

sievgene の機能追加

詳細設計書

2018 年 2 月 5 日

目次

1. 目的	3
2. 機能概要	4
2. 1. MD トラジェクトリ読み込み機能	4
2. 2. トラジェクトリ座標でのスコア計算機能追加	5
3. トラジェクトリ対応 sievgene の設計方針	9
4. 修正内容	10
4. 1. 修正内容一覧	10
4. 1. 1. DIAV 用 ASA 解析の取り込み	11
4. 1. 2. DIAV 解析フラグの追加	12
4. 1. 3. DIAV 用定数宣言の追加	13
4. 1. 4. DIAV 用 GB/SA 解析の取り込み	14
4. 1. 5. torsion 解析用 Z-matrix 入力追加	15
4. 1. 6. 蛋白・化合物指定ファイル読み込み	16
4. 1. 7. DIAV 用オプション入力	17
4. 1. 8. DIAV 用構造体宣言追加	19
4. 1. 9. GB/SA スケーリング処理の取り込み	20
4. 1. 10. Richmond-ASA 解析処理の削除	21
4. 1. 11. 原子質量の追加	22
4. 1. 12. frozen-atom, 蛋白・化合物指定ファイルのオプション追加	23
4. 1. 13. 主制御の変更	24
4. 1. 14. トラジェクトリ読み込みとスコア解析	25
4. 1. 15. Grid 生成表示の変更	26
4. 1. 16. DIAV 解析	27
4. 1. 17. 蛋白・化合物間 1-5 相互作用計算	28

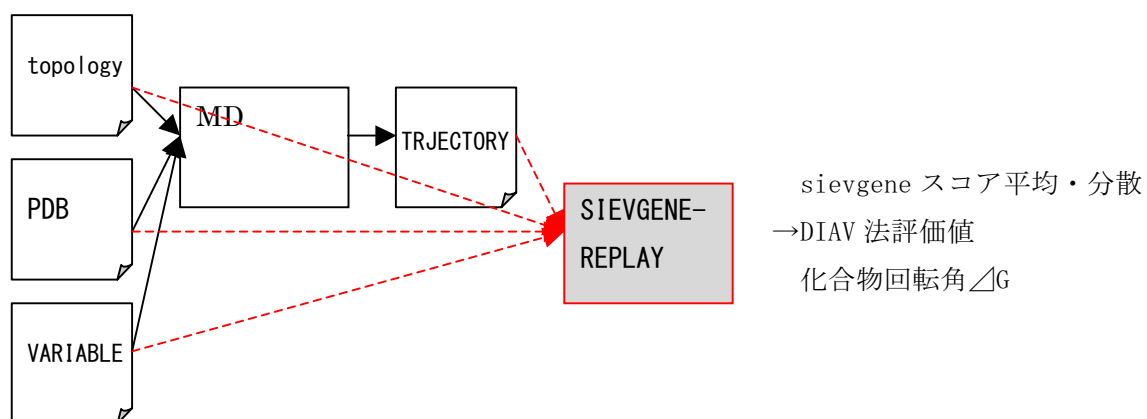
1. 目的

タンパク質・化合物ドッキングソフト myPresto/sievgene を改変し、タンパク質・化合物の MD での座標トラジェクトリを読み込み、MD 中に発生したドッキングポーズでの sievgene スコアを計算する機能を追加する。

また、座標トラジェクトリの各座標において計算された sievgene スコア平均の分散・平均値を出力する機能を追加する。

本プログラムは MD での座標を再現しながらスコア値を算出するため、仮称を SIEVGENE-REPLAY とし、以降 SIEVGENE-REPLAY と呼ぶ

SIEVGENE-REPLAY はトラジェクトリを生成した際の MD 入力と MD のトラジェクトリファイルを入力し、従来の mol2 ファイルからの化合物入力を行わないこととする。(3p に検討メモを記載)



SIEVGENE-REPLAY の入出力

2. 機能概要

2. 1. MD トrajekトリ読み込み機能

(1) インタフェース

制御ファイルは sievgene の制御ファイル仕様をベースとし、下記の記述を拡張する。

(1-1) ANALYZE フェーズの追加

sievgene に ANALYZE フェーズを追加し、“EXE>ANA”のキーワードが出現した場合にトラジェクトリ読み込み機能での DIAV 計算、sievgene スコア計算を行う。

