Ensemble docking screening with Universal Active Prove(UAP)

USER MANUAL

Version 1.0

目次

1　　概要

本設計書では、Universal Active Prove (UAP)を化合物に用いたスクリーニングシステムについて記述する。本システムは、以下の 7 つのフェーズからなる。

A) sievgene 実行環境作成フェーズ
cosgene の実行結果から標的蛋白質データを作成し、sievgene を実行するための環境を構築する。

B) 標的蛋白質選別のための sievgene 実行フェーズ
sievgene を実行し、相互作用行列における**エラー! 参照元が見つかりません。**②～④の部分を作成する。ここで、③の部分は、2～3 万低分子化合物×追加蛋白質 M 種の作成を行う。

C) 標的蛋白質選別のための MTS 法グループスクリーニング実行フェーズ
MTS 法グループスクリーニングの結果から標的蛋白質を少数に選別する。

D) 標的蛋白質を用いた sievgene 実行フェーズ
200 万低分子化合物×選別した標的蛋白質で③を作成する。

E) 標的蛋白質を用いた MTS 法グループスクリーニング実行フェーズ
200 万低分子化合物と選別した標的蛋白質を用いて MTS 法グループスクリーニングを実行する。

F) MTS 法総合スクリーニング実行フェーズ
E)のグループスクリーニングの結果得られる各化合物グループの上位を用いて、全化合物グループでの MTS 法スクリーニングを実行する。

G) ドッキングポーズの再計算フェーズ
F)の結果得られた有力ドッキング化合物のドッキングポーズを再計算する。

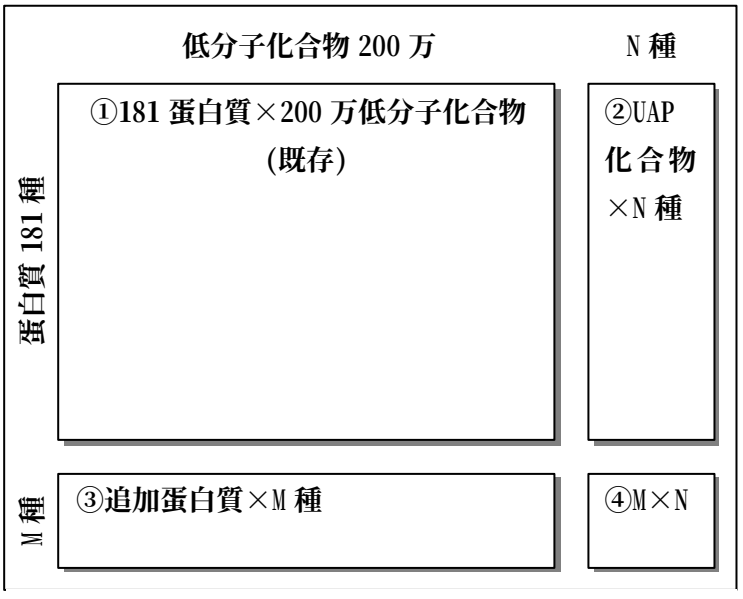

図 1-1　相互作用行列

2　　前提条件

本システムは、in silico スクリーニングシステムを用いて低分子 200 万×蛋白質 181 の相互作用行列が必要である。

システム全体の処理のフローを図 2-1 に示す。

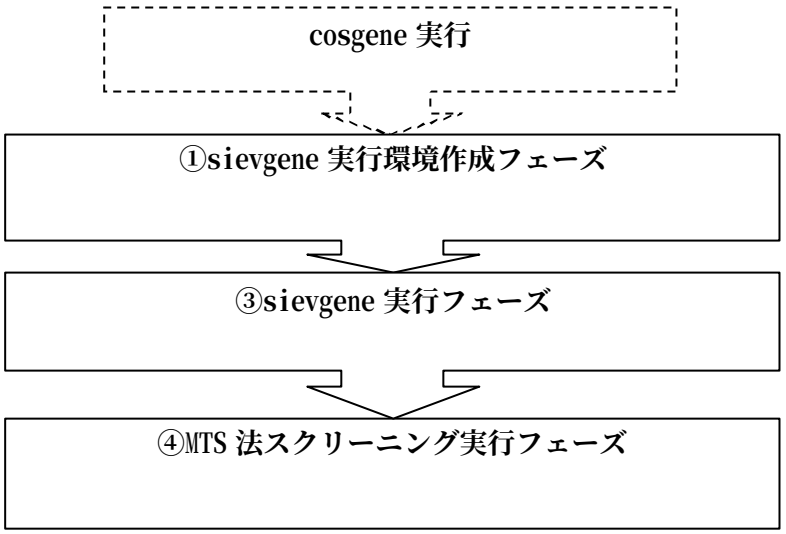

図 2-1　システム全体のフロー

システムの初期状態のディレクトリ構成を図 2-2 に示す。相互作用行列作成用ディレクトリ構成と同じ状態である。以後、作業を進めるに従い、ディレクトリを追加する。

低分子 200 万×蛋白質 181 の相互作用行列(**エラー！参照元が見つかりません。**①)を作成したときと同じディレクトリを使用する必要はないが、ligand ディレクトリ、protein ディレクトリ、grid ディレクトリ、input ディレクトリは同じファイルを使用するため、シンボリックリンクを張るなどしておくと良い。

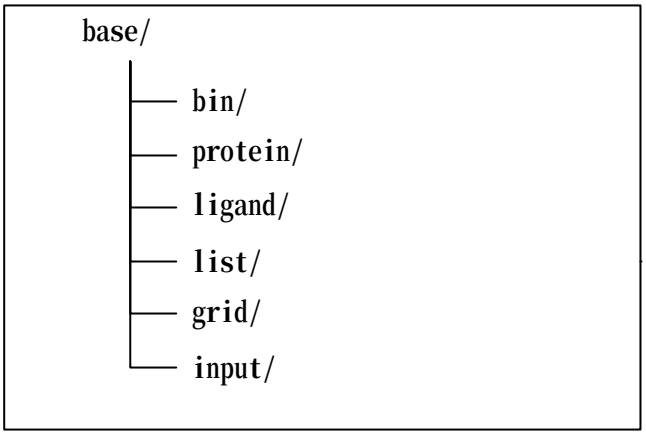

図 2-2　ディレクトリ構成

3　　sievgene 実行環境作成フェーズ

3.1　　概要

cosgene の実行結果を用いて、標的蛋白質データと sievgene 実行のための環境を作成する。下の順序で行う。

① 各 cosgene 実行ディレクトリで、トラジェクトリファイルから PDB ファイルを生成する。

② ドッキングジョブで使用する標的蛋白質として使用する PDB ファイルのリストを作成する。

③ ②で作成したリストを元に、protein ディレクトリへ蛋白質データの配置を行い、list ディレクトリに蛋白質リストを作成する。

3.2　　ディレクトリ構成

ディレクトリ構成を図 3-1 に示す。図 3-1 のディレクトリ構成に md ディレクトリを追加し、さらに md ディレクトリに cosgene を実行する各蛋白質のディレクトリを配置する。

