

LibMyPresto 詳細設計書

2018 年 2 月 5 日

Copyright (C) 2006-2018 Next Generation Natural Product Chemistry (N<sup>2</sup>PC)

---

## — 目次 —

1. はじめに.....	1
2. ライブラリの呼び出しについて.....	2
2-1. ライブラリ呼び出し関数、サブルーチンについて.....	2
2-1-1. c 言語ソースから呼び出す場合.....	2
2-1-2. f77 ソースから呼び出す場合.....	5
2-1-3. f90 ソースから呼び出す場合.....	6
3. ライブラリの機能について.....	7
3-1. 処理の流れ.....	7
3-2. pdb ファイル読み込み.....	7
3-3. cif ファイル読み込み.....	11
3-4. 分子情報の pdb、cif の相互変換対応一覧.....	17
3-5. SSBOND 情報、LINK 情報の対応一覧.....	18
3-5-1. “SSBOND” 対応の “disulf”.....	18
3-5-2. “LINK” 対応の “covale”、“metalc”.....	19
3-6. COMPND 情報の対応一覧.....	20
3-7. MODRES 情報の対応一覧.....	21
3-8. バイオロジカルユニット情報の対応一覧.....	22
3-9. フォーマット変換処理.....	24
3-9-1. 分子情報の変換(pdb から cif への変換時).....	24
3-9-2. 分子情報の変換(cif から pdb への変換時).....	25
3-9-3. ジスルフィド結合情報の変換(pdb から cif への変換時).....	26
3-9-4. ジスルフィド結合情報の変換(cif から pdb への変換時).....	26
3-9-5. 共有結合、イオン結合情報の変換(pdb から cif への変換時).....	27
3-9-6. 共有結合、イオン結合情報の変換(cif から pdb への変換時).....	27
3-9-7. エントリ内の分子情報(COMPND)の変換(pdb から cif への変換時).....	28
3-9-8. エントリ内の分子情報(COMPND)の変換(cif から pdb への変換時).....	28
3-9-9. 残基の修飾に関する情報(MODRES)の変換(pdb から cif への変換時).....	28
3-9-10. 残基の修飾に関する情報(MODRES)の変換(cif から pdb への変換時).....	29
3-9-11. バイオロジカルユニットに関する情報の変換(cif から pdb への変換時) ..	30
3-10. pdb ファイル出力.....	31
3-10-1. 分子情報の出力(cif から pdb への変換時).....	31
3-11-2. ジスルフィド結合情報の出力(cif から pdb への変換時).....	32
3-12-3. 共有結合情報、イオン結合情報の出力(cif から pdb への変換時).....	33
3-13-4. エントリ内の分子情報(COMPND)の出力.....	35
3-14-5. 残基の修飾に関する情報(MODRES)の出力.....	35
3-15-6. バイオロジカルユニットに関する情報の変換(cif から pdb への変換時) ..	35
3-11. “cif” ファイル出力.....	36
3-11-1. 分子情報の出力(pdb から cif への変換時).....	36
3-11-2. ジスルフィド結合、共有結合、イオン結合情報の出力(pdb から cif への変換時) .....	38
3-11-3. エントリ内の分子情報(COMPND)の出力.....	41
3-11-4. エントリ内の分子情報(COMPND)の変換(cif から pdb への変換時).....	41
3-11-5. 残基の修飾に関する情報(MODRES)の変換(pdb から cif への変換時).....	41
4. ライブラリ関数等説明.....	43
4-1. 構造体変数.....	43
4-2. 関数について.....	47
5. プログラムサンプルについて.....	53

5-1. c 言語サンプルプログラム .....	53
5-2. f77 言語サンプルプログラム .....	55
5-3. f90 言語サンプルプログラム .....	58
6. 制限事項について.....	64
6-1. cif ファイル読み込み範囲について .....	64
6-2. Fortran から呼び出す場合のメモリ使用量について .....	64
6-3. pdb ファイルの読み込み範囲について .....	64
6-4. PDB 出力時の問題について .....	65

## 1. はじめに

本設計書では pdbx/mmCIF、pdb ファイルコンバートライブラリ (LibMyPresto) について示します。

LibMyPresto は、c 言語、f77、f90 で作成されたソースに対してコンパイル時にリンクする事により、これらのプログラムに pdbx/mmCIF(以下 cif と示す)、pdb の入出力機能を提供する事ができます。

2 章以降、以下の内容を示します。

- 2 章. ライブラリの呼び出しについて
- 3 章. ライブラリの機能について
- 4 章. ライブラリ関数等説明
- 5 章. プログラムサンプルについて
- 6 章. 制限事項について

## 2. ライブラリの呼び出しについて

本章では、c 言語、f77、f90 プログラムソースからライブラリを呼び出す方法を示します。

### 2-1. ライブラリ呼び出し関数、サブルーチンについて

c 言語、f77、f90 でそれぞれ本ライブラリを呼び出す関数、サブルーチンについて示します。

#### 2-1-1. c 言語ソースから呼び出す場合

c 言語から本ライブラリの機能呼び出す為には、以下の準備が必要となります。

- ①include 文の設定
- ②関数プロトタイプの宣言
- ③変数の宣言
- ④関数呼び出し

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "ToolDefine.h"           // ①ヘッダーファイルのインクルード
#include "ToolmyPresto.h"       // ①

void read_cif_c(char *in_file, SysPtr sys);           // ②プロトタイプの宣言
void write_pdb_c(char *out_file, SysPtr sys);       // ②
RemoveStruct(SysPtr sys);                           // ②
void *Malloc(int size);                             // ②

int main(int argc, char *argv[])
{
    char in_file[FILEBUFSIZE+1];                     // ③変数の宣言
    char out_file[FILEBUFSIZE+1];                   // ③
    SysPtr sys;                                       // ③
    char style[100];                                  // ③

    if((sys = (SysPtr)Malloc(sizeof(Sys))) == 0)     // ④関数呼び出し
    {
        printf(" ERROR> memory allocate\n");
        exit(1);
    }

    if(argc != 4 && argc != 3) {
        printf(" ERROR> Input Err , InputFile OutFile \n");
        printf("    ccif2pdb <in_file(cif)> <out_file(pdb)>\n");
        printf("        or\n");
        printf("    ccif2pdb <in_file(cif)> <out_file(pdb)> <residue_style>\n");
    }
}
```

```

    printf("    residue_style [presto|stdpdb|moe|schrodinger]¥n");
    printf("    presto: myPresto pdb style(ARG+, GLU-, CYSS, etc)¥n");
    printf("    stdpdb: standard pdb style¥n");
    printf("    moe   : MOE pdb style(HID, HIE, HIP)¥n");
    printf("    schrodinger: SCHRODINGER pdb style(HID, HIE, HIP, CYX, etc)¥n");
    exit(1);
}

if(strlen(argv[1]) > FILEBUFSIZE) {
    printf(" ERROR> Inputname is long¥n");
    exit(1);
}
if(strlen(argv[2]) > FILEBUFSIZE) {
    printf(" ERROR> Outname is long¥n");
    exit(1);
}

strcpy(in_file, argv[1]);
strcpy(out_file, argv[2]);

sys->intRType = READCIF;           // mmCIF ファイルを読み込むことを示す。
sys->intWType = WRITEPDB;         // PDB ファイルを出力することを示す。

sys->style = NOCHANGE;           // 残基名の出力スタイルを定義する。

// stdpdb を入力した場合は、標準の pdb 形式で出力する。
// presto を入力した場合は、myPresto の pdb 形式で出力する。ARG+, GLU-, HIS+, CYSS など
// moe を入力した場合は、MOE の pdb 形式で出力する。HID, HIE, HIP など
// schrodinger を入力した場合は、SCHRODINGER の形式で出力する。HID, HIE, HIP, CYX など
if(argc == 4) {
    if(strcmp(style, "stdpdb") == 0) sys->style = STANDARD;
    else if(strcmp(style, "presto") == 0) sys->style = PRESTO;
    else if(strcmp(style, "moe") == 0) sys->style = MOE;
    else if(strcmp(style, "schrodinger") == 0) sys->style = SCHRODINGER;
}

/* read file */
read_cif_c(in_file, sys);           // ④関数呼び出し

/* write file */
write_pdb_c(out_file, sys);       // ④

/* memory free */
RemoveStruct(sys);               // ④

return 0;

```

図. c 言語からのライブラリ呼び出し例

## ①include 文の設定

本ライブラリを使用するには、ToolmyPresto.h、ToolDefine.h の 2 つのインクルードファイルが必要となります。c 言語ソースディレクトリにこれら 2 つのファイルをコピーして

使用します。また、c 言語ソースの include 文で、これら 2 つのファイルをインクルードする必要があります。

## ②プロトタイプの宣言

使用するライブラリの関数のプロトタイプ宣言をします。以下の関数についてプロトタイプ宣言します。

プロトタイプ宣言 : `void read_cif_c(char *in_file, SysPtr sys);`

機能 : cif 入力ファイルの読み込みを行います。(拡張子は任意のものに対応)

引数 : `char *in_file`      入力ファイル名  
      `SysPtr sys`          分子系の情報、分子情報、原子情報に関する構造体変数

プロトタイプ宣言 : `void write_pdb_c(char *out_file, SysPtr sys);`

機能 : 指定したファイル名の cif ファイルを出力します。(拡張子は任意のものに対応)

引数 : `char *out_file`    出力ファイル名  
      `SysPtr sys`          分子系の情報、分子情報、原子情報に関する構造体変数

プロトタイプ宣言 : `void RemoveStruct(SysPtr sys);`

機能 : メモリ確保した `sys` 構造体変数等のメモリの解放を行います。

引数 : `SysPtr sys`          分子系の情報、分子情報、原子情報に関する構造体変数

プロトタイプ宣言 : `void *Malloc(int size);`

機能 : 変数のメモリ確保を行います。

引数 : `int size`            確保するメモリサイズ

## ③変数の宣言

本ライブラリを使用するには、入力ファイル名、出力ファイル名、分子系に関する構造体変数ポインタと宣言する必要があります。また、残基名の出力書式を特定の書式に変換する場合には、残基名スタイルに関する変数を準備する必要があります。

```
char in_file[FILEBUFSIZE+1];      // 入力ファイル名
char out_file[FILEBUFSIZE+1];     // 出力ファイル名
SysPtr sys;                        // 分子系に関する構造体変数ポインタ
char style[100];                   // 残基名の出力スタイル
```

## ④関数呼び出し

②で宣言した関数を使用する事ができます。

#### ⑤その他の関数について

本ライブラリでは、上記で説明した関数以外に以下の関数を宣言、呼び出しを行うことができます。

プロトタイプ宣言 : `void read_pdb_c(char *in_file, SysPtr sys);`

機能 : pdb 入力ファイルの読み込みを行います。(任意の拡張子に対応)

引数 : `char *in_file`      入力ファイル名  
       `SysPtr sys`            分子系の情報、分子情報、原子情報に関する構造体変数

プロトタイプ宣言 : `void write_cif_c(char *out_file, SysPtr sys);`

機能 : cif ファイルを出力します。(任意の拡張子に対応)

引数 : `char *out_file`    出力ファイル名  
       `SysPtr sys`            分子系の情報、分子情報、原子情報に関する構造体変数

#### 2-1-2. f77 ソースから呼び出す場合

f77 言語から本ライブラリの機能呼び出す為には、以下の準備が必要となります。

- ①パラメータ変数の設定とパラメータ変数の型宣言
- ②COMMON 文の設定と各変数の型宣言
- ③EXTERNAL 文の設定
- ④関数呼び出し

#### ①パラメータ変数の設定とパラメータ変数の型宣言

系全体の原子数、最大モデル数、最大ジスルフィド結合数、最大リンク情報数等を設定します。

#### ②COMMON 文の設定と各変数の型宣言

原子情報(F77ATM)、高分子鎖(F77ENP)、修正アミノ酸/核酸情報(F77MOD)、バイオリジカルユニット情報(F77AGEN、F77OLST)、モデル情報(F77MOL)、系全体の情報(F77SYS)、LINK 情報(F77LNK)、ジスルフィド結合(F77SSBN)に関する COMMON 文を設定します。

#### ③EXTERNAL 文の設定

ライブラリ内の関数を宣言します。

```
EXTERNAL read_cif_f77, write_pdb_f77
```

```
EXTERNAL read_pdb_f77, write_cif_f77
```

など。



## ④関数呼び出し

③で宣言した関数を呼び出すことができます。

## 2-1-3. f90 ソースから呼び出す場合

f90 言語から本ライブラリの機能呼び出す為には、以下の準備が必要となります。なお、①、②については、f90MolStruct.f90 を準備していますので、これをそのまま使用することができます。

- ①パラメータの設定
- ②構造体の設定
- ③use 文の設定
- ④external 文の宣言
- ⑤構造体変数の定義

## ①パラメータの設定

最大モデル数、最大原子数、最大ジスルフィド結合数、最大リンク情報数等を設定します。f90MolStruct.f90 に記載されています。本ファイルをそのまま使用する事ができます。

## ②構造体の設定

分子系に関する構造体変数等を設定します。f90MolStruct.f90 に記載されています。本ファイルをそのまま使用する事ができます。

## ③use 文の設定

①、②を使用する為に、use 文を使用します。

