ハンズオンセミナー別添資料

タンパク質間ドッキングによる複合体構造予測

Protein Quaternary Structure Prediction with Protein-Protein Docking

大上 雅史

東京工業大学 大学院情報理工学研究科 計算工学専攻 秋山研究室 博士課程 3 年

Masahito Ohue

Depertment of Computer Science, Graduate School of Information Science and Engineering,
Tokyo Institute of Technology, Japan.

Email: ohue@bi.cs.titech.ac.jp

URL: http://www.bi.cs.titech.ac.jp/~ohue/

大上ハンズオンセミナー用のファイル群のダウンロード

本ハンズオンセミナーで使用するファイル群を以下でダウンロードしてください.

```
$ wget http://www.bi.cs.titech.ac.jp/~ohue/bps2013/handson2013.tgz
$ tar xzvf handson2013.tgz
$ cd tutorial
```

MEGADOCK のチュートリアル (1PPE)

PDB ID:1PPE という複合体を例にとって，MEGADOCK を動かしてみます．1PPE は β-Trypsin と Trypsin Inhibitor I (CMTI-I) の複合体結晶構造です．対応する単体構造は，1BTP(β-Trypsin) と，1LU0 chain A(CMTI-I)になります．これらのドッキング計算は，2 CPU コアによるスレッド並列により，約 3 分〜5 分で完了します．

```
$ export OMP_NUM_THREADS=2
```

（たくさんの CPU コアが使える PC の場合は数字を増やしてみてください）

（csh や tcsh を使っているマニアックな方は `setenv` コマンドを使用してください）

```
$ megadock -R pdb/1BTP.pdb -L pdb/1LU0_A.pdb -o 1PPE -N 1000
```

（対象 PDB 2 つ，出力ファイルの名前を 1PPE(.out)に，候補構造を 1000 個出力）

```
$ ./create.pl 1PPE.out
```

（out ファイルに対応する複合体の pdb ファイルを生成する）

```
$ pymol 1PPE.1.pdb
```

（評価値 1 位の予測複合体の構造(pdb ファイル)を見る）

ZRANK で構造の順位付けを行う

ZRANK は MEGADOCK の複合体構造に対し，原子レベルのエネルギースコア計算を行って順位を再評価するツールです．エネルギースコアを計算する際に，タンパク質構造(pdb ファイル)に水素を付加する必要があるため，reduce というツールで水素付加を行った後に ZRANK を実行します．

```
$ ./bin/reduce pdb/1BTP.pdb > pdb/1BTP.pdbh
$ ./bin/reduce pdb/1LU0_A.pdb > pdb/1LI0_A.pdbh
$ sed 's/.pdb/.pdbh/g' 1PPE.out > 1PPE.outh
$ ./bin/zrank32 1PPE.outh 1 1000
$ wc -l 1PPE.outh.zr.out   （行数の確認，1000 行であることを確認する）
$ less 1PPE.outh.zr.out   （エネルギースコアの確認）
$ sort -n -k 2 1PPE.outh.zr.out | less   （並べ変わった順位の確認）
```

正解の複合体結晶構造との RMSD を計算する

RMSD は構造予測の精度を測る指標の 1 つです．複合体構造予測では，予測構造(decoy)のレセプター側を重ねあわせたときのリガンドタンパク質同士の RMSD (**Ligand-RMSD**) をよく用います．今回は，レセプタータンパク質の Cα 原子同士での重ねあわせを行った時のリガンドタンパク質の全重原子間での RMSD を計算するスクリプト「rmsd.sh」を用いて，Ligand-RMSD (Å) を算出します．RMSD の計算には，1PPE 複合体の結晶構造をバラした 1PPE_r_b.pdb と 1PPE_l_b.pdb を用い，これを正解構造として decoy と比較します．

```
$ ./tools/rmsd.sh pdb/1PPE_r_b.pdb pdb/1PPE_l_b.pdb 1PPE.out
$ less 1PPE.rmsd.txt   （結果の確認．２列目が Ligand-RMSD [Å]）
```

RMSD の小さい順に並べ替えて確認

```
$ sort -n -k 2 1B6C.rmsd.txt | less
```

L-RMSD が 5Å 以下の decoy は「near-native」という言い方をし，正解と定義します．（ただし near-native にはいくつかの規準が存在するので注意が必要です．）

