

tpIgene システム設計書
tpIgeneL

- ・ tpIgeneシステム設計書
 - ・ tpIgeneLシステム設計書
-

myPresto/tplgene システム設計書
(Topology Generator 編)
第 1 版

Copyright (C) 2006-2008 National Institute of Advanced Industrial Science and Technology (AIST)
Copyright (C) 2006-2008 Japan Biological Informatics Consortium (JBIC)

目次 -

1 . はじめに	- 4
2 . システム概要	- 5
2-1 高分子用トポロジージェネレータと低分子用トポロジージェネレータの関係 .	- 5
2-2 高分子トポロジージェネレータ tplgene のシステム概要	- 8
3 . トポロジージェネレータの処理	-10
3-1 高分子トポロジージェネレータ tplgene の処理	-10
3-1-1 分子情報の取得	-11
3-1-2 カ場パラメータ DB 読込	-14
3-1-3 トポロジー生成	-14
3-1-4 環状分子の補正	-14
3-1-5 トポロジーファイルの出力	-14
3-1-6 カーテシアン座標の生成	-14
3-1-7 座標ファイルの出力	-14
4 . 高分子トポロジージェネレータ tplgene の関数機能説明	-16

1. はじめに

新規 MD システム myPresto を用いてエネルギー最小化・MD シミュレーションを行う場合は、まず始めにその分子系のトポロジー情報を記述したトポロジーファイルを作成する必要があります。このトポロジーファイルを作成するサブシステムとして、以下の 2 つのトポロジーファイル作成プログラム（トポロジージェネレータ）を開発している。

- ・高分子トポロジージェネレータ : tplgene
タンパク質や DNA、RNA 等の生体高分子用トポロジージェネレータ
- ・低分子トポロジージェネレータ : low_tplgene
リガンド分子等の低分子用トポロジージェネレータ

高分子トポロジージェネレータ tplgene は、タンパク質や DNA、RNA 等、生体高分子のトポロジーファイルを作成することができる。トポロジーファイル作成の際には、不足水素原子についての情報を補い、myPresto 用のトポロジーファイルと初期構造のカーテシアン座標を得ることができる。

低分子トポロジージェネレータ low_tplgene は、tplgene で対応していない新規のリガンド等、低分子について、トポロジーファイルを作成することができる。

このように作成した高分子と低分子を含む複数の系のコンフォメーションエネルギー計算を行いたい場合は、tplgene や low_tplgene でそれぞれ作成したトポロジーファイル中のデータをまとめ、1 つのファイルにする。この全系のトポロジーファイルを用いて、myPresto によるシミュレーションを行うことができる。

以下に高分子、低分子トポロジージェネレータの機能について示す。

<高分子トポロジージェネレータ>

高分子のトポロジー作成

与えられた原子の位置 (PDB 形式) とアミノ酸結合情報、結合角情報、二面角情報等の情報ファイルと照合し、各アミノ酸配列・核酸配列のトポロジー情報を作成する。その際、PDB 形式ファイルに含まれる分子には水素原子等を省略している場合がある。このように原子を省略している場合に、分子のライブラリに基づき、省略された原子を自動生成する。

また、初期内部回転角情報 (DIHED フォーマット) を伴うアミノ酸配列を与えられた場合にも、高分子のトポロジー情報を作成する。その際、システイン側鎖同士が酸化して共有結合を形成し、シスチンとなった場合にも対応する。

高分子力場パラメータ設定

で作成した高分子のトポロジー情報を元に分子内のポテンシャルパラメータを設定する。設定するポテンシャルパラメータは以下のものとする。

1)atom 2)charge 3)bond 4)angle 5)torsion 6)improper torsion 7)nonbond

<低分子トポロジージェネレータ>

低分子のトポロジー作成

与えられた原子の位置 (Z-matrix 情報) と電荷情報、結合次数情報から、各分子のトポロジー情報を作成する。

低分子分子力場パラメータ設定

分子のトポロジー情報を元に分子内のポテンシャルパラメータを設定する。設定するポテンシャル関数は AMBER : parm99、及び、AMBER : General Amber Force Field(GAFF)とする。

<データの取りまとめ>

全体の分子のトポロジーと座標生成

生体高分子とリガンド分子のシミュレーション等を行う場合は、高分子、低分子トポロジージェネレータで作成したトポロジー情報を1つにまとめ複合体のトポロジーファイルとし、同様に、座標情報を1つにまとめ、複合体の座標ファイルとする。

<MD 計算実行>

myPresto

上記 ~ で作成したトポロジー情報、座標情報、力場パラメータ情報を元に myPresto によるコンフォメーションエネルギー計算を行う。

2-2 高分子トポロジージェネレータ tplgene のシステム概要

以下の高分子トポロジージェネレータ tplgene のシステム構成を示す。

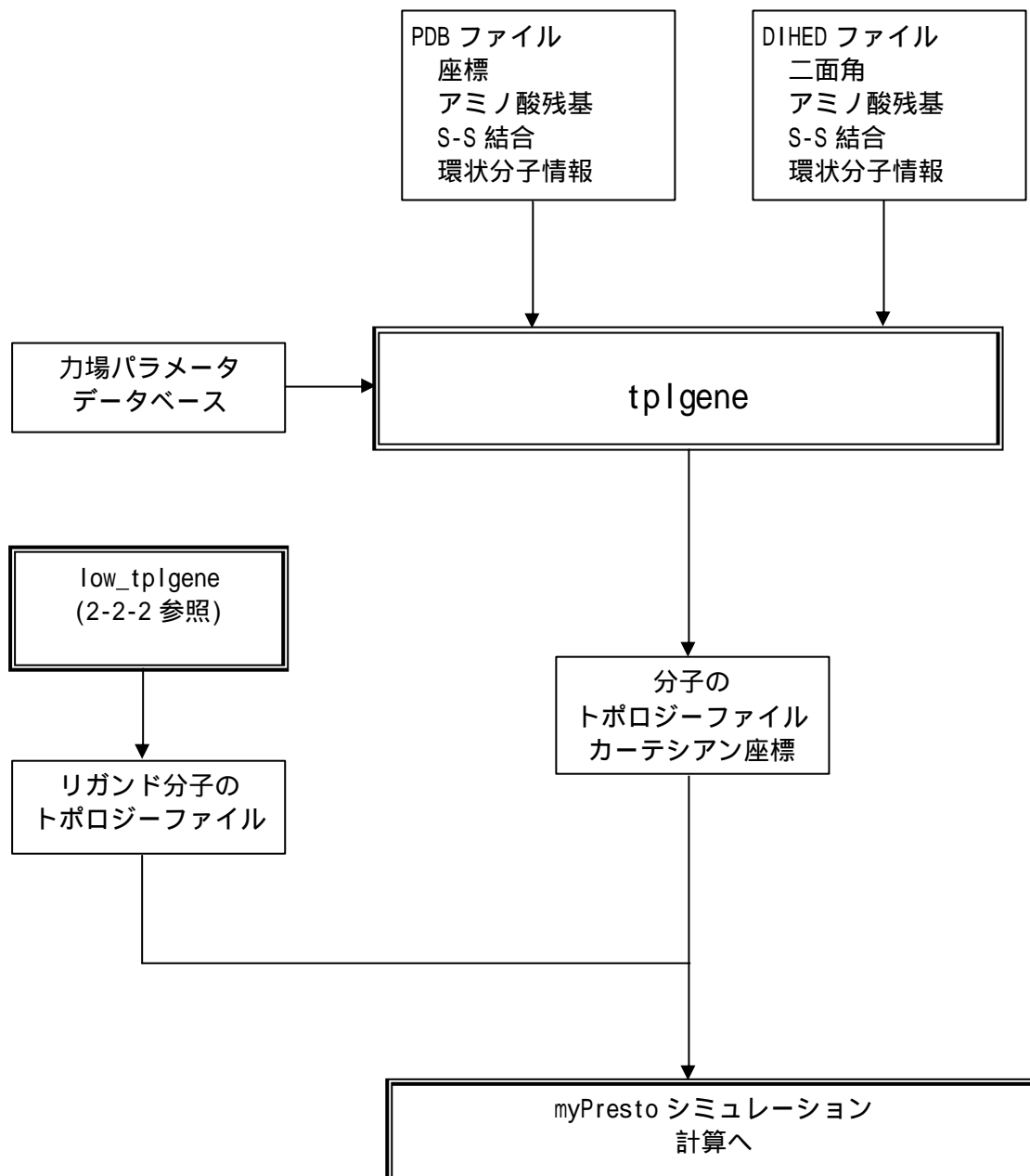


図 2 - 2 . tplgene のシステム構成

3 . トポロジージェネレータの処理

トポロジージェネレータは新規 MD システム myPresto 用のトポロジーファイルを作成するシステムである。以下に、高分子、及び、低分子トポロジージェネレータの処理の流れを示す。

3-1 高分子トポロジージェネレータ tplgene の処理

タンパク質やDNA、RNA等の生体高分子に対しては高分子トポロジージェネレータ tplgene を用いてトポロジーファイルを作成する。tplgene は以下の流れで、PDB、あるいは DIHED ファイルからトポロジーと分子座標ファイルを作成する。

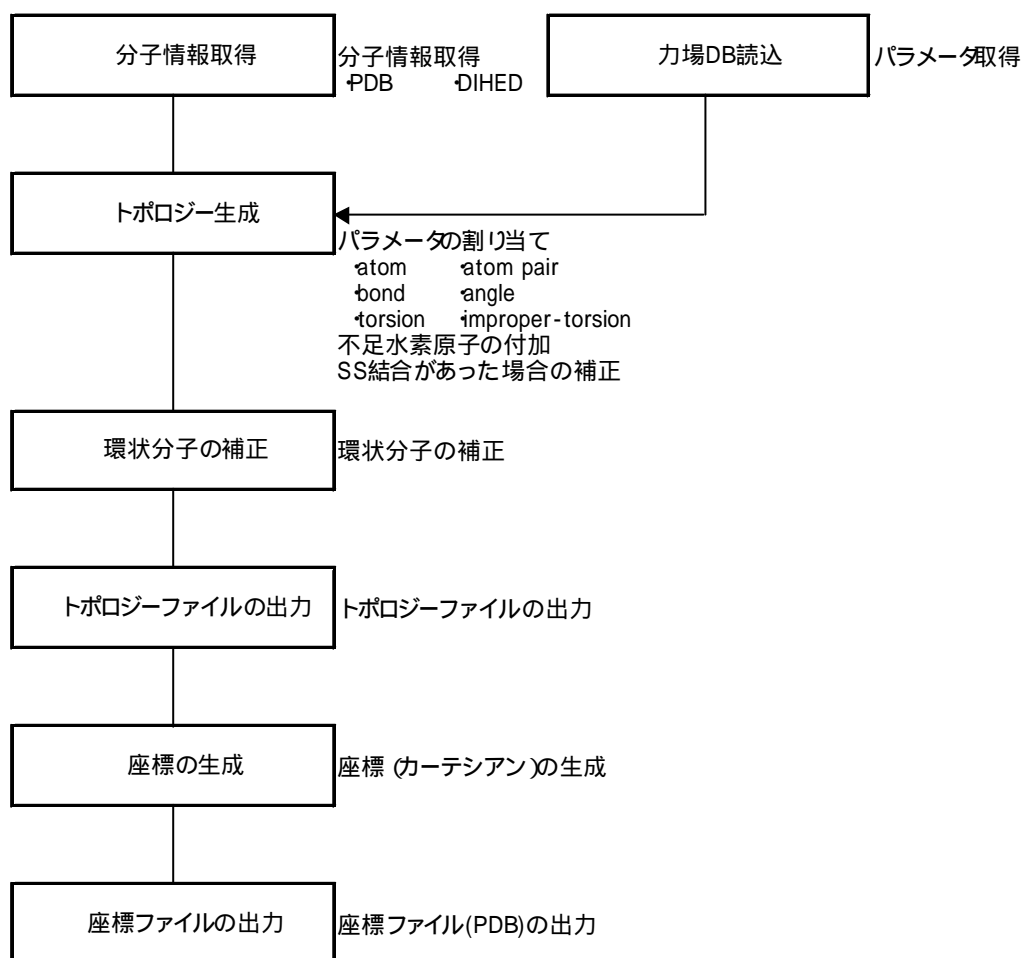


図 3 - 1 . 高分子ジェネレータ tplgene の処理概要

3-1-1 分子情報の取得

高分子トポロジージェネレータ tplgene は入力ファイルとして、PDB ファイルフォーマットと DIHED ファイルフォーマットに対応している。

(1)PDB ファイルフォーマットについて

tplgene の入力には標準的な PDB 形式ファイルに対応している。ジスルフィド結合は通常の PDB 形式に従って定義される。環状分子 (N 末と C 末が結合している分子) を計算する場合には、ATOM 行よりも前の行に、CIRCLE のキーワードを指定する (図 3 - 2 参照)。

PDB ファイル中に複数鎖の情報が含まれている場合には、それらすべての分子について計算を行う。

```
SSBOND  1 CYS A   6   CYS A  11
CIRCLE
ATOM    20  N   GLU A   4   33.037 -5.952 10.469
ATOM    21  CA  GLU A   4   33.629 -7.247 10.859
ATOM    22  C   GLU A   4   32.721 -7.845 11.909
ATOM    23  O   GLU A   4   32.470 -9.061 11.856
ATOM    24  CB  GLU A   4   35.029 -7.100 11.439
ATOM    25  CG  GLU A   4   36.081 -6.452 10.545
ATOM    26  CD  GLU A   4   35.906 -5.028 10.096
ATOM    27  OE1 GLU A   4   35.591 -4.102 10.842
ATOM    28  OE2 GLU A   4   36.158 -4.867  8.851
TER
```

図 3 - 2 PDB 入力ファイル例

tplgene には標準的な PDB 形式ファイルを入力として使用するが、以下のキーワードをもとに計算を行う。

SSBOND について

```
          1          2          3          4
1234567890123456789012345678901234567890
SSBOND  1 CYS E  48   CYS E  51
SSBOND  2 CYS E 252   CYS E 285
```

SSBOND 行の以下の情報を用いてジスルフィド結合を認識する。

- ・ 16,30 カラム目の chain ID
- ・ [18-21], [32-35]カラム目の残基番号

HETATM について

現バージョンでは、tplgene は HETATM 行で記述している原子には対応していない。

ATOM について

	1	2	3	4	5	6	7	8
12345678901234567890123456789012345678901234567890123456789012345678901234567890								
ATOM	147	C VAL A	25	30.447	15.105	58.363	1.00 12.34	A1 C
ATOM	148	O VAL A	25	29.520	15.059	59.174	1.00 15.65	A1 O
ATOM	149	CB AVAL A	25	30.385	17.437	57.230	0.28 13.88	A1 C
ATOM	150	CB BVAL A	25	30.166	17.399	57.373	0.72 15.41	A1 C

ATOM 行の以下の情報を用いて計算を行う。

- ・ [13-16]カラム目の原子名
- ・ 17 カラム目の Alternate location indicator
Alternate location indicator が検出された場合は、プログラムは終了する。
(対応) 重複している原子を削除し、Alternate location indicator を削除して再計算する。
- ・ [18-20]カラム目の残基名
- ・ 22 カラム目の chainID
- ・ [23-26]カラム目の残基番号
- ・ [31-38], [39-46], [47-54]カラム目のカーテシアン座標

TER について

ATOM 行の最後に記述する。1 つの PDB ファイル中に複数の分子を記述する場合には各分子の ATOM 行の最後にそれぞれ記述する。

(2)DIHED ファイルフォーマットについて

タンパク質 1 分子のカーテシアン座標を生成したいときは、DIHED フォーマットファイルが有効である。このフォーマットではアミノ酸残基とジスルフィド結合情報、環状分子情報を記述するだけで使用できる。二面角情報を記述しない場合は、伸びた鎖構造を生成する。二面角情報を記述した場合は、それらの値に従った鎖構造を生成する。

PDB 中アミノ酸残基は以下のキーワードを入力する。環状分子を計算する場合は 1 行目に以下を記述する。

PRE>CIRCULAR

: この分子が環状分子であることを示す。

続いて、ジスルフィド結合を有する場合には、PRE>SSBOND を指定し、次の行以降に、ジスルフィド結合している残基番号の組を記述する。以下のように記述する。

PRE>SSBOND

3 9
20 51

: 1 行目、この分子が SS 結合を持つことを示す。
: 3 番と 9 番の残基が SS 結合している。
: 20 番と 51 番の残基が SS 結合している。

次に、アミノ酸残基情報を記述する。キーワード 'PRE>SEQUENCE' の次の行から N 末端側から順次、アミノ酸名を入力していく。1 行に 1 つのアミノ酸名を記述する。

PRE>SEQUENCE

最後に二面角情報を記述する。PRE> DIHEDRAL-ANGLES で指定し、次の行以降に、N 末

端から C 末端までの、 ϕ 、 ψ 、 ω 、 χ_1 、 χ_2 ... の順番に値を記述する。1 行ごとに 10 個以内の角度値を記述する。

```
PRE>DIHEDRAL-ANGLES
180 180 180 0
180 180 180 0
180 180 180 0
```

：以下に二面角の角度情報記述することを示す。

1~3 残基目の ϕ 、 ψ をそれぞれ 180 と指定している。

例として、DODECA-PEPTIDE の DIHED ファイルは以下のように記述される。このペプチド鎖は 12 残基からなり、3、9 残基目の CYS-CYS 間でジスルフィド結合している。

```
PRE>SSBONDS
3 9
PRE>SEQUENCE
```

ASP	: 1
LYS	: 2
CYS	: 3 -----+
CYS	: 4
HIS	: 5
HIS	: 6 S-S BRIDGE
LEU	: 7
TRP	: 8
CYS	: 9 -----+
GLN	: 10
GLU	: 11
GLU	: 12

PDB、DIHED フォーマット共に計算を行う際に内部で以下の処理を行う。

- ‘+’；プロトン化された LYS、ARG、HIS の場合に残基名の後ろに付加する。
- ‘-’；非プロトン化された ASP、GLU の場合に付加する。
- ‘E’；HIS の中で、AN HD 水素のかわりに、AN HE 水素を持つもの場合に付加する。
- ‘S’；CYS の中で、ジスルフィド結合をつくるもの場合に付加する。

N および C 末端には、以下の処理を行う。

- ‘N+’；プロトン化された、N 末端の場合
- ‘N ’’；ニュートラルな、N 末端の場合
- ‘C-’；非プロトン化された、C 末端の場合
- ‘C ’’；ニュートラルな、C 末端の場合

現バージョンでは、以下の処理を自動的に行う。

- N 末端の場合は、‘N+’を自動的付加する。
- C 末端の場合は、‘C-’を自動的付加する。
- LYS、ARG の場合は、‘+’を自動的付加する。
- ASP、GLU の場合は、‘-’を自動的付加する。
- ジスルフィド結合している CYS に対しては、‘S’を自動的付加する。

3-1-2 カ場パラメータ DB 読込

アミノ酸や核酸の残基毎の以下に示すトポロジー情報を読み込む。

- atom 情報
- bond 情報
- angle 情報
- torsion 情報
- improper-torsion 情報
- Z-matrix 座標情報

分子力学プログラム AMBER と CHARMM のカ場パラメータに対応する。

3-1-3 トポロジー生成

3-1-2 で読み込んだカ場パラメータ情報を各残基のトポロジー情報をもとにトポロジーを生成し、以下のパラメータを割り当てる。

- atom 情報
- bond 情報
- angle 情報
- torsion 情報
- improper-torsion 情報
- Z-matrix 座標情報

この際、分子内、分子間にジスルフィド結合があった場合の補正を行う。

3-1-4 環状分子の補正

計算する分子が(N 末と C 末が繋がっている)環状分子については、N 末-C 末の連結部分の補正を行う。

3-1-5 トポロジーファイルの出力

作成したトポロジーファイルを出力する。

3-1-6 カートesian座標の生成

入力が DIHED ファイルフォーマットであった場合、カ場パラメータ DB から取得した Z-matrix 情報をもとに分子のカーテシアン座標を生成する。

また、入力が PDB ファイルフォーマットであった場合でも、不足している水素原子があれば、これを自動付加し、座標を計算する。

3-1-7 座標ファイルの出力

生成した座標ファイルを PDB ファイルフォーマットで出力する。

4 . 高分子トポロジージェネレータ tplgene の関数機能説明

以下に tplgene の各関数の構成図を示す。

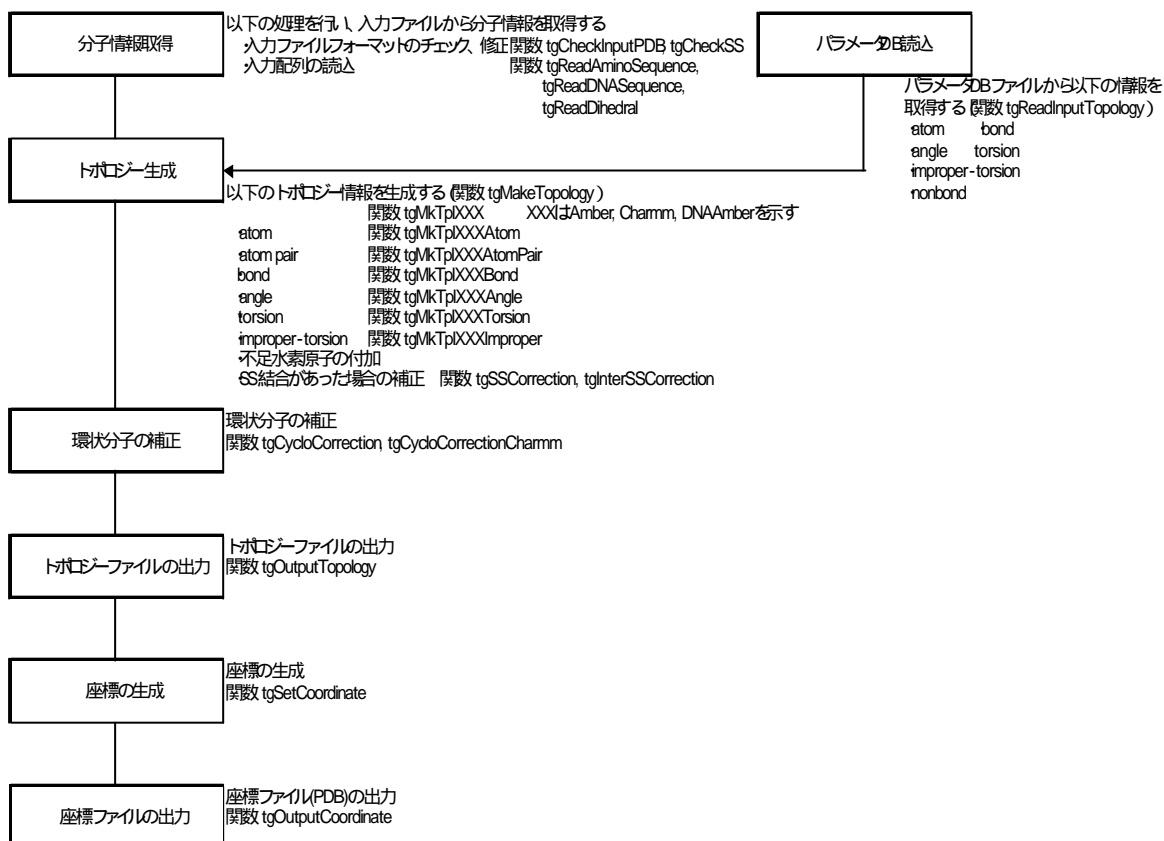


図 4 . tplgene の関数構成

4-1 main

機能概要

- tplgene のメインプログラム

機能説明

- 入力データ (タイトル、分子名、ペプチドかヌクレオチドの選択、PDB と DIHED の選択、トポロジーDB ファイル名、出力座標ファイル名、出力トポロジーファイル名) を環境ファイルから読み込む、あるいは対話式に各データを入力する。(関数 tgReadEnv を呼び出す)
- 座標ファイル、トポロジーDB ファイルを入力する。(関数 tgReadInputSequence, tgReadInputTopology を呼び出す)
- 分子内トポロジーを作成する。(関数 tgMakeTopology を呼び出す)
- 環状ペプチドの処理をする。(関数 tgCycloCorrection, tgCycloCorrectionCharmm を呼び出す)
- トポロジーファイルの出力をする。(関数 tgOutputTopology を呼び出す)
- 全原子の Cartesian 座標の創生をする。(関数 tgSetCoordinate を呼び出す)
- 全原子の Cartesian 座標の出力をする。(関数 tgOutputCoordinate を呼び出す)

引数

```
int argc  
char *argv[]
```

備考

なし。

4-2 tgReadEnv

機能概要

- ・ 入出力ファイル名を環境ファイルから読み込むか、対話式に入力する。

機能説明

- ・ タイトル名の読み込みを行う。
- ・ 分子名の読み込みを行う。
- ・ ペプチドかヌクレオチドの選択の読み込みを行う。
- ・ PDB 及び DIHED の選択の読み込みを行う。
- ・ 座標ファイル名の読み込みを行う。
- ・ トポロジーDB ファイル名の読み込みを行う。
- ・ トポロジーファイル名の読み込みを行う。

引数

```
int argc  
char *argv[]
```

備考

なし。

4-3 tgReadInputSequence

機能概要

- ・入力座標ファイルの読み込みを行う。

機能説明

- ・入力ファイルが PDB ファイルか DIHED ファイルかを選択する。
- ・入力する配列がペプチドかヌクレオチドかを選択する。
- ・入力ファイルが PDB ファイルの場合、PDB ファイルフォーマットのチェックを行う。
(関数 tgCheckInputPDB を呼び出す)
- ・入力ファイルが PDB ファイルの場合、SS 結合に関する補正を行う。(関数 tgCheckSS を呼び出す)
- ・入力ファイルが PDB ファイルで、その配列がペプチドの場合、入力ファイルを読み込む。(関数 tgReadAminoSequence を呼び出す)
- ・入力ファイルが DIHED ファイルの場合、入力ファイルを読み込む。(関数 tgReadDihedral を呼び出す)
- ・入力ファイルが PDB ファイルで、その配列がヌクレオチドの場合、入力ファイルを読み込む。(関数 tgReadDNASequence を呼び出す)

引数

TGSystemPtr amber

TGSystemPtr pdb

WHEN MODEC=1 THEN DEFAULT INNER-COORDINATES ARE USED,
WHEN MODEC=2 THEN GIVEN DIHEDRAL ANGLES ARE USED, AND
WHEN MODEC=3 THEN GIVEN CARTESIAN COORDINATES ARE USED

備考

なし。

4-4 tgCheckInputPDB

機能概要

- ・ PDB ファイルフォーマットのチェックを行う。

機能説明

- ・ PDB ファイルフォーマットのチェックを行う。

引数

なし。

備考

なし。

4-5 tgCheckSS

機能概要

- ・ SS 結合に関する補正を行う。

機能説明

- ・ SS 結合があった場合、原子、残基、分子の通し番号の修正を行う。
- ・ SS 結合があった場合、chainID の修正を行う。

引数

なし。

備考

なし。

4-6 tgReadAminoSequence

機能概要

- ・ アミノ酸配列を入力 PDB ファイルから読み込む。

機能説明

- ・ PDB 入力ファイルの読み込みを行う。
- ・ ATOM 行、HETA 行、CIRC 行、SSBO 行の読み込みを行う。
- ・ LYS、ARG の処理を行う。
- ・ ASP、GLU の処理を行う。
- ・ HISE、HIS+の処理を行う。
- ・ N 末端、C 末端の処理を行う。
- ・ CIRC 行のとき、環状分子のフラグを立てる。
- ・ 次の残基のメモリを確保する。(関数 tgNextResidue を呼び出す)
- ・ 次の分子のメモリを確保する。(関数 tgNextMol を呼び出す)

引数

TGASystemPtr amber
int output_flag

WHEN MODEC=1 THEN DEFAULT INNER-COORDINATES ARE USED,
WHEN MODEC=2 THEN GIVEN DIHEDRAL ANGLES ARE USED, AND
WHEN MODEC=3 THEN GIVEN CARTESIAN COORDINATES ARE USED

備考

なし。

4-7 tgNextResidue

機能概要

- ・次のアミノ酸残基のメモリ確保を行う。

機能説明

- ・次のアミノ酸残基のメモリ確保を行う。

引数

TGASystemPtr amber

備考

なし。

4-8 tgNextMol

機能概要

- ・ 次の分子のメモリ確保を行う。

機能説明

- ・ 次の分子のメモリ確保を行う。

引数

TGASystemPtr amber

備考

なし。

4-9 tgReadDihedral

機能概要

- ・ DIHED 配列を入力 DIHED ファイルから読み込む。

機能説明

- ・ シークエンス行か、DIHED 行か、SSBO 行かの分岐
- ・ シークエンス行のとき
 - ・ LYS、ARG の処理
 - ・ ASP、GLU の処理
 - ・ HISE、HIS+の処理
 - ・ N 末端、C 末端の処理
- ・ DIHED 行のとき、DIHED の読み込み
- ・ SSBO 行のとき、SS 結合のフラグを立てる。
- ・ TER 行のとき、終了処理を行う。

引数

TGASystemPtr amber

int output_flag

WHEN MODEC=1 THEN DEFAULT INNER-COORDINATES ARE USED,
WHEN MODEC=2 THEN GIVEN DIHEDRAL ANGLES ARE USED, AND
WHEN MODEC=3 THEN GIVEN CARTESIAN COORDINATES ARE USED

備考

なし。

4-10 tgReadDNASequenece

機能概要

- DNA 配列を入力 PDB ファイルから読み込む。

機能説明

- PDB ファイルの読み込みを行う。
- ATOM 行、TER 行の読み込みを行う。
 - 5' 末端の処理
 - 3' 末端の処理
- 原子識別名、塩基名、塩基番号、座標の読み込み
- 塩基名の最初が空白だったときの処理
- 原子と塩基番号の対応
- 原子と塩基名の対応
 - 1) 塩基番号が一番ならば
 - CRESNM に 5*を追加
 - 2) 塩基番号が一番でないならば
 - 塩基名の格納
 - 塩基数
- TER 行のとき、
 - 塩基数の格納
 - 最後の塩基に 3*の追加

引数

TGASystemPtr amber

int output_flag

WHEN MODEC=1 THEN DEFAULT INNER-COORDINATES ARE USED,
WHEN MODEC=2 THEN GIVEN DIHEDRAL ANGLES ARE USED, AND
WHEN MODEC=3 THEN GIVEN CARTESIAN COORDINATES ARE USED

