのスクリプトプログラム(install.sh)を用意しています。このインストールプログラムでは、FORTRAN コンパイラと C コンパイラを使用します。現在は、GNU のコンパイラ(gfortran と gcc)、もしくは、インテル社のコンパイラ(ifort と icc)のどちらか一方がインストールされていることを前提としています(および一般的なシステムにインストールされている C コンパイラ(cc))。他のコンパイラを使用するためには、各プログラムをコンパイルするための Makefile を編集する必要があります。

GNU のコンパイラを使用してインストールするには、以下のコマンドを実行します。Intel コンパイラを使用したインストールは後述します。

```
% cd cosgene_packYYMMDD
% bin/install.sh
```

ここでは、まず、ディレクトリを cosgene_packYYMMDD/に移動して、その後に、cosgene_packYYMMDD/bin/にある install.sh を実行しています。install.sh を引数なしで実行した場合には、GNU のコンパイラ(gfortran, gcc)を使ってプログラムをコンパイルします。

インテル社のコンパイラを使用してインストールするためには、次のように"intel"を引数に与えて install.sh を実行します。

```
% bin/install.sh intel
```

install.sh は、カレントディレクトリ(現在のディレクトリ)がどこでも実行することができますが、install.sh を実行するためには、install.sh へのパス(相対パスもしくは絶対パス)を install.sh の前に付ける必要があります。上述のコマンド(bin/install.sh)において、install.sh の前にある"bin/"は、cosgene_packYYMMDD/から install.sh への相対パスを表しています。カレントディレクトリが、cosgene_packYYMMDD/bin ならば、"./install.sh"と実行します。cosgene_packYYMMDD.tar.gz を解凍したディレクトリならば、"cosgene_packYYMMDD/bin/install.sh"と実行します。このように、カレン

トディレクトリによって、実行時のコマンドが若干異なります。カレントディレクトリと cosgene_packYYMMDD/bin/との位置関係を意識しておくことが重要です。cosgene_packYYMMDD/bin/へのパスを環境変数 PATH に設定すると相対パスの指定なしに、その場所にあるコマンドを実行することも可能ですが、本マニュアルではパスの設定をせずに、相対パスを指定してコマンドを実行する方法で、以下説明します。cosgene_packYYMMDD/bin/の下に、以下の実行ファイルが作成されていれば、インストールは問題なく完了しています。

- cosgene
- tplgeneX
- tplgeneL
- pdbcheck
- Hgene
- setwater
- add_ion
- tpl2capbc
- SHAKEinp

実行ファイルが作成されていないものについては、各プログラムのコンパイル用に用意されている Makefile の内容をチェックしてください。これらのバイナリの存在確認をするコマンドは、次のものです。(カレントディレクトリが cosgene_packYYMMDD/の場合)

```
% bin/check_binary.sh
```

引数を与えずに install.sh を実行した場合には、以下のコンパイラを使います。

- cosgene: gfortran　(cosgene_MPI は非対応)
- tplgeneX: gcc
- tplgeneL: /usr/bin/cc
- Hgene: gfortran, gcc
- pdbcheck: gfortran
- setwater: gfortran
- add_ion: gfortran
- tpl2capbc: gfortran
- SHAKEinp: cc

“intel”を引数に与えて install.sh を実行した場合には、以下のコンパイラを使います。

- cosgene: ifort　(cosgene_MPI は非対応)
- tplgeneX: gcc
- tplgeneL: /usr/bin/cc
- Hgene: ifort, icc
- pdbcheck: ifort
- add_ion: ifort
- setwater: ifort
- tpl2capbc: ifort
- SHAKEinp: cc

3. cosgene_pack のディレクトリ構成

cosgene_pack のディレクトリ構成は以下のようになっています。

cosgene_pack のディレクトリ構成:

```
cosgene_packYYMMDD/
 |--README
 |--bin/
 |    |--install.sh
 |    |--exec_tplgeneX.sh
 |    |--exec_tplgeneL.sh
 |    |--get_pdb_info.pl
 |    |--select_chain.pl
 |    |--select_res.pl
 |    |--del_res.pl
 |    |--test_MD_4lxz.sh
 |    |--check_binary.sh
 |    |--clean_binary.sh
 |    |--(cosgene)         (install.sh 実行後に出現)
 |    |--(tplgeneX)        (install.sh 実行後に出現)
 |    |--(tplgeneL)        (install.sh 実行後に出現)
 |    |--(Hgene)           (install.sh 実行後に出現)
 |    |--(pdbcheck)        (install.sh 実行後に出現)
 |    |--(setwater)        (install.sh 実行後に出現)
 |    |--(add_ion)         (install.sh 実行後に出現)
 |    |--(tpl2capbc)       (install.sh 実行後に出現)
 |    `--(SHAKEinp)        (install.sh 実行後に出現)
 |--sample/
 |    |--min.inp
 |    |--md.inp
 |    `--pdb4lxz.ent
 |--src/
 |    |--Makefile
 |    |--cosgene/
 |    |--tplgeneX/
 |    |--tplgeneL/
 |    |--Hgene/
 |    |--pdbcheck/
 |    |--setwater/
 |    |--add_ion/
 |    |--tpl2capbc/
 |    `--SHAKEinp/
 `--doc/
      |-- Hgene_manual_v5.0_en_YYMMDD.pdf
      |-- Hgene_manual_v5.0_ja_YYMMDD.pdf
      |-- cosgene_manual_v5.0_en_YYMMDD.pdf
      |-- cosgene_manual_v5.0_ja_YYMMDD.pdf
      |-- cosgene_sample_manual_v5.0_en_YYMMDD.pdf
      |-- cosgene_sample_manual_v5.0_ja_YYMMDD.pdf
      |-- cosgene_tutorial_v.5.0_ja_YYMMDD.pdf
      |-- pdbcheck_manual_v5.0_ja_YYMMDD.pdf
      |-- tplgeneL_manual_v5.0_en_YYMMDD.pdf
      |-- tplgeneL_manual_v5.0_ja_YYMMDD.pdf
      |-- tplgeneX_manual_v5.0_en_YYMMDD.pdf
      |-- tplgeneX_manual_v5.0_ja_YYMMDD.pdf
      `-- trj2pdb_manual_v5.0_ja_YYMMDD.pdf
            (YYMMDD は年月日を表す数字で、作成日を示すものです。)
```

4. テストプログラムの実行

インストールに成功したら、動作テスト用プログラムを実行するとよいでしょう。このプログラムは、サンプルとして用意したもので、タンパク質と低分子化合物の複合体(PDB ID: 4lxz)の周りに球状に水分子を配置し、イオンを付加して、エネルギー極小化を行い、その後に、MD 計算を実行します。(テストプログラムは、実行後、完了までに約 15 分かかります。)

cosgene_packYYMMDD/の下で、次のコマンドを実行してください。

```
% bin/test_MD_4lxz_CAP_cosgene.sh
```

計算の出力先ディレクトリは、cosgene_packYYMMDD/test_MD_4lxz/です。

カレントディレクトリを確認するコマンドは、pwd です。カレントディレクトリを確認して、cosgene_packYYMMDD/とは別のディレクトリにいる場合には、cd コマンドを使って移動してください。

5. MD 計算の手順

以下の手順でドッキングシミュレーションを実施します。

(1) オリジナルの PDB ファイルの内容を確認します。
(2) ドッキングで使用する分子を検討します。特に、計算に使用しない分子で、計算に使用しない分子を検討します。結晶構造情報の中には、結晶化の安定剤として添加された分子が入っていることもあります。また、結晶構造では、同じユニットが、複数個含まれていることもありますが、MD 計算の際には、１つの天然構造のユニットに対して計算を行うとよいでしょう。
(3) タンパク質と低分子化合物との複合体の MD 計算の準備段階では、一旦、タンパク質と低分子化合物を分離して別のファイルにします。これは、タンパク質の処理は tplgeneX で、低分子化合物の処理は tplgeneL で行うためです。別々のファイルに保存して、それぞれのプログラムの入力ファイルにします。タンパク質に結合した金属イオンは、タンパク質の方に含めます。
(4) (3) で切り出した低分子化合物の PDB ファイルを Hgene で処理します。この処理によって、tplgeneL で扱えるファイル形式になります。
(5) tplgeneL で低分子化合物を処理して、PDB ファイルとトポロジーファイルを作成します。
(6) (3)で分離したタンパク質ファイルの末尾に、(5)で作成した化合物の PDB ファイルを追加します。
(7) (6)で作成した複合体の PDB を tplgeneX で処理します。この際に、(5)で作成した低分子化合物のトポロジーファイルを使用します。これで、複合体の PDB ファイルとトポロジーファイルが作成されます。
(8) 複合体の周辺に配置する水分子を、setwater で発生させます。
(9) add_ion でイオンを発生させます。
(10) 水分子を球状の範囲に閉じ込めておくための束縛ファイルを tpl2capbc で発生させます。
(11) エネルギー極小化計算用の cosgene 用入力ファイルを作成します。
(12) エネルギー極小化計算を実行します。
(13) SHAKE 設定ファイルを作成します。
(14) MD の本計算用の cosgene 用入力ファイルを作成します。
(15) MD の本計算を実行します。

6. MD 計算実行までの手順の具体例

ヒストン脱アセチル化酵素2(HDAC2)と阻害剤の複合体に対してMD計算を行う手順について説明します。ここでは、球状に配置した水の中に複合体を置いた系に対する計算を行うことにします。HDAC2・阻害剤複合体の初期構造は、X線結晶構造(PDB ID: 4lxz)を使用します。結晶構造からMD計算用の系を作成する手順は、少し頻雑です。

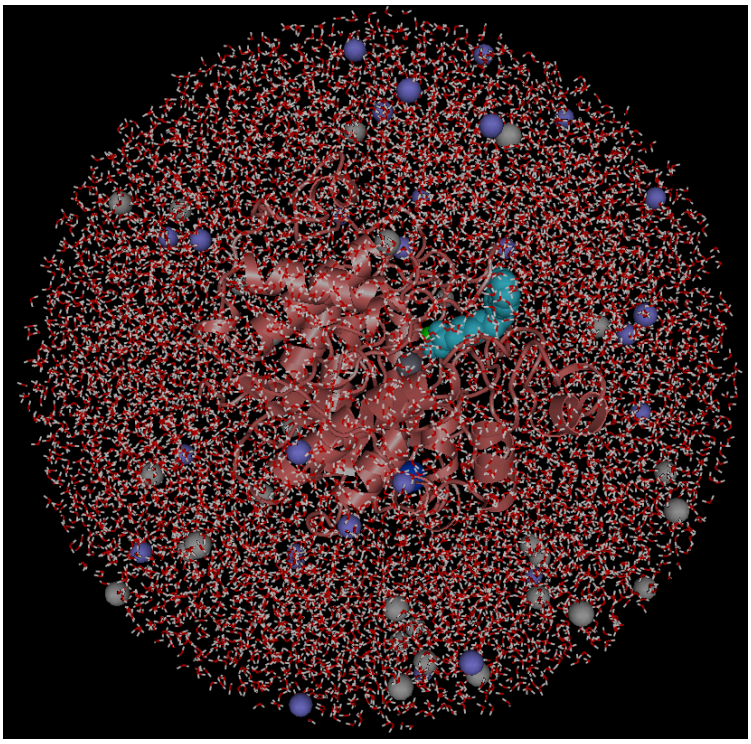

図 1. イオンを含む水球中でのタンパク質・阻害剤複合体の系の例

6.1. 準備と PDB ファイル(X 結晶構造情報)の確認

MD計算で用いる系の作成には、多くの場合、Protein Data Bank(PDB)に登録されている構造情報ファイル(PDBファイル)を用います。しかし、PDBファイルからMD計算でも用いる系の座標情報を作成するには、いくつかの処理が必要です。

