

myPresto/sievgene 方式設計書
(ドッキングエンジン編)

第4版

目次 -

1. はじめに	2
1.1 ドッキングエンジンの目的	2
1.2 ドッキングエンジンの概要	2
1.3 処理内容	4
1.4 処理フロー	5
2. 探索方法	7
2.1 Global search	7
2.2 Local search	14
3. 原子タイプ	15
4. スコア関数系	16
4.1 概要	16
4.2 van der Waals score	18
4.3 Coulomb score	19
4.4 ASA のスコア計算	21
4.5 水素結合のスコア計算	21
4.6 Grid ポテンシャルのスミージング	21
5. スコアの計算	22
6. 結果の表示	23
7. 配座の生成方法	24
7.1 Z-matrix の生成	24
7.2 環の検出と開環	26
7.3 複数配座の大量生成	26
8. リガンド分子のトポロジー自動生成	27
9. 入出力形式	28
10. 制約事項	34

1. はじめに

本設計書では、蛋白質 化合物ドッキングエンジン(sievgene)の方式について説明する。

1.1 ドッキングエンジンの目的

計算機による低分子化合物データベースを用いた蛋白質とリガンドの仮想的ドッキングによる低分子化合物の選択は、現在の HIT 化合物探索において、重要なプロセスだが、十分な精度と速度を満たすことは一般に困難である。複合体形成時に起こりうる induced fit は、ligand をフレキシブルにするドッキングにおいても十分に考慮されず複合体構造の精度を低下させ、活性値は各種の近似的・経験的な方法では十分に表現されない。

本ソフトウェアの目的は以下の通りである。ここで、厳密な実験データの再現が目的ではない点に注意されたい。

- (1) 構造変化に鈍感な蛋白質表面のラフな形状を考慮するリガンドの配座のみをフレキシブルとしたドッキングを行う、in silico スクリーニングのための蛋白質・低分子有機化合物ドッキングの実現。
- (2) 蛋白質・低分子有機化合物結合自由エネルギーないし活性値のラフな見積り。

1.2 ドッキングエンジンの概要

蛋白質 リガンド相互作用は、

- ・ van der Waals (vdW) ポテンシャル、
- ・ クーロン引力、
- ・ 水素結合、
- ・ 疎水性相互作用

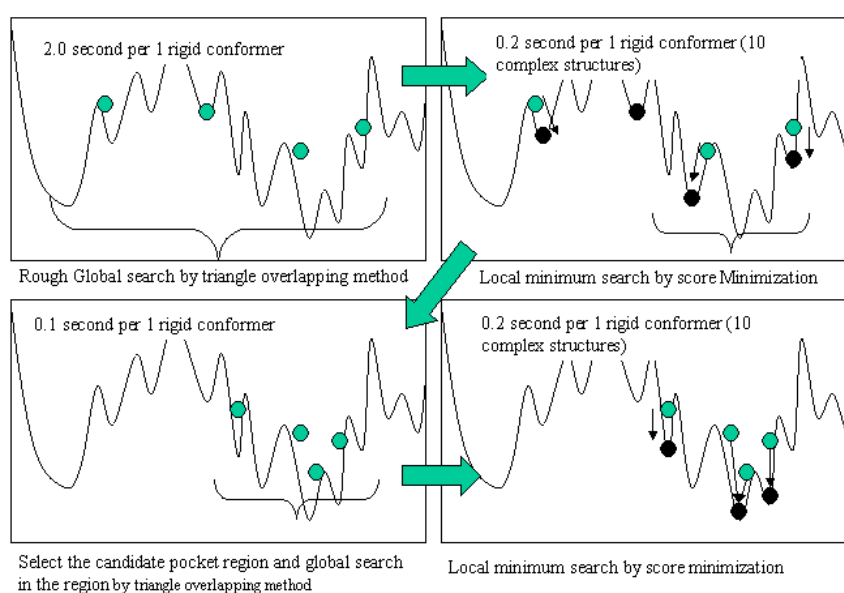
に分け、レセプター(蛋白質)の作る力場には Grid ポテンシャルを適用する。vdW 相互作用、クーロン相互作用の情報を Grid に載せるため、AMBER の原子タイプに準拠した約 10~20 種類の相互作用のタイプを用意し、補間多項式を適用する。蛋白質の表面よりも外側の領域は AMBER タイプのポテンシャル形を適用するが、蛋白質の内側の領域では、表面からの距離の関数を用いて滑らかなポテンシャルが形成されるようにする。ポテンシャル関数は、蛋白質表面において解析的に接続され連続であるように作る。