備考

なし。

4-11 tgReadInputTopology

機能概要

- ・トポロジーDB ファイルを読み込む。

機能説明

- ・トポロジーDB ファイルのタイプ(amber、CHARMm)の選択を行う。
- ・トポロジーDB ファイルのタイプが amber の場合、関数 tgReadAmber の呼び出しを行う。
- ・トポロジーDB ファイルのタイプが CHARMm の場合、関数 tgReadCharmm の呼び出しを行う。
- ・トポロジーDB ファイルのタイプが上記以外の場合、エラー出力を行う。

引数

TGADBPtr DB

備考

なし。

4-12 tgReadAmber

機能概要

- Amber、CHARMm トポロジーDB ファイルを読み込む。

機能説明

- キーワードの読み込みを行う。
- 分子の読み込みを行う。(関数 tga_ReadMolecules を呼び出す)
- 原子の読み込みを行う。(関数 tga_ReadAtom を呼び出す)
- 結合の読み込みを行う。(関数 tga_ReadBond を呼び出す)
- ANGLE の読み込みを行う。(関数 tga_ReadAngle を呼び出す)
- TORSION の読み込みを行う。(関数 tga_ReadTorsion を呼び出す)
- IMPROPER-TORSIONS の読み込みを行う。(関数 tga_ReadImproper)
- FUNCTIONS の読み込みを行う。(関数 tga_ReadFunction を呼び出す)
- NONBONDED 相互作用パラメータの作成を行う。(関数 tga_ReadNonBond を呼び出す)

引数

TGADBPtr DB

備考

なし。

4-13 tga_ReadMolecule

機能概要

- ・ Amber、CHARMm トポロジーDB ファイルの分子情報を取得。

機能説明

- ・ キーワード PRE>MOLECULES 行の読み込みを行う。
- ・ 次の残基、塩基のメモリ確保を行う。

引数

TGADBPtr DB

備考

なし。

4-14 tga_ReadAtom

機能概要

- ・ Amber、CHARMm トポロジーDB ファイルの原子情報を取得。

機能説明

- ・ キーワード PRE>ATOMS 行の読み込みを行う。
- ・ 次の原子のメモリ確保を行う。

引数

TGADBPtr DB

int molnum

備考

なし。

4-15 tga_ReadBond

機能概要

- ・ Amber、CHARMm トポロジーDB ファイルの結合情報を取得。

機能説明

- ・ キーワード PRE>BONDS 行の読み込みを行う。
- ・ 次の結合のメモリ確保を行う。

引数

TGADBPtr DB

備考

なし。

4-16 tga_ReadAngle

機能概要

- ・ Amber、CHARMm トポロジーDB ファイルの結合角情報を取得。

機能説明

- ・ キーワード PRE>ANGLES 行の読み込みを行う。
- ・ 次の結合角のメモリ確保を行う。

引数

TGADBPtr DB

備考

なし。

4-17 tga_ReadTorsion

機能概要

- ・ Amber、CHARMm トポロジーDB ファイルの Torsion 情報を取得。

機能説明

- ・ キーワード PRE>TORSIONS 行の読み込みを行う。
- ・ 次の Torsion のメモリ確保を行う。

引数

TGADBPtr DB

備考

なし。

4-18 tga_ReadImproper

機能概要

- ・ Amber、CHARMm トポロジーDB ファイルの Improper Torsion 情報を取得。

機能説明

- ・ キーワード PRE>IMPROPER-TORSIONS 行の読み込みを行う。
- ・ 次の Improper Torsion のメモリ確保を行う。

引数

TGADBPtr DB

備考

なし。

4-19 tga_ReadFunction

機能概要

- ・ Amber、CHARMm トポロジーDB ファイルのポテンシャル関数のタイプを取得。

機能説明

- ・ キーワード PRE>FUNCTIONS 行の読み込みを行う。
- ・ 次のポテンシャル関数のメモリ確保を行う。

引数

TGADBPtr DB

備考

なし。

4-20 tga_ReadNonBond

機能概要

- ・ Amber、CHARMm トポロジーDB ファイルの non-bond パラメータを取得。

機能説明

- ・ キーワード PRE>NONBONDS 行の読み込みを行う。
- ・ 次の non-bond パラメータのメモリ確保を行う。

引数

TGADBPtr DB

備考

なし。

4-21 tgMakeTopology

機能概要

- ・トポロジーファイルを創生する。

機能説明

- ・トポロジーDB ファイルタイプの選択を行う。
- ・ペプチドの Amber トポロジーファイルを作成する。(関数 tgMkTpIAmber を呼び出す)
- ・ペプチドの CHARMM トポロジーファイルを作成する。(関数 tgMkTpICharmm を呼び出す)
- ・ヌクレオチドの Amber トポロジーファイルを作成する。(関数 tgMkTpIDNAAmber を呼び出す)

引数

TGASystemPtr amber

TGADBPtr DB

備考

なし。

4-22 tgMkTplAmber

機能概要

- ・ペプチドの Amber トポロジーファイルを創生する。

機能説明

- ・原子情報の創生を行う。(関数 tgMkTplAmberAtom を呼び出す)
- ・1-3、1-4 相互作用情報の修正を行う。(関数 tgMkTplAmberAtomPair を呼び出す)
- ・結合情報の創生を行う。(関数 tgMkTplAmberBond を呼び出す)
- ・結合角情報の創生を行う。(関数 tgMkTplAmberAngle を呼び出す)
- ・二面角情報の創生を行う。(関数 tgMkTplAmberTorsion を呼び出す)
- ・IMPROPER-TORSIONS 情報の創生を行う。(関数 tgMkTplAmberImproper を呼び出す)
- ・SS 結合時の修正を行う。(関数 tgSSCorrection を呼び出す)
- ・分子間 SS 結合時の原子情報の修正を行う。(関数 tgInterSSCorrection を呼び出す)

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-23 tgMkTplAmberAtom

機能概要

- ・ペプチドの Amber トポロジーファイル中の原子の情報を創生する。

機能説明

- ・原子情報の創生を行う。
- ・2 残基目以降のアミノ酸残基の N、H、CA、CH3 原子について、原子座標情報の修正を行う
- ・2 残基目以降のアミノ酸残基名に N+がついていた場合、分子間 SS 結合のフラグを立てる。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-24 tgMkTplAmberAtomPair

機能概要

- ・ペプチドの Amber トポロジーファイル中の原子情報中の 1-3、1-4 相互作用情報を修正する。

機能説明

- ・CA、O 原子の場合、1-4 相互作用情報を修正する。
 - ・次のアミノ酸残基が PRO である場合の処理を行う。
 - ・次のアミノ酸残基が PRO でない場合の処理を行う。
- ・C 原子の場合、1-3、1-4 相互作用情報を修正する。
 - ・次のアミノ酸残基が PRO である場合の処理を行う。
 - ・次のアミノ酸残基が PRO でない場合の処理を行う。

引数

 TGASystemPtr amber

備考

なし。

4-25 tgMkTplAmberBond

機能概要

- ・ペプチドの Amber トポロジーファイル中の結合情報を創生する。

機能説明

- ・結合情報の創生を行う。
- ・分子間 SS 結合がある場合、SS 結合している C 末端側のアミノ酸残基に関わる原子の結合情報を修正する。
- ・分子間 SS 結合がない場合、2 残基目以降のアミノ酸残基に関わる結合情報のうち、1 つ前の残基と相互作用する部分について修正を行う。
- ・分子間 SS 結合がある場合、そのアミノ酸残基の結合情報数の修正を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-26 tgMkTplAmberAngle

機能概要

- ・ペプチドの Amber トポロジーファイル中の結合角情報を創生する。

機能説明

- ・結合角情報の創生を行う。
- ・分子間 SS 結合がある場合、SS 結合している C 末端側のアミノ酸残基に関わる原子の結合角情報を修正する。
- ・分子間 SS 結合がない場合、2 残基目以降のアミノ酸残基に関わる結合角情報のうち、1 つ前の残基と相互作用する部分について修正を行う。
- ・分子間 SS 結合がある場合、そのアミノ酸残基の結合角情報数の修正を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-27 tgMkTplAmberTorsion

機能概要

- ・ペプチドの Amber トポロジーファイル中の二面角情報を創生する。

機能説明

- ・二面角情報の創生を行う。
- ・分子間 SS 結合がある場合、SS 結合している C 末端側のアミノ酸残基に関わる原子の二面角情報を修正する。
- ・分子間 SS 結合がない場合、2 残基目以降のアミノ酸残基に関わる二面角情報のうち、1 つ前の残基と相互作用する部分について修正を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-28 tgMkTplAmberImproper

機能概要

- ペプチドの Amber トポロジーファイル中の Improper Torsion 情報を創生する。

機能説明

- Improper Torsion 情報の創生を行う。
- 分子間 SS 結合がある場合、SS 結合している C 末端側のアミノ酸残基に関わる原子の Improper Torsion 情報を修正する。
- 分子間 SS 結合がない場合、2 残基目以降のアミノ酸残基に関わる Improper Torsion 情報のうち、1 つ前の残基と相互作用する部分について修正を行う。
- 分子間 SS 結合がある場合、そのアミノ酸残基の Improper Torsion 情報数の修正を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-29 tgSSCorrection

機能概要

- Amber トポロジーDB ファイルを用いて、SS 結合分子の計算を行う場合、原子、結合、結合角、二面角の各情報の修正を行う。

機能説明

- SS 結合時の原子情報の修正を行う。(関数 tgSSAtomCorrection を呼び出す)
- SS 結合時の結合情報の修正を行う。(関数 tgSSBondCorrection を呼び出す)
- SS 結合時の結合角情報の修正を行う。(関数 tgSSAngleCorrection を呼び出す)
- SS 結合時の二面角情報の修正を行う。(関数 tgSSTorsionCorrection を呼び出す)

引数

TGASystemPtr amber

備考

なし。

4-30 tgSSAtomCorrection

機能概要

- Amber トポロジーDB ファイルを用いて、SS 結合を有する分子の計算を行う場合、原子情報の修正を行う。

機能説明

- SS 結合しているアミノ酸残基を見つける。
- SS 結合している残基で、N 末端側の残基について、CA、CB、HB2、HB3、SG 原子について、1-3、1-4 相互作用の数、及び、相互作用する原子の通し番号を修正する。
- SS 結合している残基で、C 末端側の残基について、CA、CB、HB2、HB3、SG 原子について 1-2、1-3、1-4 相互作用の数を修正する。

引数

TGASystemPtr amber

備考

なし。

4-31 tgSSBondCorrection

機能概要

- Amber トポロジーDB ファイルを用いて、SS 結合を有する分子の計算を行う場合、結合情報の修正を行う。

機能説明

- SS 結合しているアミノ酸残基を見つける。
- SS 結合している N 末端側のアミノ酸残基がそれぞれ、CYSS、CYSSN+、CYSSC-の場合の処理を行う。
- SS 結合している残基で、N 末端側の残基の SG 原子が結合している SG 原子の番号を修正する。
- SS 結合している残基で、C 末端側の残基について、1-2 相互作用の数を修正する。

引数

TGASystemPtr amber

備考

なし。

4-32 tgSSAngleCorrection

機能概要

- Amber トポロジーDB ファイルを用いて、SS 結合を有する分子の計算を行う場合、結合角情報の修正を行う。

機能説明

- SS 結合しているアミノ酸残基を見つける。
- SS 結合している N 末端側のアミノ酸残基がそれぞれ、CYSS、CYSSN+、CYSSC-の場合の処理を行う。
- SS 結合している残基で、N 末端側の残基中の原子と C 末端側とで、1-3 相互作用している部分について、1-3 相互作用する原子の番号の修正を行う。
- SS 結合している残基で、C 末端側の残基について、1-3 相互作用の数を修正する。

引数

TGASystemPtr amber

備考

なし。

4-33 tgSSTorsionCorrection

機能概要

- Amber トポロジーDB ファイルを用いて、SS 結合を有する分子の計算を行う場合、二面角情報の修正を行う。

機能説明

- SS 結合しているアミノ酸残基を見つける。
- SS 結合している N 末端側のアミノ酸残基がそれぞれ、CYSS、CYSSN+、CYSSC-の場合の処理を行う。
- SS 結合している残基で、N 末端側の残基中の原子と C 末端側で 1-4 相互作用しているものについて、1-4 相互作用する原子の番号を修正する。
- SS 結合している残基で、C 末端側の残基について、1-4 相互作用の数を修正する。

引数

TGASystemPtr amber

備考

なし。

4-34 tgInterSSCorrection

機能概要

- ・ Amber トポロジーDB ファイルを用いて、計算分子中に分子間 SS 結合がある分子を計算する場合、SS 結合しているアミノ酸残基部分の原子情報の修正を行う。

機能説明

- ・ 分子間 SS 結合しているアミノ酸残基ペアを見つける。
- ・ 分子間 SS 結合があるならば、結合部分の残基の原子情報を修正する。(関数 tgInterSSAtomCorrection を呼び出す)

引数

TGASystemPtr amber

備考

なし。

4-35 tgInterSSAtomCorrection

機能概要

- ・ Amber トポロジーDB ファイルを用いて、計算分子中に分子間 SS 結合がある分子を計算する場合、SS 結合しているアミノ酸残基部分の原子情報の修正を行う。

機能説明

- ・ C 末端アミノ酸残基中の CA、C、O 原子の、1-3、1-4 相互作用数の修正を行う。
- ・ N 末端アミノ酸残基中の N、CA 原子の、原子座標情報の修正を行う。

引数

TGASystemPtr amber

備考

なし。

4-36 tgMkTpICharmm

機能概要

- ・ CHARMM トポロジーファイルを創生する。

機能説明

- ・ 原子情報の創生を行う。(関数 tgMkTpICharmmAtom を呼び出す)
- ・ 1-3、1-4 相互作用情報の修正を行う。(関数 tgMkTpICharmmAtomPair を呼び出す)
- ・ 結合情報の創生を行う。(関数 tgMkTpICharmmBond を呼び出す)
- ・ 結合角情報の創生を行う。(関数 tgMkTpICharmmAngle を呼び出す)
- ・ 二面角情報の創生を行う。(関数 tgMkTpICharmmTorsion を呼び出す)
- ・ IMPROPER-TORSIONS 情報の創生を行う。(関数 tgMkTpICharmmImproper を呼び出す)
- ・ SS 結合時の修正を行う。(関数 tgSSCorrectionCharmm を呼び出す)
- ・ 分子間 SS 結合時の原子情報の修正を行う。(関数 tgInterSSCorrectionCharmm を呼び出す)

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-37 tgMkTplCharmmAtom

機能概要

- ・ CHARMM トポロジーファイル中の原子の情報を創生する。

機能説明

- ・ 原子情報の創生を行う。
- ・ 2 残基目以降のアミノ酸残基の N、H、CA、CH3 原子について、原子座標情報の修正を行う
- ・ 2 残基目以降のアミノ酸残基名に N+がついていた場合、分子間 SS 結合のフラグを立てる。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-38 tgMkTplCharmmAtomPair

機能概要

- ・ CHARMM トポロジーファイル中の原子の 1-3、1-4 相互作用情報を修正する。

機能説明

- ・ CA、O 原子の場合、1-4 相互作用情報を修正する。
 - ・ 次のアミノ酸残基が PRO である場合の処理を行う。
 - ・ 次のアミノ酸残基が PRO でない場合の処理を行う。
- ・ C 原子の場合、1-3、1-4 相互作用情報を修正する。
 - ・ 次のアミノ酸残基が PRO である場合の処理を行う。
 - ・ 次のアミノ酸残基が PRO でない場合の処理を行う。

引数

 TGASystemPtr amber

備考

なし。

4-39 tgMkTplCharmmBond

機能概要

- ・ CHARMM トポロジーファイル中の結合情報を創生する。

機能説明

- ・ 結合情報の創生を行う。
- ・ 分子間 SS 結合がある場合、SS 結合している C 末端側のアミノ酸残基に関わる原子の結合情報を修正する。
- ・ 分子間 SS 結合がない場合、2 残基目以降のアミノ酸残基に関わる結合情報のうち、1 つ前の残基と相互作用する部分について修正を行う。
- ・ 分子間 SS 結合がある場合、そのアミノ酸残基の結合情報数の修正を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-40 tgMkTpICharmAngle

機能概要

- ・ CHARMM トポロジーファイル中の結合角情報を創生する。

機能説明

- ・ 結合角情報の創生を行う。
- ・ 分子間 SS 結合がある場合、SS 結合している C 末端側のアミノ酸残基に関わる原子の結合角情報を修正する。
- ・ 分子間 SS 結合がない場合、2 残基目以降のアミノ酸残基に関わる結合角情報のうち、1 つ前の残基と相互作用する部分について修正を行う。
- ・ 分子間 SS 結合がある場合、そのアミノ酸残基の結合角情報数の修正を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-41 tgMkTpICharmmTorsion

機能概要

- ・ CHARMM トポロジーファイル中の二面角情報を創生する。

機能説明

- ・ 二面角情報の創生を行う。
- ・ 分子間 SS 結合がある場合、SS 結合している C 末端側のアミノ酸残基に関わる原子の二面角情報を修正する。
- ・ 分子間 SS 結合がない場合、2 残基目以降のアミノ酸残基に関わる二面角情報のうち、1 つ前の残基と相互作用する部分について修正を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-42 tgMkTpICharmmImproper

機能概要

- CHARMM トポロジーファイル中の Improper Torsion 情報を創生する。

機能説明

- Improper Torsion 情報の創生を行う。
- 分子間 SS 結合がある場合、SS 結合している C 末端側のアミノ酸残基に関わる原子の Improper Torsion 情報を修正する。
- 分子間 SS 結合がない場合、2 残基目以降のアミノ酸残基に関わる Improper Torsion 情報のうち、1 つ前の残基と相互作用する部分について修正を行う。
- 分子間 SS 結合がある場合、そのアミノ酸残基の Improper Torsion 情報数の修正を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-43 tgSSCorrectionCharmm

機能概要

- CHARMM のトポロジーDB ファイルを用いて、SS 結合を有する分子の計算を行う場合、原子、結合、結合角、二面角の各情報の修正を行う。

機能説明

- SS 結合時の原子情報の修正を行う。(関数 tgSSAtomCorrectionCharmm を呼び出す)
- SS 結合時の結合情報の修正を行う。(関数 tgSSBondCorrectionCharmm を呼び出す)
- SS 結合時の結合角情報の修正を行う。(関数 tgSSAngleCorrectionCharmm を呼び出す)
- SS 結合時の二面角情報の修正を行う。(関数 tgSSTorsionCorrectionCharmm, tgSSTorsionCorrectionCharmm19 を呼び出す)

引数

TGASystemPtr amber

備考

なし。

4-44 tgSSAtomCorrectionCharmm

機能概要

- CHARMM トポロジーDB ファイルを用いて、SS 結合を有する分子の計算を行う場合、原子情報の修正を行う。

機能説明

- SS 結合しているアミノ酸残基を見つける。
- SS 結合している残基で、N 末端側の残基について、CA、CB、HB2、HB3、SG 原子について、1-3、1-4 相互作用の数、及び、相互作用する原子の通し番号を修正する。
- SS 結合している残基で、C 末端側の残基について、CA、CB、HB2、HB3、SG 原子について 1-2、1-3、1-4 相互作用の数を修正する。

引数

TGASystemPtr amber

備考

なし。

4-45 tgSSBondCorrectionCharmm

機能概要

- ・ CHARMM トポロジーDB ファイルを用いて、SS 結合を有する分子の計算を行う場合、結合情報の修正を行う。

機能説明

- ・ SS 結合しているアミノ酸残基を見つける。
- ・ SS 結合している N 末端側のアミノ酸残基がそれぞれ、CYSS、CYSSN+、CYSSC-の場合の処理を行う。
- ・ SS 結合している残基で、N 末端側の残基の SG 原子が結合している SG 原子の番号を修正する。
- ・ SS 結合している残基で、C 末端側の残基について、1-2 相互作用の数を修正する。

引数

TGASystemPtr amber

備考

なし。

4-46 tgSSAngleCorrectionCharmm

機能概要

- ・CHARMm トポロジーDB ファイルを用いて、SS 結合を有する分子の計算を行う場合、結合角情報の修正を行う。

機能説明

- ・SS 結合しているアミノ酸残基を見つける。
- ・SS 結合している N 末端側のアミノ酸残基がそれぞれ、CYSS、CYSSN+、CYSSC-の場合の処理を行う。
- ・SS 結合している残基で、N 末端側の残基中の原子と C 末端側とで、1-3 相互作用している部分について、1-3 相互作用する原子の番号の修正を行う。
- ・SS 結合している残基で、C 末端側の残基について、1-3 相互作用の数を修正する。

引数

TGASystemPtr amber

備考

なし。

4-47 tgSSTorsionCorrectionCharmm

機能概要

- CHARMM22 トポロジーDB ファイルを用いて、SS 結合を有する分子の計算を行う場合、二面角情報の修正を行う。

機能説明

- CHARMM22 トポロジーDB ファイルを使用している場合に、以下の処理を行う。
- SS 結合しているアミノ酸残基を見つける。
- SS 結合している N 末端側のアミノ酸残基がそれぞれ、CYSS、CYSSN+、CYSSC-の場合の処理を行う。
- SS 結合している残基で、N 末端側の残基中の原子と C 末端側で 1-4 相互作用しているものについて、1-4 相互作用する原子の番号を修正する。
- SS 結合している残基で、C 末端側の残基について、1-4 相互作用の数を修正する。

引数

TGASystemPtr amber

備考

なし。

4-48 tgSSTorsionCorrectionCharmm19

機能概要

- ・ CHARMM19 トポロジーDB ファイルを用いて、SS 結合を有する分子の計算を行う場合、二面角情報の修正を行う。

機能説明

- ・ CHARMM19 トポロジーDB ファイルを使用している場合に、以下の処理を行う。
- ・ SS 結合しているアミノ酸残基を見つける。
- ・ SS 結合している N 末端側のアミノ酸残基がそれぞれ、CYSS、CYSSN+、CYSSC-の場合の処理を行う。
- ・ SS 結合している残基で、N 末端側の残基中の原子と C 末端側で 1-4 相互作用しているものについて、1-4 相互作用する原子の番号を修正する。
- ・ SS 結合している残基で、C 末端側の残基について、1-4 相互作用の数を修正する。

引数

TGASystemPtr amber

備考

なし。

4-49 tgInterSSCorrectionCharmm

機能概要

- ・CHARMm トポロジーファイルに関して、計算分子中に分子間 SS 結合がある場合、SS 結合しているアミノ酸残基部分の原子情報の修正を行う。

機能説明

- ・分子間 SS 結合しているアミノ酸残基ペアを見つける。
- ・分子間 SS 結合があるならば、結合部分の残基の原子情報を修正する。(関数 tgInterSSAtomCorrection を呼び出す)

引数

TGASystemPtr amber

備考

なし。

4-50 tgInterSSAtomCorrectionCharmm

機能概要

- ・ CHARMM トポロジーファイルに関して、計算分子中に分子間 SS 結合がある場合、SS 結合しているアミノ酸残基部分の原子情報の修正を行う。

機能説明

- ・ C 末端アミノ酸残基中の CA、C、O 原子の、1-3、1-4 相互作用数の修正を行う。
- ・ N 末端アミノ酸残基中の N、CA 原子の、座標情報の修正を行う。

引数

TGASystemPtr amber

備考

なし。

4-51 tgMkTpIDNAAmber

機能概要

- ・ヌクレオチドの Amber トポロジーファイルを創生する。

機能説明

- ・原子情報の創生および修正を行う。(関数 tgMkTpIDNAAmberAtom を呼び出す)
- ・1-3、1-4 相互作用の創生および修正を行う。(関数 tgMkTpIDNAAmberAtomPair を呼び出す)
- ・結合情報の創生および修正を行う。(関数 tgMkTpIDNAAmberBond を呼び出す)
- ・結合角情報の創生および修正を行う。(関数 tgMkTpIDNAAmberAngle を呼び出す)
- ・二面角情報の創生および修正を行う。(関数 tgMkTpIDNAAmberTorsion を呼び出す)
- ・IMPROPER-TORSIONS 情報の創生および修正を行う。(関数 tgMkTpIDNAAmberImproper を呼び出す)

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-52 tgMkTpIDNAAmberAtom

機能概要

- ・ヌクレオチドの Amber トポロジーファイル中の原子の情報を創生する。

機能説明

- ・原子情報の創生を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-53 tgMkTpIDNAAmberAtomPair

機能概要

- ・ヌクレオチドの Amber トポロジーファイル中の 1-3、1-4 相互作用情報を修正する。

機能説明

- ・ C3' 原子の場合、1-4 相互作用情報を修正する。
- ・ O3' 原子の場合、1-3、1-4 相互作用情報を修正する。

引数

TGASystemPtr amber

備考

なし。

4-54 tgMkTpIDNAAmberBond

機能概要

- ・ヌクレオチドの Amber トポロジーファイル中の結合情報を創生する。

機能説明

- ・結合情報の創生を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-55 tgMkTpIDNAAmberAngle

機能概要

- ・ヌクレオチドの Amber トポロジーファイル中の結合角情報を創生する。

機能説明

- ・結合角情報の創生を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-56 tgMkTpIDNAAmberTorsion

機能概要

- ・ヌクレオチドの Amber トポロジーファイル中の二面角情報を創生する。

機能説明

- ・二面角情報の創生を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-57 tgMkTpIDNAAmberImproper

機能概要

- ・ヌクレオチドの Amber トポロジーファイル中の Improper Torsion 情報を創生する。

機能説明

- ・ Improper Torsion 情報の創生を行う。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-58 tgCycloCorrection

機能概要

- Amber トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、原子、結合、結合角、二面角、Improper Torsion の各情報の修正を行う。

機能説明

- 原子情報の修正を行う。(関数 tgAtomZmatCorrection を呼び出す)
- 1-2、1-3、1-4 相互作用情報の修正を行う。(関数 tgAtomPairCorrection を呼び出す)
- 結合情報の修正を行う。(関数 tgBondCorrection を呼び出す)
- 結合角情報の修正を行う。(関数 tgAngleCorrection を呼び出す)
- 二面角情報の修正を行う。(関数 tgTorsionCorrection を呼び出す)
- IMPROPER-TORSIONS 情報の修正を行う。(関数 tgImproperCorrection を呼び出す)

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-59 tgAtomZmatCorrection

機能概要

- Amber トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、原子座標情報の修正を行う。

機能説明

- N 末端のアミノ酸残基の N 原子の座標情報を修正する。
- N 末端のアミノ酸残基の N 原子以外の座標情報を修正する。
- 末端以外のアミノ酸残基の N 原子の座標情報を修正する。
- C 末端のアミノ酸残基の O 原子の座標情報を修正する。

引数

TGASystemPtr amber

備考

なし。

4-60 tgAtomPairCorrection

機能概要

- Amber トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、1-2、1-3、1-4 相互作用情報の修正を行う。

機能説明

- C 末端アミノ酸残基の CA 原子の 1-4 相互作用情報の修正を行う。
- C 末端アミノ酸残基の C 原子の 1-3、1-4 相互作用情報の修正を行う。
- C 末端アミノ酸残基の O 原子の 1-4 相互作用情報の修正を行う。
- N 末端アミノ酸残基の各原子の 1-2、1-3、1-4 相互作用情報の修正を行う。

引数

TGASystemPtr amber

備考

なし。

4-61 tgBondCorrection

機能概要

- Amber トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、結合情報の修正を行う。

機能説明

- N 末端のアミノ酸残基において、結合情報の追加を行う。
- C 末端のアミノ酸残基において、結合情報の修正を行う。
- 他のアミノ酸残基において、結合数を修正する。
- 分子内、あるいは、分子間 SS 結合がある場合に、結合の数を修正する。
- SS 結合がある場合、結合情報を修正する。

引数

```
TGASystemPtr amber  
TGADBPtr amberDB  
int *icnres)
```

備考

なし。

4-62 tgAngleCorrection

機能概要

- Amber トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、結合角情報の修正を行う。

機能説明

- N 末端のアミノ酸残基において、結合角情報の修正を行う。
- C 末端のアミノ酸残基において、結合角情報の修正を行う。
- 他のアミノ酸残基において、結合角数を修正する。
- 分子内、あるいは、分子間 SS 結合がある場合に、結合角数を修正する。

引数

TGASystemPtr amber

TGADBPtr amberDB

int *icnre

備考

なし。

4-63 tgTorsionCorrection

機能概要

- Amber トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、二面角情報の修正を行う。

機能説明

- N 末端のアミノ酸残基において、二面角情報の修正を行う。
- C 末端のアミノ酸残基において、二面角情報の修正を行う。
- 他のアミノ酸残基において、二面角数を修正する。
- 分子内、あるいは、分子間 SS 結合がある場合に、二面角数を修正する。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-64 tglImproperCorrection

機能概要

- Amber トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、Improper Torsion 情報の修正を行う。

機能説明

- N 末端のアミノ酸残基において、Improper Torsion 情報の修正を行う。
- C 末端のアミノ酸残基において、Improper Torsion 情報の修正を行う。
- 他のアミノ酸残基において、Improper Torsion 数を修正する。
- 分子内、あるいは、分子間 SS 結合がある場合に、Improper Torsion 数を修正する。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-65 tgCycloCorrectionCharmm

機能概要

- ・ CHARMM トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、原子、結合、結合角、二面角、Improper Torsion の各情報の修正を行う。

機能説明

- ・ 原子情報の修正 (関数 tgAtomZmatCorrectionCharmm を呼び出す)
- ・ 1-2、1-3、1-4 相互作用情報の修正を行う。(関数 tgAtomPairCorrectionCharmm を呼び出す)
- ・ 結合情報の修正 (関数 tgBondCorrectionCharmm を呼び出す)
- ・ 結合角情報の修正 (関数 tgAngleCorrectionCharmm を呼び出す)
- ・ 二面角情報の修正 (関数 tgTorsionCorrectionCharmm, tgTorsionCorrectionCharmm19 を呼び出す)
- ・ IMPROPER-TORSIONS 情報の修正
(関数 tgImproperCorrectionCharmm, tgImproperCorrectionCharmm19 を呼び出す)

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-66 tgAtomZmatCorrectionCharmm

機能概要

- ・ CHARMM トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、原子座標情報の修正を行う。

機能説明

- ・ N 末端のアミノ酸残基の N 原子の座標情報を修正する。
- ・ N 末端のアミノ酸残基の N 原子以外の座標情報を修正する。
- ・ 末端以外のアミノ酸残基の N 原子の座標情報を修正する。
- ・ C 末端のアミノ酸残基の O 原子の座標情報を修正する。

引数

TGASystemPtr amber

備考

なし。

4-67 tgAtomPairCorrectionCharmm

機能概要

- CHARMM トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、1-2、1-3、1-4 相互作用情報の修正を行う。