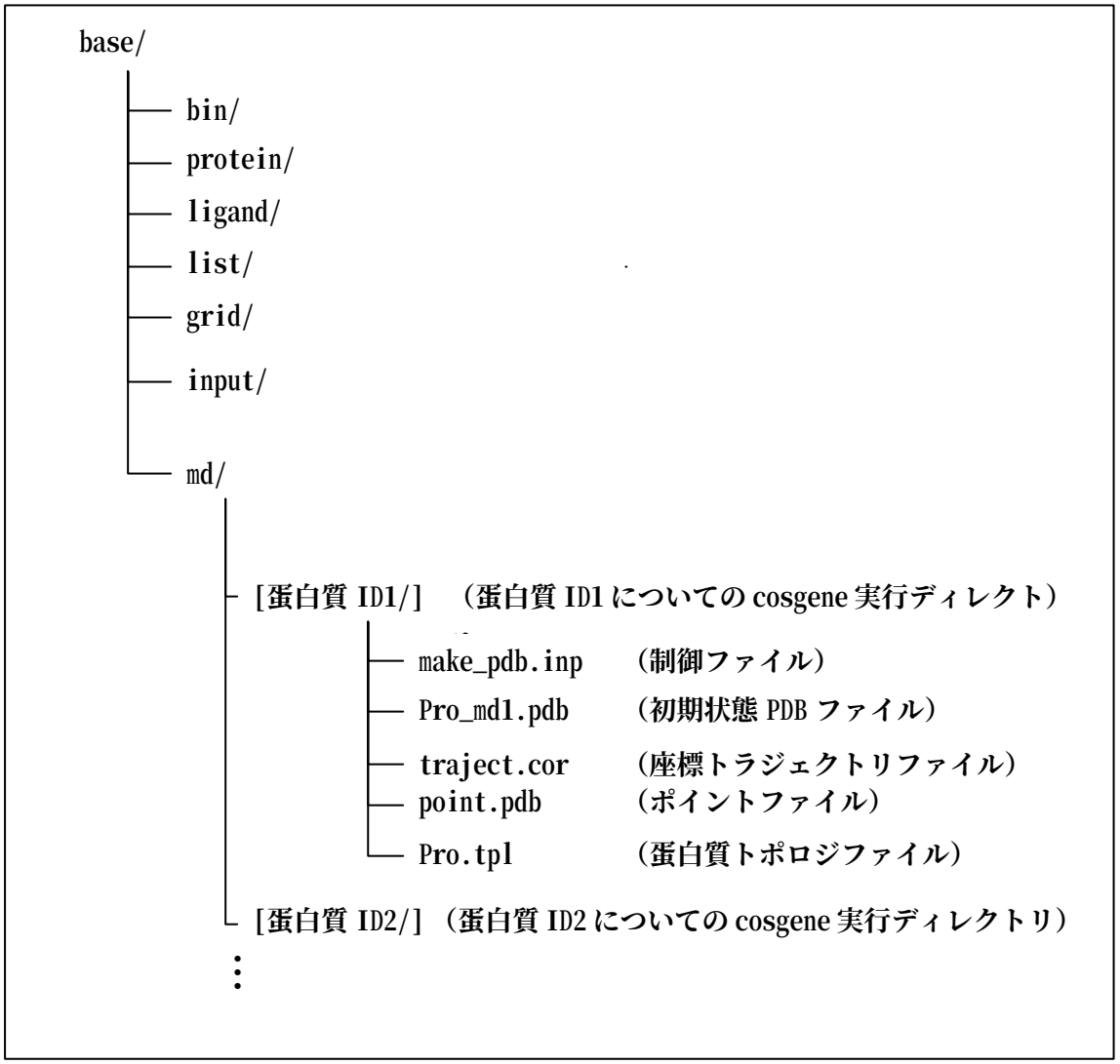

図 3-1　sievgene 実行環境作成フェーズのディレクトリ構成

md 配下のディレクトリの各[*蛋白質 ID*]ディレクトリで cosgene を実行し、ファイルを同ディレクトリに作成した状態を初期配置とする。

cosgene 実行後に生成されるファイルの内、必須のものを表 3-1 に示す。

表 3-1　md/蛋白質ディレクトリに準備するファイル一覧

#	ファイル名	用途
1	traject.cor	cosgene 実行により生成される座標トラジェクトリファイル。
2	Pro.tpl	基質、溶媒分子を取り除いた蛋白質トポロジファイル。
3	Pro_md1.pdb	cosgene 実行時に使用した蛋白質の初期状態 PDB ファイル。
4	point.pdb	蛋白質のドッキングポケットの座標データとなるポイントファイル
5	make_pdb.inp	traject.cor から PDB ファイルを生成するための制御ファイル。
6	pro.list	traject.cor から生成した PDB ファイルの内、ドッキングジョブに使用する PDB ファイル名を記述したファイル。

3.3　　ファイル入出力

md/蛋白質ディレクトリのファイル入出力を図 3-2 に示す。

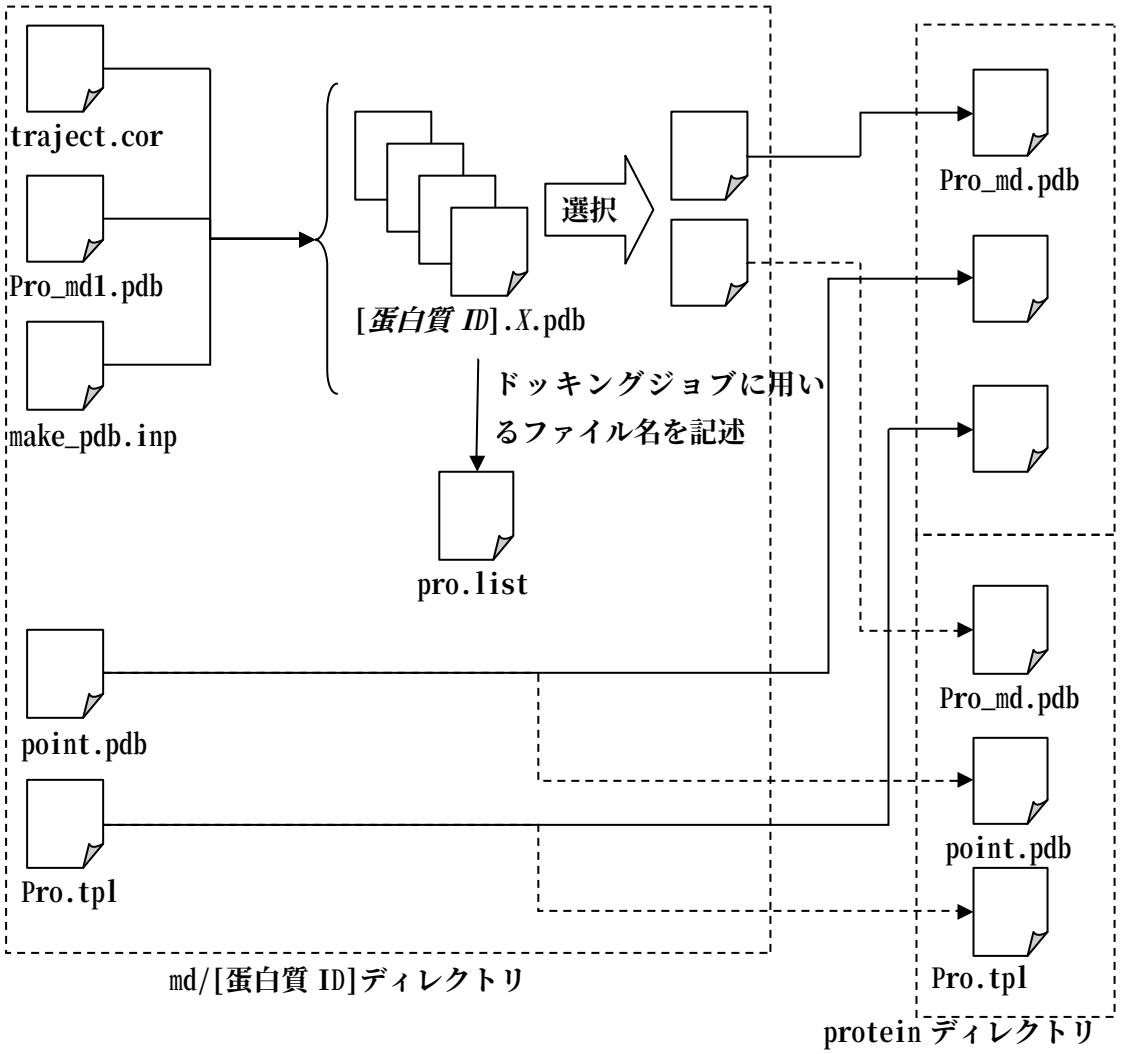

図 3-2　md/蛋白質 ID ディレクトリのファイル入出力

3.4　　プログラムの準備

sievgene 実行環境作成フェーズで使用するプログラムを表 3-2 に示す。これらプログラムは、**エラー！参照元が見つかりません。**の bin ディレクトリに格納する。

エラー！参照元が見つかりません。 の内、make_grid.csh と sievgene は、相互作用行列作成用プログラムとして既存のものを用いる。特に sievgene は、低分子 200 万×蛋白質 181 の相互作用行列(**エラー！参照元が見つかりません。**①)を作成したものと同じプログラムを使用すること。

trj2pdb、extract_pdb.pl、prepare_ proteins.pl については後述する。

表 3-2　sievgene 実行環境作成に必要なプログラム一覧

#	プログラム名	用途
1	make_grid.csh	グリッドファイルを作成するスクリプト
2	sievgene	make_grid.csh から呼ばれ、グリッドファイルを生成する。
3	trj2pdb	トラジェクトリファイルから PDB ファイルを生成する。
4	extract_pdb.pl	trj2pdb を実行するスクリプト。
5	prepare_ protein.pl	追加する標的蛋白質を protein ディレクトリに配置するスクリプト。

3.5　　制御ファイル準備

base/md/[*蛋白質 ID*]/make_pdb.inp は、トラジェクトリファイルから PDB ファイルを作成する際の制御情報が記述されたファイルであり、事前に作成する。制御ファイルのフォーマットを図 3-3 に示す。

```
蛋白質 ID                    (蛋白質名)
Pro_md1.pdb                  (座標トラジェクトリファイルから生成する PDB の数)
traject.cor                  (初期配置 PDB ファイル)
50                           (座標トラジェクトリファイル)
4010                         (座標トラジェクトリファイルの蛋白質原子数)
s                            (座標トラジェクトリファイルのデータ形式)
```