探索方法は、ラフな Global minimum search と Local minimum search の 2 段階から構成している。ラフな Global minimum search では、複合体形成時には少なくともリガンドの 3 原子が蛋白質側の原子と接触するものとする。蛋白質側の分子表面の静電ポテンシャル

が極大・極小な点と静電ポテンシャルがほぼゼロである点をほぼ同数選び、これらをダミー原子とする。これらダミー原子の任意の3点のなす三角形を Hash テーブルに格納し、リガンド側の3原子が対応しうる三角形を検索し、リガンド分子を蛋白質側の表面に重ね合わせて構造探索する方法（幾何学的 Hash）を用いる。リガンド分子は、環の自由度以外の直鎖部分をフレキシブルに変形できるフレキシブルドッキングを適用する。

更に、異なる conformer でも同一のポケット近辺に結合すると仮定し、pocket の絞込みを行う方法を組合せる。すなわち、広い範囲でラフなドッキングを行った後、スコアのよいドッキング位置周辺を新たに探索範囲に選定し、その範囲内でドッキングを再開する。

また、同時に、良いスコアを記録した配座に類似の配座を生成し、ドッキングを行う。



探索の過程は、ラフな探索において、

- (1) リガンド分子の配座の選択
- (2) リガンド分子の併進・回転による位置の選択
- (3) 蛋白質がわのドッキング探索範囲（ポケット）の特定・絞込み

の3種類を行い、その後全原子を free にした energy minimization を行う。

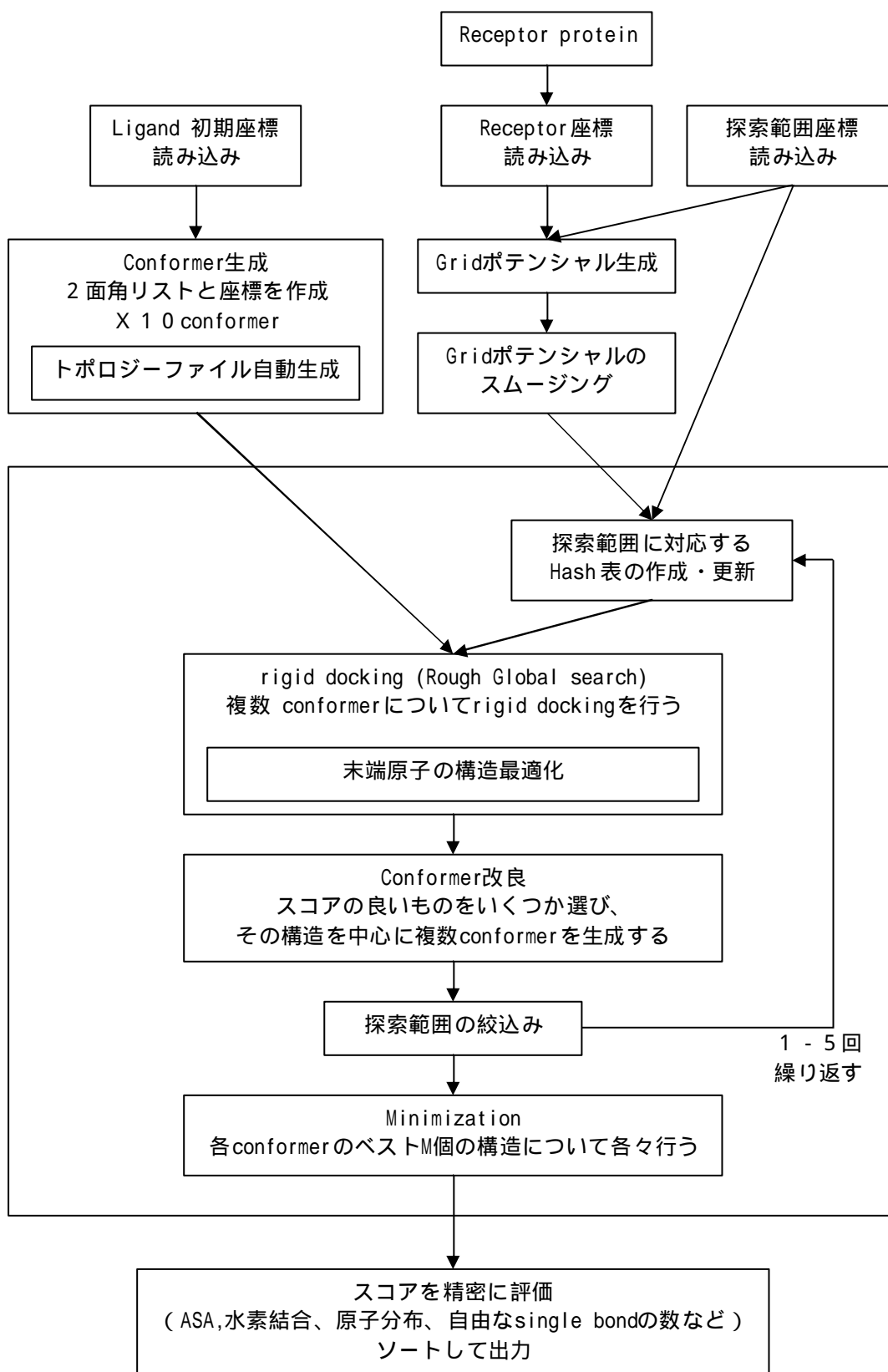
最後に、スコア上位の化合物座標を mol2 ファイル形式または PDB ファイル形式で、各種スコアとともに出力する。

