

MolSite

Ligand-binding pocket prediction

USER MANUAL

Version 1.0

Copyright (C) 2006-2010 National Institute of Advanced Industrial Science and Technology (AIST)

Copyright (C) 2006-2010 Japan Biological Informatics Consortium (JBIC)

目次

1	事前準備	1
1.1	作業用ディレクトリの準備	1
1.2	実行プログラムの配置	2
1.3	制御ファイルの配置	2
1.4	蛋白質データの配置	3
1.5	化合物データの配置	3
1.6	化合物リストの作成	3
1.7	ディレクトリ構成まとめ	4
2	ポイントファイルの作成	7
2.1	make_pre_point.pl 用制御ファイルの作成	7
2.2	make_pre_point.pl の実行	7
2.3	ディレクトリ構成	7
3	グリッドファイルの作成	8
4	ドッキングジョブの実行	9
4.1	通常の mol2 ファイルを使用した場合の実行方法	9
4.2	マルチ mol2 ファイルを使用した場合の実行方法	9
4.3	ジョブ実行	9
4.4	ディレクトリ構成	9
5	相互作用行列ファイルの作成	11
5.1	通常の mol2 ファイルを使用した場合の実行方法	11
5.2	マルチ mol2 ファイルを使用した場合の実行方法	11
5.3	相互作用行列作成確認	11
5.4	ディレクトリ構成	11
6	ドッキングスコアの集計とポイントデータの選択	11
6.1	作業用ディレクトリの作成	11
6.2	make_scoring_grid.pl の実行	12
6.3	スコアリンググリッドの選択	12
7	ドッキングポケットの中心座標の計算	13
8	ドッキングポケットの検証	13

1 事前準備

本システムは、ProbeSite 法を用いて蛋白質のドッキングポケット座標を定量的に求めるためのシステムである。はじめに、ドッキングポケット座標を求めたい蛋白質の主鎖から選択した原子の配置座標をドッキングポケット座標候補とし、ポイントファイルを複数作成する。次に、作成されたポイントファイル、蛋白質 PDB ファイル及びトポロジファイルを用いて 1 つの蛋白質データとしてドッキングジョブを実行し、各ポイント位置でのドッキングスコアを算出する。また、そのときのドッキング化合物のドッキング座標を用いて統計的にドッキングポイント座標を計算する。

本マニュアルでは、1a28 蛋白質のドッキングポケット座標を求めるものとする。

1.1 作業用ディレクトリの準備

事前準備として、作業用ディレクトリを作成する。ここでは任意の場所に work ディレクトリを作成し、ここを作業のトップディレクトリとする。

```
$ mkdir work      ($はプロンプトを表す。以下同様)
```

作業用ディレクトリについて表 1.1-1 に示す。

表 1.1-1 work ディレクトリに作成するディレクトリ一覧

#	ディレクトリ名	概要	名称
1	base	各種コマンドを実行するためのディレクトリ	任意
2	protein	蛋白質データを格納するディレクトリ	固定
3	ligand	化合物データを格納するディレクトリ	固定
4	base/input	sievgene プログラムの制御ファイルを格納するディレクトリ	固定
5	base/list	蛋白質リスト、化合物リストを格納するディレクトリ	固定

ここでは、base ディレクトリを base とする。

```
$ cd work
$ mkdir base protein ligand
$ cd base
$ mkdir input list
$ cd ..
```

1.2 実行プログラムの配置

各種実行用スクリプト、プログラムを base/bin に配置する。使用するプログラムは全て base/bin に格納する。スクリプト、プログラムの一覧を表 1.2-1 に示す。