機能説明

- C 末端アミノ酸残基の CA 原子の 1-4 相互作用情報の修正を行う。
- C 末端アミノ酸残基の C 原子の 1-3、1-4 相互作用情報の修正を行う。
- C 末端アミノ酸残基の O 原子の 1-4 相互作用情報の修正を行う。
- N 末端アミノ酸残基の各原子の 1-2、1-3、1-4 相互作用情報の修正を行う。

引数

TGASystemPtr amber

備考

なし。

4-68 tgBondCorrectionCharmm

機能概要

- ・ CHARMM トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、結合情報の修正を行う。

機能説明

- ・ N 末端のアミノ酸残基において、結合情報の修正を行う。
- ・ C 末端のアミノ酸残基において、結合情報の修正を行う。
- ・ 他のアミノ酸残基において、結合数を修正する。
- ・ 分子内、あるいは、分子間 SS 結合がある場合に、結合数を修正する。

引数

TGASystemPtr amber

TGADBPtr amberDB

int *icnres

備考

なし。

4-69 tgAngleCorrectionCharmm

機能概要

- CHARMM トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、結合角情報の修正を行う。

機能説明

- N 末端のアミノ酸残基において、結合角情報の修正を行う。
- C 末端のアミノ酸残基において、結合角情報の修正を行う。
- 他のアミノ酸残基において、結合角数を修正する。
- 分子内、あるいは、分子間 SS 結合がある場合に、結合角数を修正する。

引数

 TGASystemPtr amber

 TGADBPtr amberDB

 int *icnres

備考

なし。

4-70 tgTorsionCorrectionCharmm

機能概要

- CHARMM22 トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、二面角情報の修正を行う。

機能説明

- N 末端のアミノ酸残基において、二面角情報の修正を行う。
- C 末端のアミノ酸残基において、二面角情報の修正を行う。
- 他のアミノ酸残基において、二面角数を修正する。
- 分子内、あるいは、分子間 SS 結合がある場合に、二面角数を修正する。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-71 tgTorsionCorrectionCharmm19

機能概要

- CHARMM19 トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、二面角情報の修正を行う。

機能説明

- N 末端のアミノ酸残基において、二面角情報の修正を行う。
- C 末端のアミノ酸残基において、二面角情報の修正を行う。
- 他のアミノ酸残基において、二面角数を修正する。
- 分子内、あるいは、分子間 SS 結合がある場合に、二面角数を修正する。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-72 tglImproperCorrectionCharmm

機能概要

- CHARMM22 トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、Improper Torsion 情報の修正を行う。

機能説明

- N 末端のアミノ酸残基において、Improper Torsion 情報の修正を行う。
- C 末端のアミノ酸残基において、Improper Torsion 情報の修正を行う。
- 他のアミノ酸残基において、Improper Torsion 数を修正する。
- 分子内、あるいは、分子間 SS 結合がある場合に、Improper Torsion 数を修正する。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-73 tglImproperCorrectionCharmm19

機能概要

- CHARMM19 トポロジーファイル作成時で、環状分子を計算する場合に、N 末端と C 末端を結合し、Improper Torsion 情報の修正を行う。

機能説明

- N 末端のアミノ酸残基において、Improper Torsion 情報の修正を行う。
- C 末端のアミノ酸残基において、Improper Torsion 情報の修正を行う。
- 他のアミノ酸残基において、Improper Torsion 数を修正する。
- 分子内、あるいは、分子間 SS 結合がある場合に、Improper Torsion 数を修正する。

引数

TGASystemPtr amber

TGADBPtr amberDB

備考

なし。

4-74 tgOutputTopology

機能概要

- ・トポロジーファイルの出力する。

機能説明

- ・トポロジーファイルへの書き出し
 - ・タイトルの出力
 - ・分子名、分子数の出力
 - ・分子名、原子数の出力
 - ・原子名、原子タイプ名、原子タイプ番号、残基名、残基番号
質量、原子半径、部分電荷、1-2 原子数、1-3 原子数、1-4 原子数
1-2 相対原子番号、1-3 相対原子番号、1-4 相対原子番号、
相対結合の番号、相対結合角の番号、相対内部回転角の番号、
相対結合の結合距離、相対結合角の角度、相対内部回転角の角度、
原子の通し番号の出力
- ・結合の出力
 - ・結合数の出力
 - ・結合リスト、力の定数、平衡結合距離、結合の通し番号の出力
- ・結合角の出力
 - ・結合角数の出力
 - ・結合角リスト、力の定数、平衡結合角、結合角の通し番号の出力
- ・内部回転角の出力
 - ・内部回転角数の出力
 - ・内部回転角リスト、力の定数、内部回転角の総数、対称数、位相の出力
- ・インプロパー内部回転角の出力
 - ・インプロパー内部回転角数の出力
 - ・インプロパー内部回転角リスト、力の定数、総数、対称数、位相の出力
- ・ファンクションの出力
 - ・ノンボンドのファンクションの種類を出力
- ・ノンボンドの出力
 - ・ノンボンド数の出力
 - ・原子タイプダミー、関数タイプ、原子半径、エネルギーの深さ、1-4
van der Waals 力、静電相互作用のエネルギーの出力

引数

TGASystemPtr amber
TGADBPTr amberDB
TGASystemPtr pdb

備考

なし。

4-75 tgCheckCoordinate

機能概要

- ・座標のチェックを行う。

機能説明

- ・OT1 の処理を行う。
- ・OT2 の処理を行う。
- ・トポロジーファイルに存在しない原子名のチェックを行う。

引数

TGASystemPtr amber

TGASystemPtr pdb

備考

なし。

4-76 tgSetCoordinate

機能概要

- ・座標のセットを行う。

機能説明

- ・二面角をセットする。(関数 tgSetTorsion を呼び出す)
- ・原子座標を創生する。(関数 tgGenerateCoordinate を呼び出す)
- ・使用した座標系が 1 ならば二面角をセットする。
- ・使用した座標系が 2 ならば
 - 1) トーションテーブルに最初の二面角を与える
 - 2) 既に存在する値を元に他の二面角をセットする。
- ・使用した座標系が 3 ならば
 - 1) 可能ならば、与えられた座標より最初の二面角をセットする。
 - 2) 既に存在する値を元に他の二面角をセットする。

引数

TGASystemPtr pdb

TGASystemPtr amber

備考

なし。

4-77 tgSetTorsion

機能概要

- ・ Torsion のセットを行う。

機能説明

- ・ 二面角の計算
- ・ 外積の計算
- ・ 内積の計算

引数

TGASystemPtr amber

double *rtors

int *iflgtr

備考

なし。

4-78 tgGenerateCoordinate

機能概要

- ・座標の創生を行う。

機能説明

- ・回転マトリクスの計算

引数

double **rxyz

double *bat

double *xyz

備考

なし。

4-79 tgOutputCoordinate

機能概要

- ・座標の出力を行う。

機能説明

- ・使用した座標系が 0 のとき座標を出力
- ・使用した座標系が 1 のとき PDB 形式座標を出力
- ・使用した座標系が 2 のときバイナリー形式座標を出力
- ・使用した座標系が 3 のとき PDB とバイナリー形式の座標を出力

引数

TGASystemPtr amber

備考

なし。

4-80 tgError

機能概要

- ・ 各関数の各種エラー処理を行う。

機能説明

- ・ 各種エラーに対応したエラーメッセージを出力する。
- ・ プログラムを終了する。

引数

int error_code

char *buf

char *buf1

char *buf2

int num

int num1

備考

なし。

myPresto/tplgeneL システム設計書
(Topology Generator 編)

第 1.0 版

Copyright (C) 2006-2008 National Institute of Advanced Industrial Science and Technology (AIST)
Copyright (C) 2006-2008 Japan Biological Informatics Consortium (JBIC)

目次 -

1 . はじめに	- 4
2 . システム概要	- 5
2-1 高分子トポロジージェネレータと低分子トポロジージェネレータの関係	- 5
2-2 tplgeneL のシステム概要	- 7
3 . tplgeneL の処理	- 8
3-1 分子情報取得	- 9
3-1-1 tplgeneL オリジナル入力ファイル	- 9
3-1-2 Sybyl mol2 入力ファイル	-10
3-2 原子タイプ割り当て基本情報取得	-11
3-2-1 原子タイプ、デフォルト原子タイプについて	-11
3-2-2 原子タイプ割り当て基本情報取得	-11
3-3 分子トポロジー生成	-15
3-4 原子タイプ割り当て規則 DB 読み込み	-16
3-4-1 原子タイプ割り当て規則 DB のファイルフォーマットについて	-17
3-4-2 原子タイプ割り当て規則 DB の読み込み (中間情報の作成)	-21
3-5 原子タイプ割り当て	-25
3-5-1 原子タイプ割り当て条件の比較	-25
3-5-2 GAFF 原子タイプ割り当て	-26
3-5-3 AMBER parm99 原子タイプ割り当て	-30
3-6 パラメータ DB 読み込み	-35
3-6-1 パラメータ DB のファイルフォーマットについて	-35
3-6-2 nonbond 情報 DB のファイルフォーマットについて	-41
3-7 パラメータ割り当て	-43
3-7-1 デフォルトパラメータを用いてのパラメータ補填	-43
3-7-2 パラメータの動的計算による補填	-44
3-8 フラグメント DB 読み込み	-46
3-8-1 フラグメント DB のファイルフォーマットについて	-47
3-9 フラグメント DB パラメータ割り当て	-53
3-10 トポロジーファイル出力	-55
3-11 PDB ファイル出力	-55
4 . tplgeneL 関数機能説明	-56
5 . 低分子トポロジージェネレータ用入力ファイル作成ツール ReadGamess	-162
付録 . 原子タイプ割り当て規則ファイル (抜粋)	-167

1. はじめに

新規 MD システム myPresto を用いてエネルギー最小化・MD シミュレーションを行う場合は、まず始めにその分子系のトポロジー情報を記述したトポロジーファイルを作成する必要があります。このトポロジーファイルを作成するサブシステムとして、以下の 2 つのトポロジーファイル作成プログラム (トポロジージェネレータ) を開発している。

- ・高分子トポロジージェネレータ : tplgene
タンパク質や核酸等の生体高分子用トポロジージェネレータ
- ・低分子トポロジージェネレータ : tplgeneL
リガンド分子等の低分子用トポロジージェネレータ

高分子トポロジージェネレータ tplgene は、タンパク質や核酸等の生体高分子のトポロジーファイルを作成することができる。トポロジーファイル作成の際には、不足水素原子についての情報を補い、myPresto 用のトポロジーファイルと初期構造のカーテシアン座標を得ることができる。

低分子トポロジージェネレータ tplgeneL は、tplgene で対応していない新規のリガンド等の低分子について、トポロジーファイルを作成することができる。

このように作成した高分子と低分子を含む複数の系のコンフォメーションエネルギー計算を行いたい場合は、tplgene や tplgeneL でそれぞれ作成したトポロジーファイル中のデータをまとめ、1 つのファイルにする。この全系のトポロジーファイルを用いて、myPresto によるシミュレーションを行うことができる。

2. システム概要

2-1 高分子トポロジージェネレータと低分子トポロジージェネレータの関係

本システムは高分子、低分子トポロジージェネレータの2つのサブシステムで構成している。高分子トポロジージェネレータ(tplgene)は、タンパク質や、核酸等の生体高分子についてトポロジーファイル、座標ファイルを作成し、低分子トポロジージェネレータ(tplgeneL)はリガンド等低分子のトポロジーファイルを作成する。以下に両者の関係を示す。

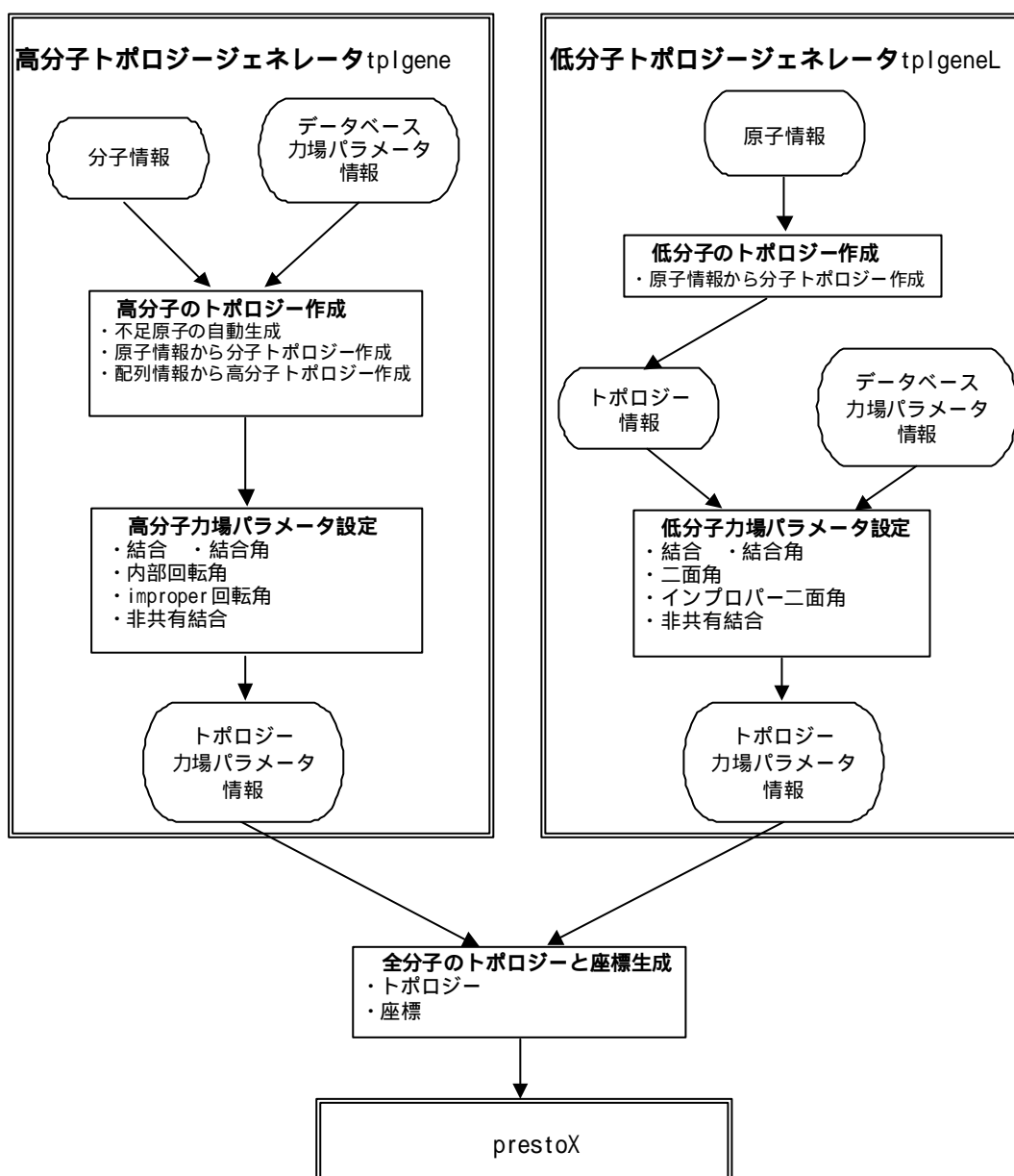


図 2 - 1 各機能の関係

以下に高分子、低分子トポロジージェネレータの機能について示す。

< tplgene >

高分子のトポロジー作成

与えられた原子の位置 (PDB 形式) とアミノ酸結合情報、結合角情報、二面角情報等の情報ファイルと照合し、各アミノ酸配列・核酸配列のトポロジー情報を作成する。

高分子力場パラメータ設定

で作成した高分子のトポロジー情報を元に分子内のポテンシャルパラメータを設定する。設定するポテンシャルパラメータは以下のものとする。

1)atom 2)charge 3)bond 4)angle 5)torsion 6)improper torsion 7)nonbond

< tplgeneL >

低分子のトポロジー作成

与えられた原子の位置 (Z-matrix 情報) と電荷情報、結合次数情報から、各分子のトポロジー情報を作成する。

低分子分子力場パラメータ設定

分子のトポロジー情報を元に分子内のポテンシャルパラメータを設定する。設定するポテンシャル関数は AMBER : General Amber Force Field(GAFF)、及び、AMBER : parm99 とする。

< データの取りまとめ >

全体の分子のトポロジーと座標生成

生体高分子とリガンド分子のシミュレーション等を行う場合は、高分子、低分子トポロジージェネレータで作成したトポロジー情報を 1 つにまとめ複合体のトポロジーファイルとし、同様に、座標情報を 1 つにまとめ、複合体の座標ファイルとする。

< MD 計算実行 >

myPresto

上記 ~ で作成したトポロジー情報、座標情報、力場パラメータ情報を元に myPresto によるコンフォメーションエネルギー計算を行う。

2-2 tplgeneL のシステム概要

以下に、tplgeneL の構成を示す。

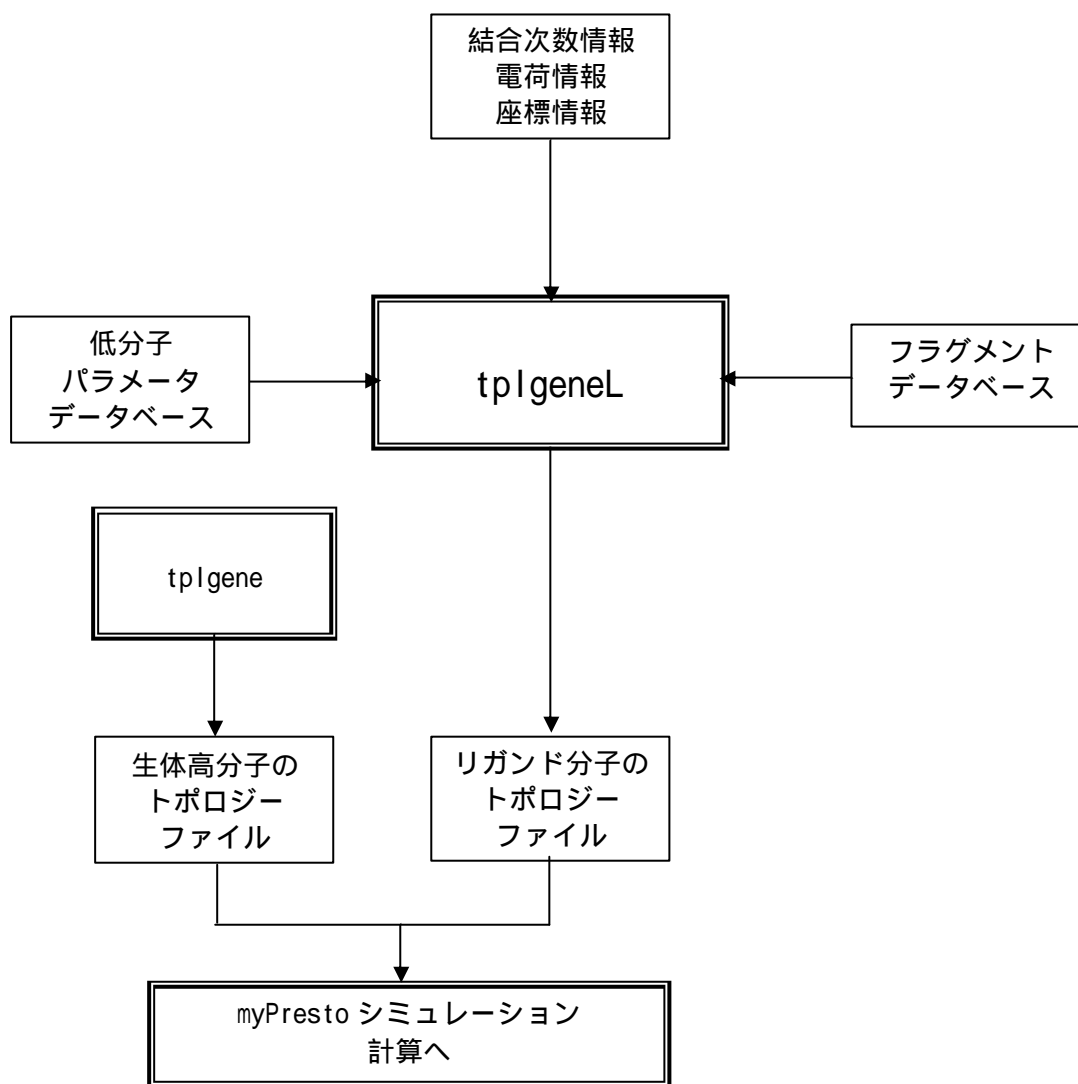


図 2 - 2 . tplgeneL のシステム構成

3 . tplgeneL の処理

tplgene では計算できないリガンド等の低分子については、tplgeneL を使用してトポロジーファイルを作成する。tplgeneL の処理として、量子化学計算を行った後、その結果から必要な情報を抽出し、それらの情報を用いてトポロジーファイルを作成する。また、tplgeneL はより一般的なファイル形式として Sybyl mol2 ファイル形式の入力についても対応する。tplgeneL の処理の概略は以下の図 (図 3 - 1) に示す。

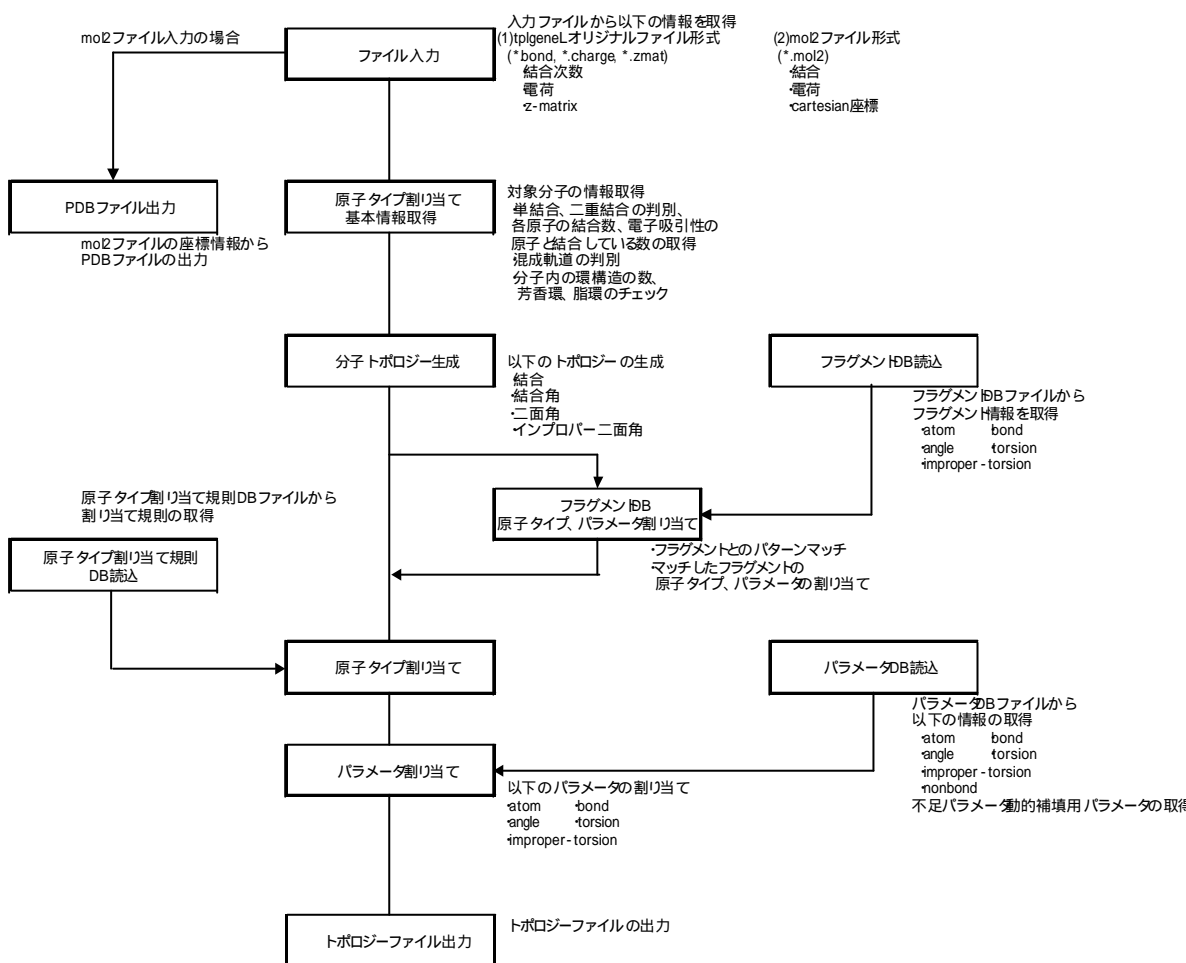


図 3 - 1 . 低分子トポロジージェネレータ tplgeneL の処理概要

3-1 分子情報取得

tplgeneL は、電荷情報、結合情報、座標情報を入力とする。現在、tplgeneL は以下の二つのファイル形式に対応している。

- ・ tplgeneL オリジナルファイル形式
- ・ Sybyl mol2 ファイル形式

3-1-1 tplgeneL オリジナル入力ファイル

電荷情報、結合情報、座標(Z-matrix)情報ファイルをもとに計算を行う。

(1) 電荷情報ファイル(XXX.charge、XXX は拡張子を除くファイル名、以下同様)について

電荷情報を原子名、元素記号、Mulliken 電荷、RESP 電荷、原子タイプ指定フラグ(s/j)、原子タイプの順に指定する。原子名、元素記号、Mulliken 電荷は必須とする。

C1 ^{†1}	C ^{†2}	0.7405 ^{†3}	0.780592 ^{†4}	s ^{†5}	c ^{†6}
N2	N	-0.7807	-0.716801	s	n
C3	C	-0.0909	-0.180378	j	
C4	C	-0.2982	-0.106306	j	
C5	C	0.0017	-0.108226	s	ca

†1:原子名(必須)
†2:元素記号(必須)
†3:Mulliken 電荷(必須)
†4:RESP 電荷
†5:原子タイプ指定フラグ(s/j)
s: †6の原子タイプを指定
j:指定無し
†6:原子タイプ

(2) 結合情報ファイル(XXX.bond)について

結合情報を結合している原子の番号の組、結合距離、結合次数の順に指定する。この情報は全て必須とする。

1 ^{†1}	3 ^{†2}	1.5180 ^{†3}	0.8310 ^{†4}
1	11	1.2080	1.8100
1	12	1.3460	0.8820
2	3	1.4720	0.8860

†1:結合している原子の番号
†2:結合している原子の番号
†3:結合距離
†4:結合次数

(3) 座標(Z-matrix)情報ファイル(XXX.zmat)について

Z-matrix 情報を以下のフォーマットで指定する。Z-matrix 情報ファイルは tplgeneL の実行に必須ではなく、このファイルがある場合には、トポロジーファイルの Z-matrix 情報を出力する部分にその値を反映する。

C						
N	1	2.4455870				
C	1	1.5183720	2	34.5282		
C ^{†1}	3 ^{†2}	1.5377660 ^{†3}	1 ^{†4}	112.3394 ^{†5}	2 ^{†6}	122.7222 ^{†7}
C	4	1.5155030	3	117.6078	1	177.0822
C	5	1.4008330	4	123.5493	3	346.7479

†1:元素記号
†2:結合している原子の番号
†3:結合している原子との原子間距離()
†4: †2と結合角をなす原子の番号
†5:結合角(°)
†6: †2と†4と二面角をなす原子の番号
†7:二面角(°)

3-1-2 Sybyl mol2 入力ファイル

3-1-1 の tplgeneL オリジナルファイル形式に加え、より一般的に使用されている Sybyl mol2 ファイル形式に対応している。フォーマットについては以下の URL を参照している。

<http://www.tripos.com/custResources/mol2Files/>

本プログラムでは Sybyl mol2 ファイルのフォーマットの中で、ATOM ブロック、及び、BOND ブロックの情報を取得し、その情報を用いてトポロジーファイルを作成する。以下に mol2 ファイル例を示す。ATOM 項 (@<TRIPOS>ATOM) BOND 項 (@<TRIPOS>BOND) は必須とする。

@<TRIPOS>MOLECULE

NH3.mol2

4 3 0 0 0

SMALL

NO_CHARGES

@<TRIPOS>ATOM

1	N ^{†1}	0.0530	0.0380	-0.1390 ^{†2}	N.4	0	<1>	-0.9299 ^{†3}
2	H	-0.8360	-0.4780	-0.1390	H	0	<1>	0.3100
3	H	0.8360	-0.6290	-0.1390	H	0	<1>	0.3098
4	H	0.1070	0.6290	0.7000	H	0	<1>	0.3100

@<TRIPOS>BOND

1	1	2 ^{†4}	1 ^{†5}
2	1	3	1
3	1	4	1

†1: 原子名
†2: カーテシアン座標
†3: 電荷
†4: 結合情報
†5: 結合のタイプ(1:単結合、 2:二重結合、 3:三重結合、ar:芳香族結合)

3-2 原子タイプ割り当て基本情報取得

3-2-1 原子タイプ、デフォルト原子タイプについて

cosgene で MD 計算を行う場合には分子内の各 atom、bond、angle、torsion、improper-torsion のパラメータを割り当てる必要がある。tplgeneL ではパラメータを割り当てる為の基本情報として、各原子の環境(元素名、結合の数、混成軌道のタイプ、結合している原子の環境、等)毎に原子タイプ、デフォルト原子タイプを割り当てている。

(1)原子タイプについて

tplgeneL では、AMBER General Amber Force Field(GAFF)、及び、AMBER parm99 力場に対応する原子タイプを使用している。各力場の原子タイプについては 3-5-2、3-5-3 を参照の事。

(2)デフォルト原子タイプについて

tplgeneL では、AMBER GAFF、AMBER parm99 力場で定義されていない原子タイプを定義する場合には、デフォルト原子タイプとして定義を行う。

また、原子タイプは定義されているものの、対応する結合、結合角、二面角、インプロパー二面角パラメータが存在しない場合には、デフォルト原子タイプについて定義されているデフォルトパラメータを使用する。そこで、デフォルト原子タイプの定義が必須となる。

tplgeneL プログラム内では、原子タイプとデフォルト原子タイプは対となっている必要があり、両者をセットとして登録する。

3-2-2 原子タイプ割り当て基本情報取得

3-1 で取得した分子情報から、原子タイプを割り当てるための必要な情報を作成する。原子タイプ割り当てに必要な情報は以下の通りである。

- ・ 結合のタイプ
- ・ 混成軌道のタイプ
- ・ 分子内の環構造の数、芳香環、脂環か否か

(1) 結合のタイプの判別

入力分子の各結合の結合次数からその結合が単結合、芳香族結合、二重結合、三重結合かの判別を行う。

結合次数が 0.2 以上、1.2 未満の場合には、単結合と見なす。

結合次数が 1.2 以上、1.5 未満の場合には、芳香族結合と見なす。

結合次数が 1.5 以上、2.15 未満の場合には、二重結合と見なす。

結合次数が 2.15 以上の場合には、三重結合と見なす。

(2) 混成軌道のタイプの判別

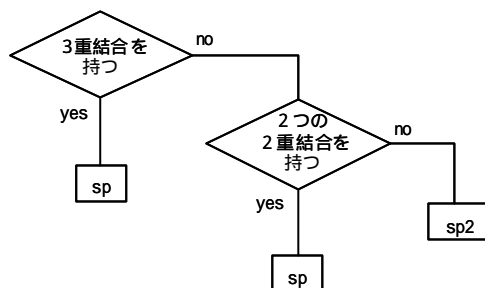
以下の表に従って混成軌道のタイプを取得する。ここでは、各原子の元素記号と結合の数、結合している原子の情報から混成軌道を判別する。

