pdbcheck_v2
詳細設計書

2018 年 2 月 5 日

目次

１．目的

pdbcheck はタンパク質の PDB データを入力し、以下の情報を出力する。

・解析したタンパク質の情報
・加工した PDB ファイル

本開発では pdbcheck に表の#9～#13 の残基の加工機能を追加する。

項番	機能概要	オプション名	加工内容
1	alternate location indicator の検出と加工	-alt	先頭の原子の情報を有効とする
2	主鎖原子の欠損の検出と加工	-bb	主鎖原子が欠損している残基を削除する
3	末端残基の検出と加工	-cap	N 末は NTER, C 末は CTER 構造に変換する
4	SS-BOND 、 CYS-(CYM) の検出と加工	-ss	硫黄間の距離が 2.0Å以内の場合に SSBOND 行を出力する。CYS-が存在する場合残基名を CYM とする。
5	近接原子の検出と加工	-hyd	水素と重原子の bond 距離を 0.5 倍した位置に水素を移動する
6	直行する二面角の検出と加工	-dih	4 つの原子が直行する場合、乱数で座標をずらす
7	HETATM の無効化	-disableHet	入力 PDB データの HETATM 行を読込まない。また出力 PDB にも出力しない。
8	入力 PDB の TER の保持	-keepTer	入力 PDB の TER を保持する。（入力 PDB の TER/END 行以外でのチェイン分割をしない。）
9	REMARK の出力	-remark	入力 PDB データの REMAKR 行(REM 行)を出力 PDB に出力する。
10	欠損原子の付加と余剰原子の削除	-mod	残基のモデルに応じて原子の追加・削除を行う
11	残基種別の変更	-mut	水素結合が増加するように残基を置換する
12	末端の回転	-rot	水素結合が増加する向きに側鎖を回転させる
13	金属配位の変換	-cmp	金属配位が増えるように残基種を変更する

２．ユーザインタフェース

２．１．コマンド

PDB チェックツールは”pdbcheck”コマンドで実行し、標準入力から下記情報を入力する。

1 行目：入力 PDB ファイル名
2 行目：出力 PDB ファイル名
3 行目以降：オプション値［ 先頭チェイン、最終チェイン、先頭残基、最終残基 ］
＃通常は、ファイルからのリダイレクションで入力する。

下記条件の場合、エラーを標準出力に出力し、プログラムを中断する。

(1) 1 行目に指定されたファイルが存在しない、あるいは読み込めない場合
→”ERROR: can not read file ファイル名”

(2) 2 行目に指定されたファイルを作成できない場合
→”ERROR: can not write file ファイル名”

(3) 3 行目以降の文字列が、“-alt”, “-cap”, “-hyd”, “-ss”, “-bb”, “-dih”,”-disableHet” ,”-keepTer” ,”-remark” のいずれでもない場合
→”ERROR: invalid option: 文字列”

オプション値の後に先頭チェイン、最終チェイン、先頭残基、最終残基を記載した場合、加工オプションの範囲が限定される。
複数記載した場合は、最後の値が設定される。

２．２．入力ファイル

入力ファイルは、コマンドで指定された PDB ファイルを読み込む。
なお、PDB ファイル読込みの詳細については、「３．１．入力ファイル読込み」を参照のこと。

２．３．出力ファイル

出力ファイルは、コマンドで指定された名前の PDB ファイルとして作成する。
出力ファイルは全ての加工が修正した時点のデータとして作成する。
なお、PDB ファイル出力の詳細については、「３．２．PDB 出力」を参照のこと。

２．４．ログ

ログは標準出力とし、入力情報、各種診断情報、出力情報の３つを表示する。

(1) 入力情報

```
PDB CHECK TOOL v1.1  2012. Sep. 28

INFORMATION> INPUT
  1) INPUT FILE
     入力 PDB ファイル名

  2) OUTPUT FILE
     出力 PDB ファイル名

  3) SPECIFIED OPTION
     指定したオプション値
```

(2) 各種診断情報

各機能の診断情報は「３．処理概要」に記述する。

(3) 出力情報

```
INFORMATION>  OUTPUT
  OUTPUT FILE
     出力 PDB ファイル名
```

３．処理方式

以下、各種機能の処理概要を示す。

３．１．入力ファイル読込み

以下に、入力 PDB ファイルの読込み詳細について説明する。

(1) RECORD TYPE

PDB チェックツールが読込む PDB データの RECORD TYPE は以下のとおりである。

項番	RECORD TYPE	概要
1	ATOM	原子情報を読込む。
2	HETATM	ヘッダが”HETA”で始まる行を、HETATM 行と認識し読込む。ただし、-disableHet オプション指定時は読込まない。
3	TER	チェインの区切りとして認識する。
4	END	TER と同意とみなし､チェインの区切りとして認識する。
5	REMARK	ヘッダが”REM”で始まる行を、REMARK 行と認識し読込む。ただし、-remark オプションが指定されていない場合は読込まない。

(2) ATOM / HETATM の残基分類

読込んだ ATOM / HETATM は、下記の表のとおりに残基で分類分けする。PDB チェックツールは、この残基分類により、各種チェックの実施有無、出力 PDB での出力の有無を決める。なお、出力の詳細については、「３．２．PDB 出力」を参照のこと。

※ただし、-disableHet オプション指定時は、残基の分類はせず ATOM 行で記述されている残基は全て、アミノ酸残基とみなし、各チェック・加工が行われることに注意すること。

項番	残基分類	概要
1	アミノ酸	残基名が、以下のものをアミノ酸とみなす。 ALA / ARG / ASN / ASP / CYS / GLN / GLU / GLY / HIS / ILE / LEU / LYS / MET / PHE / PRO / SER / THR / TRP / TYR / VAL また、以下のものは条件付きでアミノ酸とみなす。条件に一致しないときは、リガンドとみなす。

		・ACE 条件：同一チェインにおいて、ACE 残基の直後(PDB 出現順)の残基がアミノ酸である ・NME / NHE / NH2 条件：同一チェインにおいて、NME / NHE / NH2 残基の直前(PDB 出現順)の残基がアミノ酸である ・ABA / NLE / SEP / TYP / THP / LYN / CYM / MML / DML / TML / MMA / ADA / SDA 条件：同一チェインにおいて、上記残基の直前、または直後(いずれも、PDB 出現順)の残基がアミノ酸である
2	核酸	残基名が、以下のものを核酸とみなす。 _DA / _DG / _DT / _DC / _DU / __A / __G / __T / __C / __U / ADE / GUA / CYT / URA / DAD / DGU / DCY / DTH / DMC
3	水グループ	残基名が、以下のものを水グループとみなす。 TIP / HOH / WAT / H2O / SPC
4	Na イオン グループ	残基名が、以下のものを Na イオングループとみなす。 CIP / _NA / NA_ / NA2 / NA5 / NA6 / NAO / NAW
5	Cl イオン グループ	残基名が、以下のものを Cl イオングループとみなす。 CIM / _CL
6	リガンド	上記以外の残基はリガンドとみなす。

※_は半角空白を意味する。__は半角空白 2 つを意味する。

次に、残基分類について、各検出・加工対象の有無を以下に示す。

検出・加工内容	残基分類					
	アミノ酸	核酸	水グループ	Na イオングループ	Cl イオングループ	リガンド
alternate location indicator	対象	対象	対象	対象	対象	対象
主鎖欠損原子	対象	対象外	対象外	対象外	対象外	対象外
末端残基	対象(※1)	対象外	対象外	対象外	対象外	対象
SSBOND	対象	対象外	対象外	対象外	対象外	対象外
近接原子	対象	対象	対象外	対象外	対象外	対象
直行する二面角	対象	対象外	対象外	対象外	対象外	対象外

※1…通常、「-cap」オプション指定時に、チェインの N 末端に ACE 残基、C 末端

に、NME 残基を追加する。ただし、チェインの N 末端および C 末端が非アミノ酸残基の場合は ACE 残基および、NME 残基の追加は行わない。
末端残基の検出・加工の詳細は、「３．２．主鎖原子欠損の検出と補正」を参照のこと。

(3) チェインの判定

PDB チェックツールは入力 PDB に対し、以下の順序で、チェインを認識する。ただし、オプション "-keepTer" が指定されている場合は、処理順序 1 (TER/END 行による認識)のみでチェインの判定を行い、処理順序 2 以降の判定は行なわない。

処理順序	対象とする残基分類	概要
1	全て	TER または END が現れた場合、チェインを区切る。 なお、初めの ATOM / HETATM 行のチェイン ID が空白だった場合、TER / END にあわせて、内部でチェイン ID を割り当てる。※1
2	全て	同一チェインにおいて、N 残基と N+1 残基でチェイン ID が違う場合、チェインを区切る。
3	核酸	同一チェインにおいて、核酸残基 N と非核酸残基 N+1 が存在する場合、N と N+1 の間でチェインを区切る。 本処理終了後、核酸(の塊)は独立したチェインとなる
4	水グループ	同一チェインにおいて、水グループ残基 N と非水グループ残基 N+1 が存在する場合、N と N+1 の間でチェインを区切る。 本処理終了後、水グループ(の塊)は独立したチェインとなる。
5	Na イオングループ	同一チェインにおいて、Na イオングループ残基 N と非 Na イオングループ残基 N+1 が存在する場合、N と N+1 の間でチェインを区切る。 本処理終了後、Na イオングループ(の塊)は独立したチェインとなる。
6	Cl イオングループ	同一チェインにおいて、Cl イオングループ残基 N と非 Cl イオングループ残基 N+1 が存在する場合、N と N+1 の間でチェインを区切る。 本処理終了後、Cl イオングループ(の塊)は独立したチェインとなる

7	アミノ酸	同一チェインにおいて、以下のいずれかの場合チェインを区切る。 ・ACE 残基での区切り アミノ酸と分類された ACE 残基（N）が存在した場合、N 残基と N-1 残基でチェインを区切る。 ・NME, NHE, NH2 残基での区切り アミノ酸と分類された上記残基（N）が存在した場合、N 残基と N+1 残基でチェインを区切る。 ・OXT 原子での区切り アミノ酸と分類された残基(N)が、OXT 原子を保持している場合、N 残基と N+1 残基でチェインを区切る
8	リガンド（と非リガンド）	同一チェインにおいて、以下のいずれかの場合チェインを区切る。※2 ・リガンド残基(N)と、リガンドの N+1 残基で、リガンド名が異なる場合、N 残基と N+1 残基でチェインを区切る ・リガンド残基(N)が存在し、非リガンドの N-1 残基との全重原子間の距離が 3.0Å より離れている場合、N 残基と N-1 残基でチェインを区切る ・リガンド残基(N)が存在し、非リガンドの N+1 残基との全重原子間の距離が 3.0Å より離れている場合、N 残基と N+1 残基でチェインを区切る ・連続した同名のリガンド残基、N、N+1、N+2、・・・N+M が存在し、以下の場合、チェインを区切る。 残基(N)の全重原子について、N+1～N+M 残基の全重原子との距離が 3.0Å より離れているとき、その残基と他の残基との間でチェインを区切る。

※1…内部でチェイン ID を割り当てる場合、下記メッセージを出力する。

“WARNING: chain id is empty. chain id is assigned automatically.”