表 1.2-1 work/bin に配置するスクリプト/プログラム一覧

#	プログラム名	用途
1	RUN_docking_ps.pl	ドッキングジョブ投入用スクリプト
2	calc_average_coord	ドッキング化合物の平均座標を求める
3	calc_center_coord	全ドッキング化合物の平均座標を求める
4	calc_sigma	ドッキングスコアの平均値、最大値、標準偏差を求める
5	calc_verification_data	求めたドッキングポケットを検証する
6	create_pre_point	蛋白質 PDB ファイルからポイントデータを作成する
7	make_docking_pocket_coordinate.pl	calc_center_coord を呼び出す
8	make_docking_score_multi.pl	マルチ mol2 ファイルを使用した場合のドッキングジョブ投入スクリプト
9	make_docking_score_ps.csh	ドッキングジョブ投入スクリプト
10	make_grid.csh	グリッドファイルを作成する
11	make_multi.pl	マルチ mol2 ファイルを作成する
12	make_pre_point.pl	create_pre_point を呼び出す
13	make_score_data_multi_ps.pl	マルチ mol2 ファイルを使用した場合の相互作用行列作成スクリプト
14	make_score_data_ps.csh	相互作用行列作成スクリプト
15	make_scoring_grid.pl	相互作用行列から各蛋白質に calc_sigma を呼び出す。
16	sievgene	ドッキングジョブ実行プログラム
17	verify_point.pl	各蛋白質に calc_verification_data を呼び出す

1.3 制御ファイルの配置

base/input ディレクトリに sievgene 実行用の制御ファイルを配置する。配置するファイルをに示す。

表 1.3-1 work/input ディレクトリに配置する制御ファイル一覧

#	ファイル名	概要
1	s0.inp	sievgenen によるドッキングジョブ実行用制御ファイル
2	s0grid.inp	make_grid.csh によるグリッドファイル作成用制御ファイル

s0.inp は、本システム用のに記述する箇所がある。表 1.3-2 の項目を設定する。

表 1.3-2 s0.inp の設定項目

#	設定項目	値	概要
1	COORDInate	PDB	ドッキングポーズ用ファイル形式
2	NAMECOordinate	pose.pdb	ドッキングポーズファイル名
3	CANDIDatenumber	1	1 化合物当たりの出力するドッキングポーズ数
4	SCORENumber	1	1 化合物当たりの出力するスコアの数

1.4 蛋白質データの配置

protein ディレクトリに、ドッキングポケット座標を求めたい蛋白質データを配置する。ここでは、1a28 蛋白質のデータを配置するため、protein/1a28 ディレクトリを作成し、protein/1a28 ディレクトリに Pro_md.pdb、Pro.tpl ファイルを格納する。

```
$ cd protein
$ mkdir 1a28
$ cp Pro_md.pdb のオリジナルファイルパス .
$ cp Pro.tpl のオリジナルファイルパス .
```

1.5 化合物データの配置

ligand ディレクトリに、化合物グループディレクトリを作成し、化合物グループディレクトリに化合物データ mol2 ファイルを格納する。

化合物グループデータ名及び格納される mol2 ファイルの数は任意とする。

ここでは、化合物グループ名を c001 とし、格納する mol2 ファイルを 100 個とする。

相互作業行列作成で使用した ligand ディレクトリでも良い。

1.6 化合物リストの作成

base/list ディレクトリに、化合物リストを作成する。化合物リストは、ligand ディレ

クトリに配置した化合物グループディレクトリ名と同じ名前である。ここでは、base/list ディレクトリに c001 ファイルを作成し、ligand/c001/に含まれる mol2 ファイル名を 1 行に 1 化合物記述する。

```
$ cd base/list  
$ ls ../../ligand/c001 > c001
```