```
use parameters
```

```
use cmolstruct
```

## ④external 文の宣言

ライブラリ内の関数を宣言します。

```
external read_cif, write_pdb
```

```
external read_pdb, write_cif
```

など。

## ⑤構造体変数の設定

②で定義した構造体変数を使用します。

```
type(fsys) :: sys
```

### 3. ライブラリの機能について

#### 3-1. 処理の流れ

本ライブラリを使用する際には、以下の処理の流れに従って、ファイルの生成を行います。各処理は以下の節で示します。

- (1) pdb ファイルまたは cif ファイル読み込み
- (2) フォーマット変換処理
- (3) cif ファイルまたは pdb ファイル出力

#### 3-2. pdb ファイル読み込み

以下に pdb ファイル例を示します。ファイルフォーマットについては web サイト (<http://www.wwpdb.org/documentation/format33/v3.3.html>) 等をご参照下さい。

HEADER	SPASMOLYTIC PROTEIN		05-JAN-94		1PSP					
(略)										
COMPND	MOL_ID: 1;									
COMPND	2 MOLECULE: PANCREATIC SPASMOLYTIC POLYPEPTIDE;									
COMPND	3 CHAIN: A, B;									
COMPND	4 ENGINEERED: YES									
SSBOND	1	CYS A	6	CYS A	104		1555	1555	2.37	
SSBOND	2	CYS A	8	CYS A	35		1555	1555	2.03	
SSBOND	3	CYS A	19	CYS A	34		1555	1555	2.01	
SSBOND	4	CYS A	29	CYS A	46		1555	1555	2.01	
SSBOND	5	CYS A	58	CYS A	84		1555	1555	2.02	
SSBOND	6	CYS A	68	CYS A	83		1555	1555	2.00	
SSBOND	7	CYS A	78	CYS A	95		1555	1555	1.99	
SSBOND	8	CYS B	6	CYS B	104		1555	1555	2.17	
LINK		C	PCA A	1		N	LYS A	2	1555 1555 1.33	
LINK		C	PCA B	1		N	LYS B	2	1555 1555 1.31	
(略)										
HETATM	1	N	PCA A	1	25.686	44.892	82.141	1.00	38.54	N
HETATM	2	CA	PCA A	1	26.200	43.828	83.075	1.00	34.31	C
HETATM	3	CB	PCA A	1	25.263	42.626	83.061	1.00	36.68	C
HETATM	4	CG	PCA A	1	23.858	43.318	82.693	1.00	40.13	C
HETATM	5	CD	PCA A	1	24.263	44.619	81.973	1.00	40.74	C
HETATM	6	OE	PCA A	1	23.468	45.568	81.936	1.00	42.13	O
HETATM	7	C	PCA A	1	27.561	43.450	82.561	1.00	31.62	C
HETATM	8	O	PCA A	1	27.930	43.823	81.431	1.00	30.50	O
ATOM	9	N	LYS A	2	28.323	42.728	83.371	1.00	28.24	N
ATOM	10	CA	LYS A	2	29.751	42.577	83.065	1.00	26.86	C
ATOM	11	C	LYS A	2	30.019	42.226	81.596	1.00	26.15	C
(略)										
MODRES	1PSP	PCA A	1	GLU	PYROGLUTAMIC ACID					
MODRES	1PSP	PCA B	1	GLU	PYROGLUTAMIC ACID					

※上記の例は、本ツールで読み込む項目のみ示しており、実際には他の項目も存在します。

本プログラムの読み込み項目

(1) “HEADER” キーワード情報

pdb id を読み込む。

(2) “ATOM”, “HETATM” キーワード情報

下記の項目を読み込む。

No.	位置	内容	データ例
1	1 - 6	ATOM/HETATM	HETATM
2	7 - 11	原子の通し番号	1
3	13 - 16	原子名	N
4	17	Alternate location 識別子	
5	18 - 20	残基名	PCA
6	22	鎖名	A
7	23 - 26	残基番号	1
8	27	残基の挿入コード	
9	31 - 38	原子の X 座標の値	25.686
10	39 - 46	原子の Y 座標の値	44.892
11	47 - 54	原子の Z 座標の値	82.141
12	55 - 60	占有率	1.00
13	61 - 66	温度因子	38.54
14	77 - 78	元素記号	N
15	79 - 80	原子の電荷	

(3) “SSBOND” キーワード情報

下記の項目を読み込む。

No.	位置	内容	データ例
1	1 - 6	SSBOND	SSBOND
2	8 - 10	通し番号	1
3	12 - 14	残基名 (CYS)	CYS
4	16	鎖名	A
5	18 - 21	残基番号	6
6	22	挿入コード	
7	26 - 28	残基名 (CYS)	CYS

8	30	鎖名	A
9	32 - 35	残基番号	104
10	36	挿入コード	
11	60 - 65	残基 1 の対称操作	1555
12	67 - 72	残基 2 の対称操作	1555
13	74 - 78	ジスルフィド結合距離	2.37

## (4) “LINK” キーワード情報

下記の項目を読み込む。

No.	位置	内容	データ例
1	1 - 6	LINK	LINK
2	13 - 16	原子 1 の原子名	C
3	17	原子 1 の Alternate Location 識別子	
4	18 - 20	原子 1 の残基名	PCA
5	22	原子 1 の鎖名	A
6	23 - 26	原子 1 の残基番号	1
7	27	原子 1 の挿入コード	
8	43 - 46	原子 2 の原子名	N
9	47	原子 2 の Alternate Location 識別子	
10	48 - 50	原子 2 の残基名	LYS
11	52	原子 2 の鎖名	A
12	53 - 56	原子 2 の残基番号	2
13	57	原子 2 の挿入コード	
14	60 - 65	原子 1 の対称操作	1555
15	67 - 72	原子 2 の対称操作	1555
16	74 - 78	結合距離	1.33

## (5) “MODEL” キーワード情報

下記の項目を読み込む。

No.	位置	内容	データ例
1	1 - 6	MODEL	MODEL
2	11 - 14	番号	1

## (6) “TER” キーワード情報

下記の項目を読み込む。

No.	位置	内容	データ例
1	1 - 6	TER	TER

## (7) “ENDMDL” キーワード情報

下記の項目を読み込む。

No.	位置	内容	データ例
1	1 - 6	ENDMDL	ENDMDL

## (8) “MODRES” キーワード情報

下記の項目を読み込む。

No.	位置	内容	データ例
1	1 - 6	MODRES	MODRES
2	8 - 11	Pdb id	1PSP
3	13 - 15	残基名	PCA
4	17	鎖名	A
5	19 - 22	残基番号	1
6	23	挿入コード	
7	25 - 27	標準の残基名	GLU
8	30 - 70	コメント	PYROGLUTAMIC ACID

## (9) “COMPND” キーワード情報

下記の項目を読み込む。本キーワードのない pdb ファイルに異なる鎖 ID の分子を持つ場合には、それらは、別の分子 ID を付与するものとする。

No.	位置	内容	データ例
1	1 - 6	COMPND	COMPND
2	8 - 10	継続を示す。	
3	11 - 80	分子の詳細情報 (MOL_ID、CHAIN 情報のみ読み込みを行う)	
		MOL_ID : 各要素の番号	
		CHAIN : 鎖名	

## 3-3. cif ファイル読み込み

本プログラムでは、cif ファイル読み込みを行う。以下にファイル例を示す。

mmCIF 形式ファイルフォーマットについては web サイト (<http://mmCIF.wwpdb.org/>) を参照の事。

```

data_1PSP
#
_entry.id 1PSP
#
(略)
#
loop_
_struct_conn.id
_struct_conn.conn_type_id
_struct_conn.pdbx_PDB_id
_struct_conn.ptnr1_label_asym_id
_struct_conn.ptnr1_label_comp_id
_struct_conn.ptnr1_label_seq_id
_struct_conn.ptnr1_label_atom_id
_struct_conn.pdbx_ptnr1_label_alt_id
_struct_conn.pdbx_ptnr1_PDB_ins_code
_struct_conn.pdbx_ptnr1_standard_comp_id
_struct_conn.ptnr1_symmetry
_struct_conn.ptnr2_label_asym_id
_struct_conn.ptnr2_label_comp_id
_struct_conn.ptnr2_label_seq_id
_struct_conn.ptnr2_label_atom_id
_struct_conn.pdbx_ptnr2_label_alt_id
_struct_conn.pdbx_ptnr2_PDB_ins_code
_struct_conn.ptnr1_auth_asym_id
_struct_conn.ptnr1_auth_comp_id
_struct_conn.ptnr1_auth_seq_id
_struct_conn.ptnr2_auth_asym_id
_struct_conn.ptnr2_auth_comp_id
_struct_conn.ptnr2_auth_seq_id
_struct_conn.ptnr2_symmetry
_struct_conn.pdbx_ptnr3_label_atom_id
_struct_conn.pdbx_ptnr3_label_seq_id
_struct_conn.pdbx_ptnr3_label_comp_id
_struct_conn.pdbx_ptnr3_label_asym_id
_struct_conn.pdbx_ptnr3_label_alt_id
_struct_conn.pdbx_ptnr3_PDB_ins_code
_struct_conn.details
_struct_conn.pdbx_dist_value
_struct_conn.pdbx_value_order
disulf1 disulf ? A CYS 6 SG ? ? ? 1_555 A CYS 104 SG ? ? A CYS 6 A CYS 104 1_555 ? ? ? ? ? ? ? 2.366 ?
disulf2 disulf ? A CYS 8 SG ? ? ? 1_555 A CYS 35 SG ? ? A CYS 8 A CYS 35 1_555 ? ? ? ? ? ? ? 2.028 ?
disulf3 disulf ? A CYS 19 SG ? ? ? 1_555 A CYS 34 SG ? ? A CYS 19 A CYS 34 1_555 ? ? ? ? ? ? ? 2.007 ?
disulf4 disulf ? A CYS 29 SG ? ? ? 1_555 A CYS 46 SG ? ? A CYS 29 A CYS 46 1_555 ? ? ? ? ? ? ? 2.006 ?
disulf5 disulf ? A CYS 58 SG ? ? ? 1_555 A CYS 84 SG ? ? A CYS 58 A CYS 84 1_555 ? ? ? ? ? ? ? 2.018 ?
disulf6 disulf ? A CYS 68 SG ? ? ? 1_555 A CYS 83 SG ? ? A CYS 68 A CYS 83 1_555 ? ? ? ? ? ? ? 2.001 ?

```

```

disulf7 disulf ? A CYS 78 SG ? ? ? 1_555 A CYS 95 SG ? ? A CYS 78 A CYS 95 1_555 ? ? ? ? ? ? 1.992 ?
disulf8 disulf ? B CYS 6 SG ? ? ? 1_555 B CYS 104 SG ? ? B CYS 6 B CYS 104 1_555 ? ? ? ? ? ? 2.171 ?
disulf9 disulf ? B CYS 8 SG ? ? ? 1_555 B CYS 35 SG ? ? B CYS 8 B CYS 35 1_555 ? ? ? ? ? ? 2.150 ?
disulf10 disulf ? B CYS 19 SG ? ? ? 1_555 B CYS 34 SG ? ? B CYS 19 B CYS 34 1_555 ? ? ? ? ? ? 2.083 ?
disulf11 disulf ? B CYS 29 SG ? ? ? 1_555 B CYS 46 SG ? ? B CYS 29 B CYS 46 1_555 ? ? ? ? ? ? 2.129 ?
disulf12 disulf ? B CYS 58 SG ? ? ? 1_555 B CYS 84 SG ? ? B CYS 58 B CYS 84 1_555 ? ? ? ? ? ? 2.193 ?
disulf13 disulf ? B CYS 68 SG ? ? ? 1_555 B CYS 83 SG ? ? B CYS 68 B CYS 83 1_555 ? ? ? ? ? ? 2.163 ?
disulf14 disulf ? B CYS 78 SG ? ? ? 1_555 B CYS 95 SG ? ? B CYS 78 B CYS 95 1_555 ? ? ? ? ? ? 2.236 ?
covale1 covale ? A PCA 1 C ? ? ? 1_555 A LYS 2 N ? ? A PCA 1 A LYS 2 1_555 ? ? ? ? ? ? 1.326 ?
covale2 covale ? B PCA 1 C ? ? ? 1_555 B LYS 2 N ? ? B PCA 1 B LYS 2 1_555 ? ? ? ? ? ? 1.313 ?

```

#

(略)

loop\_

\_atom\_site.group\_PDB

\_atom\_site.id

\_atom\_site.type\_symbol

\_atom\_site.label\_atom\_id

\_atom\_site.label\_alt\_id

\_atom\_site.label\_comp\_id

\_atom\_site.label\_asym\_id

\_atom\_site.label\_entity\_id

\_atom\_site.label\_seq\_id

\_atom\_site.pdbx\_PDB\_ins\_code

\_atom\_site.Cartn\_x

\_atom\_site.Cartn\_y

\_atom\_site.Cartn\_z

\_atom\_site.occupancy

\_atom\_site.B\_iso\_or\_equiv

\_atom\_site.Cartn\_x\_esd

\_atom\_site.Cartn\_y\_esd

\_atom\_site.Cartn\_z\_esd

\_atom\_site.occupancy\_esd

\_atom\_site.B\_iso\_or\_equiv\_esd

\_atom\_site.pdbx\_formal\_charge

\_atom\_site.auth\_seq\_id

\_atom\_site.auth\_comp\_id

\_atom\_site.auth\_asym\_id

\_atom\_site.auth\_atom\_id

\_atom\_site.pdbx\_PDB\_model\_num

```

ATOM 1 N N . PCA A 1 1 ? 25.686 44.892 82.141 1.00 38.54 ? ? ? ? ? ? 1 PCA A N 1
ATOM 2 C CA . PCA A 1 1 ? 26.200 43.828 83.075 1.00 34.31 ? ? ? ? ? ? 1 PCA A CA 1
ATOM 3 C CB . PCA A 1 1 ? 25.263 42.626 83.061 1.00 36.68 ? ? ? ? ? ? 1 PCA A CB 1
ATOM 4 C CG . PCA A 1 1 ? 23.858 43.318 82.693 1.00 40.13 ? ? ? ? ? ? 1 PCA A CG 1
ATOM 5 C CD . PCA A 1 1 ? 24.263 44.619 81.973 1.00 40.74 ? ? ? ? ? ? 1 PCA A CD 1
ATOM 6 O OE . PCA A 1 1 ? 23.468 45.568 81.936 1.00 42.13 ? ? ? ? ? ? 1 PCA A OE 1
ATOM 7 C C . PCA A 1 1 ? 27.561 43.450 82.561 1.00 31.62 ? ? ? ? ? ? 1 PCA A C 1
ATOM 8 O O . PCA A 1 1 ? 27.930 43.823 81.431 1.00 30.50 ? ? ? ? ? ? 1 PCA A O 1
ATOM 9 N N . LYS A 1 2 ? 28.323 42.728 83.371 1.00 28.24 ? ? ? ? ? ? 2 LYS A N 1
ATOM 10 C CA . LYS A 1 2 ? 29.751 42.577 83.065 1.00 26.86 ? ? ? ? ? ? 2 LYS A CA 1
ATOM 11 C C . LYS A 1 2 ? 30.019 42.226 81.596 1.00 26.15 ? ? ? ? ? ? 2 LYS A C 1
ATOM 12 O O . LYS A 1 2 ? 29.572 41.173 81.090 1.00 31.03 ? ? ? ? ? ? 2 LYS A O 1
ATOM 13 C CB . LYS A 1 2 ? 30.437 41.584 84.005 1.00 21.01 ? ? ? ? ? ? 2 LYS A CB 1

```

(略)