まず、作業用ディレクトリを作成し、作成したディレクトリの下に移動します。ここでは、作業ディレクトリの名前はwork_MD_4lxzにします。作業ディレクトリを作成するカレントディレクトリがcosgene_packYYMMDD/ではない場合には、そこに移動してください。(他のディレクトリでも計算可能ですが、bin/への相対パスが異なります。)

```
% pwd        （カレントディレクトリを表示するコマンド。）
cosgene_packYMMDD でない場合は、そこに移動してから以下のコマンドを実行してください。
% mkdir work_MD_4lxz
% cd work_MD_4lxz
```

test_MD_4lxzは、テスト実行プログラム(bin/test_MD_4lxz.sh)の出力先のディレクトリ名として設定されていますので、それとは違う名前にしています。

次に、サンプルファイルを作業ディレクトリにコピーします。

```
% cp ../sample/pdb4lxz.ent .
```

このサンプルファイルを使って、MD計算の手順について説明します。最初に行うべきことは、PDBファイルに含まれる分子の構成を調査をすることです。次のコマンドで、PDBファイルの内容を確認します。

```
% perl ../bin/get_pdb_info.pl pdb4lxz.ent
```

get_pdb_info.plの出力例:

```
No SSBOND line.
No of peptide chain:3
Chain ID 1: A
Chain ID 2: B
Chain ID 3: C

No of ligand name:7
Ligand name 1:  ZN
Ligand name 2:  CA
Ligand name 3:  NA
Ligand name 4: PG4
Ligand name 5: SHH
Ligand name 6: NHE
Ligand name 7: HOH
```

この出力を見ると、pdb4lxz.pdbには、ペプチド鎖が3本(A鎖、B鎖、C鎖)、ペプチド鎖以外の分子(つまり、HETATMから始まる行に登録されている分子)が、ZN、CA、NA、PG4、SHH、NHE、HOHの7種類が含まれていることが確認できます。PDBフ

ァイルの内容をよく確認すると、3本のペプチド鎖は同じもので、阻害剤はSHHであることが分かります。PG4とNHEは、結晶化の際の添加剤と思われます。1つのペプチド鎖の内部には、亜鉛イオン、カルシウムイオン、ナトリウムイオンが結合しています。さらに、結晶構造における水分子の酸素原子座標も含まれています。この結晶構造自体でなく、水溶液中での複合体構造に興味がある場合には、水溶液中の状態に近いと考えられる複合体構造のみを取り出します。ここでは、結晶化の添加剤と考えられるPHEとPG4を取り除き、水溶液中では三量体でなく単量体だと考えられますので、A鎖のみを取り出すとよいでしょう。

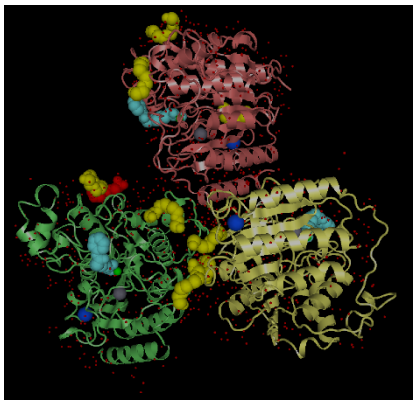

図 2. pdb4lxz.pdb に含まれる X 線結晶構造
ペプチド鎖はリボン表示で、水分子の酸素原子はドット表示(赤色(見えにくくなっています))で、その他の低分子化合物と金属イオンは vdW 球表示で描画されている。低分子化合物としては、SHH(水色)、PG4(黄色)、PHE(赤色)が含まれており、金属イオンとしては、ZN(灰色)、CA(緑色)、NA(青色)が含まれている。

A鎖と、A鎖に結合したSHH、ZN、CA、NAを使用し、残りは使用しないことにします。

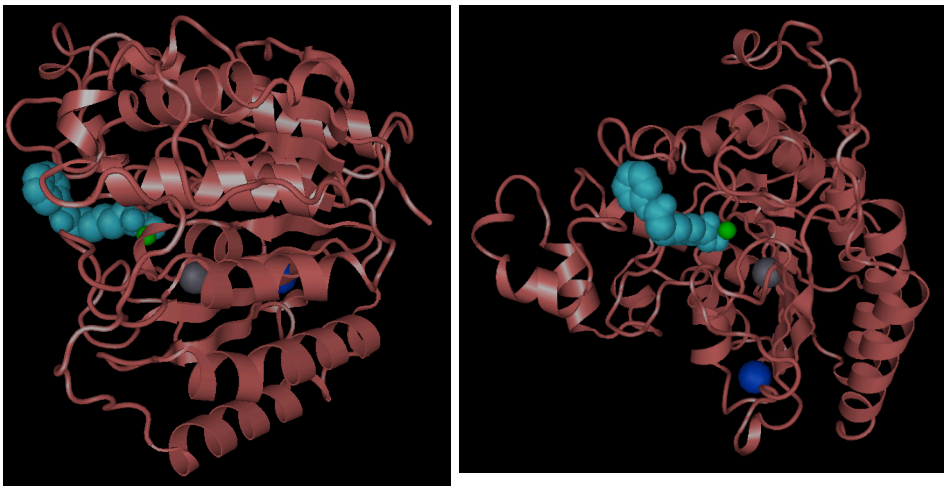

図 3. MD で使用するタンパク質・阻害剤・金属イオン
2 つの図は異なる角度から見たもの

6.2. PDB ファイルの加工

前節で検討した結果、pdb4lxz.entからA鎖と、A鎖に結合したSHH、ZN、CA、NAを取り出して使用することにしました。次に行うべきことは、低分子化合物に対して適切な処理を行うことです。そのため、pdb4lxz.entからA鎖に結合したSHHの情報を抽出して、１つのファイルに保存します。また、A鎖と、A鎖に結合した金属イオン(ZN、CA、NA)も別のファイルに抽出して保存します。この２つのファイルを作成するには、以下のコマンドを実行します。

```
% perl ../bin/select_chain.pl A pdb4lxz.ent > tmp_selectChain
（A 鎖と A 鎖に結合した分子・金属イオンの抽出）
% perl ../bin/del_res.pl HOH tmp_selectChain > tmp_delRes1
（HOH の削除）
% perl ../bin/del_res.pl PG4 tmp_delRes1 > tmp_delRes2
（PG4 の削除）
% perl ../bin/del_res.pl SHH tmp_delRes2 > tmp_delRes3
（SHH の削除）
```

A鎖と、A鎖に結合した金属イオンの情報がtmp_delRes3に保存されています。
実際に、tmp_delRes3の内容が想定した通りなのかをget_gdb_info.plで確認しましょう。

```
% perl ../bin/get_pdb_info.pl tmp_delRes3
```

get_pdb_info.pl の出力:

```
No SSBOND line.
No of peptide chain:1
Chain ID 1: A

No of ligand name:3
Ligand name 1:  ZN
Ligand name 2:  CA
Ligand name 3:  NA
```

ペプチド鎖はA鎖のみで、リガンドはZN、CA 、NAのみとなっていることが確認できます。

また、次のコマンドで、阻害剤のみをtmp_ligandに取り出します。

```
% perl ../bin/select_res.pl SHH tmp_selectChain > tmp_ligand
```

6.3. pdbcheck の実行

前節で、タンパク質と金属イオンを含むPDBファイルと、低分子化合物を含むPDBファイルの２つを作成しました。この段階で、PDBファイルのイレギュラーを検出、もしくは、修正するために、pdbcheckを実行します。pdbcheckを実行するためには、まず、pdbcheck制御用の入力ファイルを用意します。ここでは、次の３行を含むテキストファイル(inp_pdbcheck (ファイル名は任意))を用意します。

inp_pdbcheck の内容:

```
tmp_delRes3
tmp_pdb_checked.pdb
-alt
```

1行目は読み込むPDBファイルの名前、2行目は出力するPDBファイルの名前、3行目以降は、オプションを記述します。ここで使用している-altオプションは、同じ原子について登録されている複数の座標情報(alternate location indicator)が含まれている場合に、最初の座標のみを出力するオプションです。他に、ジスルフィド結合を自動検出して、PDBファイルの先頭にSSBOND行(ジスルフィド結合をしている原子ペアの情報)を書き込むオプション(-ss)等があります。ここでは、get_pdb_info.plの出力に、SSBOND 行が含まれていませんでしたので、−ssオプションは使用していません。
inp_pdbcheckは、テキストエディタで用意してもかまいませんが、次のコマンド群を使用することにより、テキストエディタを使用しないで作成することができます。

```
% echo tmp_delRes3 > inp_pdbcheck
% echo tmp_pdb_checked.pdb >> inp_pdbcheck
% echo -alt >> inp_pdbcheck
```

作成したinp_pdbcheckの確認は、次のコマンドで行います。

```
% cat inp_pdbcheck
```

pdbcheckは、次のように実行します。

```
% ../bin/pdbcheck < inp_pdbcheck
```

pdbcheck の出力:

```
PDB CHECK TOOL v1.1.1  2014. Sep. 19

  1st   : input PDB file name
  2nd   : output PDB file name
  3rd.. : -cap -hyd -alt -bb -dih -ss -as
          -disableHet -keepTer -remark
          -mod -mut -rot -cmp
INFORMATION INPUT
  1) INPUT FILE
    tmp_delRes3
  2) OUTPUT FILE
    tmp_pdb_checked.pdb
  3) SPECIFIED OPTION
     -alt
```

```
INFORMATION> DIVISION OF CHAINS.
  CHAIN NAME   RESIDUE NAME   RESIDUE ID   REASON(EXCEPT TER AND CHAIN ID)
  A     A       CA      NA     402   403   TERMINAL OF NA ION GROUP(TOP)
  A     A       ZN      CA     401   402   DIFFERENCE OF LIGAND RESIDUE NAME
  A     A      PRO      ZN     379   401   DISTANCE IS FAR

INFORMATION> EXIST ALTERNATE LOCATION INDICATOR
  RESIDUE NAME   RESIDUE ID   ATOM NAME
  LYS              36          CE
  LYS              36          NZ
  ARG              54          CG
  ARG              54          CD
  ARG              54          NE
  ARG              54          CZ
  ARG              54          NH1
  ARG              54          NH2
 (中略)
  ARG             376          CB
  ARG             376          CG
  ARG             376          CD
  ARG             376          NE
  ARG             376          CZ
  ARG             376          NH1
  ARG             376          NH2

INFORMATION> REPAIR ALTERNATE LOCATION INDICATOR
  RESIDUE NAME   RESIDUE ID   ATOM NAME
  LYS              36          CE
  LYS              36          NZ
  ARG              54          CG
  ARG              54          CD
  ARG              54          NE
  ARG              54          CZ
  ARG              54          NH1
  ARG              54          NH2
(中略)
  ARG             376          CB
  ARG             376          CG
  ARG             376          CD
  ARG             376          NE
  ARG             376          CZ
  ARG             376          NH1
  ARG             376          NH2

INFORMATION> LACKED MAIN CHAIN
  RESIDUE NAME   RESIDUE ID   ATOM NAME

INFORMATION> TERMINAL RESIDUE
  RESIDUE NAME   RESIDUE ID   DISTANCE

INFORMATION> SSBOND CANDIDATES

INFORMATION> CYS-(CYM) CANDIDATES
  CHAIN ID   RESIDUE NAME   RESID ID

INFORMATION> EXIST NEAR ATOM
  RESIDUE NAME   RESIDUE ID   ATOM NAME

INFORMATION> LINEAR TORSION ATOM
  RESIDUE ID           ATOM NAME

 WARNING: TOPOLOGY FILE NOT EXIST:C99_aa.tpl
          SKIP RESIDUE ANALYSIS
INFORMATION> OUTPUT
```

```
  OUTPUT FILE
    tmp_pdb_checked.pdb
```