ANALYZE フェーズではトラジェクトリファイルを DIAV 計算で有効なオプション (vdW 関数形、化合物の回転可能 torsion での ΔG 推算、各項の係数) の入力が可能とする。

(1-2) INPUT フェーズの拡張

DIAV 法で指定する蛋白・リガンド残基指定のキーワードを有効とする。

また、座標トラジェクトリが自由原子で構成された場合の対応のため、cosgene での frozen-atom 指定のキーワードを有効とする。

<pre>; control file for SIEVGENE-TRAJECTORY ; PHASE> INPUT TOPOLOGY= FORM NAMETO= Astemizole.tpl COORDINA= PDB NAMECO= Astemizole.pdb POINTC= NORE SETVAR = READ NAMEVA = test.var CMBREA= READ NAMECM= cmb.inp QUIT PHASE> GRID PROBDist = 6.5 ; MARGIN = 6.5 ; search margin ITERAT = 2 ; iteration of Grid potential smoothing RADVDW = 0.6 ; vdW boundary RADELE = 0.6 ; coulomb boundary RADMESH = 1.4 ; probe radius DAMPVW = 1.25d0 USEPBG = NO ; not use PB QUIT EXE> ANA COMBIN= YES NAMETR= md.cor DATATY= COR4 STARTT= 0.50D0 ENDTIM= 1.0D0 ESTROT = YES ; ESTimate dG of ligand ROTation COEROT = 0.2 ; COEfficient of ROTatin out of ring COERNG = 0.15 ; COEfficient of rotatin in RiNG</pre>	<p>INPUT フェーズ sievgene と共通</p> <p>frozen-atom 指定 蛋白・リガンド指定</p> <p>GRID フェーズ sievgene と共通、グリッドファイルの入出力は行わないようにシステムが設定する (READ/WRITE が指定された場合は、NORE, NOWR に設定する)</p> <p>ANA フェーズ ELIE (DIAV) 法の cosgene と共通で、“EXE>ANA”の終了後に解析処理を実行しプログラムを終了する</p>
--	--

sievgene-REPLAY の制御ファイル例

2. 2. トラジェクトリ座標でのスコア計算機能追加

トラジェクトリ読み込み機能を追加した sievgene に対し、各時刻でのタンパク質・化合物間の vdW、静電エネルギー、タンパク質の構造揺らぎを計算し、これらの値の平均と分散からコンセンサススコアを計算する DIA (Direct Interaction Approximation) の機能を追加する。

DIAV 計算のインタフェースは cosgene と共通とする。

DIA には下記の複数の手法があり、本バージョンでは現在の DIAV 法でサポート済みの (1)DIAV model, (2)DIAV_L model をサポートする。

DIAV 解析のバリエーションが増えた場合に自動で機能が実装されるように、DIAV 解析手続きは、DIAV 解析用 cosgene とソースを共通化する。

(1)DIAV model :

The original DIAV form

$$\Delta G = \alpha \sum_i \langle E^{vdW}(i) \rangle + \beta \sum_i \langle E^{ele}(i) \rangle + \tau \times S_x$$

(2)DIAV_L model :

The ligand entropy term is added to the original DIAV model

$$\Delta G = \alpha \sum_i \langle E^{vdW}(i) \rangle + \beta \sum_i \langle E^{ele}(i) \rangle + \tau \times S_x \\ + w_1 k_B T \times N_{rot} \times \ln 3 + w_2 k_B T \times (N_{rot-ring} - 3N_{ring}) \times \ln 2$$

(3)DIAV_W model :

The weight for each residue is calculated from the hydration solvent water.

$$\Delta G = \alpha \sum_i \exp(\gamma \langle E_i \rangle) \langle E^{vdW}(i) \rangle + \beta \sum_i \exp(\gamma \langle E_i \rangle) \langle E^{ele}(i) \rangle + \tau \times S_x$$

(4)DIAV_LW model :