※2…リガンドを区切る場合、下記のように、チェイン ID、残基名、残基 ID、チェイン分割の原因(TER/END とチェイン ID の相違による分割は除く)を出力する。

```
INFORMATION> DIVISION OF CHAINS.
  CHAIN NAME   RESIDUE NAME   RESIDUE ID   REASON(EXCEPT TER AND CHAIN ID)
  A     A      LHG      NA     162  1157   TERMINAL OF NA ION GROUP(TOP)
  A     A       NA     LHG    1157  1158   TERMINAL OF NA ION GROUP(LAST)
  A     A      LHG     UMQ    1161  1162   DIFFERENCE OF LIGAND RESIDUE NAME
  A     A      UMQ     LHG    1163  1164   DIFFERENCE OF LIGAND RESIDUE NAME
```

```
A     A     LHG     LHG     1158  1159   DISTANCE IS FAR
A     A     LHG     LHG     1159  1160   DISTANCE IS FAR
A     A     LHG     LHG     1160  1161   DISTANCE IS FAR
A     A     UMQ     UMQ     1162  1163   DISTANCE IS FAR
```

３．２．PDB 出力

以下に、PDB ファイル出力の詳細について説明する。

(1) RECORD TYPE

PDB チェックツールが出力する RECORD TYPE の行は以下のとおりである。

項番	RECORD TYPE	概要
1	REMARK	-remark オプション指定時に、入力 PDB の REMARK 行(REM 行)をまとめて出力する。
2	SSBOND	-ss オプション指定時に、SSBOND を検出した場合、SSBOND レコードを出力する。
3	ATOM	アミノ酸残基を ATOM レコードで出力する。※１ 核酸残基も ATOM レコードで出力する。
4	HETATM	以下を HETATM レコードで出力する。 ・リガンド残基 ・水グループ残基※１ ・Na イオングループ残基※１ ・Cl イオングループ残基※１ ただし、-disableHet 指定時は出力しない。
5	TER	チェインの区切りとして出力する。※２

※１…例えば、アミノ酸と分類された ACE 残基なども ATOM 行で出力される。

※２…水グループ、Na イオングループ、Cl イオングループは、チェイン毎には出力せず、それぞれのグループをまとめて出力する。また、各グループの最後に TER を出力する。ただし、”-keepTer”オプションを指定している場合は、入力 PDB での出現順序で出力される。

(2) シリアル NO / ID の補正

シリアル NO、チェイン ID、残基 ID は、以下のルールに従い値を補正して出力する。

項番	RECORD TYPE	概要
1	シリアル NO	１からの通し番号で、ATOM / HETATM を出力するたびカウントアップする。99999 を超えた場合、１からカウントアップしなおす。 また、TER 行はシリアル NO のカウントアップ対象外とする。
2	チェイン ID	A から始まり、１チェインを出力するたび B、C、D、、と割り振る。Z を超えた場合、A から割り振りなおす。
3	残基 ID	チェイン毎に１から割り振る通し番号で、残基を出力するたびカウントアップする。9999 を超えた場合、１からカウントアップしなおす。

(3) 残基名の補正

一部の残基については、残基名の補正を行い、出力する。

項番	元の残基名	補正後の残基名	補正の条件
1	CYS	CYSS	・-ss オプションを指定している ・SS 結合検出されている ※CYSS で出力するためには Makefile を修正しコンパイル必要がある。 詳細は、「３．４．SS-BOND の検出と補正」を参照のこと。
2	CYS	CYM	・-ss オプションを指定している ・CYM 検出されている ・詳細は、「３．４．SS-BOND の検出と補正」を参照のこと。

(4) その他補足事項

PDB 出力について、その他の補足事項を下記に述べる。

・TER 行は、TER という文字列のみ出力する。
・ATOM 行 / HETATM 行について、tempFactor まで(66 カラム目まで)の情報(質量と電荷に相当)のみ出力する。
・alternate location indicator を示す英字は出力しない。
※入力 PDB に alternate location indicator が存在し、かつ、-alt 加工オプションを指定していない場合、alternate location indicator の情報が失われることに注意すること。

３．３．alternate location indicator の検出と補正

alternate location indicator は当該 PDB ファイルの作成時に、X 線解析で取得した座標が複数発生する場合に挿入される重複した情報を持つ行である。(PDB ファイルでは 1 残基の原子名はユニークである)

図 3.1-(1)では CB, CG, CD1 の 3 原子に対し 17 カラム目に alternate location indicator を示す英字(A,B)が存在する。

```
ATOM    368  N    LEU A  67      13.128   1.687   5.504  1.00 12.15           N
ATOM    369  CA   LEU A  67      13.370   3.044   5.070  1.00 13.59           C
ATOM    370  C    LEU A  67      12.914   3.230   3.627  1.00 13.04           C
ATOM    371  O    LEU A  67      12.884   2.265   2.872  1.00 15.11           O
ATOM    372  CB ALEU A  67      14.853   3.401   5.185  0.52 12.71           C
ATOM    373  CB BLEU A  67      14.843   3.434   5.152  0.48 13.07           C
ATOM    374  CG ALEU A  67      15.419   3.241   6.598  0.52 17.21           C
ATOM    375  CG BLEU A  67      15.680   3.054   6.360  0.48 14.48           C
ATOM    376  CD1ALEU A  67      16.215   1.950   6.720  0.52 17.10           C
ATOM    377  CD1BLEU A  67      17.125   3.519   6.218  0.48 15.34           C
```

図 3.1-(1) alternate location indicator

(1) 対象残基分類

本機能が対象とする残基分類は以下のとおりである。

項番	残基分類	対象	備考
1	アミノ酸	○	-
2	核酸	○	-
3	水グループ	○	-
4	Na イオングループ	○	-
5	Cl イオングループ	○	-
6	リガンド	○	-

(2) alternate locatitoin indicator の表示

入力 PDB ファイルに alternate location indicator を示す英字が存在する場合、alternate location indicator として、残基名、残基 ID、原子名を下記フォーマットで標準出力にログを出力する。

```
INFORMATION> EXIST ALTERNATE LOCATION INDICATOR
  RESIDUE NAME   RESIDUE ID   ATOM NAME
  LEU              67          CB
  LEU              67          CG
  LEU              67          CD1
```

図 3.1-(2) alternate location indicator のログ表示

(3) alternate location indicator の補正

加工指定(-alt)指定があった場合、初めの座標の原子を残し、2 番目以降の座標の原子は削除する。また、下記の修正ログを出力する。

```
INFORMATION> REPAIR ALTERNATE LOCATION INDICATOR
  RESIDUE NAME   RESIDUE ID   ATOM NAME
  LEU              67          CB
  LEU              67          CG
  LEU              67          CD1
```

図 3.1-(3) alternate location indicator の補正のログ表示

３．２．主鎖原子欠損の検出と補正

タンパク質の主鎖原子(N-CA-C-O)のいずれかが不足している場合に主鎖原子の欠損としてログを出力する。

(1) 対象残基分類

本機能が対象とする残基分類は以下のとおりである。

項番	残基分類	対象	備考
1	アミノ酸	○	ACE 残基、NME/NHE/NH2 残基については主鎖原子欠損のチェックを行わない。
2	核酸	×	-
3	水グループ	×	-
4	Na イオングループ	×	-
5	Cl イオングループ	×	-
6	リガンド	×	-

(2) 主鎖原子欠損の表示

１つの残基内に原子名("N", "CA", "C", "O")の原子が存在するかを確認し、存在しない場合下記のフォーマットで残基名、残基 ID、欠損原子を表示する。

```
INFORMATION> LACKED MAIN CHAIN
  RESIDUE NAME   RESIDUE ID   ATOM NAME
  MET              32          C
  MET              32          O
```

図 3.2-(1) 主鎖原子欠損のログ表示

(3) 主鎖原子欠損の補正

加工指定(-bb)指定があった場合、主鎖原子が欠損している残基を削除し、当該残基の出力を抑止する。また、下記のログを出力する。

```
INFORMATION> DELETE INVALID RESIDUE
  RESIDUE NAME   RESIDUE ID
  MET              32
```

図 3.2-(2) 主鎖欠損している残基の削除のログ表示

※アミノ酸残基と分類分けされた ACE 残基、NME/NHE/NH2 残基について主鎖原子欠損の検出および加工を行わない。

３．３．末端残基の検出と補正

タンパク質の１チェインのアミノ酸残基同士の２残基間の主鎖原子(C-N)の距離を計算し、距離が２Åより離れている場合、末端残基としてメッセージを出力する。N 原子または C 原子が存在しない場合、メッセージ"WARNING: main chain invalid: 残基名 残基 ID"を出力する。アミノ酸残基と隣り合った残基が、アミノ酸残基以外の場合、”WARNING: skip terminal residue check: 残基名１ 残基 ID1 残基名２ 残基 ID2”を出力する。

図 3.3-(1)では VAL の C と TYR の N の距離 ((-7.392+4.029)**2 + (11.752-13.510) **2 + (10.569 - 4.941)**2)=6.78 は 2 よりも大きいため末端残基としてログを出力する。

```
ATOM      9  N   VAL A  17      -5.836  10.093  11.314  1.00 20.93           N
ATOM     10  CA  VAL A  17      -7.260  10.331  11.080  1.00 17.41           C
ATOM     11  C   VAL A  17      -7.392  11.752  10.569  1.00 16.64           C
ATOM     12  O   VAL A  17      -6.832  12.677  11.154  1.00 19.77           O
ATOM     13  CB  VAL A  17      -8.094  10.141  12.358  1.00 26.68           C
ATOM     14  CG1 VAL A  17      -9.575  10.371  12.104  1.00 21.77           C
ATOM     15  CG2 VAL A  17      -7.912   8.749  12.939  1.00 23.11           C
ATOM     24  N   TYR A  20      -4.029  13.510   4.941  1.00 18.53           N
ATOM     25  CA  TYR A  20      -3.970  13.447   3.479  1.00 20.66           C
ATOM     26  C   TYR A  20      -2.807  12.547   3.070  1.00 23.87           C
ATOM     27  O   TYR A  20      -1.885  12.281   3.827  1.00 18.98           O
```

図 3.3-(1) 残基中の末端の例

(1) 対象残基分類

本機能が対象とする残基分類は以下のとおりである。

項番	残基分類	対象	備考
1	アミノ酸	○	・-cap 加工オプション指定時に、チェインの N 末端、C 末端に、それぞれ ACE 残基、NME 残基を追加する。ただし、以下の場合は、ACE 残基、NME 残基を追加しない。 ・N 末端、C 末端が非アミノ酸残基である ・N 末端が、ACE 残基である ・C 末端が、NME/NHE/NH2 残基である ・C 末端の残基が OXT 原子を保持している
2	核酸	×	
3	水グループ	×	
4	Na イオングループ	×	
5	Cl イオングループ	×	
6	リガンド	×	

(2) 残基中の末端原子の表示

末端原子を検出した場合、末端の残基、残基 ID、距離を下記のフォーマットで標準出力に出力する。

```
INFORMATION> TERMINAL RESIDUE
  RESIDUE NAME    RESIDUE ID    DISTANCE
  VAL     TYR       17    20        6.78
```

図 3.3-(2) 末端残基のログ表示

(3) 末端残基の補正

加工指定(-cap)が指定された場合、以下の補正を行う。ただし、加工指定(-keepTer)も指定された場合は、チェイン分割は行わない(末端に ACE/NME 残基の追加のみ行う)