図 3-3　制御ファイルの例

初期配置 PDB ファイルは、PDB ファイルのフォーマット情報を得するために参照する。座標トラジェクトリファイルからは、PDB ファイルの座標データが得られるため、初期配置 PDB ファイルの座標データと入れ替えて、PDB ファイルとして作成する。

座標トラジェクトリファイルは、バイナリデータ 2 種類をサポートする。指定方法を表 3-3 に示す。

表 3-3　制御ファイルで指定するトラジェクトリファイルのデータ形式

#	データ形式	指定方法
1	バイナリ 4 バイト	s
2	バイナリ 8 バイト	d

トラジェクトリファイルから生成する PDB の数は、トラジェクトリに含まれる構造の数より多くてもかまわないが、その場合はトラジェクトリに含まれる構造の数分の出力しか行われない。

3.6　PDB 生成コマンドの実行

cosgene 実行によって作成されたトラジェクトリファイルから、指定した個数分の PDB ファイルを作成する。PDB ファイルの作成は、base ディレクトリで以下のスクリプトを実行する。

```
>> ./bin/extract_pdb.pl  [蛋白質 ID]
```

スクリプトを実行すると、コマンドライン引数で指定した[*蛋白質 ID*]について、base/md/[*蛋白質 ID*]/make_pdb.inp が読み込まれ、base/md/[*蛋白質 ID*]ディレクトリに PDB ファイルが作成される。base/md/[*蛋白質 ID*]/make_pdb.inp が無い場合はエラーで終了する。

作成された N 個分の PDB ファイルは、以下に示すように“蛋白質 ID”をプレフィックスにもち、インデックスが付与されたファイル名となる。

また、extract_pdb.pl スクリプトは、3.11 で説明する trj2pdb プログラムを呼び出すスクリプトである。

***蛋白質 ID*.*X*.pdb　（X：1～N）**

3.7　　時系列蛋白質リストファイルの作成

sievgene 実行に用いる蛋白質の PDB ファイルを用意したら、その PDB ファイルのリストを base/md/[蛋白質 ID]ディレクトリに作成する。フォーマットは、base/list ディレクトリに作成する蛋白質リストファイルと同じである。図 3-4 に例を示す。

リストには初期構造 PDB や最終構造 PDB を含んでも良いが、base/md/[蛋白質 ID]ディレクトリに該当する PDB ファイルが存在していなければならない。ここでリストされたものは、後のドッキングジョブでそれぞれ個別の蛋白質として実行される。

[*蛋白質 ID*].1.pdb
[*蛋白質 ID*].8.pdb
[*蛋白質 ID*].15.pdb
・・・{省略}・・・

図 3-4　時系列蛋白質リストファイルの例

3.8　　sievgene 実行環境の作成

各 base/md/[*蛋白質 ID*]ディレクトリで PDB ファイル及び蛋白質リストファイルを作成したら、以下のスクリプトを実行して protein ディレクトリに 3.7 でリストした蛋白質を配置する。

>> bin/prepare_ protein.pl　[*蛋白質 ID*]　[*蛋白質リストファイル*]

スクリプトを実行すると、base/md/[*蛋白質 ID*]/[*蛋白質リストファイル*]が読み込まれ、”base/protein/[*蛋白質 ID*]/”ディレクトリが作成され、さらに必要なファイルがコピーされる。作成されたディレクトリ構成を図 3-5 に示す。

ディレクトリが作成されると、その中にトポロジファイル、ポイントファイル、蛋白質 PDB ファイルがコピーされる。蛋白質 PDB ファイルは、蛋白質リストファイルに書かれている蛋白質 PDB ファイルをコピーした後、Pro_md.pdb にリネームされる。

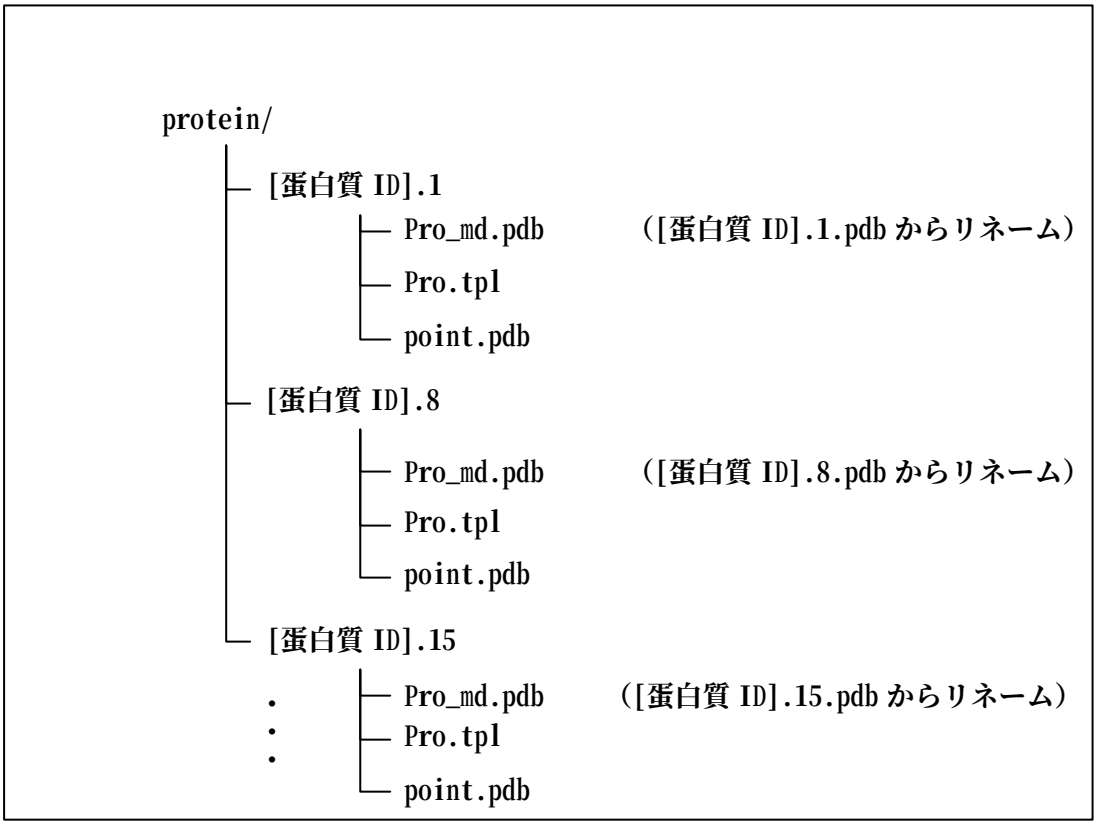

図 3-5　prepare_md_protein.pl スクリプト実行後の protein ディレクトリの構成

3.9　追加蛋白質リストファイルの作成

base/list ディレクトリに、今回追加となった蛋白質を記述したリストファイルを作成する。リストファイルは、3.7 で作成した蛋白質リストファイルを使用することができる。複数の[蛋白質 ID]ディレクトリを作成した場合には、それぞれの蛋白質リストファイルを連結して 1 つにまとめたものを使用する。

3.10　UAP 化合物データの配置

UAP 化合物データは、相互作用行列作成時に作成した ligand ディレクトリに UAP のディレクトリを作成し、mol2 ファイルを格納する。さらに、list ディレクトリに、sievgene 実行対象の化合物リストを作成する。UAP 化合物の数が多い場合は、マルチ mol2 ファイルにする。

