

myPresto 5.0

- *sievgene* -

TUTORIAL

2018/01/12

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. 本チュートリアルの概要..... | 1 |
| 2. インストール方法..... | 2 |
| 3. ディレクトリ構成..... | 4 |
| 4. 動作検証用テストプログラムの実行 | 5 |
| 5. ドッキングシミュレーションに必要なファイル | 10 |
| 6. ドッキングシミュレーションの手順 | 11 |
| 7. 手順の具体例..... | 12 |
| 7.1. リゾチーム(4HP0)の例..... | 12 |
| 7.2. AmpC(2PU2)の例..... | 19 |
| 7.3. Factor Xa(2W26)の例 | 21 |
| 7.4. トロンビン(2PKS)の例..... | 23 |
| 8. ツールプログラムについての説明 | 28 |
| 9. リファレンス..... | 31 |

1. 本チュートリアル の概要

本チュートリアルは、`sievgene_pack` に含まれるプログラム、および、サンプルデータを使用して、タンパク質と低分子化合物との最適ドッキングポーズを探索・評価する手順について説明したものです。`sievgene_pack` は、`sievgene` 実行のために必要なツールプログラムをセットにしたものです。

`sievgene_pack` には、以下のプログラムが含まれています。

- `sievgene_for_dockingpose` (ドッキングポーズを精度良く計算するための `sievgene`)
- `sievgene_for_screening` (スクリーニング用の `sievgene`)
- `tplgeneX` (タンパク質、核酸等の高分子用)
- `tplgeneL` (低分子化合物用)
- `Hgene` (化合物への水素原子付加、原子の部分電荷計算など)
- `make_point` (ドッキングの範囲を指定するためのプローブ点を発生させるプログラム)
- `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)
 - `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)
- インストーラープログラム
 - `install.sh` (インストーラープログラム)
- テスト計算自動実行プログラム
 - `test_4HP0.sh` (`4HP0.pdb` を使ったドッキングの自動実行プログラム)
 - `test_2W26.sh` (`2W26.pdb` を使ったドッキングの自動実行プログラム)
 - `test_2PU2.sh` (`2PU2.pdb` を使ったドッキングの自動実行プログラム)
 - `test_2PKS.sh` (`2PKS.pdb` を使ったドッキングの自動実行プログラム)
 - `test_all.sh` (`test_4HP0.sh`, `test_2W26.sh`, `test_2PU2.sh`, `test_2PKS.sh` を自動実行して、結果をまとめて出力するプログラム)

`sievgene_pack` に含まれているプログラムは以上です。

2. インストール方法

sievgene_pack は、sievgene_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 環境で、個人的に使用することを想定しています。つまり、ユーザーのホームディレクトリの下にインストールして、自分のみで使用する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。ホームディレクトリの下どこかに、sievgene_packYYMMDD.tar.gz を配置して、次のコマンドで展開してください。

```
% tar -xzvf sievgene_packYYMMDD.tar.gz
```

展開後に sievgene_packYYMMDD という名前のディレクトリができます。以下のコマンドを実行すると、インストールを開始します。終了まで少し時間がかかります。

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

引数なしで install.sh を実行した場合には、ifort を使わずに、プログラムをコンパイルします。次のように”intel”を引数に与えると ifort を使ってプログラムをコンパイルします。

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

sievgene_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。

- sievgene_dk
- sievgene_sc
- tplgeneX
- tplgeneL
- Hgene
- pdbcheck
- make_point

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。

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。

- `sievgene_for_dockingpose(sievgene_dk)`: gfortran
- `sievgene_for_screening(sievgene_sc)`: gfortran
- `tplgeneX`: gcc
- `tplgene`: gcc
- `tplgeneL`: /usr/bin/cc
- `Hgene`: gfortran, gcc
- `pdbcheck`: gfortran
- `make_point`: gfortran

“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。

- `sievgene_for_dockingpose(sievgene_dk)`: ifort
- `sievgene_for_screening(sievgene_sc)`: ifort
- `tplgeneX`: gcc
- `tplgene`: gcc
- `tplgeneL`: /usr/bin/cc
- `Hgene`: ifort, icc
- `pdbcheck`: ifort
- `make_point`: ifort

3. ディレクトリ構成

sievgene_pack のディレクトリ構成は以下のようになっています。

sievgene_pack のディレクトリ構成:

```
sievgene_packYYMMDD/
--OREADME
--bin/
  --install.sh
  --check_binary.sh
  --clean_binary.sh
  --exec_tplgeneX.sh
  --exec_tplgene.sh
  --exec_tplgeneL.sh
  --get_pdb_info.pl
  --select_chain.pl
  --select_res.pl
  --test_4HPO.sh
  --test_2W26.sh
  --test_2PU2.sh
  --test_2PKS.sh
  --test_all.sh
  --(sievgene_dk) (install.sh 実行後に出現)
  --(sievgene_sc) (install.sh 実行後に出現)
  --(tplgeneX) (install.sh 実行後に出現)
  --(tplgene) (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)
  --(make_point) (install.sh 実行後に出現)
--input/
  --inp_sievgene1
--src/
  --Makefile
  --tplgeneX/
  --tplgene/
  --tplgeneL/
  --Hgene/
  --pdbcheck/
  --make_point/
  --sievgene/
--doc/
  --sievgene_manual_v4.400_ja.pdf
  --sievgene_tutorial_v4.400_ja.pdf (このドキュメント)
  --tplgeneX_manual_v4.400_ja.pdf
  --Hgene_manual_v4.400_ja.pdf
  --pdbcheck_manual_v4.400_ja.pdf
--sample/
  --4HPO.pdb
  --2W26.pdb
  --2PU2.pdb
  --2PKS.pdb
  --2PKS_mod.pdb
  --c001-1.mol2
--sample2/
  --allrun.sh
  --all_for_2010.inp
  --pro_list
  --1aj5/
  --1c1e/
```

4. 動作検証用テストプログラムの実行

インストールに成功したら、動作検証用テスト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タンパク質と低分子化合物に対して、準備からドッキング計算までを自動的に実行します。sievgene_pack には、test_4HP0.sh, test_2PKS.sh, test_2W26.sh, test_2PU2.sh の4つのテスト計算プログラムと、それらを全て実行する test_all.sh が用意されています。

まず、test_4HP0.sh を実行してみましょう。sievgene_pack/の下で、次のコマンドで実行してください。

```
% bin/test_4HP0.sh
```

test_4HP0.sh を引数なしで実行すると sievgene_dk を使ったドッキング計算を実行します。このプログラムが最後まで問題なく実行されれば、4HP0.pdb のタンパク質に対して、1つの化合物(c001_1.mol2)をドッキングさせた計算が終了しています。ドッキング計算の結果は、sievgene_packYYMMDD/dock_4HP0/の下に保存されています。ex.score にドッキングスコアが、ex.mol2 にドッキングポーズが出力されています。次のように、screening を引数に与えて test_4HP0.sh を実行すると sievgene_sc を使ったドッキング計算を実行します。

```
% bin/test_4HP0.sh screening
```

dock_4HP0/ex.score の内容(GNU コンパイラ使用, sievgene_dk, MacBookPro):

```
INFORMATION> BEFORE MINIMIZE RANKING :EXPERIMENT ID=1
INFORMATION> LOCAL SCORE RANKING          5

      SCORE(/100)  dG    LE/10  NHA  rotNum   ASA    ELE    HYD    VDW    SURFACE    RMSD
1    -1.8835  -5.17  -1.35  14   0   -170.89  1.18  -18.64  0.00  0.00  24.95
2    -1.8627  -5.08  -1.33  14   0   -170.24  0.28  -16.31  0.00  0.00  25.30
3    -1.8387  -4.94  -1.31  14   0   -170.54  -2.02  -11.31  0.00  0.00  25.95
4    -1.8378  -5.06  -1.31  14   0   -163.36  -1.26  -19.16  0.00  0.00  25.57
5    -1.8372  -5.05  -1.31  14   0   -166.38  1.47  -18.80  0.00  0.00  25.11

REFERENCE COORDINATE DATA
999.90  999.90  0.00  0.00

INFORMATION> AFTER MINIMIZE RANKING :EXPERIMENT ID=1
INFORMATION> LOCAL SCORE RANKING          5

      COMPOUND NAME : HTS1404-00000229-01
      FILE NAME      : ligand.mol2

      SCORE(/100)  dG    LE/10  NHA  rotNum   ASA    ELE    HYD    VDW    SURFACE    RMSD
@ 1    -1.9406  -5.30  -1.39  14   0   -177.04  2.37  -18.70  -0.69  68.30  24.98
@ 2    -1.8983  -5.21  -1.36  14   0   -172.38  1.18  -18.64  0.00  68.26  24.95
@ 3    -1.8974  -5.09  -1.36  14   0   -177.17  -0.64  -11.43  -0.50  66.48  25.98
@ 4    -1.8839  -5.15  -1.35  14   0   -170.71  1.69  -18.54  -0.82  69.56  25.22
@ 5    -1.8771  -5.12  -1.34  14   0   -171.68  0.28  -16.31  0.00  66.50  25.30

REFERENCE COORDINATE DATA
999.90  999.90  0.00  0.00

INFORMATION> COMPOUND RANKING          1

      COMPOUND NAME          SCORE(/100)  dG    LE(/10)    NHA
1 HTS1404-00000229-01      -1.94    -5.30  -1.39      14

INFORMATION> LIGAND EFFICIENCY RANKING

      COMPOUND NAME          LE(/10)
1 HTS1404-00000229-01      -1.39
```