・検出した末端原子を持つ 2 つの残基間でチェインを分割する。
・チェインの全末端に下記の補正用原子を追加する。
  ・N 末端残基の直前に残基名が ACE の C 原子、CA 原子を配置する
  ・C 末端残基の直後に残基名が NME の N 原子を配置する

ただし、以下の場合、メッセージ"WARNING: main chain invalid: 残基名 残基 ID"を出力し補正を行わない。
  ・N 末端残基に N 原子、CA 原子、C 原子のいずれかが存在しない
  ・C 末端残基に主鎖原子(N,CA,C,O)のいずれかが存在しない

また、以下の場合についても、補正を行わない(この時、メッセージは出力しない)。
  ・チェインの N 末端残基が ACE 残基である、またはアミノ酸残基ではない
  ・チェインの N 末端残基が NME / NHE / NH2 残基である、またはアミノ酸残基ではない。または、アミノ酸残基であるが OXT 原子を保持している。

なお、末端残基の補正処理で追加した原子(残基)の残基 ID には-1 を設定している。以後のチェックまたは補正処理のログで残基 ID が-1 と表示されたものは、本処理で追加した原子(残基)である。ただし、PDB 出力時にはチェイン毎に 1 から割り振られた残基 ID が出力される。

末端を補正した場合、対象残基の残基名、残基 ID、追加残基を下記のフォーマットで標準出力に出力する。

```
INFORMATION> CAPED TERMINAL RESIDUE
  RESIDUE NAME   RESIDUE ID   CAP
  VAL              17         ACE
  VAL              17         NME
  VAL              20         ACE
  VAL              20         NME
```

図 3.3-(5) 末端残基の補正ログ表示

(2-a)N 末端の場合

N 末端の N 原子の直前に残基名 ACE、C 原子と CA 原子を配置する。

```
ATOM     22  CA  ACE A  19      -5.215  14.453   7.487
ATOM     23  C   ACE A  19      -4.782  14.104   6.094
ATOM     24  N   TYR A  20      -4.029  13.510   4.941
ATOM     25  CA  TYR A  20      -3.970  13.447   3.479
ATOM     26  C   TYR A  20      -2.807  12.547   3.070
ATOM     27  O   TYR A  20      -1.885  12.281   3.827
```

図 3.3-(3) ACE 残基の配置例

配置する C 原子の座標は、N 末端残基の C→N と同一ベクトルで、距離 1.5Åの位置に配置する。

下図の C-N 間のベクトルを v1(x1, y1, z1)、v1 の長さを V、N の座標を N1(nx,ny,nz)とした場合、追加する C の座標は(nx+x1/V*1.5, ny+y1/V*1.5, nz+z1/V*1.5)となる

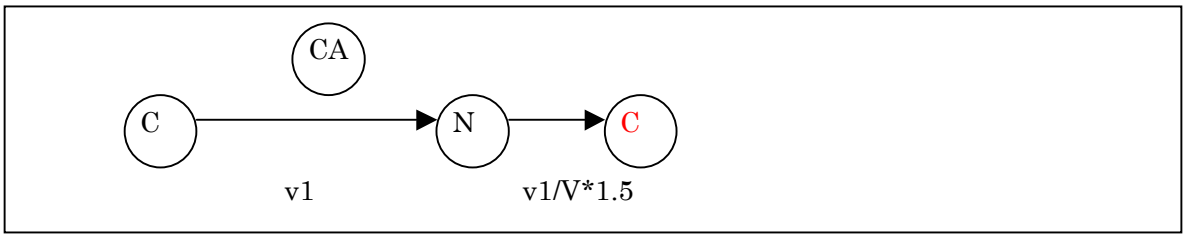


図 3.3-(4) ACE 残基の C 原子配置例

CA 原子の座標は、N 末端の残基の CA と ACE 残基に追加した C 原子、CA→C と同一ベクトルで、距離 1 .5Åの位置に配置する。

下図の CA-C 間のベクトルを v2(x2, y2, z2)、v2 の長さを V2、C の座標を C2(cx2,cy2,cz2)とした場合、追加する C の座標は(cx2+x2/V2*1.5, cy2+y2/V2*1.5, cz2+z2/V2*1.5)となる

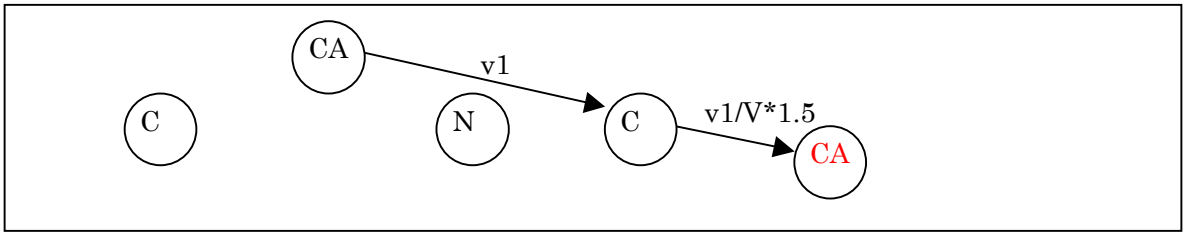


図 3.3-(5) ACE 残基の CA 原子配置例

(2-b)C 末端の場合

C 末端の C 原子の直後に残基名 NME、残基 ID+1 の N 原子を配置する。

```
ATOM      9  N   VAL A  17      -5.836  10.093  11.314
ATOM     10  CA  VAL A  17      -7.260  10.331  11.080
ATOM     11  C   VAL A  17      -7.392  11.752  10.569
ATOM     12  O   VAL A  17      -6.832  12.677  11.154
ATOM     13  N   NME A  18      -8.209  12.035   9.344
```

図 3.3-(6) NME 残基の追加例

追加する N 原子(以下では Nx で表記)の座標は、z-matrix により決定する。z-matrix のパラメータは以下のとおりとする。

- bond 距離：追加する Nx と C 末端残基の C 原子の距離を 1.5Å とする。
- angle 角：Nx-C-CA 間の angle 角は 120.0° とする。
- dihedral 角：Nx-C-CA-N の dihedral 角は、O-C-CA-N の dihedral 角+180° (点対称)の位置とする。

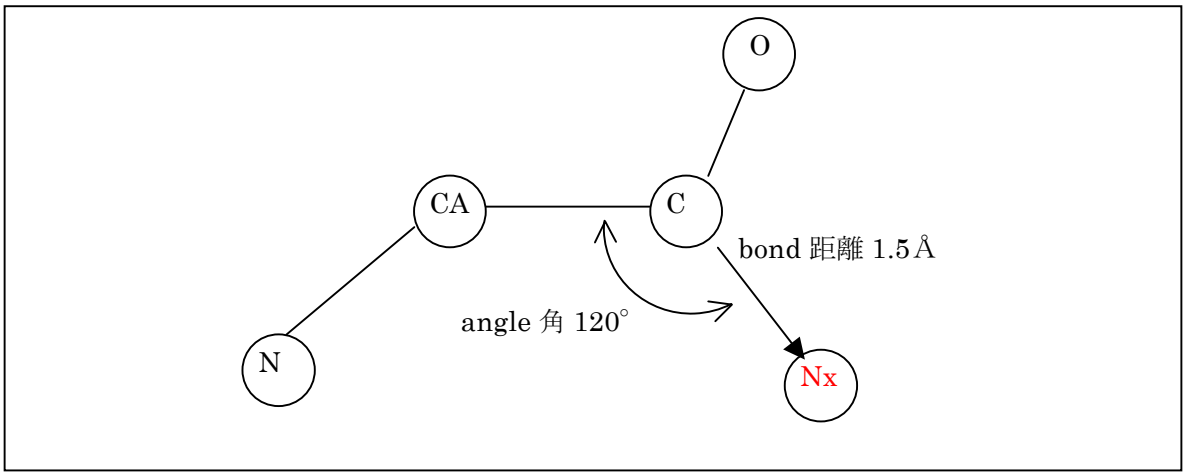


図 3.3-(7) NME 残基の N 原子配置例

３．４．SS-BOND の検出と補正

全チェインの CYS 残基の S 原子(硫黄)間の距離が 4.5Å以内の場合に最も近い SS 結合候補のメッセージを出力する。

CYS-(CYM)残基が存在している場合メッセージを出力する。なお、CYS-(CYM)残基の判定については、「(4) CYS-(CYM)の表示」を参照のこと。

(1) 対象残基分類

本機能が対象とする残基分類は以下のとおりである。

項番	残基分類	対象	備考
1	アミノ酸	○	-
2	核酸	×	-
3	水グループ	×	-
4	Na イオングループ	×	-
5	Cl イオングループ	×	-
6	リガンド	×	-

(2) SS-BOND の表示

SS-BOND の候補に対し、"SSBOND", ID(1 から昇順)、残基 1 の名称("CYS")、チェイン ID、残基 ID、残基 2 の名称("CYS")、チェイン ID、残基 ID を表示する。

```
INFORMATION> SSBOND CANDIDATES
SSBOND   1 CYS A   22    CYS A   157
```

図 3.4-(1)　SSBOND のログ表示

(3) SSBOND 情報の出力

加工指定(-ss)指定があった場合、下記のメッセージを出力する。また、同一残基内に HS 原子(HG)が存在する場合、HS 原子(HG)を削除し、HS 原子(HG)情報の出力を抑止する。

```
INFORMATION> APPEND SSBOND LINE
SSBOND   1 CYS A   22    CYS A   157
```

図 3.4-(2) SSBOND 行出力のログ表示

また、出力 PDB ファイルの先頭に同じ文字列を出力する。

※上記で出力するメッセージのチェイン ID、残基 ID と、出力 PDB ファイルに出力されているチェイン ID、残基 ID は異なる場合があることに注意すること。(メッセージ中のチェイン ID、残基 ID は入力 PDB ファイルに記述されていた ID を表示するが、出力 PDB ファイルのチェイン ID、残基 ID は ID を割り振りなおしたもので出力される)

SS 結合している CYS 残基の残基名を CYSS として出力する場合は、Makefile の FFLAGS に「-D ENABLE_RENAME_CYSS」を追加し make コマンドを再実行する必要がある。(SS 結合した CYS 残基を CYSS として出力した PDB を tplgene 入力するとエラーになる。そのため、デフォルトでは、CYS 残基名で出力する。)

```
FFLAGS = -Warn -D ENABLE_RENAME_CYSS
```

(4) CYS-(CYM)の表示

CYS-(CYM)の候補に対し、チェイン ID、残基名("CYS")、残基 ID を表示する。

なお、本プログラムでは、以下の残基を CYS-(CYM)の候補とみなす。

・SSBOND 結合していない、S 原子を持った CYS 残基である

・入力 PDB に水素原子が付与されている

(CYS 残基が HA 原子を保持している場合、水素原子が付与されているとみなす)

・HG 原子は保持していない

また、入力 PDB 上で CYM 残基と記述されているものは検出対象外である。

```
INFORMATION> CYS-(CYM) CANDIDATES
  CHAIN ID   RESIDUE NAME   RESID ID
  A          CYS              95
```

図 3.4-(1) SSBOND のログ表示

※tplgene で SS 結合した PDB ファイルを作成した場合の注意点

SS 結合している PDB を tplgene で作成し、2 つの CYS 残基の SG 原子間の距離が 2.0 Å より離れている場合、PDB チェックツールでは、2 つの CYS 残基を CYM と判定することに注意すること。これは、tplgene が出力した CYS 残基が、HA 原子を保持しており、かつ HG 原子を保持しておらず、また、PDB チェックツールとしては、SSBOND していないとみなすためである。