```

loop_
_pdbx_struct_assembly_gen.assembly_id
_pdbx_struct_assembly_gen.oper_expression
_pdbx_struct_assembly_gen.asym_id_list
1 1 A,C
2 1 B,D
#
_pdbx_struct_oper_list.id                1
_pdbx_struct_oper_list.type              'identity operation'
_pdbx_struct_oper_list.name              1_555
_pdbx_struct_oper_list.symmetry_operation x, y, z
_pdbx_struct_oper_list.matrix[1][1]     1.0000000000
_pdbx_struct_oper_list.matrix[1][2]     0.0000000000
_pdbx_struct_oper_list.matrix[1][3]     0.0000000000
_pdbx_struct_oper_list.vector[1]        0.0000000000
_pdbx_struct_oper_list.matrix[2][1]     0.0000000000
_pdbx_struct_oper_list.matrix[2][2]     1.0000000000
_pdbx_struct_oper_list.matrix[2][3]     0.0000000000
_pdbx_struct_oper_list.vector[2]        0.0000000000
_pdbx_struct_oper_list.matrix[3][1]     0.0000000000
_pdbx_struct_oper_list.matrix[3][2]     0.0000000000
_pdbx_struct_oper_list.matrix[3][3]     1.0000000000
_pdbx_struct_oper_list.vector[3]        0.0000000000
#
(略)

```

※上記の例は、本ツールで読み込む項目のみ示しており、実際には他の項目も存在する。

本プログラムの読み込み項目

(1) “\_entry.id” キーワード情報

分子名を読み込む。

(2) “\_atom\_site.” キーワード情報

“ATOM”、“HETATM” の項目名を読み込む。

(3) “\_struct\_conn.” キーワード情報

“disulf”、“covale”、“metalc” の項目名を読み込む。その他 hydrog(水素結合)情報などは今回の読み込み対象外とする。

(4) “ATOM”、“HETATM” キーワード情報

スペース区切りで下記の項目を読み込む。

No.	項目名	データ例
1	group_PDB	ATOM
2	id	1



3	type_symbol	N
4	label_atom_id	N
5	label_alt_id	.
6	label_comp_id	PCA
7	label_asym_id	A
8	label_entity_id	1
9	label_seq_id	1
10	pdbx_PDB_ins_code	?
11	Cartn_x	25.686
12	Cartn_y	44.892
13	Cartn_z	82.141
14	occupancy	1.00
15	B_iso_or_equiv	38.54
16	Cartn_x_esd	?
17	Cartn_y_esd	?
18	Cartn_z_esd	?
19	occupancy_esd	?
20	B_iso_or_equiv_esd	?
21	pdbx_formal_charge	?
22	auth_seq_id	1
23	auth_comp_id	PCA
24	auth_asym_id	A
25	auth_atom_id	N
26	pdbx_PDB_model_num	1

## (5) “disulf”、“covale”、“metalc” キーワード情報

No.	項目名	データ例1	データ例2	データ例3
1	id	disulf1	covale1	metalc1
2	conn_type_id	disulf	covale	metalc
3	pdbx_PDB_id	?	?	?
4	ptnr1_label_asym_id	A	A	B
5	ptnr1_label_comp_id	CYS	PCA	ZN
6	ptnr1_label_seq_id	6	1	.
7	ptnr1_label_atom_id	SG	C	ZN
8	pdbx_ptnr1_label_alt_id	?	?	?

9	pdbx_ptnr1_PDB_ins_code	?	?	?
10	pdbx_ptnr1_standard_comp_id	?	?	?
11	ptnr1_symmetry	1_555	1_555	1_555
12	ptnr2_label_asym_id	A	A	E
13	ptnr2_label_comp_id	CYS	LYS	HOH
14	ptnr2_label_seq_id	104	2	.
15	ptnr2_label_atom_id	SG	N	0
16	pdbx_ptnr2_label_alt_id	?	?	?
17	pdbx_ptnr2_PDB_ins_code	?	?	?
18	ptnr1_auth_asym_id	A	A	A
19	ptnr1_auth_comp_id	CYS	PCA	ZN
20	ptnr1_auth_seq_id	6	1	301
21	ptnr2_auth_asym_id	A	A	A
22	ptnr2_auth_comp_id	CYS	LYS	HOH
23	ptnr2_auth_seq_id	104	2	409
24	ptnr2_symmetry	1_555	1_555	1_555
25	pdbx_ptnr3_label_atom_id	?	?	?
26	pdbx_ptnr3_label_seq_id	?	?	?
27	pdbx_ptnr3_label_comp_id	?	?	?
28	pdbx_ptnr3_label_asym_id	?	?	?
29	pdbx_ptnr3_label_alt_id	?	?	?
30	pdbx_ptnr3_PDB_ins_code	?	?	?
31	details	?	?	?
32	pdbx_dist_value	2.366	2.366	1.897
33	pdbx_value_order	?	?	?

## (6) “\_pdbx\_struct\_assembly\_gen.” キーワード情報

以下の各項目の項目名、値を読み込む。

No.	項目名	データ例
1	assembly_id	1
2	oper_expression	1
3	asym_id_list	A, C

## (7) “\_pdbx\_struct\_oper\_list.” キーワード情報

以下の各項目の項目名、値を読み込む。

No.	項目名	データ例
1	id	1
2	type	'identity operation'
3	symmetry_operation	1_555
4	matrix[1][1]	x, y, z
5	matrix[1][2]	1.0000000000
6	matrix[1][3]	0.0000000000
7	vector[1]	0.0000000000
8	matrix[2][1]	0.0000000000
9	matrix[2][2]	1.0000000000
10	matrix[2][3]	0.0000000000
11	vector[2]	0.0000000000
12	matrix[3][1]	0.0000000000
13	matrix[3][2]	0.0000000000
14	matrix[3][3]	1.0000000000
15	vector[3]	0.0000000000

## 3-4. 分子情報の pdb、cif の相互変換対応一覧

“ATOM”、“HETATM” 対応の一覧を下記に示す。

No.	内容 (pdb)	内容 (cif)	データ例 (pdb)	データ例 (cif)
1	ATOM/HETATM	group_PDB	HETATM	ATOM
2	原子の通し番号	id	1	1
3	原子名	auth_atom_id	N	N
4	Alternate location 識別子	label_alt_id		.
5	残基名	auth_comp_id	PCA	PCA
6	鎖名	auth_asym_id	A	A
7	残基番号	auth_seq_id	1	1
8	残基の挿入コード	pdxb_PDB_ins_code		?
9	原子の X 座標の値	Cartn_x	25.686	25.686
10	原子の Y 座標の値	Cartn_y	44.892	44.892
11	原子の Z 座標の値	Cartn_z	82.141	82.141
12	占有率	occupancy	1.00	1.00
13	温度因子	B_iso_or_equiv	38.54	38.54
14	元素記号	type_symbol	N	N
15	原子の電荷	pdxb_formal_charge		?
16	—	pdxb_PDB_model_num	—	1
17	—	label_atom_id	—	N
18	—	label_comp_id	—	PCA
19	—	label_asym_id	—	A
20	—	label_entity_id	—	1
21	—	label_seq_id	—	1
22	—	Cartn_x_esd	—	?
23	—	Cartn_y_esd	—	?
24	—	Cartn_z_esd	—	?
25	—	occupancy_esd	—	?
26	—	B_iso_or_equiv_esd	—	?

## 3-5. SSBOND 情報、LINK 情報の対応一覧

“SSBOND” 対応の “disulf”、“LINK” 対応の “covale”、“metalc” 情報の対応一覧を下記に示す。

## 3-5-1. “SSBOND” 対応の “disulf”

No.	内容(pdb)	内容(cif)	データ例(pdb)	データ例(cif)
1	SSBOND	conn_type_id	SSBOND	disulf
2	通し番号	id	1	disulf1
3	残基名 1(CYS)	ptnr1_auth_comp_id	CYS	CYS
4	鎖名 1	ptnr1_auth_asym_id	A	A
5	残基番号 1	ptnr1_auth_seq_id	6	6
6	挿入コード 1	pdbx_ptnr1_PDB_ins_code		?
7	残基名 2(CYS)	ptnr2_auth_comp_id	CYS	CYS
8	鎖名 2	ptnr2_auth_asym_id	A	A
9	残基番号 2	ptnr2_auth_seq_id	104	104
10	挿入コード 2	pdbx_ptnr2_PDB_ins_code		?
11	残基 1 の対称操作	ptnr1_symmetry	1555	1_555
12	残基 2 の対称操作	ptnr2_symmetry	1555	1_555
13	ジスルフィド結合距離	pdbx_dist_value	2.37	2.366
14	—	pdbx_PDB_id	—	?
15	—	ptnr1_label_asym_id	—	A
16	—	ptnr1_label_comp_id	—	CYS
17	—	ptnr1_label_seq_id	—	6
18	—	ptnr1_label_atom_id	—	SG
19	—	pdbx_ptnr1_label_alt_id	—	?
20	—	pdbx_ptnr1_standard_comp_id	—	?
21	—	ptnr2_label_asym_id	—	A
22	—	ptnr2_label_comp_id	—	CYS
23	—	ptnr2_label_seq_id	—	104
24	—	ptnr2_label_atom_id	—	SG
25	—	pdbx_ptnr2_label_alt_id	—	?

26	—	pdbx_ptnr3_label_atom_id	—	?
27	—	pdbx_ptnr3_label_seq_id	—	?
28	—	pdbx_ptnr3_label_comp_id	—	?
29	—	pdbx_ptnr3_label_asym_id	—	?
30	—	pdbx_ptnr3_label_alt_id	—	?
31	—	pdbx_ptnr3_PDB_ins_code	—	?
32	—	details	—	?
33	—	pdbx_value_order	—	?

## 3-5-2. “LINK” 対応の “covale”、“metalc”

No.	内容(pdb)	内容(cif)	データ例 (pdb)	データ例 (cif)
1	LINK	conn_type_id	LINK	covale
2	原子1の原子名	ptnr1_label_atom_id	C	C
3	原子1のAlternate Location 識別子	pdbx_ptnr1_label_alt_id		?
4	原子1の残基名	ptnr1_auth_comp_id	PCA	PCA
5	原子1の鎖名	ptnr1_auth_asym_id	A	A
6	原子1の残基番号	ptnr1_auth_seq_id	1	1
7	原子1の挿入コード	pdbx_ptnr1_PDB_ins_code		?
8	原子2の原子名	ptnr2_label_atom_id	N	N
9	原子2のAlternate Location 識別子	pdbx_ptnr2_label_alt_id		?
10	原子2の残基名	ptnr2_auth_comp_id	LYS	LYS
11	原子2の鎖名	ptnr2_auth_asym_id	A	A
12	原子2の残基番号	ptnr2_auth_seq_id	2	2
13	原子2の挿入コード	pdbx_ptnr2_PDB_ins_code		?
14	原子1の対称操作	ptnr1_symmetry	1555	1_555
15	原子2の対称操作	ptnr2_symmetry	1555	1_555
16	結合距離	pdbx_dist_value	1.33	1.326
17	—	id	—	covale1
18	—	pdbx_PDB_id	—	?
19	—	ptnr1_label_asym_id	—	A
20	—	ptnr1_label_comp_id	—	PCA
21	—	ptnr1_label_seq_id	—	1

22	—	pdbx_ptnr1_standard_comp_id	—	?
23	—	ptnr2_label_asym_id	—	A
24	—	ptnr2_label_comp_id	—	LYS
25	—	ptnr2_label_seq_id	—	2
26	—	pdbx_ptnr3_label_atom_id	—	?
27	—	pdbx_ptnr3_label_seq_id	—	?
28	—	pdbx_ptnr3_label_comp_id	—	?
29	—	pdbx_ptnr3_label_asym_id	—	?
30	—	pdbx_ptnr3_label_alt_id	—	?
31	—	pdbx_ptnr3_PDB_ins_code	—	?
32	—	details	—	?
33	—	pdbx_value_order	—	?

## 3-6. COMPND 情報の対応一覧

“COMPND” に対応する “\_entity\_poly” 情報の対応一覧を下記に示す。

No.	内容(pdb)	内容(cif)	データ例(pdb)	データ例(cif)
1	MOL_ID 項目	entity_id	1	1
2	—	type	—	'polypeptide(L)'
3	—	nstd_linkage	—	no
4	非標準のモノマーの有無	nstd_monomer	—	yes
5	—	pdbx_seq_one_letter_code	—	(PCA)KPAACR CSRQD
6	—	pdbx_seq_one_letter_code_c an	—	EKPAACRCSRQ D
7	CAHIN 項目	pdbx_strand_id	A, B	A, B

## 3-7. MODRES 情報の対応一覧

“MODRES” に対応する “\_pdbx\_struct\_mod\_residue” 情報の対応一覧を下記に示す。

No.	内容 (pdb)	内容 (cif)	データ例 (pdb)	データ例 (cif)
1	PDBID	-	1PSP	-
2	非標準残基の名称	label_comp_id	PCA	PCA
3	鎖 ID	label_asym_id	A	A
4	残基番号	label_seq_id	1	1
5	挿入コード	PDB_ins_code	-	(PCA) KPAACR CSRQD
6	標準残基名	parent_comp_id	GLU	EKPAACRCSRQ D
7	注釈	details	PYROGLUTAMI C ACID	A, B
8	-	auth_asym_id	-	A
9	-	auth_seq_id	-	1
10	-	auth_comp_id	-	PCA



## 3-8. バイオロジカルユニット情報の対応一覧

“REMARK 350”に対応する “\_pdbx\_struct\_assembly\_gen”、“\_pdbx\_struct\_oper\_list” 情報の対応一覧を下記に示す。

No.	内容 (pdb)	内容 (cif)	データ例 (pdb)	データ例 (cif)
1	BIOMOLECULE 項目	_pdbx_struct_assembly_gen. assembly_id	1	1
2	APPLY THE FOLLOWING TO CHAINS 項目	_pdbx_struct_assembly_gen. asym_id_list	A	A, C, D, G
3	ID 1 (BIOMT1 の 1 項 目目)	_pdbx_struct_oper_list.id		1
4	行列 1 (BIOMT1 の 2-4 項目目)	_pdbx_struct_oper_list.mat rix[1][1] _pdbx_struct_oper_list.mat rix[1][2] _pdbx_struct_oper_list.mat rix[1][3]	-1.000000 0.000000 0.000000	-1.00000000 00 0.00000000 0 0.00000000 0
5	ベクトル 1 (BIOMT1 の 5 カラム目)	_pdbx_struct_oper_list.vec tor[1]	109.10335	109.1033540 338
6	ID 2 (BIOMT2 の 1 項 目目)	_pdbx_struct_oper_list.id	0.000000	1
7	行列 2 (BIOMT2 の 2-4 項目目)	_pdbx_struct_oper_list.mat rix[2][1] _pdbx_struct_oper_list.mat rix[2][2] _pdbx_struct_oper_list.mat rix[2][3]	0.000000 1.000000 0.000000	0.00000000 0 1.00000000 0 0.00000000 0
8	ベクトル 2 (BIOMT2 の 5 カラム目)	_pdbx_struct_oper_list.vec tor[2]	0.000000	0.00000000 0
9	ID 3 (BIOMT3 の 1 項 目目)	_pdbx_struct_oper_list.id	0.000000	1
10	行列 3 (BIOMT3 の	_pdbx_struct_oper_list.mat	0.000000	0.00000000

	2-4 項目目)	rix[3][1] _pdbx_struct_oper_list.mat rix[3][2] _pdbx_struct_oper_list.mat rix[3][3]	0.000000 -1.000000	0 0.00000000 0 -1.00000000 00
11	ベクトル 3(BIOMT3 の 5 カラム目)	_pdbx_struct_oper_list.vec tor[3]	85.18392	85.18391565 60
12	-	_pdbx_struct_oper_list.sym metry_operation	-	'identity operation'
13	-	_pdbx_struct_assembly_gen. oper_expression	-	1,2

### 3-9. フォーマット変換処理

#### 3-9-1. 分子情報の変換(pdb から cif への変換時)

##### (1) ATOM/HETATM

pdb で読み込んだ内容通りとする。(cif では非標準アミノ酸、非標準核酸は ATOM 行で記載されて、pdb では HETATM で記載される。本ツールでは、その残基が標準か、非標準残基かの判定を行わない為、読み込んだ内容通り出力するものとする。

##### (2) label\_alt\_id

Alternate location 識別子を出力する。空白の場合、. を出力する。

##### (3) label\_asym\_id

pdb で読み込んだ内容通りとする。

##### (4) label\_entity\_id

全ての原子データについて 1 とする。entity\_id は cif で新たに定義されたものであり、pdb 読み込みにおいて、本情報を正しく作成する事ができない。

##### (5) label\_seq\_id

プログラムで設定して出力する。

1 から “label\_comp\_id” が同一の間、同じ番号を出力する。

##### (6) pdbx\_PDB\_ins\_code

残基の挿入コードを出力する。空白の場合、? を出力する。

##### (7) pdbx\_formal\_charge

原子の電荷を出力する。空白の場合、? を出力する。

3-9-2. 分子情報の変換(cif から pdb への変換時)

(1) ATOM/HETATM

cif で読み込んだ内容通りとする。(cif では非標準アミノ酸、非標準核酸について ATOM 行で記載されるが、pdb では HETATM 行で記載される。本ツールでは、その残基が標準か、非標準かの判定を行わない為、読み込んだ内容通り出力するものとする。

(2) 原子の通し番号

内部シーケンス番号を出力する。

(3) 原子名