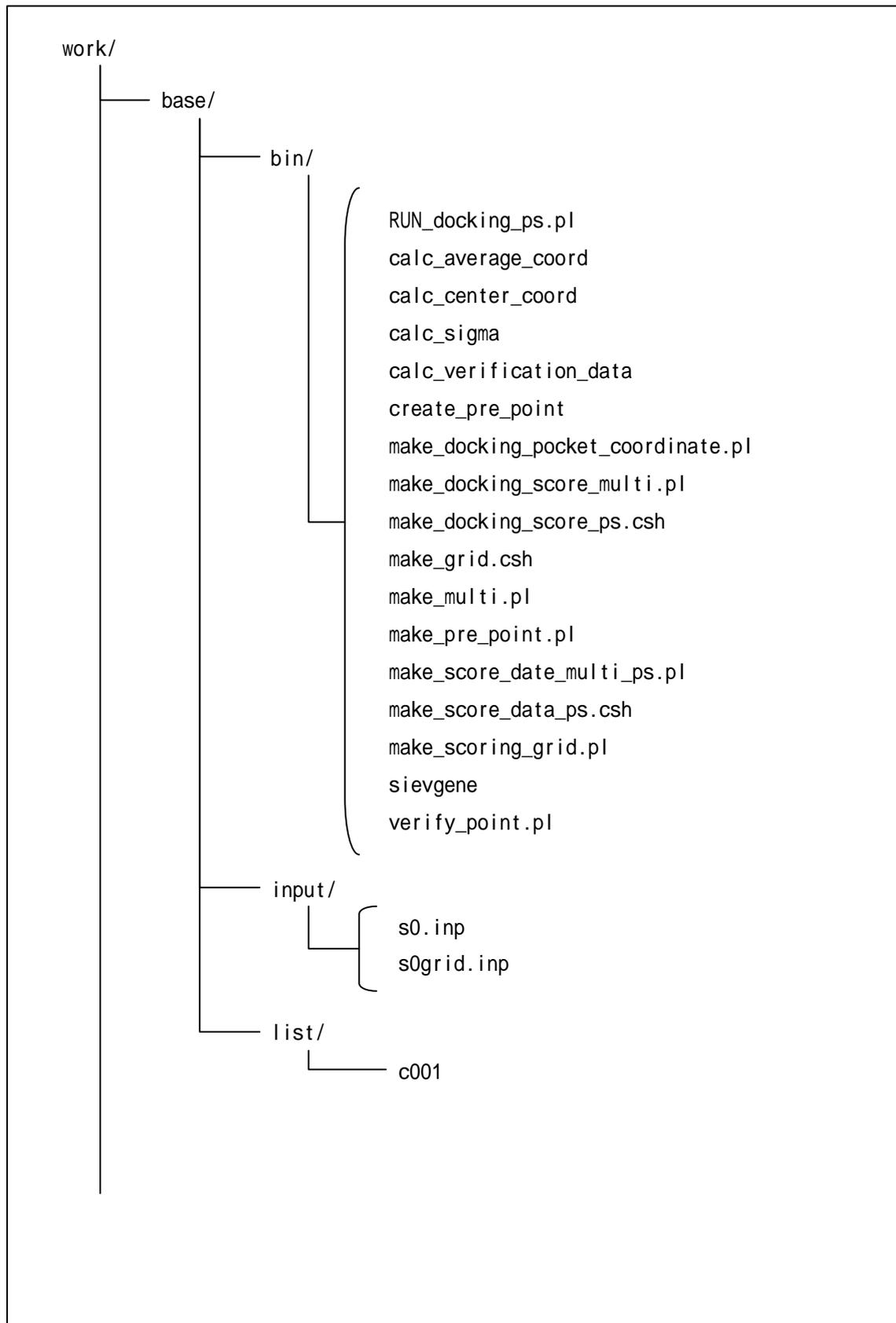
マルチ mol2 ファイルを使用する場合は、以下のコマンドでマルチ mol2 ファイルを作成する。

```
$ ./bin/make_multi.pl c001 100
```