3.11　トラジェクトリファイルから PDB ファイルを生成するコマンド

3.11.1　概要

cosgeneを実行した後に生成されるトラジェクトリファイルから指定した個数分のPDBファイルを生成する（以下、PDB ファイル生成コマンドと呼ぶ）。

3.11.2　動作イメージ

動作イメージを図 3-6 に示す。

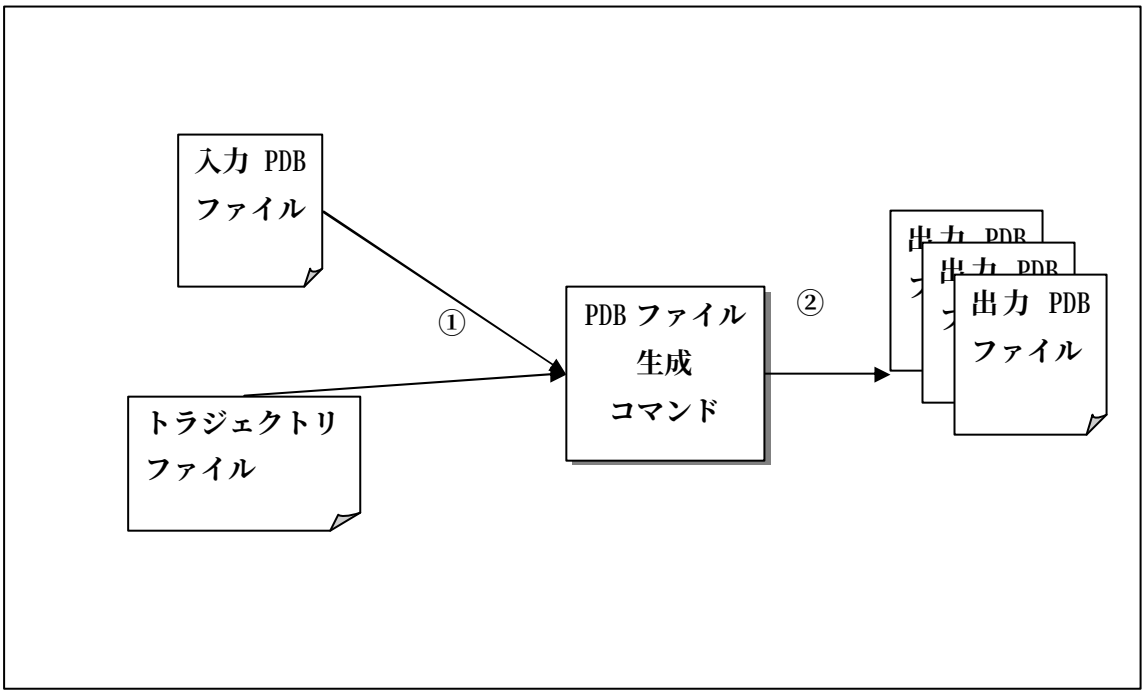

図 3-6　PDB ファイル生成コマンドの動作イメージ図

① PDB ファイル生成コマンドを実行し、入力 PDB ファイルとトラジェクトリファイルを読み込む。

② 出力 PDB ファイルを指定した数だけ出力する。

3.11.3　処理フロー

PDB 生成コマンドの処理フローを図 3-7 に示し、説明を後述する。

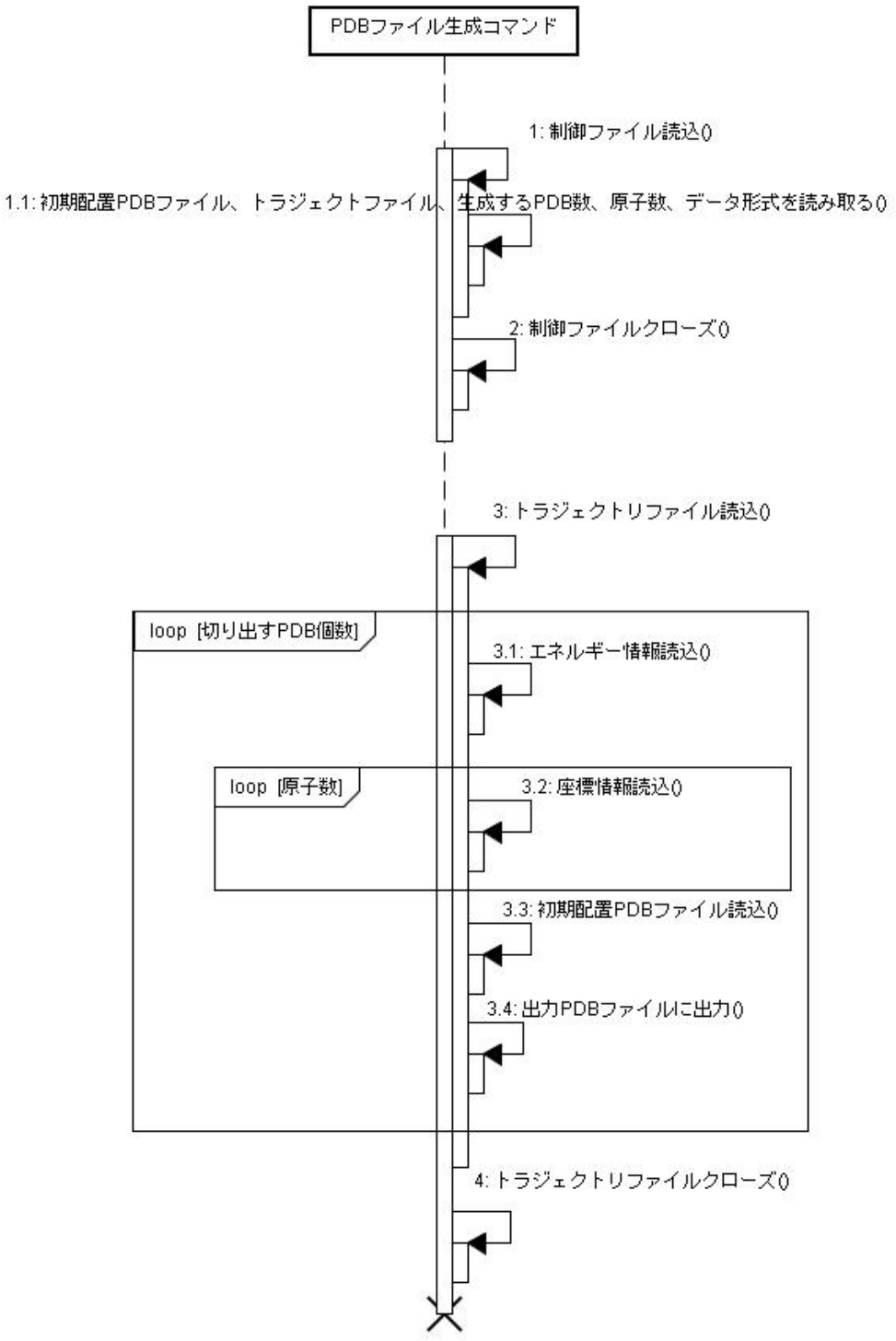

図 3-7　PDB 生成コマンドの処理フロー

① PDB 生成コマンドは、制御ファイルを読み込む
② 制御ファイルをクローズする。
③ 入力データとなる座標トラジェクトリファイルを読み込み、エネルギー情報の 11 種類のパラメータを読み込む。
④ トラジェクトリファイルの座標情報（ｘ、ｙ、ｚ）を原子数分読み込む。
⑤ 初期配置 PDB ファイルを読み込み、座標トラジェクトリファイルから読み取った座標情報に置換して新規 PDB ファイルとして出力する
⑥ ③～⑤を、生成する PDB ファイルの個数分繰り返す。
⑦ PDB ファイルの個数分の生成が終了したか、座標トラジェクトリファイルのデータが全て読み込まれた状態になったら、座標トラジェクトリファイルをクローズして終了する。

3.11.4 出力

正常終了すると、指定した個数分の PDB ファイルを生成する。ただし、トラジェクトリファイルの構造数以上を指定した場合は、トラジェクトリファイルのデータ数の最大数個分が出力される。

3.11.5 開始条件

コマンドライン実行により開始する。

3.11.6 終了条件

正常終了もしくは異常終了で終了する。

4 標的蛋白質選別のための sievgene 実行フェーズ

4.1 概要

本フェーズは、cosgene の実行結果から選択した蛋白質と UAP を使用して sievgene を実行する。

cosgene の実行結果から選択した蛋白質が多い場合(仮に数十個程度とする)と、全ての蛋白質で 200 万低分子化合物に対する相互作用行列を作成するのは時間やディスク容量を圧迫する。そのため、数万低分子化合物を対象とした小規模の相互作用行列を作成し、MTS 法スクリーニングを行って、10 個程度に蛋白質を選別する。

4.2 方法

低分子化合物グループ c001～c002 と cosgene の実行結果から選択した全蛋白質を用いて相互作用行列を作成する(**エラー! 参照元が見つかりません。**①)。また、②④については

全 UAP 化合物と 181 蛋白質及び cosgene の実行結果から選択した全蛋白質を用いて相互作用行列を作成する。

作成の方法は、in silico スクリーニングシステムの操作説明書を参照のこと。

4.3 sievgene 実行結果確認

sievgene 実行結果確認用プログラム一覧を、表 4-1 に示す。

表 4-1 sievgene 実行結果確認用プログラム

#	プログラム名	用途
1	checkExited.pl	sievgene終了ステータスが異常終了しているジョブについて出力する。
2	get_failed_score _compound_file.pl	スコア出力が行われなかった mol2 ファイルを回収する。checkError.pl と gather_compound_file.pl のラッパースクリプト。
3	checkError.pl	スコア出力が行われなかった蛋白質と化合物のリストを作成する。
4	gather_compound_file.pl	3 で作成したリストに記載された mol2 ファイルを回収する。