auth\_atom\_id を出力する。条件により、出力位置に対応する。

(4) Alternate location 識別子

label\_alt\_id を出力する。 . の場合は空白とする。

(5) 残基の挿入コード

pdbx\_PDB\_ins\_code を出力する。 ? の場合、空白とする。

(6) 原子の電荷

pdbx\_formal\_charge を出力する。 ? の場合、空白とする。

3-9-3. ジスルフィド結合情報の変換 (pdb から cif への変換時)

(1) `pdbx_ptnr1_PDB_ins_code`

挿入コードを出力する。空白の場合、?を出力する。

(2) `pdbx_ptnr2_PDB_ins_code`

挿入コードを出力する。空白の場合、?を出力する。

(3) `ptnr1_symmetry`

残基 1 の対称操作を出力する。フォーマットを変換して出力する。

(4) `ptnr2_symmetry`

残基 2 の対称操作を出力する。フォーマットを変換して出力する。

3-9-4. ジスルフィド結合情報の変換 (cif から pdb への変換時)

(1) 挿入コード

`pdbx_ptnr1_PDB_ins_code` を出力する。?の場合は、空白を出力する。

(2) 原子 1 の挿入コード

`pdbx_ptnr2_PDB_ins_code` を出力する。?の場合は、空白を出力する。

(3) 残基 1 の対称操作

`ptnr1_symmetry` を出力する。フォーマットを変換して出力する。

(4) 残基 2 の対称操作

`ptnr2_symmetry` を出力する。フォーマットを変換して出力する。

3-9-5. 共有結合、イオン結合情報の変換(pdb から cif への変換時)

(1) `pdbx_ptnr1_label_alt_id`

原子 1 の Alternate Location 識別子を出力する。空白の場合は、?を出力する。

(2) `pdbx_ptnr1_PDB_ins_code`

原子 1 の挿入コードを出力する。空白の場合は、?を出力する。

(3) `pdbx_ptnr2_label_alt_id`

原子 2 の Alternate Location 識別子を出力する。空白の場合は、?を出力する。

(4) `pdbx_ptnr2_PDB_ins_code`

原子 2 の挿入コードを出力する。空白の場合は、?を出力する。

3-9-6. 共有結合、イオン結合情報の変換(cif から pdb への変換時)

(1) 原子 1 の Alternate Location 識別子

`pdbx_ptnr1_label_alt_id` を出力する。?の場合は、空白を出力する。

(2) 原子 1 の挿入コード

`pdbx_ptnr1_PDB_ins_code` を出力する。?の場合は、空白を出力する。

(3) 原子 2 の Alternate Location 識別子

`pdbx_ptnr2_label_alt_id` を出力する。?の場合は、空白を出力する。

(4) 原子 2 の挿入コード

`pdbx_ptnr2_PDB_ins_code` を出力する。?の場合は、空白を出力する。

3-9-7. エントリ内の分子情報 (COMPND) の変換 (pdb から cif への変換時)

(1) entity\_id

COMPND 項目の MOL\_ID 項目の値を出力する。

(2) pdbx\_starnd\_id

COMPND 項目の CHAIN 項目の値を出力する。

3-9-8. エントリ内の分子情報 (COMPND) の変換 (cif から pdb への変換時)

(1) COMPND MOL\_ID 項目

entity\_id 項目の値を出力する。

(2) COMPND CHAIN 項目

pdbx\_strand\_id 項目の値を出力する。

3-9-9. 残基の修飾に関する情報 (MODRES) の変換 (pdb から cif への変換時)

(1) label\_asym\_id

MODRES 行の 4 項目目を出力する。

(2) label\_seq\_id

MODRES 行の 5 項目目を出力する。

(3) label\_comp\_id

MODRES 行の 3 項目目を出力する。

(4) auth\_asym\_id

(1) と同じ。

(5) auth\_seq\_id

(2) と同じ。

(6) auth\_comp\_id

(3) と同じ

(7) PDB\_ins\_code

MODRES 行の 6 項目目を出力する。値がない場合は?を出力する。

(8) parent\_comp\_id

MODRES 行の 7 項目目を出力する。

(9) details

?を出力する。

3-9-10. 残基の修飾に関する情報 (MODRES) の変換 (cif から pdb への変換時)

(1) PDBID

\_entry.id の値を出力する。

(2) 非標準残基の名称

label\_comp\_id の値を出力する。

(3) 鎖 ID

label\_asym\_id の値を出力する。

(4) 残基番号

label\_seq\_id の値を出力する。

(5) 挿入コード

PDB\_ins\_code を出力する。値が?の場合には”“(空白文字)とする。

(6) 標準残基名

parent\_comp\_id の値を出力する。



3-9-11. バイオロジカルユニットに関する情報の変換(cif から pdb への変換時)

(1) REMARK 350 BIOMOLECULE 項目

\_pdbx\_struct\_assembly.id の値を出力する。

(2) REMARK 350 APPLY THE FOLLOWING TO CHAINS 項目

\_pdbx\_struct\_assembly\_gen.asym\_id\_list の値を出力する。

(3) REMARK 350 BIOMT1 項目

\_pdbx\_struct\_oper\_list.id、\_pdbx\_struct\_oper\_list.matrix、  
\_pdbx\_struct\_oper\_list.vector の各情報を出力する。

※バイオロジカルユニットの変換は cif から pdb への変換のみに対応しており、pdb から cif への逆変換には対応してません。

3-10. pdb ファイル出力

「3-3. pdb ファイル読み込み」で示した項目を出力する。

3-10-1. 分子情報の出力(cif から pdb への変換時)

(1) ATOM/HETATM

group\_PDB を基とし、ATOM/HETATM を出力する。

(2) 原子の通し番号

内部シーケンス番号を出力する。

(3) 原子名

auth\_atom\_id を出力する。

(4) Alternate location 識別子

label\_alt\_id を出力する。

(5) 残基名

auth\_comp\_id を出力する。

(6) 鎖名

auth\_asym\_id を出力する。

(7) 残基番号

auth\_seq\_id を出力する。

(8) 残基の挿入コード

pdbx\_PDB\_ins\_code を出力する。

(9) 原子の X 座標の値

Cartn\_x を出力する。

(10) 原子の Y 座標の値

Cartn\_y を出力する。

(11) 原子の Z 座標の値

Cartn\_z を出力する。

- (12) 占有率  
occupancy を出力する。
- (13) 温度因子  
B\_iso\_or\_equiv を出力する。
- (14) 元素記号  
type\_symbol を出力する。
- (15) 原子の電荷  
pdbx\_formal\_charge を出力する。

3-11-2. ジスルフィド結合情報の出力(cif から pdb への変換時)

- (1) SSBOND  
SSBOND を出力する。
- (2) 通し番号  
内部シーケンス番号を出力する。
- (3) 残基名(CYS)  
ptnr1\_auth\_comp\_id を出力する。
- (4) 鎖名  
ptnr1\_auth\_asym\_id を出力する。
- (5) 残基番号  
ptnr1\_auth\_seq\_id を出力する。
- (6) 挿入コード  
pdbx\_ptnr1\_PDB\_ins\_code を出力する。
- (7) 残基名(CYS)  
ptnr2\_auth\_comp\_id を出力する。

- (8) 鎖名  
ptnr2\_auth\_asym\_id を出力する。
- (9) 残基番号  
ptnr2\_auth\_seq\_id を出力する。
- (10) 挿入コード  
pdbx\_ptnr2\_PDB\_ins\_code を出力する。
- (11) 残基 1 の対称操作  
ptnr1\_symmetry を出力する。
- (12) 残基 2 の対称操作  
ptnr2\_symmetry を出力する。
- (13) ジスルフィド結合距離  
pdbx\_dist\_value を出力する。

3-12-3. 共有結合情報、イオン結合情報の出力 (cif から pdb への変換時)

- (1) LINK  
LINK を出力する。
- (2) 原子 1 の原子名  
ptnr1\_label\_atom\_id を出力する。
- (3) 原子 1 の Alternate Location 識別子  
pdbx\_ptnr1\_label\_alt\_id を出力する。
- (4) 原子 1 の残基名  
ptnr1\_auth\_comp\_id を出力する。
- (5) 原子 1 の鎖名  
ptnr1\_auth\_asym\_id を出力する。

- (6) 原子 1 の残基番号  
ptnr1\_auth\_seq\_id を出力する。
- (7) 原子 1 の挿入コード  
pdbx\_ptnr1\_PDB\_ins\_code を出力する。
- (8) 原子 2 の原子名  
ptnr2\_label\_atom\_id を出力する。
- (9) 原子 2 の Alternate Location 識別子  
pdbx\_ptnr2\_label\_alt\_id を出力する。
- (10) 原子 2 の残基名  
ptnr2\_auth\_comp\_id を出力する。
- (11) 原子 2 の鎖名  
ptnr2\_auth\_asym\_id を出力する。
- (12) 原子 2 の残基番号  
ptnr2\_auth\_seq\_id を出力する。
- (13) 原子 2 の挿入コード  
pdbx\_ptnr2\_PDB\_ins\_code を出力する。
- (14) 原子 1 の対称操作  
ptnr1\_symmetry を出力する。
- (15) 原子 2 の対称操作  
ptnr2\_symmetry を出力する。
- (16) 結合距離  
pdbx\_dist\_value を出力する。

3-13-4. エントリ内の分子情報 (COMPND) の出力

- (1) COMPND MOL\_ID 項目  
entity\_id 項目の値を出力する。
- (2) COMPND CHAIN 項目  
pdbx\_strand\_id 項目の値を出力する。

3-14-5. 残基の修飾に関する情報 (MODRES) の出力

- (1) PDBID  
\_entry.id の値を出力する。
- (2) 非標準残基の名称  
label\_comp\_id の値を出力する。
- (3) 鎖 ID  
label\_asym\_id の値を出力する。
- (4) 残基番号  
label\_seq\_id の値を出力する。
- (5) 挿入コード  
PDB\_ins\_code を出力する。値が?の場合には”“(空白文字)とする。
- (6) 標準残基名  
parent\_comp\_id の値を出力する。

3-15-6. バイオロジカルユニットに関する情報の変換 (cif から pdb への変換時)

- (1) REMARK 350 BIOMOLECULE 項目  
\_pdbx\_struct\_assembly.id の値を出力する。
- (2) REMARK 350 APPLY THE FOLLOWING TO CHAINS 項目  
\_pdbx\_struct\_assembly\_gen.asym\_id\_list の値を出力する。
- (3) REMARK 350 BIOMT1 項目  
\_pdbx\_struct\_oper\_list.id、\_pdbx\_struct\_oper\_list.matrix、  
\_pdbx\_struct\_oper\_list.vector の各情報を出力する。

3-11. “cif” ファイル出力

「2-4. “cif” ファイル読み込み」で示した項目を出力する。

3-11-1. 分子情報の出力(pdb から cif への変換時)

(1) group\_PDB

ATOM/HETATM を出力する。

(2) 原子の通し番号

内部シーケンス番号を出力する。

(3) type\_symbol

元素記号を出力する。

(4) label\_atom\_id

auth\_atom\_id を出力する。

(5) label\_alt\_id

Alternate location 識別子を出力する。空白の場合、. を出力する。

(6) label\_comp\_id

auth\_comp\_id を出力する。

(7) label\_asym\_id

auth\_asym\_id を出力する。

(8) label\_entity\_id

変換処理で設定した値を出力する。

(9) label\_seq\_id

変換処理で設定した値を出力する。

(10) pdbx\_PDB\_ins\_code

残基の挿入コードを出力する。空白の場合、? を出力する。

(11) Cartn\_x

原子の X 座標の値を出力する。

- (12) Cartn\_y  
原子の Y 座標の値を出力する。
- (13) Cartn\_z  
原子の Z 座標の値を出力する。
- (14) Occupancy  
占有率を出力する。
- (15) B\_iso\_or\_equiv  
温度因子を出力する。
- (16) Cartn\_x\_esd  
デフォルトで?を出力する。
- (17) Cartn\_y\_esd  
デフォルトで?を出力する。
- (18) Cartn\_z\_esd  
デフォルトで?を出力する。
- (19) occupancy\_esd  
デフォルトで?を出力する。
- (20) B\_iso\_or\_equiv\_esd  
デフォルトで?を出力する。
- (21) pdbx\_formal\_charge  
原子の電荷を出力する。空白の場合、?を出力する。
- (22) auth\_seq\_id  
残基番号を出力する。
- (23) auth\_comp\_id  
残基名を出力する。



(24) auth\_asym\_id

鎖名を出力する。

(25) auth\_atom\_id

原子名を出力する。

(26) pdbx\_PDB\_model\_num

“MODEL” の番号を出力する。

3-11-2. ジスルフィド結合、共有結合、イオン結合情報の出力(pdb から cif への変換時)

(1) id

“disulf”、“covale”、“metalc” に内部シーケンス番号を付加して出力する。

(2) conn\_type\_id

“disulf”、“covale”、“metalc” を出力する。

(3) pdbx\_PDB\_id

デフォルトで?を出力する。

(4) ptrn1\_label\_asym\_id

ptrn1\_auth\_asym\_id を出力する。

(5) ptrn1\_label\_comp\_id

ptrn1\_auth\_comp\_id を出力する。

(6) ptrn1\_label\_seq\_id

ptrn1\_auth\_seq\_id を出力する。

(7) ptrn1\_label\_atom\_id

“SSBOND” の場合は「SG」固定で、“LINK” の場合は「原子 1 の原子名」を出力する。

(8) pdbx\_ptrn1\_label\_alt\_id

“SSBOND” の場合は「?」、 “LINK” の場合は「原子 1 の Alternate Location 識別子」を出力する。空白の場合は、?を出力する。

- (9) `pdbx_ptnr1_PDB_ins_code`  
“SSBOND”の場合は「挿入コード」、 “LINK”の場合は「原子 1 の挿入コード」を出力する。空白の場合は、?を出力する。
- (10) `pdbx_ptnr1_standard_comp_id`  
デフォルトで?を出力する。
- (11) `ptnr1_symmetry`  
“SSBOND”の場合は「残基 1 の対称操作」、 “LINK”の場合は「原子 1 の対称操作」を出力する。フォーマットを変換して出力する。
- (12) `ptnr2_label_asym_id`  
`ptnr2_auth_asym_id`を出力する。
- (13) `ptnr2_label_comp_id`  
`ptnr2_auth_comp_id`を出力する。
- (14) `ptnr2_label_seq_id`  
`ptnr2_auth_seq_id`を出力する。
- (15) `ptnr2_label_atom_id`  
“SSBOND”の場合は「SG」固定、 “LINK”の場合は「原子 2 の原子名」を出力する。
- (16) `pdbx_ptnr2_label_alt_id`  
“SSBOND”の場合は「?」、 “LINK”の場合は「原子 2 の Alternate Location 識別子」を出力する。空白の場合は、?を出力する。
- (17) `pdbx_ptnr2_PDB_ins_code`  
“SSBOND”の場合は「挿入コード 2」、 “LINK”の場合は「原子 2 の挿入コード」を出力する。空白の場合は、?を出力する。
- (18) `ptnr1_auth_asym_id`  
“SSBOND”の場合は「鎖名 1」、 “LINK”の場合は「原子 1 の鎖名」を出力する。

- (19) ptnr1\_auth\_comp\_id  
“SSBOND”の場合は「残基名 1(CYS)」、 “LINK”の場合は「原子 1 の残基名」を出力する。
- (20) ptnr1\_auth\_seq\_id  
“SSBOND”の場合は「残基番号 1」、 “LINK”の場合は「原子 1 の残基番号」を出力する。
- (21) ptnr2\_auth\_asym\_id  
“SSBOND”の場合は「鎖名 2」、 “LINK”の場合は「原子 2 の鎖名」を出力する。
- (22) ptnr2\_auth\_comp\_id  
“SSBOND”の場合は「残基名 2(CYS)」、 “LINK”の場合は「原子 2 の残基名」を出力する。
- (23) ptnr2\_auth\_seq\_id  
“SSBOND”の場合は「残基番号 2」、 “LINK”の場合は「原子 2 の残基番号」を出力する。
- (24) ptnr2\_symmetry  
“SSBOND”の場合は「残基 2 の対称操作」、 “LINK”の場合は「原子 2 の対称操作」を出力する。フォーマットを変換して出力する。
- (25) pdbx\_ptnr3\_label\_atom\_id  
デフォルトで?を出力する。
- (26) pdbx\_ptnr3\_label\_seq\_id  
デフォルトで?を出力する。
- (27) pdbx\_ptnr3\_label\_comp\_id  
デフォルトで?を出力する。
- (28) pdbx\_ptnr3\_label\_asym\_id  
デフォルトで?を出力する。
- (29) pdbx\_ptnr3\_label\_alt\_id  
デフォルトで?を出力する。

(30) pdbx\_ptnr3\_PDB\_ins\_code

デフォルトで?を出力する。

(31) details

デフォルトで?を出力する。

(32) pdbx\_dist\_value

“SSBOND” の場合は「ジスルフィド結合距離」、 “LINK” の場合は「結合距離」を出力する。

(33) pdbx\_value\_order

デフォルトで?を出力する。

### 3-11-3. エントリ内の分子情報 (COMPND) の出力

(1) entity\_id

COMPND 項目の MOL\_ID 項目の値を出力する。

(2) pdbx\_strand\_id

COMPND 項目の CHAIN 項目の値を出力する。

### 3-11-4. エントリ内の分子情報 (COMPND) の変換 (cif から pdb への変換時)

(1) COMPND MOL\_ID 項目

entity\_id 項目の値を出力する。

(2) COMPND CHAIN 項目

pdbx\_strand\_id 項目の値を出力する。

### 3-11-5. 残基の修飾に関する情報 (MODRES) の変換 (pdb から cif への変換時)

(1) label\_asym\_id

MODRES 行の 4 項目目を出力する。

(2) label\_seq\_id

MODRES 行の 5 項目目を出力する。

- (3) label\_comp\_id  
MODRES 行の 3 項目目を出力する。
  
- (4) auth\_asym\_id  
(1) と同じ。
  
- (5) auth\_seq\_id  
(2) と同じ。
  
- (6) auth\_comp\_id  
(3) と同じ
  
- (7) PDB\_ins\_code  
MODRES 行の 6 項目目を出力する。値がない場合は?を出力する。
  
- (8) parent\_comp\_id  
MODRES 行の 7 項目目を出力する。
  
- (9) details  
?を出力する。

## 4. ライブラリ関数等説明

本章では、本ライブラリで使用される構造体、関数について示します。

## 4-1. 構造体変数

システム構造体変数を下記に示す。