make_multi.csh を実行すると、work/ligand_multi ディレクトリが作成される。ここは、相互作業行列作成作業一般と同様の処理となる。

1.7 ディレクトリ構成まとめ

ここまでで作成したディレクトリ及びファイルの構成について、図 1.7-1 に示す。



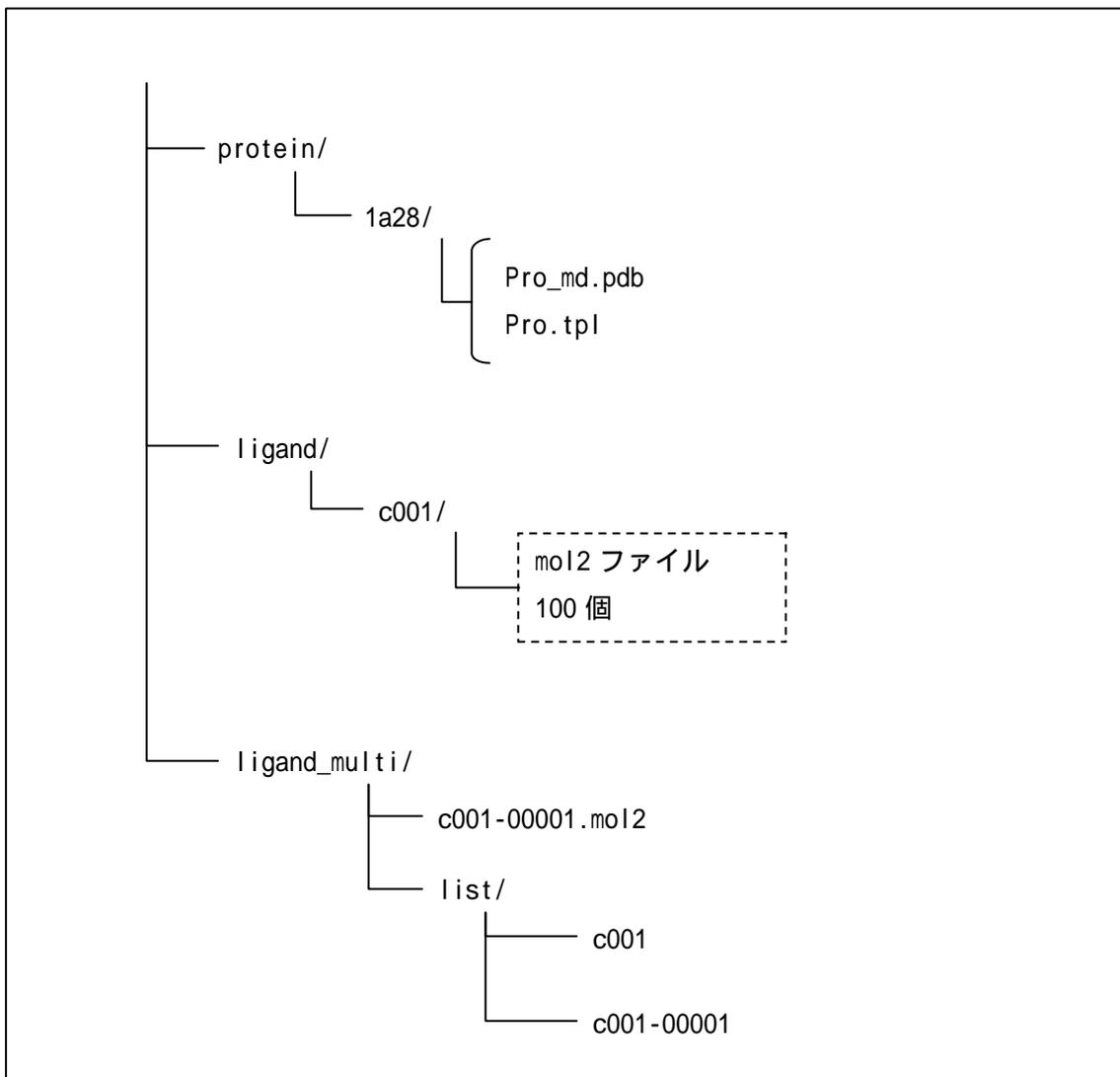


図 1.7-1 事前準備が完了したディレクトリ構成

2 ポイントファイルの作成

2.1 make_pre_point.pl 用制御ファイルの作成

ポイントファイルを作成するための制御ファイルを作成する。ここでは 1a28 蛋白質の主鎖の C 原子をポイント候補の中心座標とする。制御ファイルを init_protein とし、以下を記述する。

```
1a28 CA
```