4.3.1 sievgene 終了ステータスの確認

sievgene 実行用スクリプト make_docking_score.csh 及び make_docking_score_multi.pl は、バッチジョブとして sievgene を実行する。何らかの理由により、sievgene ジョブが異常終了した場合、どの化合物と蛋白質の組み合わせで異常終了したのかを特定し、以降、その化合物は使用しないなどの対応が必要となる。ここではバッチシステム LSF を対象とし、異常終了確認するために、base ディレクトリで以下のプログラムを実行する。

```
./bin/checkExited.pl
```

プログラムを実行すると、base/work ディレクトリ以下にあるジョブのログから異常終了のあったログを検索する。異常終了が確認された場合、標準出力に図 4-1 に示すメッセージを出力する。

図 4-1 では、異常終了のあったログのヘッダーから、計算機、ジョブ ID、ジョブ開始日時、ジョブ終了日時、終了コードが 1 行で出力される。

```
Sender: LSF System <lsfadmin@zunou212>,Subject: Job 503847:
<D_208J_init1_c081-00001> Exited,Started at Wed Jul  7 22:42:30 2010,Results
reported at Wed Jul  7 23:56:48 2010,Exited with exit code 174.
Sender: LSF System <lsfadmin@zunou235>,Subject: Job 503421:
<D_3K5K_apo_4_c039-00001> Exited,Started at Wed Jul  7 20:56:57 2010,Results
reported at Wed Jul  7 21:17:54 2010,Exited with exit code 174.
```

図 4-1　sievgene で異常終了があったログを検出したときの標準出力例

4.3.2　sievgene エラー化合物の特定と回収

sievgene 実行で、スコアの出力に失敗した場合に、その失敗した化合物の mol2 ファイルを回収する場合に、base ディレクトリで以下のプログラムを実行する。

./bin/get_failed_score_compound_file.pl

上記スクリプトは、result ディレクトリ配下にあるスコアファイルから、スコアが出力されていない化合物と蛋白質の組み合わせを検索し、さらに化合物のファイルを回収する。

スクリプトを実行すると、result ディレクトリに failed_score ディレクトリが作成される。

以下の 2 つのプログラムを実行するためのラッパースクリプトである。

① スコア出力に失敗した化合物と蛋白質の mol2 ファイル名のリストを作成する。

リストは、failed_score_compound_list というファイル名で result ディレクトリの failed_score ディレクトリに格納される。リストの例を図 4-2 に示す。

```
12as,cUAP_dud,conf1.mol2,../../../../ligand/cUAP_dud/1ldm.mol2
12as,cUAP_dud,conf1.mol2,../../../../ligand/cUAP_dud/1mbi.mol2
12as,cUAP_pdb_gpcr,conf1.mol2,../../../../ligand/cUAP_pdb_gpcr/1ldm.mol2
12as,cUAP_pdb_gpcr,conf1.mol2,../../../../ligand/cUAP_pdb_gpcr/1mbi.mol2
12as,cUAP_pdb_gpcr,histamine.SM2,../../../../ligand/cUAP_pdb_gpcr/h1hist.mol2
16gs,cUAP_dud,conf1.mol2,../../../../ligand/cUAP_dud/1ldm.mol2
```

図 4-2　スコア出力に失敗した化合物と蛋白質の出力例

図 4-2 は、蛋白質名、化合物グループ名、mol2 ファイルに記述された化合物名、各スコアファイルからの mol2 ファイルの相対パスを CSV 形式で出力する。

また、リストは以下②でも使用される。

② スコア出力に失敗した化合物と蛋白質の mol2 ファイルを回収する。

①で出力したリストに書かれている mol2 ファイルを ligand ディレクトリから回収する。

回収したファイルは、result ディレクトリの failed_score ディレクトリに格納される。

5　標的蛋白質選別のための MTS 法グループスクリーニング実行フェーズ

5.1　　概要

スクリーニングの実行方法は、既存の in silico スクリーニングシステムの操作説明書のグループスクリーニング実行手順と同じであるが、UAP を使用して標的蛋白質を選別する過程が新しく追加される処理である。ここでは、その方法について記述する。

5.2　　ディレクトリ構成

ここでのディレクトリ構成を、図 5-1 に示す。

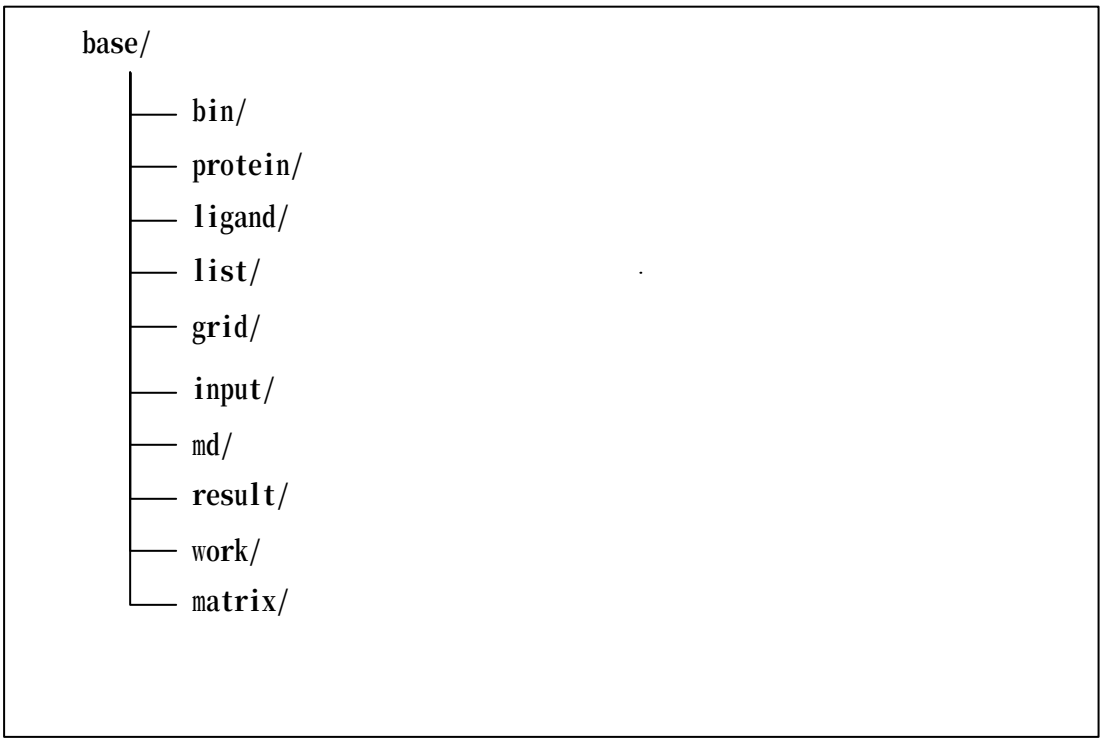

図 5-1　MTS 実行開始前のディレクトリ構成

sievgene 実行し、相互作用行列を作成した後のディレクトリ構成は、result ディレクトリ、work ディレクトリ及び matrix ディレクトリが追加された状態である。

この base ディレクトリに mts ディレクトリを作成し、mts ディレクトリで作業を行う。

mts ディレクトリに用意するファイルを表 5-1 に示す。

表 5-1　mts ディレクトリに用意するファイル

#	ファイル名	用途
1	matrix_list	相互作用行列データリストファイルの雛形
2	lig_gre.list	化合物グループリストファイル
3	pro.list	蛋白質リストファイル
4	target_pro.list	標的蛋白質リストファイル
5	uap_list	UAP リストファイル