(5) CYS-(CYM)残基の残基名補正

加工指定(-ss)指定があった場合、下記のメッセージを出力し、出力 PDB での残基名を CYM で出力する。

```
INFORMATION> REPAIR CYS-(CYM) CANDIDATES
  CHAIN ID   RESIDUE NAME   RESID ID
  A          CYS              95
```

図 3.4-(2) SSBOND 行出力のログ表示

※上記で出力するメッセージのチェイン ID、残基 ID と、出力 PDB ファイルに出力されているチェイン ID、残基 ID は異なる場合があることに注意すること。(メッセージ中のチェイン ID、残基 ID は入力 PDB ファイルに記述されていた ID を表示するが、出力 PDB ファイルのチェイン ID、残基 ID は ID を割り振りなおしたもので出力される)

３．５．近接原子の検出と補正

水素原子に対し、従属する重原子以外(他の残基の原子)が 1Å以内に存在する場合、近接原子としてメッセージを出力する。

(1) 対象残基分類

本機能が対象とする残基分類は以下のとおりである。

項番	残基分類	対象	備考
1	アミノ酸	○	-
2	核酸	○	-
3	水グループ	×	水グループ、Na イオングループ、Cl イオングループについては以下のとおりとする。 ・水素原子を保持していたとしても、近接原子の確認を行わない。 ・ある水素原子に対する、近接原子の確認対象としても対象外である。
4	Na イオングループ	×	
5	Cl イオングループ	×	
6	リガンド	○	-

(2) 近接原子の表示

全残基に対し、近接原子の存在する水素原子の残基名、残基 ID、原子名と、近接原子の残基名、残基 ID、原子名を表示する。

```
INFORMATION> EXIST NEAR ATOM
  RESIDUE NAME   RESIDUE ID   ATOM NAME
  LEU     GLY      67   100    HB1  HD1
```

図 3.5-(1)　近接原子のログ表示

(3) 近接原子の補正

加工指定(-hyd)指定があった場合、下記のメッセージを出力し、水素原子の座標を補正する。ただし、同一残基中に重原子(水素以外の原子)が存在しない場合、メッセージ"WARNING: can not repair near atom: 残基名 残基 ID 原子名"を出力し座標を補正しない。

```
INFORMATION> REPAIRED NEAR ATOM
  RESIDUE NAME   RESIDUE ID   ATOM NAME
  LEU              67          HB1
```

図 3.5-(2) 近接原子の補正ログ表示

補正後の座標は、同一残基内の最も近い重原子(水素以外の原子)を検出し、この原子と水素のベクトルを求め、距離を半分にする。

下図の N-H 間のベクトルを v1(x1, y1, z1)、N の座標を N1(nx,ny,nz)とした場合、補正後の H の座標は(nx+(x1/2), ny+(y1/2), nz+(z1/1))となる

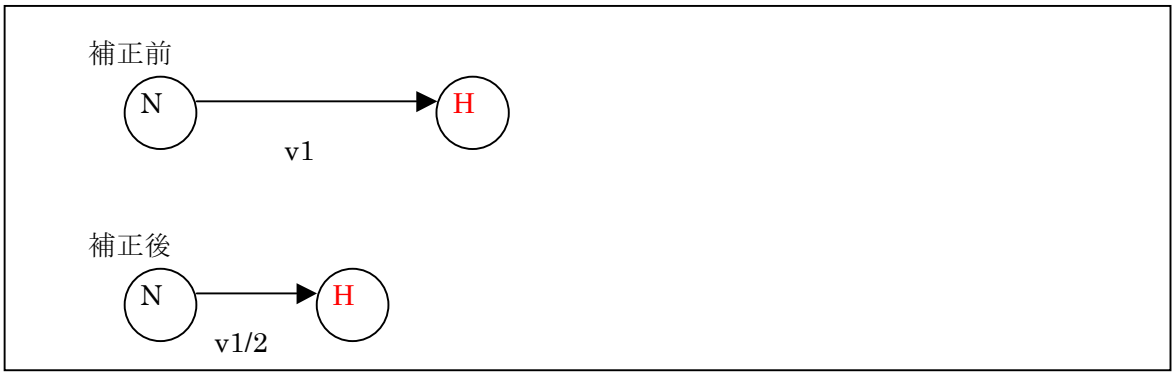


図 3.5-(3) 近接原子の座標補正

３．６．直行する二面角の検出と補正

タンパク質の主鎖原子が(N-CA-C-N-CA-C-N-...)の２面角が直行している場合に二面角の直行のメッセージを出力する。

(1) 対象残基分類

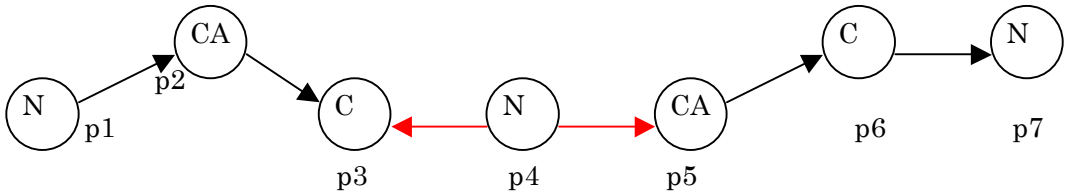
本機能が対象とする残基分類は以下のとおりである。

項番	残基分類	対象	備考
1	アミノ酸	○	-
2	核酸	×	-
3	水グループ	×	-
4	Na イオングループ	×	-
5	Cl イオングループ	×	-
6	リガンド	×	-

(2) 直行する二面角の表示

チェインの N-CA-C に対し点 p を割り当て(p1=1 番目の N の座標、p2=1 番残基の CA の座標、p3=1 番残基の C の座標、p4=2 番残基の N の座標、p5=2 番目残基の CA の座標、･･･)、p2-p1 のベクトルと p2-p3 のベクトルが同一方向かまたは完全な逆方向かを判定する。

下記の例では、p4->p3 と p4->p5 のベクトルが完全な逆方向であり、3 点が平面を構成しないため、直行する二面角を構成する原因となる。




```
INFORMATION> LINEAR POSITION ATOM
RESIDUE ID         ATOM NAME
1   2   2         C   N   CA
```

図 3.6-(1)直行する二面角のログ表示

(3) 直行する二面角の補正

加工指定(-dih)があった場合、2 番目の原子の座標(x,y,z)を乱数(-0.1～0.1 の範囲)で移動させる。ログに下記の情報を出力する。

```
INFORMATION> REPAIR LINEAR POSITION ATOM
RESIDUE ID          ATOM NAME
2                   N
```

図 3.6-(2) 直行する二面角補正のログ表示

ただし、p1 と p3 が同一座標の場合、メッセージ” WARNING: can not repair linear position atom: p2 の残基 ID p2 の原子名”を出力し補正を行わない。

３．詳細設計

３．１．ファイル構成

pdbcheck のソースに以下の機能を追加する。

項番	ファイル名	手続き名	機能	区分
1	analysis.f	Check_NearAminoLigand	蛋白・化合物距離計算	非使用
2		Get_ResidType	残基名の整数化	非使用
3		Get_AtomType	原子タイプ名の整数化	非使用
4		Set_SybylType	化合物への原子タイプ設定	非使用
5		Set_StructureData	チェイン、残基構造の生成	流用
6		Set_ShapeKind	主鎖形状設定	非使用
7		Count_ContactAminoLigand	蛋白・化合物接触の計測	非使用
8		Set_ResidGroup	分子分類の設定	流用
9		Update_ChainStructureData	チェイン構造の更新	流用
10		Analyze_RigandTerminal	リガンド分子によるチェイン構造の更新	流用
11	io.f	Input_PDB	PDB ファイル入力	流用
12		Input_MOL2	MOL2 ファイル入力	非使用
13		Output_Histogram	ヒストグラム出力	非使用
14		Output_PDB	PDB ファイル出力	流用
15		Output_AppearanceNumber	接触表出力	非使用
16	math.f	Cross_Product	外積計算	非使用
17		Calc_Angle	偏角計算	非使用
18		Calc_Torsion	二面角計算	非使用
19		Calc_Distance	2 点間距離計算	流用
20		Compare_Distance	2 点間距離の比較	流用
21	check.f	Insert_Resid	新規残基の追加	流用
22		Devide_Chain	チェインの分割	流用
23		Delete_Atom	原子の削除	流用
24		Insert_Atom	原子の追加	流用
25		Find_AtomIDByName	原子 ID の検索	流用
26		Check_Alternate	alternate location indicator チェック	流用
27		Check_MainChain	主鎖原子の欠損チェック	流用
28		Check_TerminalResid	末端残基チェック	流用

29		Check_SSBond	SSBond チェック	流用
30		Check_Contact	水素原子の近接チェック	流用
31		Check_LinearTorsion	二面角チェック	流用
32		Mutate_Residue	残基種の変更	新規
33		Check_Residue	残基の変更	新規
34	check_main.f	Check_PDB	主制御	流用
35	residUtil.f90	Read_TopologyModel	トポロジーの読み込み	新規
36		Search_Atom	原子の検索	新規
37		Get_ResidueModel	残基モデル取得	新規
38		Set_AcceptDonar	化合物の水素ドナーアクセプター解析	新規
39		Is_BridgeOfGraph	非環構造判定	新規
40		Count_HydrogenBond	水素結合数のカウント	新規
41		Get_SideChainDonarAcceptor	残基の水素ドナーアクセプター解析	新規
42		Try_RotateTerminal	残基末端の回転	新規
43		Get_NearMetalResidue	金属周辺残基の検索	新規

３．２．変数構成

PDB チェックツールでは蛋白の情報を原子(atom) 、残基(residue)、チェイン(chain)の３つのレベルで表現する。

(1)原子情報

PDB から入力した情報は原子情報として下記表の配列に登録する。

配列の n 番目の要素は n 番目の原子の情報を表し、原子数は変数 atomNum に登録する。

項番	内容	変数宣言	型
1	原子名	atomName(MAX_ATOM)	character*4
2	残基名	residName(MAX_ATOM)	character*4
3	チェイン名	chainName(MAX_ATOM)	character*2
4	残基 ID	residID(MAX_ATOM)	integer
5	原子座標	coordinate(3,MAX_ATOM)	real
6	Aternate location 識別子	altLoc(MAX_ATOM)	character
7	質量	mass(MAX_ATOM)	real
8	電荷	charge(MAX_ATOM)	real