The ligand entropy term and the weight for each residue are added to the original DIAV model. The weight for each residue is calculated from the hydration solvent water.

$$\Delta G = \alpha \sum_i \exp(\gamma \langle E_i \rangle) \langle E^{vdW}(i) \rangle + \beta \sum_i \exp(\gamma \langle E_i \rangle) \langle E^{ele}(i) \rangle + \tau \times S_x \\ + w_1 k_B T \times N_{rot} \times \ln 3 + w_2 k_B T \times (N_{rot-ring} - 3N_{ring}) \times \ln 2$$

(5)DIAV_LW2 model (直鎖と環の回転可能結合を同じに扱う) :

The ligand entropy term and the weight for each residue are added to the original DIAV model. The weight for each residue is calculated from the hydration solvent water.

$$\Delta G = \alpha \sum_i \exp(\gamma < E_i >) < E^{vdW}(i) > + \beta \sum_i \exp(\gamma < E_i >) < E^{ele}(i) > + \tau \times S_x$$

$$+ w_1 k_B T \times \{N_{rot} + (N_{rot-ring} - 3N_{ring})\} \times \ln 3$$

(6) DIAV_LW_nS model (τS なし) :

The ligand entropy term and the weight for each residue are added to the original DIAV model. The weight for each residue is calculated from the hydration solvent water.

$$\Delta G = \alpha \sum_i \exp(\gamma < E_i >) < E^{vdW}(i) > + \beta \sum_i \exp(\gamma < E_i >) < E^{ele}(i) >$$

$$+ w_1 k_B T \times N_{rot} \times \ln 3 + w_2 k_B T \times (N_{rot-ring} - 3N_{ring}) \times \ln 2$$

(7) DIAV_LW2_nS model (直鎖と環の回転可能結合を同じに扱う + τS なし) :

The ligand entropy term and the weight for each residue are added to the original DIAV model. The weight for each residue is calculated from the hydration solvent water.

$$\Delta G = \alpha \sum_i \exp(\gamma < E_i >) < E^{vdW}(i) > + \beta \sum_i \exp(\gamma < E_i >) < E^{ele}(i) >$$

$$+ w_1 k_B T \times \{N_{rot} + (N_{rot-ring} - 3N_{ring})\} \times \ln 3$$

(1) DIA 計算の出力

sievgene-REPLAY の DIA 計算結果は、DIAV 用 cosgene と同じ下記のフォーマットとする。

<pre>INFORMATION> EXTENDED LINEAR INTERACTION ENERGY total vdW average = -178.952560757154 total vdW variant = 1.38117986646209 total dielectric average = 28.7135735979065 total dielectriv variant = 0.879967198117273 total A. S. A variant = 39.8715921490601 (1) 3 term estimation (a) vdW average coefficient = 4.160000000000000E-002 (b) dielectric average coefficient = 8.999999999999999E-003 (c) A. S. A variant coefficient = -2.079700000000000E-004 energy = a * average(vdW) + b * average(dielectric) + c * variant(A. S. A) energy = -7.44442652749761 + 0.258422162381158 + -8.292095019240036E-003 @# dg_calc_3= -7.1943 (2) 5 term estimation (d) vdW average multiplier = 3.070000000000000E-002 (e) vdW variant coefficient = -1.000000000000000E-002 (f) dielectric average multiplier = 1.180000000000000E-002 (g) dielectric variant coefficient = 3.120000000000000E-003 (h) A. S. A variant multiplier = -2.431200000000000E-004 energy = SUM(d * average(vdW(i)) * EXP(e * variant(vdW(i))) + f * average(ele(i)) * EXP(g * variant(ele(i))) + h * variant(A. S. A(i))) @# dg_calc_5= -5.1631 (3) add estimate energy of ligand greedom (j) rotatable node : number = 8 : coeff = 0.100000000000000 : energy = 0.523963146752324 (k) number of rings = 5 rotatable tor. : number = 10 rotatable - 3 * rings = -5 : coeff = 0.100000000000000 rotatable ring : energy = -0.206614961913106 @# dg_calc_3+ligand= -6.8769 @# dg_calc_5+ligand= -4.8458</pre>	<p>DIAV 評価</p> <p>DIAV_W 評価</p> <p>DIAV_L 評価 DIAV_LW 評価</p>
--	---