元素	結合の数	混成軌道のタイプ	元素	結合の数	混成軌道のタイプ
H	1	s	N	1	sp
C	1	sp		2	sp, sp ² 1
	2	sp		3	sp ² , sp ³ 2
	3	sp ²		4	sp ⁴
	4	sp ³	P	2	sp ²
O	1	sp ²		3	sp ³
	2	sp ³		4	sp ³
S	1	sp ²	Ca	1	s
	2	sp ³	F	1	s
	3	sp ³	Cl	1	s
	4	sp ³	Br	1	s
			I	1	s

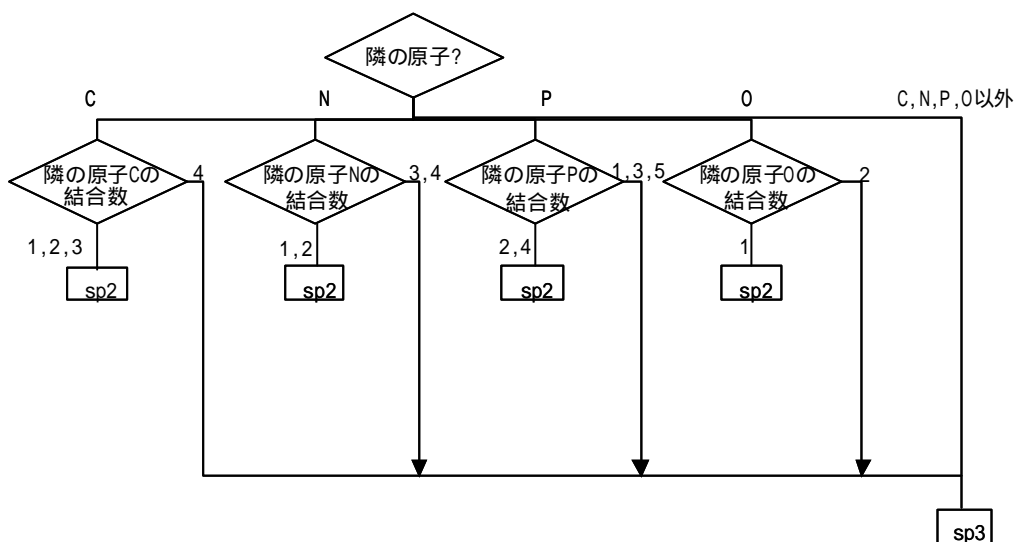
N 原子の様に同じ結合の数でも複数の混成軌道のタイプを取る場合は、得られる混成軌道は化学的に必ずしも正確なものではない。

1,2 元素 N の結合の数が 2, 3 の場合以下の図に従って、混成軌道を判別する。

1 元素 N の結合の数が 2 の場合



2 元素 N の結合の数が 3 の場合

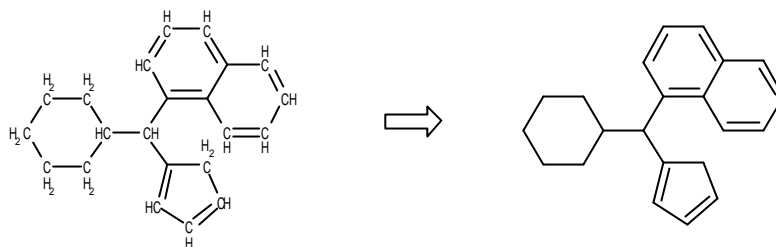


(3) 分子内の環構造の数、芳香環、脂環のチェック

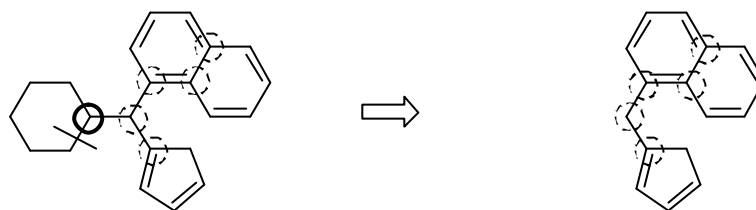
脂環、芳香環の判別は、環の判定、および、環構造内電子数のチェックの2段階で行う。

. 環の判定について

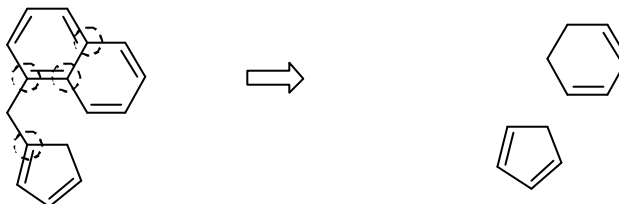
結合の数が1の原子の結合を切る。
この操作を切る結合がなくなるまで繰り返す。



結合の数が3の原子があった場合、その結合の1つを切る。
この操作によって結合の数が1になった原子について の操作を行う。



結合の数が3の原子がなくなるまで の操作を繰り返す。



最後まで残った原子は環を構成しているので、その原子番号の組を保存する。

元の分子から で得られた原子を取り除き(ただし、複数の環内で共有している原子を除く)、すべての環構造が得られるまで、 ~ の操作を繰り返す。

. 芳香族判定について

環を構成する原子の結合の数を数え、環内の原子毎に以下の規則により、その環にある二重結合に関りうる電子の数を数える。

炭素：1 電子

窒素：1 電子（結合の数が 2 本）、2 電子（結合の数が 3 本）

酸素：2 電子

硫黄：2 電子

の規則に従って環内の電子の数を数えて $4n+2$ (n は整数) 個になったときは芳香族としてのフラグを立てる。ただし、環内に結合の数 3、 sp^2 の窒素がある場合、電子の合計が 7 になっても例外的に芳香族とみなす。

3-3 分子トポロジー生成

3-1, 3-2 で取得した情報をもとに以下の分子トポロジーを生成する。

- bond
- angle
- torsion
- improper-torsion

3-4 原子タイプ割り当て規則 DB 読み込み

原子タイプの規則をプログラムを変更せずに対応する為に、原子タイプの割り当て規則を外部のファイルとして定義する。

原子タイプ割り当て規則 DB ファイルは AMBER parm99 カ場用と、AMBER GAFF 用の 2 種類を用意している。

atomtype_gaff.db	AMBER GAFF カ場用原子タイプ割り当て規則 DB ファイル
atomtype_amber99.db	AMBER parm99 カ場用原子タイプ割り当て規則 DB ファイル

各原子タイプ割り当て規則 DB ファイルには、以下の条件、及び、原子タイプを記述する。

指定できる条件：

- ・ 元素名(element)
- ・ 結合の数(nbond)
- ・ 混成軌道の種類(hybrid)：s、sp、sp²、sp³、(sp³d、sp³d² は未対応)
- ・ 結合している原子が、電子吸引性の原子と結合している数(nelectrwd)：
酸素原子、及び、窒素原子を電子吸引性原子とする。
- ・ 原子が所属する環のメンバ数(ring)：
着目原子が複数の環に所属する場合はの大きな環のメンバ数を指定する(ただし、6 員環以下)。
- ・ 芳香族フラグ(aromatic)：その原子が芳香族原子ならば 1、異なれば 0 を入れる。
- ・ 環構造フラグ(circ)：その原子が環に含まれるならば 1、異なれば 0 を入れる。
- ・ 結合次数(border)

3-4-1 原子タイプ割り当て規則 DB のファイルフォーマットについて

以下に原子タイプ割り当て規則 DB ファイルのフォーマットを示す。

DB ファイルは begin で始まり end で終了するものとする。また、begin、end キーワードはファイル内に 1 組のみ記述するものとする。各条件は begin と end の間に記述するものとする。

```
例)
begin
  各条件
  .
  .
  .
end
```

各条件は if から then の間に記述する。

各条件は以下に示す様に指定する。また、各条件は以下に示すキーワードのみ使用できる。

```
if “KEY” (“原子指定番号”) “演算子” “VAL” then
```

例) 図 3 - 4 - 1 下線部 の例

```
if element(0) = N then
```

原子指定番号とは、着目原子の周辺の原子を指定するための番号である。これを用いて着目している原子周辺の環境を指定することができる (図 3 - 4 - 2 参照)

条件キーワード)

キーワード “KEY”	説明	実際に入る 値“VAL”の 例	対応 演算子	備考
element	元素名	C, N, ...	=	
hybrid	混成軌道	s, sp, sp2, sp3	=	現状対応しているのは s, sp, sp2, sp3 のみ
nbond	結合の数	1, 2, ...	=	
nelectrwd	電子吸引性の 原子に結合し ている数	1, 2, ...	=, <, >, <=, >=	

ring	原子が所属する環のメンバー数	5, 6, ...	=, <, >, <=, >=	
circ	環の一部か否か	0 : 非環構造	=	
		1 : 環構造		
aromatic	芳香族か否か	0 : 非芳香族	=	
		1 : 芳香族		
border	結合次数	1, 1.5, 2.0, ...	=, <, >, <=, >=	結合次数は量子化学計算値ではなく、単結合(1)、芳香族結合(1.5)、2重結合(2)、3重結合(3)を判別するために用いる

条件文の中で、複数の条件を指定する事ができる。複数条件を指定する場合は、AND 条件（キーワード “and”）で記述する。OR 条件には対応しない。

（例：図3 - 4 - 1の下線部）

条件記述部は複数行にまたがっても良い。ただし、それぞれの条件式（例：element(0)=N）の途中での改行はできないものとする。

対応する例： if element (0) = C and
 nbond(0) = 4 then

対応しない例： if element(0) =
 C and nbond(0) = 4 then

原子タイプ、デフォルト原子タイプの代入文は以下のフォーマットとする。ここで KEY は “atom_type”、又は、“default_atom_type” である。

“KEY” := “VAL” ;

例)図3 - 4 - 1の下線部 の例

atom_type := ne;

キーワード)

原子タイプ : atom_type,

デフォルト原子タイプ : default_atom_type

原子タイプ、デフォルト原子タイプは begin の直後、あるいは、if 文の直後に記述する。

右の例では太字部の if 文の意味がなくなるため、このような定義を許さないものとする。

正しい例)

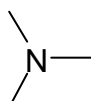
```
if nbond(0) = 3 and hybrid(0) = sp3 then
  atom_type := n3;
  default_atom_type := 3n2;
  if element(02) = C then
    if element(021) = 0 and
      nbond(021) = 1 then
      atom_type := n;
      default_atom_type := 3n1;
    endif
  endif
endif
endif
```

誤り例)

```
if nbond(0) = 3 and hybrid(0) = sp3 then
  if element(02) = C then
    if element(021) = 0 and
      nbond(021) = 1 then
      atom_type := n;
      default_atom_type := 3n1;
    endif
  endif
  atom_type := n3;
  default_atom_type := 3n2;
endif
endif
```

デフォルト原子タイプは結合の数、元素記号、数字の文字列で指定する。異なるデフォルト原子タイプを指定する際に、結合の数、元素記号が重複している場合には 3 番目の数字を重複しない数字で指定する(3n1、3n2、、、等)。

例) 結合の数 3 の窒素の例



デフォルト原子タイプ: 3n1

デフォルト原子タイプは上位の条件で指定されていた場合、省略することができる。

図 3 - 4 - 1 でデフォルト原子タイプを省略した 11~13 行目のブロックに対して上位の条件である、10 行目のデフォルト原子タイプ 2n2 を採用する。

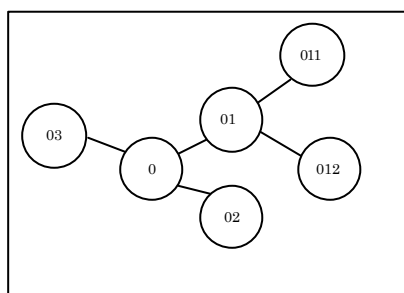
‘!’ 以降のカラムはコメント行とする。

```

1  begin
2  if element(0) = N then
3      if aromatic(0) = 0 then
4          if nbond(0) = 1 then
5              atom_type := n1;
6              default_atom_type := 1n1;
7          endif
8          if nbond(0) = 2 then
9              atom_type := n1;
10             default_atom_type := 2n2;
11             if hybrid(0) = sp2 then
12                 atom_type := n2;
13             endif
14             if border(01) = 2 and border(021) > 1.5 then
15                 atom_type := ne;
16                 default_atom_type := 2n1;
17                 if circ(0) = 1 then
18                     atom_type := nc;
19                     default_atom_type := 2n1;
20                 endif
21             endif
22         endif
23     endif
24 endif
25 end

```

図 3 - 4 - 1 . 原子タイプ割り当て規則 DB ファイル例



0 : 着目原子
01,02,03 : 着目原子の隣の原子
011, 012 : 隣の隣の原子

図 3 - 4 - 2 . 原子指定番号

3-4-2 原子タイプ割り当て規則 DB の読み込み (中間情報の作成)

3-4-1 のフォーマットで記述された割り当て規則は if 文の階層的な構造になっているため、プログラムで処理しやすい中間情報に変換し保存する。

中間情報の作成処理の流れ

(1)原子タイプ割り当て規則 DB の読み込み処理

if 文の階層的に記述された原子タイプ割り当て規則 DB を読み込み、構文木の作成を行う。

(2)原子タイプ毎の条件の結合処理

(1)で作成した構文木をもとに、原子タイプ毎に条件を結合し、一次元的な情報に変換し保存する。

(3)原子指定番号毎の条件のグループ化処理

原子タイプ毎に結合した、各条件を各原子指定番号 (着目原子、隣、その隣の原子など) 毎に整理して保存する。

以下にそれぞれの処理を説明する。

(1)原子タイプ割り当て規則 DB の読み込み処理

ファイルの内容を順次解釈し、各 if 文の情報を構造体 PRMIfNode に保存する。構造体 PRMIfNode は、if ポインタ、then ポインタ、条件保存部、原子タイプ部で構成する(図 3 - 4 - 3 参照)。

if 文の情報を保存した構造体群を、以下に示す処理によりポインタで結合し構文木を作成する(図 3 - 4 - 4 参照)。

【構文木作成処理】

if から then の間に記述された条件を、構造体 PRMIfNode の条件保存部に保存する。(root の場合はこの処理は行わない)

ファイル内に以下のキーワードがある場合に、そのキーワード毎に以下の処理を行う。

<atom_type, default_atom_type の場合>

原子タイプ、デフォルト原子タイプを保存し、 の処理を繰り返す。

<if の場合>

(i)新しい条件を保存する構造体 PRMIfNode を生成し、以下のようにつなく。

a)参照中の構造体の then ポインタが null の場合、参照中の構造体の then ポインタにつなく。

b)参照中の構造体の then ポインタが null でない場合、then ポインタにつながっている (一つ下の階層の) 構造体の if ポインタをたどり、一番最後 (if ポインタが null) の構造体の if ポインタにつなく。

(ii)新しく作成した構造体を参照するようにセットし、再帰的に本処理を呼び出す。

(i),(ii)の処理終了後、 の処理を繰り返す。

<end, endif の場合>

root の場合は “end”、それ以外の場合は “endif” キーワードが現れた場合、この処理を終了する。

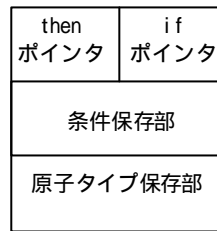


図 3 - 4 - 3 . 構造体 PRMIfNode

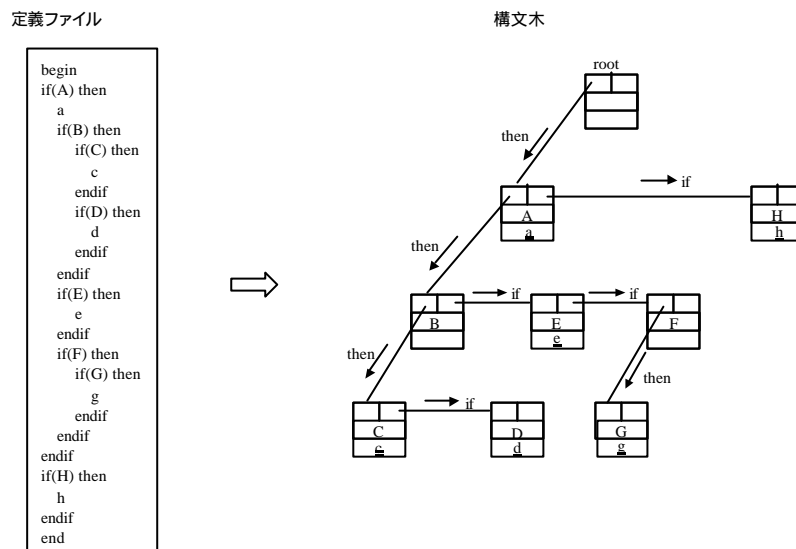


図 3 - 4 - 4 . 構文木の作成

(2)原子タイプ毎の条件の結合処理

作成した構文木をもとに以下の処理を行い、原子タイプ毎の条件を結合する。

【原子タイプ毎の条件の結合処理】

条件とデフォルト原子タイプを配列に保存する。

参照している構造体 PRMIfNode に原子タイプが保存されてる場合、配列の内容(これまで保存してきた条件群、及び、デフォルト原子タイプ)と原子タイプもとに、(3)の原子指定番号毎の条件のグループ化処理を行う。

then ポインタが null で無い場合、then ポインタに繋がっている条件に対して再帰的に本処理を呼び出す。

自分自身の条件を配列から削除する。

if ポインタが null で無い場合、if ポインタに繋がっている条件に対して再帰的に本処理を呼び出す。

(3)原子指定番号毎の条件のグループ化処理

(2)の処理によって、原子タイプ毎に結合した条件のグループ化を行う。条件のグループ化は if 毎に作成する構造体 PRMIfFrame と、各 if 文内の各条件の原子指定番号毎に作成する構造体 PRMAtomCondition の2つの構造体を用いる。構造体 PRMIfFrame は構造体 PRMAtomIfFrame へのポインタ、原子タイプ、次の if 文(構造体 PRMIfFrame)へのポインタからなる。構造体 PRMAtomCondition は原子指定番号毎の条件保存部からなる。

原子タイプを if 文の骨格用構造体 PRMIfFrame (図3 - 4 - 5) に保存する。各 if 文中の条件を各原子指定番号(着目原子、隣、その隣の原子など)毎に整理して原子指定毎条件の条件を保存する構造体 PRMAtomCondition (図3 - 4 - 5) に保存する。

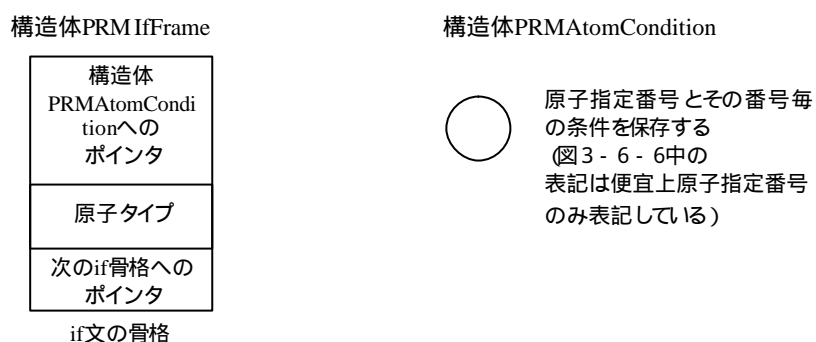


図3 - 4 - 5 . 構造体 PRMIfFrame と構造体 PRMAtomCondition

以下の処理を行い、原子タイプ毎に結合した条件を各原子指定番号毎にグループ化し、図3 - 4 - 6のような構造に保存する。

【原子指定番号毎の条件のグループ化処理】

if 文の骨格用構造体 PRMIfFrame を生成し、原子タイプ、デフォルト原子タイプを保存する。

各条件に対して、以下のように構造体を生成し、原子指定番号毎の条件を保存する。

(i)原子指定番号が if ブロック内で新規の場合、

原子指定番号毎の条件を保存する構造体 PRMAtomCondition を生成、条件を保存する。

a)上位の(原子指定番号の末尾を1つ除いた)構造体に、

生成した構造体と同じ階層の構造体がつながっていない場合、上位の構造体と生成した構造体をつなぐ。

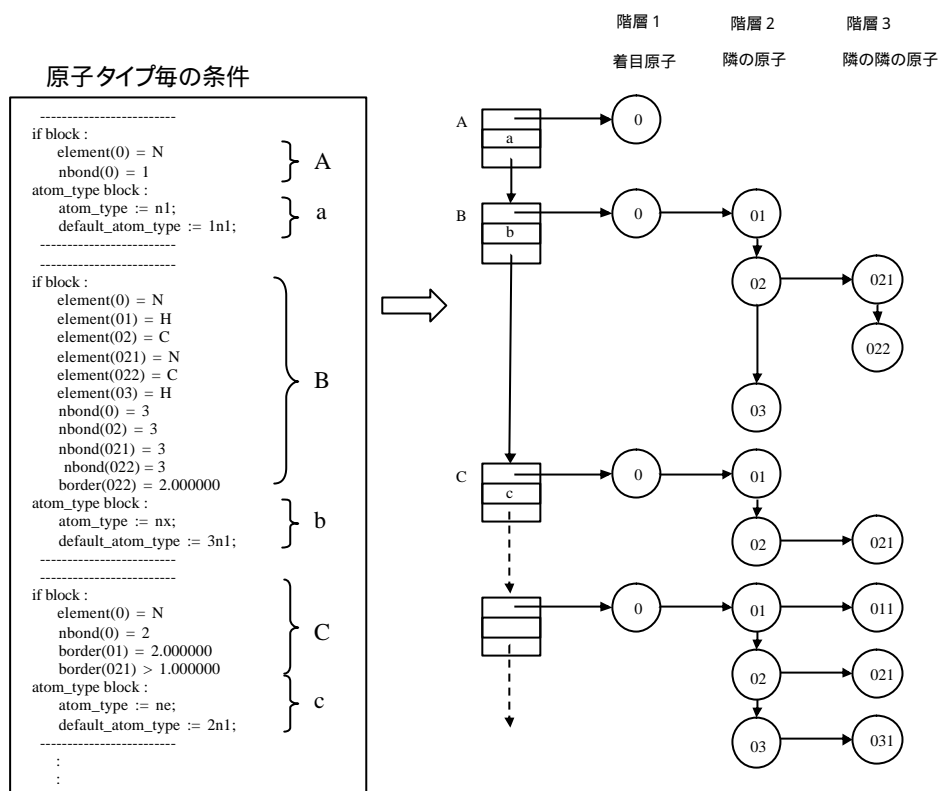
b)上位の構造体に、生成した構造体と同じ階層(注1)の構造体が既につながっている場合、同じ階層の構造体の末尾と生成した構造体をつなぐ

この際、生成した構造体より上の階層の構造体が無かった場合、その上位の構造体も生成する。

(ii)原子指定番号が if 文の条件内で既出の場合、

新たに構造体を生成せず、原子指定番号が一致する構造体に条件を保存する。

注1)階層とは着目原子からの位置関係を示し、上の階層とは参照している原子より着目原子に近い原子、下の階層とは参照している原子より着目原子から遠い原子を指す。



英大文字：条件
 英小文字：原子タイプ

図 3 - 4 - 6 . 各条件のグループ化

3-5 原子タイプ割り当て

3-2 で取得した、入力分子内の各原子の基本情報（元素名、結合の数、混成軌道の種類など）と、3-4 でグループ化、保存した原子タイプの定義情報（図 3 - 4 - 6 参照）とを比較する。原子指定番号毎の各条件 (PRMAtomCondition) が全て一致した場合、if 文の骨格用構造体 (PRMIFFrame) に保存している原子タイプを割り当てる。また、複数の原子タイプの定義情報と一致する場合は、後の方の原子タイプを上書きして割り当てる。

3-5-1 原子タイプ割り当て条件の比較

各原子指定番号毎の条件の評価は以下の処理で行い、戻り値として TRUE が返ってきた場合には、if 文の骨格用構造体に保存されている原子タイプ、デフォルト原子タイプを割り当てる。FALSE の場合には、次の if 文の条件について同様に評価する。

【原子タイプ割り当て処理】

指定された原子指定番号の条件(元素名、結合の数など)が参照している原子の情報と、

(i) 一致しない場合、

FALSE を呼び出し元関数に返す。

(ii) 一致し、かつ、指定された原子指定番号より下の階層^(注1)の条件が無い場合、

TRUE を呼び出し元関数に返す。

(iii) 一致するが、指定された原子指定番号より下の階層の条件がある場合、

参照している原子に結合している隣の原子について、本関数を再帰的に呼び出す。

下の階層の条件全てを満たす(TRUE を返す)隣の原子の組み合わせが得られた場合、TRUE を呼び出し元関数に返す

下の階層の条件全てを満たす隣の原子の組み合わせが得られ無かった場合、FALSE を呼び出し元関数に返す

注 1) 階層とは着目原子からの距離を示し、上の階層とは参照している原子より着目原子に近い原子、下の階層とは参照している原子より着目原子から遠い原子を指す。

3-5-2 GAFF 原子タイプ割り当て

以下に GAFF 原子タイプ一覧 (表 3 . 1) を示す。

表 3 . 1 GAFF(General Amber Force Field)原子タイプ一覧

元素名	原子タイプ	原子タイプ説明	元素名	原子タイプ	原子タイプ説明	
C	c	Sp2 C carbonyl group	N	n	Sp2 nitrogen in amide groups	
	c1	Sp C		n1	Sp N	
	c2	Sp2 C		n2	aliphatic Sp2 N with two connected atoms	
	c3	Sp3 C		n3	Sp3 N with three connected atoms	
	ca	Sp2 C aromatic		n4	Sp3 N with four connected atoms	
S	s2	S with one connected atom		na	Sp2 N with three connected atoms	
	s4	S with three connected atoms		nb	aromatic Sp2 N with two connected atoms	
	s6	S with four connected atoms		nh	Amine N connected one or more aromatic rings	
	sh	Sp3 S connected with hydrogen		no	Nitro N	
	ss	Sp3 S in thio-ester and thio-ether		O	o	Oxygen with one connected atom
H	h1	H bonded to aliphatic carbon with 1 electrwd. Group	oh		Oxygen in hydroxyl group	
	h2	H bonded to aliphatic carbon with 2 electrwd. Group	os		Ether and ester oxygen	
	h3	H bonded to aliphatic carbon with 3 electrwd. Group	P		p2	Phosphate with two connected atoms
	h4	H bonded to non-sp3 carbon with 1 electrwd. group			p3	Phosphate with three connected atoms, such as PH3
	h5	H bonded to non-sp3 carbon with 2 electrwd. group		p4	Phosphate with three connected atoms, such as O=P(CH3)2	
	ha	H bonded to aromatic carbon		p5	Phosphate with four connected atoms, such as O=P(OH)3	
	hc	H bonded to aliphatic carbon without electrwd. Group	F	f	Fluorine	
	hn	H bonded to nitrogen atoms	Cl	cl	Chlorine	
	ho	Hydroxyl group	Br	br	Bromine	
	hp	H bonded to phosphate	I	i	Iodine	
	hs	Hydrogen bonded to sulphur	/			
	hx	H bonded to C next to positively charged group				

3-2 で取得した原子タイプ割り当て規則基本情報をもとに、原子タイプとデフォルト原子タイプを割り当てる。以下に GAFF の原子タイプ割り当て定義一覧を示す（表 3 . 2 参照）。

なお、原子タイプ割り当て規則 DB ファイル(atom_type_gaff.db ファイル)は下表の登録順序に従って記述している。割り当て規則 DB ファイル内の条件と入力分子中の各原子の基本情報との比較を行い、マッチした場合にその原子タイプを割り当てる。

表 3 . 2 GAFF 原子タイプ割り当て定義一覧(atomtype_gaff.db)

No	原子タイプ	原子指定番号	元素名	結合数	混成軌道	結合次数	環フラグ	芳香族フラグ	環原子数
1	c1	0	C	1					
2	c1	0	C	2					
3	c2	0	C	3					
4	c	0	C	3					
		01	O	1					
5	ca	0	C	3			1		
6	c3	0	C	4					
7	n1	0	N	1					
8	n2	0	N	2	sp2				
9	n1	0	N	2	sp				
10	nb	0	N	2			1		
11	n3	0	N	3	sp3				
12	n	0	N	3	sp3				
		02	C						
		021	O	1					
13	nh	0	N	3			1		
		01					1		
14	na	0	N	3	sp2				
15	n3	0	N	3					
		01	S						
		011	O	1					
16	n	0	N	3					
		01	P						
		011	O	1					
17	n	0	N	3					
		02	C						
		021	O	1					
18	no	0	N	3					
		02	O	1					
		03	O	1					
19	n4	0	N	4					
20	o	0	O	1					
21	os	0	O	2					
22	oh	0	O	2					
		01	H						
23	ow	0	O	2					
		01	H						
		02	H						

表 3 . 2 GAFF 原子タイプ割り当て規則(atomtype_gaff.db) (続き)

24	ho	0		H								
			01	O								
25	hw	0		H								
			01	O								
				011	H							
26	hs	0		H								
			01	S								
27	hp	0		H								
			01	P								
28	hn	0		H								
			01	N								
29	ha	0		H								
			01	C						1		
30	h5	0		H								
			01	C						1	2	
31	h4	0		H								
			01	C						1	1	
32	hc	0		H								
			01	C						0		
33	hc	0		H								
			01	C	4					0		
34	h3	0		H								
			01	C	4					0	3	
35	h2	0		H								
			01	C	4					0	2	
36	h1	0		H								
			01	C	4					0	1	
37	h5	0		H								
			01	C	3					0	2	
38	h5	0		H								
			01	C	3					0	3	
39	h4	0		H								
			01	C	3					0	1	
40	hc	0		H								
			01	C	2					0		
				011	C							
41	h1	0		H								
			01	C	1					0		
42	hx	0		H								
			01	C								
				011	N	4						
43	p2	0		P	2							
44	p5	0		P	4							
45	p3	0		P	3							
46	p4	0		P	3							
			01	O	1							
47	s2	0		S	1							
48	ss	0		S	2							
49	sh	0		S	2							
			01	H								
50	s4	0		S	3							
51	s6	0		S	4							
52	f	0		F								
53	cl	0		Cl								
54	br	0		Br								
55	l	0		I								