```

typedef struct Sys // 分子系に関する構造体
{
    int intRType; // Read Type */
    int intWType; // Write Type */
    char nameHed[NAMELEN]; // pdb id 情報

    Cif itAtom;
    Cif itConn;
    Cif itEnPoly;
    Cif itChComp;

    MolPtr mol, molbegin, molend; // 分子モデルに関する構造体ポインタ

    int total_atom_num; // 系全体の原子数
    int model_num; // モデル数
    int ssbond_num; // 系の SS 結合数
    int link_num; // 系の LINK 数

    ConnPtr connB, connBbegin, connBend; // SS 結合に関する構造体ポインタ
    ConnPtr connL, connLbegin, connLend; // LINK に関する構造体ポインタ
    ModResPtr modres, modresbegin, modresend; // 修正 AA/NA に関する構造体ポインタ
    EnPolyPtr enPoly, enPolybegin, enPolyend; // 高分子鎖に関する構造体ポインタ
} Sys, *SysPtr;

typedef struct Cif // cif カテゴリ名、値に関する構造体
{
    int Cnt; // Count */
    char name[CIFITEM][CIFITEMLEN]; // Item name */
    int num[CIFITEM]; // Data number */
    int len[CIFITEM]; // Max Len */
    char type[CIFITEM][CIFTYPELEN]; // Write Type */
} Cif;

typedef struct Mol // 分子モデルに関する構造体
{ // pdb の MODEL の単位でデータを保存する。
    int atom_num;

    AtomPtr atom, atombegin, atomend; // 原子に関する構造体ポインタ

    struct Mol *before;
    struct Mol *next;
} Mol, *MolPtr;

```

```

typedef struct Atom
{
    /* for Pdb */
    char group_PDB[8];          /* 0 1-6 "ATOM" or "HETATM" */
                                /* same as _atom_site.group_PDB */

    int num;                   /* 1 serial number (1-) */
    int atom_num;              /* 2 7-11 Atom number */
    char atom_name[NAMELEN];   /* 3 13-16 Atom name */
    char altLoc[4];           /* 4 17 */
                                /* same as _atom_site.label_alt_id */
    char res_name[NAMELEN];    /* 5 18-20 Residue name */
    char chainID[8];          /* 6 22 */
    int res_num;              /* 7 23-26 Residue number */
    int res_num_serial;       /* Residue number to serialize (to use cif file) */
                                /* count up for same model or same chain, */
                                /* other model, chain, start from 1. */

    char iCode[4];            /* 8 27 */
    float x, y, z;            /* 9,10,11 31-38 39-46 47-54 Real(8.3) */
    float occupancy;          /* 12 55-60 Real(6.2) */
    float tempFactor;         /* 13 61-66 Real(6.2) */
    char element[4];          /* 14 77-78 */
                                /* same as _atom_site.type_symbol */
    char charge[4];           /* 15 79-80 */
    char cif_charge[4];

    int model_num;            /* 1 MODEL Number */

    int cif_num;              /* same _atom_site.id */

    /* for Cif */
    /* 0 _atom_site.group_PDB          use group_PDB
    * 1 _atom_site.id                  use cif_num
    * 2 _atom_site.type_symbol         use element
    * 3 _atom_site.label_atom_id      use cif_label_atom_id
    * 4 _atom_site.label_alt_id      create from altLoc
    * 5 _atom_site.label_comp_id      use cif_label_comp_id
    * 6 _atom_site.label_asym_id      use cif_label_asym_id
    * 7 _atom_site.label_entity_id    use cif_label_entity_id
    * 8 _atom_site.label_seq_id       use cif_label_seq_id
    * 9 _atom_site.pdbx_PDB_ins_code  create from iCode
    * 10 _atom_site.Cartn_x           use x
    * 11 _atom_site.Cartn_y           use y
    * 12 _atom_site.Cartn_z           use z
    * 13 _atom_site.occupancy         use occupancy
    * 14 _atom_site.B_iso_or_equiv    use tempFactor
    * 15 _atom_site.Cartn_x_esd       use cif_cartn_x_esd
    * 16 _atom_site.Cartn_y_esd       use cif_cartn_y_esd
    * 17 _atom_site.Cartn_z_esd       use cif_cartn_z_esd
    * 18 _atom_site.occupancy_esd     use cif_occupancy_esd
    * 19 _atom_site.B_iso_or_equiv_esd use cif_b_iso_or_equiv_esd
    * 20 _atom_site.pdbx_formal_charge create charge
    * 21 _atom_site.auth_seq_id       use res_num
    * 22 _atom_site.auth_comp_id      use res_name
    * 23 _atom_site.auth_asym_id      use chainID
    * 24 _atom_site.auth_atom_id      use atom_name
    * 25 _atom_site.pdbx_PDB_model_num use model_num
    */
}

```

```

char cif_label_atom_id[NAMELEN];
char cif_label_alt_id[4];
char cif_label_comp_id[NAMELEN];
char cif_label_asym_id[8];
char cif_label_entity_id[CIFDATALEN];
char cif_label_seq_id[CIFDATALEN];
char cif_pdbx_PDB_ins_code[4];
char cif_cartn_x_esd[CIFDATALEN];
char cif_cartn_y_esd[CIFDATALEN];
char cif_cartn_z_esd[CIFDATALEN];
char cif_occupancy_esd[CIFDATALEN];
char cif_b_iso_or_equiv_esd[CIFDATALEN];

struct Atom *before;
struct Atom *next;
} Atom, *AtomPtr;

typedef struct Conn
{
    int num;                /* 1 serial number (1-)   *Bond */
    int bond_num;          /*                        *Bond 2 8-10 */

    char atom_name1[NAMELEN]; /* 2 13-16 Atom name */
    char altLoc1[4];        /* 3 17 */
    char res_name1[NAMELEN]; /* 4 18-20 Residue name   *Bond 3 12-14 CYS */
    char chainID1[8];      /* 5 22                   *Bond 4 16 */
    int res_num1;         /* 6 23-26 Residue number *Bond 5 18-21 */
    char iCode1[4];       /* 7 27                   *Bond 6 22 */

    char atom_name2[NAMELEN]; /* 8 43-46 Atom name */
    char altLoc2[4];        /* 9 47 */
    char res_name2[NAMELEN]; /* 10 48-50 Residue name  *Bond 7 26-28 CYS */
    char chainID2[8];      /* 11 52                  *Bond 8 30 */
    int res_num2;         /* 12 53-56 Residue number *Bond 9 32-35 */
    char iCode2[4];       /* 13 57                  *Bond 10 36 */

    int symmetry1;        /* 14,15 60-65 67-72     *Bond 11,12 60-65 67-72 */
    int symmetry2;        /* 14,15 60-65 67-72     *Bond 11,12 60-65 67-72 */
    float length;        /* 16 74-78 Real(5.2)    *Bond 13 74-78 Real(5.2) */
    /* _struct_conn.pdbx_dist_value */

    char cif_ptnr1_symmetry[CIFDATALEN]; // 原子1の対称性情報
    char cif_ptnr2_symmetry[CIFDATALEN]; // 原子2の対称性情報

    char cif_id[CIFDATALEN]; //
    char cif_conn_type_id[CIFDATALEN];
    char cif_pdbx_PDB_id[CIFDATALEN];

    char cif_ptnr1_label_asym_id[CIFDATALEN]; /* chainID */
    char cif_ptnr1_label_comp_id[CIFDATALEN]; /* residue_name */
    char cif_ptnr1_label_seq_id[CIFDATALEN];
    char cif_pdbx_ptnr1_label_alt_id[CIFDATALEN];
    char cif_pdbx_ptnr1_PDB_ins_code[CIFDATALEN];
    char cif_pdbx_ptnr1_standard_comp_id[CIFDATALEN];

    char cif_ptnr2_label_asym_id[CIFDATALEN]; /* chainID */
    char cif_ptnr2_label_comp_id[CIFDATALEN]; /* residue_name */
    char cif_ptnr2_label_seq_id[CIFDATALEN];
    char cif_pdbx_ptnr2_label_alt_id[CIFDATALEN];
    char cif_pdbx_ptnr2_PDB_ins_code[CIFDATALEN];

```



```

    char cif_pdbx_ptrn3_label_atom_id[CIFDATALEN];
    char cif_pdbx_ptrn3_label_seq_id[CIFDATALEN];
    char cif_pdbx_ptrn3_label_comp_id[CIFDATALEN];
    char cif_pdbx_ptrn3_label_asym_id[CIFDATALEN];
    char cif_pdbx_ptrn3_label_alt_id[CIFDATALEN];
    char cif_pdbx_ptrn3_PDB_ins_code[CIFDATALEN];
    char cif_details[CIFDATALEN];
    char cif_pdbx_value_order[CIFDATALEN];

    struct Conn *before;
    struct Conn *next;
} Conn, *ConnPtr;

typedef struct EnPoly{                                // 高分子鎖(_entity_poly 情報)
    int num;

    char cif_entity_id[CIFDATALEN];                  // _entity_poly.entity_id
    char cif_type[256];                               // _entity_poly.type
    char cif_nstd_linkage[CIFDATALEN];               // _entity_poly.nstd_linkage
    char cif_nstd_monomer[CIFDATALEN];              // _entity_poly.nstd_monomer
    char cif_pdbx_seq_one_letter_code[CIFRESMAX];   // _entity_poly.pdbx_seq_one_letter_code
    char cif_pdbx_seq_one_letter_code_can[CIFRESMAX]; // _entity_poly.pdbx_seq_one_letter_code_can
    char cif_pdbx_strand_id[CIFRESMAX];             // _entity_poly.pdbx_strand_id

    int molID;
    char chainIDs[2048][8];                          /* chainIDs include same molID */
    int num_chainIDs;                                /* number if chainIDs include same molID */

    struct EnPoly *before;
    struct EnPoly *next;
} EnPoly, *EnPolyPtr;

typedef struct ModRes                                // 修正 AA/NA (_pdbx_struct_mod_residue)情報
{
    char pdbid[8];                                    // pdbid

    char id[CIFDATALEN];                             /* pdb id */
    char label_comp_id[CIFDATALEN];
    char label_asym_id[CIFDATALEN];
    char label_seq_id[CIFDATALEN];
    char auth_comp_id[CIFDATALEN];                  /* residue_name_used */
    char auth_asym_id[CIFDATALEN];                  /* strand_id(chainID) */
    char auth_seq_id[CIFDATALEN];                  /* residue_number */
    char PDB_ins_code[CIFDATALEN];                 /* Inersion code */
    char parent_comp_id[CIFDATALEN];               /* standard residue name */
    char details[256];                              /* details */