(2) sievgene スコアの出力

sievgene-REPLAY のスコア出力は、各ステップでのスコアと平均と分差を下記のフォーマットで出力する。

<pre>INFORMATION> GRID SIZE NUMBER OF GRID : 60 SAMPLING POINT NUMBER : 34 GRID BOUND : X-AXIS: -8.583733 - 9.928724 Y-AXIS: -11.04456 - 11.70127 Z-AXIS: -2.849600 - 19.23287 1 PARTICLE SIZE: 0.3137704 0.3855225 0.3742791 INFORMATION> SEARCH A. S. A FOR GRID POTENTIAL A. S. A NUMBER : 19305 INFORMATION> GENERATING A. S. A. GRID 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% INFORMATION> GENERATING COULOMB GRID 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% INFORMATION> GENERATING HYDROGEN GRID 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% ASA: -277.55 ELE: 5.71 HYD: -4.33 VDW: -6.80 SCR: -282.97 : INFORMATION> GRID SIZE NUMBER OF GRID : 60 SAMPLING POINT NUMBER : 34 GRID BOUND : X-AXIS: -8.631214 - 10.74198 Y-AXIS: -11.12636 - 11.51924 Z-AXIS: -2.367275 - 19.19534 1 PARTICLE SIZE: 0.3283593 0.3838237 0.3654680 INFORMATION> SEARCH A. S. A FOR GRID POTENTIAL A. S. A NUMBER : 19289 INFORMATION> GENERATING A. S. A. GRID 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% INFORMATION> GENERATING COULOMB GRID 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% INFORMATION> GENERATING HYDROGEN GRID 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% ASA: -270.13 ELE: 5.27 HYD: -2.76 VDW: -4.53 SCR: -272.16 : #@# SCORE AVERAGE : -277.2994 VARIANT : 31.84532</pre>	<p>1 ステップ目 Grid 生成</p> <p>1 ステップ目スコア</p> <p>n ステップ目 Grid 生成</p> <p>n ステップ目スコア</p> <p>スコア平均と分散</p>
--	---

3. トラジェクトリ対応 sievgene の設計方針

SIEVGENE-REPLAY は sievgene_for_screening に対し、DIAV 用 cosgene の DIAV 法計算手続きを組み込むことで実装する。

下記の図に、SIEVGENE-REPLAY の制御フローと、DIAV 法の利用機能の関連を示す。

SIEVGENE-REPLAY 主処理	
<ul style="list-style-type: none"> ・ 蛋白読み込み ・ グリッドオプション読み込み ・ 化合物トポロジー生成 ・ 複合体トポロジー生成 	<ul style="list-style-type: none"> sievgene 処理 sievgene 処理 sievgene 処理 sievgene 処理
do トラジェクトリ <ul style="list-style-type: none"> ・ GB スケーリング用相互作用のサンプリング enddo	DIAV 法機能を流用
do トラジェクトリ <ul style="list-style-type: none"> ・ 静電・vdW 相互作用・構造変化のサンプリング enddo	DIAV 法機能を流用
<ul style="list-style-type: none"> ・ 静電・vdW・構造変化の平均算出 	
do トラジェクトリ <ul style="list-style-type: none"> ・ 静電・vdW 相互作用・構造変化の分散計算 enddo	DIAV 法機能を流用
<ul style="list-style-type: none"> ・ DIAV 計算 	DIAV 法機能を流用
do トラジェクトリ <ul style="list-style-type: none"> ・ グリッドポテンシャル生成 ・ sievgene スコアサンプリング enddo	sievgene 処理
<ul style="list-style-type: none"> ・ sievgene スコア平均分散計算 ・ sievgene 結果出力 	<ul style="list-style-type: none"> sievgene 処理 sievgene 処理