sievgene を使った計算では乱数を使っています。乱数列は、同じシードで初期化していますので、プログラムのコンパイルに同じコンパイラを使用し、同じ計算機環境で計算を実行した場合は、同じ結果が得られます。しかし、違うコンパイラを使って実行ファイルを作成した場合や、計算機環境が違う場合には、同じ結果になりません。

次の結果は、intel コンパイラを使用してプログラムをコンパイルした場合の結果です。
dock_4HP0/ex.score の内容(intel コンパイラ使用, sievgene_dk, MacBookPro):

```
INFORMATION> BEFORE MINIMIZE RANKING :EXPERIMENT ID=1
INFORMATION> LOCAL SCORE RANKING          5
```

| | SCORE(/100) | dG | LE/10 | NHA | rotNum | ASA | ELE | HYD | VDW | SURFACE | RMSD |
|---|-------------|-------|-------|-----|--------|---------|-------|--------|------|---------|-------|
| 1 | -1.8835 | -5.17 | -1.35 | 14 | 0 | -170.89 | 1.18 | -18.64 | 0.00 | 0.00 | 24.95 |
| 2 | -1.8627 | -5.08 | -1.33 | 14 | 0 | -170.24 | 0.28 | -16.31 | 0.00 | 0.00 | 25.30 |
| 3 | -1.8387 | -4.94 | -1.31 | 14 | 0 | -170.54 | -2.02 | -11.31 | 0.00 | 0.00 | 25.95 |
| 4 | -1.8378 | -5.06 | -1.31 | 14 | 0 | -163.36 | -1.26 | -19.16 | 0.00 | 0.00 | 25.57 |
| 5 | -1.8371 | -5.05 | -1.31 | 14 | 0 | -166.38 | 1.47 | -18.80 | 0.00 | 0.00 | 25.11 |

```
REFERENCE COORDINATE DATA
999.90 999.90 0.00 0.00
```

```
INFORMATION> AFTER MINIMIZE RANKING :EXPERIMENT ID=1
INFORMATION> LOCAL SCORE RANKING          5
```

```
COMPOUND NAME : HTS1404-00000229-01
FILE NAME      : ligand.mol2
```

| | SCORE(/100) | dG | LE/10 | NHA | rotNum | ASA | ELE | HYD | VDW | SURFACE | RMSD |
|-----|-------------|-------|-------|-----|--------|---------|-------|--------|-------|---------|-------|
| @ 1 | -1.9702 | -5.44 | -1.41 | 14 | 0 | -173.67 | 0.10 | -22.87 | -0.59 | 69.84 | 24.91 |
| @ 2 | -1.9406 | -5.30 | -1.39 | 14 | 0 | -177.04 | 2.37 | -18.70 | -0.69 | 68.30 | 24.98 |
| @ 3 | -1.9054 | -5.11 | -1.36 | 14 | 0 | -177.93 | -0.41 | -11.70 | -0.50 | 63.66 | 25.96 |
| @ 4 | -1.8983 | -5.21 | -1.36 | 14 | 0 | -172.38 | 1.18 | -18.64 | 0.00 | 68.26 | 24.95 |
| @ 5 | -1.8974 | -5.09 | -1.36 | 14 | 0 | -177.17 | -0.64 | -11.43 | -0.50 | 66.48 | 25.98 |

```
REFERENCE COORDINATE DATA
999.90 999.90 0.00 0.00
```

```
INFORMATION> COMPOUND RANKING          1
```

| COMPOUND NAME | SCORE(/100) | dG | LE(/10) | NHA |
|-----------------------|-------------|-------|---------|-----|
| 1 HTS1404-00000229-01 | -1.97 | -5.44 | -1.41 | 14 |

```
INFORMATION> LIGAND EFFICIENCY RANKING
```

| COMPOUND NAME | LE(/10) |
|-----------------------|---------|
| 1 HTS1404-00000229-01 | -1.41 |

dock_4HP0/ex.mol2 の一部(GNU コンパイラ使用, sievgene_dk, MacBookPro):

```
# HTS1404-00000229-01      1 of 5
#                               SCORE = -194.0596
#                               dG-SCORE = -5.3039
#                               LIGAND_EFFICIENCY = -1.3861
# NUMBER_OF_HEAVY_ATOMS = 14
#                               ASA = -177.0405
#                               ELE = 2.3682
#                               HYD = -18.6965
#                               VDW = -0.6909
#                               RMSD = 24.9758
@<TRIPOS>MOLECULE
HTS1404-00000229-01
  21  22  0  0  0
SMALL
USER_CHARGES
@<TRIPOS>ATOM
  1 C      23.5598   5.9190  -4.0341 C. ar      1 UNK      -0.0048
  2 C      23.3936   5.8692  -2.6965 C. ar      1 UNK      -0.3298
  3 H      22.8935   5.1190  -2.1174 H          1 UNK       0.1575
  4 N      24.2111   7.0221  -4.2947 N. ar      1 UNK     -0.1297
  5 C      23.1520   5.0239  -4.9834 C. ar      1 UNK       0.0490
  6 C      23.9967   7.0849  -2.1762 C. ar      1 UNK       0.1235
  7 N      24.5135   7.7602  -3.2014 N. ar      1 UNK     -0.1664
  8 H      24.9908   8.6350  -3.2186 H          1 UNK       0.2883
  9 C      23.3570   5.0950  -6.4088 C. ar      1 UNK     -0.0999
 10 C      22.4806   3.9085  -4.5311 C. ar      1 UNK     -0.1017
 11 H      22.2979   3.7345  -3.4799 H          1 UNK       0.1449
 12 C      22.9053   4.0670  -7.1917 C. ar      1 UNK     -0.1208
 13 H      23.0935   4.0982  -8.2680 H          1 UNK       0.1467
 14 N      24.0280   7.5505  -0.9043 N. pl3     1 UNK     -0.4163
 15 H      23.5531   7.0959  -0.1620 H          1 UNK       0.2313
 16 H      24.4832   8.3973  -0.6380 H          1 UNK       0.2313
 17 C      21.9876   2.9327  -5.3270 C. ar      1 UNK     -0.1425
 18 Br     24.2342   6.3883  -7.5257 Br         1 UNK       0.0564
 19 C      22.2504   3.0105  -6.6665 C. ar      1 UNK     -0.1023
 20 H      21.9316   2.2078  -7.2979 H          1 UNK       0.1436
 21 Br     20.9708   1.5495  -4.5649 Br         1 UNK       0.0417
@<TRIPOS>BOND
  1 1 2 ar
  2 1 4 ar
  3 1 5 1
  4 2 6 ar
  5 4 7 ar
  6 5 9 ar
  7 5 10 ar
  8 9 12 ar
  9 6 14 1
 10 10 17 ar
 11 9 18 1
 12 17 19 ar
 13 17 21 1
 14 6 7 ar
 15 12 19 ar
 16 2 3 1
 17 7 8 1
 18 10 11 1
 19 12 13 1
 20 14 15 1
 21 14 16 1
 22 19 20 1
#MOLECULE END
```