リガンドについても同様に pdbcheck を実行します。

```
% echo tmp_ligand > inp_pdbcheck_lig
% echo lig.pdb >> inp_pdbcheck_lig
% echo -alt >> inp_pdbcheck_lig
% cat inp_pdbcheck_lig
```

inp_pdbcheck_lig の内容を確認して、タイプミスがないことを確認した後で、次のコマンドを実行してください。

```
% ../bin/pdbcheck < inp_pdbcheck_lig
```

lig.pdb(Hgene の入力ファイル):

```
HETATM    1  O1   SHH A   1       19.636 -19.886  -2.094  1.00  9.90
HETATM    2  O2   SHH A   1       20.493 -17.663  -0.936  1.00 11.81
HETATM    3  N1   SHH A   1       20.883 -19.782  -1.456  1.00 12.88
HETATM    4  C1   SHH A   1       21.257 -18.620  -0.886  1.00 11.91
HETATM    5  C2   SHH A   1       22.597 -18.546  -0.195  1.00  8.48
HETATM    6  C3   SHH A   1       23.558 -17.483  -0.789  1.00 11.04
HETATM    7  C4   SHH A   1       24.933 -17.701  -0.133  1.00 14.84
HETATM    8  C5   SHH A   1       25.972 -16.715  -0.701  1.00 18.97
HETATM    9  C6   SHH A   1       27.325 -17.005  -0.024  1.00 21.78
HETATM   10  C7   SHH A   1       28.304 -15.828  -0.183  1.00 22.61
HETATM   11  C8   SHH A   1       28.010 -14.702   0.779  1.00 25.92
HETATM   12  O3   SHH A   1       27.461 -13.712   0.333  1.00 27.41
HETATM   13  N2   SHH A   1       28.377 -14.832   2.094  1.00 27.90
HETATM   14  C9   SHH A   1       28.270 -13.899   3.158  1.00 28.65
HETATM   15  C10  SHH A   1       28.005 -12.534   2.969  1.00 29.96
HETATM   16  C11  SHH A   1       28.001 -11.658   4.055  1.00 30.92
HETATM   17  C12  SHH A   1       28.266 -12.130   5.340  1.00 28.90
HETATM   18  C13  SHH A   1       28.553 -13.478   5.537  1.00 28.01
HETATM   19  C14  SHH A   1       28.581 -14.346   4.451  1.00 27.71
TER
```

6.4. Hgene の実行

pdb形式のリガンドファイルはtplgeneLで扱えません。Hgeneを用いてmol2ファイルに変換します。次のコマンドを実行してください。

```
% ../bin/Hgene -ipdb lig.pdb -p -mop AM1BCC -omol2 lig.mol2
```

このコマンドによりlig.mol2が作成されます。-pオプションを使用すると、酸性/塩基性官能基が解離状態になるように水素原子を付加します。-mopオプションは、ハミルトニアンを指定してMOPAC7 の計算を行います。ここでは、AM1BCCのハミルトニアンを指定しています。Hgeneのその他の使用方法については、Hgeneのマニュアルを参照してください。

lig.mol2 (Hgene の出力ファイル) :

```
@<TRIPOS>MOLECULE
lig.mol2
 39 39 0 0 0
SMALL
USER_CHARGES

@<TRIPOS>ATOM
     1 O1          19.6360   -19.8860    -2.0940 O.3       1  SHH        -0.1622
     2 O2          20.4930   -17.6630    -0.9360 O.2       1  SHH        -0.3354
     3 N1          20.8830   -19.7820    -1.4560 N.am      1  SHH        -0.3083
     4 C1          21.2570   -18.6200    -0.8860 C.2       1  SHH         0.3019
     5 C2          22.5970   -18.5460    -0.1950 C.3       1  SHH        -0.1660
     6 C3          23.5580   -17.4830    -0.7890 C.3       1  SHH        -0.1328
     7 C4          24.9330   -17.7010    -0.1330 C.3       1  SHH        -0.1576
     8 C5          25.9720   -16.7150    -0.7010 C.3       1  SHH        -0.1500
     9 C6          27.3250   -17.0050    -0.0240 C.3       1  SHH        -0.1449
    10 C7          28.3040   -15.8280    -0.1830 C.3       1  SHH        -0.1724
    11 C8          28.0100   -14.7020     0.7790 C.2       1  SHH         0.3157
    12 O3          27.4610   -13.7120     0.3330 O.2       1  SHH        -0.3391
    13 N2          28.3770   -14.8320     2.0940 N.am      1  SHH        -0.3348
    14 C9          28.2700   -13.8990     3.1580 C.ar      1  SHH         0.0753
    15 C10         28.0050   -12.5340     2.9690 C.ar      1  SHH        -0.1565
    16 C11         28.0010   -11.6580     4.0550 C.ar      1  SHH        -0.0951
    17 C12         28.2660   -12.1300     5.3400 C.ar      1  SHH        -0.1500
    18 C13         28.5530   -13.4780     5.5370 C.ar      1  SHH        -0.0951
    19 C14         28.5810   -14.3460     4.4510 C.ar      1  SHH        -0.1565
    20 H1          19.7733   -20.1131    -3.0062 H         1  SHH         0.1923
    21 H2          21.4957   -20.5717    -1.4241 H         1  SHH         0.2369
    22 H3          22.4546   -18.3153     0.7675 H         1  SHH         0.0928
    23 H4          23.0511   -19.4339    -0.2693 H         1  SHH         0.0928
    24 H5          23.6114   -17.6097    -1.7795 H         1  SHH         0.0913
    25 H6          23.2040   -16.5713    -0.5804 H         1  SHH         0.0913
    26 H7          24.8500   -17.5559     0.8529 H         1  SHH         0.0767
    27 H8          25.2363   -18.6362    -0.3157 H         1  SHH         0.0767
    28 H9          26.0411   -16.8485    -1.6896 H         1  SHH         0.0895
    29 H10         25.6763   -15.7805    -0.5030 H         1  SHH         0.0895
    30 H11         27.1734   -17.1644     0.9515 H         1  SHH         0.0799
    31 H12         27.7314   -17.8169    -0.4431 H         1  SHH         0.0799
    32 H13         29.2340   -16.1525    -0.0106 H         1  SHH         0.1010
```

```
    33 H14          28.2348   -15.4705    -1.1143 H          1  SHH          0.1010
    34 H15          28.7867   -15.6852     2.4168 H          1  SHH          0.2260
    35 H16          27.8174   -12.1834     2.0514 H          1  SHH          0.1342
    36 H17          27.8075   -10.6871     3.9136 H          1  SHH          0.1266
    37 H18          28.2496   -11.5008     6.1171 H          1  SHH          0.1246
    38 H19          28.7385   -13.8203     6.4581 H          1  SHH          0.1266
    39 H20          28.8280   -15.3030     4.6033 H          1  SHH          0.1342
@<TRIPOS>BOND
     1     1     3   1
     2     2     4   2
     3     3     4  am
     4     4     5   1
     5     5     6   1
     6     6     7   1
     7     7     8   1
     8     8     9   1
     9     9    10   1
    10    10    11   1
    11    11    12   2
    12    11    13  am
    13    13    14   1
    14    14    15  ar
    15    14    19  ar
    16    15    16  ar
    17    16    17  ar
    18    17    18  ar
    19    18    19  ar
    20     1    20   1
    21     3    21   1
    22     5    22   1
    23     5    23   1
    24     6    24   1
    25     6    25   1
    26     7    26   1
    27     7    27   1
    28     8    28   1
    29     8    29   1
    30     9    30   1
    31     9    31   1
    32    10    32   1
    33    10    33   1
    34    13    34   1
    35    15    35   1
    36    16    36   1
    37    17    37   1
    38    18    38   1
    39    19    39   1
```

6.5. tplgeneL の実行

ここではリガンドのトポロジーファイルを作成します。まずは、tplgeneLの制御用入力ファイルを作成します。tplgeneLについては、cosgeneのマニュアルを参照してください。

作成すべきファイル(inp_tplgeneL)の内容:

```
2
lig.mol2
1
gaff21.db
no
```

このファイルの意味は、以下の通りです。

```
[1行目] 読み込むファイル形式( 1: tplgeneLオリジナル形式, 2: Sybyl mol2形式)
[2行目] 読み込むファイル名
[3行目] パラメーターが無い場合の処理
        1: デフォルトパラメータを使用
        2: パラメーターを計算する
        3: デフォルトパラメーターがある場合にはそれを使い、無い場合には計算する
[4行目] 使用するポテンシャルのデータベースファイル名
[5行目] フラグメントデータベースを使用するか (yes/no)
```

この内容のファイルは、テキストエディタで用意してもかまいませんが、以下のコマンドを実行すると、同じ内容のファイルが作成されます。

```
% echo 2 > inp_tplgeneL
% echo lig.mol2 >> inp_tplgeneL
% echo 1 >> inp_tplgeneL
% echo gaff21.db >> inp_tplgeneL
% echo no >> inp_tplgeneL

% cat inp_tplgeneL  (内容の確認)
```

tplgeneLの実行は以下のコマンドで行います。

```
% ../bin/exec_tplgeneL.sh < inp_tplgeneL
```

exec_tplgeneL.shは、そのプログラムの中でtplgeneLの実行前に必要な環境変数TPLL_DB_PATHを設定してからtplgeneLを実行するプログラムです。
tplgeneLを実行すると、低分子化合物のトポロジーファイルが作成されます。作成されるトポロジーファイルの名前は、入力ファイルの名前から拡張子を取り除いたものに、".tpl"を付加したものです。