(2)残基情報

残基情報は各残基の先頭原子、最終残基をポイントする二つの変数で表現する。

配列の n 番目の要素は n 番目の残基の情報を表し、残基数は変数 residNum に登録する。

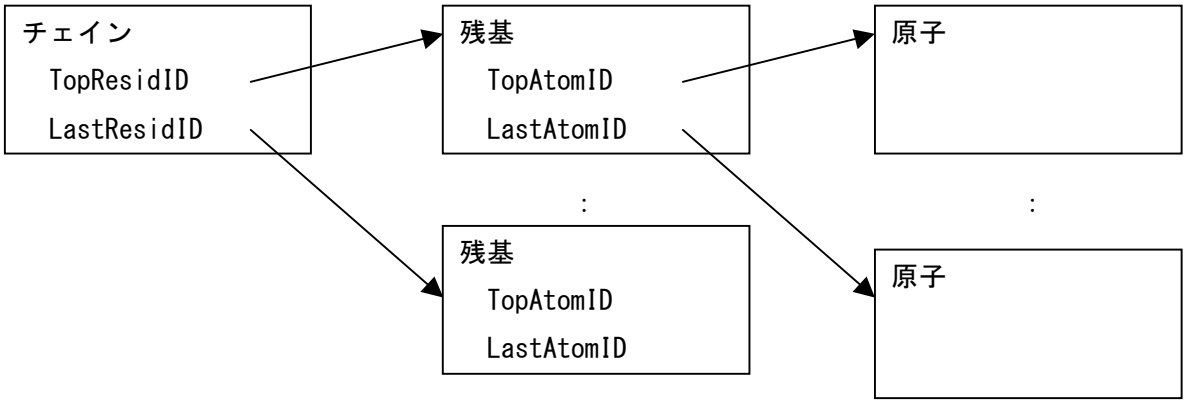
また、各残基の分類を残基分類に登録する。

項番	内容	変数宣言	型
1	先頭原子	topAtomID(MAX_RESID)	integer
2	最終原子	lastAtomID(MAX_RESID)	integer
3	残基分類	residGroup(MAX_RESID)	integer

(3)チェイン情報

チェイン情報は各チェインの先頭残基、最終残基をポイントする二つの変数で表現する。配列の n 番目の要素は n 番目のチェインの情報を表し、チェイン数は変数 chainNum に登録する。

項番	内容	変数宣言	型
1	先頭残基	topResidID(MAX_RESID)	integer
2	最終残基	lastResidID(MAX_RESID)	integer



(4)オプション情報

ユーザがオプションで指定した情報は下記の変数に格納する。

項番	オプション	変数宣言	型
1	-alt	altFlag	logical
2	-cap	capFlag	logical
3	-hyd	hydFlag	logical
4	-ss	ssFlag	logical
5	-bb	bbFlag	logical
6	-dih	dihFlag	logical
7	-disableHet	disableHetFlag	logical
8	-keepTer	keepTerFlag	logical
9	-remark	remarkFlag	logical
10	-mod	modifyFlag	logical
11	-rot	rotateFlag	logical
12	-mut	mutateFlag	logical
13	-cmp	complexFlag	logical

(5)SSBOND 情報

SSBOND の情報は下記変数に設定する。

配列の n 番目の要素は n 番目の SSBOND 情報を表し、SSBOND 数は変数 ssNum に登録する。オリジナルの SSBOND は orgssAtom に設定し個数は orgssNum に登録する。

項番	内容	変数宣言	型
1	「(2)残基情報」に対応する残基の ID(配列の添え字)	ssResidID(2, MAX_RESID)	integer
2	オリジナルの SS 結合	orgssAtom(2,MAX_RESID)	integer

(6)CYS-(CYM)情報

CYS-(CYM)の情報は下記変数に設定する。

配列の n 番目の要素は n 番目の CYS-(CYM)情報を表し、CYS-(CYM)数は変数 cymNum に登録する。

項番	内容	変数宣言	型
3	「(2)残基情報」に対応する残基の ID(配列の添え字)	cymResidID(MAX_RESID)	integer

３．３．モジュール仕様

３．３．１．check_main.f

(1)Check_PDB

呼び出し形式：program Check_PDB

引数：

なし

戻り値：

なし

機能：

PDB チェックツールの主制御を行う。

(1) 標準入力から入力 PDB ファイル名、出力 PDB ファイル名、オプションを読み込む。
(2) PDB ファイルを入力し、チェイン、残基の情報を設定する(Input_PDB 呼び出し)。
(3) 読み取った残基情報について、残基分類を設定する(Set_ResidGroup)。
(4) PDB データを解析し、チェイン構造を更新する(Update_ChainStructureData 呼び出し
(5) リガンド分子を解析し、チェイン構造を更新する(Analyze_RigandTerminal 呼び出し)。
(6) alternate location indicator のチェックを行う(Chack_Alternate 呼び出し)。
(7) 主鎖原子の欠損のチェックを行う(Chack_MainChain 呼び出し)。
(8) 末端残基のチェックを行う(Chack_TerminalResid 呼び出し)。
(9) SSBOND のチェックを行う(Chack_SSBond 呼び出し)。
(10) 水素原子の近接チェックを行う(Chack_Contact 呼び出し)。
(11) 二面角のチェックを行う(Chack_LinearTorsion 呼び出し)。
(12) PDB ファイルの出力を行う(Output_PDB 呼び出し)。

３．３．３．check.f

(1) subroutine Insert_Resid(
targetChainID, targetResidID,
chainNum, topResidID,lastResidID,
residNum, topAtomID, lastAtomID, residGroup)

引数：

integer targetChainID	挿入対象チェイン ID
integer targetResidID	挿入対象残基 ID
integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer residNum	残基数
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
integer residGroup(MAX_RESID)	n 番目残基の残基分類

戻り値：
なし

機能：
指定位置(targetChainID, targetResidID)に新規の残基を追加する。
また、残基の挿入に伴い、下記の変数の変更を行う。
・topResidID, lastResidID：targetResidID 以降の番号はシフトする。
・topAtomID, lastAtomID : targetResidID 以降はシフトする。新規の残基の番号は top=0, last=-1 を設定する。
・residNum： 1 つ増やす

例：
以下の 3 つのチェインに、それぞれ 3 つの残基が存在する構造を例に、追加のイメージを示す。
鎖 1[残 1、残 2、残 3]、鎖 2[残 4、残 5、残 6]、鎖 3[残 7、残 8、残 9]

target ChainID	Target ResidID	処理結果 (▼の箇所に残基(topAtomID=0, lastAtomID=-1)が追加される)
1	1	鎖 1[▼、残 1、残 2、残 3]、鎖 2[残 4、残 5、残 6]、鎖 3[残 7、残 8、残 9]
1	2	鎖 1[残 1、▼、残 2、残 3]、鎖 2[残 4、残 5、残 6]、鎖 3[残 7、残 8、残 9]
1	3	鎖 1[残 1、残 2、▼、残 3]、鎖 2[残 4、残 5、残 6]、鎖 3[残 7、残 8、残 9]
1	4	鎖 1[残 1、残 2、残 3、▼]、鎖 2[残 4、残 5、残 6]、鎖 3[残 7、残 8、残 9]

1	5	エラーとなり、プログラムが停止する。
2	4	鎖 1[残 1、残 2、残 3]、鎖 2[▼、残 4、残 5、残 6]、鎖 3[残 7、残 8、残 9]
2	5	鎖 1[残 1、残 2、残 3]、鎖 2[残 4、▼、残 5、残 6]、鎖 3[残 7、残 8、残 9]
3	9	鎖 1[残 1、残 2、残 3]、鎖 2[残 4、残 5、残 6]、鎖 3[残 7、残 8、残 9、▼]
3	10	エラーとなり、プログラムが停止する。

(2) subroutine Devide_Chain(
targetChainID, targetResidID,
chainNum, topResidID, lastResidID)

引数：

integer targetChainID 分割対象チェイン ID
integer targetResidID 分割対象残基 ID
integer chainNum チェイン数
integer topResidD(MAX_CHAIN) n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN) n 番目チェインの最終残基 ID

戻り値：

なし

機能：

指定位置(targetChainID, targetResidID)でチェインを分割する。具体的には、targetResidID と targetResidID +1 の残基間でチェインを分割する。
また、チェインの分割に伴い、下記の変数の変更を行う。

・topResidID, lastResidID：分割対象のチェインの lastResidID を更新する。
分割対象チェインの直後に新規のチェインを追加する。
分割対象チェイン以降の配列は+1 方向にシフトする。

・chainNum：1 つ増やす。

例：

以下の 3 つのチェインに、それぞれ 3 つの残基が存在する構造を例に、チェイン分割のイメージを示す。

鎖 1[残 1、残 2、残 3]、鎖 2[残 4、残 5、残 6]、鎖 3[残 7、残 8、残 9]

target ChainID	Target ResidID	処理結果 (鎖 M と鎖 N が分割後の状態)
1	1	鎖 M[残 1] 鎖 N[残 2、残 3]、鎖 2[残 4、残 5、残 6]、鎖 3[残 7、残 8、残 9]
1	2	鎖 M[残 1、残 2] 鎖 N[残 3]、鎖 2[残 4、残 5、残 6]、鎖 3[残 7、残 8、残 9]
1	3	エラーとなり、プログラムが停止する。

2	3	エラーとなり、プログラムが停止する。
2	4	鎖 1[残 1、残 2、残 3]、鎖 M[残 4] 鎖 N[残 5、残 6]、鎖 3[残 7、残 8、残 9]
2	5	鎖 1[残 1、残 2、残 3]、鎖 M[残 4、残 5] 鎖 N[残 6]、鎖 3[残 7、残 8、残 9]
2	6	エラーとなり、プログラムが停止する。
3	8	鎖 1[残 1、残 2、残 3]、鎖 2[残 4、残 5、残 6]、鎖 M[残 7、残 8] 鎖 N[残 9]
3	9	エラーとなり、プログラムが停止する。

(3) subroutine Delete_Atom(
chainID, residID, atomID, atomNum,
topAtomID, lastAtomID,
atomName, chainName, residName, residID, coordinate,
mass, charge)

引数：

integer chainID	削除対象原子のチェイン ID
integer residID	削除対象原子の残基 ID
integer atomID	削除対象原子 ID
integer atomNum	原子数
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
character*4 atomName(MAX_ATOM)	原子名
character*2 chainName(MAX_ATOM)	チェイン名
charactrer*4 residName(MAX_ATOM)	残基名
integer residID(MAX_ATOM)	残基 ID
real coordinate(3,MAX_ATOM)	原子座標
real mass(MAX_ATOM)	原子の質量
real charge(MAX_ATOM)	原子の電荷

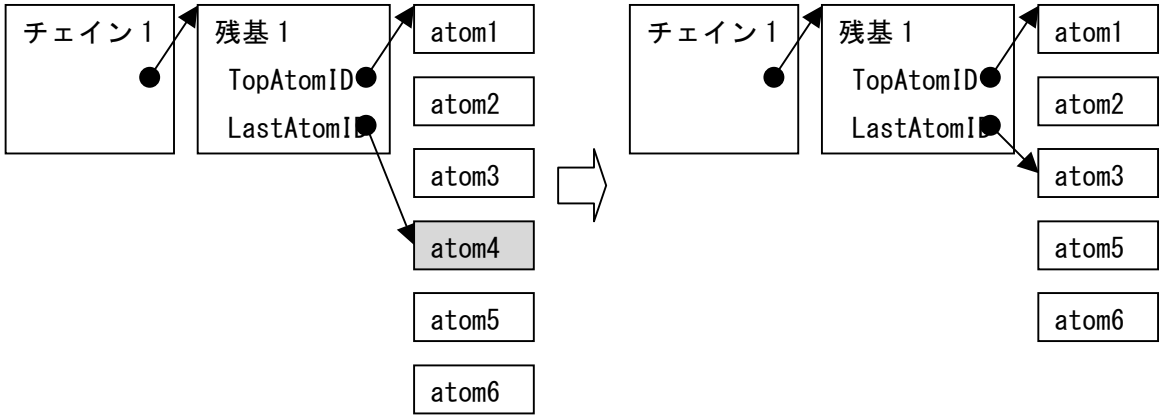
戻り値：
なし
機能：
指定位置(chainID, residID,atomID)の原子を削除する。
また、原子の削除に伴い、下記の変数の変更を行う。