1.3 処理内容

- (1) リガンド分子は、PDB/topology ファイルまたは mol2 形式ファイルで準備する。リガンド分子の入力形式は 3 種 (複数分子を含む PDB 類似、複数分子を含む mol2、及び、複数のファイル名のリスト)
- (2) 蛋白質側は、ポケット領域の指定ファイルと PDB/topology ファイルまたは mol2 形式ファイルを準備する。ポケット領域の指定ファイルは、別ソフトウェアにて事前に準備する。
- (3) ポケット領域の指定ファイルに従って、レセプター側の領域を特定し、Grid ポテンシャルを形成する。
- (4) レセプター表面に、リガンドの原子が接触しうる点を離散的に生成する。これらの点での静電ポテンシャルを計算し、電位の極大値をとる点のうち絶対値の大きい点から N1 個、電位の絶対値の小さい点 (電位がほぼゼロの点) から N2 個を選び出す。これら N1+N2 個の点がどのレセプター原子の上に接触するかを記録する。それらの座標と、原子タイプを記録、Hash 表を作成する。
- (5) Grid ポテンシャルなどのレセプターの情報はファイル出力を可能とし、リスタート時には、レセプター座標がなくともリガンド座標情報のみでドッキング計算が再開できるようにする。
- (6) リガンド分子を順番に読み込み、各リガンド分子に対し、一定の数の複数配座を生成し、ドッキングを行う。ドッキングの結果は、リガンドごとにスコアの良いもの順に記録する。
- (7) 準備した配座のドッキングの終了後、その中で良いスコアを与える座標と配座を選び、その座標周辺にポケットの探索領域を絞る。また、新たに配座を発生させ、良いスコアを与えた配座に類似のものを選ぶ。
- (8) 新しい探索領域で、これら配座のドッキングを行う。
- (9) 一定の回数、上記の操作を繰り返した後、良いスコアを与える結果に対して minimization を行う。その際、座標を少し、ランダムに座標を変化させ、複数の初期座標から Grid ポテンシャル上で、構造最適化を行う。minimization の対象数は、外部入力を与えるものとする。
- (10) 結果をスコアの良い順にソートし出力する。座標、総合スコア、及び、各スコア成分を別々に出力する。出力座標数および出力スコア数はそれぞれ独立に外部入力を与えるものとする。
- (11) 1 リガンドのドッキングが終了すると次のリガンドのドッキングを行う。
mol2 ファイルの書式については以下の URL を参照。