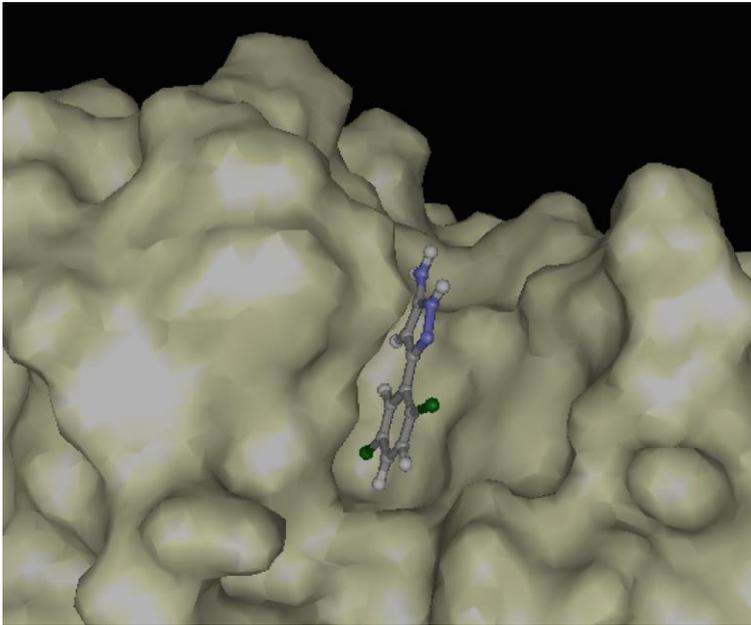


図 1. 得られたドッキングポーズ

ex.mol2 に記録された5つのドッキングポーズのうち、スコアが最も低かったもののみを表示しています。緑色の原子は臭素です。(座標は前ページのもので)

同様の動作テストプログラムが他に 3 つ用意されています。

```
% test_2W26.sh
% test_2PU2.sh
% test_2PKS.sh
```

test_4HP0.sh と同様に screening というキーワードを引数に与えると、sievgene_sc を使います。引数が無い場合には、sievgene_dk を使います。

また、上述の 4 つのテストプログラム全てを自動的に逐次実行し、結果をまとめて出力するプログラムは、以下のコマンドです。

```
% bin/test_all.sh (sievgene_dk を使用する場合)
% bin/test_all.sh screening (sievgene_sc を使用する場合)
```

5. ドッキングシミュレーションに必要なファイル

sievgene でドッキングシミュレーションを行うためには、以下の入力ファイルを用意する必要があります。括弧内は、計算用サンプルファイルとしてよく使用しているファイル名の例です。

- | | |
|--------------------------|-----------------|
| (1) タンパク質の座標ファイル | (例: Pro_md.pdb) |
| (2) タンパク質のトポロジーファイル | (例: Pro.tpl) |
| (3) ターゲットサイト指定用プローブ点ファイル | (例: point.pdb) |
| (4) リガンドの立体構造情報ファイル | (例: c001.mol2) |
| (5) 制御ファイル | (例: s1.inp) |

これらのファイルを準備する方法について説明します。

(1)と(2)の作成には、**tplgeneX** を使用します。(1)と(2)は **tplgeneX** の出力ファイルとして得られます。これらを得るためには、**tplgeneX** の入力ファイルを用意する必要があります。**tplgeneX** の入力ファイルを作成するためには、ターゲットタンパク質の PDB ファイルを加工する必要があります。この加工を手助けするプログラムは **pdbcheck** です。

(3)のファイルを用意する方法はいくつかあります。既知活性化化合物とターゲットタンパク質との複合体構造が利用可能な場合には、複合体構造の既知活性化化合物の座標のみを取り出して、それをプローブ点ファイルとして使用することができます。また、そうしたリガンドが無い場合には、**make_point** というプログラムを使ってプローブ点を発生することができます。

(4)は、自分で用意する方法もありますが、ドッキング用に用意された **MOL2** ファイルを入手して使用するのが良いでしょう。

(5)のファイルには、(1)~(4) のファイル名とシミュレーションパラメーターを記述します。これについては、サンプルの制御ファイルを編集し、ファイル名等を変更して使用すると良いでしょう。**sievgene_packYYMMDD/input/**の下に、**inp_sievgene1** という名前のサンプルの制御ファイルを用意しています。

6. ドッキングシミュレーションの手順

5.で説明したファイルの用意(図 2)を含めて、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を決定します。

1. ターゲットサイト付近において、リガンドが結合するのに排除体積効果から障害となる分子は、ドッキングの妨げになりますので削除します。
2. ターゲットサイトから比較的遠くに位置して、ターゲットサイトにおけるリガンドの結合に影響が少ないと考えられる分子(他のユニットに属するタンパク質鎖、溶媒中の水分子やイオン等)は、計算する際には、削除しても問題ないと思われま
3. タンパク質内部に結合している金属イオンは、ターゲットタンパク質の一部とみなして残した方がよいでしょう。金属イオンは比較的大きな電荷を持ち、比較的長距離に渡って影響を及ぼすので重要だと考えられます。

※get_pdb_info.pl の出力および分子ビューアでの表示も参考にするとよいでしょう。

- (3) オリジナルの PDB ファイルを加工して、ドッキングで使用しない分子を削除します。
※select_chain.pl や Unix の grep コマンドを使用するといいかもかもしれません。

- (4) pdbcheck で PDB ファイルを処理します。
※pdbcheck の実行には、制御ファイルの作成が必要です。

- (5) pdbcheck の出力ファイルとして得られた PDB ファイルを入力ファイルとして、tplgeneX を実行します。tplgeneX の出力ファイルとして、sievgene で使用可能な PDB ファイルとそれに対応したトポロジーファイルが得られます。
※tplgeneX の実行には、制御ファイルの作成が必要です。(対話的にも実行できます。)

- (6) sievgene で使用するターゲットサイト指定用プローブ点ファイルを作成します。このファイルは、複数の方法で作成できます。1つの方法は、オリジナルの PDB ファイルに含まれる阻害剤の座標を利用する方法で、他には、make_point で作成する方法、myPresto 用 GUI ソフト(MF myPresto、MolDesk)を使って、GUI 上でマウス操作によって作成する方法等があります。

- (7) ドッキングさせるリガンドの MOL2 ファイルを用意します。

- (8) sievgene の制御ファイルを作成します。

- (9) sievgene を実行します。

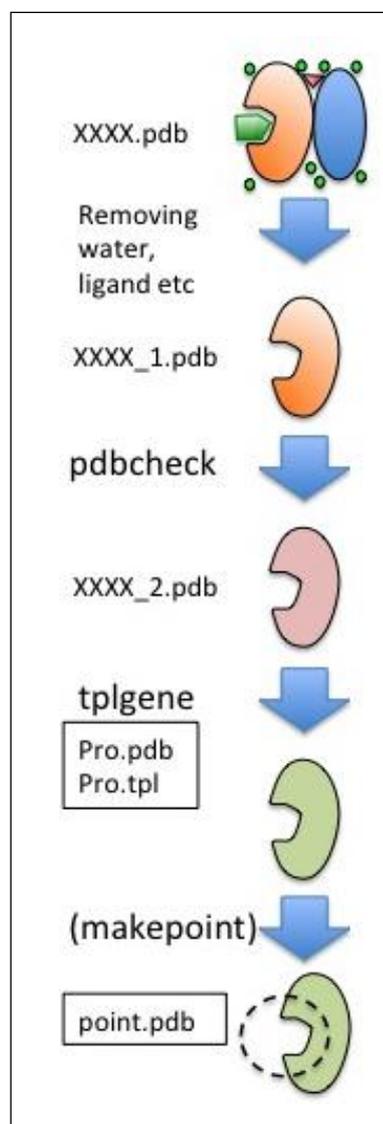


図 2. ドッキング用入力ファイルの準備プロセス

7. 手順の具体例

7.1. リゾチーム(4HP0)の例

リゾチーム(4HP0)と低分子化合物とのドッキングは以下の手順で実行します。まず、`sievgene_packYYMMDD.tar.gz` を展開したディレクトリに移動してから、以下のコマンドを実行してください。

```
% cd sievgene_packYYMMDD                (移動)
% mkdir tmp_4HP0                          (作業用ディレクトリの作成)
% cd tmp_4HP0                              (作業用ディレクトリへの移動)
% cp ../sample/4HP0.pdb .                 (オリジナル PDB ファイルの用意)
% ../bin/get_pdb_info.pl 4HP0.pdb        (オリジナル PDB ファイルの内容確認)
```