CA は、C 原子を意味する。窒素原子を指定する場合は N、酸素原子を指定する場合は O を指定する。

2.2 make_pre_point.pl の実行

base ディレクトリで以下のコマンドを実行する。

```
$ ./bin/make_pre_point.pl list/init_protein
```

protein ディレクトリに 1a28_CA+1 ~ 32 の 32 個のディレクトリが作成され、それぞれのディレクトリに point.pdb ファイルが作成されている。また、Pro_md.pdb と Pro.tpl のシンボリックリンクが作成される。

ここで、32 個のディレクトリが作成されたが、別の蛋白質を使用した場合は作成されるディレクトリの数は異なる。

また、base/list ディレクトリに protein.list ファイルが作成される。

2.3 ディレクトリ構成

make_pre_point.pl を実行後に新規に追加されたディレクトリ、ファイル構成を図 2.3-1 に示す。

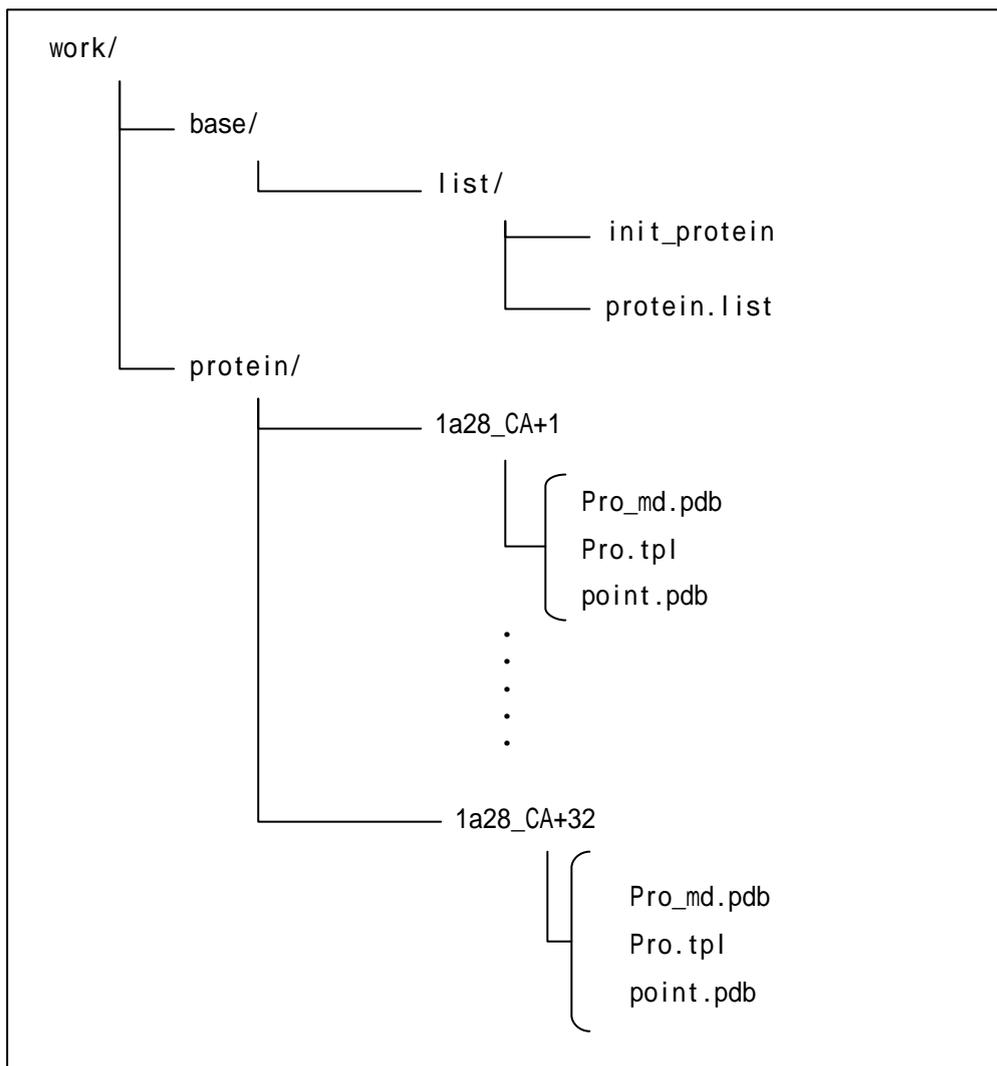


図 2.3-1 make_pre_point.pl 実行により追加されたディレクトリ、ファイル

3 グリッドファイルの作成

base ディレクトリで以下のコマンドを実行する。

```
$ ./bin/make_grid.csh
```

make_grid.csh を実行すると、base/grid ディレクトリが作成され、さらに base/grid/1a28_CA+1 ~ 1a28_CA+32 ディレクトリが作成される。グリッドファイル作成の計算が終了すると、base/grid/1a28_CA+1 ~ 1a28_CA+32 ディレクトリに grid.file が作成される。この作業は、相互作用行列作成作業一般と同様である。

4 ドッキングジョブの実行

ドッキングジョブを投入する。

4.1 通常の mol2 ファイルを使用した場合の実行方法

以下のコマンドを実行する。

```
$ ./bin/make_docking_score_ps.csh protein_list c001
```

4.2 マルチ mol2 ファイルを使用した場合の実行方法

以下のコマンドを実行する。

```
$ ./bin/make_docking_score_multi.pl protein_list c001
```

4.3 ジョブ実行

make_docking_score_ps.csh もしくは make_docking_score_multi.pl を実行すると、sievgene のドッキングジョブが投入、実行される。ここでは、1a28_CA+1 ~ 1a28_CA+32 = 32 個の蛋白質 × 100 個の化合物についてのドッキングジョブが実行される。

4.4 ディレクトリ構成

make_docking_score_ps.csh もしくは make_docking_score_multi.pl を実行後、base ディレクトリに result 及び work ディレクトリが作成される。それぞれ、sievgene の出力が格納される。ディレクトリ構成は相互作用行列作成作業一般と同様である。ここでは、マルチ mol2 ファイルを用いた場合においての、result ディレクトリ及び work ディレクトリについて、図 4.4-1 に示す。通常の mol2 ファイルを用いた場合でも、概ね同様の出力となるが、この後の作業について、この違いについて利用者が意識することはない。

