myPresto 5.0

- in silico screening -

TUTOTRIAL

2020/06/22

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. 概要 4
2. 準備 5
2.1. 使用するファイル 5
2.2. ファイルの入手先 5
2.3. インストール方法 6
3. 計算の流れ 11
4. サンプルファイル 12
5. MTS 法実行手順(screening_org_server の場合) 13
6. ML-MTS 法の実行手順(screening_org_server の場合) 21
7. DSI 法の実行手順(screening_org_server の場合) 30
8. MTS 法実行手順(screening_org_PC の場合) 35
9. ML-MTS 法の実行手順(screening_org_PC の場合) 39
10. DSI 法の実行手順(screening_org_PC の場合) 43
11. コマンドの説明 46
12. 過去のプログラムセットとの違い 48
13. 注意事項 48

1. 概要

本チュートリアルでは、myPresto でのインシリコ・スクリーニングの手順について説明します。myPresto では、複数のスクリーニング計算方法が提供されています。ターゲットタンパク質の立体構造が利用可能かどうか、また、既知活性化合物が利用可能かどうかによって、それぞれ、適したスクリーニング方法を使い分けることができます。ここでは、MTS(multiple target screening)法、機械学習 MTS 法(以下、ML-MTS 法と記述)、DSI(docking score index)法を使った計算手順について説明します。これらは、ドッキング・シミュレーションの結果を利用したスクリーニング方法です。これらの計算方法の違いは、表 1 のようになります。

表 1: MTS 法・ML-MTS 法・DSI 法の違い

手法	ターゲットタンパク質の立体構造	既知活性化合物情報
MTS 法	利用する	利用しない
ML-MTS 法	利用する	利用する
DSI 法	利用しない	利用する

MTS 法、ML-MTS 法、DSI 法では、それぞれ、計算に使用するドッキング対象が一部異なります。

- MTS 法では、ターゲットタンパク質と多数の化合物とのドッキングを行います。
- ML-MTS 法では、ターゲットタンパク質と多数の化合物とのドッキングに加えて、ターゲットタンパク質と既知活性化合物、および、リファレンスタンパク質と既知活性化合物とのドッキングを行います。
- DSI 法では、ターゲットタンパク質とのドッキングは行わず、リファレンスタンパク質と既知活性化合物とのドッキング計算のみを行います。

これらの 3 つの方法は、LigandBOX の化合物と 181 個のリファレンスタンパク質とのドッキングデータ(相互作用行列)を用います。相互作用行列は、あらかじめ計算してあるものが次世代天然物化学技術研究組合から提供されています。

2. 準備

2.1. 使用するファイル

スクリーニング用のプログラム群は、screening_packYYMMDD.tar.gz として配布しています。(YYMMDD は年月日を表す数字です。) また、myPresto で用意している化合物ライブラリ LingandBOX（スクリーニング用に提供されている 200 万化合物のもの、相互作用行列データと一緒に配布）は、screening_dataYYMMDD.tar.gz として配布しています。(YYMMDD は年月日を表す数字です。)

2.2. ファイルの入手先

スクリーニング用プログラム群(screening_packYYMMDD.tar.gz)は、次のサイトからダウンロード可能です。

https://www.mypresto5.jp

スクリーニング用 LigandBOX 化合物ライブラリをご要望の場合は、上記の myPresto5 サイトのお問い合わせフォームから、その旨をお知らせください。非公開のダウンロードサイトの案内をメールでお知らせいたします。

2.3. インストール方法

myPreto5 のサイトからダウロードしたスクリーニング用プログラムパッケージ screening_packYYMMDD.tar.gz を書き込み可能なディレクトリに配置してください。その後に、次のコマンドで screening_packYYMMDD.tar.gz を解凍してください。

```
% tar –xzvf screening_packYYMMDD.tar.gz
```

解凍すると以下のディレクトリ構造で準備されています。

```
screening_packYYMMDD/
    |—bin/
    |     |--install.sh/
    |—src/
    |—src_script/
    |-- doc/
    |—screening_org_server/
    |—screening_org_PC/
    |—ref_protein/
    `—sample_data_4HP0/
```

スクリーニング計算にLigandBOX化合物ライブラリを使用する場合には、LigandBOXの圧縮ファイルも解凍しておいてください。LigandBOXの圧縮ファイルの解凍方法は、配布時にご案内します。

LigandBOXのディレクトリ構造：

```
(LigandBOXディレクトリ)/
        |— ligand/ (LigandBOXの化合物データ)
        |        |-- c001/
        |        |      `-- c001.mol2
        |        |-- c001/
        |        |      `-- c001.mol2
        |       …
        `— mts_data/ (相互作用行列)
                |-- c001_mts.dat
                |-- c002_mts.dat
               …
```

LigandBOXディレクトリ名は、配布時期と内容によって異なります。

screening_packYYMMDD.tar.gzの解凍が済んだら、インストールコマンドを実行する前に、圧縮ファイルを解凍してできたディレクトリの下に移動します。

```
% cd screening_packYYMMDD
```

コンパイルには、intelのFORTRANコンパイラ(ifort)もしくは、GNUのFORTRANコンパイラ(gfortran)が必要です。ifortが利用可能な場合は、そちらを使用してください。intelコンパイラの方が計算が速い場合が多いようです。

インストールコマンド実行前に、使用するシステムによって、使用するスクリプトプログラムのセット(ディレクトリ)を選択します。

使用するスクリプトプログラムは、次の２種類あります。

(1) screening_org_PC: ジョブスケジューラーを使わない場合に使用します。PC、PCクラスター、クラウドコンピューター等、多くの環境で動作します。

(2) screening_org_server : ジョブスケジューラーを使う場合に使用します。初期設定では、LSFを使用し、尚かつ、all.qというキューを使用する設定になっています。多くの場合、スクリプトプログラムの修正が必要です。

使用する方のディレクトリをscreening_orgという名前でコピーしてから、インストールコマンドを実行します。

次のコマンドの一方を実行します。

```
% cp -R screening_org_PC screening_org
% cp -R screening_server screening_org
```

次に、インストール用コマンドを実行します。

インテルコンパイラ(ifort)が利用可能な場合は、次のコマンドを実行してください。

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

GNUコンパイラ(gfortran)を使用する場合には、次のコマンドを実行してください。

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

このコマンドは、src/の下に配置されているsievgene_for_screening(sievgene_sc)、selectMTS、 selectDSIをコンパイルして作成された実行バイナリを、screening_org/base/bin/にインストールします。screening_org/の下には、多数の化合物に対するドッキング計算を自動的に実行するためのプログラムやデータが配置されています。このディレクトリは、計算毎にデータを上書きしないように、コピーして使用して下さい。

sample_data_4HP0/の下には、本チュートリアルで説明する計算手順で使用するサンプルファイルが置かれています。

screening_org_PCの構造：

```
screening_org_PC/
    |— protein/ (空のディレクトリ)
    `— base/
        |—bin/
        |    |—sievgene (install.sh実行後に出現)
        |    |—selectMTS (install.sh実行後に出現)
        |    |—selectDSI (install.sh実行後に出現)
        |    |
        |    |—submit_jobs.pl (only in screening_org_PC)
        |    |
        |    |—make_grid.csh (without any job scheduler command)
        |    |
        |    |—make_docking_score.csh  (without any job scheduler command)
        |    |—RUN_docking.pl
        |    |—startDocking.pl
        |    |
        |    |—make_score_data.pl
        |    |—makeMatrix.pl
        |    |
        |    |—run_group_MTS.pl(without any job scheduler command)
        |    |—get_result_MTS.pl
        |    |
        |    |—run_group_ML-MTS.pl(without any job scheduler command)
        |    |—auto_make_matrix_selected_ML-MTS.pl(without any job scheduler command)
        |    |—make_matrix_selected_ML-MTS.pl
        |    |—merge_matrix.pl
        |    |—run_selected_ML-MTS.pl (without any job scheduler command)
        |    |—remove_active.pl
        |    |
        |    |—run_group_DSI.pl (without any job scheduler command)
        |    |—auto_make_matrix_selected_DSI.pl   (without any job scheduler command)
        |    |—make_matrix_selected_DSI.pl
        |    `—run_selected_DSI.pl   (without any job scheduler command)
        |
        |—list/
        |    |—c001
        |    |—...
        |    `—c200
        |
        `—input/
             |—s0.inp
             `—s0grid.inp
```

screening_org_serverの構造：

```
screening_org_server/
    |— protein/ (空のディレクトリ)
    `— base/
        |—bin/
        |    |—sievgene (install.sh実行後に出現)
        |    |—selectMTS (install.sh実行後に出現)
        |    |—selectDSI (install.sh実行後に出現)
        |    |
        |    |—make_grid.csh (including bsub)
        |    |
        |    |—make_docking_score.csh (including bsub)
        |    |—RUN_docking.pl
        |    |—startDocking.pl
        |    |
        |    |—make_score_data.pl
        |    |—makeMatrix.pl
        |    |
        |    |—run_group_MTS.pl (including bsub)
        |    |—get_result_MTS.pl
        |    |
        |    |—run_group_ML-MTS.pl (including bsub)
        |    |—auto_make_matrix_selected_ML-MTS.pl (including bsub)
        |    |—make_matrix_selected_ML-MTS.pl
        |    |—merge_matrix.pl
        |    |—run_selected_ML-MTS.pl (including bsub)
        |    |—remove_active.pl
        |    |
        |    |—run_group_DSI.pl (including bsub)
        |    |—auto_make_matrix_selected_DSI.pl (including bsub)
        |    |—make_matrix_selected_DSI.pl
        |    `—run_selected_DSI.pl (including bsub)
        |
        |—list/
        |    |—c001
        |    |—...
        |    `—c200
        |
        `—input/
             |—s0.inp
             `—s0grid.inp
```

ref_protein の構造:

```
ref_protein/
        |— pro_list (181  タンパク質のリスト)
        |— protein/ (181タンパク質のPDBファイル、トポロジーファイル、プローブ点ファイル)
        |        |-- 12as/
        |        |       |-- Pro_md.pdb
        |        |       |-- Pro.tpl
        |        |       `-- point.pdb
        |       …
```

sample_data_4HP0の構造:

```
sample_data_4HP0/
        |— pro_list (181  タンパク質のリスト)
        |— protein/ (181タンパク質のPDBファイル、トポロジーファイル、プローブ点ファイル)
        |        |-- 12as/
        |        |       |-- Pro_md.pdb
        |        |       |-- Pro.tpl
        |        |       `-- point.pdb
        |       …
        |— virtual_hit_list(仮想的に設定した既知活性化合物のリスト)
        |— 4HP0/ (ドッキング用のファイル)
        |        |-- Pro_md.pdb
        |        |-- Pro.tpl
        |        `-- point.pdb
        `— 4HP0_virtual_hit/ (既知活性化合物データ)
                |-- virtual_hit1.mol2
               …
                `-- virtual_hit5.mol2
```