`get_pdb_info.pl` の出力例は以下のようになります。

```
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
```

この出力を見ることにより、`4HP0.pdb` には、`SSBOND` から始まる行があり、ペプチド鎖は A 鎖の 1 本のみ、ペプチド鎖以外では、`NOJ`、`HAG`、`HOH` が含まれていることが分かります。この例では、リガンドは、`NOJ-NAG-NAG-NAG` で 1 分子となっています。つまり、オリジナルの PDB ファイルには、ペプチド鎖が 1 本、リガンドが 1 つ、水分子が多数含まれています。この例では、ペプチド鎖のみを残し、リガンドと水分子を削除します。`pdbcheck` の制御ファイルで `-disableHet` オプションをつけることにより、`HETATM` で始まる行を全て削除することができます。(`pdbcheck` の使い方の詳細は、`sievgene_packYYMMDD/doc/` の下にある `pdbcheck_manual_v4.400.pdf` を参照してください。)

`getpdb_info.pl` もしくは、分子ビューアで PDB ファイルに含まれる分子を観察することは、どの分子を残し、どの分子を削除するかを判断するための手助けになるでしょう。

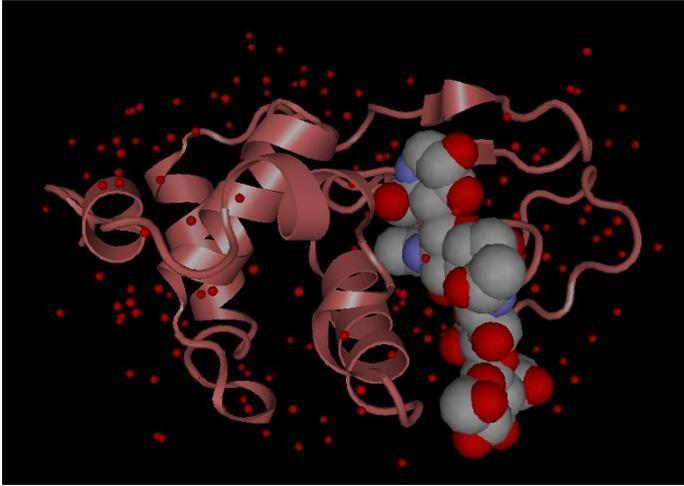


図 3. 4HP0.pdb の内容描画
タンパク質のペプチド鎖(1本)の他に阻害剤と水分子が含まれています。

次に、`pdbcheck` の制御ファイルを作成します。テキストエディタを使わずに、入力ファイルを用意するためには、以下のようにコマンドを実行します。ここでは、`pdbcheck` の制御ファイル名を `inp_pdbcheck` としています。このファイルの名前は、別のものでもかまいません。

```
(以下、pdbcheck の制御ファイルの用意)
% echo 4HP0.pdb > inp_pdbcheck
% echo 4HP0_1.pdb >> inp_pdbcheck
% echo -alt >> inp_pdbcheck
% echo -ss >> inp_pdbcheck
% echo -disableHet >> inp_pdbcheck
```

以上の手順で作成された `inp_pdbcheck` の内容は以下のようになります。

```
4HP0.pdb
4HP0_1.pdb
-alt
-ss
-disableHet
```

`pdbcheck` の制御用インプットファイルは、2行以上で構成されます。

1行目: 読み込み用 PDB ファイルの名前(パスを含めることもできます。)

2行目: 書き出し用 PDB ファイルの名前(パスを含めることもできます。)

3行目以降: 使用するオプションを1行に1つずつ記述します。上の例では、3つのオプションを使用しています。

以下のコマンドで `pdbcheck` を実行します。

```
% ../bin/pdbcheck < inp_pdbcheck (pdbcheck の実行)
```

この処理後に出来る `4HP0_1.pdb` が、`tplgeneX` の入力として適した PDB ファイルです。

次に、`tplgeneX` の制御ファイルを作成します。テキストエディタを使用しないで作成するには以下のコマンドで、ファイルを用意します。

```
(以下、tplgeneX の制御ファイルの用意)
% echo 1 > inp_tplgeneX
% echo 1 >> inp_tplgeneX
% echo 4HP0_1.pdb >> inp_tplgeneX
% echo 1 >> inp_tplgeneX
% echo Pro.pdb >> inp_tplgeneX
% echo Pro.tpl >> inp_tplgeneX
```

用意した `inp_tplgeneX` の内容は以下のようになっています。

```
1
1
4HP0_1.pdb
1
Pro.pdb
Pro.tpl
```

(現在フォーマット変更中 (数字の部分) -> `pdb/pdbx` など)

次のコマンドで実行します。

```
% ../bin/exec_tplgeneX.sh < inp_tplgeneX (tplgeneX の実行)
```

このコマンドが問題なく終了すれば、`sievgene` の入力ファイルとして使用できる PDB ファイル(`Pro.pdb`)、トポロジーファイル(`Pro.tpl`)が作成されています。

ターゲットサイト指定用プローブ点ファイルの作成:

ここでは、オリジナルの PDB ファイルに含まれている阻害剤(`NOJ-NAG-NAG-NAG`)を利用します。次のコマンドで該当部分を抽出します。

```
% grep "^HETATM" 4HP0.pdb | grep -v "HOH" > point.pdb (プローブ点データの作成)
```

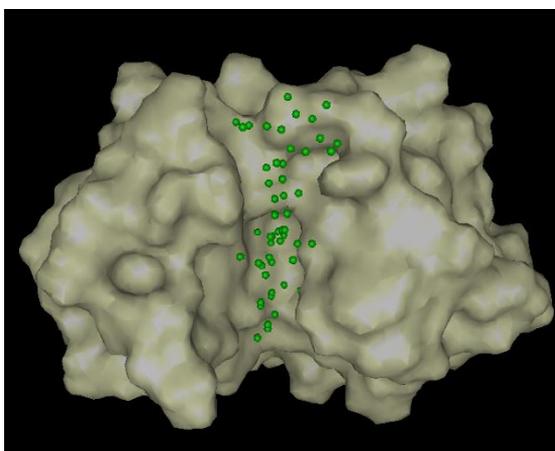


図 4. ターゲットタンパク質とプローブ点

PDB ファイルから水分子とリガンド分子を除去したものをターゲットタンパク質の PDB ファイルとし、リガンド分子の座標をプローブ点として使用しています(緑色の点)。

リガンドファイルの準備：

次のコマンドでサンプルとして用意してあるリガンドの **mol2** ファイルを作業ディレクトリにコピーします。サンプルとして用意した **sievgene** 制御ファイル内にて、読み込むリガンドファイル名を **ligand.mol2** と設定していますので、**ligand.mol2** という名前にしています。他の名前のファイルを使用するためには制御ファイル内の記述を変更する必要があります。

```
cp ../sample/c001-1.mol2 ./ligand.mol2
```

sievgene 制御用ファイルの準備：

次のコマンドで、サンプルとして用意している制御用ファイルを作業しているディレクトリにコピーします。

```
% cp ../input/inp_sievgene1 .
```

この動作テスト用プログラムでは、サンプルで用意した sievgene の入力ファイル (inp_sievgene1) を変更なしに使用できるようになっていますが、このファイルは、sievgene を実行するディレクトリに、以下の名前でファイルを用意していることを想定しています。