(参考)

tplgeneLは対話的に実行できるように設計されています。対話的に実行する場合の画面を以下にします。tplgeneLの制御用入力ファイルとして作成したinp_tplgeneLは、対話的に実行する際に、キーボードから入力する項目をファイルに保存したもので、OSのリダイレクト機能を使って、キーボードからの入力をファイルからの入力に切り替えた実行方法が、次のコマンドです。

```
% ../bin/exec_tplgeneL.sh < inp_tplgeneL
```

tplgeneL を対話的に実行した際の出力:

```
  ************************************************
 *                                              *
 *                  tplgeneL                    *
 *                                              *
 *                Dec  5, 2016                  *
 *                                              *
 ************************************************


 Please select Input File Format by the next number!
   1 : tplgeneL original (*.bond,*.charge,*.zmat)
   2 : Sybyl mol2 (*.mol2)
2   <----   (キーボードからの入力項目)

 INFORMATION> main
      Sybyl mol2 Input File was selected.


 Please select Input File Name!
./
capbc              inp_tplgeneL   Original_aminoacid_No.dat  Pro_4.tpl
command_list       inp_tplgeneX   pdb4lxz.ent                setwater.log
cry_wat_x.pdb      inp_tplgeneX2  pdbcheck_lig.log           tmp_delRes1
fort.16            lig.mol2       pdbcheck.log               tmp_delRes2
fort.30            lig.pdb        Pro_0.pdb                  tmp_delRes3
inp_add_ion        lig.tpl        Pro_0.tpl                  tmp_ligand
inp_pdbcheck       lig_tplL.pdb   Pro_1.pdb                  tmp_pdb_checked.pdb
inp_pdbcheck_lig   M_all.res      Pro_2.pdb                  tmp_selectChain
inp_setwater       min.inp        Pro_3.pdb                  wat.pdb
inp_tpl2capbc      min.out        Pro_4.pdb
lig.mol2   <----   (キーボードからの入力項目)

 INFORMATION> ltgGetFilename
      Get File Name, "./lig"

 What processing do you do if there is a missing parameter?
 Please select 1, 2 or 3! (default : 1)
   1 :  use default parameters.
   2 :  calculate parameters.
   3 :  use default parameters, when default parameters exist.
        use calculated parameters, when default parameters don't exist.
1   <----   (キーボードからの入力項目)

 Please select Input DB Name!
   (default : gaff17.db)
/home/ec2-user/cosgene_pack161210/src/tplgeneL161205/src/tplgeneL/tplgeneL/DB/
amber99.db  gaff17.db  gaff18.db  gaff21.db
gaff21.db   <----   (キーボードからの入力項目)
```

```
 Do you want to use FragmentDB ? (yes(y)/no(n) default : no)
no    <----    (キーボードからの入力項目)

 WARNING> ltgCorrectNotExistParm
      Improper-torsion parameter is not defined.
      Default parameter is used.
        13   15   14   19:   n    ca   ca   ca
      ==>                    X    X    3c1  X


 %% Program is done. %%
 %% This program ended normally. %%
```

赤字で示した入力項目を、１つにつき１行で記述したものが、制御用入力ファイルです。

つまり以下の内容のファイルです。

```
2
lig.mol2
1
gaff21.db
no
```

myPrestoに含まれるプログラムのいくつかは、このように対話的に実行するように設計されており、対話的実行をする際の入力項目が制御用入力ファイルに書かれています。

6.6. tplgeneX の実行

本章で説明するMDの準備プロセスでは、tplgeneXを数回実行します。tplgeneXは、入力されたPDBをシミュレーションに適した形式に加工すると同時に、そのPDBファイルに対応したトポロジーファイルを作成します。ここで、実行する最初のtplgeneXの実行は、PDBファイルの加工を行うためです。

まず、制御用入力ファイル(inp_tplgeneX)を作成します。

inp_tplgeneX の内容:

```
pdb
C99
tmp_pdb_checked.pdb
pdb
Pro_0.pdb
Pro_0.tpl
```

このファイルは、tplgeneX(exec_tplgeneX.sh)を対話的に実行した際の入力項目を記述したものです。tplgeneXを対話的に実行した時の出力例を以下に示します。

tplgeneX を対話的に実行した時の出力例:

```
 ************************************************
 *                                              *
 *            tplgeneX version 0.9.2            *
 *                                              *
 *                 Jul 11, 2015                 *
 *                                              *
 ************************************************
 %% Enter Inputfile Type or the number. %%
   1 : pdb
   2 : pdbx (PDBx/mmCIF)
   3 : dihed (for Amino Acid Only)
pdb    ←----    (キーボードからの入力項目)

 %% Enter Forcefield Type or the number. %%
   1 : C99
   2 : C96
   3 : charmm22(for Amino Acid Only)
   4 : charmm19(for Amino Acid Only)
   9 : other Force Field(Modified AMBER FF, etc.)
C99    ←----    (キーボードからの入力項目)
 %% Enter an Input Coordinate or Dihed File Name. %%
inp_pdbcheck              lig.tpl                  tmp_delRes2
         tmp_selectChain
inp_pdbcheck_lig lig_tplL.pdb             tmp_delRes3
lig.mol2         pdb4lxz.ent              tmp_ligand
lig.pdb                  tmp_delRes1              tmp_pdbcheck.pdb
tmp_pdbcheck.pdb    ←----    (キーボードからの入力項目)
 %% Enter Outputfile Type or the number. %%
   1 : pdb
   2 : pdbx
pdb    ←----    (キーボードからの入力項目)
 %% Enter an Output Coordinate File Name. %%
Pro_0.pdb    ←----    (キーボードからの入力項目)
 %% Enter an Output Topology File Name. %%
Pro_0.tpl    ←----    (キーボードからの入力項目)
```

```
INFORMATION> printMolInfo
            Molecule Number           :1
            Total number of residues  :368

INFORMATION> printMolInfo
            Amino acid Sequence of the protein

Molecule Number :1
GLYN+  LYS+   LYS+   LYS+   VAL    CYS    TYR    TYR    TYR    ASP-
GLY    ASP-   ILE    GLY    ASN    TYR    TYR    TYR    GLY    GLN
GLY    HIS    PRO    MET    LYS+   PRO    HIS    ARG+   ILE    ARG+
MET    THR    HIS    ASN    LEU    LEU    LEU    ASN    TYR    GLY
LEU    TYR    ARG+   LYS+   MET    GLU-   ILE    TYR    ARG+   PRO
HIS    LYS+   ALA    THR    ALA    GLU-   GLU-   MET    THR    LYS+
TYR    HIS    SER    ASP-   GLU-   TYR    ILE    LYS+   PHE    LEU
ARG+   SER    ILE    ARG+   PRO    ASP-   ASN    MET    SER    GLU-
TYR    SER    LYS+   GLN    MET    GLN    ARG+   PHE    ASN    VAL
GLY    GLU-   ASP-   CYS    PRO    VAL    PHE    ASP-   GLY    LEU
PHE    GLU-   PHE    CYS    GLN    LEU    SER    THR    GLY    GLY
SER    VAL    ALA    GLY    ALA    VAL    LYS+   LEU    ASN    ARG+
GLN    GLN    THR    ASP-   MET    ALA    VAL    ASN    TRP    ALA
GLY    GLY    LEU    HIS    HIS    ALA    LYS+   LYS+   SER    GLU-
ALA    SER    GLY    PHE    CYS    TYR    VAL    ASN    ASP-   ILE
VAL    LEU    ALA    ILE    LEU    GLU-   LEU    LEU    LYS+   TYR
HIS    GLN    ARG+   VAL    LEU    TYR    ILE    ASP-   ILE    ASP-
ILE    HIS    HIS    GLY    ASP-   GLY    VAL    GLU-   GLU-   ALA
PHE    TYR    THR    THR    ASP-   ARG+   VAL    MET    THR    VAL
SER    PHE    HIS    LYS+   TYR    GLY    GLU-   TYR    PHE    PRO
GLY    THR    GLY    ASP-   LEU    ARG+   ASP-   ILE    GLY    ALA
GLY    LYS+   GLY    LYS+   TYR    TYR    ALA    VAL    ASN    PHE
PRO    MET    ARG+   ASP-   GLY    ILE    ASP-   ASP-   GLU-   SER
TYR    GLY    GLN    ILE    PHE    LYS+   PRO    ILE    ILE    SER
LYS+   VAL    MET    GLU-   MET    TYR    GLN    PRO    SER    ALA
VAL    VAL    LEU    GLN    CYS    GLY    ALA    ASP-   SER    LEU
SER    GLY    ASP-   ARG+   LEU    GLY    CYS    PHE    ASN    LEU
THR    VAL    LYS+   GLY    HIS    ALA    LYS+   CYS    VAL    GLU-
VAL    VAL    LYS+   THR    PHE    ASN    LEU    PRO    LEU    LEU
MET    LEU    GLY    GLY    GLY    GLY    TYR    THR    ILE    ARG+
ASN    VAL    ALA    ARG+   CYS    TRP    THR    TYR    GLU-   THR
ALA    VAL    ALA    LEU    ASP-   CYS    GLU-   ILE    PRO    ASN
GLU-   LEU    PRO    TYR    ASN    ASP-   TYR    PHE    GLU-   TYR
PHE    GLY    PRO    ASP-   PHE    LYS+   LEU    HIS    ILE    SER
PRO    SER    ASN    MET    THR    ASN    GLN    ASN    THR    PRO
GLU-   TYR    MET    GLU-   LYS+   ILE    LYS+   GLN    ARG+   LEU
PHE    GLU-   ASN    LEU    ARG+   MET    LEU    PROC-


INFORMATION> setCoordinate
            All the atom positions are now set.

INFORMATION> outputTpl
            Output formatted Topology File

INFORMATION> outputCoord
            Output coordinate file

CALC. TIME = 0.1284 sec.

%% Program is done. %%
%% This program ended normally. %%
```

pdbcheckの時と同じようにechoコマンドでファイルを作成するには、以下のように実行します。(テキストエディタで作成してもかまいません。)

```
% echo pdb > inp_tplgeneX
% echo C99 >> inp_tplgeneX
% echo tmp_pdb_checked.pdb >> inp_tplgeneX
% echo pdb >> in_tplgeneX
% echo Pro_0.pdb >> inp_tplgeneX
% echo Pro_0.tpl >> inp_tplgeneX
```

次のコマンドでtplgeneX(exec_tplgeneX.sh)を実行してください。

```
% ../bin/exec_tplgeneX.sh < inp_tplgeneX
```

次のexec_tplgeneX.shは、そのプログラムの中で、最初にtplgeneX実行に必要なデータベースファイルのパスを環境変数として設定し、その後にtplgeneXを実行します。つまり、簡単な設定をした後に、tplgeneXを実行するものです。

(参考)

inp_tplgeneXは、以下のように記述して上述のinp_tplgeneXのものと同様に動作します。これは、tplgeneXからの質問項目に番号で回答したパターンになります。

inp_tplgeneX の記述例:

```
1
1
tmp_pdb_checked.pdb
1
Pro_0.pdb
Pro_0.tpl
```

6.7. タンパク質と低分子化合物の PDB ファイルのマージ

上述の手順では、タンパク質と低分子化合物について別々に処理して、それぞれのPDBファイルを作成しました。次のコマンドで２つのファイルをマージします。

```
% cat Pro_0.pdb lig_tplL.pdb > Pro_1.pdb
```