3. 計算の流れ

スクリーニング計算のおおよその流れは以下のようになります。

(1) 計算準備
(2) グリッド作成
(3) ドッキング計算
(4) ドッキングデータの集計（相互作用行列の作成）
(5) MTS 法/ML-MTS 法/DSI 法によるランキング

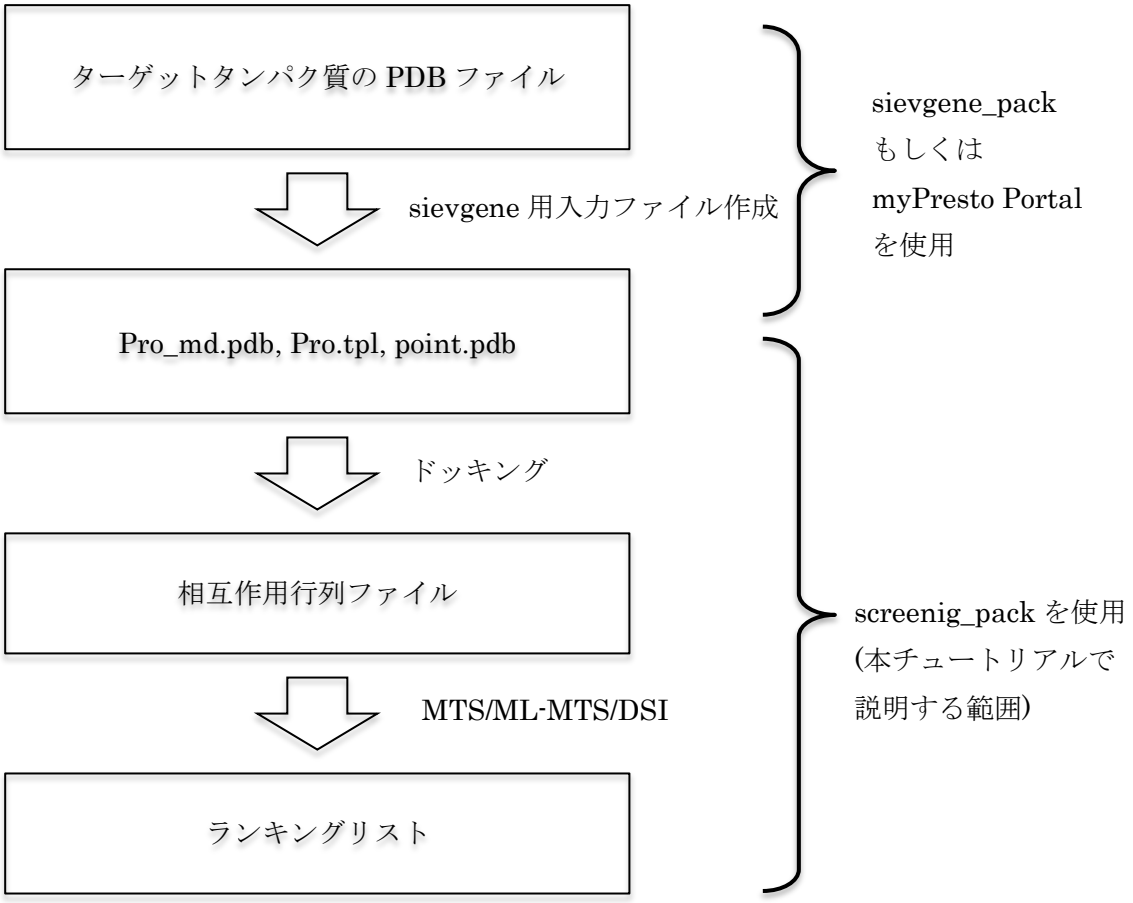

図１　重要ファイルに注目した作業過程の概要

sievgene を用いたドッキング・シミュレーションを実行するためには、まず、ドッキング用に処理を施した PDB ファイル(Pro_md.pdb)と、トポロジーファイル(Pro.tpl)、結合サイト指定ファイル(point.pdb)を作成する必要があります。これは、tplgeneX、make_point 等のプログラムを用いて作成しますが、本チュートリアルでは、これらのファイルが既に用意されているものとして、その後の手順について説明します。これらのファイル作成については、sievgene_pack のチュートリアルを参照してください。

4. サンプルファイル

本チュートリアルでは、ターゲットタンパク質にニワトリ・リゾチームを用いています。このターゲットタンパク質に対してドッキング用に用意したファイル(Pro_md.pdb、Pro.tpl、point.pdb)は、screening_packYYMMDD/sample_data_4HP0/4HP0/の下に置いてあります。これを使って本チュートリアルで説明する計算を実行することができます。

ML-MTS 法と DSI 法では、既知活性化合物を使用します。ここでは、MTS 法でのランクが、20 位、40 位、60 位、80 位、100 位だった化合物を、仮想的に既知活性化合物と見立てて計算の練習を行います。この化合物の mol2 ファイルは、screening_packYYMMDD/sample_data_4HP0/4HP0_virtual_hit/の下に置いてあります。

5. MTS 法実行手順(screening_org_server の場合)

ターゲットタンパク質と LigandBOX に含まれる化合物とのドッキング計算を実行し、リファレンスタンパク質と LigandBOX 化合物との相互作用行列を利用して、化合物毎に、ターゲットタンパク質がリファレンスタンパク質と比べて結合しやすいかどうかを評価します。

スクリーニング用 LigandBOX では、主に、1 万化合物を含むマルチ mol2 ファイルを読み込んで、1 つのジョブで 1 万化合物のドッキング計算計を行います。以下の手順では、c001-c004 に対する計算を実行しています。計算内容によって異なる箇所には色をつけています。

MTS 計算の手順(各コマンドに関する解説は、コマンド群の後にあります)

MTS-1% cp△–R△screening_org△screening_MTS_4HP0
　　△は空白文字を示しています。　　(screening_org をコピーして用います)
　　screening_MTS_4HP0 は、ディレクトリ名(フォルダ名)で任意です。
　　練習時には、これと同じ名前にすると良いと思います。
　　ここでの 4HP0 は、ターゲットの PDB ID を示しています。
MTS-2% cd△screening_MTS_4HP0
MTS-3% ln△–s△{LigandBOX ディレクトリ}/ligand△.　（最後の文字はドットです。）
(化合物ライブラリの設定)
(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。
例:　ln△-s△../../../screening_dataYYMMDD/ligand△.　　（最後の文字はドットです。）
この例では、screening_dataYYMMDD が LigandBOX のディレクトリ名ですが、これは配布時期と内容によって異なるものになります。
ligand の後ろに’/’(スラッシュ)があるとリンクに失敗します。)
MTS-4% cp△–R△../sample_data_4HP0/4HP0△protein/ (タンパク質の設定)
(ここではサンプルファイルを使用しています。)
MTS-5% cd△base/
MTS-6% csh△bin/make_grid.csh　　(グリッド作成)
------ここで、ジョブの終了待ちをします。------
MTS-7% echo△4HP0△>△list/target　　(リストファイル作成)
MTS-8% perl△bin/startDocking.pl△1△200△target (ドッキング)
------ここで、ジョブの終了待ちをします。------
MTS-9% perl△bin/makeMatrix.pl△1△200△target (データ集計)
MTS-10% cp△../../ref_protein/pro_list△list/
MTS-11% cp△list/pro_list△list/pro_list_add

MTS-12% echo△4HP0△>>△list/pro_list_add　　(リストファイル作成)
MTS-13% ln△–s△{LigandBOX ディレクトリ}/mts_data△.
(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。
例:　ln△-s△../../../screening_dataYYMMDD/mts_data△.
mts_data の後ろに’/’(スラッシュ)があるとリンクに失敗します。)
MTS-14% perl△bin/run_group_MTS.pl△1△200△4HP0△pro_list_add
(グループ毎の MTS 計算)
------ここで、ジョブの終了待ちをします。------
(次ページに続く)
MTS-15% perl△bin/get_result_MTS_sort.pl△1△200△4HP0△10000△>△ranking_list_MTS
ここでは、赤字の 10000 の部分は、化合物数の 1/200 の値を指定します。後述の説明を参照してください。
(選抜の MTS 計算)
MTS-16% perl△bin/getCatalogCode.pl△ranking_list_MTS△>△ranking_list_MTS_id
この ranking_list_MTS_id がカタログコード情報付きのランキングリストです。

(以上、ランキングリストが ranking_list_MTS に出力されています。)
MTS-17% perl△bin/get_top_compounds.pl△ranking_list_MTS△50△top
ここでの 50 は、ランキングの上位から何個ドッキングポーズを取得するかを指示する数字です。
MTS-18% mv△top/top△list
MTS-19% cd△..
MTS-20% unlink△ligand
MTS-21% mkdir△ligand
MTS-22% mv△base/top△ligand/
MTS-23% cd△base
MTS-24% csh△bin/make_docking_score.csh△target△top
------ここで、ジョブの終了待ちをします。------
MTS-25% perl△bin/get_struct.pl△target△top△top
(top/に上位化合物のドッキングポーズが出力されています。)

[MTS 法の計算手順に関する解説]

MTS-1: 複数のターゲットするドッキング計算を、1 つのディレクトリで行うと、計算 log が上書きされたり、どのターゲットに対する結果なのかが分かりにくくなったりする等の問題が発生しますので、screening_org はディレクトリごとコピーして使用します。

本チュートリアルでは、ディレクトリ名は、screening_(計算方法)_(ターゲット ID)としています。

MTS-2: コピーしたディレクトリに移動します。

MTS-3～4;

スクリーニング用スクリプトが想定している場所に、protein/と ligand/を用意します。

想定ディレクトリ構成:

```
screening_MTS_4HP0/
    |--base/
    |--protein/
    |--ligand/
```

MTS-3: screening_MTS_4HP0/の下にシンボリックリンクを張ります。シンボリックファイル名を ligand とします。シンボリックリンクを張ると、そのシンボリックリンクファイルが、リンク先と同等と見なせます。化合物のデータサイズは大きいので、ターゲット毎に、コピーするとディスク容量を消費しますので、シンボリックリンクにしてあります。

MTS-4: 今回ターゲットタンパク質として設定したのは、ニワトリ・リゾチームで、その PDB ID は、4HP0 です。

screening_pack/sample_data_4HP0/4HP0/の下に, Pro_md.pdb, Pro.tpl, point.pdb が保存されています。screening_MTS_4HP0/protein/の下に、そのディレクトリごと、コピーします。

MTS-5: screening_MTS_4HP0/base/に移動します。スクリーニング実行用に用意されているスクリプトプログラムは、このディレクトリで実行してください。