・topAtomID, lastAtomID：削除対象の残基の topAtomID, lastAtomID を更新する。
削除対象残基以降の topAtomID, lastAtomID を更新する。
削除により、削除対象の残基に原子が存在しなくなる場合は、

対象残基に対し、top=0, last=-1 を設定する。

・その他原子情報配列：削除対象原子に対し-1 シフトする。

・atomNum：1 つ減らす。



(4)subroutine Insert_Atom(

chainID, residID, atomID,

newName,

topAtomID, lastAtomID,

atomName, chainName, residName, residID, coordinate,

mass, charge)

引数：

integer chainID	追加対象原子のチェイン ID
integer residID	追加対象原子の残基 ID
integer atomID	追加対象原子 ID
character*4 atomName	追加原子名
real coordinate(3)	追加原子座標
integer atomNum	原子数
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
character*4 atomName(MAX_ATOM)	原子名
character*2 chainName(MAX_ATOM)	チェイン名
charactrer*4 residName(MAX_ATOM)	残基名
integer residID(MAX_ATOM)	残基 ID
real coordinate(3,MAX_ATOM)	原子座標
real mass(MAX_ATOM)	原子の質量
real charge(MAX_ATOM)	原子の電荷

戻り値：

なし

機能：

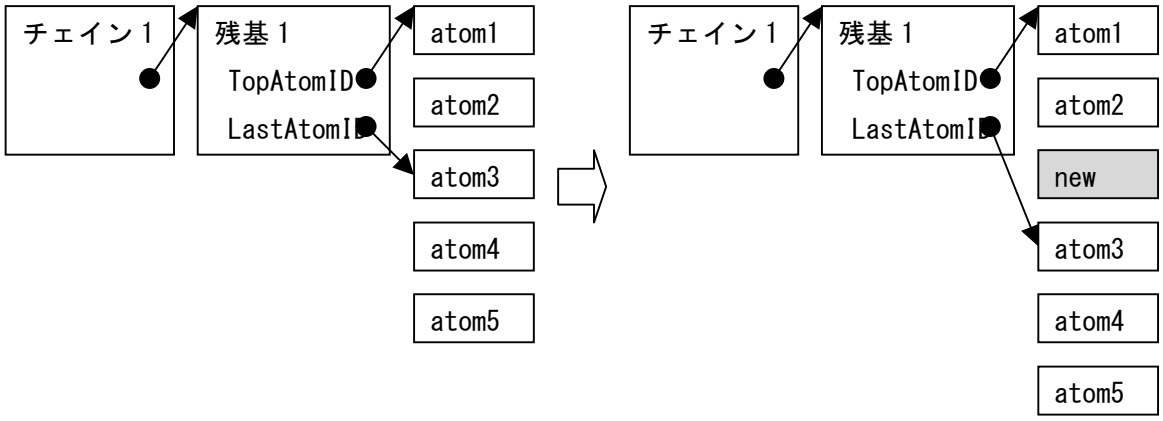
指定位置(chainID, residID,atomID)に原子を追加する。

また、原子の追加に伴い、下記の変数の変更を行う。

・topAtomID, lastAtomID：追加対象の残基の topAtomID, lastAtomID を更新する。
追加対象残基以降の topAtomID, lastAtomID を更新する。

・その他原子情報配列：追加対象原子に対し+1 シフトする。

・atomNum：1 つ増やす。



(5)function Find_AtomIDByName(
targetAtomName,
topAtomIndex,
lastAtomIndex,
atomName,
isExactMatch)

引数：

character*4 targetAtomName	追加対象原子のチェイン ID
integer topAtomIndex	検索開始位置の INDEX
integer lastAtomIndex	検索終了位置の INDEX
character*4 atomName(MAX_ATOM)	原子名
logical isExactMatch	検索方法

戻り値：

検索にヒットした原子の INDEX(添え字)。ヒットしない場合は-1。

機能：

atomName の配列から、topAtomIndex と lastAtomIndex で指定された範囲で、targetAtomName で指定された原子名を持つ原子の INDEX を検索する。検索にヒットしない場合は-1 を返す。

なお、isExactMatch に.TRUE.を指定した場合は、原子名の完全一致で検索をする。.FALSE.を指定された場合は、原子名の前方一致で検索する。

(5)subroutine Check_Alternate(

chainNum, topResidID, lastResidID,
residNum, topAtomID, lastAtomID,
atomNum, atomName, residName,
residID, chainName, coordinate, mass, charge,
altLoc, altFlag)

引数：

integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer residNum	残基数
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
integer atomNum	原子数
character*4 atomName(MAX_ATOM)	原子名
charactrer*4 residName(MAX_ATOM)	残基名
integer residID(MAX_ATOM)	残基 ID
character*2 chainName(MAX_ATOM)	チェイン名
real coordinate(3,MAX_ATOM)	原子座標
real mass(MAX_ATOM)	原子の質量
real charge(MAX_ATOM)	原子の電荷
character altLoc(MAX_ATOM)	Alternate location indicator の値
logical altFlag	修正フラグ

出力：

なし

機能：

各原子に対応する altLoc の値が存在する場合、メッセージを出力する。

オプション(-alt)指定の場合には重複座標原子について、２番目以降の原子を削除する。

(6)subroutine Check_MainChain (
chainNum, topResidID, lastResidID,
residNum, topAtomID, lastAtomID, residGroup,
atomNum, atomName, residName,
residID, chainName, coordinate, mass, charge
bbFlag)

引数：

integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer residNum	残基数
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
integer residGroup(MAX_RESID)	n 番目残基の残基分類
integer atomNum	原子数
character*4 atomName(MAX_ATOM)	原子名
charactrer*4 residName(MAX_ATOM)	残基名
integer residID(MAX_ATOM)	残基 ID
character*2 chainName(MAX_ATOM)	チェイン名
real coordinate(3,MAX_ATOM)	原子座標
real mass(MAX_ATOM)	原子の質量
real charge(MAX_ATOM)	原子の電荷
logical bbFlag	修正フラグ(back bone)

出力：

なし

機能：

残基分類が、アミノ酸の残基について、同一残基内に N, CA, C, O が存在しない場合、メッセージを出力する。

オプション(-bb)指定の場合には、残基内の全原子を削除し、残基自体も削除する。

(7)subroutine Check_TerminalResid(

chainNum, topResidID, lastResidID,
residNum, topAtomID, lastAtomID, residGroup
atomNum, atomName, residName,
residID, chainName, coordinate, mass, charge
capFlag, keepTerFlag)

引数：

integer chainNum チェイン数
integer topResidD(MAX_CHAIN) n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN) n 番目チェインの最終残基 ID
integer residNum 残基数
integer topAtomID(MAX_RESID) n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID) n 番目残基の最終原子 ID
integer residGroup(MAX_RESID) n 番目残基の残基分類
integer atomNum 原子数
character*4 atomName(MAX_ATOM) 原子名
charactrer*4 residName(MAX_ATOM) 残基名
integer residID(MAX_ATOM) 残基 ID
character*2 chainName(MAX_ATOM) チェイン名
real coordinate(3,MAX_ATOM) 原子座標
real mass(MAX_ATOM) 原子の質量
real charge(MAX_ATOM) 原子の電荷
logical capFlag 修正フラグ(-cap)
logical keepTerFlag 修正フラグ(-keepTer)

出力：

なし

機能：

１チェインのアミノ酸残基の 2 残基間の主鎖原子(C-N)の距離を計算し、２Åより離れている場合、末端残基として、メッセージを出力する。

ただし、残基分類がアミノ酸と隣り合っていないリガンド残基についても距離を計算しない。

オプション(-cap)指定されかつ、オプション(-keepTer)が指定されていない場合には、上記の末端残基間でチェインを分割する。

また、オプション(-cap)指定されている場合は、各チェインの N 末端、C 末端への原子の追加を行う。

ただし、以下の場合は N 末端または C 末端への原子の追加は行わない。

・ N 末端が ACE 残基または、非アミノ酸残基の場合は原子の追加を行わない。
・ C 末端が、NME/NHE/NH2 残基の場合、または非アミノ酸残基の場合、OXT 原子を保持している場合は原子の追加を行わない。

(8)subroutine Check_SSBond(
chainNum, topResidID, lastResidID,
residNum, topAtomID, lastAtomID, residGroup,
atomNum, atomName, residName,
residID, chainName, coordinate, mass, charge
ssNum, ssResidID,
cymNum, cymResidID,
ssFlag)

引数：

integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer residNum	残基数
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
integer residGroup(MAX_RESID)	n 番目残基の残基分類
integer atomNum	原子数
character*4 atomName(MAX_ATOM)	原子名
charactrer*4 residName(MAX_ATOM)	残基名
integer residID(MAX_ATOM)	残基 ID
character*2 chainName(MAX_ATOM)	チェイン名
real coordinate(3,MAX_ATOM)	原子座標
real mass(MAX_ATOM)	原子の質量
real charge(MAX_ATOM)	原子の電荷
integer ssNum	SS 結合数
integer ssResidID(2,MAX_RESID)	SS 結合の残基 ID
integer cymNum	CYS-(CYM)数
integer cymResidID(MAX_RESID)	CYS-(CYM)の残基 ID
logical ssFlag	修正フラグ

出力：
なし

機能：

全チェイン間で CYS 残基の S 原子の距離が 2.0Å以内の場合、メッセージを出力する。オプション(-ss)指定の場合にはSSBOND 情報を作成し、HS 原子(HG)が存在する場合は、削除する。

また、SS 結合していない CYS 残基について、HA 原子を保持し、かつ HG 原子を保持していない場合、CYM 検出のメッセージを出力する。オプション(-ss)指定の場合には、CYS-(CYM)情報を作成する。

(10) subroutine Check_Contact(
chainNum, topResidID, lastResidID,
topAtomID, lastAtomID, residGroup,
atomName, residName, residID, coordinate,
hydFlag)

引数：

integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
integer residGroup(MAX_RESID)	n 番目残基の残基分類
character*4 atomName(MAX_ATOM)	原子名
character*4 residName(MAX_RESID)	残基名
integer residID(MAX_ATOM)	残基 ID
real coordinate(3,MAX_ATOM)	原子座標
logical dihFlag	修正フラグ

出力：

なし

機能：

残基分類がアミノ酸、核酸、リガンドの残基について、残基内の水素に対し、他の残基との原子の距離が 1.0Å以内の場合、メッセージを出力する。オプション(-hyd)指定の場合には原子座標を変更する。

(9) subroutine Check_LinearTorsion(
chainNum, topResidID, lastResidID,
topAtomID, lastAtomID, residGroup,
atomName, residID, coordinate,
dihFlag)

引数：

integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
integer residGroup(MAX_RESID)	n 番目残基の残基分類
character*4 atomName(MAX_ATOM)	原子名
integer residID(MAX_ATOM)	残基 ID
real coordinate(3,MAX_ATOM)	原子座標
logical dihFlag	修正フラグ