    struct ModRes *before;
    struct ModRes *next;
} ModRes, *ModResPtr;

```

## 4-2. 関数について

## ①ライブラリ内関数一覧

本ライブラリで使用される関数一覧を示す。

No.	関数名	概要
1	TypeSetRead	ファイル拡張子から、入力ファイルが pdb か、cif かを判定する。
2	TypeSetWrite	ファイル拡張子から、出力ファイルが pdb から、cif かを判定する。
3	InitStruct	分子系に関する構造体、及び、分子、原子、SS 結合、リンク情報に関する構造体のメモリ確保、初期化を行う。
4	RemoveStruct	分子系に関する構造体、及び、分子、原子、SS 結合、リンク情報に関する構造体のメモリ解放を行う。
5	Malloc	メモリの確保と初期化を行う。
6	mallocNextMol	複数の分子モデルの情報を保存する為に、分子の構造体変数のメモリ確保、初期化を行う。
7	mallocAtom	原子の構造体変数のメモリ確保、初期化を行う。
8	mallocNextAtom	次の原子の情報を保存する為に、原子の構造体変数のメモリ確保、初期化を行う。
9	mallocNextConnB	次のジスルフィド結合の情報を保存する為に、ジスルフィド結合の構造体変数のメモリ確保、初期化を行う。
10	mallocNextConnL	次のリンク情報を保存する為に、リンク情報の構造体変数のメモリ確保、初期化を行う。
11	mallocNextModRes	次の MODRES 情報を保存する為に、MODRES 情報の構造体変数のメモリ確保、初期化を行う。
12	mallocNextEnPoly	次の COMPND 情報を保存する為に、COMPND 情報の構造体変数のメモリ確保、初期化を行う。
13	mallocNextAssemGen	次の生物学的単位情報を保存する為に、AssemGen 情報の構造体変数のメモリ確保、初期化を行う。
14	mallocNextOperList	次の生物学的単位情報を保存する為に、OperList 情報の構造体変数のメモリ確保、初期化を行う。
15	comDataGet	文字列から開始位置、終了位置、文字数を取得する。
16	ReadPDBFile	pdb ファイルの読み込みメイン処理を行う。
17	setElement	入力 pdb ファイルに元素記号が記載されていない場合

		に、元素記号をセットする。
18	pdbAtom	pdb ファイルの分子情報を読み込む。
19	pdbSsbond	pdb ファイルの接続情報 (SSBOND) を読み込む。
20	pdbLink	pdb ファイルの接続情報 (LINK) を読み込む。
21	pdbModRes	pdb ファイルの MODRES 情報を読み込む。
22	getCompndData	pdb ファイルの COMPND 情報を読み込む。
23	pdbHed	pdb ファイルの分子名を読み込む。
24	ReadCifFile	cif ファイルの読み込みメイン処理を行う。
25	cifItemAtom	cif ファイルの分子情報の項目名を読み込む。
26	cifItemConn	cif ファイルの接続情報の項目名を読み込む。
27	cifItemEnPoly	cif ファイルの高分子情報の項目名を読み込む。
28	cifItemModRes	cif ファイルの修正 AA/NA 情報の項目名を読み込む。
29	cifItemAssemGen	cif ファイルの_pdbx_struct_assembly の項目名を読み込む。生物学単位の取得に使用する。
30	cifItemOperList	cif ファイルの_pdbx_struct_oper_list の項目名を読み込む。生物学単位の取得に使用する。
31	cifAtom	cif ファイルの分子情報の詳細情報取得メイン処理を行う。
32	cifConn	cif ファイルの接続情報の詳細情報取得メイン処理を行う。
33	cifEnPoly	cif ファイルの高分子情報の詳細情報取得メイン処理を行う。
34	cifModRes	cif ファイルの修正 AA/NA 情報の詳細情報取得メイン処理を行う。
35	cifAssemGen	cif ファイルの_pdbx_struct_assembly 情報の詳細情報取得メイン処理を行う。
36	cifOperList	cif ファイルの_pdbx_struct_oper_list 情報の詳細情報取得メイン処理を行う。
37	cifAtomData	cif ファイルの分子情報の詳細情報を読み込む。
38	cifConnData	cif ファイルの接続情報の詳細情報を読み込む。
39	cifEnPolyData	cif ファイルの高分子情報の詳細情報を読み込む。
40	cifModResData	cif ファイルの修正 AA/NA 情報を読み込む。
41	cifHed	cif ファイルの分子名を読み込む。
42	cifAssemGenData	cif ファイルの_pdbx_struct_assembly 情報(値)を読み込む。

43	cifOperListData	cif ファイルの_pdbx_struct_oper_list 情報(値)を読み込む。
44	ChangeInitC	pdb→cif 変換時の項目名初期セットを行う。
45	cifConnChange	cif→pdb 変換時の接続情報の変換を行う。
46	AtomNameViewC	原子名の出力位置を文字数と条件から設定する。
47	ConnNameSet	pdb→cif 変換時の接続情報 “LINK” を条件から “covale”・“metalc” に設定する。
48	complementDataAtom	pdb→cif、cif→pdb 変換時に、原子の構造体に関するデータの補間を行う。
49	matchAtomAndConn	原子の構造体の残基番号、alternate location indicator、原子名が、リンク情報のそれと一致している場合に、conn→cif_ptnr1_label_seq_id に atom→cif_label_seq_id の値をコピーする。
50	complementDataSS	pdb→cif、cif→pdb 変換時に、ジスルフィド結合の構造体に関するデータの補完を行う。
51	complementDataLink	pdb→cif、cif→pdb 変換時に、リンク情報の構造体に関するデータの補完を行う。
52	complementDataCompnd	pdb→cif、cif→pdb 変換時に、COMPND 情報(cif では _entity_poly)の構造体に関するデータの補完を行う。
53	complementDataModRes	pdb→cif、cif→pdb 変換時に、MODRES 情報(cif では _pdbx_struct_mod_residue の構造体に関するデータの補完を行う。
54	setCifItemAtomLength	cif ファイル出力時の原子項目の出力カラム数を計算する。
55	setCifItemConnLength	cif ファイル出力時のジスルフィド結合、リンク項目の出力カラム数を計算する。
56	complementData	pdb→cif、cif→pdb 変換時に、各データの補間を行う。
57	WritePDBFile	pdb ファイルの出力メイン処理を行う。
58	pdbAtomWdata	pdb ファイルの分子情報を出力する。
59	WriteCifFile	pdb ファイルの出力メイン処理を行う。
60	cifItemASet	pdb ファイルの分子情報の項目名を出力する。
61	cifItemCSet	pdb ファイルの接続情報の項目名を出力する。
62	cifAtomW	pdb ファイルの分子情報出力メイン処理を行う。
63	cifConnW	pdb ファイルの接続情報出力メイン処理を行う。
64	cifAtom1W	pdb ファイルの分子情報が一件の時の出力処理を行う。

65	cifConnlW	pdb ファイルの接続情報が一件の時の出力処理を行う。
66	cifAtomWData	pdb ファイルの分子情報の詳細情報を出力する。
67	cifConnWData	pdb ファイルの接続情報の詳細情報を出力する。
68	read_file_f77	f77 から呼び出された場合に、入力ファイルの読み込みを行う。
69	read_cif_f77	f77 から呼び出された場合に、cif ファイルの読み込みを行う。
70	read_pdb_f77	f77 から呼び出された場合に、pdb ファイルの読み込みを行う。
71	setf77Data	f77 の COMMON 文の各変数に値をセットする。
72	write_file_f77	f77 から呼び出された場合に、ファイル出力を行う。
73	write_cif_f77	f77 から呼び出された場合に、cif ファイル出力を行う。
74	write_pdb_f77	f77 から呼び出された場合に、pdb ファイル出力を行う。
75	setcStruct_f77	f77 から呼び出された場合に、c 構造体変数に COMMON 文の各値をセットする。
76	checkEachMaxValues	入力データが、モデル数の最大値、原子数の最大値、ジスルフィド結合の最大個数、リンク情報の最大個数を超える場合に、エラー処理を行う。
77	read_file	f90 から呼び出された場合に、入力ファイルの読み込みを行う。
78	read_cif	f90 から呼び出された場合に、cif ファイルの読み込みを行う。
79	read_pdb	f90 から呼び出された場合に、pdb ファイルの読み込みを行う。
80	setString	文字列のコピーを行い、終端の” ¥0” を” “(空白文字)に変更する。
81	setfStruct	f90 構造体変数に、c 構造体変数の値をコピーする。
82	write_file	f90 から呼び出された場合に、ファイルの出力を行う。
83	write_pdb	f90 から呼び出された場合に、pdb ファイルの出力を行う。
84	write_cif	f90 から呼び出された場合に、cif ファイルの出力を行う。
85	setcString	文字列のコピーを行い、終端に” ¥0” (ヌル文字)を付与する。
86	setcStruct	f90 から呼び出された場合に、c 構造体変数に f90 構造体

		変数の各値をセットする。
87	modifyResidueName	残基名のタイプに従い残基名を変換する。
88	modifyRes2Standard	残基名を標準 PDB 形式の残基名に変換する。
89	modifyRes2Presto	残基名を myPresto 形式の残基名 (ARG+, GLU-, HIS+, CYSS 等) に変換する。
90	modifyRes2Moe	残基名を MOE 形式の残基名に変換する。
91	modifyRes2Schrodinger	残基名を SCHRODINGER 形式の残基名に変換する。
92	isSSRes	着目した残基がジスルフィドを構成しているか否かを判定する。
93	chomp	文字列の末尾の改行コードを除去する。
94	removeRightSpace	文字列の途中から末尾まで空白文字が続くような値が入っている変数に対して、空白文字をヌル文字 ' ¥0 ' に変換する。
95	checkAndStrcpy	コピー元配列の文字列長と、コピー先配列の配列サイズをチェックした上で、文字列のコピーを行う。
96	checkAndStrcat	コピー元配列の文字列長と、コピー先配列の文字列長、配列サイズをチェックした上で、文字列の結合を行う。
97	checkAndStrncpy	コピー元配列の文字列長と、コピー先配列の配列サイズをチェックした上で、文字列のコピーを行う。
98	checkAndStrncat	コピー元配列の文字列長と、コピー先配列の文字列長、配列サイズをチェックした上で、文字列の結合を行う。
99	checkAndSscanf	コピーすべき文字列の文字列長と、コピー先配列の配列サイズをチェックした上で、文字列の取得を行う。

## ②c 言語プログラムから呼び出される関数一覧

No.	関数名	概要
1	read_cif_c	cif ファイルの読み込みを行う。
2	write_cif_c	cif ファイルの出力を行う。
3	read_pdb_c	pdb ファイルの読み込みを行う。
4	write_pdb_c	Pdb ファイルの出力を行う。
5	read_file_c	入力ファイルの読み込みを行う。
6	write_file_c	ファイル出力を行う。
7	Malloc	メモリの確保と初期化を行う。
8	RemoveStruct	分子系に関する構造体、及び、分子、原子、SS 結合、リンク情報に関する構造体のメモリ解放を行う。

## ③f90 プログラムから呼び出される関数一覧

No.	関数名	概要
1	read_cif	cif ファイルの読み込みを行う。
2	write_cif	cif ファイルの出力を行う。
3	read_pdb	pdb ファイルの読み込みを行う。
4	write_pdb	Pdb ファイルの出力を行う。
5	read_file_	入力ファイルの読み込みを行う。
6	write_file_	ファイル出力を行う。

## ④f77 プログラムから呼び出される関数一覧

No.	関数名	概要
1	read_cif_f77	cif ファイルの読み込みを行う。
2	write_cif_f77	cif ファイルの出力を行う。
3	read_pdb_f77	pdb ファイルの読み込みを行う。
4	write_pdb_f77	Pdb ファイルの出力を行う。
5	read_file_f77_	入力ファイルの読み込みを行う。
6	write_file_f77_	ファイル出力を行う。

## 5. プログラムサンプルについて

各言語における本ライブラリの呼び出しサンプルプログラムを示します。

## 5-1. c 言語サンプルプログラム

以下に cif ファイルを読み込み pdb ファイルを出力する、c サンプルプログラムを示します。サンプルの太字下線部は必須の項目となります。

```

#include <stdio.h> //include 文(必須)
#include <stdlib.h>
#include <string.h>
#include "ToolDefine.h"
#include "ToolmyPresto.h"

void read_cif_c(char *in_file, SysPtr sys); //プロトタイプ宣言(必須)
void write_pdb_c(char *out_file, SysPtr sys);
void RemoveStruct(SysPtr sys);
void *Malloc(int size);

int main(int argc, char *argv[])
{
    char in_file[FILEBUFSIZE+1];
    char out_file[FILEBUFSIZE+1];
    charstyle[100];
    SysPtr sys = NULL;

    if((sys = (SysPtr) Malloc(sizeof(Sys))) == 0) // 構造体変数のメモリ確保(必須)
    {
        printf(" ERROR> memory allocate\n");
        exit(1);
    }

    if(argc != 4 && argc != 3) { // 引数の数のチェック
        printf(" ERROR> Input Err ,InputFile OutFile %n");
        /* add 2015.1.21, start */
        printf(" ccif2pdb <in_file(cif)> <out_file(pdb)>%n");
        printf(" or\n");
        printf(" ccif2pdb <in_file(cif)> <out_file(pdb)> <residue_style>%n");
        printf(" residue_style [presto|stdpdb|moe|schrodinger]%n");
        printf(" presto: myPresto pdb style(ARG+, GLU-, CYSS, etc)%n");
        printf(" stdpdb: standard pdb style\n");
        printf(" moe : MOE pdb style(HID, HIE, HIP)%n");
        printf(" schrodinger: SCHRODINGER pdb style(HID, HIE, HIP, CYX, etc)%n");
        /* add end */

        exit(1);
    }
    if(strlen(argv[1]) > FILEBUFSIZE) { // 引数のチェック
        printf(" ERROR> Inputname is long\n");
        return FALSE;
    }
    if(strlen(argv[2]) > FILEBUFSIZE) { // 引数のチェック
        printf(" ERROR> Outname is long\n");
        return FALSE;
    }
}

```



```
strcpy(in_file, argv[1]); // ファイル名の取得
strcpy(out_file, argv[2]);

if(argc == 4){
    strcpy(style, argv[3]);
}

sys->intRType = READCIF;
sys->intWType = WRITEPDB;
/* add end */

/* add 2015.1.21, start */
sys->style = NOCHANGE;
if(argc == 4){ // 出力ファイルの残基名タイプの選択(必要な場合のみ指定する。)
    if(strcmp(style, "stdpdb") == 0) sys->style = STANDARD;
    else if(strcmp(style, "presto") == 0) sys->style = PRESTO;
    else if(strcmp(style, "moe") == 0) sys->style = MOE;
    else if(strcmp(style, "schrodinger") == 0) sys->style = SCHRODINGER;
}

/* read file */
read_cif_c(in_file, sys); // ファイル読み込み

/* write file */
write_pdb_c(out_file, sys); // ファイル出力

/* memory free */
RemoveStruct(sys); // 構造体変数のメモリ解放

return 0;
```

## 5-2. f77 言語サンプルプログラム

以下に cif ファイルを読み込み、pdb ファイルを出力する f77 サンプルプログラムを示します。サンプルの太字下線部は必須の項目となります。