4. 修正内容

4. 1. 修正内容一覧

sievgene-REPLAY 開発のため、sievgene_for_screening に下記の修正を行う。

項番	修正内容	対象ファイル	区分
1	DIIV 用 ASA 解析の取り込み	AccessibleSurface_Method. f90	流用
2	DIIV 解析フラグの追加	Analyze_Class. f90	改造
3	DIIV 用定数宣言の追加	Enum_Type. f90	改造
4	DIIV 用 GB/SA 解析の取り込み	GeneralizedBorn_Method. f90	流用
5	torsion 解析用 Z-matrix 入力追加	Object_Class. f90	改造
		Input_Data. f90	改造
		Output_Data. f90	改造
6	蛋白・化合物指定ファイル読み込み	Input_Data. f90	流用
7	DIIV 用オプション入力	Input_Option. f90	改造
8	DIIV 用構造体宣言追加	Interact_Class. f90	改造
9	GB/SA スケーリング処理の取り込み	Object_Method. f90	流用
10		Control_Min. f90	改造
11	Richmond-ASA 解析処理の削除	Score_Method. f90	改造
12	原子質量の追加	SievEnum_Type. f90	改造
13	frozen-atom, 蛋白・化合物指定ファイルのオプション追加	SievInput_Option. f90	改造
14	主制御の変更	sievgene. f90	改造
15	トラジェクトリ読み込みとスコア解析	SievAnalyze_Method. f90	新規
16	Grid 生成表示の変更	Grid_Method. f90	改造
17	DIIV 解析	Analyze_Method. f90	流用
		Control_Analyze. f90	流用
18	蛋白・化合物間 1-5 相互作用計算	Combine_Method. f90	流用

4. 1. 1. DIAV 用 ASA 解析の取り込み

DIAV 計算用の `AccessibleSurface_Method.f90` を `sievgene` のコンパイル対象ソースに追加する。

4. 1. 2. DIAV 解析フラグの追加

Analyze_Class.f90 の Analyze_t 構造体に DIAV 解析実行フラグ IsAnalyzeCombine を追加する。

4. 1. 3. DIAV 用定数宣言の追加

Enum_Type.f90 に DIAV (旧 LIE) のスケーリング方法、スケーリング関数、vdW 関数形を示す定数とそれぞれの値を示す文字列を追加する。

```
!  
! LIE scaling method  
!  
integer, parameter::NO_SCALING      = 1  
integer, parameter::PROJECTION_SCALING = 2  
integer, parameter::SIZE_SCALING    = 3  
integer, parameter::ELE_PROJECTION  = 4  
integer, parameter::ELE_SIZE        = 5  
  
character (17), dimension(ELE_SIZE), parameter::          &  
      GBSAScalig_Method = (' NO SCALING      ',          &  
      ' PROJECTION (GB) ',          &  
      ' SIZE SCALING (GB) ',          &  
      ' PROJECTION (ELE) ',          &  
      ' SIZE SCALING (ELE) ' /)  
  
!  
! for LIE scaling function  
!  
integer, parameter::LIE_ORIGINAL = 1  
integer, parameter::LIE_LINEAR   = 2  
character (17), dimension(LIE_LINEAR), parameter::          &  
      LIE_FUNCTION = (' EXPONENTIAL ',          &  
      ' LINEAR     ' /)  
  
!  
! for LIE vdW function  
!  
integer, parameter::LJ_12_6 = 1  
integer, parameter::LJ_9_6  = 2  
integer, parameter::LJ_8_4  = 3  
character (17), dimension(LJ_8_4), parameter::          &  
      LJ_POTENTIAL = (' 12 - 6 ',          &  
      ' 9 - 6 ',          &  
      ' 8 - 4 ' /)
```