デフォルト原子タイプは以下の表（表 3 . 3 ）に基づき割り当てる

表 3 . 3 GAFF デフォルト原子タイプ割り当て規則(atomtype_gaff.db)

元素名	結合の数	混成軌道	その他	デフォルト 原子タイプ
C	1			1c1
	2			2c1
	3			3c1
	4			4c1
N	1			1n1
	2	sp2		2n1
		sp		2n2
	3	sp2		3n1
		sp3		3n2
	4			4n1
O	1			1o1
	2			2o1
H	1			1h1
P	2			2p1
	3			3p1
			0 と二重結合	
	4			4p1
S	1			1s1
	2			2s1
	3			3s1
	4			4s1
F	1			1f1
Cl	1			1cl1
Br	1			1br1
I	1			1i1

3-5-3 AMBER parm99 原子タイプ割り当て

以下に AMBER parm99 の原子タイプ一覧 (表 3 . 4) を示す。

表 3 . 4 AMBER parm99 原子タイプ一覧

元素名	原子タイプ	原子タイプ説明	元素名	原子タイプ	原子タイプ説明
C	C	Sp2 C carbonyl group	N	N	sp2 nitrogen in amide groups
	CA	sp2 C pure aromatic (benzene)		NA	sp2 N in 5 memb.ring w/H atom (HIS)
	CB	sp2 aromatic C, 5&6 membered ring junction		NB	sp2 N in 5 memb.ring w/LP (HIS,ADE,GUA)
	CC	sp2 aromatic C, 5 memb.ring HIS		NC	sp2 N in 6 memb.ring w/LP (ADE,GUA)
	CK	sp2 C 5 memb.ring in purines		N2	sp2 N in amino groups
	CM	sp2 C pyrimidines in pos. 5 & 6		N3	sp3 N for charged amino groups (Lys, etc)
	CN	sp2 C aromatic 5&6 memb.ring junct.(TRP)		NT	sp3 N for amino groups amino groups
	CQ	sp2 C in 5 mem.ring of purines between 2 N		N*	sp2 N
	CR	sp2 arom as CQ but in HIS	NY	nitrile N (Howard et al.JCC,16,243,1995)	
	CT	sp3 aliphatic C	O	O	carbonyl group oxygen
	CV	sp2 arom. 5 memb.ring w/1 N and 1 H (HIS)		O2	carboxyl and phosphate group oxygen
	CW	sp2 arom. 5 memb.ring w/1 N-H and 1 H (HIS)		OH	oxygen in hydroxyl group
				OS	ether and ester oxygen
	C*	sp2 arom. 5 memb.ring w/1 subst. (TRP)	P	P	phosphate,pol:JACS,112,8543,90,K.J.Miller
	CY	nitrile C (Howard et al.JCC,16,243,1995)		S	S
CZ	sp C (Howard et al.JCC,16,243,1995)	SH	S in cystine		
F	F	fluorine	H	H	H bonded to nitrogen atoms
CL	Cl	chlorine (Applequist)		HC	H aliph. bond. to C without electrwd.group
BR	Br	bromine (Applequist)		H1	H aliph. bond. to C with 1 electrwd. Group
I	I	iodine (Applequist)		H2	H aliph. bond. to C with 2 electrwd.groups
				H3	H aliph. bond. to C with 3 eletrwd.groups
				HA	H arom. bond. to C without elctrwd. Groups
				H4	H arom. bond. to C with 1 electrwd. Group
				H5	H arom.at C with 2 elctrwd. gr,+HCOO group
				HO	hydroxyl group
				HS	hydrogen bonded to sulphur (pol?)
			HP	H bonded to C next to positively charged gr	
			HZ	H bond sp C (Howard et al.JCC,16,243,1995)	

以下に AMBER parm99 の原子タイプ割り当て規則を示す (表 3 . 5 参照)。なお、原子タイプ割り当て規則 DB ファイル(atom_type_amber99.db ファイル)は下表の登録順序に従って記述している。割り当て規則 DB ファイル内の条件と入力分子中の各原子の基本情報との比較を行い、マッチした場合にその原子タイプを割り当てる。

表 3 . 5 AMBER parm99 原子タイプ割り当て規則(atomtype_amber99.db)

No	原子タイプ	原子指定番号	元素名	結合数	混成軌道	結合次数	環フラグ	芳香族フラグ	隣接する電子吸引性の原子数	環原子数
1	CZ	0	C	2						
2	CY	0	C	2						
		01	N			=3.0				
3	CT	0	C	4						
4	CA	0	C	3						
		01	N	3						
		02	N	3						
		03	N	3						
5	CA	0	C	3						
		01	N	3						
		02	N	3						
		03	N	2						
6	CA	0	C	3			1			
7	CM	0	C	3			1			
		01	C			>1.5				=6
		02	C							=6
		02	C							=6
		021	N							=6
8	CM	0	C	3			1			
		01	N							=6
		02	C							=6
		02	C			>1.5				=6
		021	C							=6
9	C	0	C	3						
		01	O	1						
10	CA	0	C	3			1			
11	CB	0	C	3				1		
		01						1		=6
		02					1			=6
		03						1		=5
12	CN	0	C	3				1		
		02	C				1			=6
		03	H					1		=5
		03	N					1		=5
		01	H					1		=6
13	CC	0	C	3				1		=5
		01	C	4			0			
		02	N	3				1		
		03	C	3				1		
14	CC	0	C	3				1		=5
		01	C	4			0			
		02	N	2				1		
		03	C	3				1		
15	CK	0	C	3				1		=5
		01	N					1		
		011								
		0111								
		02	N					1		=6
		021								
		0211	N					1		
		02111								=6
16	CR	0	C	3				1		=5
		01	N							=5
		02	H							=5

表 3 . 5 AMBER parm99 原子タイプ割り当て規則(atomtype_amber99.db) (続き)

17	CV	0				C	3				1	=5
			01			N	2				1	=5
			02			C					1	=5
			03			H						
18	CW	0				C	3				1	=5
			01			N	3				1	=5
				011		H						
			02			C					1	=5
			03			H						
19	C*	0				C	3				1	=5
			01			C	4					
			02			C					1	=5
			03			C					1	=6
				031		C					1	=6
20	CQ	0				C	3				1	=6
			01			H						
			02			N	2				1	=6
				021		C					1	=6
			03			N	2				1	=6
21	H	0				H						
			01			N						
22	HC	0				H						
			01			C						
23	H1	0				H						
			01			C	4					1
24	H2	0				H						
			01			C	4					2
25	H3	0				H						
			01			C	4					3
26	HA	0				H						
			01			C	3					
27	H4	0				H						
			01			C	3					1
28	H5	0				H						
			01			C	3					2
29	HZ	0				H						
			01			C		sp				
30	HP	0				H						
			01			C						
				011		N	4					
31	HO	0				H						
			01			O						
32	HW	0				H						
			01			O						
				011		H						
33	HS	0				H						
			01			S						

表 3 . 5 AMBER parm99 原子タイプ割り当て規則(atomtype_amber99.db) (続き)

34	NY	0					N	1	sp											
			01				C			=3.0										
35	NY	0					N	2	sp											
			01				C			=3.0										
36	NB	0					N	2												=5
			01				C					1								
			02				C					1								
37	NC	0					N	2												=6
			01				C					1								=6
			02				C					1								
38	NT	0					N	3	sp3											
39	N2	0					N	3	sp2					0						
40	N	0					N	3	sp2											
			01				C	3												
					011		O	1												
41	N*	0					N	3	sp2			1								
			01				C	3				1								
			02				C		sp2			1								
			03				C	4												
42	NA	0					N	3	sp2											=5
			01				C	3												=5
			02				C					1								=5
			03				H													
43	NA	0					N	3	sp2											=5
			01				C	3												=6
			02				C					1								=5
			03				H													
44	NA	0					N	3	sp2											=6
			01				C	3				1								
			02				H													
45	N*	0					N	3												
			01				C	4				1		0						
			02				C	3						1						
					021		C	3						1						
					022		N	2				1								
			03				C	3						1						
46	N3	0					N	4												
47	O	0					O	1												
			01				C	3												
48	O2	0					O	1												
			01				C	3												
					011		O	1												
49	O2	0					O	1												
			01				P													
					011		O	1												
50	OS	0					O	2												
51	OH	0					O	2												
			01				H													
52	OW	0					O	2												
			01				H													
			02				H													
53	P	0					P													
54	S	0					S	2												
55	SH	0					S	2												
			01				H													
56	F	0					F													
57	Cl	0					CL													
58	Br	0					BR													
59	I	0					I													

デフォルト原子タイプは以下の表（表 3 . 6 ）に基づき割り当てる

表 3 . 6 AMBER parm99 デフォルト原子タイプ
割り当て規則 (atomtype_amber99.db)

元素名	結合の数	混成軌道	その他	デフォルト 原子タイプ
C	2			2C1
	3			3C1
	4			4C1
N	1			1N1
	2	sp2		2N1
		sp		2N2
	3	sp2		3N1
		sp3		3N2
	4			4N1
O	1			1O1
	2			2O1
H	1			1H1
P	4			4P1
S	2			2S1
F				1F1
Cl				1CL1
Br				1BR1
I				1I1

3-6 パラメータ DB 読み込み

以下の DB から各パラメータを読み込む。

パラメータ DB ファイル

- ・ atom パラメータ
- ・ bond パラメータ
- ・ angle パラメータ
- ・ torsion パラメータ
- ・ improper-torsion パラメータ

nonbond 情報 DB ファイル

- ・ nonbond パラメータ

以下に各パラメータ DB ファイルの一覧を示す。

prm_gaff.db	AMBER GAFF 力場用パラメータ DB ファイル
prm_amber99.db	AMBER parm99 力場用パラメータ DB ファイル
nonbond_gaff.db	AMBER GAFF 力場用 nonbond 情報 DB ファイル
nonbond_amber99.db	AMBER parm99 力場用 nonbond 情報 DB ファイル

パラメータ DB ファイル(prm_gaff.db、 prm_amber99.db)と nonbond 情報 DB ファイル (nonbond_gaff.db、 nonbond_amber99.db)のファイルフォーマットについて示す。

3-6-1 パラメータ DB のファイルフォーマットについて

以下にパラメータ DB ファイル例とフォーマットを示す。

パラメータ DB は以下の各種データブロックに分けて記述する。各ブロックは Start ブロック名で始まり、end ブロック名で終了する。

ブロック名	意味
atom	フラグメント内の atom パラメータを記述
bond	フラグメント内の bond パラメータを記述
angle	フラグメント内の angle パラメータを記述
torsion	フラグメント内の torsion パラメータを記述
improper	フラグメント内の improper-torsion のパラメータを記述

フォーマット)

Start ブロック名

end ブロック名

例)

Start atom

:

end atom

Start bond

:

end bond

:

(1) 各種ブロック共通フォーマット

- ・各パラメータは、“,” で区切り、複数のパラメータを一行で記述することもできる。
- ・“;” 以降をコメント行とする。

(2) atom ブロック

atom ブロックで指定できるキーワードは以下の通りである。

キーワード	説明	実際の値の例
atomtype	原子タイプ	ca, na, ...
mass	原子の質量	12.011
rvdw	原子半径	1.908
evdw	電荷	0.086
level	パラメータのレベル	[rhf/6-31G*], ...
atom	元素名	C, N, ...
com	混成軌道の種類	s, sp, sp2, ...
nbond	結合の数	3, 4, ...
bond(n,atom)	隣の原子の元素名	C, N, [C,N]...
bond(n,com)	隣の原子の混成軌道の種類	s, sp, sp2, ...

カッコ内の n は最大の結合の数までの通し番号

例)

```

Start atom
  atomtype= ca ,mass= 12.01000 ,
  rvdw= 1.908000 ,evdw= 0.0860000,
  level=[rhf/6-31G*],; from AMBER gaff
  atom= C ,
  com= sp2 ,
  nbond= 3 ,
  bond(1,atom)= [C,N] , bond(1,com)= sp2,
  bond(2,atom)= [C,N] , bond(2,com)= sp2,
  bond(3,atom)= x ,
end atom

```

(3)bond ブロック

bond ブロックで指定できるキーワードは以下の通りである。

キーワード	説明	実際の値の例
atomtype(1)	結合している原子 1 の原子タイプ	ca, na,...
atomtype(2)	結合している原子 2 原子タイプ	ca, na,...
order(1,2)	結合次数の範囲	[1.7 < 3.0) 1.7 結合次数<3.0
K	バネ定数	473.7
Req	平衡結合距離	1.39

結合次数の範囲を指定は以下を用いて行う。

下限 : “ (” 境界を含まない

“ [” 境界を含む

上限と下限の区切り文字 : “ < ” (固定)

“ > ” (未対応)

上限 : “) ” 境界を含まない

“] ” 境界を含む

例)

```
Start bond
    K= 570.000 ,
    Req= 1.229,
    atomtype(1)= 1o1,atomtype(2)= 3c1,
    order(1,2)=[1.7 < 3.0 ) , ; double bond C=0
end bond
```

(4)angle ブロック

angle ブロックで指定できるキーワードは以下の通りである。

キーワード	説明	実際の値の例
atomtype(1)	結合角をなす原子 1 の原子 タイプ	ca, na, ...
atomtype(2)	結合角をなす原子 2 の原子 タイプ (結合角の中心)	ca, na, ...
atomtype(3)	結合角をなす原子 3 の原子 タイプ	ca, na, ...
K	バネ定数	63
Req	平行結合角	120

例)

```

Start angle
      K=    63.000 ,
      Req= 120.000 ,
      atomtype(1)= ca,atomno(2)=ca,atomno(3)= ca,
end   angle

```


(5)torsion ブロック

torsion ブロックで指定できるキーワードは以下の通りである。

キーワード	説明	実際の値の例
atomtype(1)	二面角をなす原子 1 の原子タイプ	ca, na, ...
atomtype(2)	二面角をなす原子 2 の原子タイプ	ca, na, ...
atomtype(3)	二面角をなす原子 3 の原子タイプ	ca, na, ...
atomtype(4)	二面角をなす原子 4 の原子タイプ	ca, na, ...
order(2,3)	atomtype(2),atomtype(3)間の結合次数の範囲	[1.7 < 3.0) 1 1.7 結合次数<3.0
no	二面角パラメータの数	1, 2, ...
pn(n) 2	回転周期	2
phase(n) 2	位相	180
pk(n) 2	バリアー高	12
idivf(n) 2	割る値	2

1 結合次数の範囲を指定は以下を用いて行う。

下限 : “ (” 境界を含まない

“ [” 境界を含む

上限と下限の区切り文字 : “ < ” (固定)

“ > ” (未対応)

上限 : “) ” 境界を含まない

“] ” 境界を含む

2 カッコ内の n は torsion パラメータの通し番号

例)

```

Start torsion
  no= 1 ,
    pn( 1)= 2 ,
    phase( 1)= 180.000 ,
    pk( 1)= 26.600 ,
    idivf( 1)= 4 ,
  !
  atomtype(1)= X ,atomtype(2)= c2,atomtype(3)= c2,atomtype(4)= X ,
  order(2,3)=[1.5 < 3.0) , ; double bond
end torsion

```

(6) improper ブロック

improper ブロックで指定できるキーワードは以下の通りである。

キーワード	説明	実際の値の例
atomtype(1)	インプロパー二面角をなす原子 1 の 原子タイプ	ca, na, ...
atomtype(2)	インプロパー二面角をなす原子 2 の 原子タイプ	ca, na, ...
atomtype(3)	インプロパー二面角をなす原子 3 の 原子タイプ (インプロパー二面角の中心)	ca, na, ...
atomtype(4)	インプロパー二面角をなす原子 4 の 原子タイプ	ca, na, ...
no	インプロパー二面角パラメータの数	1, 2, ...
pn(n)	回転周期	2
phase(n)	位相	180
pk(n)	バリアー高	12
idivf(n)	割る値	2

カッコ内の n は improper パラメータの
通し番号

例)

```

Start improper
  no= 1 ,
    pn( 1)= 2 ,
    phase( 1)= 180.000 ,
    pk( 1)= 1.100 ,
    idivf( 1)= 1 ,
  !
  atomtype(1)= na,atomtype(2)= ca,atomtype(3)= ca,atomtype(4)= c3 ,
end improper

```

3-6-2 nonbond 情報 DB のファイルフォーマットについて

以下に nonbond 情報 DB ファイル例とフォーマットを示す。
nonbond 情報 DB ファイルは以下の 2 つブロックに分けて記述する。

(1)FUNCTIONS ブロック

非共有結合相互作用を計算する関数を記述する。以下のいずれかを記述する。

LENNARD-JONES-AMBER

H-BONDING-AMBER

(2)NONBONDS ブロック

非共有結合相互作用を計算する際に用いるパラメータを記述する。

(1)FUNCTIONS ブロック

FUNCTIONS ブロックとは、PRE> FUNCTIONS から PRE> NONBONDS の前の行までを指す。
このブロックに記述される情報は以下の通りである。

非共有結合相互作用を計算する関数

原子タイプと原子タイプ番号（非共有結合のパラメータの割り当てに用いる）の
対応表

非共有結合を計算する関数

非共有結合相互作用を計算する関数名、パラメータの数を指定する。

PRE> FUNCTIONS の直後の行に関数名を以下のフォーマットで指定する。

また、“;”以降はコメント行とする。

フォーマット)

関数番号	この関数で使用する パラメータ総数	空白	関数名
%6d	%6d	6	%40s

例)

PRE> FUNCTIONS

;NONBONDED-POTENTIALS-FOR-PREPARATION

1 4 LENNARD-JONES-AMBER

; (R* -AND-E)

原子タイプと原子タイプ番号の対応表

の “ 非共有結合を計算する関数 ” 以降の行の “ ; TYPE OF ATOMS ” から PRE> NONBONDS の間に指定する。フォーマットは以下の通りで “ ; ” 以降に記述する。

(nonbond 情報 DB ファイルにおいて、通常 “ ; ” はコメントであるが、 “ ; TYPE OF ATOMS ” 以降のコメント行で原子タイプと原子タイプ番号の対応表を示す。この情報は必須である)

フォーマット)

```
; TYPE OF ATOMS
; 原子タイプ 1, 原子タイプ 2, 原子タイプ 2 : 原子タイプ番号
```

例)

```
; TYPE OF ATOMS
; c, c*, ca, cb, cc, cd, ck, cm, cn, cq, cr, cv, cw, cx, cy, cz, c1, c2 : 1
; c3, 4c1 : 2
; h : 3
```

上記の例は以下を表す。

原子タイプ c, c*, ca, ... の原子タイプ番号は 1

原子タイプ c3, 4c1 の原子タイプ番号は 2

原子タイプ h の原子タイプ番号は 3

(2)NONBONDS ブロック

NONBONDS ブロックとは、PRE> NONBONDS からファイルの終端までを指す。

このブロックには原子タイプ番号と非共有結合相互作用計算時のパラメータの対応を記述する。

tplgeneL で対応する AMBER parm99、GAFF では、非共有結合相互作用の計算に必要なパラメータは以下のとおりである。

- ・ 非共有結合を決める原子タイプ番号 i
- ・ 関数番号 (FUNCTIONS ブロックで指定した関数の番号を指定する)
- ・ van der Waals エネルギーのための原子半径 ri
- ・ van der Waals エネルギーの深さの値 ei
- 1-4 van der Waals エネルギーのための定数 Rvi
- 1-4 静電相互作用エネルギーのための定数 REj

上記パラメータを以下のフォーマットで記述する。

フォーマット)

原子タイプ番号 i	原子タイプ番号 j	関数タイプ番号	ri	ei	REij	RVij
%6d	%6d	%3d	%10.5lf	%11.6	%12.7	%8.3lf

例)

```
PRE> NONBONDS
```

```
;NUMBER OF TYPE= 40
```

```
  1  0  1  1.90800  0.086000  0.8333333  0.500; c
  2  0  1  1.90800  0.109400  0.8333333  0.500; c3
  3  0  1  0.60000  0.015700  0.8333333  0.500; h
  4  0  1  0.00000  0.000000  0.8333333  0.500; ho
```

3-7 パラメータ割り当て

3-6 で割り当てた原子タイプをもとに以下のパラメータを割り当てる。また、パラメータ DB ファイル内の各パラメータの項目に結合次数が指定されている場合は、結合次数がその範囲を満たすときのみ、そのパラメータを割り当てる。

- atom パラメータ
- bond パラメータ
- angle パラメータ
- torsion パラメータ
- improper-torsion パラメータ
- nonbond パラメータ

パラメータ DB 内に対応するパラメータが存在しない場合には、以下の2種類のいずれかの方法について不足パラメータの補填を行なう。

- デフォルトパラメータを用いてのパラメータ補填
- パラメータの動的計算による補填(動的補填は結合、結合角パラメータのみ対応)

以下に各補填方法について示す。

3-7-1 デフォルトパラメータを用いてのパラメータ補填

デフォルトパラメータとは、元素名と結合の数、混成軌道の種類で分類し、統計的に求めたパラメータであり、基本的なものは既にパラメータ DB ファイル(prm_gaff.db, prm_amber99.db)内に登録されている。

計算時に「デフォルトパラメータを使用する」を選択した場合にこれらのパラメータを使用してトポロジーファイルを作成する。

3-7-2 パラメータの動的計算による補填

計算時に「不足パラメータを計算する」を選択した場合に不足パラメータの動的補填を行う。ここでは入力構造の原子種、結合長、結合角の各値を使用して平衡結合長、平衡結合角、力の定数を計算を行う。

パラメータの動的補填は、結合パラメータ(平衡結合長、力の定数)、及び、結合角パラメータ(平衡結合角、力の定数)について、行う。

(1) 結合パラメータの補填について

結合パラメータ(力の定数)の補填は以下の式に従って計算を行う。

$$K_r = K_{ij} \left(\frac{1}{r_{ij}} \right)^m$$

ここで、mは4.5を使用する。r_{ij}は入力構造から得られる結合長の値を、K_{ij}は下表に示す各原子の組に対応する値を使用する。K_{ij}はbond.prmファイル内の情報を使用する。

また、平衡結合長は、入力構造の結合長を使用する。

(2) 結合角パラメータの補填について

結合角パラメータ(力の定数)の補填は以下の式を用いて計算を行う。

$$K_{ijk}^0 = 143.9 Z_i C_j Z_k (r_{ij}^{eq} + r_{jk}^{eq})^{-1} (q_{ijk}^{eq})^{-1/2} \exp(-2D)$$

$$D = \frac{(r_{ij}^{eq} - r_{jk}^{eq})^2}{(r_{ij}^{eq} + r_{jk}^{eq})^2}$$

ここで、r_{ij}^{eq}、r_{jk}^{eq}は入力構造から得られる結合長の値を使用する。q_{ijk}^{eq}は入力構造から得られる結合角の値を使用する。C及びZは下表に示す対応する値を使用する。C及びZはangle.prmファイル内の情報を使用する。

また、平衡結合角は入力構造の結合角を使用する。

参考論文

Wang et al. J Comput Chem 2004, 25, 1157

なおこの論文での式は-2乗と-1/2乗を間違っている。

$$\text{(誤)} K_{ijk}^0 = 143.9 Z_i C_j Z_k (r_{ij}^{eq} + r_{jk}^{eq})^{-1} (q_{ijk}^{eq})^2 \exp(-2D)$$

$$\text{(正)} K_{ijk}^0 = 143.9 Z_i C_j Z_k (r_{ij}^{eq} + r_{jk}^{eq})^{-1} (q_{ijk}^{eq})^{-1/2} \exp(-2D)$$

表 . 結合伸縮に関する力の定数を見積もる為のパラメータ一覧

No.	i	j	$\ln K_{ij}$	No.	i	j	$\ln K_{ij}$
1	H	H	4.661	27	C	S	8.117
2	C	C	7.643	28	N	O	7.526
3	N	N	7.634	29	N	F	7.475
4	O	O	7.561	30	N	Cl	8.266
5	F	F	7.358	31	N	Br	8.593
6	Cl	Cl	8.648	32	N	I	8.963
7	Br	Br	9.012	33	N	P	8.212
8	I	I	9.511	34	N	S	8.073
9	P	P	8.805	35	O	F	7.375
10	S	S	8.316	36	O	Cl	8.097
11	H	C	6.217	37	O	Br	8.276
12	H	N	6.057	38	O	I	8.854
13	H	O	5.794	39	O	P	7.957
14	H	F	5.6	40	O	S	7.922
15	H	Cl	6.937	41	F	Cl	7.947
16	H	Br	7.301	42	Cl	I	9.309
17	H	I	7.802	43	Br	I	9.38
18	H	P	7.257	44	F	P	7.592
19	H	S	7.018	45	F	S	7.733
20	C	N	7.504	46	Cl	P	8.656
21	C	O	7.347	47	Cl	S	8.619
22	C	F	7.227	48	Br	P	8.729
23	C	Cl	8.241	49	Br	S	8.728
24	C	Br	8.478	50	I	P	9.058
25	C	I	8.859	51	I	S	9.161
26	C	P	8.237	52	P	S	8.465

表 . 結合角の bend に関する力の定数を見積もる為のパラメータ一覧

Element	C	Z
H	-	0.784
C	1.339	1.183
N	1.300	1.212
O	1.249	1.219
F	-	1.166
Cl	-	1.272
Br	-	1.378
I	-	1.398
P	0.906	1.620
S	1.448	1.280

3-8 フラグメント DB 読み込み

フラグメント DB とは、3-6 節のアルゴリズムで原子タイプの割り当てができないものや、ユーザの定義したパラメータを使用したい低分子の官能基等を、それぞれ低分子のフラグメントとみなし、各種パラメータを登録した DB である。このフラグメント DB に登録してあるフラグメントと一致するパターンを計算対象分子内から見つけ、各種パラメータを割り当てる。

ここではフラグメント DB に定義されたフラグメントの以下の情報を読み込む。

- atom パラメータ
- bond パラメータ
- angle パラメータ
- torsion パラメータ
- improper-torsion パラメータ

以下に各フラグメント DB ファイルの一覧を示す。

frg_gaff.db	AMBER GAFF 力場用フラグメント DB ファイル
frg_amber99.db	AMBER parm99 力場用フラグメント DB ファイル

3-8-1 フラグメント DB のファイルフォーマットについて

以下にフラグメント DB ファイル例とフォーマットを示す。

フラグメント DB は以下の各種データブロックに分けて記述する。各ブロックは Start ブロック名で始まり、end ブロック名で終了する。

ブロック名	意味
fragment	各フラグメントの情報を記述
atom	フラグメント内の原子のパラメータを記述
bond	フラグメント内の結合のパラメータを記述
angle	フラグメント内の結合角のパラメータを記述
torsion	フラグメント内の二面角のパラメータを記述
improper	フラグメント内のインプロパー二面角のパラメータを記述

フォーマット)

Start ブロック名

end ブロック名

例)

Start fragment

:

Start atom

:

end atom

:

end fragment

(1) 各種ブロック共通フォーマット

- ・各パラメータは、“,” で区切られ、複数のパラメータを一行で記述することもできる。
- ・“;” 以降をコメント行とする。

(2) fragment ブロック

fragment ブロックで指定できるキーワードは以下の通りである。

キーワード	説明	実際の値の例
Fragment	フラグメント名	Benzene 等
ID	フラグメントの ID	1, 2, ...

また、fragment ブロック内には以下のブロックを含む。(は必須)

atom ブロック ()

bond ブロック ()

angle ブロック

torsion ブロック

improper ブロック

例)

```

Start fragment
  fragment= Benzene, ID=1,
  !
Start atom
  atomtype= ca ,mass= 12.01000 ,
  rvdw= 1.908000 ,evdw= 0.0860,
  atomno=1,
  atom= C,
  nbond= 3,
end atom
:
Start bond
  K= 473.700 ,
  Req= 1.3900,
  atomno(1)= 1 ,atomno(2)=2,
end bond
:
end fragment

```

(3)atom ブロック

atom ブロックで指定できるキーワードは以下の通りである。atom ブロックは fragment ブロック内に記述されなければならない。

キーワード	説明	実際の値の例
atomtype	原子タイプ	ca, na, ...
mass	原子の質量	12.011
rvdw	原子半径	1.908
evdw	電荷	0.086
atomno	原子の番号	1, 2, ...
atom	元素名	C, N, ...
nbond	結合の数	3, 4, ...