PDB ファイル: Pro.pdb

トポロジーファイル: Pto.tpl

ターゲットサイト指定用プローブ点ファイル: point.pdb

ドッキングさせるリガンド: ligand.mol2

inp_sievgene1 の内容を以下に示します。

inp_sievgene1:

```
.;
.; Sample input for sievgene version4.
.; Slow but more precise than the input for screening.
.;
PHASE> INPUT
LIGAND = MOL2
NAMELI = ligand.mol2 ; リガンドの座標位置ファイル
REFERE = NORE
NAMERE = ligand.mol2 ; リガンドの参照位置ファイル
TOPOLO = FORM
NAME TO = Pro.tpl ; 蛋白側のトポロジー
COORDI = PDB
NAMECO = Pro.pdb ; 蛋白側の座標
POINTC = PDB
NAMEPO = point.pdb ; ポケット領域を示すプローブ点

SETTAR = NORE
DAMPPA = 1.0d0
QUIT
.;
.; gird generation and Hash table generation
.;
PHASE> GRID
GRIDPotential = NORE ; Grid file reading SW (NORE/ASCII/BINA)
NAMEGrid = grid.file ; Grid file
OUTGRIDpotential = NOWR ; Grid file writing SW (NOWR/ASCII/BINA)

PROBDIst = 6.5 ;
MARGIN = 6.5 ; search margin
ITERAT = 3 ; iteration of Grid potential smoothing
RADVDW = 0.6 ; vDW boundary
RADELE = 0.6 ; coulomb boundary
RADMESH = 1.4 ; probe radius

DAMPVW = 0.99d0

USEPBG = NO ; not use PB
QUIT
;
```

```

; conformer generation
;
PHASE> CONF
  ATMMDL      = ALL ; UNIT      ; united atom model
  CONFLimit   = 100000 ;
  CONFORMernumber = 100 ;
  SORTATom    = YES ;
  DAMPING     = 0.7 ;
  PHASETorsion = 3 ;
  ROTTER      = YES ;
  QUIT

PHASE> DOCK
  METHOD       = FLEX
  PROSUR      = HYDR
  GENERATION  = 1 ;
  NUMCONFomer = 50000 ; 1000 ; number of triangle
  MATCHING    = 2 ; matching type
  LOWMIN      = 2.5 ;
  LOWMAX      = 3.5 ;
  UPRMIN      = 5.0 ;
  UPRMAX      = 12.0 ;

  RADIUS      = 6.0 ;
  WETVDW      = 1.0 ;
  WETASA      = 1.0
  WETELE      = 1.0
  WETHYD      = 1.0
  EVALHB      = NO
  WETANH      = 1.0d0
  ROTLOH      = NO
  ROTPSC      = NO

  MOVNUM      = 200 ; 10
  CANDID      = 10 ; 10 ;
; DOCKSP = FAST
  QUIT

EXE> MIN
  METHOD=    STEEP          CPUTIM = 360000.0
  UPRATE=    1.0           DOWNRATE= 0.3
  LOOPLI=    50           UPDATE = 100
  MONITO=    50           CONVGR = 0.1D0
  CUTMET=    RESA         CUTLEN = 22.0D0
  DIEFUN=    DIST         DIEVAL = 4.0D0
  LOGFOR=    SHOR

  QUIT

PHASE> OUTPUT
  COORDinate   = MOL2      ; coordinate file type
  NAMECOORDinate = ex.mol2 ; coordinate file
  NAMESCORE    = ex.score  ; score file
  CANDIDatenum= 5         ; number of PDB
  SCORENumber  = 5         ; number of score
  QUIT

EXE> SIEV

```

自分で編集する場合に、特に気をつける必要があるのは、赤字の箇所です。特に入力ファイルが適切に設定されていなければ、計算に失敗しますので、パスを含めて正確に記述してください。

sievgene の実行:

次のコマンドで sievgene_dk(sievgene_for_dockingpose)を実行します。

```
../bin/sievgene_dk < inp_sievgene1
```

ドッキングスコアの結果は ex.socre に、ドッキングポーズは ex.mol2 に出力されています。

```
INFORMATION> BEFORE MINIMIZE RANKING :EXPERIMENT ID=1
INFORMATION> LOCAL SCORE RANKING          5

  SCORE(/100) dG    HIT    MTS    rotNum  ASA    ELE    HYD    VDW    SURFACE    RMSD
1   -1.88   -5.15  -47.44 -180.65    0   -170.89  1.18 -18.04  0.00  0.00  24.95
2   -1.86   -5.06  -48.62 -177.76    0   -170.24  0.28 -15.83  0.00  0.00  25.30
3   -1.83   -5.01  -47.35 -175.75    0   -166.50  0.42 -17.00  0.00  0.00  25.21
4   -1.83   -5.02  -45.69 -176.35    0   -166.38  1.47 -18.08  0.00  0.00  25.11
5   -1.82   -5.00  -49.10 -175.05    0   -163.36 -1.26 -17.82  0.00  0.00  25.57

REFERENCE COORDINATE DATA
  999.90  999.90   0.00   0.00

INFORMATION> AFTER MINIMIZE RANKING :EXPERIMENT ID=1
INFORMATION> LOCAL SCORE RANKING          5

  COMPOUND NAME : HTS1404-00000229-01
  FILE NAME     : ligand.mol2

  SCORE(/100) dG    HIT    MTS    rotNum  ASA    ELE    HYD    VDW    SURFACE    RMSD
@ 1   -1.95   -5.37  -50.06 -188.10    0   -173.66  0.11 -20.98 -0.59  69.89  24.92
@ 2   -1.93   -5.28  -47.45 -186.01    0   -177.05  2.37 -18.10 -0.69  68.31  24.98
@ 3   -1.89   -5.19  -47.86 -182.02    0   -172.38  1.18 -18.04  0.00  68.26  24.95
@ 4   -1.89   -5.19  -46.62 -182.27    0   -169.47  1.55 -20.12 -0.83  71.44  25.21
@ 5   -1.88   -5.14  -46.72 -180.92    0   -170.72  1.69 -18.28 -0.82  69.61  25.22

REFERENCE COORDINATE DATA
  999.90  999.90   0.00   0.00

INFORMATION> COMPOUND RANKING          1

  COMPOUND NAME          SCORE(/100) dG    HIT    MTS
1 HTS1404-00000229-01   -1.95   -5.37  -50.06  -188.10
```

7.2. AmpC(2PU2)の例

AmpC(2PU2.pdb)をターゲットとしたドッキングの手順について説明します。
sievgene_packYYMMDD/に移動してから、以下のコマンドを実行してください。

```
% mkdir tmp_2PU2
% cd tmp_2PU2
% cp ../sample/2PU2.pdb .
% ../bin/get_pdb_info.pl 2PU2.pdb
```

次の枠内の情報は、get_pdb_info.pl の出力例です。

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

No of ligand name:3
Ligand name 1: PO4
Ligand name 2: DK2
Ligand name 3: HOH
```

この情報を参考にして、使用する分子を検討します。ここでは、A鎖のみを使用します。

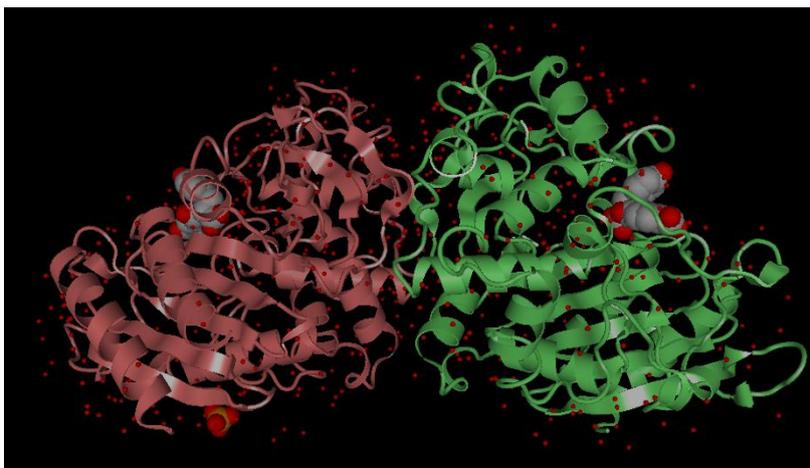


図 5. 2PU2.pdb の内容描画

2PU2.pdb には同じユニットが2つ含まれており、それぞれに阻害剤が結合しています。他に、PO4 と水分子が含まれています。

```
% ../bin/select_chain.pl A 2PU2.pdb > 2PU2_1.pdb (A鎖のみを出力します。)
% echo 2PU2_1.pdb > inp_pdbcheck
% echo 2PU2_2.pdb >> inp_pdbcheck
% echo -alt >> inp_pdbcheck
% echo -ss >> inp_pdbcheck
% echo -disableHet >> inp_pdbcheck
% ../bin/pdbcheck < inp_pdbcheck
```