4. 1. 4. DIAV 用 GB/SA 解析の取り込み

DIAV 計算用の GeneralizedBorn_Method.f90 を sievgene のコンパイル対象ソースに追加する。

4. 1. 5. torsion 解析用 Z-matrix 入力追加

DIAB の torsion 揺らぎ解析を実行するため、原子に対する Z-matrix 情報、Z-matrix 読み込み、Z-matrix 出力処理を追加する。

(1) 原子に対する Z-matrix 情報追加 : Object_Class.f90

(2) Z-matrix 入力処理追加 : Input_Data.f90

(3) Z-matrix 出力処理追加 : Output_Data.f90

4. 1. 6. 蛋白・化合物指定ファイル読み込み

Input_Data.f90 に DIAV 用 cosgene の蛋白・化合物指定ファイル読み込み手続き
Input_CombineFile 手続きを追加する。


```

'CMBCOE', & ! average(vdW) coefficient
'LIEAVC', & ! average(vdW) coefficient
'LIEAEC', & ! average(ele) coefficient
'LIEVAC', & ! variant(ASA) coefficient
'LIEAVM', & ! average(vdW) multiplier
'LIEVVC', & ! variant(vdW) coefficient
'LIEAEM', & ! average(ele) multiplier
'LIEVEC', & ! variant(ele) coefficient
'LIEVAM', & ! variant(ASA) multiplier
'ENDTIM', & ! end time of trajectory
'COEROT', & ! coefficient for rotate
'COERNG' /) ! coefficient for ring

call Init_Value(sopt, SelectOption, 'CMBSCl', 'NO ')
call Init_Value(sopt, SelectOption, 'CMBFUN', 'EXP0')
call Init_Value(ropt, RealOption, 'CMBCOE', 0.25d0)

call Init_Value(sopt, SelectOption, 'CMBVDW', '8-4 ')

call Init_Value(ropt, RealOption, 'LIEAVC', 0.0416d0)
call Init_Value(ropt, RealOption, 'LIEAEC', 0.0090d0)
call Init_Value(ropt, RealOption, 'LIEVAC', -2.0797d-4)
call Init_Value(ropt, RealOption, 'LIEAVM', 0.0307d0)
call Init_Value(ropt, RealOption, 'LIEVVC', -0.0100d0)
call Init_Value(ropt, RealOption, 'LIEAEM', 0.0118d0)
call Init_Value(ropt, RealOption, 'LIEVEC', 0.00312d0)
call Init_Value(ropt, RealOption, 'LIEVAM', -2.4312d-4)

call Init_Value(sopt, SelectOption, 'ESTROT', 'NO ')
call Init_Value(ropt, RealOption, 'COEROT', 0.1d0)
call Init_Value(ropt, RealOption, 'COERNG', 0.1d0)

call Set_Value(sopt, SelectOption, 'CMBSCl', Interact%GBSAScaling)
call Set_Value(sopt, SelectOption, 'CMBFUN', Interact%GBSAScaleFunc)
call Set_Value(ropt, RealOption, 'CMBCOE', Interact%GBSAScaleCoeff)

call Set_Value(sopt, SelectOption, 'CMBVDW', Interact%GBSATypeLJ)

call Set_Value(ropt, RealOption, 'LIEAVC', Interact%ELIE%AveVdwCoeff)
call Set_Value(ropt, RealOption, 'LIEAEC', Interact%ELIE%AveEleCoeff)

call Set_Value(ropt, RealOption, 'LIEVAC', Interact%ELIE%VarAsaCoeff)
call Set_Value(ropt, RealOption, 'LIEAVM', Interact%ELIE%AveVdwMult)
call Set_Value(ropt, RealOption, 'LIEVVC', Interact%ELIE%VarVdwCoeff)
call Set_Value(ropt, RealOption, 'LIEAEM', Interact%ELIE%AveEleMult)
call Set_Value(ropt, RealOption, 'LIEVEC', Interact%ELIE%VarEleCoeff)
call Set_Value(ropt, RealOption, 'LIEVAM', Interact%ELIE%VarAsaMult)

call Set_Value(sopt, SelectOption, 'ESTROT', Interact%ELIE%EstimateLigand)
call Set_Value(ropt, RealOption, 'COEROT', Interact%ELIE%NodeCoeff)
call Set_Value(ropt, RealOption, 'COERNG', Interact%ELIE%RingCoeff)