```

PROGRAM f77cif2pdb
IMPLICIT REAL (A-H,O-Z)
INTEGER MXMDL, MXLKNM, MXSSNM, MXEPNM, MXMDNM
INTEGER MXMDLATM
INTEGER MXAGNM, MXOLNM
C   MXMDL: max number of models
C   MXLKNM: max number of link lines
C   MXSSNM: max number of ssbond lines
C   MXMDLATM: max number of atoms in one model
C   MXEPNM: max number of entity_poly lines
C   MXMDNM: max number of pdbx_struct_mod_residue lines
C   MXAGNM: max number of pdbx_struct_assem_gen lines
C   MXOLNM: max number of pdbx_struct_oper_list lines
PARAMETER (MXMDL=2000, MXLKNM=10000, MXSSNM=2000)
PARAMETER (MXMDLATM=5000000)
PARAMETER (MXEPNM=80, MXMDNM=80)
PARAMETER (MXAGNM=80, MXOLNM=80)
COMMON F77ATM
C   GRPPDB: _atom_site.group_PDB ("ATOM" or "HETATM")
C   NUM: serial number
C   ATMNUM: atom number
C   ATMNAM: atom name, not atom element
C   ALTLOC: alternate location indicator (use to write pdb)
C   RESNAM: residue name
C   CID: chain ID
C   RESNUM: residue number
C   ICODE: insertion code
C   CODX, CODY, CODZ: atom coordinates
C   OCCUP: occupancy
C   TFAC: temp factor
C   ELEM: atom element name
C   CHARGE: charge in pdb
C   CCHARGE: charge in cif
C   AMDLNM: model number
C   CIFNUM: _atom_site.id
C   CLATID: _atom_site.label_atom_id
C   CLALID: _atom_site.label_alt_id
C   CLCPID: _atom_site.label_comp_id
C   CLASID: _atom_site.label_asym_id
C   CLETID: _atom_site.label_entity_id
C   CLSQID: _atom_site.label_seq_id
C   CIFINS: _atom_site.pdbx_PDB_ins_code
C   CXESD: _atom_site.Cartn_x_esd
C   CYESD: _atom_site.Cartn_y_esd
C   CZESD: _atom_site.Cartn_z_esd
C   COCESD: _atom_site.occupancy_esd
C   CISOEQ: _atom_site.B_iso_or_equiv_esd
COMMON/F77ATM/GRPPDB (MXMOL, MXATM), NUM (MXMOL, MXATM),
* ATMNUM (MXMOL, MXATM), ATMNAM (MXMOL, MXATM), ALTLOC (MXMOL, MXATM),
* RESNAM (MXMOL, MXATM), CID (MXMOL, MXATM), RESNUM (MXMOL, MXATM),
* ICODE (MXMOL, MXATM), CODX (MXMOL, MXATM), CODY (MXMOL, MXATM),
* CODZ (MXMOL, MXATM), OCCUP (MXMOL, MXATM), TFAC (MXMOL, MXATM),
* ELEM (MXMOL, MXATM), CHARGE (MXMOL, MXATM), CCHARGE (MXMOL, MXATM),
* AMDLNM (MXMOL, MXATM), CIFNUM (MXMOL, MXATM), CLATID (MXMOL, MXATM),
* CLALID (MXMOL, MXATM), CLCPID (MXMOL, MXATM), CLASID (MXMOL, MXATM),
* CLETID (MXMOL, MXATM), CLSQID (MXMOL, MXATM), CIFINS (MXMOL, MXATM),
* CXESD (MXMOL, MXATM), CYESD (MXMOL, MXATM), CZESD (MXMOL, MXATM),
* COCESD (MXMOL, MXATM), CISOEQ (MXMOL, MXATM)

```

```

C  VARIABLES OF F77ATM
C  CHARACTER*8 GRPPDB
C  CHARACTER*4 ALTLOC, CID, ICODE, ELEM, CHARGE, CCHRG
C  CHARACTER*4 CLALID, CLASID, CIFINS
C  CHARACTER*8 ATMNAM, RESNAM, CLATID, CLCPID
C  CHARACTER*12 CLETID, GLSQID, CXESD, CYESD, CZESD, COCESD, CISOEQ
C  INTEGER NUM, ATMNUM, RESNUM, AMDLNM, CIFNUM
C  REAL CODX, GODY, GODZ, OCGUP, TFAC
C  COMMON F77ENP
C  EID: _entity_poly.entity_id
C  TYPE: _entity_poly.type
C  LNK: _entity_poly.nstd_linkage
C  MON: _entity_poly.nstd_mononer
C  SEQ: _entity_poly.pdbx_seq_one_letter_code
C  SEQC: _entity_poly.pdbx_seq_one_letter_code_can
C  SID: _entity_poly.pdbx_strand_id
C  MLID: molID
C  CIDS: chainIDs
C  NCHS: num_chainIDs
C  COMMON/F77ENP/EID (MXEPNM), TYP (MXEPNM), LNK (MXEPNM), MON (MXEPNM),
C  * SEQ (MXEPNM), SEQC (MXEPNM), SID (MXEPNM), MLID (MXEPNM),
C  * NCHS (MXEPNM), CIDS (MXEPNM, 32768)

C  VARIABLE OF F77ENP
C  CHARACTER*8 CIDS
C  CHARACTER*12 EID, LNK, MON
C  CHARACTER*256 TYP
C  CHARACTER*32768 SEQ, SEQC, SID
C  INTEGER MLID, NCHS

C  COMMON F77MOD
C  MPID: pdbid to write pdb
C  MID: pdbid to write cif
C  MCID: _pdbx_struct_mod_residue.label_comp_id
C  MASID: _pdbx_struct_mod_residue.label_asym_id
C  MSID: _pdbx_struct_mod_residue.label_seq_id
C  MACID: _pdbx_struct_mod_residue.auth_comp_id
C  MAAID: _pdbx_struct_mod_residue.auth_asym_id
C  MAASQ: _pdbx_struct_mod_residue.auth_seq_id
C  MPICD: _pdbx_struct_mod_residue.PDB_ins_code
C  MPCID: _pdbx_struct_mod_residue.parent_comp_id
C  MDET: _pdbx_struct_mod_residue.details
C  COMMON/F77MOD/MPID (MXMDNM), MID (MXMDNM), MCID (MXMDNM), MASID (MXMDNM),
C  * MSID (MXMDNM), MACID (MXMDNM), MAAID (MXMDNM), MAASQ (MXMDNM),
C  * MPICD (MXMDNM), MPCID (MXMDNM), MDET (MXMDNM)
C  CHARACTER*8 MPID
C  CHARACTER*12 MID, MCID, MASID, MSID, MACID, MAAID, MAASQ, MPICD, MPCID
C  CHARACTER*256 MDET

C  COMMON F77AGEN
C  AASID: _pdbx_struct_assembly_gen.assembly_id
C  AOPEX: _pdbx_struct_assembly_gen.oper_expression
C  AALST: _pdbx_struct_assembly_gen.asym_id_list
C  COMMON/F77AGEN/AASID (MXAGNM), AOPEX (MXAGNM), AALST (MXAGNM)
C  CHARACTER*12 AASID
C  CHARACTER*128 AOPEX
C  CHARACTER*32768 AALST

C  COMMON F77OLST
C  OID: _pdbx_struct_oper_list.id
C  OTYP: _pdbx_struct_oper_list.type
C  ONAM: _pdbx_struct_oper_list.name
C  OSYOP: _pdbx_struct_oper_list.symmetry_operation
C  OMAT11: _pdbx_struct_oper_list.matrix[1][1]
C  OMAT12: _pdbx_struct_oper_list.matrix[1][2]

```

```

C   OMAT13: _pdbx_struct_oper_list.matrix[1][3]
C   OVECT1: _pdbx_struct_oper_list.vector[1]
C   OMAT21: _pdbx_struct_oper_list.matrix[2][1]
C   OMAT22: _pdbx_struct_oper_list.matrix[2][2]
C   OMAT23: _pdbx_struct_oper_list.matrix[2][3]
C   OVECT2: _pdbx_struct_oper_list.vector[2]
C   OMAT31: _pdbx_struct_oper_list.matrix[3][1]
C   OMAT32: _pdbx_struct_oper_list.matrix[3][2]
C   OMAT33: _pdbx_struct_oper_list.matrix[3][3]
C   OVECT3: _pdbx_struct_oper_list.vector[3]
COMMON/F77OLST/OID (MXOLNM), OTYP (MXOLNM), ONAM (MXOLNM),
*OSYOP (MXOLNM), OMAT11 (MXOLNM), OMAT12 (MXOLNM), OMAT13 (MXOLNM),
*OVECT1 (MXOLNM), OMAT21 (MXOLNM), OMAT22 (MXOLNM), OMAT23 (MXOLNM),
*OVECT2 (MXOLNM), OMAT31 (MXOLNM), OMAT32 (MXOLNM), OMAT33 (MXOLNM)
CHARACTER*12 OID, ONAM
CHARACTER*256 OTYP, OSYOP
REAL OMAT11, OMAT12, OMAT13, OVECT1
REAL OMAT21, OMAT22, OMAT23, OVECT2
REAL OMAT31, OMAT32, OMAT33, OVECT3

C   COMMON F77MOL
C   ATMNMS: number of atoms in molecule
COMMON/F77MOL/ATMNMS (MXMOL)

C   VARIABLES OF F77MOL
INTEGER ATMNMS
C   COMMON F77SYS
C   MDLNUM: number of models
C   LKNUM: number of links data
C   SSNUM: number of ssbond data
COMMON/F77SYS/MDLNUM, LKNUM, SSNUM, EPNUM, MGRNM, MDNUM, PDBID

C   VARIABLES OF F77SYS
INTEGER MDLNUM, LKNUM, SSNUM
CHARACTER*8 PDBID
C   COMMON F77LNK
C   LNUM: serial number
C   LBNDNM: bond number
C   LATNAM1: atom name1
C   LALT1: _struct_conn.pdbx_ptnr1_label_alt_id
C   LRSNAM1: residue name1
C   LCID1: chain id1
C   LRSNUM1: residue number1
C   LICOD1: insersion code1
C   LATNAM2: atom name2
C   LALT2: _struct_conn.pdbx_ptnr2_label_alt_id
C   LRSNAM2: residue name2
C   LCID2: chain id2
C   LRSNUM2: residue number2
C   LICOD2: insersion code2
C   LSYM1: symmetry operation1 (use to write pdb file)
C   LSYM2: symmetry operation1 (use to write pdb file)
C   LLENG: _struct_conn.pdbx_dist_value
C   LCSYM1: symmetry operation1 (use to wrrire cif file)
C   LCSYM2: symmetry operation1 (use to wrrire cif file)
C   LCID: _struct_conn.id
C   LCCNID: _struct_conn.conn_type_id
C   LCPID: _struct_conn.pdbx_PDB_id
C   LCASM1: _struct_conn.ptnr1_label_asym_id
C   LCCMP1: _struct_conn.ptnr1_label_comp_id
C   LCSQ1: _struct_conn.ptnr1_label_seq_id
C   LCALT1: _struct_conn.pdbx_ptnr1_label_alt_id

```

```

C      LCINS1: _struct_conn.pdbx_ptnr1_PDB_ins_code
C      LCSTC1: _struct_conn.pdbx_ptnr1_standard_comp_id
C      LCASM2: _struct_conn.ptnr2_label_asym_id
C      LCCMP2: _struct_conn.ptnr2_label_comp_id
C      LCSQ2: _struct_conn.ptnr2_label_seq_id
C      LCALT2: _struct_conn.pdbx_ptnr2_label_alt_id
C      LCINS2: _struct_conn.pdbx_ptnr2_PDB_ins_code
C      LCATID3: _struct_conn.pdbx_ptnr3_label_atom_id
C      LCSQ3: _struct_conn.pdbx_ptnr3_label_seq_id
C      LCCMP3: _struct_conn.pdbx_ptnr3_label_comp_id
C      LCASM3: _struct_conn.pdbx_ptnr3_label_asym_id
C      LCALT3: _struct_conn.pdbx_ptnr3_label_alt_id
C      LCINS3: _struct_conn.pdbx_ptnr3_PDB_ins_code
C      LDTAIL: _struct_conn.details
C      LVLOR: _struct_conn.pdbx_value_order
C      COMMON/F77LNK/LNUM (MXLKNM), LBNDNM (MXLKNM), LATNAM1 (MXLKNM),
* LALT1 (MXLKNM), LRSNAM1 (MXLKNM), LCID1 (MXLKNM), LRSNUM1 (MXLKNM),
* LICOD1 (MXLKNM), LATNAM2 (MXLKNM), LALT2 (MXLKNM),
* LRSNAM2 (MXLKNM), LCID2 (MXLKNM), LRSNUM2 (MXLKNM), LICOD2 (MXLKNM),
* LSYM1 (MXLKNM), LSYM2 (MXLKNM), LENG (MXLKNM), LGSYM1 (MXLKNM),
* LGSYM2 (MXLKNM), LGID (MXLKNM), LCGNID (MXLKNM), LCPID (MXLKNM),
* LGASM1 (MXLKNM), LCCMP1 (MXLKNM), LCSQ1 (MXLKNM),
* LCALT1 (MXLKNM), LCINS1 (MXLKNM), LCSTC1 (MXLKNM),
* LGASM2 (MXLKNM), LCCMP2 (MXLKNM), LCSQ2 (MXLKNM),
* LCALT2 (MXLKNM), LCINS2 (MXLKNM),
* LCATID3 (MXLKNM), LCSQ3 (MXLKNM), LCCMP3 (MXLKNM), LGASM3 (MXLKNM),
* LCALT3 (MXLKNM), LCINS3 (MXLKNM), LDTAIL (MXLKNM), LVLOR (MXLKNM)
! LINK 情報に関する COMMON 文

C      VARIABLES OF F77LNK
      INTEGER LNUM, LBNDNM, LRSNUM1, LRSNUM2, LSYM1, LSYM2
      CHARACTER*4 LALT1, LICOD1, LALT2, LICOD2
      CHARACTER*8 LATNAM1, LRSNAM1, LATNAM2, LRSNAM2
      CHARACTER*8 LCID1, LCID2
      CHARACTER*12 LGSYM1, LGSYM2, LGID, LCGNID, LCPID, LGASM1, LCCMP1,
* LCSQ1, LCALT1, LCINS1, LCSTC1, LGASM2, LCCMP2, LCSQ2,
* LCALT2, LCINS2, LCATID3, LCSQ3, LCCMP3, LGASM3, LCALT3,
* LCINS3, LVLOR
      CHARACTER*128 LDTAIL
      REAL LENG

C      COMMON F77SSBN
      COMMON/F77SSBN/SNUM (MXSSNM), SBNDNM (MXSSNM), SATNAM1 (MXSSNM),
* SALT1 (MXSSNM), SRSNAM1 (MXSSNM), SCID1 (MXSSNM), SRSNUM1 (MXSSNM),
* SICOD1 (MXSSNM), SATNAM2 (MXSSNM), SALT2 (MXSSNM),
* SRSNAM2 (MXSSNM), SCID2 (MXSSNM), SRSNUM2 (MXSSNM), SICOD2 (MXSSNM),
* SSYM1 (MXSSNM), SSYM2 (MXSSNM), SLENG (MXSSNM), SCSYM1 (MXSSNM),
* SCSYM2 (MXSSNM), SCID (MXSSNM), SCCNID (MXSSNM), SCPID (MXSSNM),
* SCASM1 (MXSSNM), SCCMP1 (MXSSNM), SCSQ1 (MXSSNM),
* SCALT1 (MXSSNM), SCINS1 (MXSSNM), SCSTC1 (MXSSNM),
* SCASM2 (MXSSNM), SCCMP2 (MXSSNM), SCSQ2 (MXSSNM),
* SCALT2 (MXSSNM), SCINS2 (MXSSNM),
* SCATID3 (MXSSNM), SCSQ3 (MXSSNM), SCCMP3 (MXSSNM), SCASM3 (MXSSNM),
* SCALT3 (MXSSNM), SCINS3 (MXSSNM), SDTAIL (MXSSNM), SVLOR (MXSSNM)
! SSBOND に関する COMMON 文