```
% echo 1 > inp_tplgeneX
% echo 1 >> inp_tplgeneX
% echo 2PU2_2.pdb >> inp_tplgeneX
% echo 1 >> inp_tplgeneX
% echo Pro.pdb >> inp_tplgeneX
% echo Pro.tpl >> inp_tplgeneX
% ../bin/exec_tplgeneX.sh < inp_tplgeneX
% grep "^HETATM" 2PU2_1.pdb | grep "DK2" > point.pdb
(DK2の座標をプローブ点として使用します)
% cp ../sample/c001-1.mol2 ligand.mol2
% cp ../input/inp_sievgene1 .
% ../bin/sievgene_dk < inp_sievgene1
```

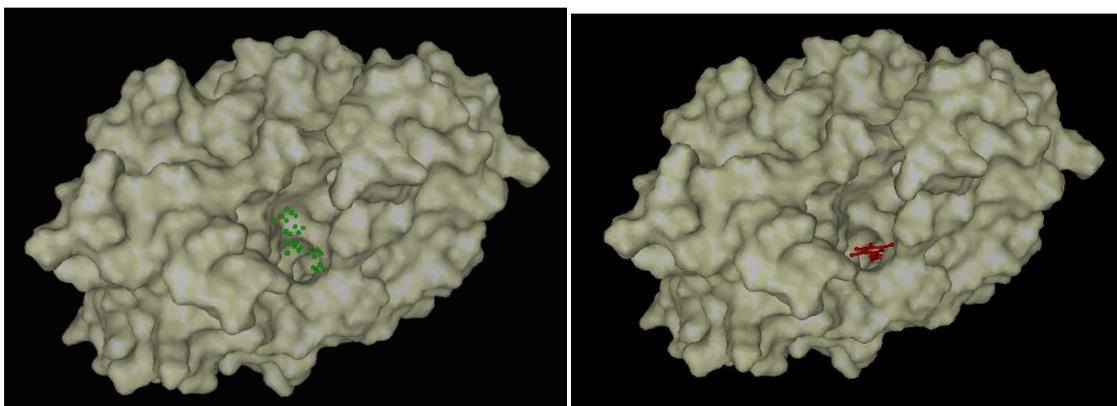


図 6. ターゲット指定用プローブ点(左図, 緑点) とドッキング構造の例(右図, 赤色) (2PU2)

7.3. Factor Xa(2W26)の例

Factor Xa(2W26.pdb)をターゲットとしたドッキングの手順について説明します。
sievgene_packYYMMDD/に移動してから、以下のコマンドを実行してください。

```
% mkdir tmp_2W26
% cd tmp_2W26
% cp ../sample/2W26.pdb .
% ../bin/get_pdb_info.pl 2W26.pdb
```

次の枠内は、get_pdb_info.pl の出力例です。

```
SSBOND information:
SSBOND 1 CYS A 22 CYS A 27 1555 1555 2.07
SSBOND 2 CYS A 42 CYS A 58 1555 1555 2.05
SSBOND 3 CYS A 122 CYS B 44 1555 1555 2.05
SSBOND 4 CYS A 168 CYS A 182 1555 1555 2.02
SSBOND 5 CYS A 191 CYS A 220 1555 1555 2.03
SSBOND 6 CYS B 1 CYS B 12 1555 1555 2.05
SSBOND 7 CYS B 8 CYS B 21 1555 1555 2.02
SSBOND 8 CYS B 23 CYS B 36 1555 1555 2.06

No of peptide chain:2
Chain ID 1: A
Chain ID 2: B

No of ligand name:3
Ligand name 1: RIV
Ligand name 2: CA
Ligand name 3: HOH
```

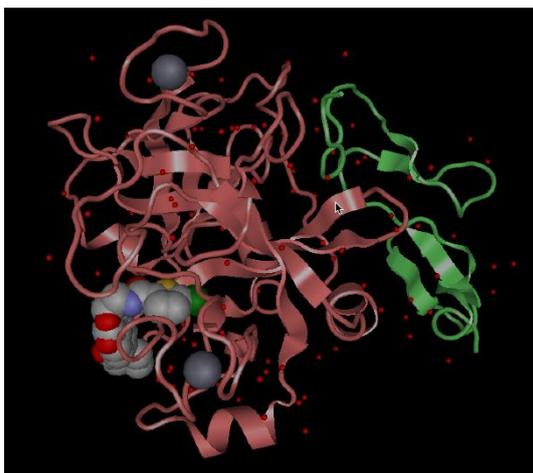


図 7. 2W26.pdb の描画

タンパク質のペプチド鎖(2本)の他に、阻害剤(RIV)、カルシウムイオン(CA)、水分子が含まれている。このタンパク質は、2本の鎖で1つのユニットを形成している。

ここでは、CA と HOH を除去し、RIV の座標をプローブ点情報として使用します。

```

% grep -v RIV 2W26.pdb > 2W26_1.pdb
% grep -v HOH 2W26_1.pdb > 2W26_2.pdb
% echo 2W26_2.pdb > inp_pdbcheck
% echo 2W26_3.pdb >> inp_pdbcheck
% echo -alt >> inp_pdbcheck
% echo -ss >> inp_pdbcheck
% echo -disableHet >> inp_pdbcheck (このオプションを使用しません。)
% ./bin/pdbcheck < inp_pdbcheck
% echo 1 > inp_tplgeneX
% echo 1 >> inp_tplgeneX
% echo 2W26_3.pdb >> inp_tplgeneX
% echo 1 >> inp_tplgeneX
% echo Pro.pdb >> inp_tplgeneX
% echo Pro.tpl >> inp_tplgeneX
% ./bin/exec_tplgenX.sh < inp_tplgeneX
% grep "^HETATM" 2W26.pdb | grep "RIV" > point.pdb
(RIVの座標をプローブ点として使用します)
% cp ../sample/c001-1.mol2 ligand.mol2
% cp ../input/inp_sievgene1 .
% ./bin/sievgene_dk < inp_sievgene1

```

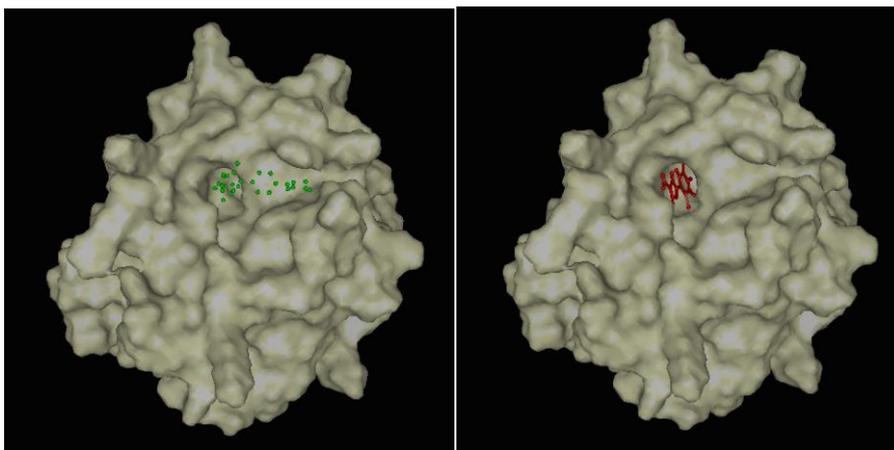


図 8. ターゲット指定用プローブ点(左図, 緑点) とドッキング構造の例(右図, 赤色) (2W26)

7.4. トロンビン(2PKS)の例

トロンビン(2PKS.pdb)をターゲットとしたドッキングの手順について説明します。まずは、sievgene_packYYMMDDに移動してから、以下のコマンドを実行してください。

```
% mkdir tmp_2PKS
% cd tmp_2PKS
% cp ../sample/2PKS.pdb .
% ../bin/get_pdb_info.pl 2PKS.pdb
```

次の枠内は、get_pdb_info.pl の出力例です。

```
SSBOND information:
SSBOND  1 CYS A   9   CYS B  155           1555  1555  2.05
SSBOND  2 CYS B  64   CYS B   80           1555  1555  2.00
SSBOND  3 CYS C 203   CYS C  217           1555  1555  2.01
SSBOND  4 CYS C 231   CYS C  261           1555  1555  2.02