なお、フラグメントと入力分子構造のパターンマッチの条件として、元素名(atom)と結合の数(nbond)を用いる。元素名をパターンマッチの条件として使用したくない場合は、ワイルドカード X で指定する(atom=X)。また、結合の数をパターンマッチの条件として使用したくない場合は、nbond の項目を指定しない。

例)

```
Start atom
  atomtype= ca ,mass= 12.01000 ,
  rvdw= 1.908000 ,evdw= 0.0860,
  atomno=1,
  atom= C,
  nbond= 3,
end atom
```

(4)bond ブロック

bond ブロックで指定できるキーワードは以下の通りである。bond ブロックは fragment ブロック内に記述されなければならない。

キーワード	説明	実際の値の例
atomno(1)	結合している原子 1 の番号	1, 2, ...
atomno(2)	結合している原子 2 の番号	1, 2, ...
order(1,2)	結合次数の範囲	[1.7 < 3.0) 1.7 結合次数<3.0
K	バネ定数	473.7
Req	平衡結合距離	1.39

結合次数の範囲を指定は以下を用いて行う。

下限 : “ (” 境界を含まない
 “ [” 境界を含む
 上限と下限の区切り文字 : “ < ” (固定)
 “ > ” (未対応)
 上限 : “) ” 境界を含まない
 “] ” 境界を含む

なお、結合次数の範囲をここで指定した場合、パターンマッチの条件として使用される。

例)

```
Start bond
  K= 473.700 ,
  Req= 1.3900,
  atomno(1)= 1 ,atomno(2)=2,
  order(1,2)=[1.7<3.0),
end bond
```

(5) angle ブロック

angle ブロックで指定できるキーワードは以下の通りである。angle ブロックは fragment ブロック内に記述されなければならない。

キーワード	説明	実際の値の例
atomno(1)	結合角をなす原子 1 の番号	1, 2, ...
atomno(2)	結合角をなす原子 2 の番号 (結合角の中心)	1, 2, ...
atomno(3)	結合角をなす原子 3 の番号	1, 2, ...
K	バネ定数	63
Req	平衡結合角	120

例)

```

Start angle
      K=    63.000 ,
      Req=  120.000 ,
      atomno(1)= 6,atomno(2)=1,atomno(3)= 2,
end   angle

```

(6) torsion ブロック

torsion ブロックで指定できるキーワードは以下の通りである。torsion ブロックは fragment ブロック内に記述されなければならない。

キーワード	説明	実際の値の例
atomno(1)	二面角をなす原子 1 の番号	1, 2, ...
atomno(2)	二面角をなす原子 2 の番号	1, 2, ...
atomno(3)	二面角をなす原子 3 の番号	1, 2, ...
atomno(4)	二面角をなす原子 4 の番号	1, 2, ...
no	二面角パラメータの数	1, 2, ...
pn(n)	回転周期	2
phase(n)	位相	180
pk(n)	バリアー高	12
idivf(n)	割る値	

カッコ内の n は torsion パラメータの通し番号

例)

```
Start torsion
  no= 1 ,
  pn( 1)= 2 ,
  phase( 1)= 180.000 ,
  pk( 1)= 5.000 ,
  idivf( 1)= 2 ,
  !
  atomno(1)= 1 ,atomno(2)=2,atomno(3)=3,atomno(4)= 4,
end torsion
```

(7) improper ブロック

improper ブロックで指定できるキーワードは以下の通りである。improper ブロックは fragment ブロック内に記述されなければならない。

キーワード	説明	実際の値の例
atomno(1)	インプロパー二面角をなす原子 1 の番号	1, 2, ...
atomno(2)	インプロパー二面角をなす原子 2 の番号	1, 2, ...
atomno(3)	インプロパー二面角をなす原子 3 の番号 (インプロパー二面角の中心)	1, 2, ...
atomno(4)	インプロパー二面角をなす原子 4 の番号	1, 2, ...
no	インプロパー二面角パラメータの数	1, 2, ...
pn(n)	回転周期	2
phase(n)	位相	180
pk(n)	バリアー高	12
idivf(n)	割る値	2

カッコ内の n は improper
パラメータの通し番号

例)

```

Start improper
  no= 1 ,
    pn( 1)= 2 ,
    phase( 1)= 180.000 ,
    pk( 1)= 1.100 ,
    idivf( 1)= 1 ,
  !
  atomno(1)= 1 ,atomno(2)= 2 ,atomno(3)=3,atomno(4)= 4,
end improper

```

3-9 フラグメント DB パラメータ割り当て

(1) パターンマッチアルゴリズム

フラグメント DB に登録されたフラグメントの構成元素名、結合情報、結合次数 ()、結合の数 ()、環上に含まれるか否かを示すフラグ(便宜的に環フラグとする) ()をもとにパターンマッチ判定を行う。

() は必須項目ではない。フラグメント DB 内で定義されている場合のみ、判定条件に加える。))

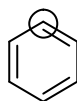
以下の手順で各フラグメントに対するパターンマッチを行う。

例として、以下の構造をフラグメントに登録した際の、パターンマッチ判定の流れを示す。

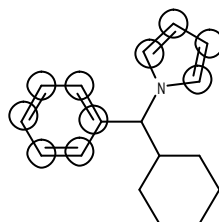


- ・ベンゼンの環構造のみ (Hは登録していない)
- ・各原子の結合の数 3
- ・結合次数の範囲を芳香族結合に設定

計算対象分子から、フラグメントの最初に定義されている原子の元素名と環フラグと結合の数が一致する原子を探し、その原子をフラグメントとの一致候補原子とする。

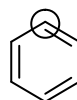


フラグメント

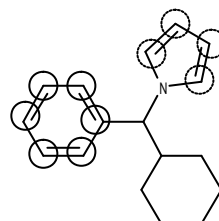


計算対象分子

着目原子 () と結合している原子の元素名、環フラグ、結合の数、結合次数がフラグメントと一致しない候補原子団を一致候補からはずす。



フラグメント



計算対象分子

フラグメント内の全ての構成原子をチェックし終えるまで、着目原子をその原子の結合している次の原子に移し、 の操作を行う。

フラグメント内の全て構成原子と一致するパターンが計算対象分子内に見つかった場合、フラグメント DB に登録されているパラメータを割り当てる。

(2) パターンのマッチしたフラグメントのパラメータ割り当て

(1) でパターンのマッチしたフラグメントにフラグメント DB で定義された以下のパラ

メータを割り当てる。定義されていないパラメータについては割り当てない。

- atom パラメータ
- bond パラメータ
- angle パラメータ
- torsion パラメータ
- improper-torsion パラメータ

3-10 トポロジーファイル出力

作成したトポロジー情報を出力する。トポロジーファイルは以下の項目を含む。

bond、angle、torsion、improper-torsion 項において、デフォルトパラメータ、または、動的補填パラメータが記述されている場合には、その項目のコメント(;以降の部分)にアスタリスク(*)を表示する。

atom 項
bond 項
angle 項
torsion 項
improper-torsion 項
nonbond 項

3-11 PDB ファイル出力

Sybyl mol2 入力ファイルで tplgeneL を実行した場合には、トポロジーファイルを作成するとともに、PDB ファイルも作成する。

作成される PDB ファイル名は XXX_tpIL.pdb(XXX は入力ファイルの拡張子を除くファイル名)である。

4 . tplgeneL 関数機能説明

以下に tplgeneL の処理の流れ、及び、各関数の構成図を示す(図4 - 1、図4 - 2 参照)。

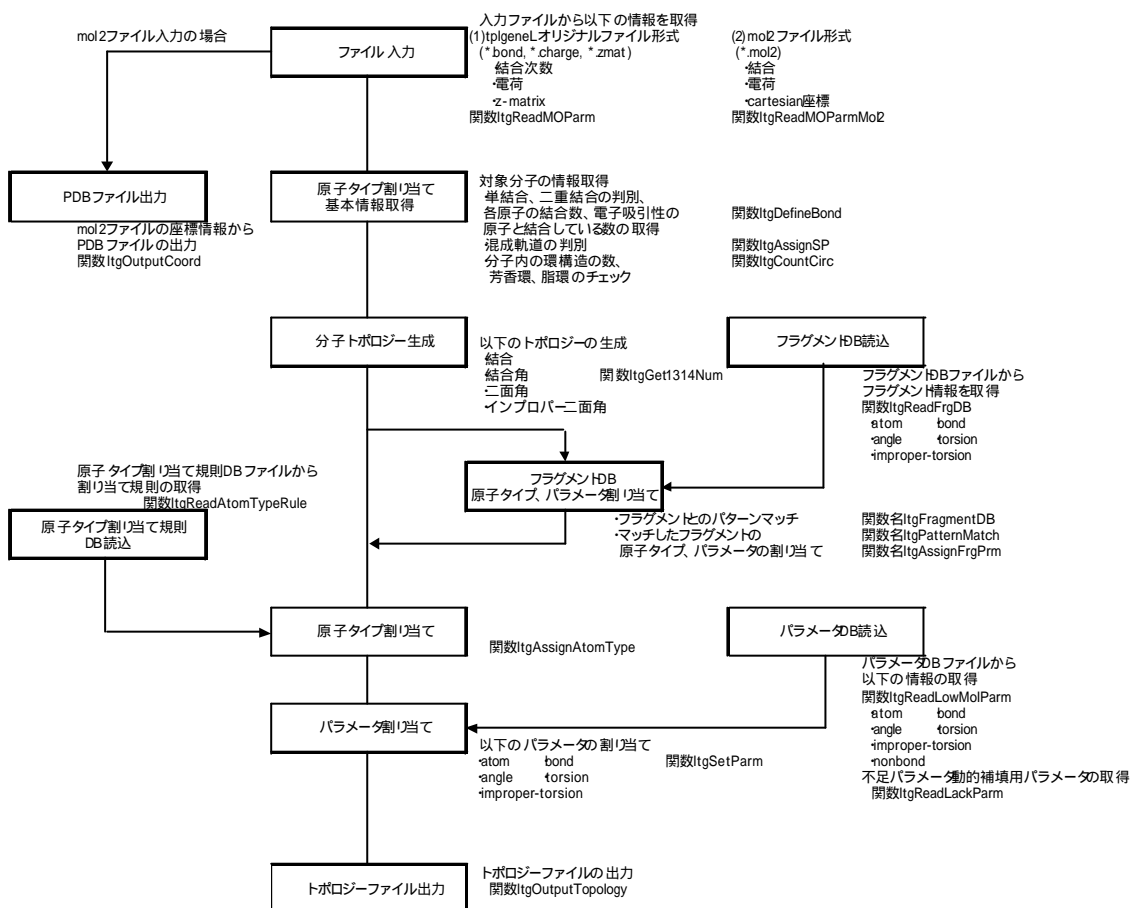


図4 - 1 . tplgeneL の処理の流れ

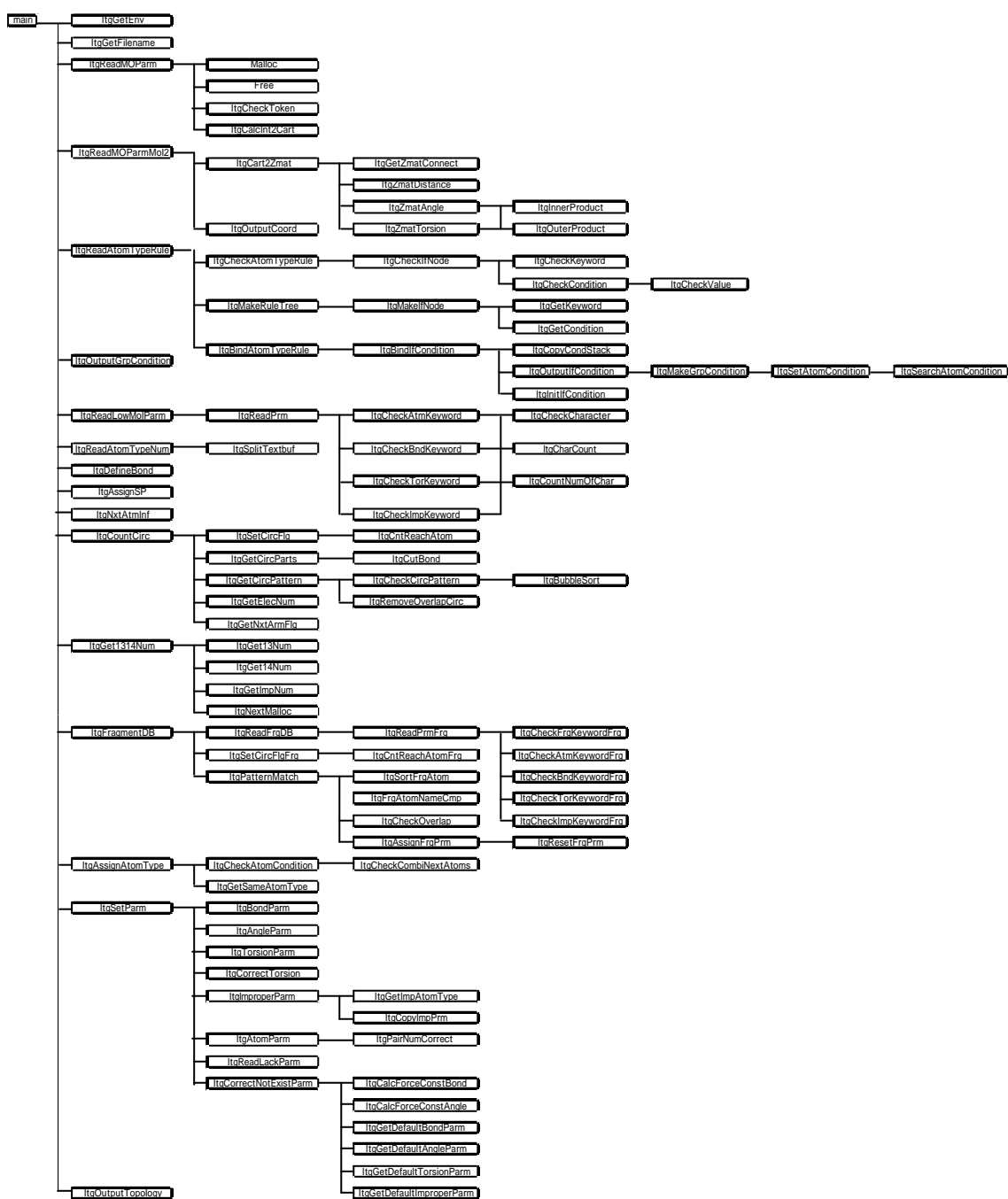


図 4 - 2 . tplgeneL の関数構成図

4-1 main

機能概要

- tplgeneL のメインプログラム

機能説明

- 環境変数から、入力、出力データの PATH を取得する(関数 ItgGetEnv を呼び出す)。
- 入力データのファイル名を取得する(関数 ItgGetFilename を呼び出す)。
- 入力ファイル形式がオリジナルフォーマットの場合の読み込み処理を行なう(関数 ItgReadMOParm を呼び出す)。
- 入力ファイル形式が mol2 フォーマットの場合の読み込み処理を行なう(関数 ItgReadMOParmMol2 を呼び出す)。
- 選択した力場情報に基づく原子タイプ割り当て規則 DB ファイル名、及び、nonbond ファイル名の取得を行なう。
- 低分子パラメータ情報の取得を行なう(関数 ItgReadLowMolPrm を呼び出す)。
- 原子タイプ番号を取得する(関数 ItgReadAtomTypeNum を呼び出す)。
- 入力構造の各結合の結合次数を取得する(関数 ItgDefineBond を呼び出す)。
- 入力構造の各原子の混成軌道を取得する(関数 ItgAssignSP を呼び出す)。
- 入力構造の各原子の隣の原子の情報を保存する(関数 ItgNxtAtmInf を呼び出す)。
- 環構造に関する処理を行なう(関数 ItgCountCirc を呼び出す)。
- 結合、結合角、二面角情報を作成する(関数 ItgGet1314Num を呼び出す)。
- フラグメント DB を用いたパターンマッチ、原子タイプの割り当てを行なう(関数 ItgFragmentDB を呼び出す)。
- 原子タイプを割り当てる(関数 ItgAssignAtomType を呼び出す)。
- 結合パラメータを割り当てる(関数 ItgBondParm を呼び出す)。
- 関数 ItgAngleParm を呼び出す。
- 関数 ItgTorsionParm を呼び出す。
- 関数 ItgCorrectTorsion を呼び出す。
- 関数 ItgImproperParm を呼び出す。
- 関数 ItgAtomParm を呼び出す。
- 関数 ItgCorrectNotExistParm を呼び出す。
- 関数 ItgOutputTopology を呼び出す。

引数

```
int argc  
char *argv[]
```

備考

なし。

4-2 ItgGetEnv

機能概要

- tplgeneL で使用する環境変数を取得する。

機能説明

- 環境変数 TPLL_DB_PATH から、パラメータ DB ファイルの置いてあるディレクトリへの PATH を取得する。
- 環境変数 TPLL_INPUT_PATH から、入力ファイルの置いてあるディレクトリへの PATH を取得する。
- 環境変数 TPLL_OUTPUT_PATH から、出力ファイルを出力するディレクトリへの PATH を取得する。

引数

char DB_PATH[], INPUT_PATH[], OUTPUT_PATH[]

備考

なし。

4-3 ItgGetFilename

機能概要

- ・ tplgeneL で使用するファイル名、計算方法などを取得する。

機能説明

<対話的実行時>

- ・ 入力ファイル名を対話的に取得する。
- ・ 入力ファイル名の種類 (tplgeneL オリジナル入力ファイル、Sybyl mol2 入力ファイル) を対話的に取得する。
- ・ パラメータ DB ファイル名を対話的に取得する。
- ・ フラグメント DB 機能を使用するか否かの情報を対話的に取得する。
- ・ 不足パラメータの補填方法を対話的に取得する。

<オプション指定実行時>

- ・ 実行時、オプションでヘルプメッセージの表示が指定されている場合、ヘルプメッセージを表示する。
- ・ 実行時、オプションで入力ファイル名が指定されている場合、入力ファイル名を取得する。
- ・ 実行時、オプションで入力ファイル名の種類が指定されている場合、入力ファイル名の種類を取得する。
- ・ 実行時、オプションでパラメータ DB ファイル名が指定されている場合、パラメータ DB ファイル名を取得する。
- ・ 実行時、オプションで残基名が指定されている場合、残基名を取得する。
- ・ 実行時、オプションでフラグメント DB 機能を使用するか否かの情報が指定されている場合、その情報を取得する。
- ・ 実行時、オプションで不足パラメータの補填方法が指定されている場合、補填方法を取得する。

なお、オプションで指定していない項目については実行時に対話的に入力する。

引数

```
int      argc, *filetype, *frgdb_used_flg,  
char     *argv[], *in_filename, char *out_filename,  
char     *db_filename, *frg_filename,  
char     *resname, int *prm
```

備考

なし。

4-4 ltgReadMOParm

機能概要

- ・電荷情報、結合次数情報、Z-matrix 情報の読み込みを行う。

機能説明

- ・電荷情報ファイル (XXX.charge) をオープンする。
- ・電荷情報ファイルから原子名、元素記号、Mulliken 電荷、resp 電荷、原子タイプ情報の読み込みを行う。
- ・結合次数情報 (XXX.bond) をオープンする。
- ・結合次数情報ファイルから結合原子のペアの番号、結合距離、結合次数情報の読み込みを行う。
- ・Z-matrix 情報ファイル (XXX.zmat) が存在する場合には、オープンする。
- ・Z-matrix 情報の読み込みを行う。
- ・メモリ領域確保の際に、関数 Malloc を呼び出す。

引数

LTGSystemPtr	sys
char	*filename

備考

なし。

4-5 Malloc

機能概要

- ・メモリ領域を確保する。

機能説明

- ・メモリ領域を確保する。
- ・メモリ領域を 0 で埋める。

引数

int size

備考

なし。

4-6 Free

機能概要

- ・確保していたメモリ領域を解放する。

機能説明

- ・確保していたメモリ領域を解放する。

引数

void *ptr

備考

なし

4-7 ItgCheckToken

機能概要

- ・関数 ItgReadMOParm のサブ関数
- ・引数として渡された文字列の型を判別する。

機能説明

- ・引数として渡された文字列がアルファベットを含む場合、“文字列型”を返す。
- ・引数として渡された文字列がアルファベットを含まず、文字 ' - '、' . ' を含む場合、“実数型”を返す。
- ・引数として渡された文字列にアルファベット、文字 ' - '、' . ' を含まない場合、“整数型”を返す。

引数

char *p

備考

なし

4-8 ItgCalcInt2Cart

機能概要

- ・関数 ItgReadMOParm のサブ関数
- ・内部座標からカーテシアン座標を計算する。

機能説明

- ・1番目の原子をカーテシアン座標の原点にセットする。
- ・2番目の原子をカーテシアン座標の X 軸上にセットする。
- ・3番目の原子を X-Y 平面上にセットする。
- ・4番目以降の原子について内部座標からカーテシアン座標への変換行列を作成する。
- ・4番目以降の原子を作成した変換行列でカーテシアン座標に変換する。

引数

LTGSystemPtr sys

備考

なし

4-5 ItgReadMOParmMol2

機能概要

- mol2 形式ファイルから電荷情報、結合、座標情報の読み込みを行う。
- mol2 形式ファイルの座標情報を PDB 形式ファイルに出力する。
- 読み込んだ座標情報から Z-matrix 座標を計算する。

機能説明

- mol2 形式ファイル (XXX.mol2) をオープンする。
- mol2 ファイル内のキーワード “ @<TRIPOS>ATOM ” を取得する。
- キーワード “ @<TRIPOS>ATOM ” 以下の行から原子名、元素記号、電荷、結合、座標情報の読み込みを行う。
- mol2 ファイル内のキーワード “ @<TRIPOS>BOND ” を取得する。
- キーワード “ @<TRIPOS>BOND ” 以下の行から結合する原子の組、及び、結合次数を取得する。
- カートesian座標から Z-matrix 座標を作成する(関数 ItgCart2Zmat を呼び出す)。
- PDB 形式ファイルの出力を行う(関数 ItgOutputCoordinate を呼び出す)。

引数

LTGSystemPtr	sys
char	*filename, *output_filename, *rename

備考

なし。

4-6 ItgCart2Zmat

機能概要

- ・カーテシアン座標から Z-matrix 情報を計算する。

機能説明

- ・カーテシアン座標から Z-matrix 情報を計算する。
- ・Z-matrix を作成する為の原子の connect 情報を作成する(関数 ItgGetZmatConnect を呼び出す)。
- ・原子間距離を計算する(関数 ItgZmatDistance を呼び出す)。
- ・結合角を計算する(関数 ItgZmatAngle を呼び出す)。
- ・二面角を計算する(関数 ItgZmatTorsion を呼び出す)。

引数

LTGSystemPtr sys

備考

なし。

4-11 ItgGetZmatConnect

機能概要

- ・関数 ItgReadMOParmMol2 のサブ関数である。
- ・内部座標の結合情報の基準原子を取得する。通常は結合している原子の番号 (num_bond)、結合角を構成する原子の番号 (num_angle)、二面角を構成する原子の番号 (num_torsion) を取得する。

機能説明

- ・1 番目の原子は num_bond、num_angle、num_torsion に 0 を代入する。
- ・2 番目の原子の num_bond を 1 とする。num_angle、num_torsion に 0 を代入する。
- ・3 番目の原子の num_bond に値を代入する。
1 番目の原子と結合している場合には 1 を、2 番目の原子と結合している場合には 2 を代入する。
- ・3 番目の原子の num_angle に値を代入する。
1 番目の原子と結合している場合には 2 を、2 番目の原子と結合している場合には 1 を代入する。
- ・3 番目の原子の num_torsion に 0 を代入する。
- ・n 番目 (n>=4) の原子の num_bond に値を代入する。
n-1 番目までの原子と結合している場合はその原子の番号を代入する。
n-1 番目までの原子と結合していない場合は n-1 を代入する。
- ・n 番目 (n>=4) の原子の num_angle に値を代入する。
n-1 番目までの原子と結合している場合には、その原子の num_bond の値を代入する。
n-1 番目までの原子と結合していない場合には、n-1 番目の num_bond の値を代入する。
- ・n 番目 (n>=4) の num_torsion に値を代入する。
n-1 番目までの原子と結合している場合には、その原子の num_angle の値を代入する。
n-1 番目までの原子と結合していない場合には、n-1 番目の num_angle の値を代入する。

引数

LTGSystemPtr sys

備考

なし。

4-12 ltgZmatDistance

機能概要

- ・ 結合距離を計算する。

機能説明

- ・ 2点間の結合距離を計算する。

引数

LTGZMatPtr	atom1
LTGZMatPtr	atom2

備考

なし。

4-13 ItgZmatAngle

機能概要

- ・結合角を計算する。

機能説明

- ・結合角を計算する。
- ・内積の計算を行う(関数 ItgInnerProduct を呼び出す)。
- ・原子間距離を計算する(関数 ItgZmatDistance を呼び出す)。

引数

LTGZMatPtr	atom1
LTGZMatPtr	atom2
LTGZMatPtr	atom3

備考

なし。

4-14 ItgInnerProduct

機能概要

- ・二つのベクトルの内積を計算する。

機能説明

- ・二つのベクトルの内積を計算する。

引数

LTGZMatPtr	atom1
LTGZMatPtr	atom2

備考

なし。

4-15 ItgZmatTorsion

機能概要

- ・二面角を計算する。

機能説明

- ・二面角を計算する。
- ・外積の計算を行う(関数 ItgOuterProduct を呼び出す)。
- ・内積の計算を行う(関数 ItgInnerProduct を呼び出す)。
- ・結合距離を求める(関数 ItgZmatDistance を呼び出す)。

引数

LTGZMatPtr	atom1
LTGZMatPtr	atom2
LTGZMatPtr	atom3
LTGZMatPtr	atom4

備考

なし。

4-16 ltgOuterProduct

機能概要

- ・二つのベクトルの外積を計算する。

機能説明

- ・二つのベクトルの外積を計算する。

引数

LTGZMatPtr	atom1
LTGZMatPtr	atom2
LTGZMatPtr	norm

備考

なし。

4-17 ltgOutputCoord

機能概要

- ・カーテシアン座標情報を PDB 形式のファイルに出力する。

機能説明

- ・ファイルをオープンする。
- ・カーテシアン座標情報を PDB 形式のファイルに出力する。
- ・出力情報は、原子名、残基名、座標情報、原子量、電荷である。

引数

LTGSystemPtr	sys
char	*filename, *resname

備考

なし。

4-18 ItgReadAtomTypeRule

機能概要

- ・原子タイプ割り当て規則 DB を読み込み、原子タイプ割り当て規則を取得する。

機能説明

- ・原子タイプ割り当て規則 DB ファイルをオープンする。
- ・原子タイプ割り当て規則 DB ファイルを配列に読み込む。
- ・原子タイプ割り当て規則 DB ファイルのフォーマットチェックのため、関数 ItgCheckAtomTypeRule を呼び出す。
- ・関数 ItgBindAtomTypeRule を呼び出し、原子タイプ割り当て規則の構文木を作成する。
- ・関数 ItgBindAtomTypeRule を呼び出し、原子タイプ毎に割り当て規則まとめる。

引数

PRMIfNodePtr	root
PRMIfFramePtr	head
char	*fname, *DB_PATH

備考

なし。

4-19 ItgCheckAtomTypeRule

機能概要

- ・原子タイプ割り当て規則の “ begin ” から対応する “ end ” までのフォーマットのチェックを行う。

機能説明

- ・キーワードを取得するため、関数 ItgCheckKeyword を呼び出す。
- ・関数 ItgCheckKeyword の戻り値が負の場合、エラー処理を行う。
- ・“ begin ” が無い場合、エラー処理を行う。
- ・“ begin ” に対応する “ end ” が無い場合、エラー処理を行う。
- ・“ begin ” から “ end ” の間に “ begin ” が存在する場合、エラー処理を行う。
- ・原子タイプ代入文の以下のフォーマットのチェックを行い、違反する場合にはエラー処理を行う。
 - ・代入演算子 “ := ” を使用しているか。
 - ・改行を示す “ ; ” を使用しているか。
 - ・原子タイプが記述されているか。
 - ・原子タイプが原子タイプの最大文字数 ATOM_TYPE_MAX を越えていないか。
 - ・原子タイプ代入文が “ then ” の直後以外で用いられていないか。
- ・“ begin ” から対応する “ end ” の間に “ if ” が存在する場合、関数 ItgCheckIfNode を呼び出す。

引数

char	textbuf[][BUFSIZE+1]
int	nline

備考

なし。

4-20 ItgCheckIfNode

機能概要

- ・関数 ItgCheckAtomTypeRule のサブ関数
- ・原子タイプ割り当て規則の “ if ” から対応する “ endif ” までのフォーマットのチェックを行う。

機能説明

- ・“ if ” から“ then ” までの各条件をチェックする関数 ItgCheckCondition を呼び出す。
- ・関数 ItgCheckCondition の戻り値が負の場合、エラー処理を行う。
- ・“ if ” から “ endif ” の間に “ and ” で区切らず、複数の条件を記述していた場合、エラー処理を行う。
- ・キーワードを取得するため、関数 ItgCheckKeyword を呼び出す。
- ・関数 ItgCheckKeyword の戻り値が負の場合、エラー処理を行う。
- ・原子タイプ代入文の以下のフォーマットのチェックを行う。
 - ・代入演算子 “ := ” を使用しているか。
 - ・改行を示す “ ; ” を使用しているか。
 - ・原子タイプが記述されているか。
 - ・原子タイプが原子タイプの最大文字数 ATOM_TYPE_MAX(8)を越えていないか。
 - ・原子タイプ代入文が “ then ” の直後以外で用いられていないか。
- ・“ if ” から対応する “ endif ” の間に “ if ” が存在する場合、本関数 ItgCheckIfNode を再帰的に呼び出す。

引数

char	textbuf[][BUFSIZE+1]
int	*line, start, nline

備考

なし。

4-21 ItgCheckKeyword

機能概要

- ・関数 ItgCheckAtomTypeRule のサブ関数
- ・引数として渡された、原子タイプ割り当て規則 DB 中の文字列からキーワードを抽出する。

機能説明

- ・キーワードを ' ('、あるいは、空白、タブまで、読み込み変数に保存する。
- ・保存するキーワードの文字数がキーワード最大文字数 KEYWORD_MAX-1(31)を越える場合、-1 を返り値として呼び出し元関数に返す。
- ・キーワードに正常に読み込めた場合、読み込んだところまでの文字数を返り値として呼び出し元関数に返す。

引数

char	*keyword
char	*line

備考

なし。

4-22 ItgCheckCondition

機能概要

- ・関数 ItgCheckAtomTypeRule のサブ関数
- ・引数として渡された、原子タイプ割り当て規則 DB 中の条件文の文字列から各条件を抽出する。

機能説明

- ・各条件(element(0)=1 など)の途中で改行が入っている場合、-1 を返り値として呼び出し元関数に返す。
- ・原子指定番号が ' (, ') ' で囲まれていない場合、-1 を返り値として呼び出し元関数に返す。
- ・条件に指定する値が、条件で指定できる最大文字数 COND_VAL_MAX より長い文字列の場合、-1 を返り値として呼び出し元関数に返す。
- ・原子指定番号について、以下のチェックを行い、違反している場合はエラー処理を行う。
 - ・原子指定番号最大文字数 NEIGHBOR_NUM-1(9)より長い文字列で記述されていないか。
 - ・原子指定番号が数字であるか。
 - ・原子指定番号が " 0 " から始まっているか。
 - ・原子指定番号 " 0 " が先頭以外で使用されていないか。
 - ・条件 " border " の原子指定番号として一桁の数字が使用されていないか。
- ・条件を指定する演算子について、以下のチェックを行い、違反している場合はエラー処理を行う。
 - ・演算子に " = ", " < ", " <= ", " > ", " >= " 以外が使用されていないか。
 - ・条件 " element ", " nbond ", " hybrid ", " circ ", " aromatic " に " = " 以外の演算子が使用されていないか。
- ・条件に指定する値について、以下のチェックを行い、違反している場合はエラー処理を行う。
 - ・条件 " circ ", " aromatic " に 0, 1 以外の値が指定されていないか。
 - ・条件 " nbond ", " nelectrwd ", " ring " に正の整数以外が指定されていないか。
 - ・条件 " border " に 1, 1.5, 2, 3 以外の数字が指定されていないか。

引数

char	*keyword
char	*linebuf
int	col
int	*line_num

備考

なし。

4-23 ItgCheckValue

機能概要

- ・関数 ItgCheckAtomTypeRule のサブ関数
- ・引数として渡された文字列の型のチェックを行う。

機能説明

- ・引数として渡された文字列の型が “ string ” の場合、1 を返り値として呼び出し元関数に返す。
- ・引数として渡された文字列の型が “ int ” で、
 - ・文字列中に数字以外の文字が無かった場合、1 を返り値として呼び出し元関数に返す。
 - ・文字列中に数字以外の文字があった場合、-1 を返り値として呼び出し元関数に返す。
- ・引数として渡された文字列の型が “ double ” で、
 - ・数字と ‘ . ’ 以外の文字が無かった場合、1 を返り値として呼び出し元関数に返す。
 - ・数字と ‘ . ’ 以外の文字があった場合、-1 を返り値として呼び出し元関数に返す。
- ・引数として渡された文字列が上記以外の場合、-1 を返り値として呼び出し元関数に返す。