Pro_0.pdbに追加したlig_tplL.pdbの内容は以下の通りです。水素原子が付加されており、部分電荷付加と原子質量の情報も付加されています。また、この低分子化合物に対するトポロジーファイルは、2行目のREMARKから始まる行に登録されています。この情報は、この低分子化合物を含む系のトポロジーファイルをtplgeneXで作成する際に使われます。その際には、この化合物のトポロジーファイル(lig.tpl)が読み込めないとエラーになります。

```
REMARK ORIGMOL2 ./lig.mol2
REMARK ORIGTPL  ./lig.tpl
ATOM      1  O1  SHH     1       19.636 -19.886  -2.094 16.00 -0.16
ATOM      2  O2  SHH     1       20.493 -17.663  -0.936 16.00 -0.34
ATOM      3  N3  SHH     1       20.883 -19.782  -1.456 14.01 -0.31
ATOM      4  C4  SHH     1       21.257 -18.620  -0.886 12.01  0.30
ATOM      5  C5  SHH     1       22.597 -18.546  -0.195 12.01 -0.17
ATOM      6  C6  SHH     1       23.558 -17.483  -0.789 12.01 -0.13
ATOM      7  C7  SHH     1       24.933 -17.701  -0.133 12.01 -0.16
ATOM      8  C8  SHH     1       25.972 -16.715  -0.701 12.01 -0.15
ATOM      9  C9  SHH     1       27.325 -17.005  -0.024 12.01 -0.14
ATOM     10  C10 SHH     1       28.304 -15.828  -0.183 12.01 -0.17
ATOM     11  C11 SHH     1       28.010 -14.702   0.779 12.01  0.32
ATOM     12  O12 SHH     1       27.461 -13.712   0.333 16.00 -0.34
ATOM     13  N13 SHH     1       28.377 -14.832   2.094 14.01 -0.33
ATOM     14  C14 SHH     1       28.270 -13.899   3.158 12.01  0.08
ATOM     15  C15 SHH     1       28.005 -12.534   2.969 12.01 -0.16
ATOM     16  C16 SHH     1       28.001 -11.658   4.055 12.01 -0.10
ATOM     17  C17 SHH     1       28.266 -12.130   5.340 12.01 -0.15
ATOM     18  C18 SHH     1       28.553 -13.478   5.537 12.01 -0.10
ATOM     19  C19 SHH     1       28.581 -14.346   4.451 12.01 -0.16
ATOM     20  H20 SHH     1       19.773 -20.113  -3.006  1.01  0.19
ATOM     21  H21 SHH     1       21.496 -20.572  -1.424  1.01  0.24
ATOM     22  H22 SHH     1       22.455 -18.315   0.767  1.01  0.09
ATOM     23  H23 SHH     1       23.051 -19.434  -0.269  1.01  0.09
ATOM     24  H24 SHH     1       23.611 -17.610  -1.780  1.01  0.09
ATOM     25  H25 SHH     1       23.204 -16.571  -0.580  1.01  0.09
ATOM     26  H26 SHH     1       24.850 -17.556   0.853  1.01  0.08
ATOM     27  H27 SHH     1       25.236 -18.636  -0.316  1.01  0.08
ATOM     28  H28 SHH     1       26.041 -16.849  -1.690  1.01  0.09
ATOM     29  H29 SHH     1       25.676 -15.780  -0.503  1.01  0.09
ATOM     30  H30 SHH     1       27.173 -17.164   0.952  1.01  0.08
ATOM     31  H31 SHH     1       27.731 -17.817  -0.443  1.01  0.08
ATOM     32  H32 SHH     1       29.234 -16.152  -0.011  1.01  0.10
ATOM     33  H33 SHH     1       28.235 -15.470  -1.114  1.01  0.10
ATOM     34  H34 SHH     1       28.787 -15.685   2.417  1.01  0.23
ATOM     35  H35 SHH     1       27.817 -12.183   2.051  1.01  0.13
ATOM     36  H36 SHH     1       27.808 -10.687   3.914  1.01  0.13
ATOM     37  H37 SHH     1       28.250 -11.501   6.117  1.01  0.12
ATOM     38  H38 SHH     1       28.738 -13.820   6.458  1.01  0.13
ATOM     39  H39 SHH     1       28.828 -15.303   4.603  1.01  0.13
TER
```

6.8. setwater の実行

前節でタンパク質と低分子化合物を含むPDBファイルを準備しました。次に、複合体の周辺に水を配置します。ここでは、複合体の周囲に球状に、タンパク質表面から8オングストロームのマージンで、水分子を配置することにしします。setwaterの制御用入力ファイルは、次のものを使ってください。

inp_setwater の内容:

```
Pro_1.pdb
N
wat.pdb
S
-8.0
C
1.0
1.0
3
Y
setwater.log
N
```

このファイルは、対話的に実行した際の入力項目になります。各行に記述した内容は、対話的実行での質問項目に対する答えです。質問内容は、次の出力例で確認してください。

setwater を対話的に実行した際の出力例:

```
--- setwater ---
Input file name (PDB of target molecule) ?
Pro_1.pdb    <----    (キーボードからの入力項目)
  -> Pro_1.pdb
Do you use crystal water file (Y or N) ?
N    <----    (キーボードからの入力項目)
  -> not use.
Input file name (output of water coord) ?
wat.pdb    <----    (キーボードからの入力項目)
  -> wat.pdb
Input cell type (sphere="S", ellipsoid="E", cube="C", parallelepiped="P") ?
S    <----    (キーボードからの入力項目)Input radius (R) ?
(positive vale:radius,negative value:mergin)
-8.0    <----    (キーボードからの入力項目)
  -> sphere :    -8.0000000000
Input center of water (mass center="C", 3D-coordinate="D") ?
C    <----    (キーボードからの入力項目)
  -> mass center
Input density of water (usually 1.0) ?
1.0    <----    (キーボードからの入力項目)
  ->     1.0000000000
Input vdW damping factor (usually 1.0) ?
1.0    <----    (キーボードからの入力項目)
  ->     1.0000000000
Input water model(TIP3P="3", TIP4P="4") ?
3    <----    (キーボードからの入力項目)
  -> TIP3P
Do you use precise model (Y or N) ?
Y    <----    (キーボードからの入力項目)
Input file name (information output) ?
setwater.log    <----    (キーボードからの入力項目)
  -> setwater.log
```

```
Do you use membrane (Y or N) ?
N   ←――   (キーボードからの入力項目)
 numatomp        5881
 ntype=           1        4477
 ntype=           2        1634
 ntype=           3        4477
 ntype=           4        3944
 ntype=           5        3728
 ntype=           6        4774
 n_str=           1 npot=      225420
 ave_&_max_potential=  0.249992028      0.948207736
 ave_&_max_potential=  0.372312039      0.948207736
 no_crywat=         861
 @@ n_atom=         5881
 min/max  =  -12.927000000000000        43.008000000000003
 min/max  =  -48.764000000000003        6.4470000000000001
 min/max  =  -31.616000000000000        26.285000000000000
  xyz     =  -16.000000000000000       -16.000000000000000       -16.000000000000000
```

setwater 制御用入力ファイル作成のコマンドは以下の通りです。テキストエディタで作成してもよいでしょう。

```
% echo Pro_1.pdb > inp_setwater
% echo N >> inp_setwater
% echo wat.pdb >> inp_setwater
% echo S >> inp_setwater
% echo -8.0 >> inp_setwater
% echo C >> inp_setwater
% echo 1.0 >> inp_setwater
% echo 1.0 >> inp_setwater
% echo 3 >> inp_setwater
% echo Y >> inp_setwater
% echo setwater.log >> inp_setwater
% echo N >> inp_setwater

確認コマンド
% cat inp_setwater
```

setwater の実行方法は次の通りです。

```
% ../bin/setwater < inp_setwater
```

setwater が作成した wat.pdb の内容:

```
HETATM    1  O   WAT    1       0.421 -51.084 -29.423 16.00 -0.83
HETATM    2  H1  WAT    1       1.035 -50.471 -29.905  1.01  0.42
HETATM    3  H2  WAT    1       0.214 -50.495 -28.654  1.01  0.42
 (中略)
HETATM   94  O   WAT   33      16.824   8.854  28.904 16.00 -0.83
HETATM   95  H1  WAT   33      16.499   8.252  28.214  1.01  0.42
```

```
HETATM   96  H2  WAT    33      16.015   8.966  29.454  1.01  0.42
REMARK     SETWATER(v2.0) RESULTS
REMARK       TARGET MOLECULE   -> Pro_1.pdb
REMARK       CRYSTAL WATER     -> NOT USE
REMARK       CELL    ->      S   89.607026   89.607026   89.607026
REMARK       CENTER  ->      C   12.984611  -22.805681   -1.301643
REMARK       DENSITY ->     1.000000
REMARK       DAMP F. ->     1.000000    (PROBE R. =     1.400000A)
REMARK       NUMBER OF CRYSTAL WATER   :        0
REMARK       NUMBER OF WATER MOLECULE  :    10032
REMARK       NUMBER OF WATER ATOM      :    30096
```

setwaterの出力は水分子のみです。読み込んだタンパク質(とリガンド)のPDBは、出力されませんので、タンパク質（とリガンド）と水の両方を含むファイルを作成するには、次のコマンドを実行します。

```
% cat Pro_1.pdb wat.pdb > Pro_2.pdb
```

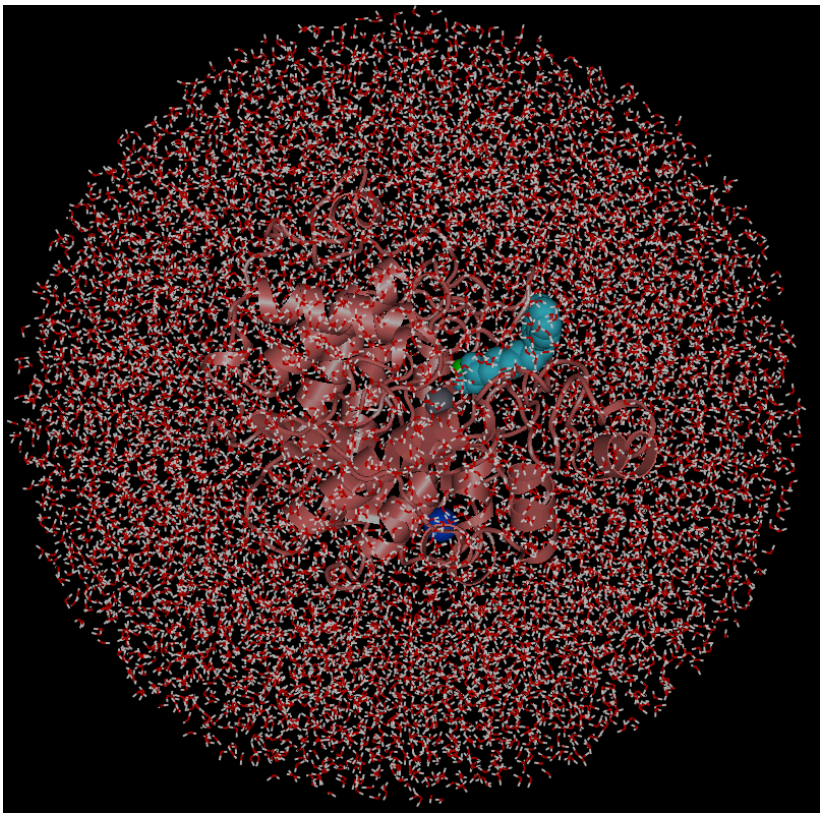

図 4. setwater で発生させた水分子を配置させた系

setwater で発生させた水分子を配置させた系

タンパク質・リガンド複合体と水分子からなる系

（水溶液中のイオンはまだ付加していない）

6.9. add_ion の実行

次に、add_ionを使って水溶液中にイオンを発生させます。制御用入力ファイルは、次のように記述します。

inp_add_ion の内容:

```
Pro_2.pdb
Pro_3.pdb
4
Y
6.0
0.2
```

このファイルも対話的実行時の入力項目をファイルに保存したものです。対話的実行の出力例を次に示します。各行に記述すべき内容は、この出力例の質問項目を参照してください。