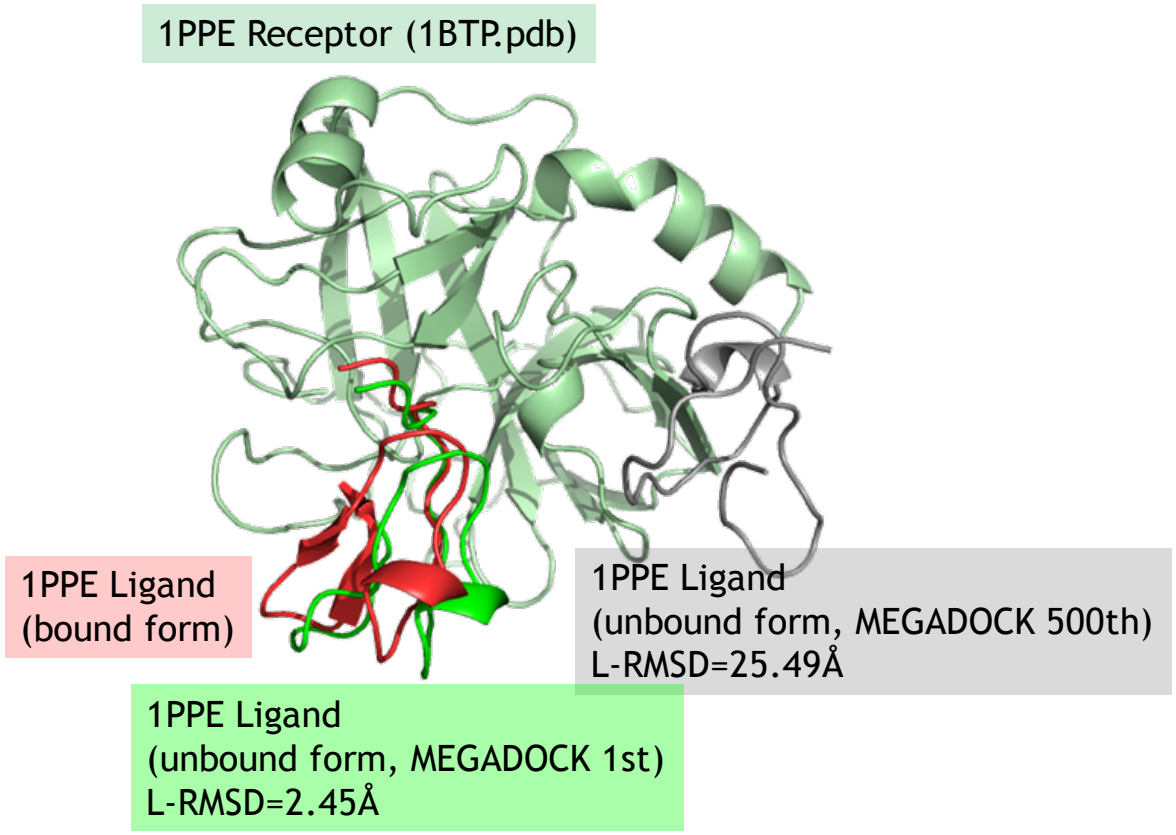
以下は少しだけ複雑な Unix のテクニックを用いています．以下のコマンドを試し，元のファイルなどと比べてみましょう．巻末に付した Unix のコマンド解説も参照ください．

（ZRANK スコア順に並べ替えた RMSD を確認する）

```
$ paste 1PPE.outh.zr.out 1PPE.rmsd.txt | sort -n -k 2 | cut -f 3,4 | less
```

（ZRANK スコア順に並べ替えて順位を振り直し，RMSD の小さい順に並べて見る）

```
$ paste 1PPE.outh.zr.out 1PPE.rmsd.txt | sort -n -k 2 | cut -f 4 | cat -n | sort -n -k 2 | less
```



PDB ID: 1PPE 複合体のドッキング例．MEGADOCK による予測 1 位と 500 位の decoy をそれぞれ示した．(`$ pymol tutorial/results/pymol/pymolsession_1PPE.pse` で描画可)

---- *MEMO* ----

演習 MEGADOCK と ZRANK による 1B6C 複合体予測

FK506 Binding Protein (1D6O_A) と TGF-β Receptor (1IAS_A) の複合体である PDB ID: 1B6C を用いて，MEGADOCK から ZRANK，RMSD 計算までの一連の流れをもう一度行ってみましょう．（ドッキング計算は，2 CPU コアで約 7 分くらいかかります.）

実行例を以下に示します.

MEGADOCK の実行

```
$ megadock -R pdb/1D6O_A.pdb -L pdb/1IAS_A.pdb -o 1B6C -N 1000
```

decoy PDB の生成

```
$ ./create.pl 1B6C.out
```

reduce による水素付加

```
$ ./bin/reduce pdb/1D6O_A.pdb > pdb/1D6O_A.pdbh
$ ./bin/reduce pdb/1IAS_A.pdb > pdb/1IAS_A.pdbh
```

out ファイルを編集（水素付加したものに変更）

```
$ sed 's/.pdb/.pdbh/g' 1B6C.out > 1B6C.outh
```

ZRANK の実行

```
$ ./bin/zrank32 1B6C.outh 1 1000
```

ZRANK スコアの確認

```
$ sort -n -k 2 1B6C.outh.zr.out | less
```

RMSD の計算

```
$ ./tools/rmsd.sh pdb/1B6C_r_b.pdb pdb/1B6C_l_b.pdb 1B6C.out
```

結果の確認

```
$ less 1B6C.rmsd.txt
```

RMSD の小さい順に並べ替えて確認

```
$ sort -n -k 2 1B6C.rmsd.txt | less
```

ZRANK 順に並べた RMSD の確認

```
$ paste 1B6C.outh.zr.out 1B6C.rmsd.txt | sort -n -k 2 | cut -f 3,4 | less
```