5.3　　ファイル入出力

base/mts ディレクトリのファイル入出力を図 5-2 と図 5-3 に示す。

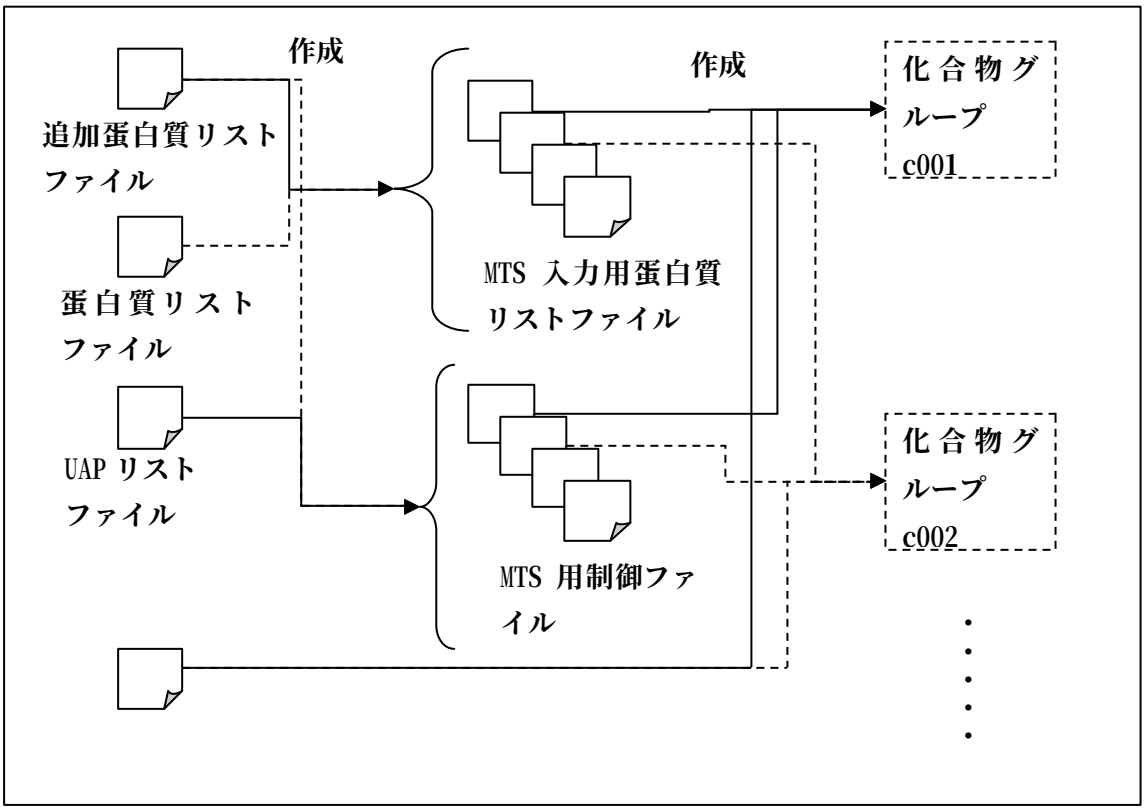

図 5-2　base/mts ディレクトリのファイル入出力

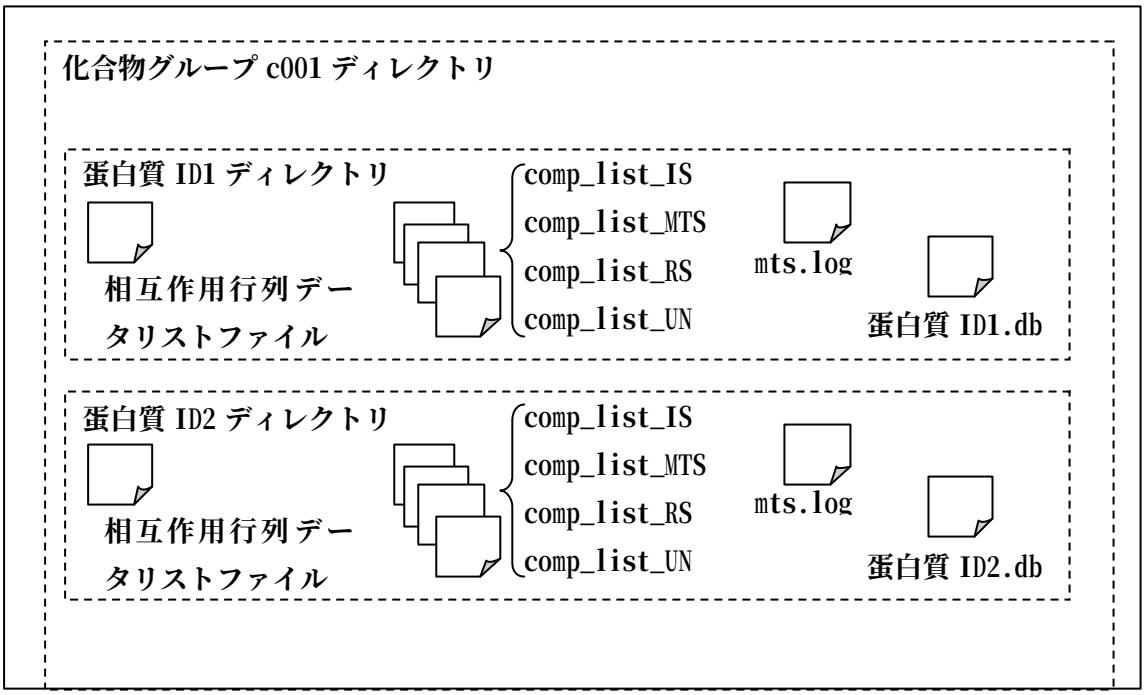

図 5-3　base/mts/化合物グループのファイル入出力

5.4　　プログラムの準備

MTS 法グループスクリーニングを行うために必要なプログラムを表 5-2 に示す。これらは mts ディレクトリに配置される。

表 5-2　mts ディレクトリに配置するプログラム

#	プログラム	用途
1	make_inp.pl	標的蛋白質ごとにMTS用制御ファイルを作成するスクリプト。
2	run_mts.sh	mts 法グループスクリーニングを実行するためのスクリプト。
3	extract_db_enrichment.pl	mts 実行後に各低分子グループの各標的蛋白質でデータベースエンリッチメントを抽出し、さらに AUC を計算する。
4	rank_target_protein.pl	標的蛋白質について、AUC による順位付けを行う。

5.5　　MTS 入力用制御ファイルの作成

UAP を用いた MTS 法グループスクリーニングでは、低分子グループごとに標的蛋白質分の MTS 法スクリーニング結果が出力される。

まず、**エラー! 参照元が見つかりません。**の pro.list、target_pro.list、UAP_list を作成する。

pro.list は、既存の相互作用行列を作成するときの蛋白質名をリストしたもので、相互作用行列作成時に使用した base/list/にある蛋白質リストを指定する。

target_pro.list は、追加した蛋白質名をリストしたもので、3.9 の追加蛋白質リストファイルを指定する。

ディレクトリで以下のスクリプトを実行する。

./make_inp.pl　*target_pro_list*　*UAP_list*　*pro.list*

上記のスクリプトを実行すると、mts ディレクトリに、pro.list に標的蛋白質を 1 つ追加した MTS 入力用蛋白質リストファイルが作成される。MTS 入力用蛋白質リストファイルのフォーマット例を図 5-4 に示す。MTS 入力用蛋白質リストファイル名は以下の規則で命名される。

MTS 入力用蛋白質リストファイル名： **pro.*蛋白質 ID*.list**

図 5-4　MTS 入力用蛋白質リストファイルのフォーマット例

さらに、lig_grp.list リストに記述された低分子化合物グループごとに target_pro.list に記述された蛋白質を標的蛋白質とする MTS 入力用制御ファイルが作成される。MTS 入力用制御ファイルのフォーマット例を図 5-5 に示す。MTS 入力用制御ファイル名は以下の規則で決定される。

MTS 入力用制御ファイル名： mts.*蛋白質 ID.UAP_list*.inp

../../蛋白質 ID.list	(MTS 入力用蛋白質リストファイルへのパス)
../../UAP_list	(UAP リストファイル)
matrix_list	(相互作用行列データリストファイル)
10000	(出力順位数)
蛋白質 ID	(標的蛋白質名)
0	(スコア補正モード選択)
0	(機械学習回数)
0	(ランダム試行回数)

図 5-5　MTS 入力用制御ファイルフォーマット例

MTS 入力用蛋白質リストファイルと UAP リストファイル、相互作用行列データリストファイルは、次項 5.8 で作成される MTS 法グループスクリーニング実行ディレクトリ mts/[*化合物グループディレクトリ*]/[*標的蛋白質ディレクトリ*]からの相対パスとなる。なお、相互作用行列データリストファイルは MTS 実行前に自動生成される。

5.6　相互作用行列データリストファイルの雛形の作成

既存の相互作用行列(**エラー！ 参照元が見つかりません。**の①)と 4 で作成した相互作用行列(**エラー！ 参照元が見つかりません。**②～④)のパスが書かれた相互作用行列データリストファイルの雛形(matrix_list)を作成する。例を図 5-6 に示す。

```
../../../matrix/pro.list_#LIGAND_GROUP#.dat
../../../matrix/target_pro.list_#LIGAND_GROUP#.dat
../../../matrix/target_pro.list_cUAP.dat
../../../matrix/pro.list_cUAP.dat
```

図 5-6　matrix_list の例

相対パスの深さは、次項で作成される MTS 法グループスクリーニング実行ディレクトリ mts/[*化合物グループディレクトリ*]/[*標的蛋白質ディレクトリ*]から base/matrix ディレクトリに対しての相対パスとする。

5.7　化合物グループリストファイルの作成

化合物グループ名をリストしたファイルを作成する。ここでは、**エラー! 参照元が見つかりません。**で準備した追加蛋白質から、AUC の高い順の上位を選別するため、全ての化合物グループで MTS を実行する必要はない。よって、2～3 の化合物グループを用いて MTS 法グループスクリーニングを実施し、化合物グループごとに AUC の傾向に差がないことを確認し、上位の標的蛋白質を選別する。