MTS-6: グリッド作成を行います。このスクリプトは、データ出力先のフォルダを作成した後に、sievgene を実行してグリッドファイルを作成します。グリッドファイルの出力先は、base/grid/4HP0/です。そこに、grid.file という名前のファイルが作成されます。このスクリプトを実行すると、base/work が(もし存在しなければ)作成され、その下で計算が実行されます。make_grid.csh は、ジョブスケジューラーへのジョブ投入コマンドを含んでいます。初期設定では、bsub コマンドを使用しています。この部分は、環境に依存して変更すべき箇所ですので、ユーザーの環境に合わせて変更が必要となる

場合があります。

MTS-7: スクリーニング計算の対象とするタンパク質を登録したリストファイルを作成します。MTS 計算では、ターゲットタンパク質のみを登録します。ここでは、target という名前にしますが、このファイルの名前は任意に設定することができます。ここでは、echo コマンドで、4HP0 をファイルに書き込んでいますが、テキストエディタで list/target を用意してもかまいません。場所は、base/list/下に置いてください。

MTS-8: ドッキング計算を実行します。200 万化合物の場合は、c001.mol2～c200.mol2 に対して、list/target に指定したタンパク質とのドッキングを行います。startDocking.pl の最初の２つの引数は、c001～c200 の何番から何番までを計算するかを指定する数字です。その後の引数は、計算対象とするタンパク質のリストを記入したファイル名を指定します。このファイルは list/の下に置いてあるものを想定しており、この引数ではファイルへのパスは含めません。MTS-7 で、該当ファイル名を target としましたので、この第３引数は、それに対応して target とします。

startDocking.pl の中では、bin/make_docking_score.csh を実行しています。

例: csh△bin/make_docking_score.csh△c001△target

この例では、１つの make_docking_score.csh コマンドでは、list/c001 の中に記録されている mol2 ファイルで、ligand/c001/の下にある mol2 ファイルに対して実行します。make_docking_score.csh は、make_grid.csh と同様に、ジョブスケジューラーへのジョブ投入コマンドを含んでいます。お使いの計算環境に応じて、変更が必要です。

この計算が開始すると、計算結果が base/result の下に書き出されます。4HP0 と c001.mol2 との計算結果は、最終的には、result/4HP0/c001.scores に出力されます。ただし、計算途中では、result/4HP0/c001/c001.score に出力され、計算が終了すると計算途中に出力していたファイルは消えて、一つ上のディレクトリに c001.scores が作成されます。スクリーニング計算において最も重要な結果である sievgene スコアは、@を含む行に記録されています。1 化合物に対する sievgene スコアは、@を含む１行に記録されています。計算途中で、どれだけの化合物に対して計算が終了しているか知るためには、result/4HP0/c001 へ移動して、次のコマンドを実行します。(4HP0 と c001 の組み合わせの場合)

```
% grep△'^@△'△c001.score△|△wc△-l
```

このコマンドは、c001.score の中から@ を含む行を抜き出します。その出力をパイプ(縦棒)で、wc コマンドの入力として使用しています。wc は、行数を数えるコマンドです。-l オプションを使用すると、行数のみを出力します。計算結果の一部はバッファリングされて、ある程度溜るまでファイルに書き出されませんので、このカウントは正確な値ではありません。

計算途中において、c001～c200 の全てに対して、計算完了化合物数を知りたい場合には、result/4HP0 に移動してから次のコマンドを実行します。

```
% grep△'^@△'△*/*.score△ | △wc
```

計算完了時には、c???/c???.score は消えていますので注意してください。計算が問題なく終了した後では、result/4HP0/で次のコマンドを実行すると計算が問題なく完了した化合物数をカウントできます。

```
% grep△'^@△'△*.scores△ | △wc△–l
```

ジョブが完了しているかどうかは、ジョブスケジューラーのコマンドを使用して確認することができます。これは、計算機システムによって異なるコマンドを使用します。
例: bjobs(LSF の場合), qstat(UGE 場合)

MTS-9: 計算結果を集計します。makeMatrix.pl に与えるコマンドは、startDocking.pl (MTS-8)で使用したものと同じものを使用してください。引数の意味は、startDocking.pl のものを同じです。このコマンドは、make_docking_score.pl を使用します。例えば、次のように、c001～c200 の１つに対して実行するコマンドを、引数で与えた範囲の数だけ実行します。

```
% perl△bin/make_docking_score.pl△c001△target
```

この結果は、base/matrix/の下に出力されます。
ファイル名は、target_c???.dat (???の部分は、001～200 のいずれか) となっています。
フォーマットの例を以下に示します。
taeget_c001.dat の例:

```
P 4HP0
L     1 HTS1404-00000229-01   -2.15000
L     2 HTS1404-00000392-02   -2.48000
L     3 HTS1404-00000490-02   -2.90000
L     4 HTS1404-00000524-01   -2.57000
L     5 HTS1404-00000597-02   -2.64000
以下省略。
```

MTS-10～12: MTS 計算用プログラム selectMTS で読み込む、MTS 計算の対象とするタンパク質のリストファイルを作成します。181 個のリファレンスファイルに、ターゲットタンパク質(今回は 4HP0)を加えたファイル(pro_list_add)を作成します。181 個のリファレンスタンパク質のリストは、screening_pack/ref_protein/pro_list にあります。4HP0 を追加する操作は、テキストエディタで行ってもかまいません。
MTS-13: MTS 計算で使用するデータファイルを格納したフォルダへのシンボリックリンクを張ります。

MTS-14: c001～c200 の各グループで、MTS 計算を行うジョブを投入するスクリプトを実行します(run_group_MTS.pl)。run_group_MTS.pl は、引数は４つ指定して実行します。

第 1 引数: 計算対象範囲の開始番号. c???の???の箇所に入る番号(1-200)
第 2 引数: 計算対象範囲の終了番号. c???の???の箇所に入る番号(1-200)
第 3 引数: ターゲットタンパク質の ID. protein/の下に配置したディレクトリ名.
第 4 引数: MTS 計算の対象となるタンパク質リストを記録したファイル名

run_group_MTS.pl は、内部で以下のコマンドを実行しています。実行場所は、work/4HP0_group_MTS/c???/です。(???には、001～200 のいずれかが入ります)

% selectMTS△<△inp_selectMTS

inp_selectMTS の内容は以下の通りです。

inp_selectMTS の内容:

```
../../../list/pro_list_add
n
merge_list
500
4HP0
0
60
10
```

selectMTS 用制御ファイルにおける各行の意味は以下の通りです。

1 行目: タンパク質リストファイル名（パスつきで）
2 行目: ヒットリストファイル名、もしくは、n(機械学習を行わない場合)(今回は n)
3 行目: 読み込む相互作用行列ファイルリストを記録したファイル名
4 行目: 出力件数
5 行目: ターゲットタンパク質の ID
6 行目: 計算モード
7 行目: AI 回数上限
8 行目: 試行回数

selectMTS は、MTS 法と ML-MTS 法の計算で使用します。MTS 法の場合には、２行目を n とし、また、6 行目の計算モードを 0 とします。6 行目を 0 とすると、7 行目と 8 行目は無視されます。

selectMTS 用制御ファイル 3 行目に指定するファイル(merge_list)の例:

```
../../../mts_data/c001_mts.dat
../../../matrix/target_c001.dat
```

これは、c001 に対しては、c001.mol2 に含まれている低分子化合物とリファレンスタンパク質 181 個のドッキングデータを記録した c001_mts.dat と、c001.mol2 に含まれている低分子化合物とターゲットタンパク質とのドッキングデータを記録した target_c001.dat を使用することを宣言しています。

run_group_MTS.pl は、ジョブスケジューラーのジョブ投入コマンドを使用しています。そのため、引数に、1 と 200 を与えると 200 個のジョブ投入を行います。ジョブ投入コマンドは、使用している計算機システムに依存して変更する必要があります。

MTS-15: get_result_MTS.pl は、run_group_MTS.pl で計算した各グループの上位 100 個を集めた集団に対して selectMTS を再度実行します。
この計算では最終的に MTS 法でのランキングにおける上位化合物について、sievgene スコアで再ソートしています。sievgene スコアでソートする理由は、元々の MTS 法(最後のソートを含まないもの)では、182 個のタンパク質集団の中で結合しやすいタンパク質を選択する方法として提案されたため、1/200 程度のグループに絞り込む方法として適当と考えられます。1/200 程度の集団の中では、sievgene スコアの順に並べた方が上位にヒット化合物が来る傾向があることが経験的に分かってきたため、最後に、元々の MTS 法の上位集団を sievgene スコアで再ソートしたものを、現在は MTS 法の最終的なランキングとしています。

MTS 法で得られたランキングリスト(ranking_list_MTS)の例:

Rank	Group	Compound code	Score	Target rank
1	c025	HTS1404-04305469-04	4.3700	15
2	c042	HTS1404-03383093-01	4.2900	55
3	c091	HTS1404-03613456-02	4.2800	30
4	c079	HTS1404-01459405-01	4.2800	50
5	c095	HTS1404-04664958-02	4.2700	35
6	c026	HTS1404-04066032-02	4.2600	56
7	c121	HTS1404-04171154-02	4.1900	52
8	c015	HTS1404-04093412-01	4.1800	26
9	c065	HTS1404-04130768-03	4.1800	42
10	c165	HTS1404-04675667-01	4.1700	34

このランキングリストは使用する計算機環境や使用するコンパイラに依存して若干異なります。ドッキング・シミュレーションは、莫大な数のドッキングポーズの中から、乱数を使った有限回の探索で最適なドッキングポーズを探索する計算です。ドッキングポーズの候補が莫大になる理由は、ドッキングポーズが、化合物の重心位置、角度、コンフォーメーションの自由度を持つためです。sievgene は、同じ計算環境で同じコンパイラを使った場合には、同じ結果になるように設計されていますが、CPU のアーキテクチャーが異なる場合や、異なるコンパイラで実行ファイルを作成した場合には、使用する乱数列が異なるため、結果が異なります。

6. ML-MTS 法の実行手順(screening_org_server の場合)

ML-MTS 法では、ターゲットタンパク質とライブラリの化合物とのドッキングに加えて、既知活性化合物とターゲットタンパク質、既知活性化合物とリファレンスタンパク質とのドッキングを行います。既知活性化合物の順位が良くなるようにスコアを変更します。実際の計算で異なる可能性が高い箇所は色付きの文字で示しています。