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

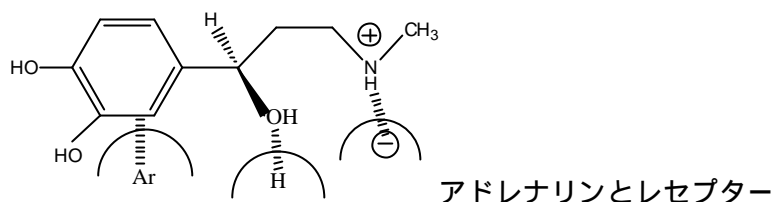
1.4 処理フロー



2. 探索方法

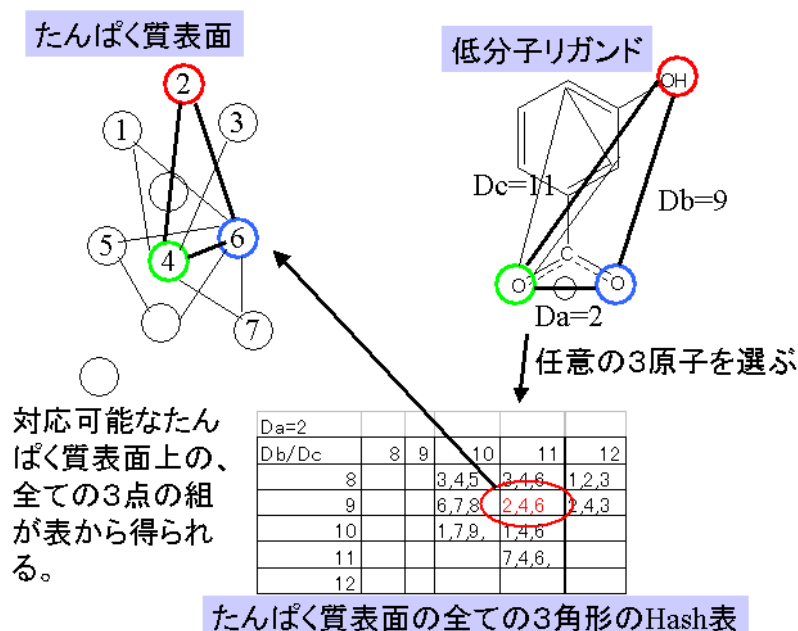
2.1 Global search

蛋白質に化合物が結合するとき、化合物の立体構造が固定されるには、蛋白質と化合物が最低3点で接触しなければ位置が定まらない。こういった性質を利用しもっともらしい蛋白質 リガンドの空間的配置を探索する。



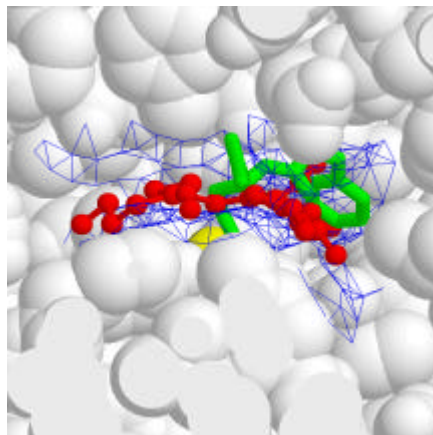
蛋白質側は、「リガンド側相手の原子がレセプター原子に接触しうる場所 (mesh 点: ダミー原子)」の点の集合を準備する。

リガンド分子の任意の3原子を選び、これら3原子の作る三角形とほぼ等しい三角形を形成しうる蛋白質表面の3点を上記集合の中から探す。リガンド側の3原子と、蛋白質表面で対応する3点の RMSD が最小となるようにリガンド分子を配置し、蛋白質 リガンドの相互作用を見積る。衝突のないスコアの良いものを選ぶ。衝突のある場合は、リガンド分子の衝突した部分を調べ新たな conformer を生成してやり直す。