C      VARIABLES OF F77SSBN
      INTEGER SNUM, SBNDNM, SRSNUM1, SRSNUM2, SSYM1, SSYM2
      CHARACTER*4 SALT1, SICOD1, SALT2, SICOD2
      CHARACTER*8 SATNAM1, SRSNAM1, SATNAM2, SRSNAM2
      CHARACTER*8 SCID1, SCID2
      CHARACTER*12 SCSYM1, SCSYM2, SCID, SCCNID, SCPID, SCASM1, SCCMP1,
* SCSQ1, SCALT1, SCINS1, SCSTC1, SCASM2, SCCMP2, SCSQ2,
* SCALT2, SCINS2, SCATID3, SCSQ3, SCCMP3, SCASM3, SCALT3,
* SCINS3, SVLOR
      CHARACTER*128 SDTAIL
      REAL SLENG

```

```
EXTERNAL read_file_f77, write_file_f77                ! 外部関数の宣言
  
CHARACTER*1024 in_file, out_file
C  OTHER VARIABLES
MDLNUM = 0
LNKNUM = 0
SSNUM = 0
EPNUM = 0
MGRNM = 0
MDNUM = 0
AGNUM = 0
OPNUM = 0
  
C  set input filename
print *, " %% INPUT FILE NAME OF ATOM COORD. %%"
read (*,'(A)') in_file                                ! ファイル名取得
  
C  set output filename
print *, " %% INPUT FILE NAME OF OUTPUT FILE %%"
read (*,'(A)') out_file                              ! ファイル名取得
  
C  read inputfile
call read_cif_f77(in_file, MXATM, MXMOL, MXSSNM, MXLKNM, MXEPNM,
* MXMDNM)                                           ! ファイル読み込み
  
C  write outputfile
call write_pdb_f77(out_file)                          ! ファイル出力
  
END PROGRAM f77cif2pdb
```

## 5-3. f90 言語サンプルプログラム

f90 サンプルプログラムを示します。サンプルは f90MolStruct.f90 と f90cif2pdb.f90 からなります。

f90MolStruct.f90

```

module parameters
  implicit none
  integer, parameter :: MAXMDLATM = 5000000 ! max number of models, 系の最大原子数
  integer, parameter :: MAXMDL = 2000 ! max number of models, 最大モデル数
  ! not number of proteins !
  integer, parameter :: MAXLNKNUM = 10000 ! max number of link lines 最大LINK行数
  integer, parameter :: MAXSSNUM = 2000 ! max number of ssbond lines 最大SSBOND行数
  integer, parameter :: MAXEPNUM = 80 ! max number of entity_poly lines 最大高分子情報数
  integer, parameter :: MAXMDNUM = 80 ! mac number of pdbx_struct_mod_residue 最大修正AA数

  integer, parameter :: MAXAGNUM = 80 ! max number of pdbx_struct_assem_gen lines 生物学単位最大数
  integer, parameter :: MAXOLNUM = 80 ! max number of pdbx_struct_oper_list lines 生物学単位最大数

  integer, parameter :: NAMELEN = 8 ! 原子名、残基名の最大文字数
  integer, parameter :: CIFDATALEN = 12 ! cifデータの配列サイズ
  integer, parameter :: BUFSIZE = 1024 ! バッファサイズ
  integer, parameter :: CIFRESMAX = 32768
end module parameters

module cmolstruct
  use parameters

  !-----
  ! struct fatom, structure of atom information 原子情報に関する構造体
  !-----
  type fatom
    sequence
    character(len=8) :: group_PDB ! record name 1: ATOM, 2: HETATM
    integer :: num ! serial number
    integer :: atom_num ! atom number
    character(len=NAMELEN) :: atom_name ! atom name
    character(len=4) :: altLoc ! alternate location indicator
    character(len=NAMELEN) :: res_name ! residue name
    character(len=8) :: chainID ! chain id
    integer :: res_num ! residue number
    character(len=4) :: iCode ! insertion code
    real :: x, y, z ! coordinate
    real :: occupancy ! occupancy
    real :: tempFactor ! same _atom_site.B_iso_or_equiv
    character(len=4) :: element ! atom element
    character(len=4) :: charge ! formal charge
    character(len=4) :: cif_charge ! formal charge (use to write cif file)

    integer :: model_num ! model number
    integer :: cif_num ! same _atom_site.id

    character(len=NAMELEN) :: cif_label_atom_id ! (cif出力用)原子名
    character(len=4) :: cif_label_alt_id ! alternate location indicator
    character(len=NAMELEN) :: cif_label_comp_id ! 残基名
    character(len=8) :: cif_label_asym_id ! chain id
    character(len=CIFDATALEN) :: cif_label_entity_id ! entity_id
    character(len=CIFDATALEN) :: cif_label_seq_id ! 残基番号
    character(len=4) :: cif_pdbx_PDB_ins_code ! insersion code
  end type fatom

```

```

character(len=CIFDATALEN) :: cif_cartrn_x_esd      ! The standard uncertainty
character(len=CIFDATALEN) :: cif_cartrn_y_esd      ! The standard uncertainty
character(len=CIFDATALEN) :: cif_cartrn_z_esd      ! The standard uncertainty
character(len=CIFDATALEN) :: cif_occupancy_esd     ! The standard uncertainty
character(len=CIFDATALEN) :: cif_b_iso_or_equiv_esd ! b_iso_or_equiv_esd
end type fatom

!-----
! struct fssbond, structure of ssbond, covalent,      LINK、SSBOND 情報に関する構造体
! ion bonds information
!-----
type fconn
sequence
integer :: num          ! serial number
integer :: bond_num     ! serial number of SSBOND line
character(len=NAMELEN) :: atom_name1 ! atom name of connected atom1
character(len=4) :: altLoc1 ! alternate location indicator of atom1
character(len=NAMELEN) :: res_name1 ! residue name of connected atom1
character(len=8) :: chainid1 ! chain id of atom1
integer :: res_num1     ! residue number of atom1
character(len=4) :: icode1 ! insersion code of atom1
character(len=NAMELEN) :: atom_name2 ! atom name of connected atom2
character(len=4) :: altLoc2 ! alternate location indicator of atom2
character(len=NAMELEN) :: res_name2 ! residue name of connected atom2
character(len=8) :: chainid2 ! chain id of atom2
integer :: res_num2     ! residue number of atom2
character(len=4) :: icode2 ! insersion code of atom2

integer :: simmetry1 ! simmetry operation1
integer :: simmetry2 ! simmetry operation2

real :: length ! connection distance

character(len=CIFDATALEN) :: cif_ptnr1_symmetry ! 原子 1 の対称性
character(len=CIFDATALEN) :: cif_ptnr2_symmetry ! 原子 2 の対称性

character(len=CIFDATALEN) :: cif_id ! cif id
character(len=CIFDATALEN) :: cif_conn_type_id ! type
character(len=CIFDATALEN) :: cif_pdbx_PDB_id ! pdb id

character(len=CIFDATALEN) :: cif_ptnr1_label_asym_id ! 原子 1 の chain id
character(len=CIFDATALEN) :: cif_ptnr1_label_comp_id ! 同残基名
character(len=CIFDATALEN) :: cif_ptnr1_label_seq_id ! 同残基番号
character(len=CIFDATALEN) :: cif_pdbx_ptnr1_label_alt_id ! 同 alternate location indicator
character(len=CIFDATALEN) :: cif_pdbx_ptnr1_PDB_ins_code ! 同 insertion code
character(len=CIFDATALEN) :: cif_pdbx_ptnr1_standard_comp_id ! 同残基名

character(len=CIFDATALEN) :: cif_ptnr2_label_asym_id ! 原子 2 の chain id
character(len=CIFDATALEN) :: cif_ptnr2_label_comp_id ! 同残基名
character(len=CIFDATALEN) :: cif_ptnr2_label_seq_id ! 同残基番号
character(len=CIFDATALEN) :: cif_pdbx_ptnr2_label_alt_id ! 同 alternate location indicator
character(len=CIFDATALEN) :: cif_pdbx_ptnr2_PDB_ins_code ! 同 insertion code

character(len=CIFDATALEN) :: cif_pdbx_ptnr3_label_atom_id ! 原子 3 の原子名
character(len=CIFDATALEN) :: cif_pdbx_ptnr3_label_seq_id ! 同残基番号
character(len=CIFDATALEN) :: cif_pdbx_ptnr3_label_comp_id ! 同残基名
character(len=CIFDATALEN) :: cif_pdbx_ptnr3_label_asym_id ! 同 chain id
character(len=CIFDATALEN) :: cif_pdbx_ptnr3_label_alt_id ! 同 alternate location indicator
character(len=CIFDATALEN) :: cif_pdbx_ptnr3_PDB_ins_code ! 同 insersion code
character(len=128) :: cif_details ! A description of special aspects of the connection.
character(len=CIFDATALEN) :: cif_pdbx_value_order ! The chemical bond order associated with
end type fconn ! the specified atom in this contact.

```



```

!-----
! struct fempoly, structure of entity_poly information          高分子鎖に関する構造体
!-----
type fempoly
sequence
integer :: num          ! serial number

character(len=CIFDATALEN) :: cif_entity_id
character(len=256)        :: cif_type
character(len=CIFDATALEN) :: cif_nstd_linkage
character(len=CIFDATALEN) :: cif_nstd_monomer
character(len=CIFRESMAX)  :: cif_pdbx_seq_one_letter_code
character(len=CIFRESMAX)  :: cif_pdbx_seq_one_letter_code_can
character(len=CIFRESMAX)  :: cif_pdbx_strand_id

integer :: molID
character, dimension(32786,8) :: chainIDs
integer :: num_chainIDs

end type fempoly

!-----
! struct fmodres, structure of _pdbx_struct_mod_residue information  修正 AA/NA に関する構造体
!-----
type fmodres
sequence
character(len=8)          :: pdbid
character(len=CIFDATALEN) :: id
character(len=CIFDATALEN) :: label_comp_id      ! 修飾 AA/NA の残基名
character(len=CIFDATALEN) :: label_asym_id     ! 鎖 ID
character(len=CIFDATALEN) :: label_seq_id      ! 残基番号
character(len=CIFDATALEN) :: auth_comp_id
character(len=CIFDATALEN) :: auth_asym_id     ! ユーザ指定鎖 ID
character(len=CIFDATALEN) :: auth_seq_id     ! ユーザ指定残基番号
character(len=CIFDATALEN) :: PDB_ins_code
character(len=CIFDATALEN) :: parent_comp_id   ! ベースとなる AA/NA 残基名
character(len=256) :: details

end type fmodres

!-----
! struct fassemGen, structure of _pdbx_struct_assem_gen information  生物学的単位作成用構造体
!-----
type fassemGen
sequence
character(len=CIFDATALEN) :: assembly_id
character(len=128) :: oper_expression
character(len=32768) :: asym_id_list
end type fassemGen

!-----
! struct foperList, structure of _pdbx_struct_oper_list information  生物学的単位作成用構造体
!-----
type foperList
sequence
character(len=CIFDATALEN) :: id
character(len=128) :: type
character(len=CIFDATALEN) :: name
character(len=128) :: symmetry_operation
real :: matrix11, matrix12, matrix13
real :: matrix21, matrix22, matrix23
real :: matrix31, matrix32, matrix33
real :: vector1, vector2, vector3
end type foperList

```

```

!-----
! struct fmol, structure of molecule information      分子に関する構造体
!-----
type fmol
  sequence
  integer :: atom_num;                               ! 原子数
  type(fatom) atoms(1:MAXATOM)                     ! 原子に関する構造体
end type fmol

!-----
! struct fsys, structure of system information        ! 分子系に関する構造体
!-----
type fsys
  sequence
  character(len=NAMELEN) :: pdbid                   ! pdb id
  integer :: model_num, ssbond_num, link_num        ! モデル数、SSBOND 数、LINK 数

  type(fmol) :: mols(1:MAXMDL)                      ! 分子に関する構造体
  type(fconn) :: links(1:MAXLNKNUM)                 ! LINK に関する構造体
  type(fconn) :: ssbonds(1:MAXSSNUM)                ! SSBOND に関する構造体
end type fsys

end module cmolstruct

```

f90cif2pdb.f90

```

program f90cif2pdb
  use parameters                                     ! モジュールの使用
  use cmolstruct                                    ! f90MolStruct の内容を使用する

  external read_cif, write_pdb                       ! 外部関数定義
  character(len=BUFSIZE) :: in_file, out_file
  type(fsys) :: sys;                                ! 構造体変数
  integer :: istyle                                   ! 残基名のタイプを選択、選択不要時は 0

! set input filename
  print *, " %% INPUT FILE NAME OF ATOM COORD. %%"
  read (*, '(A)') in_file

! set output filename
  print *, " %% INPUT FILE NAME OF OUTPUT FILE %%"
  read (*, '(A)') out_file

  istyle = 0
  call read_cif(in_file, istyle, MAXMDLATM, MAXMDL, MAXSSNUM, MAXLNKNUM, MAXEPNUM, MAXMDNUM, MAXAGNUM,
MAXOLNUM, sys)                                     ! ファイル読み込み処理

  call write_pdb(out_file, sys)                       ! ファイル出力処理
end program f90cif2pdb

```

## 6. 制限事項について

### 6-1. cif ファイル読み込み範囲について

本ライブラリで pdbx/mmCIF を読み込む際に、読み込み対象を以下のカテゴリーとしています。よって、今回のコンバート機能は、読み込み対象カテゴリーの範囲内で変換可能な機能を実装しています。よって、以下に示す以外のカテゴリーの内容が必要となる場合には、その部分の変換、出力は行われません。

読み込み対象カテゴリー：

_entry.id	pdb id に相当する情報
_struct_conn カテゴリー	pdb の SSBOND、LINK 行に相当する情報
_atom_site カテゴリー	pdb の ATOM、HETATM 行に相当する情報
_pdbx_struct_mod_residue カテゴリー	pdb の MODRES 行に相当する情報
_entity_poly カテゴリー	pdb の COMPND MOLID、COMPND CHAIN に相当する部分のみを対象とします。

### 6-2. Fortran から呼び出す場合のメモリ使用量について

f90、f77 からライブラリを呼び出す場合に、前者では構造体を設定し、後者では COMMON 文で c ライブラリに渡す情報を定義しているが、各変数のサイズ、メモリ使用サイズを最大モデル数、最大原子数の積のメモリを使用する事となります。最大モデル数、最大原子数を大きく設定すると、それに従いメモリ使用量も大きくなります。

c 言語からライブラリを呼び出す場合には、動的にメモリを確保していますので、fortran 程メモリを使用する事はありません。

### 6-3. pdb ファイルの読み込み範囲について

本ライブラリで pdb を読み込む際に、MODRES キーワード情報を読み込むようにします。これは、HETATM 行の残基がアミノ酸/核酸として認識するか、それ以外(リガンド、金属、水等)として認識するかの指標として使用します。

よって、修飾アミノ酸/核酸が存在する場合に、MODRES キーワードが存在しない場合には、その残基は修飾アミノ酸/核酸以外の残基(分子)として認識します。

#### 6-4. PDB 出力時の問題について

cif ファイルでは、大きな分子の場合において `chainID(_atom_site.label_asym_id, _atom_site.auth_asym_id)` が 1 文字とは限らず複数文字で表現されている場合があり、このようなファイルを pdb に変換、出力する場合に、現状ではその文字数の chainID の 1 文字目のみが出力されます。よって、複数文字の chainID の場合は、ChainID が必ずしも正しく変換できない問題が発生します。