No of peptide chain:4
Chain ID 1: A
Chain ID 2: B
Chain ID 3: C
Chain ID 4: D

No of ligand name:4
Ligand name 1: TYS
Ligand name 2: NA
Ligand name 3: G44
Ligand name 4: HOH
```

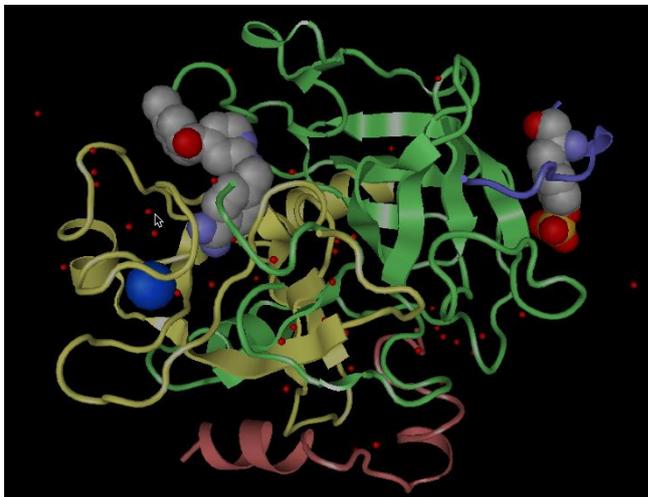


図 9. 2PKS.pdb の描画

タンパク質のペプチド鎖(4本)の他に、阻害剤(G44)、ナトリウムイオン(NA)、水分子が含まれています。TYS は、改変アミノ酸で、硫酸化チロシンであり、タンパク質の一部です。

get_pdb_info.pl において ligand name と報告されているものは、PDB ファイルにおいて HETATM から始まるデータ行の残基名です。この PDB ファイルにおける TYS は、少し特殊です。これは、タンパク質の鎖の途中で 1 残基のデータに相当する部分が HETATM で始まり、残基名が TYS となっています。これは改変アミノ酸です。TYR が硫酸化したものです。このような状況は、get_pdb_info.pl の出力のみでは分かりに

くいで、分子ビューアで PDB ファイルに含まれる情報を GUI で確認し、さらに、テキストエディタで PDB ファイルの中身をさっと確認するとよいでしょう。こうした改変アミノ酸は、そのままでは計算に使用できません。TYS を削除すると該当残基が全て取り除かれ、そこで鎖が分断してしまいます。この改変が重要でない場合には、改変アミノ酸を、改変前のアミノ酸に変更するとよいでしょう。テキストエディタでファイルを編集することによって、この変更は可能です。下の例のように、硫酸基部分の原子を除去し、HETATM を ATOM に、TYS を TYR に変更します。具体的には、原子番号 2316～2319 までを削除します。

2PKS.pdb に含まれる改変アミノ酸の例:

| | | | | | | | | | | | |
|--------|------|-----|-----|---|-----|--------|-------|--------|------|-------|---|
| HETATM | 2306 | N | TYS | D | 363 | 15.835 | 3.000 | -5.497 | 1.00 | 82.02 | N |
| HETATM | 2307 | CA | TYS | D | 363 | 16.520 | 2.761 | -4.197 | 1.00 | 81.45 | C |
| HETATM | 2308 | CB | TYS | D | 363 | 15.431 | 2.501 | -3.109 | 1.00 | 80.17 | C |
| HETATM | 2309 | CG | TYS | D | 363 | 14.488 | 3.694 | -2.956 | 1.00 | 78.55 | C |
| HETATM | 2310 | CD1 | TYS | D | 363 | 15.004 | 4.952 | -2.635 | 1.00 | 76.64 | C |
| HETATM | 2311 | CD2 | TYS | D | 363 | 13.108 | 3.574 | -3.136 | 1.00 | 76.01 | C |
| HETATM | 2312 | CE1 | TYS | D | 363 | 14.181 | 6.068 | -2.495 | 1.00 | 74.14 | C |
| HETATM | 2313 | CE2 | TYS | D | 363 | 12.282 | 4.699 | -3.003 | 1.00 | 74.26 | C |
| HETATM | 2314 | CZ | TYS | D | 363 | 12.802 | 5.961 | -2.659 | 1.00 | 72.73 | C |
| HETATM | 2315 | OH | TYS | D | 363 | 12.032 | 7.109 | -2.549 | 1.00 | 72.04 | O |
| HETATM | 2316 | S | TYS | D | 363 | 11.117 | 7.359 | -1.320 | 1.00 | 68.15 | S |
| HETATM | 2317 | O1 | TYS | D | 363 | 10.102 | 6.300 | -1.225 | 1.00 | 65.01 | O |
| HETATM | 2318 | O2 | TYS | D | 363 | 12.036 | 7.495 | -0.203 | 1.00 | 68.32 | O |
| HETATM | 2319 | O3 | TYS | D | 363 | 10.478 | 8.653 | -1.507 | 1.00 | 69.42 | O |
| HETATM | 2320 | C | TYS | D | 363 | 17.600 | 1.668 | -4.333 | 1.00 | 81.84 | C |
| HETATM | 2321 | O | TYS | D | 363 | 17.961 | 1.040 | -3.360 | 1.00 | 82.05 | O |

改変アミノ酸部分の修正例:

| | | | | | | | | | | | |
|------|------|-----|-----|---|-----|--------|-------|--------|------|-------|---|
| ATOM | 2306 | N | TYR | D | 363 | 15.835 | 3.000 | -5.497 | 1.00 | 82.02 | N |
| ATOM | 2307 | CA | TYR | D | 363 | 16.520 | 2.761 | -4.197 | 1.00 | 81.45 | C |
| ATOM | 2308 | CB | TYR | D | 363 | 15.431 | 2.501 | -3.109 | 1.00 | 80.17 | C |
| ATOM | 2309 | CG | TYR | D | 363 | 14.488 | 3.694 | -2.956 | 1.00 | 78.55 | C |
| ATOM | 2310 | CD1 | TYR | D | 363 | 15.004 | 4.952 | -2.635 | 1.00 | 76.64 | C |
| ATOM | 2311 | CD2 | TYR | D | 363 | 13.108 | 3.574 | -3.136 | 1.00 | 76.01 | C |
| ATOM | 2312 | CE1 | TYR | D | 363 | 14.181 | 6.068 | -2.495 | 1.00 | 74.14 | C |
| ATOM | 2313 | CE2 | TYR | D | 363 | 12.282 | 4.699 | -3.003 | 1.00 | 74.26 | C |
| ATOM | 2314 | CZ | TYR | D | 363 | 12.802 | 5.961 | -2.659 | 1.00 | 72.73 | C |
| ATOM | 2315 | OH | TYR | D | 363 | 12.032 | 7.109 | -2.549 | 1.00 | 72.04 | O |
| ATOM | 2320 | C | TYR | D | 363 | 17.600 | 1.668 | -4.333 | 1.00 | 81.84 | C |
| ATOM | 2321 | O | TYR | D | 363 | 17.961 | 1.040 | -3.360 | 1.00 | 82.05 | O |

tplgeneX で処理後の該当箇所:

| | | | | | | | | | | |
|------|------|-----|-----|---|---|---------------|--------------|---------------|-------|-------|
| ATOM | 4584 | N | TYR | D | 9 | <u>15.835</u> | <u>3.000</u> | <u>-5.497</u> | 14.01 | -0.42 |
| ATOM | 4585 | H | TYR | D | 9 | 14.932 | 2.562 | -5.614 | 1.01 | 0.27 |
| ATOM | 4586 | CA | TYR | D | 9 | <u>16.520</u> | <u>2.761</u> | <u>-4.197</u> | 12.01 | -0.00 |
| ATOM | 4587 | HA | TYR | D | 9 | 16.996 | 3.682 | -3.862 | 1.01 | 0.09 |
| ATOM | 4588 | CB | TYR | D | 9 | 15.431 | 2.501 | -3.109 | 12.01 | -0.02 |
| ATOM | 4589 | HB2 | TYR | D | 9 | 14.839 | 1.630 | -3.388 | 1.01 | 0.03 |
| ATOM | 4590 | HB3 | TYR | D | 9 | 15.913 | 2.320 | -2.148 | 1.01 | 0.03 |
| ATOM | 4591 | CG | TYR | D | 9 | 14.488 | 3.694 | -2.956 | 12.01 | -0.00 |
| ATOM | 4592 | CD1 | TYR | D | 9 | 15.004 | 4.952 | -2.635 | 12.01 | -0.19 |
| ATOM | 4593 | HD1 | TYR | D | 9 | 16.067 | 5.079 | -2.488 | 1.01 | 0.17 |
| ATOM | 4594 | CE1 | TYR | D | 9 | 14.181 | 6.068 | -2.495 | 12.01 | -0.23 |
| ATOM | 4595 | HE1 | TYR | D | 9 | 14.611 | 7.030 | -2.257 | 1.01 | 0.17 |
| ATOM | 4596 | CZ | TYR | D | 9 | 12.802 | 5.961 | -2.659 | 12.01 | 0.32 |
| ATOM | 4597 | OH | TYR | D | 9 | 12.032 | 7.109 | -2.549 | 16.00 | -0.56 |
| ATOM | 4598 | HH | TYR | D | 9 | 11.095 | 6.945 | -2.676 | 1.01 | 0.40 |
| ATOM | 4599 | CE2 | TYR | D | 9 | 12.282 | 4.699 | -3.003 | 12.01 | -0.23 |
| ATOM | 4600 | HE2 | TYR | D | 9 | 11.215 | 4.605 | -3.142 | 1.01 | 0.17 |
| ATOM | 4601 | CD2 | TYR | D | 9 | 13.108 | 3.574 | -3.136 | 12.01 | -0.19 |
| ATOM | 4602 | HD2 | TYR | D | 9 | 12.676 | 2.614 | -3.378 | 1.01 | 0.17 |
| ATOM | 4603 | C | TYR | D | 9 | 17.600 | 1.668 | -4.333 | 12.01 | 0.60 |
| ATOM | 4604 | O | TYR | D | 9 | 17.961 | 1.040 | -3.360 | 16.00 | -0.57 |

tplgeneX によって、水素原子付加および欠損原子の付加が行われるので、原子番号はオリジナルのものより多くなります。tplgeneX では残基番号は変更されます。上の例では、主鎖の N(緑色の字、かつ、下線)と主鎖の CA(青色の字、かつ、下線)の座標が一致しています。

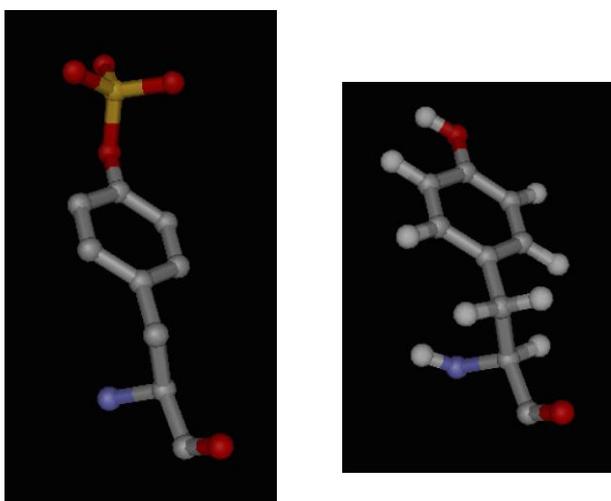


図 10. 硫酸チロシン残基とチロシン残基

2PKS.pdb に含まれる硫酸チロシン残基の構造(左図、水素はこの時点では付加していません)と、チロシン残基に変更後に tplgeneX で処理後の原子配置(右図、水素が付加されています)。

テキストエディタで、TYS を TYR に変更したファイルを、2PKS_mod.pdb
として tmp_2PKS/の下に保存します。2PKS_mod.pdb は、sample/に保存してありま
す。bin/test_2PKS.sh では、この 2PKS_mod.pdb を使用しています。
再度、get_pdb_info.pl を使用して、PDB ファイルの内容を確認します。