引数

char	*value
char	*type

備考

なし。

4-24 ItgMakeRuleTree

機能概要

- ・関数 ItgReadAtomTypeRule のサブ関数
- ・原子タイプ割り当て規則の “begin” から対応する “end” までの構文木作成処理を行う。

機能説明

- ・キーワードを取得するため、関数 ItgGetKeyword を呼び出す。
- ・原子タイプ、デフォルト原子タイプの代入文が存在する場合は原子タイプ、デフォルト原子タイプを保存する。
- ・“begin” から対応する “end” の間に “if” が存在する場合、新しい構文木用構造体を生成し、以下の処理を行い、関数 ItgMakeIfNode を呼び出す。
 - ・参照している構文木の構造体のポインタ then_ptr に何もつながっていない場合、ポインタ then_ptr に生成した構造体をつなぐ。
 - ・参照している構文木の構造体のポインタ then_ptr に既につながっているものがある場合、ポインタ then_ptr につながっている構造体のポインタ if_ptr の一番最後に生成した構造体をつなぐ。

引数

PRMIfNodePtr	root
char	textbuf[][BUFSIZE+1]
int	nline

備考

なし。

4-25 ItgMakeIfNode

機能概要

- ・関数 ItgReadAtomTypeRule のサブ関数
- ・原子タイプ割り当て規則の “ if ” から対応する “ endif ” までの構文木作成処理を行う。

機能説明

- ・“ if ” から “ then ” までの各条件を読み込む関数 ItgGetCondition を呼び出す。
- ・キーワードを取得するため、関数 ItgGetKeyword を呼び出す。
- ・原子タイプ、デフォルト原子タイプの代入文が存在する場合は原子タイプ、デフォルト原子タイプを保存する。
- ・“ if ” から対応する “ endif ” の間に “ if ” が存在する場合、新しい構文木用構造体を生成し、以下の処理を行い、本関数 ItgMakeIfNode を再帰的に呼び出す。
 - ・参照している構文木の構造体のポインタ then_ptr に何もつながっていない場合、ポインタ then_ptr に生成した構造体をつなぐ。
 - ・参照している構文木の構造体のポインタ then_ptr に既につながっているものがある場合、ポインタ then_ptr につながっている構造体のポインタ if_ptr の一番最後に生成した構造体をつなぐ。

引数

PRMIfNodePtr	cond
char	textbuf[][BUFSIZE+1]
int	*line, start, nline

備考

なし。

4-26 ItgGetKeyword

機能概要

- ・関数 ItgReadAtomTypeRule のサブ関数
- ・引数として渡された、原子タイプ割り当て規則 DB 中の文字列からキーワードを抽出する。

機能説明

- ・キーワードを ' ('、あるいは、空白、タブまで、読み込み変数に保存する。
- ・保存するキーワードの文字数がキーワード最大文字数 KEYWORD_MAX-1(31)を越える場合、-1 を返り値として呼び出し元関数に返す。
- ・キーワードに正常に読み込めた場合、読み込んだところまでの文字数を返り値として呼び出し元関数に返す。

引数

char *keyword
char *line

備考

なし。

4-27 ItgGetCondition

機能概要

- ・関数 ItgReadAtomTypeRule のサブ関数
- ・引数として渡された、原子タイプ割り当て規則 DB 中の条件文の文字列から各条件を抽出する。

機能説明

- ・“elemnt”に関する条件がある場合、“element”の条件として、原子指定番号、演算子、指定した値を保存する。
- ・“nbond”に関する条件がある場合、“nbond”の条件として、原子指定番号、演算子、指定した値を保存する。
- ・“hybrid”に関する条件がある場合、“hybrid”の条件として、原子指定番号、演算子、指定した値を保存する。
- ・“border”に関する条件がある場合、“border”の条件として、原子指定番号、演算子、指定した値を保存する。
- ・“aromatic”に関する条件がある場合、“aromatic”の条件として、原子指定番号、演算子、指定した値を保存する。
- ・“circ”に関する条件がある場合、“circ”の条件として、原子指定番号、演算子、指定した値を保存する。
- ・“nelectrwd”に関する条件がある場合、“nelectrwd”の条件として、原子指定番号、演算子、指定した値を保存する。
- ・“ring”に関する条件がある場合、“ring”の条件として、原子指定番号、演算子、指定した値を保存する。

引数

PRMIfNodePtr	cond
char	*keyword
char	*line

備考

なし。

4-28 ItgBindAtomTypeRule

機能概要

- ・原子タイプ割り当て規則の構文木から原子タイプ毎に条件をまとめる。

機能説明

- ・構文木の root から原子タイプ毎に条件をまとめる関数 ItgBindIfCondition を呼び出す。

引数

PRMIfNodePtr	root
PRMIfFramePtr	head

備考

なし。

4-29 ItgBindIfCondition

機能概要

- ・関数 ItgBindAtomTypeRule のサブ関数
- ・原子タイプ割り当て規則の構文木から原子タイプ毎に条件をまとめる。

機能説明

- ・関数 ItgCopyCondStack を呼び出し、参照している構文木の構造体の条件とデフォルト原子タイプを配列に保存する。
- ・参照している構文木の構造体に原子タイプが保存されている場合、関数 ItgOutputIfCondition を呼び出し、配列に保存されている情報を出力する。
- ・参照している構文木の構造体のポインタ then_ptr につながっているものがある場合、つながっている構造体を参照し、本関数 ItgBindIfCondition を再帰的に呼び出す。
- ・ポインタ then_ptr につながっている構造体に対する処理が終わった、あるいは、ポインタ then_ptr につながっている構造体が無かった場合、関数 ItgInitIfCondition を呼び出し、本関数で保存した条件とデフォルト原子タイプを配列から削除する。
- ・参照している構文木の構造体のポインタ if_ptr につながっているものがある場合、つながっている構造体を参照し、本関数 ItgBindIfCondition を再帰的に呼び出す。

引数

PRMIfNodePtr	cond
PRMIfCondition	*stack
int	stack_num
PRMIfFramePtr	if_frame

備考

なし。

4-30 ItgCopyCondStack

機能概要

- ・関数 ItgBindAtomTypeRule のサブ関数
- ・参照している原子タイプ割り当て規則の構文木から条件とデフォルト原子タイプを抽出し、配列に保存する。

機能説明

- ・“elemnt”に関する条件がある場合、“element”の条件として、原子指定番号、演算子、指定した値を配列に保存する。
- ・“nbond”に関する条件がある場合、“nbond”の条件として、原子指定番号、演算子、指定した値を配列に保存する。
- ・“hybrid”に関する条件がある場合、“hybrid”の条件として、原子指定番号、演算子、指定した値を配列に保存する。
- ・“border”に関する条件がある場合、“border”の条件として、原子指定番号、演算子、指定した値を配列に保存する。
- ・“aromatic”に関する条件がある場合、“aromatic”の条件として、原子指定番号、演算子、指定した値を配列に保存する。
- ・“circ”に関する条件がある場合、“circ”の条件として、原子指定番号、演算子、指定した値を配列に保存する。
- ・“nelectrwd”に関する条件がある場合、“nelectrwd”の条件として、原子指定番号、演算子、指定した値を配列に保存する。
- ・“ring”に関する条件がある場合、“ring”の条件として、原子指定番号、演算子、指定した値を配列に保存する。

引数

PRMIfNodePtr	cond
PRMIfCondition	*stack
int	stack_num

備考

なし。

4-31 ItgOutputIfCondition

機能概要

- ・関数 ItgBindAtomTypeRule のサブ関数
- ・原子タイプ毎の条件を原子指定番号毎にグループ化する関数 ItgMakeGrpCondition に渡す。

機能説明

- ・引数として渡された、条件を保存している配列から、もっとも後ろに保存されているデフォルト原子タイプを取得し、原子タイプとデフォルト原子タイプを if 文の骨格用構造体に保存する。
- ・引数として渡された条件を保存する配列の全条件を関数 ItgMakeGrpCondition に渡し、原子指定番号毎のグループ化を行う。

引数

PRMIfNodePtr	cond
PRMIfCondition	*stack
int	stack_num
PRMIfFramePtr	if_frame

備考

なし。

4-32 ItgMakeGrpCondition

機能概要

- ・原子タイプ毎の条件を原子指定番号毎にグループ化する。

機能説明

- ・引数として渡された各条件に対して、関数 ItgSetAtomCondition を呼び出し、原子指定番号毎にグループ化し保存する。

引数

PRMIfFramePtr	if_frame
PRMIfConditionPtr	if_cond

備考

なし。

4-33 ItgSetAtomCondition

機能概要

- ・関数 ItgMakeGrpCondition のサブ関数
- ・原子タイプ毎の条件を原子指定番号毎に保存する。

機能説明

- ・引数として渡された条件の原子指定番号を取得し、関数 ItgSearchAtomCondition を呼び出し、取得した原子指定番号の条件を保存する構造体を探す。
- ・条件を取得した構造体に保存する。

引数

PRMIfFramePtr	if_frame
PRMIfConditionPtr	if_cond
char	*cond_type
int	idx_cond

備考

なし。

4-34 ItgSearchAtomCondition

機能概要

- ・関数 ItgMakeGrpCondition のサブ関数
- ・原子タイプ毎の条件を原子指定番号毎に保存する構造体を探す。

機能説明

- ・引数として渡された原子指定番号の原子指定番号毎条件用構造体が存在する場合、その構造体のアドレスを返り値として呼び出し元関数に返す。
 - ・引数として渡された原子指定番号の原子指定番号毎条件用構造体が存在しない場合、その構造体を生成し、以下の処理を行い、生成した構造体のアドレスを呼び出し元関数に返す。
 - a) 上位の(原子指定番号の末尾を1つ除いた)構造体に、生成した構造体と同じ階層の構造体がつながっていない場合、上位の構造体と生成した構造体をつなく。
 - b) 上位の構造体に、生成した構造体と同じ階層^(注1)の構造体が既につながっている場合、同じ階層の構造体の末尾と生成した構造体をつなく
この際、生成した構造体より上の階層の構造体が無かった場合、その上位の構造体も生成する。
- 注1) 階層とは着目原子からの位置関係を示し、上の階層とは参照している原子より着目原子に近い原子、下の階層とは参照している原子より着目原子から遠い原子を指す。

引数

PRMIfFramePtr	if_frame
char	*specified_num

備考

なし。

4-35 ItgInitIfCondition

機能概要

- ・関数 ItgBindAtomTypeRule のサブ関数
- ・原子タイプ割り当て条件を削除する。

機能説明

- ・引数として渡された原子タイプの割り当て条件とデフォルト原子タイプを削除する。

引数

PRMIfConditionPtr cond

備考

なし。

4-36 ItgOutputGrpCondition

機能概要

- ・関数 ItgMakeGrpCondition のデバッグ用サブ関数
- ・原子指定番号毎にグループ化された原子タイプ割り当て規則をファイルに出力する。

機能説明

- ・各条件の原子タイプ、デフォルト原子タイプを出力する。
- ・関数 ItgOutputGrpCondition を呼び出し、各原子指定番号毎にグループ化された条件を出力する。

引数

PRMIfFramePtr head

備考

なし。

4-37 ItgOutputAtomCondition

機能概要

- ・関数 ItgMakeGrpCondition のデバッグ用サブ関数
- ・原子指定番号毎にグループ化された原子タイプ割り当て規則をファイルに出力する。

機能説明

- ・構造体に保存されている原子指定番号を出力する。
- ・構造体に保存されている “ element ” の条件を出力する。
- ・構造体に保存されている “ nbond ” の条件を出力する。
- ・構造体に保存されている “ hybrid ” の条件を出力する。
- ・構造体に保存されている “ border ” の条件を出力する。
- ・構造体に保存されている “ circ ” の条件を出力する。
- ・構造体に保存されている “ aromatic ” の条件を出力する。
- ・構造体に保存されている “ nelectrwd ” の条件を出力する。
- ・構造体に保存されている “ ring ” の条件を出力する。
- ・下の階層^(注1)の原子指定番号毎の条件が存在する場合、その条件を保存している構造体に対して、本関数 ItgOutputAtomCondition を呼び出す。
- ・同じ階層^(注1)の原子指定番号毎の条件が存在する場合、その条件を保存している構造体に対して、本関数 ItgOutputAtomCondition を呼び出す。

注 1) 階層とは着目原子からの位置関係を示し、上の階層とは参照している原子より着目原子に近い原子、下の階層とは参照している原子より着目原子から遠い原子を指す。

引数

PRMIfFramePtr head

備考

なし。

4-38 ItgReadLowMolPrm

機能概要

- ・ tplgeneL 用のパラメータ DB ファイルを読み込む。

機能説明

- ・ パラメータ DB ファイル中にキーワード “ Start atom ” の文字列があった場合、以下の行から原子パラメータを読み込む。
- ・ パラメータ DB ファイル中にキーワード “ Start bond ” の文字列があった場合、以下の行から結合パラメータを読み込む。
- ・ パラメータ DB ファイル中にキーワード “ Start angl ” の文字列があった場合、以下の行から結合角パラメータを読み込む。
- ・ パラメータ DB ファイル中にキーワード “ Start tors ” の文字列があった場合、以下の行から二面角パラメータを読み込む。
- ・ パラメータ DB ファイル中にキーワード “ Start impr ” の文字列があった場合、以下の行から Improper-Torsion パラメータを読み込む。
- ・ 上記のキーワードが出てきた場合、関数 ItgReadPrm を呼び出す。

引数

```
PRMDBPtr prmdb  
char *db_filename  
char *nonbond_filename  
char *DB_PATH
```

備考

なし。

4-39 ItgReadPrm

機能概要

- ・ tplgeneL 用のパラメータ DB ファイルを読み込む。

機能説明

- ・ パラメータ DB ファイルから読み込んだ文字列情報から、“;” 及び空白文字を取り除く。
- ・ 変数 prm が “atom” の場合、関数 ItgCheckAtmKeyword を呼び出す。
- ・ 変数 prm が “bond” の場合、関数 ItgCheckBndKeyword を呼び出す。
- ・ 変数 prm が “angle” の場合、関数 ItgCheckBndKeyword を呼び出す。
- ・ 変数 prm が “torsion” の場合、関数 ItgCheckTorKeyword を呼び出す。
- ・ 変数 prm が “improper” の場合、関数 ItgCheckImpKeyword を呼び出す。

引数

PRMDBPtr	prmdb
FILE	*fp
char	*prm
int	*line

備考

なし。

4-40 ItgCheckAtmKeyWord

機能概要

- tplgeneL 用のパラメータ DB ファイルから atom 情報を読み込む。

機能説明

- 関数 ItgCheckCharacter を呼び出す。
- 関数 ItgCountNumOfChar を呼び出す。
- 原子タイプを読み込み、prmdb->atom->type に代入する。
- 原子量を読み込み、prmdb->atom->mass に代入する。
- ファンデルワールス半径を読み込み、prmdb->atom->rvdw に代入する。
- 電荷を読み込み、prmdb->atom->evdw に代入する。
- パラメータの分子軌道計算レベルを読み込み、prmdb->atom->level に代入する。
- 原子名を読み込み、prmdb->atom->atom に代入する。
- 混成軌道のタイプを読み込み、prmdb->atom->com に代入する。
- その原子の結合の数を読み込み、prmdb->atom->nbond に代入する。
- 結合している各原子の原子名を読み込み、prmdb->atom->nxtatm->atom に代入する。
- 結合している各原子の混成軌道のタイプを読み込み、prmdb->atom->nxtatm->com に代入する。

引数

PRMDBPtr	prmdb
char	*textbuf
int	*line

備考

なし。

4-41 ItgCheckBndKeyWord

機能概要

- tplgeneL 用のパラメータ DB ファイルから bond 合、angle 情報を読み込む。

機能説明

- 関数 ItgCheckCharacter を呼び出す。
- 関数 ItgCountNumOfChar を呼び出す。
- 力の定数を読み込み、prmdb->bond->K、又は、prmdb->angle->K に代入する。
- 結合情報の読み込み時には結合長を読み込み、prmdb->bond->Req に代入する。
- 結合角情報の読み込み時には結合角度を読み込み、prmdb->angl->K に代入する。
- 各原子の原子タイプを読み込み、prmdb->bond->type1、type2、又は、prmdb->angl->type1、type2、type3 に代入する。
- パラメータ DB ファイル内に結合次数の記載がある場合、その結合次数の範囲を読み込み、prmdb->bond->order12_min、及び、prmdb->bond->order12_max に代入する。

引数

PRMDBPtr	prmdb
char	*textbuf
char	*prm

備考

なし。

4-42 ItgCheckTorKeyWord

機能概要

- tplgeneL 用のパラメータ DB ファイルから torsion 情報を読み込む。

機能説明

- 関数 ItgCheckCharacter を呼び出す。
- 関数 ItgCountNumOfChar を呼び出す。
- Torsion パラメータの数を読み込み、prmdb->tors->no に代入する。
- 二面角の総数を読み込み、prmdb->tors->parts->idivf に代入する。
- 2 を単位とした回転対称の数を読み込み、prmdb->tors->parts->idivf に代入する。
- 位相を読み込み、prmdb->tors->parts->phase に代入する。
- 1-4 相互作用における非共有結合のカウントのためのフラグを読み込み、prmdb->tors->parts->pn に代入する。
- 二面角を構成する各原子の原子タイプを読み込み、prmdb->tors->type1、type2、type3、type4 に代入する。
- 二面角を構成する各結合の結合次数の読み込みを行い、prmdb->tors->order23_min、及び、prmdb->tors->order23_max に代入する。

引数

PRMDBPtr	prmdb
char	*textbuf
int	*line

備考

なし。

4-43 ItgCheckImpKeyWord

機能概要

- ・ tplgeneL 用のパラメータファイルから improper-torsion 情報を読み込む。

機能説明

- ・ 関数 ItgCheckCharacter を呼び出す。
- ・ 関数 ItgCountNumOfChar を呼び出す。
- ・ 二面角の総数を読み込み、prmdb->impr->parts->idivf に代入する。
- ・ 2 を単位とした回転対称の数を読み込み、prmdb->impr->parts->idivf に代入する。
- ・ 位相を読み込み、prmdb->impr->parts->phase に代入する。
- ・ インプロパー二面角を構成する各原子の原子タイプを読み込み prmdb->impr->type1、type2、type3、type4 に代入する。

引数

PRMDBPtr	prmdb
char	*textbuf
int	*line

備考

なし。

4-44 ItgChekcCharacter

機能概要

- ・ tplgeneL 用のパラメータ DB ファイル読み込み時に、その文字列をチェックする。

機能説明

- ・ 引数として渡された文字列が行の先頭ではなく、前の文字が “ , ” “ ; ” “ ! ” 以外の場合、エラー処理を行う。

引数

int	j
char	*p
int	line
char	*textbuf

備考

なし。

4-45 ItgCharCount

機能概要

- ・ tplgeneL 用のパラメータファイル読み込み時に、そのトークンに含まれる文字数を数える。

機能説明

- ・ トークンに含まれる文字数を数える。

引数

char	*p
int	j,k
int	*k2

備考

なし。

4-46 ItgCountNumOfChar

機能概要

- ・ tplgeneL 用のパラメータファイル読み込み時に、そのトークンに含まれる文字数を数える。

機能説明

- ・ 関数 ItgCharCount を呼び出し、トークンの文字数を数える。
- ・ 文字列の先頭の行のカラム番号と文字数の和を返り値とし、呼び出し元関数に返す。

引数

int	j
char	*p
int	char_num
int	*k0
int	*k2

備考

なし。

4-47 ItgReadAtomTypeNum

機能概要

- ・ nonbond ファイルの読み込みを行い、原子タイプ - 原子タイプ番号の一覧を作成する。

機能説明

- ・ nonbond ファイルの読み込みを行う。
- ・ 原子タイプ番号を取得する(関数 ItgSplitTextbuf を呼び出す)。

引数

PRMDBPtr	prmdb
char	*nondond_filename

備考

なし。

4-48 ItgSplitTextbuf

機能概要

- ・各原子の原子タイプとその原子タイプ番号の組を取得する。

機能説明

- ・各原子の原子タイプとその原子タイプ番号の組を取得する。

引数

char	*textbuf
PRMDBPtr	prmdb

備考

なし。

4-49 ItgDefineBond

機能概要

- ・分子内に含まれる全結合に対して、単結合、芳香族結合、二重結合を割り当てる。

機能説明

- ・その結合の結合次数が 2.15 以上の場合、三重結合を割り振り、
sys->mol->bond->bond_type に 8 を代入する。
- ・その結合の結合次数が定数 1.5 以上の場合、二重結合を割り振り、
sys->mol->bond->bond_type に 4 を代入する。
- ・その結合の結合次数が定数 1.2 以上の場合、芳香族共有結合と割り振り、
sys->mol->bond->bond_type に 2 を代入する。
- ・その結合の結合次数が定数 0.2 以上の場合、単結合と割り振り、
sys->mol->bond->bond_type に 1 を代入する。
- ・各原子の結合の数を数え、sys->mol->atom->bond_num に保存する。
- ・電子吸引性の原子に結合している数を数える。

引数

LTGSystemPtr sys

備考

なし。

4-50 ItgAssignSP

機能概要

- ・各原子について、結合の数から混成軌道の種類を割り当てる。

機能説明

- ・炭素原子について、結合の数が4の場合、sp³軌道を割り当てる。
結合の数が3の場合、sp²軌道を割り当てる。
結合の数が2の場合、sp軌道を割り当てる。
結合の数が1の場合、sp軌道を割り当てる。
- ・窒素原子について、結合の数が4の場合、sp³軌道を割り当てる。
結合の数が3の場合、sp²軌道を割り当て、以下に示す表による補正を行う。
結合の数が2の場合、sp²軌道を割り当てる。
結合の数が2で単結合と三重結合、あるいは、2つの二重結合を持つ場合、sp軌道を割り当てる。
結合の数が1の場合、sp軌道を割り当てる。
- ・酸素原子について、結合の数が2の場合、sp³軌道を割り当てる。
結合の数が1の場合、sp²軌道を割り当てる。
- ・水素原子について、s軌道を割り当てる。
- ・リン原子について、結合の数が4の場合、sp³軌道を割り当てる。
結合の数が3の場合、sp³軌道を割り当てる。
結合の数が2の場合、sp²軌道を割り当てる。
- ・硫黄原子について、結合の数が4の場合、sp³軌道を割り当てる。
結合の数が3の場合、sp³軌道を割り当てる。
結合の数が2の場合、sp³軌道を割り当てる。
結合の数が1の場合、sp²軌道を割り当てる。
- ・Ca、F、Cl、Br、I原子について、s軌道を割り当てる。

表．結合の数3の窒素原子に対する混成軌道の補正

窒素原子の隣の原子の環境	補正後の混成軌道
結合の数が3または2の炭素原子と結合	sp ²
結合の数が2または1の窒素原子と結合	sp ²
結合の数が2または3のリン原子と結合	sp ²
結合の数が1の酸素原子と結合	sp ²
結合の数4の硫黄原子と結合しており、その硫黄原子が結合の数1の酸素原子と結合	sp ³
結合している炭素原子の持っている結合がすべて単結合	sp ³

引数

LTGSystemPtr sys

備考

なし。

4-51 ltgNxtAtmInf

機能概要

- ・各原子に結合している原子の情報を読み込む。

機能説明

- ・結合している原子の原子名を読み込む。
- ・結合している原子の混成軌道の種類を読み込む。
- ・結合している原子の元素記号を読み込む。
- ・結合している原子の通し番号を読み込む。
- ・結合している原子の結合の数を読み込む。
- ・結合している原子について、原子吸引性原子との結合の数を読み込む。

引数

LTGSystemPtr sys

備考

なし

4-52 ItgCountCirc

機能概要

- ・分子内に含まれる環構造の数を計算する。
- ・環状分子の計算の場合、芳香環、脂環のチェックを行う。

機能説明

- ・分子内に含む環構造の数を計算する。
- ・環状分子計算の場合、環構造に含まれる原子の番号の組を取得する (関数 ItgGetCircParts を呼び出す)。
- ・環状分子計算の場合、6員環以内の環の構成原子の番号を取得する (関数 ItgGetCircPattern を呼び出す)。
- ・環状分子計算の場合、電子数を求める (関数 ItgGetElecNum を呼び出す)。
- ・環状分子計算の場合、着目原子の隣の原子が芳香族か否かをチェックする (関数 ItgGetNxtArmFlg を呼び出す)。

引数

LTGSystemPtr sys

備考

アダマンタンやピレン等の構造については正しい環の数を計算する事ができない。

4-53 ItgSetCircFlg

機能概要

- ・環構造を構成する原子にフラグを立てる。

機能説明

- ・結合の数が1の原子の結合を切れるところまで切る。
- ・結合の数が2以上になった原子に繋がっている原子の数を数える。
(関数 ItgCntReachAtom を呼び出す)。
- ・結合の数が2以上の原子の結合を1つ切って、繋がっている原子の数が増えた場合、その結合は環の一部ではないので、その結合を切ったままにする。
- ・最終的に残った原子に対して、フラグを立てる。

引数

LTGMolPtr mol

備考

なし。

4-54 ltgCntReachAtom

機能概要

- ・仮想的に結合を切断した場合に、その切断後の着目原子を含む構造の原子数を取得する。

機能説明

- ・仮想的に結合を切断した場合に、その切断後の着目原子を含む構造の原子数を取得し、返り値として返す。

引数

LTGMolPtr	mol
int	atom
int	*bond1, *bond2

備考

なし。

4-55 ItgGetCircParts

機能概要

- ・環状分子のそれぞれの環構造を構成する原子の組を得る。

機能説明

- ・環状分子のそれぞれの環構造に分ける。
- ・結合の数が1の原子の結合を切る（関数 ItgCutBond を呼び出す）。
- ・環状分子のそれぞれの環構造を構成する原子の組を得る。

引数

LTGSystemPtr sys

備考

複数の環が1つ以上の原子で直接結合している場合は、便宜上、それらの環を1つの環と見なして原子の組を取得する。

4-30 ltgCutBond

機能概要

- ・結合の数が1の原子の結合を切る。

機能説明

- ・結合の数が1の原子の結合を切る。

引数

int *bond1, *bond2, *bond_num
LTGSystemPtr sys

備考

なし

4-57 ItgGetCircPattern

機能概要

- ・環状分子の環構造のメンバを取得する。

機能説明

- ・3～6員環の環に含まれる原子の番号の組を取得する。
- ・3～6員環の環構造をチェックし、そのメンバを取得し、配列に保存する(関数 ItgCheckCircPattern を呼び出す)。
- ・配列中の環構造のメンバを原子の番号順にソートする(関数 ItgBubbleSort を呼び出す)。
- ・環構造が完全に重複しているものを削除する(関数 ItgRemoveOverlapCirc を呼び出す)。
- ・環構造に属する各原子に環のメンバ数を保存する。ただし、複数の環構造に属する場合は最大の環のメンバ数を保存する。

引数

LTGSystemPtr sys

備考

なし。

4-58 ItgCheckCircPattern

機能概要

- ・環状分子の環構造のチェックをする。

機能説明

- ・最大6個先の原子が着目原子と一致しなかった場合、呼び出し元関数に FALSE を返す。
- ・最小3個先の原子が着目原子と一致した場合、着目している原子の番号を配列に保存し、呼び出し元関数に TRUE を返す。
- ・上記にあてはまらない場合、参照している原子が結合している原子に対して、本関数 ItgCheckCircPattern を再帰的に呼び出す。
- ・呼び出した関数 ItgCheckCircPattern の戻り値が TRUE である場合、参照している原子の番号を配列に保存する。
- ・参照している原子が結合している原子全てに対して呼び出した関数 ItgCheckCircPattern の戻り値が全て FALSE であった場合、呼び出し元関数に FALSE を返す。
- ・参照している原子が結合している原子全てに対して呼び出した関数 ItgCheckCircPattern の戻り値の中に一つでも TRUE があった場合、呼び出し元関数に TRUE を返す。

引数

LTGSystemPtr	sys
LTGAtomPtr	atom
int	**array
int	pre_atom_num
int	root[][6]
int	*nroot

備考

なし。

4-59 ItgBubbleSort

機能概要

- ・環構造のメンバを原子の番号順にソートする。

機能説明

- ・引数として渡された、環構造のメンバを原子の番号昇順にソートする。

引数

int	data[]
int	n

備考

なし。

4-60 ItgRemoveOverlapCirc

機能概要

- ・ 環構造が完全に重複しているものを削除する。

機能説明

- ・ 引数として渡された環構造のメンバのリストのうち、完全に環構造が重複しているものを削除する。

引数

int	root[][6]
int	*nroot

備考

なし。

4-61 ItgGetElecNum

機能概要

- ・環状分子のそれぞれの環構造の原子の 電子の数を数え、芳香族に相当するものに芳香族のフラグを立てる。

機能説明

- ・環状分子のそれぞれの環構造の 電子の数を数える。
 - 炭素：1 電子
 - 窒素：1 電子（結合の数が 2 本）、2 電子（結合の数が 3 本）
 - 酸素：2 電子
 - 硫黄：2 電子
- ・電子の数が $4n+2$ (n は整数) 個になったときは芳香族としてのフラグを立てる。
- ・環内に結合の数 3、 sp^2 の窒素がある場合、電子の合計が 7 になっても芳香族としてのフラグを立てる。

引数

LTGSystemPtr sys

備考

なし

4-62 ItgGetNxtArmFlg

機能概要

- ・各原子と結合している原子の芳香族フラグを取得する。

機能説明

- ・各原子と結合している原子の芳香族フラグを取得する。

引数

LTGSystemPtr sys

備考

なし。

4-63 ItgGet1314Num

機能概要

- ・bond 情報から angle、torsion、improper-torsion を構成する原子の番号を計算する。

機能説明

- ・1-3 相互作用する原子の組の数、及び、その組の原子の番号を取得する
(関数 ItgGet13Num を呼び出す)。
- ・1-4 相互作用する原子の組の数、及び、その組の原子の番号を取得する
(関数 ItgGet14Num を呼び出す)。
- ・インプロパー相互作用する原子の組、及び、その組の原子の番号を取得する
(関数 ItgGetImpNum を呼び出す)。