2.1.1 探索範囲の発生

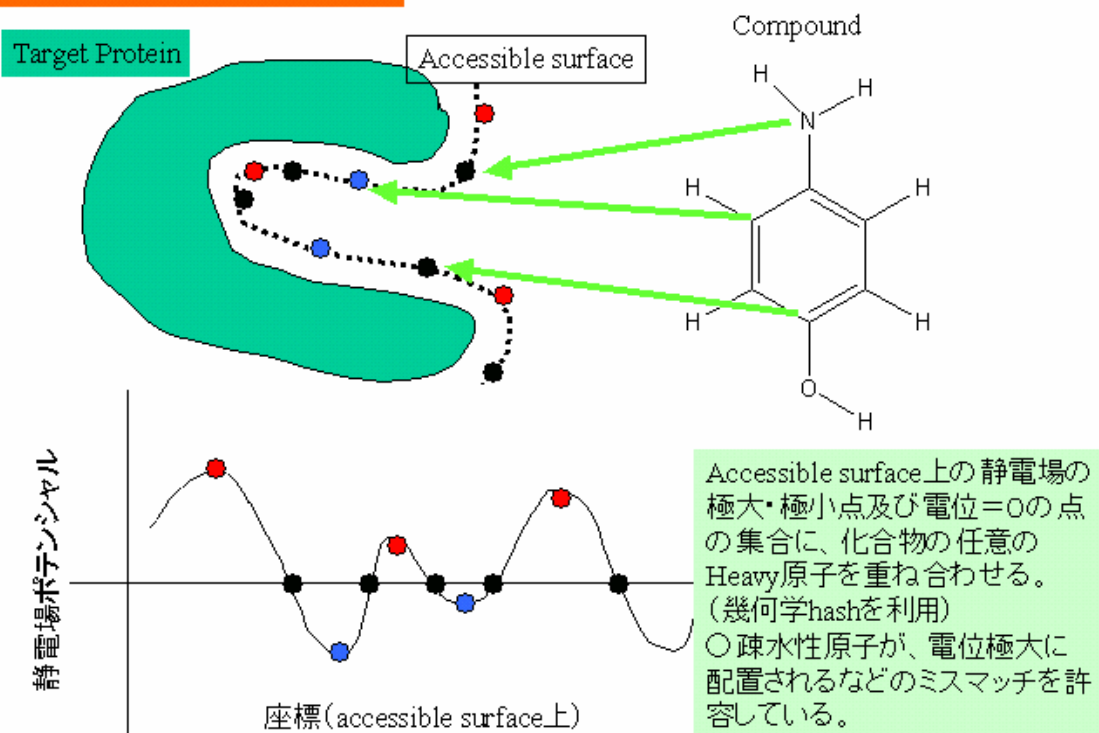
ポケットを認識させるためのプローブ点から一定の距離にある原子に対して、mesh 点を発生させる。mesh 点を発生させるためのパラメータである、プローブ点からの距離、及び、mesh 点の間隔、は外部から与えられるものとする。



上図は、レセプター側のポケット中心の原子（黄色，sphere）を中心に半径 8 の範囲にプローブ点（赤，ball and stick）を 24 点発生。更に、これらの点から半径 4 以内の領域に「リガンド側相手の原子が存在しうる場所（mesh 点）」の集合を、2 間隔で生成した（青，wire frame）場合。

これら mesh 点での静電ポテンシャルを計算し、電位の極大値をとる点のうち絶対値の大きい点から N1 個、電位の絶対値の小さい点（電位がほぼゼロの点）から N2 個を選び出す。これら N1+N2 個の点がどのレセプター原子の上に接触するかを記録する。N1, N2 は、動作モードにより異なるが、20-30 点である。蛋白質と薬物の結合においては、疎水性相互作用が結合エネルギーを稼ぐのに重要であり、親水性相互作用は特異性を出すのに重要である。Sievgene では、特に蛋白質表面と化合物表面の相補性に重点を置き、疎水性相互作用を大きく評価するように構成されている。

Global search



NOTE

プローブ点の生成は別ソフトウェアで、事前に行う。

プローブ点の座標は PDB 書式で保存され、sievene で読み込まれる。

2.1.2 Hash table の生成

$N1+N2$ 個の選ばれた mesh 点の集合に含まれる任意の 3 点組 $i-j-k$ に対し、 $i-j$, $j-k$, $i-k$ の 3 辺の長さを $dist_{ij}$, $dist_{jk}$, $dist_{ik}$ のように整数化し、配列 G に格納する (計算では、長さ 3 以上 12 以下の辺のみを 0.5 刻みで整数化している)。serial は、同じ三角形を与える点の組が $i-j-k$ 以外にもある場合に、それらの組を見分けるため通し番号を振って区別するためである。

$$i = G(dist_{ij}, dist_{jk}, dist_{ik}, 1, serial)$$

$$j = G(dist_{ij}, dist_{jk}, dist_{ik}, 2, serial)$$

$$k = G(dist_{ij}, dist_{jk}, dist_{ik}, 3, serial)$$

辺の長さの上限値・下限値・及び刻み幅は可変とし、入力で変更できるようにする。こうしておけば、任意の 3 角形を指定したとき、その三角形を形成しうる蛋白質表面の 3 点 i, j, k の組が配列 G より見出せる。同じサイズの三角形を与える 3 点の組は多数存在するため、この配列のサイズは非常に大きい。上記の例では、整数型 $20 \times 20 \times 20 \times 3 \times 1500$ である。

mesh 点には、それぞれの原子・原子タイプの表面にあるかを記載する。

3 点がどの原子タイプの上にあるかによって、不要な三角形を選別する。リガンドの疎水性原子が蛋白質の疎水性原子上に、リガンドの親水性原子が蛋白質の親水性原子上にあるとき大きな値を割り振り、疎水性原子と親水性原子が重なる場合に小さい値を割り振る。