```
% cp ../sample/2PKS_mod.pdb . (サンプルとして提供しているものをそのまま使う場合)
% ../bin/get_pdb_info.pl 2PKS_mod.pdb
```

以下の出力例が得られます。

```
SSBOND information:
SSBOND  1 CYS A    9    CYS B  155          1555  1555  2.05
SSBOND  2 CYS B   64    CYS B   80          1555  1555  2.00
SSBOND  3 CYS C  203    CYS C  217          1555  1555  2.01
SSBOND  4 CYS C  231    CYS C  261          1555  1555  2.02

No of peptide chain:4
Chain ID 1: A
Chain ID 2: B
Chain ID 3: C
Chain ID 4: D

No of ligand name:3
Ligand name 1: NA
Ligand name 2: G44
Ligand name 3: HOH
```

TYYS が ligand としてレポートされなくなりました。ここでは、以下のコマンドを使っ
て、A 鎖～D 鎖までを使用し、NA を残し、G44 と HOH は削除してから、pdbcheck、
tplgeneX、 sievgene を順番に実行します。

```
% grep -v G44 2PKS_mod.pdb > 2PKS_1.pdb
% grep -v HOH 2PKS_1.pdb > 2PKS_2.pdb
% echo 2PKS_2.pdb > inp_pdbcheck
% echo 2PKS_3.pdb >> inp_pdbcheck
% echo -alt >> inp_pdbcheck
% echo -ss >> inp_pdbcheck
% echo -disableHet >> inp_pdbcheck (このオプションを使用しません。)
% ../bin/pdbcheck < inp_pdbcheck
% echo 1 > inp_tplgeneX
% echo 1 >> inp_tplgeneX
% echo 2PKS_3.pdb >> inp_tplgeneX
% echo 1 >> inp_tplgeneX
% echo Pro.pdb >> inp_tplgeneX
% echo Pro.tpl >> inp_tplgeneX
% ../bin/exec_tplgeneX.sh < inp_tplgeneX
```

```
% grep "^HETATM" 2PKS_mod.pdb | grep "G44" > point.pdb
% cp ../sample/c001-1.mol2 ligand.mol2
% cp ../input/inp_sievgene1 .
% ../bin/sievgene_dk < inp_sievgene1
```

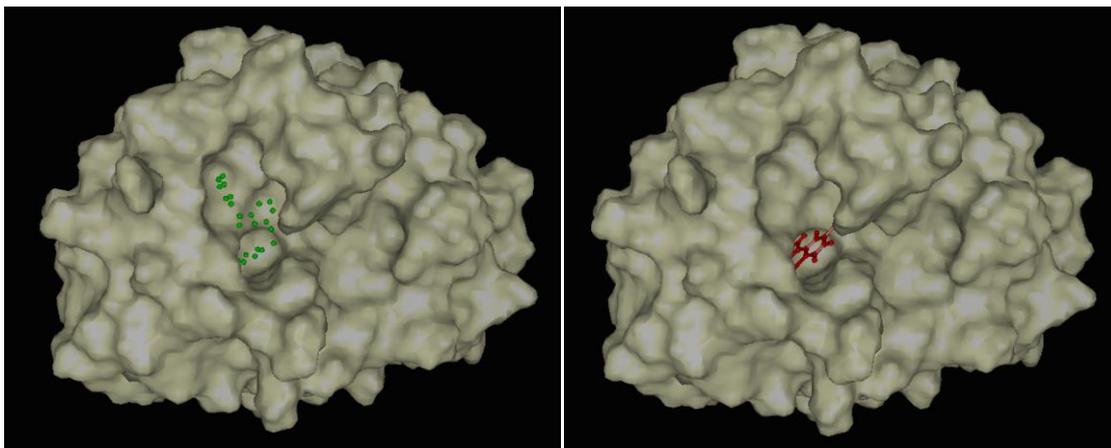


図 11. ターゲット指定用プローブ点(左図, 緑点) とドッキング構造の例(右図, 赤色)
(2PKS)

8. ツールプログラムについての説明

本章では、`sievgene_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。`sievgene`, `pdbcheck`, `tplgeneX`, `tplgeneL`, `Hgene` 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。`make_point` については、`sievgene` のマニュアル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するといいでしょう。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:

```
% ../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:

```
% ../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:

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

- `exec_tplgeneX.sh`

このプログラムは `sievgene_pack/bin/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
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`

このプログラムは `sievgene_pack/bin/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
DIR=$(cd $(dirname $0); cd ..; pwd)
echo $DIR
export PATH=$DIR/bin:$PATH
export TPLL_DB_PATH=$DIR/src/tplgeneL/tplgeneL/DB
echo "TPLL_DB_PATH:$TPLL_DB_PATH"
which tplgeneL
tplgeneL
```

9. リファレンス

- [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] “user_manual_v4.400_ja.pdf” .
- [3] “pdbcheck_manual_v4.400_ja.pdf” .
- [4] “sievgene_manual_v4.400_ja.pdf” .
- [5] “tplgeneX_manual_v4.400_ja.pdf” .