add_ion の対話的実行の出力例:

```
--- addion version 2.002 2014/01/06 ---
 Biomolecule file (input file) name =
Pro_2.pdb     ←――     (キーボードからの入力項目)

 Biomolecule file (output file) name =
Pro_3.pdb     ←――     (キーボードからの入力項目)

 Biomolecule file (input): Pro_2.pdb
 Biomolecule file (output): Pro_3.pdb
 mode 1:direct inp, 2:minimum , 3 : density
  in case 3, 0.00277: normal saline solution
 4 : normal saline solution (dencity=0.00277)
4     ←――     (キーボードからの入力項目)
 Do you want to replace all water?(Y/N)
Y     ←――     (キーボードからの入力項目)

 Number of atoms (Biomolecule file) =       35623

 Number of water molecules =        9914
 Number of atoms in water =           3
 Total charge =   +2

 Number of Na+ ions to be added =          27
 Number of Cl- ions to be added =          29
 Total charge after couter-ion addition =   +0
 mol density of Na/Cl =   2.76999990E-03

 Exclusion distance =
6.0     ←――     (キーボードからの入力項目)

 Exclusion distance =   6.000
 probability of replacing water by ion(value=0.0-1.0: 0.2 is recommended)=
0.2
 probability value =  0.200000003


            1 ion added
(中略)
           35 ions added (random)
(中略)
           56 ions added (random)
```

add_ionの制御用入力ファイルの作成は以下のコマンドで行います(テキストエディタで作成するのでもかまいません。)

```
% echo Pro_2.pdb > inp_add_ion
% echo Pro_3.pdb >> inp_add_ion
% echo 4 >> inp_add_ion
% echo Y >> inp_add_ion
% echo 6.0 >> inp_add_ion
% echo 0.2 >> inp_add_ion

確認コマンド
% cat inp_add_ion
```

add_ionは、次のように実行します。

```
% ../bin/add_ion < inp_add_ion
```

add_ionの出力は、タンパク質(とリガンド)と水に加えて水中のイオンを含むPDBファイルです。

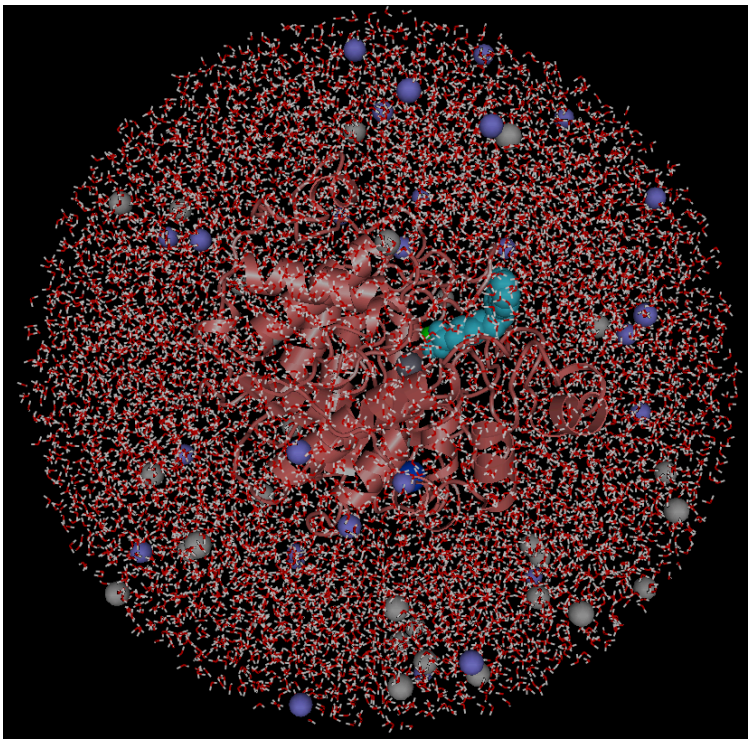

図 5. add_ion の出力ファイルを描画したもの

6.10. tplgeneX の実行(2 回目)

前節までで、タンパク質とリガンドと水とイオンを含む系のPDBファイルを準備できました。これは、MDで計算対象とする分子を全て含むものです。この系について、cosgeneで使用するためのトポロジーファイルを作成するために、tplgeneXを再度実行します。まず、次のように制御用入力ファイルを作成します。

inp_tplgeneX2 の内容:

```
pdb
C99
Pro_3.pdb
pdb
Pro_4.pdb
Pro_4.tpl
tip3p
```

このファイルも、tplgeneXを対話的に実行する場合の入力項目を記述したものです。このファイルは、1回目のtplgeneX実行のために作成したinp_tplgeneと比べて１行多くなっています。水分子が含まれている時には、次の質問が追加されるため、その回答が１つ多くなります。

```
%% Enter Water Model or the number. %%
   1: tip3p, 2: tip4p
1   ←――   （キーボードからの入力項目）
```

inp_tplgeneX2の別の記述例：

```
1
1
Pro_3.pdb
1
Pro_4.pdb
Pro_4.tpl
1
```

今回は低分子化合物を含む系についてのtplgeneXの実行ですので、実行時に低分子化合物のトポロジーファイルを読み込む必要があります。この制御用入力ファイルでは、低分子化合物のトポロジーファイルを指定していませんが、低分子化合物のトポロジーファイルはPro_3.pdbの中に記述されています。Pro_3.pdbに含まれる次の行です。

```
REMARK ORIGTPL  ./lig.tpl
```

本章の説明通りに、準備作業を行っている場合には問題ありませんが、途中から他のディレクトリで作業をする場合には、低分子化合物のトポロジーファイルもコピーしておかないとtplgeneXの実行時にエラーとなります。次のようなエラーメッセージが出力されます。

```
ERROR> outputLigandTplMolInfo
      File Open Error!
      Can not open Lignad topology File, "./lig.tpl".
```

上述の制御用入力ファイル(inp_tplgeneX2)を作成するコマンドは以下の通りです。

```
% echo pdb > inp_tplgeneX2
% echo C99 >> inp_tplgeneX2
% echo Pro_3.pdb >> inp_tplgeneX2
% echo pdb >> inp_tplgeneX2
% echo Pro_4.pdb >> inp_tplgeneX2
% echo Pro_4.tpl >> inp_tplgeneX2
% echo tip3p >> inp_tplgeneX2

確認コマンド：
% cat inp_tplgeneX2
```

次のコマンドで、2回目のtplgeneXを実行します。

```
% ../bin/exec_tplgeneX.sh < inp_tplgeneX2
```

tplgeneX の実行が完了したら、Pro_4.pdb と Pro_4.tpl が作成されていることを確認してください。次のコマンドを実行するとファイルサイズも同時に確認できます。

```
% ls -l
```

出力例:

```
-rw-r--r--  1 username  staff  2379322  7 27 03:06 Pro_4.pdb
-rw-r--r--  1 username  staff  4066743  7 27 03:06 Pro_4.tpl
```

ファイルサイズがゼロでない（もしくは小さすぎない）ことを確認してください。この例では、Pro_4.pdbが約2.4MB(2379322バイト)、Pro_4.tplが約4MB(4066743バイト)です。

　また、分子が適切に登録されているかを確認するために、Pro_4.tplの先頭から20行程度を確認するとよいでしょう。次のコマンドを実行するとPro_4.tplの先頭から20行を確認できます。

```
% head -20 Pro_4.tpl
```

headコマンドはファイルの冒頭からn行を出力するコマンドです。ここでは、第１引数でマイナス記号の後に続く数字で表示する行数を指定しています。”-20”の部分は適当に調整してください。

Pro_4.tpl の先頭部分:

```
TPL> TITLE
This tpl file made by tplgene 2015/07/27 03:06:42

TPL> MOLECULES
; NUMBER OF MOLECULES = 10037
pro                                                 1
Zn                                                  1
Ca                                                  1
Na1                                                 1
lig                                                 1
tip3p_water                                      9958
Cl                                                 29
Na                                                 27
```

TPL> MOLECULESから始まる部分を見ますと、pro 1となっているのはペプチド鎖が１本であることを示しています。Zn、Caが１個ずつ含まれています。Na1は、タンパク質内部に結合したナトリウムイオンです。下の方に出てくる水溶液中のカウンターイオンのNaと区別された表記になっています。lig 1の行から、化合物が１個であることが分かります。水分子(tip3pモデルの水分子)が9976 個、Cl(Cl-)が29個、Na(Na+)が27個含まれていることも分かります。setwaterでは10032個の水を用意しましたが、add_ionでイオンを付加した際に、イオンが置かれた場所に存在していた水分子は取り除かれていますので、水分子の数が減少しています。

6.11. tpl2capbc の実行

tpl2capbcは、球状に配置した水分子集団(CAP水)の境界情報を用意するコマンドです。同時に、ペプチド鎖の主鎖の重原子(CA, N, C, O)について位置束縛をするための入力ファイルを作成します。次の入力ファイル(inp_tpl2capbc)を用意して下さい。

inp_tpl2capbc の内容:

```
Pro_4.tpl
wat.pdb
capbc
M_all.res
```

これも、tpl2capbcを対話的に実行した際の質問項目への回答を記述したものです。

各行の意味は以下の通りです。

[1行目] 対象のトポロジーファイル名
[2行目] 水分子のみのPDBファイル名(setwaterの出力ファイル名)
[3行目] CAP水の境界情報の出力先ファイル名
[4行目] ペプチド鎖主鎖の重原子を位置束縛するための情報の出力先ファイル名

tpl2capbcを対話的に実行した画面を以下に示します。

tpl2capbc を対話的に実行した画面:

```
input TPL file name
Pro_4.tpl    <----    (キーボードからの入力項目)
 input WAT file name
wat.pdb    <----    (キーボードからの入力項目)
 input CAP file name
capbc    <----    (キーボードからの入力項目)
BOUND> INCLUDE
pro                          1          1 YES
 Zn                           1          1 YES
 Ca                           1          1 YES
 Na1                          1          1 YES
 lig                          1          1 YES
 tip3p_water                  1       9858 YES
 Cl                           1         29 YES
 Na                           1         27 YES

BOUND> CENTER
  COORDINATEs     15.0405      -21.1585       -2.6655
BOUND> RADIUS
     44.2508
input POS file name
M_all.res    <----    (キーボードからの入力項目)
GROUP> LIST
     1     4     1  2000 CA  *  1.0 MASS YES
     1     4     1  2000 N   *  1.0 MASS YES
     1     4     1  2000 C   *  1.0 MASS YES
     1     4     1  2000 O   *  1.0 MASS YES
END
GROUP> STOP
```