引数

LTGSystemPtr sys

備考

なし

4-64 ltgGet13Num

機能概要

- ・ bond 情報から angle を構成する原子の番号を計算する。

機能説明

- ・ bond 情報から angle を構成する原子の番号を計算する。
- ・ angle の組の数を取得する。

引数

LTGSystemPtr sys

備考

なし

4-65 ltgGet14Num

機能概要

- ・ bond 情報から torsion を構成する原子の番号を計算する。

機能説明

- ・ bond 情報を組み合わせて torsion を構成する原子の番号を計算する。

引数

LTGSystemPtr sys

備考

なし

4-66 ItgGetImpNum

機能概要

- ・ bond 情報から improper-torsion を構成する原子の番号を計算する。

機能説明

- ・ bond 情報を組み合わせて improper-torsion を構成する原子の番号を計算する。

引数

LTGSystemPtr sys

備考

なし

4-67 ItgNextMalloc

機能概要

- ・メモリ領域の確保を行う。

機能説明

- ・atom 情報についてメモリ領域の確保を行う。
- ・bond 情報についてメモリ領域の確保を行う。
- ・angle 情報についてメモリ領域の確保を行う。
- ・torsion 情報についてメモリ領域の確保を行う。
- ・improper-torsion 情報についてメモリ領域の確保を行う。

引数

LTGSystemPtr sys
int next_code

備考

なし

4-68 ItgFragmentDB

機能概要

- ・フラグメント DB に登録されているフラグメントと入力分子のパターンマッチを行う。

機能説明

- ・フラグメント DB に登録されているフラグメントと入力分子のパターンマッチを行う。
- ・フラグメント DB ファイル情報の読み込みを行う(関数 ItgReadFrgDB を呼び出す)。
- ・フラグメントの環内にある原子にフラグを立てる(関数 ItgSetCircFlgFrg を呼び出す)。
- ・フラグメント構造と入力構造のパターンマッチを行う(関数 ItgPatternMatch を呼び出す)。

引数

LTGSystemPtr sys
char *frg_filename

備考

なし

4-69 ItgReadFrgDB

機能概要

- ・フラグメント DB ファイルを読み込む。

機能説明

- ・パラメータ DB ファイルとして prm_amber99.db を使用する場合、フラグメント DB ファイルとして、frg_amber99.db をオープンする。
- ・パラメータ DB ファイルとして prm_gaff.db を使用する場合、フラグメント DB ファイルとして、frg_gaff.db をオープンする。
- ・フラグメント DB ファイルがない場合は、メッセージを表示し、フラグメント DB の処理をスキップする。
- ・フラグメント DB ファイル中にキーワード “ Start fragment ” の文字列があった場合、フラグメントのパラメータを読み込む関数 ItgCheckFrgKeywordFrg を呼び出す。
- ・フラグメント DB ファイル中にキーワード “ Start atom ” の文字列があった場合、原子パラメータを読み込む。
- ・フラグメント DB ファイル中にキーワード “ Start bond ” の文字列があった場合、結合パラメータを読み込む。
- ・フラグメント DB ファイル中にキーワード “ Start angl ” の文字列があった場合、結合角パラメータを読み込む。
- ・フラグメント DB ファイル中にキーワード “ Start tors ” の文字列があった場合、二面角パラメータを読み込む。
- ・フラグメント DB ファイル中にキーワード “ Start impr ” の文字列があった場合、インプロパー二面角パラメータを読み込む。
- ・ “ Start fragment ” 以外の上記のキーワードが出てきた場合、関数 ItgReadPrmFrg を呼び出し、各種パラメータを読み込む。

引数

FRGDBPtr	frgdb
char	*frg_filename

備考

なし

4-70 ItgReadPrmFrg

機能概要

- ・フラグメント DB ファイルからパラメータを読み込む。

機能説明

- ・フラグメント DB ファイルから読み込んだ文字列情報から、“;” 及び空白文字を取り除く。
- ・変数 prm が “atom” の場合、関数 ItgCheckAtmKeywordFrg を呼び出す。
- ・変数 prm が “bond” の場合、関数 ItgCheckBndKeywordFrg を呼び出す。
- ・変数 prm が “angle” の場合、関数 ItgCheckBndKeywordFrg を呼び出す。
- ・変数 prm が “torsion” の場合、関数 ItgCheckTorKeywordFrg を呼び出す。
- ・変数 prm が “improper” の場合、関数 ItgCheckImpKeywordFrg を呼び出す。

引数

FRGDBPtr	frgdb
FILE	*fp
char	*prm

備考

なし

4-71 ItgCheckAtmKeywordFrg

機能概要

- ・フラグメント DB ファイルから atom 情報を読み込む。

機能説明

- ・原子タイプが定義されている場合は、原子タイプを読み込み、フラグを立てる。
- ・原子量が定義されている場合は、原子量を読み込み、フラグを立てる。
- ・ファンデルワールス半径が定義されている場合は、ファンデルワールス半径を読み込み、フラグを立てる。
- ・部分電荷が定義されている場合は、部分電荷を読み込み、フラグを立てる。
- ・パラメータの分子軌道計算レベルが定義されている場合は、分子軌道レベルを読み込み、フラグを立てる。
- ・原子名を読み込む。
- ・混成軌道のタイプが定義されている場合は、混成軌道のタイプを読み込む。
- ・原子の結合の数が定義されている場合は、結合の数を読み込む。
- ・環のフラグが定義されている場合は、環のフラグを読み込む。

引数

FRGDBPtr	frgdb
char	*textbuf

備考

なし

4-72 ItgCheckBndKeywordFrg

機能概要

- ・フラグメント DB ファイルから bond、angle 情報を読み込む。

機能説明

- ・力の定数が定義されていた場合は、力の定数を読み込みフラグを立てる。
- ・結合長が定義されていた場合は、結合長を読み込み、フラグを立てる。
- ・結合角が定義されていた場合は、結合角度を読み込み、フラグを立てる。
- ・各原子のフラグメント内の原子通し番号を読み込み、frgdb->frag->bond->bond1、bond2、又は、frgdb->frag->angle->angl1、angl2、angl3 に代入する。
- ・フラグメント DB ファイル内に結合次数の記載がある場合、その結合次数の範囲を読み込み、frgdb->bond->order12_min、及び、frgdb->bond->order12_max に代入する。

引数

FRGDBPtr	frgdb
char	*textbuf
char	*prm

備考

なし。

4-73 ItgCheckTorKeywordFrg

機能概要

- ・フラグメント DB ファイルから torsion 情報を読み込む。

機能説明

- ・ torsion パラメータの数が定義されていた場合は、torsion パラメータの数を読み込み、フラグを立てる。
- ・二面角の総数が定義されていた場合は、二面角の総数を読み込み、フラグを立てる。
- ・2 を単位とした回転対称の数が定義されていた場合は、回転対象の数を読み込み、フラグを立てる。
- ・位相が定義されていた場合は、位相を読み込み、フラグを立てる。
- ・1-4 相互作用における非共有結合のカウントのためのフラグが定義されている場合、フラグを読み込み、フラグを立てる。
- ・二面角を構成する各原子の原子タイプを読み込み、frgdb->tors->type1、type2、type3、type4 に代入する。
- ・二面角を構成する各結合の結合次数の読み込みを行い、frgdb->tors->order23_min、及び、frgdb->tors->order23_max に代入する。

引数

FRGDBPtr	frgdb
char	*textbuf

備考

なし。

4-74 ItgCheckImpKeywordFrg

機能概要

- ・フラグメント DB ファイルから improper-torsion 情報を読み込む。

機能説明

- ・improper-torsion パラメータの数が定義されていた場合は、improper-torsion パラメータの数を読み込み、フラグを立てる。
- ・二面角の総数が定義されていた場合は、二面角の総数を読み込み、フラグを立てる。
- ・2 を単位とした回転対称の数が定義されていた場合は、回転対象の数を読み込み、フラグを立てる。
- ・位相が定義されていた場合は、位相を読み込み、フラグを立てる。
- ・インプロパー二面角を構成する各原子の原子タイプを読み込み、frgdb->impr->type1、type2、type3、type4 に代入する。に代入する。

引数

FRGDBPtr	frgdb
char	*textbuf

備考

なし。

4-75 ItgSetCircFlgFrg

機能概要

- ・フラグメントの環構造を構成する原子にフラグを立てる。

機能説明

- ・結合の数が1の原子の結合を切れるところまで切る。
- ・結合の数が2以上になった原子に繋がっている原子の数を数える
(関数 ItgCntReachAtomFrg を呼び出す)。
- ・結合の数が2以上の原子の結合を1つ切って、繋がっている原子の数が増えた場合、その結合は環の一部ではないので、その結合を切ったままにする。
- ・最終的に残った原子に対して、フラグを立てる。

引数

FRGFragmentPtr fragment

備考

なし。

4-76 ItgCntReachAtomFrg

機能概要

- ・ 指定した原子に繋がっている原子の数を数える。

機能説明

- ・ 指定した原子に繋がっている原子の数を数えて返り値として返す。

引数

```
FRGFragmentPtr fragment  
int             atom  
int             *bond1, *bond2
```

備考

なし。

4-77 ItgPatternMatch

機能概要

- ・フラグメントと計算対象分子とのパターンマッチとパラメータの割り当てを行う。

機能説明

- ・フラグメントの原子の番号を結合に沿って振り直す
(関数 ItgSortFrgAtom を呼び出す)。
- ・フラグメントと計算対象分子とのパターンマッチとパラメータの割り当てを行う。
- ・フラグメントと計算対象分子と元素名の比較を行う
(関数 ItgFrgAtomNameCmp を呼び出す)。
- ・フラグメントとパターンが一致した原子団の重複をチェックする
(関数 ItgCheckOverlap を呼び出す)。
- ・フラグメントとパターンが一致した場合、パラメータの割り当てを行う
(関数 ItgAssignFrgPrm を呼び出す)。

引数

LTGSystemPtr	sys
FRGFragmentPtr	fragment

備考

なし。

4-78 ltgSortFrgAtom

機能概要

- ・フラグメント内の原子の番号の振り直しを行う。
結合に沿って原子の番号を振り直す。

機能説明

- ・フラグメント内の原子の番号の振り直しを行う。結合に沿って原子の番号を振りなおす。

引数

LTGSystemPtr sys

FRGFragmentPtr fragment

備考

なし。

4-79 ItgFrgAtomNameCmp

機能概要

- ・フラグメントと計算対象分子と元素名の比較を行う。

機能説明

- ・フラグメント側と計算対象分子側の元素名が一致した場合、TRUE を返す。
- ・フラグメント側の元素名がワイルドカード “X” で定義されていた場合、計算対象分子側の元素名が何であっても TRUE を返す。
- ・フラグメント側の元素名が複数表記 (例 “[C,N,S]”) されていて、計算対象分子の元素名がこの中の 1 つと一致する場合、TRUE を返す。
- ・それ以外の場合は FALSE を返す。

引数

char *frg_atom
char *sys_atom

備考

なし。

4-80 ItgCheckOverlap

機能概要

- ・フラグメントと一致した原子の組が重複していないかチェックを行う。

機能説明

- ・フラグメントと一致した原子の通し番号の組を原子の通し番号でソートする。
- ・ソートしたフラグメントと一致した原子の組と履歴(フラグメントと一致した原子の組をソートしたもの)と比較を行い、同じものが無い場合、この原子の組を履歴に保存し、TRUE を返す。
- ・重複している場合は FALSE を返す。

引数

int	**sortedlist
int	*list
int	list_num
int	atom_num

備考

なし。

4-81 ltgAssignFrgPrm

機能概要

- ・フラグメントと一致した原子の組にパラメータを割り当てる。

機能説明

- ・フラグメントと一致した原子の組にパラメータを割り当てる。
- ・原子タイプが定義されていた場合、原子タイプと原子タイプ番号を割り当てる。
- ・原子量が定義されていた場合、原子量を割り当てる。
- ・原子半径が定義されていた場合、原子半径を割り当てる。
- ・部分電荷が定義されていた場合、部分電荷を割り当てる。
- ・結合について、力の定数が定義されていた場合、力の定数を割り当てる。
- ・結合について、平衡結合距離が定義されていた場合、平衡結合距離を割り当てる。
- ・結合角について、平衡結合角が定義されていた場合、平衡結合角を割り当てる。
- ・二面角について、パラメータの個数が定義されていた場合、パラメータの個数を割り当てる。
- ・二面角について、力の定数が定義されていた場合、力の定数を割り当てる。
- ・二面角について、二面角の総数が定義されていた場合、二面角の総数を割り当てる。
- ・二面角について、回転対称数が定義されていた場合、回転対称数を割り当てる。
- ・二面角について、位相が定義されていた場合、位相を割り当てる。
- ・インプロパー二面角について、パラメータの個数が定義されていた場合、パラメータの個数を割り当てる。
- ・インプロパー二面角について、力の定数が定義されていた場合、力の定数を割り当てる。
- ・インプロパー二面角について、二面角の総数が定義されていた場合、二面角の総数を割り当てる。
- ・インプロパー二面角について、回転対称数が定義されていた場合、回転対称数を割り当てる。
- ・インプロパー二面角について、位相が定義されていた場合、位相を割り当てる。
- ・割り当てた各パラメータについてフラグを立てる。

引数

LTGMolPtr	mol
FRGFragmentPtr	frg
int	*match_table_frg

備考

なし。

4-82 ItgAssignAtomType

機能概要

- ・計算対象分子の各原子の基本情報と、原子タイプ割り当て規則を比較して、原子タイプの割り当てを行う。

機能説明

- ・計算対象分子の各原子に対して、関数 ItgCheckAtomCondition を呼び出し、各原子情報と原子タイプ割り当て規則との比較を行う。
- ・tplgeneL オリジナル入力ファイルで原子タイプが指定されていない場合、条件が一致した原子タイプとデフォルト原子タイプを割り当てる。
- ・tplgeneL オリジナル入力ファイルで原子タイプが指定されている場合、条件が一致したデフォルト原子タイプのみを割り当てる。
- ・割り当てた原子タイプに対して、関数 ItgGetSameAtomType を呼び出し、原子タイプのパラメータを取得する。

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb
PRMIfFramePtr	ifframe_head

備考

なし。

4-83 ItgCheckAtomCondition

機能概要

- ・計算対象分子の各原子の基本情報と、原子タイプ割り当て規則の比較を行う。

機能説明

- ・参照している原子指定番号の条件と原子の情報が一致しない場合、
戻り値として FALSE を呼び出し元関数に返す。
- ・参照している原子指定番号の条件と原子の情報が一致する場合以下の処理を行う。
 - ・参照している原子指定番号より下の階層の条件が無い場合、
戻り値として TRUE を呼び出し元関数に返す。
 - ・参照している原子指定番号より下の階層の条件がある場合、
参照している原子の隣の原子に対して、本関数 ItgCheckAtomCondition を呼び出す。
 - ・隣の原子に対して呼び出した ItgCheckAtomCondition の戻り値が TRUE の場合、
その隣の原子を候補として保存する。
- ・参照している原子指定番号より下の階層の条件に対して、候補となる原子が全てみつ
かった場合、関数 ItgCheckCombiNextAtoms を呼び出し、戻り値が TRUE の場合、呼び
出し元関数に TRUE を返す。

引数

LTGSystemPtr	sys
LTGAtomPtr	atom
int	pre_atm_num
PRMAtomConditionPtr	cond

備考

なし。

4-84 ItgCheckCombiNextAtoms

機能概要

原子指定番号における隣の原子の候補として、重複せずに別々の原子を割り振ることが可能か確認する

機能説明

- ・引数として渡された配列に保存された、原子番号をそれぞれの行で重複しない組合せを取ることが可能か確認する。
- ・重複しない組合せを取ることが可能な場合、呼び出し元関数に TRUE を返す。
- ・重複しない組合せを取ることができない場合、呼び出し元関数に FALSE を返す。

引数

```
int match_table[6][6]
```

備考

なし。

4-85 ItgGetSameAtomType

機能概要

- ・低分子パラメータファイル中の原子情報を代入する。

機能説明

- ・関数 ItgAssignAtomType で割り当てられた原子タイプとパラメータ DB ファイル中の原子タイプが一致した場合、各原子パラメータの取得を行う。

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb

備考

なし。

4-86 ItgSetParm

機能概要

- atom、bond、angle、torsion、improper-torsion パラメータの割り当てを行う。

機能説明

- bond パラメータを割り当てる。(関数 ItgBondParm)
- angle パラメータを割り当てる。(関数 ItgAngleParm)
- torsion パラメータを割り当てる。(関数 ItgTorsionParm)
- torsion パラメータの補正を行う。(関数 ItgCorrectTorsion)
- improper-torsion を割り当てる。(関数 ItgImproperParm)
- atom パラメータを割り当てる。(関数 ItgAtomParm)
- 不足パラメータ補填方法として動的補填が選択去れている場合、不足パラメータの動的補填を行う際に必要となる情報の取得を行う。(関数 ItgReadLackParm)
- 不足しているパラメータを補填する。(関数 ItgCorrectNotExistParm)

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb
int	prm
char	*DB_PATH

備考

なし。

4-87 ItgBondParm

機能概要

- ・結合している各原子の原子タイプ、及び、その結合の結合次数情報と、低分子パラメータ DB ファイルとを比較して、bond 情報を割り当てる。

機能説明

- ・原子タイプ名を低分子パラメータ DB と比較する。
- ・結合次数を低分子パラメータ DB と比較する。
- ・原子タイプ、結合次数ともに条件を満たしている場合、力の定数、及び、平衡結合長を代入する。

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb

備考

なし

4-88 ltgAngleParm

機能概要

- ・ angle パラメータの割り振りを行う。

機能説明

- ・ 結合角を構成する各原子の原子タイプを低分子パラメータ DB 中のものと比較する。
- ・ 原子タイプが一致した場合、力の定数を代入する。
- ・ 原子タイプが一致した場合、結合角度を代入する。

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb

備考

なし

4-89 lTgTorsionParm

機能概要

- ・ torsion パラメータの割り振りを行う。

機能説明

- ・ 二面角を構成する各原子の原子タイプを低分子パラメータ DB 中のものと比較する。
- ・ 二面角中に含む結合の結合次数を、低分子パラメータ DB 中のものと比較する。
- ・ 原子タイプと結合次数が一致した場合、力の定数を代入する。
- ・ 原子タイプと結合次数が一致した場合、二面角の総数を代入する。
- ・ 原子タイプと結合次数が一致した場合、位相を代入する。
- ・ 原子タイプと結合次数が一致した場合、1-4 相互作用における非共有結合のカウントのためのフラグを代入する。
- ・ 原子タイプと結合次数が一致した場合、2 を単位とした回転対称の数を代入する。

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb

備考

なし

4-90 ltgCorrectTorsion

機能概要

- ・ 1-4 相互作用における非共有結合のカウントのためのフラグの補正を行う。

機能説明

- ・ 結合、結合角を構成する両末端の原子が二面角を構成する両末端の原子と一致した場合、1-4 相互作用における非共有結合のカウントのためのフラグの補正を行う。

引数

LTGSystemPtr sys

備考

なし

4-91 ItgImproperParm

機能概要

- ・ improper-torsion を構成する原子の組を低分子パラメータ DB 中のものと比較する。

機能説明

- ・ インプロパー二面角を構成する原子の原子タイプを取得する (関数 ItgGetImpAtomType を呼び出す)。
- ・ インプロパー二面角を構成する各原子の原子タイプを低分子パラメータ DB 中のものと比較する。
- ・ 上記で一致した場合、インプロパー二面角パラメータを代入する (関数 ItgCopyImpPrm を呼び出す)。

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb

備考

なし

4-92 ltgGetImpAtomType

機能概要

- ・ improper-torsion を構成する原子の原子タイプを取得する。

機能説明

- ・ improper-torsion を構成する原子の原子タイプを取得する。

引数

LTGSystemPtr sys

備考

なし

4-93 ltgCopyImpPrm

機能概要

- ・ improper-torsion パラメータを代入する。

機能説明

- ・ improper-torsion パラメータを代入する。

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb

備考

なし

4-94 ItgAtomParm

機能概要

- ・ atom パラメータの割り当てを行う。

機能説明

- ・ 各原子の結合の数を数える。
- ・ 原子と結合している他の原子の相対番号を得る。
- ・ 各原子がとる結合角の数を数える。
- ・ 結合角をなす原子の相対番号を得る。
- ・ 各原子がとる二面角の数を数える。
- ・ 二面角をなすの原子の相対番号を得る。
- ・ 結合、結合角を構成する原子の原子番号と、二面角を構成する原子の原子番号が一致する場合、結合、結合角を構成する原子の原子番号を削除する。その際に結合の数、結合角の数の補正を行う(関数 ItgPairNumCorrect を呼び出す)。

引数

LTGSystemPtr sys

備考

なし

4-95 ItgPairNumCorrect

機能概要

- ・各原子における結合、結合角、二面角の数の補正を行う。

機能説明

- ・各原子における結合、結合角、二面角の数の補正を行う。
- ・結合を構成する原子の原子番号と、二面角を構成する原子の原子番号が一致する場合、結合を構成する原子の原子番号を削除する。その際に結合の数の補正を行う。
- ・結合角を構成する原子の原子番号と、二面角を構成する原子の原子番号が一致する場合、結合角を構成する原子の原子番号を削除する。その際に結合角の数の補正を行う。

引数

int* arrangement
int num

備考

なし

4-96 ItgReadLackParm

機能概要

- ・不足パラメータの動的補填を行う際に必要となる情報の取得を行う。

機能説明

- ・結合に関する力の定数を計算する為に必要となる定数をファイル bond.prm をオープンし、情報を取得する。
- ・結合角に関する力の定数を計算する為に必要となる定数をファイル angle.prm をオープンし、情報を取得する。

引数

PRMLackPrmPtr lprm
char *DB_PATH

備考

なし

4-97 ItgCorrectNotExistParm

機能概要

- bond、angle、torsion、improper-torsion の各パラメータが存在しない場合、パラメータを補填する。

機能説明

- bond パラメータが存在しない場合、選択した不足パラメータの補填方法によってパラメータを取得する。
 - デフォルトパラメータの使用が選択されていた場合、関数 ItgGetDefaultBondParm を呼び出し、不足している結合パラメータにデフォルトパラメータを代入する。
 - パラメータの動的補填が選択されていた場合、関数 ItgCalcForceConstBond を呼び出し、不足している結合パラメータを計算する。
- angle パラメータが存在しない場合、選択した不足パラメータの補填方法によってパラメータを取得する。
 - デフォルトパラメータの使用が選択されていた場合、関数 ItgGetDefaultAngleParm を呼び出し、不足している結合角パラメータにデフォルトパラメータを代入する。
 - パラメータの動的補填が選択されていた場合、関数 ItgCalcForceConstAngle を呼び出し、不足している結合角パラメータを計算する。
- torsion パラメータが存在しない場合、関数 ItgGetDefaultTorsionParm を呼び出し、不足している二面角パラメータにデフォルトパラメータを代入する。
- improper-torsion パラメータが存在しない場合、関数 ItgGetDefaultImproperParm を呼び出し、不足しているインプロパー二面角にデフォルトパラメータを代入する。

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb
PRMLackPrmPtr	lprm
Int	prm

備考

なし

4-98 ltgForceConstBond

機能概要

- bond パラメータの計算を行なう。この関数は、パラメータの動的補填を選択した場合に呼び出される。

機能説明

- 結合距離を計算する。
- 結合に関する力の定数を計算する。
- 力の定数を求める際に必要となる定数 K_{ij} は、ファイル bond.prm より取得する。
- 力の定数は以下の式を用いて計算を行う。ここで m は 4.5 を使用する。

$$K_r = K_{ij} \left(\frac{1}{r_{ij}} \right)^m$$

引数

LTGSystemPtr	sys
PRMLackPrmPtr	lprm

備考

なし

4-99 lTgForceConstAngle

機能概要

- ・ angle パラメータの計算を行なう。この関数は、パラメータの動的補填を選択した場合に呼び出される。

機能説明

- ・ 結合角を計算する。
- ・ 結合角に関する力の定数を計算する。
- ・ 力の定数を求める際に必要となる定数 Zi、Cj、Zk は、ファイル angle.prm より取得する。
- ・ 力の定数は以下の式を用いて計算を行う。

$$K_{ijk}^0 = 143.9 Z_i C_j Z_k (r_{ij}^{eq} + r_{jk}^{eq})^{-1} (q_{ijk}^{eq})^{-2} \exp(-2D)$$

$$D = \frac{(r_{ij}^{eq} - r_{jk}^{eq})^2}{(r_{ij}^{eq} + r_{jk}^{eq})^2}$$

引数

LTGSystemPtr	sys
PRMLackPrmPtr	lprm

備考

なし

4-100 ItgGetDefaultBondParm

機能概要

- ・ デフォルトの bond パラメータを取得する。

機能説明

- ・ デフォルト原子タイプをもとに、デフォルトの bond パラメータを取得する。

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb

備考

なし

4-101 ItgGetDefaultAngleParm

機能概要

- ・ デフォルトの angle パラメータを取得する。

機能説明

- ・ デフォルト原子タイプをもとに、デフォルトの angle パラメータを取得する。

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb

備考

なし

4-102 ItgGetDefaultTorsionParm

機能概要

- ・ デフォルトの torsion パラメータを取得する。

機能説明

- ・ デフォルト原子タイプをもとに、デフォルトの torsion パラメータを取得する。

引数

LTGSystemPtr		sys
PRMDBPtr	prmdb	

備考

なし

4-103 ItgGetDefaultImproperParm

機能概要

- ・ デフォルトの improper-torsion を取得する。

機能説明

- ・ デフォルト原子タイプをもとに、デフォルトの improper-torsion を取得する。

引数

LTGSystemPtr	sys
PRMDBPtr	prmdb

備考

なし

4-104 ItgOutputTopology

機能概要

- ・トポロジーファイルの出力を行う。

機能説明

- ・atom 情報の出力を行う。
- ・bond 情報の出力を行う。
- ・angle 情報の出力を行う。
- ・torsion 情報の出力を行う。
- ・improper-torsion 情報の出力を行う。
- ・nonbond 情報の出力を行う。

引数

```
LTGSystemPtr sys  
char          *filename, *nonbond_filename  
char          *resname
```

備考

なし

5 . 低分子トポロジージェネレータ用入力ファイル作成ツール ReadGamess

以下に量子化学計算ソフト GAMESS の出力ファイルから低分子トポロジージェネレータ用入力ファイル(電荷情報、結合情報、Z-Matrix 情報)を作成するツール ReadGamess の各関数の構成図を示す。

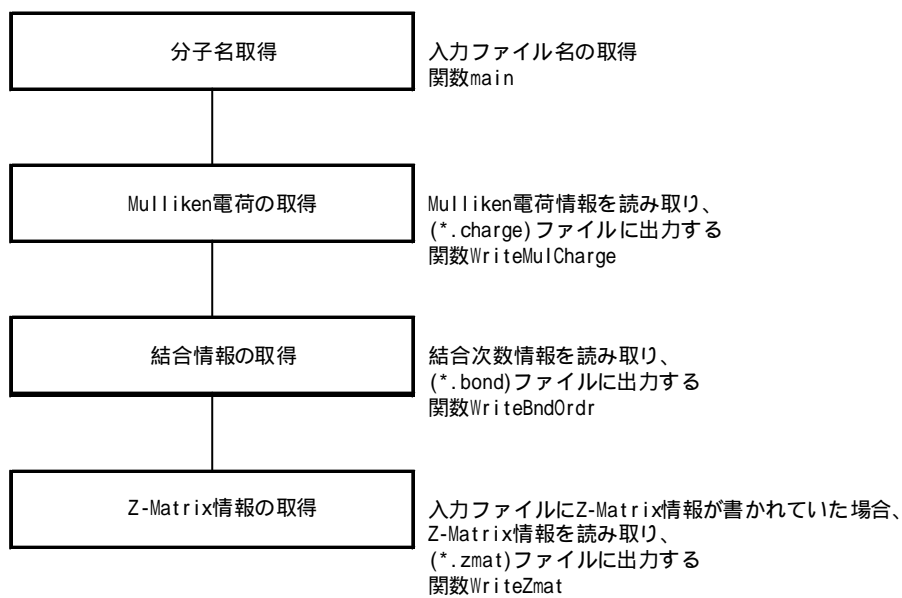


図 5 . ReadGamess の関数構成

5-1 main

機能概要

- ・低分子トポロジージェネレータ用入力ファイル作成ツール ReadGamess のメインプログラム

機能説明

- ・実行時の引数から入力ファイル名（量子化学計算ソフト GAMESS の出力ファイル名）を取得する。
- ・関数 WriteMulCharge を呼び出す。
- ・関数 WriteBndOrdr を呼び出す。
- ・入力ファイル内に Z-Matrix 情報が書かれている場合、関数 WriteZmat を呼び出す。
- ・GAMESS で構造最適化を行っていた場合は、最適化後の構造の Mulliken 電荷、結合、Z-Matrix 情報に対して、それぞれ関数 WriteMulCharge, WriteBndOrdr, WriteZmat を呼び出す。

引数

```
int argc  
char *argv[]
```

備考

なし

5-2 WriteMulCharge

機能概要

- ・ Mulliken 電荷情報をファイルに出力する。

機能説明

- ・ 入力ファイルから Mulliken 電荷情報を抽出し、電荷情報ファイル(*.charge)に出力する。

引数

FILE* fp
FILE* fpw

備考

なし

5-3 WriteBndOrdr

機能概要

- ・結合情報をファイルに出力する。

機能説明

- ・入力ファイルから結合次数情報を抽出し、結合情報ファイル(*.bond)に出力する。

引数

FILE* fp

FILE* fpw

備考

なし

5-4 WriteZmat

機能概要

- ・ Z-Matrix 情報をファイルに出力する。

機能説明

- ・ 入力ファイルから Z-Matrix 情報を抽出し、Z-matrix 情報ファイル (*.zmat) に出力する。

引数

- FILE* fp
- FILE* fpw

備考

- なし

付録 . 原子タイプ割り当て規則ファイル (抜粋)

```
!=====  
! if element(0) is C  
!=====  
if element(0) = C then  
  if nbond(0) = 1 then  
    atom_type := c1;  
    default_atom_type := 1c1;  
  endif  
  if nbond(0) = 2 then  
    atom_type := c1;  
    default_atom_type := 2c1;  
  endif  
  if nbond(0) = 3 then  
    default_atom_type := 3c1;  
    atom_type :=c2;  
  
    if element(01) = N and nbond(01) = 3 then  
      if element(02) = N and nbond(02) = 3 then  
        atom_type := ca;  
      endif  
    endif  
  
    if element(01) = O and nbond(01) = 1 then  
      atom_type := c;  
    endif  
    if aromatic(0) = 1 then  
      atom_type := ca;  
    endif  
  endif  
  if nbond(0) = 4 then  
    atom_type := c3;  
    default_atom_type := 4c1;  
  endif  
endif
```