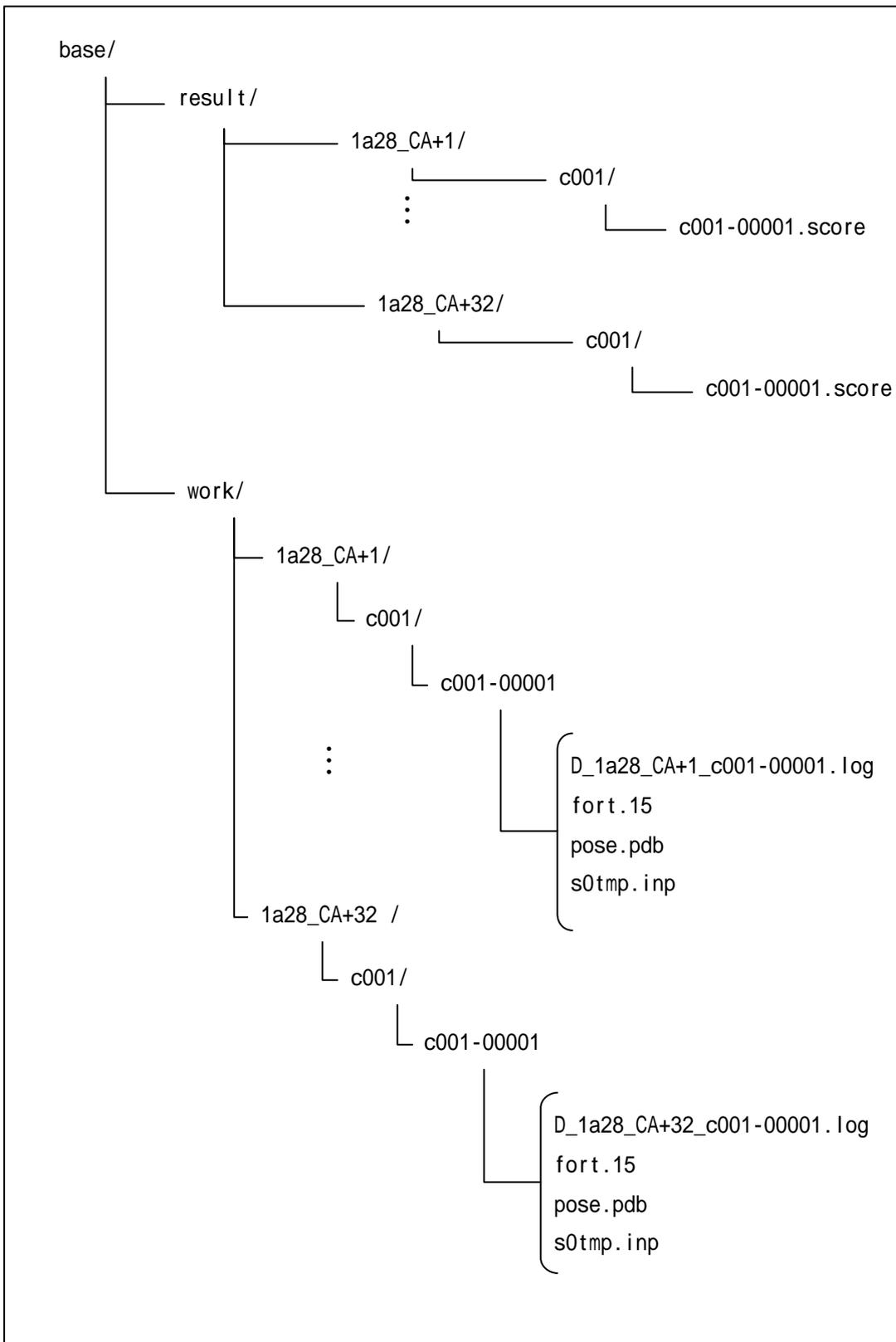


図 4.4-1 ドッキングジョブ実行後に追加されたのディレクトリ/ファイル構成

5 相互作用行列ファイルの作成

ドッキングジョブ実行が全て終了したら、相互作用行列を作成する。

5.1 通常の mol2 ファイルを使用した場合の実行方法

以下のコマンドを実行する。

```
$ ./bin/make_score_data_ps.csh protein_list c001
```

5.2 マルチ mol2 ファイルを使用した場合の実行方法

```
$ ./bin/make_score_data_multi_ps.pl protein_list c001
```

5.3 相互作用行列作成確認

make_score_data_ps.csh もしくは make_score_data_multi_ps.pl を実行すると、base/matrix ディレクトリが作成される。さらに base/matrix ディレクトリに、protein.list_#c001#.dat ファイルが作成される。