上述の制御用入力ファイル(inp_tpl2capbc)を作成するコマンドは以下の通りです。
(テキストエディタで作成してもかまいませんが、以下のものはテキストエディタを使わない作成方法です。)

```
% echo Pro_4.tpl > inp_tpl2capbc
% echo wat.pdb >> inp_tpl2capbc
% echo capbc >> inp_tpl2capbc
% echo M_all.res >> inp_tpl2capbc

確認コマンド：
% cat inp_tpl2capbc
```

tpl2capbcは、次のように実行します。

```
% ../bin/tpl2capbc < inp_tpl2capbc
```

これを実行すると、capbcとM_all.resが作成されます。それぞれの内容を以下に示します。このファイル名を、cosgeneの制御用入力ファイルで指定します。

capbc の内容:

```
BOUND> INCLUDE
 pro                          1           1 YES
 Zn                           1           1 YES
 Ca                           1           1 YES
 Na1                          1           1 YES
 lig                          1           1 YES
 tip3p_water                  1        9958 YES
 Cl                           1          29 YES
 Na                           1          27 YES

BOUND> CENTER
  COORDINATEs     15.0405      -21.1585       -2.6655
BOUND> RADIUS
     44.2508
```

M_all.res の内容:

```
GROUP> LIST
     1     4     1  2000 CA  *  1.0 MASS YES
     1     4     1  2000 N   *  1.0 MASS YES
     1     4     1  2000 C   *  1.0 MASS YES
     1     4     1  2000 O   *  1.0 MASS YES
END
GROUP> STOP
```

6.12. min.inp の作成

MD計算を実行する前には、まずエネルギー極小化計算を実行します。これは、系におけるひずみ（原子間隔が近すぎたり遠すぎたりしている状況）を補正するためです。

エネルギー極小化計算もcosgeneで行うことができます。ここでは、エネルギー極小化計算をcosgeneで行うための制御用入力ファイル(min.inp(ファイル名は任意))を作成します。cosgeneの制御用入力ファイルは行数が多いので、echoコマンドを打ち込んで実行することは困難です。テキストエディタで編集する方法が適切です。一から入力するのではなく、サンプルの入力ファイルを編集すると良いでしょう。cosgene_pack/sample/の下に、サンプルのmin.inpを用意していますので、ここでは、それをコピーして使用することにします。次のコマンドでコピーします。

```
% cp ../sample/min.inp .
```

min.inp の例:

```
    EXE> INPUT
         TOPOLOGY=   FORM     NAMETO=    Pro_4.tpl
         COORDINA=   PDB      NAMECO=    Pro_4.pdb
         REFCOORD=   PDB      NAMERE=    Pro_4.pdb
         POSITION=   READ     NAMEPO=    ./M_all.res
         SETBOU=     READ     NAMEBO=    ./capbc

         QUIT
    EXE> MINI
         METHOD=  STEEP       CPUTIM=     360000.0
         LOOPLI=  500     UPDATE=     20
         MONITO=  100     CONVGR=     0.001D0
         CUTMET=  RESA        CUTLEN=    10.0D0
         DIEFUN=  CONS        DIEVAL=     1.0D0
         CALPSR=  CALC        WETPSR=     1.00
         CALCAP=  CALC        FORCAP=    10.0

         QUIT
    EXE> OUTPUT
         COORDINATE=  PDB     NAMECO=   Pro_4_min.pdb
         QUIT
    EXE> END
```

読み込むファイルの名前が間違っていないかを確認してください。

計算時間に影響するパラメーターは、LOOPLI です。このサンプルファイルでは、テスト計算を短時間で完了するために、計算のステップ数(LOOPLI=500)を小さくしています。実際の計算では、適切な値を使ってください。適切な値は、系の大きさによっても異なります。最初は少し小さめ(LOOPLI=10000 程度)で実行してみて、 極小化の終盤でほとんどエネルギーの減少がほぼ無くなっていればそれで OK とし、そうでなければ LOOPLI を大きくしてから再計算をするとよいでしょう。MONITO で計算途中の情報(エネルギー等)をレポートする間隔を調整します。cosgene の出力か

ら、ステップ数とエネルギーの情報を抽出してグラフ化するとエネルギーの下がり具合を視覚的に確認がしやすいです。

6.13. cosgene の実行(エネルギー極小化)

cosgene制御用入力ファイルが準備できたら、次のコマンドでcosgeneを実行します。

```
% ../bin/cosgene < min.inp > min.out &
```

コマンドの末尾に&をつけると、バックグラウンドで計算を行います。cosgene を使った計算は比較的時間がかかります。min.out に計算の状況が逐次書き込まれていますので、計算が問題なく進んでいるか確認してください。次のコマンドで min.out の末尾を出力するといいでしょう。

```
% tail min.out
```

tail コマンドは、ファイルの末尾から n 行を出力するコマンドです。指定しない場合には末尾から 10 行を出力します。行数を指定する場合には、ハイフンに続いて行数を指定します。
行数を指定した例:

```
% tail -20 min.out
```

持続的にファイルの末尾を監視するには、次のコマンドを使います。

```
% tail -f min.out
```

このコマンドはファイルが更新される度に、ファイルの末尾を新しく出力します。このコマンドを終了するには、Ctrl+cを押してください。

計算の終了は、min.out の末尾に、以下のように”Job has finished.”が記録されているかどうかで判断できます。

```
 ++++++++++++++++++++++++++++++++++++++++
 +                                      +
 + INFORMATION> cosgene(13/01/17)       +
 +       Job has finished.              +
 +                                      +
 ++++++++++++++++++++++++++++++++++++++++


     CPU TIME FOR TOTAL    :     147.8713 (S)
```

計算の終了は、エネルギー極小化した構造の PDB ファイル(Pro_4_min.pdb, min.inp の中で指定)が作成されているかどうかでも判断できます。

6.14. SHAKEinp の実行

MD計算を効率的に実行するために、SHAKE法による原子束縛の設定情報を作成します。SHAKE法は、原子間距離が一定になるように拘束する計算方法です。ここでは、水素原子を含む結合だけをエネルギー極小化後の長さに拘束します。これにより、MD計算を効率的に実行できます。この設定情報を作成するためのプログラムはSHAKEinpです。SHAKEinpの制御用入力ファイルを以下の内容で用意します。

inp_shake の内容:

```
Pro_4.tpl
Pro_4_min.pdb
shake_inp
yes
```

このファイルも対話的実行をする場合の入力項目を記述したものです。SHAKEinp を対話的に実行した画面を以下に示します。実行画面を見ると分かるように inp_shake の各行の意味は以下の通りです。

[1行目] 読み込むトポロジーファイル名
[2行目] 読み込むPDBファイル名
[3行目] 出力ファイル名
[4行目] TIP3Pの水分子モデルを使用するか(yes/no)

SHAKEinp を対話的に実行した際の画面:

```
 ************************************************
 *                                              *
 *                   SHAKEinp                   *
 *                                              *
 *                 Mar 11, 2014                 *
 *                                              *
 ************************************************


 Please input TPL filename.
Pro_4.tpl

 Please input PDB filename.
Pro_4_min.pdb

 Please input SHAKE filename.
shake_inp

 INFORMATION>
        H2O was detected.
        Do you want to use TIP3P model?[yes/no]
yes

 INFORMATION> toolWriteTip3p
        The file "tip3_shk.model" does not exist in the current directory.
        Information given by the system is used for the Tip3p model.


 %% Program is done. %%
 %% This program is normal end. %%
```

上述の制御用入力ファイル(inp_tpl2capbc)を作成するコマンドは以下の通りです。
(テキストエディタで作成してもかまいませんが、以下のものはテキストエディタを使わない作成方法です。)

```
% echo Pro_4.tpl > inp_shake
% echo Pro_4_min.pdb >> inp_shake
% echo shape_inp >> inp_shake
% echo yes >> inp_shake

確認コマンド：
% cat inp_shake
```

SHAKEinpは次のコマンドで実行します。

```
% ../bin/SHAKEinp < inp_shake
```

6.15. md.inp の作成

前節までで、MD計算の準備がほとんど整いました。残された準備は、MD計算用の制御用インプットファイルの作成のみです。min.inpの作成と同様に、このインプットファイルも行数が多いので、サンプルファイルを修正して使用するとよいでしょう。

cosgene_pack/sample/にサンプルのmd.inpを用意してありますので、次のコマンドでコピーして使用します。

```
% cp ../sample/md.inp .
```

md.inp の内容:

```
;
    EXE> INPUT
         TOPOLOGY=   FORM     NAMETO=  Pro_4.tpl
         COORDINA=   PDB      NAMECO=  Pro_4_min.pdb
         REFCOORD=   PDB      NAMERE=  Pro_4.pdb
         SETSHAKE =  READ     NAMESH = shake_inp
         SETBOU=     READ     NAMEBO=  ./capbc
         POSITION=   READ     NAMEPO=  ./M_all.res
         QUIT
    EXE> MD
         LOOPLI=     500
         SETTIM=     500000.0D0   CPUTIM= 36000000.0D0
         UPDATE=     20
         TIMEST=     2.0D0
         OUTTRJ=     50
         OUTLOG=     50
         LOGFOR=     DETA

         METHOD=     CANONICAL
         SETTEM=     310.0D0
         INITIA=     SET
         STARTT=     310.0D0
         RANDOM=     254341
         SHAKEM=     HBON

         RESTARt=    NO
         NAMERO=     restart_1.rst

         MNTRCO=     SING
         OUTCOO=     5
         NAMECO=     traject_0.cor

         TEMPCO=     YES
         BESTFI=     YES

         CALPSR=   CALC  WETPSR=        0.1
         CALCAP=     CALC
         FORCAP=     10.0       ; FUKCAP = 150.0D0

         CUTMET=     RESA     CUTLEN=  12.0D0
         DIEFUN=     CONS     DIEVAL=  1.0D0

; FMM
         USEFMM=     YES
         FMMSPD= HIGH
         FMTREE=     4
         FMPOLE=     6
```

```
        CALCMM=     NOCALC
        CALV15=     CALC       CALE15=   CALC
        CALHYD=     NOCALC
        CALV5N=     NOCALC     CALE5N=   NOCALC
        CALH5N=     NOCALC

        QUIT
    EXE> MD
        LOOPLI=     500
        SETTIM=     500000.0D0   CPUTIM= 36000000.0D0
        UPDATE=     20
        TIMEST=     2.0D0
        OUTTRJ=     50
        OUTLOG=     50
        LOGFOR=     DETA

        METHOD=     CANONICAL
        SETTEM=     310.0D0
        INITIA=     SET
        STARTT=     310.0D0
        RANDOM=     654321
        SHAKEM=     HBON
        STOPCE=  BOTH

        RESTARt=    YES
        NAMERI=     restart_1.rst
        NAMERO=     restart_2.rst

        MNTRCO=     SING
        OUTCOO=     5
        NAMECO=     traject_1.cor

        BESTFI=     YES

        CALPSR=   CALC  WETPSR=      0.01
        CALCAP=     CALC
        FORCAP=     10.0

        CUTMET=     RESA     CUTLEN=  12.0D0
        DIEFUN=     CONS     DIEVAL=  1.0D0

; FMM
        USEFMM=     YES
        FMMSPD= HIGH
        FMTREE=     4
        FMPOLE=     6
        CALCMM=     NOCALC
        CALV15=     CALC       CALE15=   CALC
        CALHYD=     NOCALC
        CALV5N=     NOCALC     CALE5N=   NOCALC
        CALH5N=     NOCALC

        QUIT
    EXE> OUTPUT
        COORDINATE=  PDB     NAMECO=   Pro_4_md_1.pdb
        QUIT
    EXE> END
exit
```