よって、ここで作成する化合物グループリストファイルは、図 5-7 のようになる。

図 5-7 では、c001 と c002 の化合物グループを使用することを意味する。

```
c001
c002
```

図 5-7　化合物グループリストファイルの例

5.8　MTS 法グループスクリーニングの実行

以下のスクリプトを実行して MTS 法グループスクリーニングを実行する。

```
./run_mts.sh
```

上記のスクリプトを実行すると、化合物グループリストファイル lig_grp.list に記述された化合物グループ名のディレクトリが作成され、さらに中に、target_pro.list に記述された標的蛋白質名のディレクトリが作成され、selectMTS の入力ファイルがコピーされる。

上記スクリプト実行後の mts ディレクトリ構成を図 5-8 に示す。ディレクトリ及びファイルは、以下の順序で作成される。

① 化合物グループリストに記述された化合物グループのディレクトリ(c001 と c002)が作成される。

② ①で作成されたディレクトリ(c001 と c002)に、target_pro.list に記述された蛋白質名のディレクトリが作成される。

③ ②で作成されたディレクトリに、matrix_list の内容が更新されたファイル(#LIGAND_GROUP#がそれぞれ c001、c002 に置換される)がコピーされる。

④ MTS 計算終了後、②で作成されたディレクトリに、comp_list_IS、comp_list_MTS、comp_list_RS、comp_list_UN、mts.log が出力される。

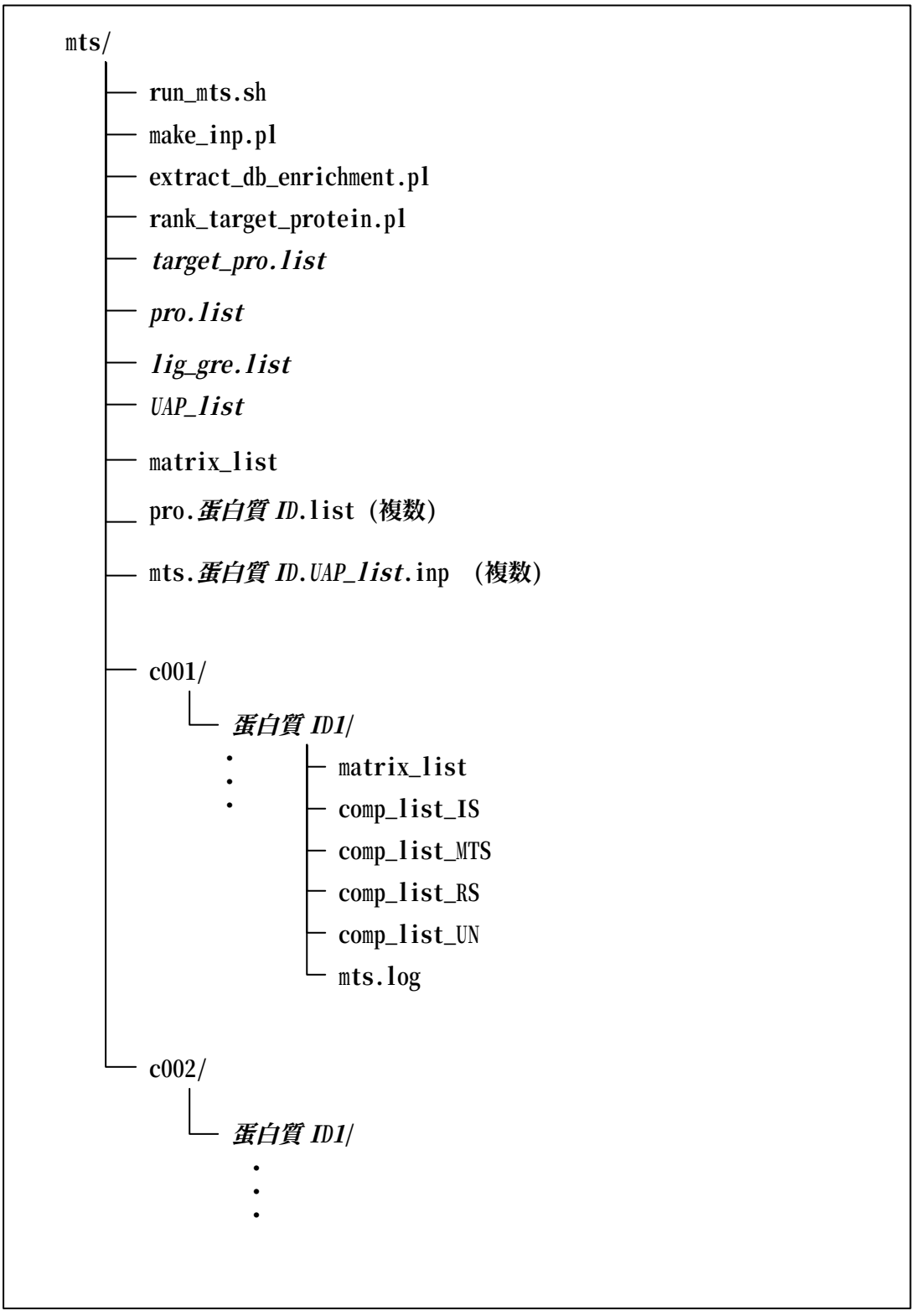

図 5-8　MTS 法グループスクリーニング実行後の mts ディレクトリの構成

5.9　MTS 法グループスクリーニング実行結果の確認

5.8 で実行した全ての MTS 法グループスクリーニングが終了したら、それぞれの MTS 計算結果から AUC を算出する。さらに、各標的蛋白質で AUC の順位付けを行い、AUC 上位の標的蛋白質を選別する。

5.9.1　AUC の計算

mts ディレクトリで、以下のスクリプトを実行する。

./extract_db_enrichment.pl [*対象ディレクトリツリーのトップディレクトリ*]

上記のスクリプトを実行すると、mts/[*化合物グループ*]/[*蛋白質 ID*]/*蛋白質 ID*.db が作成される。これは、データベースエンリッチメント及び AUC の結果が出力されている。出力例を図 5-9 に示す。

```
1, 0.0000, 0.0000
2, 0.0000, 0.0000
3, 0.0000, 0.0000
4, 0.0000, 0.0000
5, 0.0000, 0.0000
6, 0.4545, 0.4545
7, 0.0000, 0.4545
8, 0.0000, 0.4545
9, 0.4545, 0.9091
10, 0.4545, 1.3636
…{中略}…
90, 0.0000, 100.0000
91, 0.0000, 100.0000
92, 0.0000, 100.0000
93, 0.0000, 100.0000
94, 0.0000, 100.0000
95, 0.0000, 100.0000
96, 0.0000, 100.0000
97, 0.0000, 100.0000
98, 0.0000, 100.0000
99, 0.0000, 100.0000
100, 0.0000, 100.0000
AUC 57.2
```

図 5-9　*蛋白質 ID*.db の出力例

5.9.2　化合物グループ間の偏り確認と標的蛋白質のランク付けによる蛋白質の選別

5.9.1 において、各化合物グループ/各蛋白質の AUC が求まるので、それを用いて AUC で順位付けを行う。

mts/化合物グループディレクトリで以下のスクリプトを実行する。

./rank_target_protein.pl [*化合物グループディレイクトリ*]

上記のスクリプトを実行すると、全*蛋白質 ID*.db の AUC と蛋白質 ID のリストを AUC_RANK という名前のファイルに出力する。出力例を図 5-10 に示す。

```
1,蛋白質 ID32,63.44
2，蛋白質 ID13,62.786671
3，蛋白質 ID4,62.346667
4，蛋白質 ID19,60.742222
5，蛋白質 ID11,60.066667
6，蛋白質 ID8,59.773336
7，蛋白質 ID21,59.662219
8，蛋白質 ID6,59.577785
9，蛋白質 ID3,58.333336
10，蛋白質 ID10,57.648886
…{省略}…
```

図 5-10　rank_target_protein.pl 実行による標準出力例

複数の化合物グループディレクトリ(c001 と c002)に対して上記のスクリプトを実行し、AUC の順位付けのされ方をグラフにして比較することで、化合物グループ間による偏りがあるかを調べる。偏りがない場合は、化合物グループ間においてランダムであると判断し、どれか一つの化合物グループ(例：c001)の AUC の順位から上位を選別する。

6　標的蛋白質を用いた sievgene 実行フェーズ

5 で選別した標的蛋白質と 200 万低分子化合物を用いて 4 で作成した相互作用行列(**エラー！参照元が見つかりません。**②)を作成する。作成方法は、4 と同様である。