1PPE の例では，ZRANK をかけてもかけなくても 1 位の予測が near-native(正解)となっていましたが，1B6C では MEGADOCK が 1 位と出力した decoy は不正解(L-RMSD=38.57Å)となっています．ZRANK をかけると MEGADOCK の評価値が 157 位のものが 1 位になり，L-RMSD も 2.80Å で near-native となることが確認できました．時間に余裕がありましたら，PyMOL を用いて実際に構造を見てみましょう.

```
$ pymol 1B6C.1.pdb
$ pymol 1B6C.157.pdb
```

参考　handson2013.tar.gz の中身

```
$ wget http://www.bi.cs.titech.ac.jp/~ohue/bps2013/handson2013.tgz
```

で得られるファイル群の詳細です.

```
tutorial
 |- create.pl              out ファイルから pdb を生成する perl スクリプト
 |- create_lig             create.pl が用いるバイナリ
 |- bin
 |    |- reduce            水素付加ツール
 |    |- zrank32           ZRANK (32bit 用)
 |    |- zrank64           ZRANK (64bit 用)
 |- pdb
 |    |- 1BTP.pdb          1PPE のレセプター (unbound フォーム)
 |    |- 1LU0_A.pdb        1PPE のリガンド (unbound フォーム)
 |    |- 1D6O_A.pdb        1B6C のレセプター (unbound フォーム)
 |    |- 1IAS_A.pdb        1B6C のリガンド (unbound フォーム)
 |    |- 1PPE_r_b.pdb      1PPE のレセプター (bound フォーム，RMSD 計算用)
 |    |- 1PPE_l_b.pdb      1PPE のリガンド (bound フォーム，RMSD 計算用)
 |    |- 1B6C_r_b.pdb      1B6C のレセプター (bound フォーム，RMSD 計算用)
 |    |- 1B6C_l_b.pdb      1B6C のリガンド (bound フォーム，RMSD 計算用)
 |- tools
 |    |- rmsd.sh           RMSD 計算用シェルスクリプト
 |    |- burecfitrmsd.py   rmsd.sh が用いる python スクリプト(ProDy ライブラリ使用)
 |- results                ハンズオンセミナーで生成されるファイル群（結果確認用）
 |    |- bin
 |    |    |- megadock     megadock の実行ファイル（ubuntu12.04.2 でコンパイル済）
 |    |    |- tarball      各 tarball ファイル
 |    |    |    |- ProDy-1.4.3.tar.gz
 |    |    |    |- fftw-3.3.3.tar.gz
 |    |    |    |- megadock-2.6.tgz
 |    |- decoy
 |    |    |- 1PPE         1PPE の decoy PDB が含まれるディレクトリ（1000 個）
 |    |    |- 1B6C         1B6C の decoy PDB が含まれるディレクトリ（1000 個）
 |    |- outfile
```

```
|   |   |- 1PPE.out          1PPE のドッキング out ファイル
|   |   |- 1B6C.out          1B6C のドッキング out ファイル
|   |- protonation
|   |   |- 1PPE.outh         1PPE のドッキング out ファイル(水素付きファイルを参照する)
|   |   |- 1B6C.outh         1B6C のドッキング out ファイル(水素付きファイルを参照する)
|   |   |- 1BTP.pdbh         1BTP.pdb に reduce で水素付加したもの
|   |   |- 1LU0_A.pdbh       1LU0_A.pdb に reduce で水素付加したもの
|   |   |- 1D6O_A.pdbh       1D6O_A.pdb に reduce で水素付加したもの
|   |   |- 1IAS_A.pdbh       1IAS_A.pdb に reduce で水素付加したもの
|   |- pymol
|   |   |- pymolsession_1PPE.pse    この資料の図を描いたときの PyMOL セッションファイル
|   |- rmsdfile
|   |   |- 1PPE.rmsd.txt     1PPE の 1000 decoy の L-RMSD
|   |   |- 1B6C.rmsd.txt     1B6C の 1000 decoy の L-RMSD
|   |- zrankoutput
|   |   |- 1PPE.outh.zr.out  1PPE の ZRANK エネルギースコア
|   |   |- 1B6C.outh.zr.out  1B6C の ZRANK エネルギースコア
```

---- *MEMO* ----