```

4. 1. 8. DIAV 用構造体宣言追加

Interact_Class.f90 に DIAV 用の構造体 ELIE_t を追加する。

```
!
! extended linear interaction energy
!
! energy = A * ave(vdW) + B * ave(ele) + C * var(ASA)
!
! or
!
! energy = D * ave(vdW) * EXP(E * var(vdW))
!           + F * ave(ele) * EXP(G * var(ele))
!           + H * var(ASA)
!
type ELIE_t
  real*8::AveVdwCoeff      ! A
  real*8::AveEleCoeff     ! B
  real*8::VarAsaCoeff     ! C
  real*8::AveVdwMult      ! D
  real*8::VarVdwCoeff     ! E
  real*8::AveEleMult      ! F
  real*8::VarEleCoeff     ! G
  real*8::VarAsaMult      ! H

  integer::RingNum        ! number of rings
  integer::RotRing        ! number of rotatable torsion in ring
  integer::RotNode        ! number of rotatable torsion out ring
  logical::EstimateLigand ! estimate ligand freedom
  real*8::RingCoeff       ! coefficient of ring torsion
  real*8::NodeCoeff       ! coefficient of node torsion

end type ELIE_t
```

4. 1. 9. GB/SA スケーリング処理の取り込み

GB/SA スケーリング用手続き"Set_GBSAScale"が実装されている DIAV 用 cosgene の Object_Method.f90 を差し替える。

本変更に伴い、DIAV 用 cosgenen のベストフィット処理 Calc_LeastSquareFitting の引数が変更されているため、sievgene で Calc_LeastSquareFitting を使用している Control_Min.f90 の引数を変更する。

4. 1. 10. Richmond-ASA 解析処理の削除

Score_Method.f90 では Richmond-ASA 解析を AccessibleSurface_Method.f90 の手続きを使用していた。従来の AccessibleSurface_Method.f90 と DIAV 法の AccessibleSurface_Method.f90 はインタフェースが異なり、sievgene-REPLAY ではスコア解析を行わないため、これらの手続き呼び出しをコメントアウトする。

4. 1. 1 1. 原子質量の追加

sievgene では mol2 ファイルの原子名から原子番号を特定しているが、cosgene のトポロジーファイルでは原子名は蛋白の原子名 (HA, CA 等が存在) となっているためと元素の対応が付かない。

質量を利用して各原子と原子番号の対応をとるために、質量の情報を SievEnum_Type.f90 の原子テーブルに追加する。

4. 1. 12. frozen-atom, 蛋白・化合物指定ファイルのオプション追加

SievInput_Option.f90 の Input フェーズ入力に frozen-atom および蛋白・化合物指定ファイルの入力機能を追加する。

4. 1. 13. 主制御の変更

sievgene.f90 のメイン処理に下記の変更を行う。

(1)Grid フェーズ入力時の処理から Grid 作成処理を削除する。

sievgene-REPLAY ではトラジェクトリを読み込みながら Grid を生成するため、Grid フェーズ入力時の Grid 生成は行わない。

(2)Analyze フェーズ入力処理を追加し、入力後に DIA 解析および各トラジェクトリでの sievgene スコアを計算する。

4. 1. 14. トラジェクトリ読み込みとスコア解析

トラジェクトリファイルを読み込みながらスコア計算を行い、全トラジェクトリでの sievgene スコアの平均と分散を出力する手続き Analyze_SievgeneScore を追加する。

Analyze_SievgeneScore の処理フローを以下に示す。

- (1) 化合物単体の原子トポロジーを作成
- (2) 化合物単体の相互作用トポロジーを作成
- (3) 各ステップのトラジェクトリファイルを入力し、sievgene スコアを計算
 - ・当該ステップのリガンド座標をポケットとしグリッドを生成
 - ・当該ステップのリガンド座標でのスコア計算
 - ・ASA 計算補正
- (4) スコアの平均と分散の計算と表示

4. 1. 15. Grid 生成表示の変更

グリッド生成ステップのパーセンテージ表示を1行に出力する変更を行う。

4. 1. 16. DIAV 解析

DIAV 解析用手続き Analyze_Main を使用するため、Control_Analyze.f90 と Analyze_Method.f90 をコンパイル対象ファイルに追加する。

4. 1. 17. 蛋白・化合物間 1-5 相互作用計算

蛋白・化合物相互作用計算解析用手続き Combine_Main を使用するため、Combine_Method.f90 をコンパイル対象ファイルに追加する。

以上