ML-MTS 計算の手順

ML-MTS-1% cp△–R△screening_org△screening_ML-MTS_**4HP0**
(screening_org をコピーして用いる)
ML-MTS-2% cd△screening_ML-MTS_**4HP0**
ML-MTS-3% rmdir△protein
ML-MTS-4% cp△–R△../ref_protein/protein△. (タンパク質の設定)
ML-MTS-5% cp△–R△../sample_data_4HP0/4HP0△protein/ (タンパク質の設定)
ML-MTS-6% mkdir△ligand
ML-MTS-7% cp△–R△../sample_data_4HP0/4HP0_virtual_hit△ligand/
(既知活性化合物の設定)
ML-MTS-8% cd△ligand/4HP0_virtual_hit
ML-MTS-9% ls△*.mol2△>△../../base/list/4HP0_virtual_hit
ML-MTS-10% cd△../../base
ML-MTS-11% csh△bin/make_grid.csh (グリッド作成)
------ここで、ジョブの終了待ちをします。------
ML-MTS-12% cp△../../ref_protein/pro_list△list/ (リストファイル作成)
ML-MTS-13% cp△list/pro_list△list/pro_list_add (リストファイル作成)
ML-MTS-14% echo△**4HP0**△>>△list/pro_list_add
ML-MTS-15% csh△bin/make_docking_score.csh△pro_list_add△4HP0_virtual_hit
(ドッキング 1: 既知活性化合物と 182 タンパク質)
------ここで、ジョブの終了待ちをします。------
ML-MTS-16% perl△bin/make_score_data.pl△pro_list_add△4HP0_virtual_hit (集計)
ML-MTS-17% cd△..
ML-MTS-18% mv△ligand△ligand2
ML-MTS-19% ln△–s△{LigandBOX ディレクトリ}/ligand△. (LigandBOX の設定)
(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。
例: ln△-s△../../../screening_dataYYMMDD/ligand△.
ligand の後ろに’/’(スラッシュ)があるとリンクに失敗します。)

ML-MTS-20% cd△base

ML-MTS-21% echo△4HP0△>△list/target

ML-MTS-22% perl△bin/startDocking.pl△1△200△target

(ドッキング 2: スクリーニング用 LigandBOX の化合物とターゲットタンパク質)

------ここで、ジョブの終了待ちをします。------

ML-MTS-23% perl△bin/makeMatrix.pl△1△200△target (データ集計)

ML-MTS-24% cd△matrix

ML-MTS-25% ls△pro_list_add_4HP0_virtual_hit.dat△>△../list/matrix_list

ML-MTS-26% cd△..

ML-MTS-27% ln△–s△{LigandBOX ディレクトリ}/mts_data△.

(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。

例:　ln△-s△../../../screening_dataYYMMDD/mts_data△.

mts_data の後ろに’/’(スラッシュ)があるとリンクに失敗します。)

ML-MTS-28% cp△../../sample_data_4HP0/virtual_hit_list△list/

ML-MTS-29%　perl △ bin/run_group_ML-MTS.pl △ 1 △ 200 △ 4HP0 △ pro_list_add △ virtual_hit_list△matrix_list　　(グループ毎の MTS 計算)

この ML-MTS-29 のコマンドは２行になっていますが、実際には１行で実行します。

------ここで、ジョブの終了待ちをします。------

ML-MTS-30% perl△bin/auto_make_matrix_selected_ML-MTS.pl△1△200△virtual_hit_list△4HP0

------ここで、ジョブの終了待ちをします。------

ML-MTS-31% perl△bin/merge_matrix.pl△1△200△work/4HP0_group_MTS△matrix_united.dat

(グループの上位化合物の相互作用行列を作成する)

------このコマンドは時間がかかります。------

ML-MTS-32% ls△matrix_united.dat△>>△list/matrix_list

ML-MTS-33% mv△matrix_united.dat matrix

ML-MTS-34%　perl△bin/run_selected_ML-MTS.pl△4HP0△pro_list_add△virtual_hit_list△matrix_list　　(選抜の ML-MTS 計算)

２行になっていますが、１行で実行します。

------ここで、ジョブの終了待ちをします。------

ML-MTS-35%　perl △ bin/remove_active.pl △ work/4HP0_selected_MTS/comp_list_MTS △ list/virtual_hit_list△ranking_list_ML-MTS

２行になっていますが、１行で実行します。

ML-MTS-36% perl△bin/modify_ranking_list.pl△ranking_list_ML-MTS

work/4HP0_group_MTS△ML-MTS△>△ranking_list_ML-MTS_mod

２行になっていますが、１行で実行します。

(次ページに続く)

ML-MTS-37% perl△bin/get_top_compounds.pl△ranking_list_ML-MTS_mod△50△top
ML-MTS-38% mv△top/top△list
ML-MTS-39% cd△..
ML-MTS-40% unlink△ligand
ML-MTS-41% mkdir△ligand
ML-MTS-42% mv△base/top△ligand/
ML-MTS-43% cd△base
ML-MTS-44% csh△bin/make_docking_score.csh△target△top
------ここで、ジョブの終了待ちをします。------
ML-MTS-45% perl△bin/get_struct.pl△target△top△top
(top/に上位化合物のドッキングポーズが出力されています。)

[ML-MTS 法の計算手順に関する解説]

ML-MTS-1: 複数のターゲットするドッキング計算を、１つのディレクトリで行うと、計算 log が上書きされて、どのターゲットに対する結果なのかが分かりにくくなるため、screening_org は、ディレクトリごとコピーして使用します。

本チュートリアルでは、ディレクトリ名は、screening_(計算方法)_(ターゲット ID)としています。

ML-MTS-2: コピーしたディレクトリに移動します。

ML-MTS 法の計算では、既知活性化合物も使用します。ドッキング計算は、２段階で行います。

まず、最初に、既知活性化合物と、182 個のタンパク質とのドッキング計算を行います。182 個のタンパク質とは、1 個のターゲットタンパク質(4HP0)と 181 個のリファレンスタンパク質です。(ML-MTS-3～ML-MTS-16)

次に、ターゲットタンパク質(4HP0)とライブラリ化合物とのドッキング計算を行います。２段階のドッキング計算が終了したら、それらのデータを合わせて使用して、最終的なランキングを決定します。

ML-MTS-3: 一旦、protein ディレクトリを消します。これは、ref_protein/の下に用意されている protein ディレクトリを、ディレクトリごとコピーしてくるためです。

ML-MTS-4: ref_protein/protein/を、screening_ML-MTS_4HP0/の下にコピーします。ref_protein/protein/には、MTS 計算で使用する 181 個のリファレンスタンパク質について、それぞれ、Pro_md.pdb、Pro.tpl、point.pdb が用意されています。protein/の下にある１つのディレクトリが、１つのタンパク質のものです。ディレクトリ名がタンパク質の ID となっています。

ML-MTS-5: 今回のターゲットタンパク質(4HP0)の Pro_md.pdb、Pro.tpl、point.pdb を含むディレクトリを、protein/の下にコピーします。

ML-MTS-6: 既知活性化合物の mol2 ファイルを配置する ligand ディレクトリを作成します。

ML-MTS-7: 既知活性化合物を含むディレクトリを ligand/の下にコピーします。ここでは、実際の活性化合物ではなく、計算の練習用に選出した化合物を用いています。ディレクトリ名は、計算対象のディレクトリを指定する際に使います。

ML-MTS-8～9: 計算対象の mol2 ファイルを指定するリストを作成します。このリストの保存先は、screening_ML-MTS_4HP0/base/list/の下です。ファイル名は、ディレクトリ名と同じにする必要があります。mol2 ファイルのある場所に移動して、ls *.mol2 で、mol2 ファイルのリストを出力し、その出力をファイルに保存します。

ML-MTS-10: base/に移動します。make_grid.csh を始めとする base/bin/に含まれるほ

とんどのスクリプトファイルは、このディレクトリで実行します。

ML-MTS-11: protein/の下に存在するディレクトリのそれぞれに対して、グリッドファイルが用意されていないものがあれば、そのグリッドファイルを作成します。ここでは、182 個のタンパク質に対するグリッドファイルを作成します。このスクリプトプログラム(make_grid.csh)は、内部で bsub コマンドを実行しています。そのため、環境に応じて、make_grid.csh のコードを修正する必要があります。ジョブの終了を確認するコマンドも、環境に応じて異なります。LSF の場合には、bjobs で、自分が投入したジョブのリストを表示します。この出力を見ることによって、グリッド作成が終了しているかどうかを確認できます。また、次のコマンドを実行する前に、実際にグリッドが作成されていることを確認するとよいでしょう。グリッドファイルは、base/grid/(タンパク質 ID)/grid.file として作成されています。base/にいる場合には、次のコマンドで、grid.file が何個できているかをカウントできますので、この出力が 182 となっていれば、今回使用する全てのグリッドファイルが作成されています。

```
% ls△grid/*/grid.file△|△wc
```

ML-MTS-12～ML-MTS-14: 計算対象とするタンパク質のリストを作成します。

リファレンスタンパク質のリストは、ref_protein/pro_list として用意されていますので、まず、これを list/の下にコピーします(ML-MTS-12)。さらに、それを pro_list_add としてコピーします(ML-MTS-13)。pro_list_add に、今回のターゲットである 4HP0 を追加書き込みします(ML-MTS-14)。

この 4HP0 は、screening_ML-MTS_4HP0/protein/の下にあるディレクトリ名の記述と対応するようにしてください。例えば、4hp0 ではダメです。ここでは説明を簡便にするために echo コマンドを使用していますが、テキストエディタで pro_list_add を開いて編集する方法でも問題ありません。

ML-MTS-15: 182 個のタンパク質と、既知活性化合物（ここでは仮想的に設定したもの）とのドッキングを行います。make_docking_score.csh の第 1 引数は、計算対象のタンパク質の ID(protein/の下のディレクトリ名)を記述したリストファイル名を、第 2 引数は、計算対象の低分子化合物の mol2 ファイル名を記述したファイル名を指定します。これらのファイルは、base/list/の下にあらかじめ配置してください。引数には、パスの情報は含めません。make_docking_score.sch は、ジョブスケジューラーのコマンド(bsub)を含んでいますので、環境に応じて書き換えが必要です。

ML-MTS-16: ML-MTS-15 の計算が終了後に実行します。182 個のタンパク質と、既知活性化合物とのドッキングデータを集計します。このコマンドを実行すると base/の下に、matrix/が作成され、その下に集計結果が保存ます。make_score_data.pl の 2 つの引数は、make_docking_score.csh と同じものを指定します。出力ファイル名は、(第 1 引数)_(第 2 引数).dat となります。ここでは pro_list_add_4HP0_virtual_hit.dat です。

ML-MTS-17～ML-MTS-23: ターゲットタンパク質とスクリーニング用化合物とのドッキング計算を行います。

ML-MTS-17～ML-MTS-19: ドッキング計算対象の化合物を、既知活性化合物のものからスクリーニング用化合物ライブラリに切り替えます。上の手順で使用していた既知活性化合物を含む ligand/を ligand2/とリネームします(ML-MTS-17)。
screeing_data/ligand へのシンボリックリンクを作成します(ML-MTS-19)。このシンボリックリンクによって、この場所に、screening_data/ligand が存在しているように扱えます。ちなみに、このシンボリックリンクを消すコマンドは、unlink ligand です。

ML-MTS-20: base/に移動します。

ML-MTS-21: ドッキングの計算対象とするタンパク質の ID を、list/の下の target という名前のファイルに書き込みます。ここでは、echo コマンドを使っていますが、テキストエディタで作成してもかまいません。