出力：

なし

機能：

残基分類がアミノ酸の残基について、主鎖原子（N、CA、C）のいずれかで構成される 3 点(p1, p2, p3)について以下の処理を行う。

・p2→p1、p2→p3 が同一方向または完全な逆方向の場合メッセージを出力する。

・p2 と p1　または、p2 と p3 が同座標の場合メッセージを出力する

・p1 と p3 が同座標の場合警告メッセージを出力する。（dih 指定時のみ）

また、オプション(-dih)指定の場合には原子座標を補正する。

(10) subroutine Check_Residue(

chainNum, topResidID, lastResidID, residNum,
topAtomID, lastAtomID, atomNum,
residGroup, atomName, residName,
residID, chainName, coordinate,
mass, charge, orgssNum, orgssAtom,
modifyFlag, mutateFlag, rotateFlag, complexFlag)

引数：

integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
integer residGroup(MAX_RESID)	n 番目残基の残基分類
character*4 atomName(MAX_ATOM)	原子名
character*4 residName(MAX_RESID)	残基名
integer residID(MAX_ATOM)	残基 ID
real coordinate(3,MAX_ATOM)	原子座標
character*2chainName(MAX_ATOM)	チェイン名
real mass(MAX_ATOM)	原子質量
real charge(MAX_ATOM)	原子電荷
integer orgssNum	SS 結合数
integer orgssAtom(2,MAX_RESID)	SS 結合の対象原子
logical modifyFlag	欠損原子の付加と余剰原子の削除実行
logical rotateFlag	残基末端の回転の実行
logical muateFlag	水素結合向け残基種変更の実行
logical complexFlag	金属配位向け残基種変更の実行

出力：

なし

機能：

以下の処理を行う。

(1)トポロジーモデルの読み込み

(2)残基の欠損原子追加と余剰原子削除

(3)リガンド、金属の検出とドナー・アクセプター解析

(4)アミノ酸のドナー・アクセプター解析

(5)HIS/HISE, ASP/ASN, CLU/GLN の変更による水素結合解析と変更

(6)ASH, CYC, GLJ, SER, THR, TYR の末端回転による水素結合解析と変更

(7)金属配位による CYS の CYM 化と HIS/HISE 置換の解析と変更

３．３．４．io.f

(1) subroutine Input_PDB(
chainNum,
topResidID,
lastResidID,
residNum,
topAtomID,
lastAtomID,
atomNum,
atomName,
residName,
residID,
chainName,
coordinate,
altLoc,
hetFlag,
pdbFile)

引数：

integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer residNum	残基数
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
integer atomNum	原子数
character*4 atomName(MAX_ATOM)	原子名
charactrer*4 residName(MAX_ATOM)	残基名
integer residID(MAX_ATOM)	残基 ID
character*2 chainName(MAX_ATOM)	チェイン名
real coordinate(3,MAX_ATOM)	原子座標
character altLoc(MAX_ATOM)	Alternate location indicator の値
logical hetFlag	HETATM 読込みフラグ
chatacter(NAME_SIZE) pdbFile	入力ファイル名

戻り値

なし

機能：

入力 PDB ファイルをオープンし、ATOM 行を読み込む。-hetFlag が.TRUE.の場合、HETATM 行も読込み、内部構造に設定する。

チェイン構造については、本サブルーチンでは TER 行/END 行によってのみチェインを区切る。(チェイン ID などによる区切りは別サブルーチンで行う)

1 つめの ATOM 行のチェイン ID が空白の場合、下記メッセージを出し、内部で、A からチェイン ID を割り振る。

「WARNING: chain id is empty. chain id is assigned automatically.」

また、TER/END 行が現れるたびに新しいチェイン ID をインクリメントしていく。

(2) subroutine Output_PDB(

chainNum, topResidID, lastResidID,
topAtomID, lastAtomID, residGroup,
atomName, residName, residID,
chainName, coordinate,
ssNum, ssResidID,
cysNum, cysResidID,
hetFlag, disableHetFlag, keepTerFlag, remarkFlag,
inPdbFile, outPdbFile)

引数：

integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
integer residGroup(MAX_RESID)	n 番目残基の残基分類
character*4 atomName(MAX_ATOM)	原子名
charactrer*4 residName(MAX_ATOM)	残基名
integer residID(MAX_ATOM)	残基 ID
character*2 chainName(MAX_ATOM)	チェイン名
real coordinate(3,MAX_ATOM)	原子座標
integer ssNum	SS 結合数
integer ssResidID(2,MAX_RESID)	SS 結合の残基 ID
integer cymNum	CYS-(CYM)数
integer cymResidID(2,MAX_RESID)	CYS-(CYM)の残基 ID
logical disableHetFlag	-disableHet 指定
logical remarkFlag	-remark 指定
chatacter(NAME_SIZE) inPdbFile	入力ファイル名
chatacter(NAME_SIZE) outPdbFile	出力ファイル名

戻り値

なし

機能：

出力ファイルをオープンし、SSBOND 行、ATOM 行、HETATM 行、TER 行をファイルに出力する。

・ -disableHet が.TRUE.のときは HETATM 行は出力しない。

- -remark が.TRUE.のときは、入力 PDB の REMARK 行を読込み、そのまま出力 PDB に出力する。
- 入力 PDB ファイルに記述されていたチェイン ID は使用せず、新たに割り振ったチェイン ID を出力する。
- 入力 PDB ファイルに記述されていた残基 ID は使用せず、チェイン毎に 1 から新たに割り振った残基 ID を出力する。
- ATOM 行は、原子の tempFactor まで(66 カラム目まで)情報を出力する。
- 残基分類が、水グループ、Na イオングループ、Cl イオングループの残基は、チェイン毎には出力せず、それぞれのグループでまとめて出力し、それぞれのグループの出力が終了したら TER を出力する。

３．３．４．analysis.f

(1) subroutine Set_ResidGroup(
chainNum,
topResidID,
lastResidID,
topAtomID,
residName,
residGroup,
hetFlag)

引数

integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
charactrer*4 residName(MAX_ATOM)	残基名
integer residGroup(MAX_RESID)	n 番目残基の残基分類
logical hetFlag	-het 指定

戻り値

なし

機能

残基名や、前後の残基の種類から、残基分類を residGroup に設定する。
ただし、hetFlag が.FALSE.の場合は、全てをアミノ酸残基とみなす。
分類分けの詳細は、「３．１．入力ファイル読込み」を参照のこと。

(2) subroutine Update_ChainStructureData(
chainNum,
topResidID,
lastResidID,
topAtomID,
lastAtomID,
residGroup,
atomName,
residName,
chainName,
hetFlag)

引数

integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
integer residGroup(MAX_RESID)	n 番目残基の残基分類
charactrer*4 atomName(MAX_ATOM)	原子名
charactrer*4 residName(MAX_ATOM)	残基名
charactrer*4 atomName(MAX_ATOM)	チェイン名
logical hetFlag	-het 指定

戻り値

なし

機能

以下に従い、チェインの構造を更新(チェインを分割)します。

- N 残基と N+1 残基で、チェイン ID が異なる
- 非核酸残基 N と核酸残基 N+1 が存在する場合、N 残基と N+1 残基間でチェインを分割する
- 非水グループ残基 N と水グループ残基 N+1 が存在する場合、N 残基と N+1 残基間でチェインを分割する
- 非 Na イオングループ残基 N と Na イオングループ残基 N+1 が存在する場合、N 残基と N+1 残基間でチェインを分割する
- 非 Cl イオングループ残基 N と Cl イオングループ残基 N+1 が存在する場合、N 残基と N+1 残基間でチェインを分割する

- アミノ酸と分類された ACE 残基 N が存在する場合、N 残基と N-1 残基間でチェインを分割する
- アミノ酸と分類された NME/NHE/NH2 残基 N が存在する場合、N 残基と N+1 残基間でチェインを分割する
- アミノ酸と分類された残基 N が、OXT 原子を保持しているとき、、N 残基と N+1 残基間でチェインを分割する

(3) subroutine Analyze_RigandTerminal(
chainNum,
topResidID,
lastResidID,
topAtomID,
lastAtomID,
residGroup,
residName,
residID,
coordinate,
hetFlag)

引数

integer chainNum	チェイン数
integer topResidD(MAX_CHAIN)	n 番目チェインの先頭残基 ID
integer lastResidID(MAX_CHAIN)	n 番目チェインの最終残基 ID
integer topAtomID(MAX_RESID)	n 番目残基の先頭原子 ID
integer lastAtomID(MAX_RESID)	n 番目残基の最終原子 ID
integer residGroup(MAX_RESID)	n 番目残基の残基分類
charactrer*4 residName(MAX_ATOM)	残基名
integer residID(MAX_ATOM)	残基 ID
real coordinate(3,MAX_ATOM)	原子座標
logical hetFlag	-het 指定

戻り値

なし

機能

リガンドの残基について、1 分子毎にチェインを区切る。1 分子は以下の判定で行う。

・リガンド残基(N)とリガンド残基(N+1)で残基名が異なる場合、N 残基と N-1 残基でチェインを区切る
・リガンド残基(N)が存在し、非リガンドの N-1 残基との距離が 3.0Åより離れている場合、N 残基と N-1 残基でチェインを区切る
・リガンド残基(N)が存在し、非リガンドの N+1 残基との距離が 3.0Åより離れている場合、N 残基と N+1 残基でチェインを区切る
・連続したリガンド残基、N、N+1、N+2、・・・N+M が存在し、以下の場合、チェインを区切る。
残基(N)の全重原子について、N+1～N+M 残基の全重原子との距離が 3.0Åより離れているとき、その残基と他の残基との間でチェインを区切る。

３．３．５．residUtil.f90

(1) subroutine Read_TopologyModel(model, filename)

引数

type(ResidModel_t),dimension(200),intent(out)::model　残基モデル

character(*),intent(in)::filename　ファイル名

戻り値

なし

機能：

myPresto のトポロジーファイルを読み込み、残基を構成する原子のモデル情報を設定する。

(2) Search_Atom(searchName, pos, atomName, dir, atomNum) result (id)

引数

```
character*4,intent(in)::searchName   ! search target
integer,intent(in)::pos              ! start position
character*4,intent(in),dimension(MAX_ATOM)::atomName
integer,intent(in)::dir              ! search direction (sign)
integer,intent(in)::atomNum          ! number of atoms
```

戻り値

integer::id

機能：

searchName で指定した原子名の原子を開始位置から dir（正または負）の方向に向かって検索する。

(3) subroutine Get_ResidueModel(model, residName, nterm, cterm, nth)

引数

type(ResidModel_t),dimension(200)::model　残基モデル

character*4::residName　検索残基名

logical::nterm　検索対象は N 末

logical::cterm　検索対象は C 末

integer::nth　モデルの添字

戻り値

なし

機能：

残基名に該当し、N 末/C 末の条件に適合するモデルの配列添字を nth②設定する。

(4) Set_AcceptDonar(numatom,co,nbond,numbond,mat,
num_don_l,list_don,list_do2,list_do3,list_do4,
num_acc_l,list_acc,n_type_lhd,n_type_lha,
h_vecl,use_atom,speed)