protein.list_#c001#.dat ファイルは、相互作用行列作成作業一般と同様のファイルであるが、ファイル名が異なる。化合物グループ名が#で囲まれていることを確認する。

5.4 ディレクトリ構成

相互作用行列作成後に追加されたディレクトリ構成を、図 5.4-1 に示す。

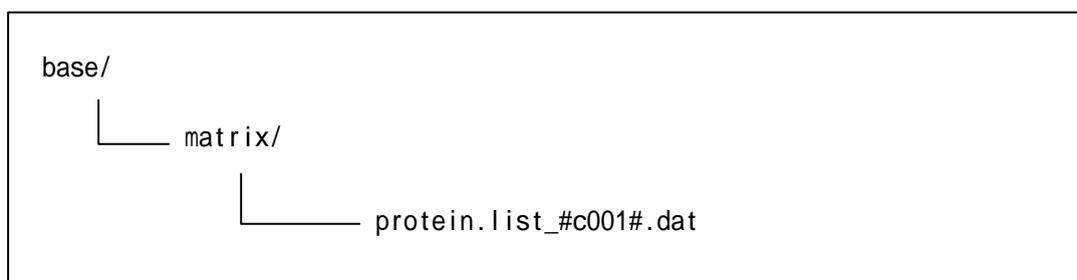


図 5.4-1 相互作用行列作成後に追加されるディレクトリ/ファイル構成

6 ドッキングスコアの集計とポイントデータの選択

6.1 作業用ディレクトリの作成

相互作用行列に出力されているスコア及び順番の情報を用いて、ドッキングポケットの統計データを作成する。

ここでは、作業用ディレクトリとして、base/prediction ディレクトリを作成する。

```
$ cd base
$ mkdir prediction
```