ML-MTS-22: 200 万化合物に対するドッキング計算を実行します。startDocking.pl は、内部で以下のコマンドを実行しています。

```
csh△bin/make_docking_score.csh△target△c001
csh△bin/make_docking_score.csh△target△c002
…
csh△bin/make_docking_score.csh△target△c199
csh△bin/make_docking_score.csh△target△c200
```

引数の 1 と 200 は、c001～c200 までの計算を意味しています。スクリーニング用の LigandBOX 化合物ライブラリには、1 つの mol2 ファイルに 1 万化合物が含まれています。ML-MTS-22 のコマンドを実行すると、1 万化合物ずつ、200 ジョブに分けて、合計 200 万化合物のドッキング計算を実行します。

ML-MTS-23: ML-MTS-22 の計算が終了してから、データを集計します。
makeMatrix.pl 1 200 は内部で、以下のコマンドを実行しています。

```
perl△bin/make_score_data.pl△target△c001
perl△bin/make_score_data.pl△target△c002
…
perl△bin/make_score_data.pl△target△c199
perl△bin/make_score_data.pl△target△c200
```

ML-MTS-24～ML-MTS-28: ML-MTS 計算の準備をします。ML-MTS 計算を実行するプログラムは、selectMTS です。ここでは、selectMTS を内部で実行するスクリプトプログラム使用して効率的に ML-MTS 計算を実行します。ここで準備するファイルは、selectMTS の入力ファイルです。

ML-MTS-24～ML-MTS-26: selectMTS の入力ファイルの 1 つを作成しています。読み込む相互作用行列（タンパク質と化合物とのドッキングデータ)のファイル名を入力

ファイルに出力しています。ここでは、ls コマンドの出力をファイルに書き出すことで、このファイルを作成していますが、テキストエディタで作成してもかまいませんし、echo コマンドで作成してもかまいません。

ML-MTS-27: スクリーニング用 LigandBOX の化合物とリファレンスタンパク質との相互作用行列ファイルを格納したディレクトリ(screening_data/mts_data)へのシンボリックファイルを base/の下に作成します。

ML-MTS-28: 本チュートリアルでのテスト計算で使用している化合物の名前を含むリスト(virtual_hit_list)を、sample_data_4HP0/の下から base/list/の下へコピーします。このリストは、mol2 ファイルのリスト名ではなく、mol2 ファイル中の@<TRIPOS>MOLECULE という記述の次行に記述されている化合物名です。

ML-MTS-29: run_group_ML-MTS.pl は、1 万化合物のグループ毎に MTS 計算を実行して、上位 100 個を選出しています。run_group_ML-MTS.pl は、内部で以下のコマンドを実行しています。実行場所は、work/4HP0_group_MTS/c???/です。(???には、001～200 のいずれかが入ります)

```
% selectMTS△<△inp_selectMTS
```

inp_selectMTS の内容は以下の通りです。これは、機械学習なしの MTS 法の計算実行と同じコマンドですが、inp_selectMTS のファイルの内容が異なっています。

inp_selectMTS の内容(ML-MTS 用):

```
../../../list/pro_list_add
../../../list/hit_test_name_list
merge_list
500
4HP0
1
60
10
```

selectMTS 用制御ファイルにおける各行の意味は以下の通りです。

1 行目: タンパク質リストファイル名（パスつきで）
2 行目: ヒットリストファイル名、もしくは、n(機械学習を行わない場合)
3 行目: 読み込む相互作用行列ファイルリストを記録したファイル名
4 行目: 出力件数
5 行目: ターゲットタンパク質の ID
6 行目: 計算モード
7 行目: AI 回数上限
8 行目: 試行回数

ML-MTS 法の場合には、２行目にヒットリストファイル名を指定します。ヒットリストは、既知活性化合物の化合物名のリストですが、この化合物名は、mol2 ファイルにおいて@<TRIPOS>MOLECULE の次の行に記述されている分子名です。また、ヒットリストファイルの 1 行目は、'X'のみ記述してください。例を以下に示します。

selectMTS 用制御ファイル 2 行目に指定するヒットリストファイルの例:
(本チュートリアルでは virtual_hit_list を使用)

X
HTS1404-04376179-02forTest
HTS1404-03625339-01forTest
HTS1404-04166694-02forTest
HTS1404-01459298-01forTest
HTS1404-04660192-03forTest

また、機械学習用 selectMTS 制御ファイルの 6 行目は、計算モードを 1 にします。
7 行目には機械学習の繰り返し回数、8 行目には機械学習の試行回数をしています。

ML-MTS-30: グループ毎に、選抜した化合物のデータのみを抽出した相互作用行列ファイルを作成します。
ML-MTS-31: 200 のグループで作成された相互作用行列を統合します。
ML-MTS-32: 統合した相互作用行列ファイル名を matrix_list に追加します。テキストエディタで追加してもかまいません。
ML-MTS-33: 統合した相互作用行列ファイルを matrix/の下に配置します。
ML-MTS-34: 選抜した化合物 2 万個に対して、ML-MTS 計算を実行します。
ML-MTS-35: ML-MTS-34 の計算結果は、既知活性化合物と混ざったランキングリストとなっていますので、このコマンドで既知活性化合物をリストから削除します。このコマンドで得られる ranking_list_ML-MTS が最終的なランキングリストです。

ML-MTS 法でのランキングリスト（ranking_list_ML-MTS）の例：

(既知活性化合物は除外済)

1	5779	HTS1404-02737258-01	4.4100	29
2	1279	HTS1404-04085880-01	3.8500	29
3	8131	HTS1404-04435677-01	4.0700	29
4	14645	HTS1404-04406230-01	4.4400	29
5	11847	HTS1404-04346488-01	3.8500	29
6	19671	HTS1404-03240645-01	3.9600	29
7	7914	HTS1404-04111945-02	4.7000	29
8	19645	HTS1404-04608132-01	4.2300	29
9	1167	HTS1404-02369245-01	4.1600	29
10	3648	HTS1404-03834473-01	4.0800	29

既知活性化合物のリスト(remove_from_comp_list)の例：

(１番左は、除外する前の順位、上のランキングリストは抜けを詰めてあります)

1	245	ZINC03814777	5.0400	29
2	290	ZINC01894681	4.5600	29
41	203	ZINC03814724	4.5300	30
42	169	ZINC03814692	4.4900	30
44	151	ZINC03814677	4.5600	30
45	222	ZINC03814744	4.8700	30
52	216	ZINC03814737	4.5300	30
59	164	ZINC03814689	4.7400	30
61	165	ZINC03814690	4.7500	30
64	233	ZINC03814760	4.6300	30
71	40	ZINC03814574	4.1500	30

分離前の comp_list_MTS（selectMTS からの直接の出力）：

1	245	ZINC03814777	5.0400	29
2	290	ZINC01894681	4.5600	29
3	5779	HTS1404-02737258-01	4.4100	29
4	1279	HTS1404-04085880-01	3.8500	29
5	8131	HTS1404-04435677-01	4.0700	29
6	14645	HTS1404-04406230-01	4.4400	29
7	11847	HTS1404-04346488-01	3.8500	29
8	19671	HTS1404-03240645-01	3.9600	29
9	7914	HTS1404-04111945-02	4.7000	29
10	19645	HTS1404-04608132-01	4.2300	29

7. DSI 法の実行手順(screening_org_server の場合)

DSI 法では、ターゲットタンパク質を使いません。既知活性化合物とリファレンスタンパク質とのドッキングを行い、既知活性化合物とドッキングプロファイルが近いものを選出します。

DSI 計算の手順

DSI-1% cp△–R△screening_org△screening_DSI_**4HP0**
(screening_org をコピーして用いる)
DSI-2% cd△screening_DSI_**4HP0**
DSI-3% rmdir△protein
DSI-4% cp△–R△../ref_protein/protein△. (タンパク質の設定)
DSI-5% mkdir△ligand
DSI-6% cp△–R△../sample_data_4HP0/4HP0_virtual_hit△ligand/
(既知活性化合物の設定)
DSI-7% cd△ligand/4HP0_virtual_hit
DSI-8% ls△*.mol2△>△../../base/list/4HP0_virtual_hit
DSI-9% cd△../../base/
DSI-10% csh△bin/make_grid.csh (グリッド作成)
------ここで、ジョブの終了待ちをします。(1 分程度)------
DSI-11% cp△../../ref_protein/pro_list△list/ (リストファイル作成)
DSI-12% csh△bin/make_docking_score.csh△pro_list△4HP0_virtual_hit (ドッキング)
------ここで、ジョブの終了待ちをします。------
DSI-13% perl△bin/make_score_data.pl△pro_list△4HP0_virtual_hit (集計)
DSI-14% cd△matrix
DSI-15% ls△pro_list_4HP0_virtual_hit.dat△>△../list/matrix_list
DSI-16% cd△..
DSI-17% cp△../../sample_data_4HP0/virtual_hit_list△list/
DSI-18% ln△–s△{LigandBOX ディレクトリ}/mts_data△.
(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。
例: ln△-s△../../../screening_dataYYMMDD/mts_data△.
mts_data の後ろに’/’(スラッシュ)があるとリンクに失敗します。)

(次ページに続く)

DSI-19% perl△bin/run_group_DSI.pl△1△200△pro_list△virtual_hit_list△matrix_list
(グループ毎の DSI 計算)
------ここで、ジョブの終了待ちをします。------
DSI-20% perl△bin/auto_make_matrix_selected_DSI.pl△1△200△virtual_hit_list
------ここで、ジョブの終了待ちをします。-----
DSI-21% perl△bin/merge_matrix.pl△1△200△work/group_DSI△matrix_merged.dat
------このコマンドは時間がかかります。------
DSI-22% ls△matrix_merged.dat△>>△list/matrix_list
DSI-23% mv△matrix_merged.dat△matrix
DSI-24% perl△bin/run_selected_DSI.pl△pro_list△virtual_hit_list△matrix_list
(選抜した化合物に対する DSI 計算)
------ここで、ジョブの終了待ちをします。------
DSI-25% perl △ bin/remove_active.pl △ work/selected_DSI/comp_list_PCA △ list/virtual_hit_list △ ranking_list_DSI
(ここで得られた ranking_list_DSI が DSI 法によるランキングリストです。学習用に使用した化合物については、removed_from_comp_list に出力されています。)
DSI-26%perl△bin/modify_ranking_list.pl△ranking_list_DSI△work/group_DSI/△DSI△>△ranking_list_DSI_mod
2 行になっていますが、1 行で実行します。
DSI-27%cd△..
DSI-28%mv△ligand△ligand2
DSI-29%ln△–s△{LigandBOX ディレクトリ}/ligand△.
(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。
例:　ln△-s△../../../screening_dataYYMMDD/ligand△.
ligand の後ろに’/’(スラッシュ)があるとリンクに失敗します。)
DSI-30%cd△base
DSI-31%perl△bin/get_top_compounds_with_rank.pl△ranking_list_DSI_mod△50△top
(top/に上位化合物のドッキングポーズが出力されています。)