md.inpでは、EXE> MDのブロックが２つあります。このように、１回のMD計算の実行で、２つの異なる計算条件を設定することができます。エネルギー極小化をしていても、MD開始直後には、まだひずみが残っていて分子の挙動が非常に不安定で異常な振る舞いをする可能性があります。第１のMD計算では、主にこのひずみを取ることを目的としたMD計算を行います。このような計算を緩和計算/平衡化計算と呼びます。十分に系を緩和させた後で、第2のMD計算を実行して、この結果を解析対象とします。

```
LOOPLI:(Loop limit) MD シミュレーションのループ回数
SETTIM:(Set time limit) シミュレーション時間の上限(ps)
CPUTIM:(CPU time limit) CPU 時間での上限(秒)
UPDATE: 相互作用テーブルの更新頻度
TIMEST: (Time step) タイムステップ(fs)
OUTTRJ: トラジェクトリを保存するステップ間隔
OUTLOG:計算結果出力をする MD ステップの回数
LOGFOR:(Log format) 出力形式(SHOR(Short), DETA(Detail))

METHOD:(Method) アンサンブルの発生方法 (MIC: Micro-canonical,CANO: Canonical 等)
SETTEM:(Set temperature) 系の目標温度(K)
INITIA:(Initial velocity) 初期速度の設定に仕方(ZERO, SET)
STARTT:(Start temperature) 初期温度の平均値(K)
RANDOM:(Random seed) 速度分布を得るための乱数のシード

RESTAR:(Restart) リスタートの設定(YES, NO)
NAMERI: 読み込むリスタートファイルの名前
NAMERO: 書き出すリスタートファイルの名前

MNTRCO:(Monitor coordinate) 座標トラジェクトファイルの形式
OUTCOO:(Output coordinate) 座標トラジェクトリの出力タイミング
NAMECO:(Name coordinate) 座標トラジェクトリのファイル名
```

cosgene制御用入力ファイルの記述方法については、”cosgene USER MANUAL”に詳しく書かれていますので、そちらを参照してください。

6.16. cosgene の実行(MD)

次のコマンドでMD計算を開始します。

```
% ../bin/cosgene < md.inp > md.out &
```

極小化の時と同様に、tailコマンドを使って計算が問題なく進んでいるかチェックしてください。MD 計算の進行状況を確認するのには、次のコマンドも役に立ちます。

```
% grep LOOP md.out
```

grep の出力例(計算が完了した例):

```
          LOOP LIMIT      :          500
          LOOP LIMIT    :         1000
         HEATLOOP STEPS  :            0
             MOLECULAR DYNAMICS LOOP START
 MD LOOP NUMBER        :              50 TIME (PSEC)            :          0.10000
 MD LOOP NUMBER        :             100 TIME (PSEC)            :          0.20000
 MD LOOP NUMBER        :             150 TIME (PSEC)            :          0.30000
 MD LOOP NUMBER        :             200 TIME (PSEC)            :          0.40000
 MD LOOP NUMBER        :             250 TIME (PSEC)            :          0.50000
 MD LOOP NUMBER        :             300 TIME (PSEC)            :          0.60000
 MD LOOP NUMBER        :             350 TIME (PSEC)            :          0.70000
 MD LOOP NUMBER        :             400 TIME (PSEC)            :          0.80000
 MD LOOP NUMBER        :             450 TIME (PSEC)            :          0.90000
 MD LOOP NUMBER        :             500 TIME (PSEC)            :          1.00000
 MD LOOP NUMBER        :             501 TIME (PSEC)            :          1.00200
             MD LOOP           :       296.25635
         LOOP LIMIT      :          500
          LOOP LIMIT    :         1000
         HEATLOOP STEPS  :            0
    LAST MD LOOP          :          500
 MD LOOP NUMBER        :             501 TIME (PSEC)            :          1.00200
             MOLECULAR DYNAMICS LOOP START
 MD LOOP NUMBER        :             550 TIME (PSEC)            :          1.10000
 MD LOOP NUMBER        :             600 TIME (PSEC)            :          1.20000
 MD LOOP NUMBER        :             650 TIME (PSEC)            :          1.30000
 MD LOOP NUMBER        :             700 TIME (PSEC)            :          1.40000
 MD LOOP NUMBER        :             750 TIME (PSEC)            :          1.50000
 MD LOOP NUMBER        :             800 TIME (PSEC)            :          1.60000
 MD LOOP NUMBER        :             850 TIME (PSEC)            :          1.70000
 MD LOOP NUMBER        :             900 TIME (PSEC)            :          1.80000
 MD LOOP NUMBER        :             950 TIME (PSEC)            :          1.90000
 MD LOOP NUMBER        :            1000 TIME (PSEC)            :          2.00000
 MD LOOP NUMBER        :            1001 TIME (PSEC)            :          2.00200
             MD LOOP           :       313.83679
```

6.17. 解析

MD計算が終了したら、MDのトラジェクトリーについて解析を行います。解析については、” cosgene USER MANUAL”を参照してください。

7. 発展

本マニュアルでは、基本的なMD計算の準備方法・計算開始方法について説明しました。実際に、より長いMD計算を実行するためには、複数の計算機コアを同時に使用して高速に計算を実行するcosgene_MPIを使用します(もしくは、GPUを使うpsygene-Gを使用します。cosgene_MPIのコンパイル、および、実行方法は、計算機環境に依存します。cosgene_MPIを使用する場合でも、入力ファイルは同じですので、入力ファイルの準備およびテストは、本マニュアルの手順に沿って行うと良いと思います。LSFやSun Grid Engine等のジョブスケジューラーを使用する環境では計算ジョブの投入方法についても確認する必要があります。計算ジョブの投入方法は、計算機環境毎に異なりますので、計算ジョブ投入方法を確認するには、計算機システムの使用マニュアルを参照する、もしくは、管理者に聞く必要があります。

8. ツールプログラムについての説明

cosgene_pack に含まれるツールプログラムの一部について説明します。cosgene, tplgeneX, tplgeneL, Hgene, pdbcheck については、それぞれのマニュアルがありますので、そちらを参照してください。

- get_pdb_info.pl

  このプログラムは、PDB ファイルに含まれる分子についての情報を出力します。ドッキングに使用する分子、使用しない分子を選択する際に参考にします。引数で与えた PDB ファイルを解析し、以下の内容をレポートします。

  - ヘッダーにおける SSBOND 行
  - ペプチド鎖の数と各 ID
  - HETATM から始まる行に出現する残基名とその種類の数

  使用方法:

```
% (パス)/get_pdb_info.pl  (pdb ファイル名)
```

  使用例(P10 参照):

```
% ../bin/get_pdb_info.pl 4HP0.pdb
```

  出力例:

```
SSBOND information:
SSBOND   1 CYS A    6    CYS A  127                          1555   1555  2.04
SSBOND   2 CYS A   30    CYS A  115                          1555   1555  2.07
SSBOND   3 CYS A   64    CYS A   80                          1555   1555  2.05
SSBOND   4 CYS A   76    CYS A   94                          1555   1555  2.04

No of peptide chain:1
Chain ID 1: A

No of ligand name:3
Ligand name 1: NOJ
Ligand name 2: NAG
Ligand name 3: HOH
```

  このレポートから、以下のことが分かります。4HP0.pdb には、

  - SSBOND 行も含まれている
  - ペプチド鎖は A 鎖のみ
  - HETATM 行から始まる行に登場する残基名の種類は 3 種類で、それらは、NOJ, NAG, HOH

- select_chain.pl

  このプログラムは、PDB ファイルの中から特定の鎖 ID を持つものを標準出力に出力します。

  使用方法:

```
% select_chain.pl (鎖の ID) (入力 PDB ファイル名) > (出力 PDB ファイル名)
```

  使用例(P12 参照):

```
% ../bin/select_chain.pl A 2PU2.pdb > 2PU2_1.pdb
```

- select_res.pl

  このプログラムは、PDB ファイルの中から特定の残基名のものを標準出力へ出力します。

  使用方法:

```
% select_res.pl (残基名に記述された ID) (入力 PDB ファイル名) > (出力 PDB ファイル名)
```

  使用例(P12 参照):

```
% ../bin/select_res.pl DK2 2PU2.pdb > point.pdb
```

- del_res.pl

  このプログラムは、PDB ファイルの中から特定の残基名のものを削除し、残りを標準出力へ出力します。

  使用方法:

```
% del_res.pl (残基名に記述された ID) (入力 PDB ファイル名) > (出力 PDB ファイル名)
```

  使用例(P12 参照):

```
% ../bin/del_res.pl DK2 2PU2.pdb > tmpDel_DK2_2PU2.pdb
```

- exec_tplgeneX.sh(P.21 参照)

  このプログラムは、tplgeneX を実行するためのプログラムで、tplgeneX を実行する前に、環境変数 TPL_DB_PATH を、このスクリプトプログラム内で設定します。プログラムの中身は以下の通りです。

  exec_tplgeneX.sh:

```
#!/bin/bash
# 2018/1/18

tplgeneX=tplgeneX180118

DIR=$(cd $(dirname $0); cd ..; pwd)
echo $DIR
export PATH=$DIR/bin:$PATH
export TPL_DB_PATH=$DIR/src/${tplgeneX}/tplgeneX/DB
echo "TPL_DB_PATH:$TPL_DB_PATH"
which tplgeneX
tplgeneX
```

- exec_tplgeneL.sh(P.18 参照)

  このプログラムは、tplgeneL を実行するためのプログラムで、tplgeneL を実行する前に、環境変数 TPLL_DB_PATH を、このスクリプトプログラム内で設定します。プログラムの中身は以下の通りです。

  exec_tplgeneL.sh:

```
#!/bin/bash
# 2020/8/12

tplgeneL=tplgeneL200118

DIR=$(cd $(dirname $0); cd ..; pwd)
echo $DIR
export PATH=$DIR/bin:$PATH
export TPLL_DB_PATH=$DIR/src/${tplgeneL}/src/tplgeneL/tplgeneL/DB
echo "TPLL_DB_PATH:$TPLL_DB_PATH"
which tplgeneL
tplgeneL
```

9. 参照文献

[1] 神谷成敏・肥後順一・福西快文・中村春木, タンパク質計算科学 ―基礎と創薬への応用―, 共立出版, 2009 年 8 月
[2] PDB チェックツール設計書.doc
[3] PDB チェックツール操作説明書.doc