必ず、base/ディレクトリの下に作業用ディレクトリを作成すること。

次に、prediction/に、matrix_list を作成する。prediction/から base/matrix/*.dat の相対パスを記入する。例を示す。

matrix_list

```
../matrix/ protein.list_#c001#.dat
```

次に、prediction/に、lig_grp を作成する。ドッキングジョブを実行したときの化合物グループ名を記入する。例を示す。

lig_grp

```
c001
```

6.2 make_scoring_grid.pl の実行

prediction ディレクトリで、以下のコマンドを実行する。

```
$ ../bin/make_scoring_grid.pl 1a28
```

make_scoring_grid.pl コマンドを実行すると、prediction ディレクトリに 1a28_CA+1_ave.coord ~ 1a28_CA+32_ave.coord 、 1a28_CA+1_sigma.score ~ 1a28_CA+32_sigma.score、1a28_selected_protein_list、1a28_sigma.score が作成される。

6.3 スコアリンググリッドの選択

1a28_sigma.score は、1a28 蛋白質の全ポイントで出力されたスコアの平均値、最大値の順位が出力されている。このファイルから最適と思われる蛋白質_ポイントの組み合わせを選択し、リストに記述する。ここでは、1a28_protein.list というファイルを作成し、1a28_CA+19、1a28_CA+24 の2つを記述する。

```
1a28_CA+19
1a28_CA+24
```

仮に、2.1 で CA 以外に N を指定した蛋白質も書かれていた場合、1a28_sigma.score には CA と N が含まれた結果となる。

7 ドッキングポケットの中心座標の計算

ドッキングポケットの中心座標を計算する。

prediction ディレクトリで以下のコマンドを実行する。

```
$ ../bin/make_docking_pocket_coordinate.pl 1a28_protein.list
```

make_docking_pocket_coordinate.pl を実行すると、1a28_CA+19_ave.coord、1a28_CA+24_ave.coord、1a28_CA_center.coord が作成される。

ここで着目するのは、1a28_CA_center.coord である。1a28_CA_center.coord には、選択した CA 原子をポケットとした場合においての、ポケットの中心座標が計算される。出力例を以下に示す。

```
P 1a28_CA+19
@ave      43.67729    39.75393    32.266408
@sigma    1.001717     0.5455856   0.1208649
P 1a28_CA+24
@ave      61.63492    37.72311    32.292023
@sigma    1.6766767   0.198657    0.8784733
```

8 ドッキングポケットの検証

既に 1a28 蛋白質のドッキングポケットが既知である場合、既知のドッキングポケットの中心座標との距離、及びドッキングポケットの最短距離を計算し、どの程度、求めたポケット位置が正しかを検証することができる。

prediction ディレクトリで以下のコマンドを実行する。

```
$ ../bin/verify_point.pl point.pdb 1a28_CA_center.coord
```

ここでは、CA 原子で選択したポケットで得られたポケット中心座標結果である 1a28_CA_center.coord の検証を行う。point.pdb は、既知のドッキングポケットの PDB ファイルであり、事前に準備する。

実行後、1a28_CA.verify ファイルが出力される。

@center-to-center distance は、既知ドッキングポケットの中心座標と、求めたポケットの座標との距離を表す。

@minimum distance は、既知ドッキングポケット原子と求めたポケットの座標との最短距離を表す。

P 1a28_CA+19

@center-to-center distance: 31.22828

@minimum distance : 20.62262

P 1a28_CA+24

@center-to-center distance: 11.98828

@minimum distance : 6.12223

検証結果では、1a28_CA+19 よりも 1a28_CA+24 のポケットのほうが、実際の値に近いことが分かる。