[DSI 法の計算手順に関する解説]

DSI-1: 複数のターゲットするドッキング計算を、１つのディレクトリで行うと、計算 log が上書きされて、どのターゲットに対する結果なのかが分かりにくくなるため、screening_org は、ディレクトリごとコピーして使用します。

本チュートリアルでは、ディレクトリ名は、screening_(計算方法)_(ターゲット ID)としています。

DSI-2: コピーしたディレクトリに移動します。

DSI-3: DSI 法では、ターゲットタンパク質とのドッキング計算は行いません。代わりに、181 個のリファレンスタンパク質とのドッキング計算を行い、リファレンスタンパク質とのドッキングパターンが、既知活性化合物と類似している度合いによってランキングを決定します。リファレンスタンパク質を含む ref_protein/protein/を、ディレクトリごとコピーするために、既存の protein/を削除します。

DSI-4: ref_protein/protein/ を 、 screening_DSI_4HP0/ の 下 に コ ピ ー し ま す 。 ref_protein/protein/には、MTS 計算で使用する 181 個のリファレンスタンパク質について、それぞれ、Pro_md.pdb、Pro.tpl、point.pdb が用意されています。protein/の下にある１つのディレクトリが、１つのタンパク質のものです。ディレクトリ名がタンパク質の ID となっています。

DSI-5: 既知活性化合物の mol2 ファイルを配置する ligand/を作成します。

DSI-6: サンプルとして提供している練習用の既知活性化合物（仮想的に設定したもの）の mol2 ファイルを含むディレクトリ(4HP0_virtual_hit/)を、ligand/の下にコピーします。

本チュートリアルでは、ファイルサイズが小さなものについてはコピーし、大きなものについてはシンボリックリンクを使用しています。

DSI-7～DSI-8: 計算対象とする化合物のファイル名リストを作成しています。このリストのファイル名は、ディレクトリ名と同じにしてください。保存先は、base/list/です。

DSI-9: スクリプトプログラムを実行するディレクトリに移動します。

DSI-10: グリッドを作成します。make_grid.csh は、ジョブスケジューラーのコマンド(bsub)を含んでいます。これは環境に応じて変更する必要があります。

DSI-11: 計算対象とするタンパク質のリスト(pro_list)を、ref_protein/から base/list/にコピーします。

DSI-12: 181 個のリファレンスタンパク質と既知活性化合物とのドッキング計算を実行します。make_docking_score.csh は、ジョブスケジューラーのコマンド(bsub)を含んでいます。これは環境に応じて変更する必要があります。

DSI-13: DSI-12 の計算が終了後に実行します。ドッキング結果を集計し、相互作用行列を作成します。作成された相互作用行列は、base/matrix/の下に保存されています。

DSI-14～DSI-16: selectDSI の入力ファイルを作成しています。使用する相互作用行列のファイル名を記述したファイルを作成しています。テキストエディタで作成してもかまいません。

DSI-17: 本チュートリアルでのテスト計算で使用している化合物の名前を含むリスト(virtual_hit_list)を、sample_data_4HP0/の下から base/list/の下へコピーします。このリストは、mol2 ファイルのリスト名ではなく、mol2 ファイル中の @<TRIPOS>MOLECULE という記述の次行に記述されている化合物名です。

DSI-18: スクリーニング用 LigandBOX の化合物とリファレンスタンパク質との相互作用行列ファイルを格納したディレクトリ(screening_data/mts_data)へのシンボリックファイルを base/の下に作成します。

DSI-19: run_group_DSI.pl は、selectDSI を 1 万化合物毎に実行するプログラムです。それぞれのグループ(c001 等)から上位 100 化合物を選抜します。

DSI-20: グループ毎に、選抜した化合物のデータのみを抽出した相互作用行列ファイルを作成します。

DSI-21: 200 のグループで作成された相互作用行列を統合します。

DSI-22: 統合した相互作用行列ファイル名を matrix_list に追加します。テキストエディタで追加してもかまいません。

DSI-23: 統合した相互作用行列ファイルを matrix/の下に配置します。

DSI-24: 選抜した化合物 2 万個に対して、DSI 計算を実行します。

DSI-25: DSI-24 の計算結果は、既知活性化合物と混ざったランキングリストとなっていますので、このコマンドで既知活性化合物をリストから削除します。このコマンドで得られる ranking_list_DSI が最終的なランキングリストです。学習用に使用した既知活性化合物は、removed_from_comp_list に出力されています。

DSI 法で得られたランキングリスト(ranking_list_DSI)の例：

（既知活性化合物は除外してあります）

1	2665	HTS1404-03378466-01	0.9492
2	12109	HTS1404-03733381-01	1.2320
3	531	HTS1404-02888870-01	1.2863
4	5704	HTS1404-00750020-02	1.3164
5	19850	HTS1404-04196791-02	1.3824
6	18161	HTS1404-04389872-01	1.4848
7	3464	HTS1404-04097986-01	1.5072
8	2096	HTS1404-04653958-04	1.5150
9	3860	HTS1404-01572908-02	1.5353
10	17959	HTS1404-04317118-01	1.5497

ランキングリストから除いた既知活性化合物のリスト(removed_from_comp_list)の例:

11	4	HTS1404-01459298-01forTest	1.5551
19	1	HTS1404-04376179-02forTest	1.6809
98	2	HTS1404-03625339-01forTest	2.0120
132	5	HTS1404-04660192-03forTest	2.0860
146	3	HTS1404-04166694-02forTest	2.1160

8. MTS 法実行手順(screening_org_PC の場合)

前章までは計算機サーバーを使用した計算手順について説明しましたが、本章では、1 台の PC を使用して行う MTS 計算の方法について説明します。サーバーを使う場合には、5 章を参照してください。インストール時に、screening_org が存在していない状態で、bin/install.sh を実行していれば、screening_org_PC が screening_org にコピーされています。

最近の PC では、1 つの CPU に 2 コア、もしくは 4 コアが格納されている場合があります。事前に、使用する PC のコア数を確認しておくとしいでしょう。

スクリーニング計算で使用しているドッキングプログラム sievgene は、マルチ mol2 ファイルを読み込むことによって、1 度の計算で多数の化合物に対するドッキング計算を行うことができます。スクリーニング用の LigandBOX では、化合物が 1 万化合物毎に 1 つの mol2 ファイル(マルチ mol2 ファイル)となっています。そのため、1 つのジョブで 1 万化合物に対するドッキングを行うことができます。

本章では、4 コアを使って、4 万化合物を対象とした計算する例を紹介します。メインのドッキング計算に 1 時間～2 時間かかります。ほぼ同じ手順で 200 万化合物を対象とした計算も可能です。もし、4 万化合物の計算に約 2 時間かかる場合には、200 万化合物の計算には、25 倍の約 50 時間かかります。まずは、使う PC のコア数と同じ並列数(2 コアなら 2 並列で 2 万化合物、4 コアなら 4 並列で 4 万化合物)に対して、テスト計算をするとよいでしょう。

intel 製の CPU でハイパースレッディング機能が利用可能な CPU では、仮想的に 2 倍のコア数が利用可能な場合があります。その場合には、4 コアの場合でも、8 並列でジョブ投入すると計算効率が良い場合があります。

ジョブ終了は、タスクマネージャー(Windows の場合)やアクティビティモニタ(macOS の場合で CPU の負荷をモニターしていると分かりやすいかもしれません。

MTS 計算の手順(各コマンドに関する解説は、コマンド群の後にあります)

MTS-1% cp△–R△screening_org△screening_MTS_4HP0
(screening_org をコピーして用います)
MTS-2% cd△screening_MTS_4HP0
MTS-3% ln△–s△{LigandBOX ディレクトリ}/ligand△. (化合物の設定)
(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。
例: ln△-s△../../../screening_dataYYMMDD/ligand△.
ligand の後ろに’/’(スラッシュ)があるとリンクに失敗します。)
(次ページに続く)
MTS-4% cp△–R△../sample_data_4HP0/4HP0△protein/ (タンパク質の設定)

MTS-5% cd△base/

MTS-6% csh△bin/make_grid.csh　　(グリッド作成ジョブの作成)

MTS-7% perl△bin/submit_jobs.pl△joblist_grid△1△grid (ジョブの投入)

------ここで、少しだけジョブの終了待ちをします。------

MTS-8% echo△4HP0△>△list/target　　(リストファイル作成)

MTS-9% perl△bin/startDocking.pl△1△4△target (ドッキングジョブの作成)

MTS-10% perl△bin/submit_jobs.pl△joblist_docking△4△dock (ジョブの投入)

------ここで、ジョブの終了待ちをします。------

MTS-11% perl△bin/makeMatrix.pl△1△4△target (データ集計)

MTS-12% cp△../../ref_protein/pro_list△list/

MTS-13% cp△list/pro_list△list/pro_list_add

MTS-14% echo△4HP0△>>△list/pro_list_add　　(リストファイル作成)

MTS-15% ln△–s△{LigandBOX ディレクトリ}/mts_data△.

(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。

例:　ln△-s△../../../screening_dataYYMMDD/mts_data△.

mts_data の後ろに’/’(スラッシュ)があるとリンクに失敗します。)

MTS-16% perl△bin/run_group_MTS.pl△1△4△4HP0△pro_list_add

MTS-17% perl△bin/submit_jobs.pl△joblist_group_MTS△4△group_MTS　(ジョブ投入)

------ここで、ジョブの終了待ちをします。------　　(グループ毎の MTS 計算)

MTS-18% perl△bin/get_result_MTS_sort.pl△1△4△4HP0△200 >△ranking_list_MTS

ここで赤字の箇所(200)は、化合物数の 1/200 の値を設定します。

(統合したランキングリストの作成)

(以上、ランキングリストが ranking_list_MTS に出力されています。)

MTS-19% perl△bin/getCatalogCode.pl△ranking_list_MTS△>△ranking_list_MTS_id

この ranking_list_MTS_id がカタログコード情報付きのランキングリストです。

MTS-20% perl△bin/get_top_compounds.pl△ranking_list_MTS△50△top

(ranking_list_MTS_id でも可、50 は取得する化合物数、top は出力先ディレクトリ名)

MTS-21% mv△top/top△list/

MTS-22% cd△..