7　標的蛋白質を用いた MTS 法グループスクリーニング実行フェーズ

5 で選別した標的蛋白質と 200 個の化合物グループを用いて MTS 法グループスクリーニングを実行する。実行方法は 5 と同様である。

8　MTS 法総合スクリーニング

8.1　　概要

7 で行った MTS 法グループスクリーニングの結果を用いて、MTS 法総合スクリーニングを実行する。

① MTS 法総合スクリーニングの実行環境を構築する。
② 相互作用行列の再構成を行う。
③ MTS 法総合スクリーニングを実行する。

8.2　　実行環境の構築

mts ディレクトリで以下のスクリプトを実行する。

./prepare_total_mts.pl ***制御ファイル***

制御ファイルのフォーマットを図 8-1 に示す。

```
標的蛋白質リストファイル
ヒットリストファイル
相互作用行列(エラー！参照元が見つかりません。①)ファイル
相互作用行列(エラー！参照元が見つかりません。②)ファイル
相互作用行列(エラー！参照元が見つかりません。③)ファイル
相互作用行列(エラー！参照元が見つかりません。④)ファイル
```

図 8-1　prepare_total_mts.pl 用制御ファイルフォーマット

標的蛋白質、ヒットリストファイルは、MTS 法グループスクリーニングで用いたファイルを指定する。また、相互作用行列(**エラー！参照元が見つかりません。**①～④)ファイルは次の例に従って相対パスを指定する。

制御ファイルの記述例を図 8-2 に示す。

```
pro.list
hit.list
../../matrix/pro.list_#LIGAND_GROUP#.dat
../../../matrix/pro.list_lig01.dat
../../matrix/pro01.list_#LIGAND_GROUP#.dat
../../../matrix/pro01.list_lig01.dat
```

図 8-2　prepare_total_mts.pl 用制御ファイル例

スクリプト実行後のディレクトリ構成を図 8-3 に示す。

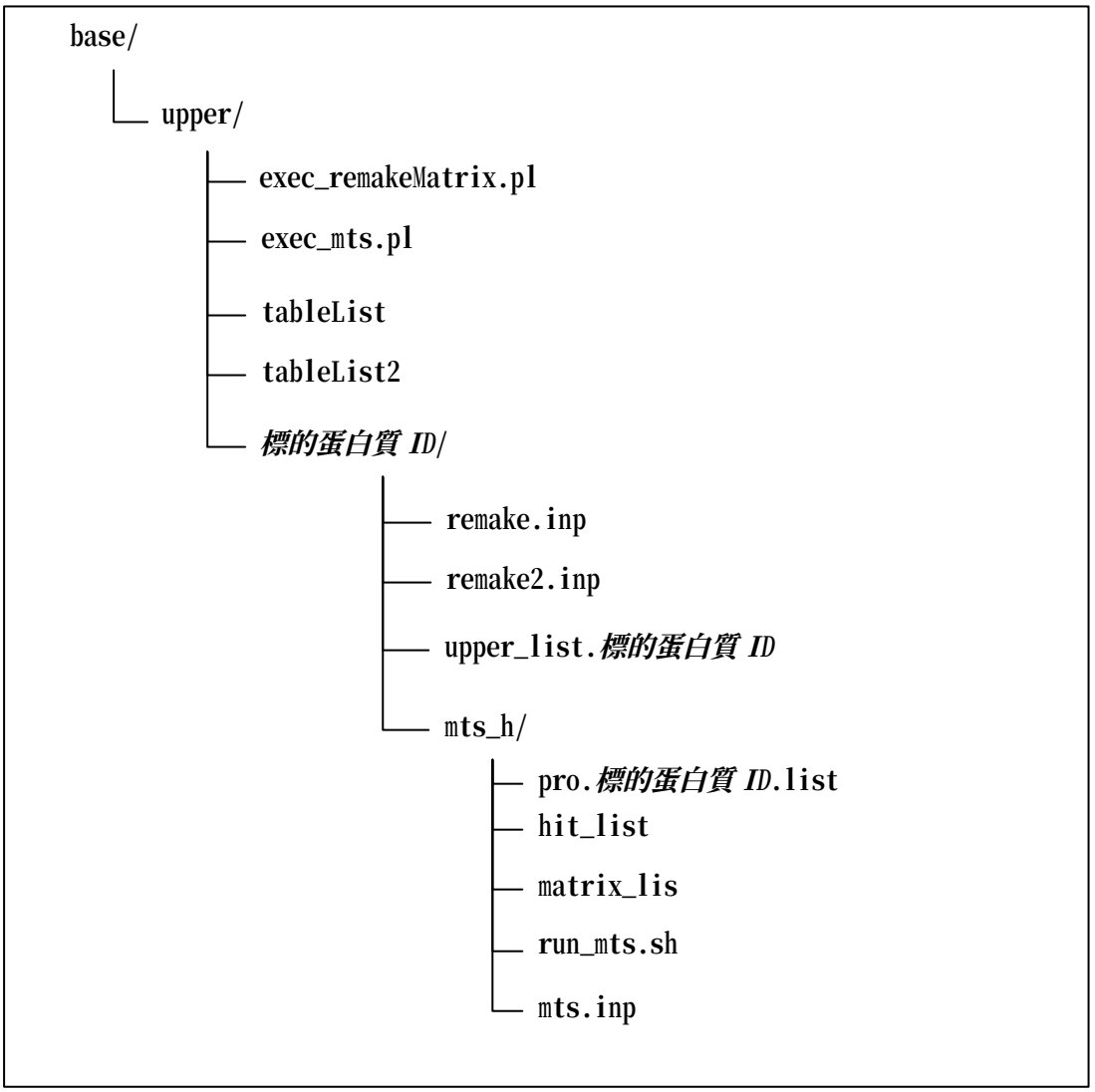

図 8-3　prepare_total_mts.pl 実行後のディレクトリ構成

スクリプトを実行すると、base/upper ディレクトリが作成され、さらに base/upper ディレクトリに各標的蛋白質名のディレクトリが作成される。
各標的蛋白質ディレクトリは、MTS 法総合スクリーニング環境となるディレクトリである。base/upper/*標的蛋白質 ID*/mts_h は MTS 法総合スクリーニングの実行および結果を格納するディレクトリとなる。

各ファイルについて、表 8-1 に示す。

表 8-1　upper ディレクトリ以下のファイル一覧

#	ファイル名	意味
1	exec_remakeMatrix.pl	各標的蛋白質の相互作用行列データ抽出設定ファイルの作成を行うスクリプト。remakeMatrix のラッパースクリプト。本スクリプトは自動で作成される。
2	exec_mts.pl	各蛋白質の MTS 法総合スクリーニングの実行を行う。本スクリプトは自動で作成される。
3	tableList	相互作用行列データリストファイル(**エラー! 参照元が見つかりません。**①用) in silico スクリーニングマニュアル参照
4	tableList2	相互作用行列データリストファイル(**エラー! 参照元が見つかりません。**③用) in silico スクリーニングマニュアル参照
5	remake.inp	総合スクリーニング用相互作用行列作成設定ファイル(**エラー! 参照元が見つかりません。**②用)。 in silico スクリーニングマニュアル参照
6	remake2.inp	総合スクリーニング用相互作用行列作成設定ファイル(**エラー! 参照元が見つかりません。**④用)。 in silico スクリーニングマニュアル参照
7	upper_list.*標的蛋白質 ID*	総合スクリーニング用低分子リストファイル。
8	pro.*標的蛋白質 ID*.list	蛋白質リストファイル。
9	hit_list	ヒットリストファイル in silico スクリーニングマニュアル参照
10	matrix_list	相互作用行列データリストファイル in silico スクリーニングマニュアル参照
11	run_mts.sh	MTS 法総合スクリーニング実行スクリプト。 in silico スクリーニングマニュアル参照
12	mts.inp	MTS 法総合スクリーニング制御ファイル。 in silico スクリーニングマニュアル参照

8.3　相互作用行列の再構成

base/upper ディレクトリで、以下のスクリプトを実行する。

./ exec_remakeMatrix.pl

上記のスクリプトを実行すると、各標的蛋白質のディレクトリで総合スクリーニング用相互作用行列を作成するためのジョブが投入される。

ジョブ終了後、各標的蛋白質ディレクトリに matrix.dat、matrix2.dat が生成される。

8.4 MTS 法総合スクリーニング実行

base/upper ディレクトリで、以下のスクリプトを実行する。

```
./exec_mts.pl
```

上記のスクリプトを実行すると、base/upper/*標的蛋白質 ID*/mts_h ディレクトリにおいて MTS 法総合スクリーニングのジョブが投入される。

ジョブ終了後、base/upper/*標的蛋白質 ID*/mts_h ディレクトリに comp_list_IS、comp_list_UN、comp_list_MTS、comp_list_RS、mts.log が生成される。