引数

```
integer*4,intent(in)::numatom                                    ! number of atoms
real*4,intent(in),dimension(3,MAX_LIGAND)::co                    ! coordinate of atoms
integer*4,intent(inout),dimension(2,MAX_LIGAND)::nbond   ! bond atom ID
integer*4,intent(in)::numbond                                    ! number of bonds
character*2,intent(in),dimension(MAX_LIGAND)::mat        ! atomic identifier

integer*4,intent(inout)::num_don_l                               ! ligand donar number
integer*4,intent(out),dimension(MAX_LIGAND)::list_don ! ligand donar ID
integer*4,intent(out),dimension(MAX_LIGAND)::list_do2 ! ligand donar ID
integer*4,intent(out),dimension(MAX_LIGAND)::list_do3 ! ligand donar ID
integer*4,intent(out),dimension(MAX_LIGAND)::list_do4 ! ligand donar ID

integer*4,intent(inout)::num_acc_l                               ! ligand acceptor num.
integer*4,intent(out),dimension(MAX_LIGAND)::list_acc ! ligand acceptor ID
integer*4,intent(out),dimension(MAX_LIGAND)::n_type_lha ! acceptor type
integer*4,intent(out),dimension(MAX_LIGAND)::n_type_lhd ! ligand donar type
real*4,intent(out),dimension(3,MAX_LIGAND)::h_vecl       ! atom distance vector
logical,intent(out),dimension(MAX_LIGAND)::use_atom      ! use atom flag
integer*4,intent(in)::speed                                      ! dock speed
```

戻り値

なし

機能

化合物の結合と原子種を解析し、の水素ドナーとアクセプターのリストを返す。

(5) Is_BridgeOfGraph(numatom,numbond,nbond,nc1,nc2,one_graph)

引数

```
integer*4,intent(in)::numatom                                    ! number of atoms
integer*4,intent(in)::numbond                                    ! number of bonds
integer*4,intent(in),dimension(2,MAX_LIGAND)::nbond   ! bond pair
integer*4,intent(in)::nc1                                        ! cut point
integer*4,intent(in)::nc2                                        ! cut point
logical,intent(out)::one_graph                                   ! results
```

戻り値

なし

機能

指定した 2 つの原子間結合を切断した場合に、グラフ的に分割されるかを判定する。
false の場合は他に結合があるためこの 2 点は環構造に含まれることを示す。

(6) Count_HydrogenBond(

comment, id,
donarNum, donarList,
accNum, accList,
num_don_l, list_don,
num_acc_l, list_acc,
donCount, accCount,
atomName, residName, resID,
ligandName, ligResName, ligResID,
coordinate, ligandCord, maxHbond,
chainNum, topResidID, lastResidID,
topAtomID, bestResid)

引数

character(*),intent(in)::comment　表示用注釈
integer,intent(in)::id　残基 ID
integer,intent(in)::donarNum　! number of residue donars
integer,intent(in)::accNum　! number of residue acceptors
integer,intent(in),dimension(MAX_LIGAND)::donarList　! donar ID list
integer,intent(in),dimension(MAX_LIGAND)::accList　! acceptor ID list
integer,intent(in)::num_don_l　! number of ligand donars
integer,intent(in)::num_acc_l　! number of ligand acceptors
integer,intent(in),dimension(MAX_LIGAND)::list_don　! donar ID list
integer,intent(in),dimension(MAX_LIGAND)::list_acc　! acceptor ID list
integer,intent(out),dimension(MAX_LIGAND)::donCount　! donar ID list
integer,intent(out),dimension(MAX_LIGAND)::accCount　! acceptor ID list
character*4,intent(in),dimension(MAX_ATOM)::atomName ! atom name
character*4,intent(in),dimension(MAX_ATOM)::residName ! residue name
integer*4,intent(in),dimension(MAX_ATOM)::resID ! residue ID
real*4,intent(in),dimension(3,MAX_LIGAND)::ligandCord　化合物の座標
character*2,intent(in),dimension(MAX_LIGAND)::ligandName　化合物原子名
character*4,intent(in),dimension(MAX_LIGAND)::ligResName　化合物の残基名
integer,intent(in),dimension(MAX_LIGAND)::ligResID　化合物の残基 ID
real*4,intent(in),dimension(3,MAX_ATOM)::coordinate　蛋白原子座標

integer,intent(in)::chainNum　　チェイン数
integer,intent(in),dimension(MAX_CHAIN)::topResidID　　先頭残基 ID
integer,intent(in),dimension(MAX_CHAIN)::lastResidID　　最終残基 ID
integer,intent(in),dimension(MAX_RESID)::topAtomID　　先頭原子 ID
character*4,intent(out),dimension(MAX_RESID)::bestResid　　置換候補残基列

戻り値

なし

機能

蛋白と化合物のドナー・アクセプター間の水素結合をカウントする。
以前の水素結合数よりも多い場合には、現在の残基の並びを置換構造残基列に設定する。

(7) Get_SideChainDonarAcceptor(
residName, atomName, co, group,
chainNum, topResidID, lastResidID,
topAtomID, lastAtomID,
numdonar_p, numaccep_p,
listp_don, listp_acc)

引数

character*4,intent(in),dimension(MAX_ATOM)::residName　! residue name
character*4,intent(in),dimension(MAX_ATOM)::atomName　! first atom name
real*4,intent(in),dimension(3,MAX_ATOM)::co　! atom coordinate
integer,intent(in),dimension(MAX_RESID)::group　構造種別(残基、水、イオン)
integer,intent(in)::chainNum　チェイン数
integer,intent(in),dimension(MAX_CHAIN)::topResidID　先頭残基 ID
integer,intent(in),dimension(MAX_CHAIN)::lastResidID　最終残基 ID
integer,intent(in),dimension(MAX_RESID)::topAtomID　先頭原子 ID
integer,intent(in),dimension(MAX_RESID)::lastAtomID　最終原子 ID
integer*4,intent(inout)::numdonar_p　! donar number
integer*4,intent(inout)::numaccep_p　! acceptor number
integer*4,intent(out),dimension(MAX_LIGAND)::listp_acc ! acceptor atom ID
integer*4,intent(out),dimension(MAX_LIGAND)::listp_don ! acceptor atom ID

戻り値

なし

機能

アミノ酸のドナー・アクセプターを解析し、リストを返す。

(8)Try_RotateTerminal(

id,
donarNum, donarList,
accNum, accList,
num_don_l, list_don,
num_acc_l, list_acc,
donCount, accCount,
atomName, residName, resID,
ligandName, ligResName, ligResID,
coordinate, ligandCord, maxHbond,
chainNum, topResidID, lastResidID,
topAtomID, lastAtomID, bestResid, rotate,
setPhase)

引数

integer,intent(in)::id ! residue ID
integer,intent(in)::donarNum ! number of residue donars
integer,intent(in)::accNum ! number of residue acceptors
integer,intent(in),dimension(MAX_LIGAND)::donarList ! donar ID list
integer,intent(in),dimension(MAX_LIGAND)::accList ! acceptor ID list
integer,intent(in)::num_don_l ! number of ligand donars
integer,intent(in)::num_acc_l ! number of ligand acceptors
integer,intent(in),dimension(MAX_LIGAND)::list_don ! donar ID list
integer,intent(in),dimension(MAX_LIGAND)::list_acc ! acceptor ID list
integer,intent(out),dimension(MAX_LIGAND)::donCount ! donar ID list
integer,intent(out),dimension(MAX_LIGAND)::accCount ! acceptor ID list
character*4,intent(in),dimension(MAX_ATOM)::atomName ! atom name
character*4,intent(in),dimension(MAX_ATOM)::residName ! residue name
integer*4,intent(in),dimension(MAX_ATOM)::resID ! residue ID
real*4,intent(in),dimension(3,MAX_LIGAND)::ligandCord 化合物の座標
character*2,intent(in),dimension(MAX_LIGAND)::ligandName 化合物原子名
character*4,intent(in),dimension(MAX_LIGAND)::ligResName 化合物の残基名
integer,intent(in),dimension(MAX_LIGAND)::ligResID 化合物の残基 ID
real*4,intent(in),dimension(3,MAX_ATOM)::coordinate 蛋白原子座標
integer,intent(in)::chainNum チェイン数
integer,intent(in),dimension(MAX_CHAIN)::topResidID 先頭残基 ID
integer,intent(in),dimension(MAX_CHAIN)::lastResidID 最終残基 ID
integer,intent(in),dimension(MAX_RESID)::topAtomID 先頭原子 ID
integer,intent(in),dimension(MAX_RESID)::lastAtomID 最終原子 ID

character*4,intent(out),dimension(MAX_RESID)::bestResid 置換候補残基列
real*4,intent(out)::rotate 回転角
integer,intent(out)::maxHbond 水素結合の最大数
real*4,intent(in)::setPhase 指定回転角

戻り値

なし

機能

対象残基の側鎖が回転可能な場合は回転を試行し、最も水素結合の多い角度を rotate に設定する。

指定回転角に 0.0 以外が指定された場合は、この角度で側鎖を回転した座標を設定する。

(9) Get_NearMetalResidue(

chainNum,
topResidID, lastResidID,
topAtomID, lastAtomID,
atomName, residName,
coordinate,
metalCord,
cysResidID, cysDist, cysNum,
hisResidID, hisNum,
aspCount, gluCount, cymCount,
minCys, minHis, minAsp, minGlu, minCym)

引数

integer,intent(in)::chainNum　　チェイン数
integer,intent(in),dimension(MAX_CHAIN)::topResidID　先頭残基 ID
integer,intent(in),dimension(MAX_CHAIN)::lastResidID　最終残基 ID
integer,intent(in),dimension(MAX_RESID)::topAtomID　先頭原子 ID
integer,intent(in),dimension(MAX_RESID)::lastAtomID　最終原子 ID
character*4,intent(in),dimension(MAX_ATOM)::atomName ! atom name
character*4,intent(in),dimension(MAX_ATOM)::residName ! residue name
real*4,dimension(3,MAX_ATOM)::coordinate　　蛋白質の原子座標
real*4,dimension(3)::metalCord　　金属の座標
real*4,dimension(MAX_RESID)::cysDist　　! metal - CYS:HG length
integer,dimension(MAX_RESID)::cysResidID! CYS residue ID
integer::cysNum　　! CYS count (near metal)
integer,dimension(MAX_RESID)::hisResidID! HIS/HISE residue ID
integer::hisNum　　! HIS count (near metal)
integer::aspCount　　! ASP count (near metal)
integer::gluCount　　! GLU count (near metal)
integer::cymCount　　! CYM count (near metal)
real*4 minCys, minHis, minAsp, minGlu, minCym

戻り値

なし

機能

金属に近い(5.5Å以内)の HIS、HISE, HSD, GLU, ASP の金属配位する可能性のある原子の個数を返す

CYS の場合は残基 ID と距離、HIS/HISE/HISD の場合は残基 ID と個数を返す。

(10) Set_EditScope(
topChainID, lastChainID,
topResidID, lastResidID, apply)

引数

integer,intent(in)::topChainID ! absolute top chain ID for edit
integer,intent(in)::lastChainID ! absolute last chain ID for edit
integer,intent(in)::topResidID ! absolute top resid ID for edit
integer,intent(in)::lastResidID ! absolute last resid ID for edit
logical,intent(in)::apply ! apply edit region

戻り値

なし

機能：

加工対象のチェインと残基の範囲を設定する。

(11) Is_EditTarget(currChain, currResid, topResidID) result (inScope)

引数

integer,intent(in)::currChain 当該チェイン ID
integer,intent(in)::currResid 当該残基 ID
integer,intent(in)::topResidID 当該チェインの先頭残基 ID

戻り値

logical::inScope

機能

当該チェイン、残基が加工範囲かどうかを判定する

―以上―