MTS-23% unlink△ligand

MTS-24% mkdir△ligand

MTS-25% mv△base/top△ligand/

MTS-26% cd△base

MTS-27% rm△joblist_docking

(次ページに続く)

MTS-28% csh△bin/make_docking_score.csh△target△top　(ドッキングジョブの作成)
MTS-29% perl△bin/submit_jobs.pl△joblist_docking△4△dock　(ジョブの投入)
------ここで、ジョブの終了待ちをします。------
MTS-30% perl△bin/get_struct.pl△target△top△top
(top/に上位化合物のドッキングポーズが出力されています。上位化合物のオリジナルの mol2 ファイルは screening_MTS_4HP0/ligand/top/に保存されています。)

(注意)

submit_jobs.pl の第 2 引数は、並列数です。使用する PC のコア数に合わせるとよいでしょう。intel 製の CPU の場合には、実際のコア数の 2 倍までは計算効率が上がる場合があります。第 3 引数は作成するジョブファイル名のタグなので、任意です。

MTS-18 の get_result_MTS_sort.pl では、第 4 引数で指定する順位までを sievgene スコアでソートしています。ここで指定する数字や、全体の化合物数の約 1/180～1/200 を指定してください。MTS 法自体は、化合物を約 1/180～1/200 に絞り込むための方法です。絞り込んだ化合物については、sievgene スコア順にすると上位にヒット化合物がきやすい傾向があります。

MTS-20 の get_top_compounds.pl の第 2 引数は、取得する上位化合物数です。例では 50 にしてあります。取得するドッキングポーズ数を変更する場合には、この数字を変更してください。

200 万化合物について計算する場合には、以下の箇所を変更します。
(submit_jobs.pl 以外で 4 となっている箇所を 200 にする)

MTS-9% perl△bin/startDocking.pl△1△200△target (ドッキングジョブの作成)
MTS-11% perl△bin/makeMatrix.pl△1△200△target (データ集計)
MTS-16% perl△bin/run_group_MTS_sort.pl△1△200△4HP0△pro_list_add
MTS-18% perl△bin/get_result_MTS_sort.pl△1△200△4HP0△10000 >△ranking_list_MTS

ranking_list_MTS_id の例(c001～c004 に対する計算):

Rank	Group	Compound code	Score	Target rank	SOURCE_ID
1	c003	HTS1508-03738796-01	4.1990	61	NS-08496426
2	c004	HTS1508-00390701-03	4.0111	59	NS-00542103
3	c001	HTS1508-04166307-01	3.9970	56	NS-09881771
4	c002	HTS1508-04714865-01	3.9901	33	NS-10819122
5	c004	HTS1508-04040051-02	3.9773	27	NS-09714739
6	c001	HTS1508-02505385-01	3.9752	63	NS-04695617
7	c003	HTS1508-04652671-01	3.9661	29	NS-10754275
8	c001	HTS1508-02044309-02	3.9561	47	NS-03166310
9	c004	HTS1508-04676732-01	3.9328	67	NS-10779444
10	c002	HTS1508-03107408-01	3.9308	67	NS-06369315

使用するシステム、コンパイラによって異なる場合があります。

9. ML-MTS 法の実行手順(screening_org_PC の場合)

ここでは、PC を使った ML-MTS 計算の手順について説明します。サーバーを使う場合には、6 章を参照してください。

ここでは、4 コアの PC を使う場合を想定して、4 万化合物について計算する例を紹介します。計算機の性能にもよりますが、4 万化合物についての計算は 4 コアで約 3 時間で計算が完了します。メインのドッキング計算は 1 時間～2 時間です。例えば、200 万化合物についても、ほぼ同じ手順で計算できます。その場合にはメインのドッキング計算部分が 25 倍になり、その部分が 2 日程度かかります。

まずは、サンプルファイルを用いて、使用する PC のコア数と同じ並列数でテスト計算するとよいでしょう。2 コアの場合には 2 並列で 2 万化合物に対して、4 コアの場合には 4 並列で 4 万化合物に対して、計算するとよいでしょう。intel 製の CPU の場合には、ハイパースレッディングが機能すれば、実際のコア数の 2 倍までは並列数を増やした場合に計算効率がよくなることがあります。

ML-MTS 計算の手順

ML-MTS-1% cp△–R△screening_org△screening_ML-MTS_4HP0
(screening_org をコピーして用いる)
ML-MTS-2% cd△screening_ML-MTS_4HP0
ML-MTS-3% rmdir△protein
ML-MTS-4% cp△–R△../ref_protein/protein△. (タンパク質の設定)
ML-MTS-5% cp△–R△../sample_data_4HP0/4HP0△protein/ (タンパク質の設定)
ML-MTS-6% mkdir△ligand
ML-MTS-7% cp△–R△../sample_data_4HP0/4HP0_virtual_hit△ligand/
(既知活性化合物の設定)
ML-MTS-8% cd△ligand/4HP0_virtual_hit
ML-MTS-9% ls△*.mol2△>△../../base/list/4HP0_virtual_hit
ML-MTS-10% cd△../../base
ML-MTS-11% csh△bin/make_grid.csh (グリッド作成ジョブの作成)
ML-MTS-12% perl△bin/submit_jobs.pl△joblist_grid△**4**△grid (ジョブの投入)
------ここで、ジョブの終了待ちをします。------
ML-MTS-13% cp△../../ref_protein/pro_list△list/ (リストファイル作成)
ML-MTS-14% cp△list/pro_list△list/pro_list_add (リストファイル作成)
ML-MTS-15% echo△**4HP0**△>>△list/pro_list_add
ML-MTS-16% csh△bin/make_docking_score.csh△pro_list_add△4HP0_virtual_hit
(次ページに続く)

ML-MTS-17% perl△bin/submit_jobs.pl△joblist_docking△4△dock (ジョブの投入)
(ドッキング 1: 既知活性化合物と 182 タンパク質)
------ここで、ジョブの終了待ちをします。------
ML-MTS-18% perl△bin/make_score_data.pl△pro_list_add△4HP0_virtual_hit (集計)
ML-MTS-19% cd△..
ML-MTS-20% mv△ligand△ligand2
ML-MTS-21% ln△–s△{LigandBOX ディレクトリ}/ligand△. (LigandBOX の設定)
(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。
例: ln△-s△../../../screening_dataYYMMDD/ligand△.
ligand の後ろに’/’(スラッシュ)があるとリンクに失敗します。)
ML-MTS-22% cd△base
ML-MTS-23% echo△4HP0△>△list/target
ML-MTS-24% rm joblist_docking
ML-MTS-25% perl△bin/startDocking.pl△1△4△target
ML-MTS-26% perl△bin/submit_jobs.pl△joblist_docking△4△dock (ジョブの投入)
(ドッキング 2: スクリーニング用 LigandBOX の化合物とターゲットタンパク質)
------ここで、ジョブの終了待ちをします。------
ML-MTS-27% perl△bin/makeMatrix.pl△1△4△target (データ集計)
ML-MTS-28% cd△matrix
ML-MTS-29% ls△pro_list_add_4HP0_virtual_hit.dat△>△../list/matrix_list
ML-MTS-30% cd△..
ML-MTS-31% ln△–s△{LigandBOX ディレクトリ}/mts_data△.
(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。
例: ln△-s△../../../screening_dataYYMMDD/mts_data△.
mts_data の後ろに’/’(スラッシュ)があるとリンクに失敗します。)
ML-MTS-32% cp△../../sample_data_4HP0/virtual_hit_list△list/
ML-MTS-33% perl△bin/run_group_ML-MTS.pl△1△4△4HP0△pro_list_add△virtual_hit_list△matrix_list (本来は１行のコマンドですが、２行になっています。)
ML-MTS-34% perl△bin/submit_jobs.pl△joblist_group_ML-MTS△4△group_ML-MTS
(グループ毎の MTS 計算)
------ここで、ジョブの終了待ちをします。------ (次ページに続く)

ML-MTS-35% perl△bin/auto_make_matrix_selected_ML-MTS.pl△1△4△virtual_hit_list△4HP0

ML-MTS-36% perl△bin/submit_jobs.pl△joblist_auto_make_ML-MTS△4△auto_make_ML-MTS

------ここで、ジョブの終了待ちをします。------

ML-MTS-37% perl△bin/merge_matrix.pl△1△4△work/4HP0_group_MTS△matrix_united.dat

(グループの上位化合物の相互作用行列を作成する)

ML-MTS-38% ls△matrix_united.dat△>>△list/matrix_list

ML-MTS-39% mv△matrix_united.dat matrix

ML-MTS-40% perl△bin/run_selected_ML-MTS.pl△4HP0△pro_list_add△virtual_hit_list△matrix_list (本来は１行のコマンドですが、２行になっています。) (選抜の ML-MTS 計算)

ML-MTS-41% perl△bin/submit_jobs.pl△joblist_select_ML-MTS△4△select_ML-MTS

------ここで、ジョブの終了待ちをします。------

ML-MTS-42% perl △ bin/remove_active.pl △ work/4HP0_selected_MTS/comp_list_MTS △ list/virtual_hit_list△ranking_list_ML-MTS

(本来は１行のコマンドですが、２行になっています。)

ML-MTS-43% perl△bin/modify_ranking_list.pl△ranking_list_ML-MTS△ work/4HP0_group_MTS△ML-MTS△>△ranking_list_ML-MTS_mod

(本来は１行のコマンドですが、２行になっています。)

ML-MTS-44% perl△bin/get_top_compounds.pl△ranking_list_ML-MTS_mod△50△top

ML-MTS-45% mv△top/top△list

ML-MTS-46% cd△..

ML-MTS-47% unlink△ligand

ML-MTS-48% mkdir△ligand

ML-MTS-49% mv△base/top△ligand/

ML-MTS-50% cd△base

ML-MTS-51% rm△joblist_docking

ML-MTS-52% csh△bin/make_docking_score.csh△target△top (ドッキングジョブの作成)

ML-MTS-53% perl△bin/submit_jobs.pl△joblist_docking△4△dock (ジョブの投入)

------ここで、ジョブの終了待ちをします。------

ML-MTS-54% perl△bin/get_struct.pl△target△top△top

(top/に上位化合物のドッキングポーズが出力されています。)

カタログコードの情報が必要な場合には、以下のコマンドを実行してください。

```
ML-MTS-55% cd△..
ML-MTS-56% mv△ligand△ligand3
ML-MTS-57% ln△–s△{LigandBOX ディレクトリ}/ligand△.
(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。
例:  ln△-s△../../../screening_dataYYMMDD/ligand△.
ligand の後ろに’/’(スラッシュ)があるとリンクに失敗します。)
ML-MTS-58% cd△base
ML-MTS-59% perl△bin/getCatalogCode.pl△ranking_list_ML-MTS_mod△>
△ranking_list_ML-MTS_mod_id
                                    (本来は１行のコマンドですが、２行になっています)
```

(注意)

getCatalogCode.pl を実行する際に、引数で渡すランキングリストファイルは、ML-MTS-43 で実行する modify_ranking_list.pl の出力を指定してください。つまり、左から 2 列目のカラムがグループ名(c001～c200)になっている必要があります。また、screening_ML–MTS_4HP0/ligand が、screening_dataYYMMDD/ligand へのシンボリックリンクになっている必要があります。

ranking_list_ML-MTS_mod_id の例(c001～c004 に対する計算):

```
     1    c003 KSH2016-02782481-01    3.6665    67
     2    c001 KSH2016-02663150-02    3.9901    68
     3    c002 KSH2016-03482975-02    4.0240    68
     4    c001 KSH2016-03981691-01    3.5740    69
     5    c001 KSH2016-04001152-03    3.8602    71
     6    c001 KSH2016-02706799-02    3.8037    72
     7    c003 KSH2016-02998714-01    4.0719    73
     8    c003 KSH2016-03249774-02    3.9838    73
     9    c004 KSH2016-02786915-02    3.8198    73
    10    c001 KSH2016-02790708-01    3.7248    73
(以下省略)
```

使用するシステム、コンパイラによって異なる場合があります。

10.　DSI 法の実行手順(screening_org_PC の場合)

ここでは、PC を使った ML-MTS 計算の手順について説明します。サーバーを使う場合には、7 章を参照してください。

DSI 法では、ターゲットタンパク質を使いません。既知活性化合物とリファレンスタンパク質とのドッキングを行い、既知活性化合物とドッキングプロファイルが近いものを選出します。ターゲットタンパク質と化合物ライブラリとのドッキング計算が不要で計算量が比較的小さいので、4 コア程度の PC でも 200 万化合物に対する計算が可能です。以下では 200 万化合物を対象とした計算手順について説明します。

DSI 計算の手順

DSI-1% cp△–R△screening_org△screening_DSI_**4HP0**
(screening_org をコピーして用いる)
DSI-2% cd△screening_DSI_**4HP0**
DSI-3% rmdir△protein
DSI-4% cp△–R△../ref_protein/protein△. (タンパク質の設定)
DSI-5% mkdir△ligand
DSI-6% cp△–R△../sample_data_4HP0/4HP0_virtual_hit△ligand/
(既知活性化合物の設定)
DSI-7% cd△ligand/4HP0_virtual_hit
DSI-8% ls△*.mol2△>△../../base/list/4HP0_virtual_hit
DSI-9% cd△../../base/
DSI-10% csh△bin/make_grid.csh　(グリッド作成ジョブの作成)
DSI-11% perl△bin/submit_jobs.pl△joblist_grid△**4**△grid　(ジョブの投入)
------ここで、ジョブの終了待ちをします。------
DSI-12% cp△../../ref_protein/pro_list△list/　(リストファイル作成)
DSI-13% csh△bin/make_docking_score.csh△pro_list△**4HP0_virtual_hit**　(ドッキング)
DSI-14% perl△bin/submit_jobs.pl△joblist_docking△**4**△dock
------ここで、ジョブの終了待ちをします。------
DSI-15% perl△bin/make_score_data.pl△pro_list△4HP0_virtual_hit　(集計)
DSI-16% cd△matrix
DSI-17% ls△pro_list_4HP0_virtual_hit.dat△>△../list/matrix_list
DSI-18% cd△..
DSI-19% cp△../../sample_data_4HP0/virtual_hit_list△list/
(次ページに続く)

DSI-20% ln△–s△{LigandBOX ディレクトリ}/mts_data△.

(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。

例:　ln△-s△../../../screening_dataYYMMDD/mts_data△.

mts_data の後ろに’/’(スラッシュ)があるとリンクに失敗します。)

DSI-21% perl△bin/run_group_DSI.pl△1△4△pro_list△virtual_hit_list△matrix_list

DSI-22% perl△bin/submit_jobs.pl△joblist_group_DSI△4△group_DSI　(ジョブの投入)

(グループ毎の DSI 計算)

------ここで、ジョブの終了待ちをします。------

DSI-23% perl△bin/auto_make_matrix_selected_DSI.pl△1△4△virtual_hit_list

DSI-24% perl△bin/submit_jobs.pl△joblist_auto_make_DSI△4△auto_make_DSI

(ジョブの投入)

------ここで、ジョブの終了待ちをします。-----

DSI-25% perl△bin/merge_matrix.pl△1△4△work/group_DSI△matrix_merged.dat

------このコマンドは時間がかかります。------

DSI-26% ls△matrix_merged.dat△>>△list/matrix_list

DSI-27% mv△matrix_merged.dat△matrix

DSI-28% perl△bin/run_selected_DSI.pl△pro_list△virtual_hit_list△matrix_list

DSI-29% perl△bin/submit_jobs.pl△joblist_select_DSI△4△select_DSI　(ジョブの投入)

(選抜した化合物に対する DSI 計算)

------ここで、ジョブの終了待ちをします。------

DSI-30%　perl △ bin/remove_active.pl △ work/selected_DSI/comp_list_PCA △ list/virtual_hit_list △ ranking_list_DSI

2 行になっていますが、1 行で実行します。

(ここで得られた ranking_list_DSI が DSI 法によるランキングリストです。学習用に使用した化合物については、removed_from_comp_list に出力されています。)

DSI-31%　perl△bin/modify_ranking_list.pl△ranking_list_DSI△work/group_DSI△DSI△>△ranking_list_DSI_mod

DSI-32% cd△..

DSI-33% mv△ligand△ligand2

DSI-34% ln△–s△{LigandBOX ディレクトリ}/ligand△.

(注意: {LigandBOX ディレクトリ}には、パス付きでディレクトリ名を記述します。

例:　ln△-s△../../../screening_dataYYMMDD/ligand△.

ligand の後ろに’/’(スラッシュ)があるとリンクに失敗します。)

(次ページに続く)

```
DSI-35% cd△base
DSI-36% perl△bin/get_top_compounds_with_rank.pl△ranking_list_DSI_mod△50△top
```
(top/に上位化合物の mol2 ファイルが出力されています。DSI 法では、ターゲットタンパク質とのドッキングを行っていませんので、出力された mol2 ファイルは、MTS 法や ML-MTS 法のものとは異なり、ドッキングポーズではありません。計算の際に使用したマルチ mol2 ファイルから該当箇所を切り出したものです。)

(注意)

submit_jobs.pl の第 2 引数は、並列数です。使用する PC のコア数に合わせるとよいでしょう。intel 製の CPU の場合には、実際のコア数の 2 倍までは計算効率が上がる場合があります。第 3 引数は作成するジョブファイル名のタグなので任意です。

DSI-36 の get_top_compounds_with_rank.pl の第 2 引数は、取得する上位化合物数です。例では 50 にしてあります。取得したい化合物数を変更する場合には、この数字を変更してください。第 3 引数は出力先のディレクトリ名です。

ナミキ ID が必要な場合には、以下のコマンドを実行してください。

```
% perl△bin/getCatalogCode.pl△ranking_list_DSI_mod△>△ranking_list_DSI_mod_id
```

(注意)

このコマンドを実行する際に、引数で渡すランキングリストファイルは、DSI-31 で実行する modify_ranking_list.pl の出力を指定してください。また、screening_DSI_4HP0/ligand が、screening_dataYYMMDD/ligand へのシンボリックリンクになっている必要があります。

ranking_list_DSI_mod_id の出力例:

```
     1    c033 HTS1508-03567913-01    0.9540    NS-07201437
     2    c072 HTS1508-03568003-02    1.2106    NS-07201529
     3    c073 HTS1508-01160905-01    1.4037    NS-01758482
     4    c104 HTS1508-04347880-02    1.4065    NS-10317507
     5    c014 HTS1508-03027203-01    1.4093    NS-06115298
     6    c033 HTS1508-04238066-04    1.4368    NS-10146393
     7    c014 HTS1508-02467393-02    1.4565    NS-04274810
     8    c009 HTS1508-02430009-01    1.4566    NS-04056608
     9    c038 HTS1508-00336414-01    1.4634    NS-00481628
    10    c136 HTS1508-00541855-01    1.5046    NS-00717407
(以下省略)
```

使用するシステム、コンパイラによって異なる場合があります。

11.　コマンドの説明

make_grid.csh

グリッド作成を行うプログラム。protein/の下に用意されているタンパク質に対して、グリッドが既に作成されているかどうかを自動的に判断して、グリッドがないものに対して、新たにグリッドを作成します。

make_docking_score.csh

２つの引数をとり、多くのドッキングを自動的に実行してくれます。２つの引数は、それぞれ、タンパク質のリストファイル名と、化合物のリストファイル名で、これらのリストファイルは、list/の下にあることを想定しています。例えば、

% csh△bin/make_dockin_score.csh△target△c001

と実行した場合、target に含まれるタンパク質と c001 に含まれる化合物とのドッキングを自動的に実行します。target が１つで c001 が１万化合物の場合には、このコマンド１回で１万組のドッキングが実行されます。タンパク質を複数登録したリストを指定することもできます。

化合物のリストファイル名は、ligand/の下のディレクトリ名と同じでないといけません。

startDocking.pl

このプログラムは、スクリーニング用の LigandBOX(c001～c200)を使う場合に使用します。３つの引数をとり、最初の２つが c001～c200 の中で何番から何番までを計算するかを指定する引数で、３つ目の引数は、タンパク質のリストファイル名を指定します。内部で make_docking_score.csh を実行しています。

make_score_data.pl

ドッキング終了後に、ドッキング結果を相互作用行列にするコマンドです。これまで使われてきた make_docking_score.csh と同じようにタンパク質のリストと化合物のリストを引数にとります。

makeMatrix.pl

これは、startDocking.pl で実行したデータセットに対して相互作用行列を作成するプログラムです。

run_group_MTS.pl

１万化合物毎に 200 グループに対して MTS 計算を実行します。

get_result_MTS.pl

run_group_MTS.pl 実行後に、各グループから上位 100 個を集めて、集めた 2 万個の中で、まず、MTS 法で上位 1 万位を決定し、その後に、上位 1 万個の化合物を、sievgene スコアで再ソートします。

run_group_ML-MTS.pl

MTS 法を各グループで実行します。

auto_make_matrix_selected_ML-MTS.pl

MTS で選んだ化合物の相互作用行列を作成します。

merge_matrix.pl

複数のグループの相互作用行列を 1 つにします。

run_selected_ML-MTS.pl

各グループから選出した化合物を集めて、それに対して再度 ML-MTS 計算を実行します。

run_group_DSI.pl

DSI 法を各グループで実行します。

auto_make_matrix_selected_DSI.pl

DSI で選んだ化合物の相互作用行列を作成します。

run_selected_DSI.pl

各グループから選出した化合物を集めて、それに対して再度 DSI 計算を実行します。

12. 過去のプログラムセットとの違い

make_score_data.csh, make_score_data_multi.pl, DataMaker は本チュートリアルでは使用していません。また、マルチ mol2 とシングル mol2 ファイルで、ほぼ同じように扱える方法を使用しています。

13. 注意事項

myPresto のプログラムの一部は、LSF を使用して、計算ジョブ投入を実行します。そのため、LSF を使っていない計算機では、そのままでは動作しないことがあります。