< 例 >

リガンド原子 蛋白質原子の対毎にスコアを振り、3 つの mesh 点が原子 1, 2, 3 の上にあり、そのスコアが

$score(1)$, $score(2)$, $score(3)$ であるとき、

$$score(1) + score(2) + score(3) < \text{一定値}$$

であるならば、その三角形を採用する。具体的には「2.1.3 リガンド分子の 3 原子の選択と mesh 点とのマッチング」参照。

生成した Hash 表は、ファイル出力(アスキー形式、バイナリー形式)できるようにする。プログラムは、レセプターの PDB が入力されなくても、ファイルに保存された Hash 表の情報を使って、リスタートできるようにする。

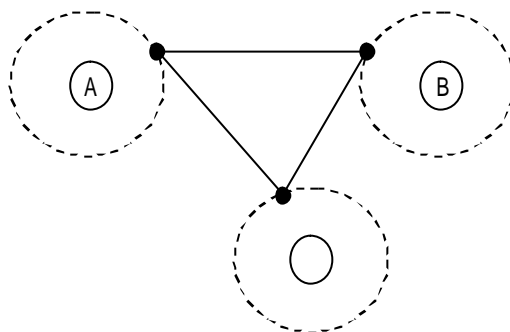
mesh 点の所属する原子の指定

特定の原子表面上の mesh 点が探索の対象になるように、mesh 点の所属する原子を指定できるようにする。指定した原子表面上の mesh 点を三角形の点として採用し、Hash テーブルに登録する。1 ~ 3 個までの原子を指定し、それぞれの原子表面上の mesh 点から 1 つずつ選択するようにする。

< 例 >

2つの原子A、Bを指定した場合：

原子A表面上の mesh 点1つと、原子B表面上の mesh 点1つ、そして、原子A、B以外の原子表面上の mesh 点1つ、からなる三角形を採用する。



2.1.3 リガンド分子の3原子の選択と mesh 点とのマッチング

リガンド分子の H 以外の原子 a,b,c を任意に 3 原子選ぶ。この 3 原子を結んで作られる三角形の辺の長さを整数化し e_ab, e_cb, e_ac とする。この三角形に合致する蛋白質表面の点は、

$$i1 = G(e_{ab}, e_{cb}, e_{ac}, 1, \text{serial})$$

$$j1 = G(e_{ab}, e_{cb}, e_{ac}, 2, \text{serial})$$

$$k1 = G(e_{ab}, e_{cb}, e_{ac}, 3, \text{serial}) \text{ for serial}=1 \sim \text{max number of pairs}$$

i1, j1, k1 である。a i1, b j1, c k1 に重なる可能性がある。

また、蛋白質 (mesh) 側の原子と、リガンド側原子の、原子タイプの対に対し、下記のようなスコア表を準備し、3つの原子対に対し、各々の原子タイプによってスコア表のスコアを加算して、ある一定値以上のスコアの得られるマッチングを採用する。疎水性の原子同士 (C - C のような場合)、親水性の原子同士 (N - N のような場合) が重なる場合に高いスコアを与える。疎水性 親水性のペアでなくても、CH- 相互作用や、カチオン- 相互作用が期待される H、金属と C との間にもある程度のスコアを与える。

	OH	NH/PH/SH	H/Na/Mg/Ca	N/O/F	C/S	I/Cl/Br	P
OH	0	0	0	2	1	2	0
NH/PH/SH	0	0	0	2	1	2	0
H/Na/Mg/Ca	0	0	1	1	1	0	1
N/O/F	2	2	1	2	0	1	2
C/S	1	1	1	0	2	0	0
I/Cl/Br	2	2	0	1	0	0	0
P	0	0	1	2	0	0	0

OH/NH/PH/SH などは、OH の H, NH の H, PH の H, SH の H を表す。

リガンド分子の大きさによって三角形の比較の試行の回数には大きな差が出るため、試行回数をおおまかにそろえるために、辺の長さの上限値・下限値を自動調整する。試行回数が多すぎる場合は、上限値を長く、下限値を短くするように調整する。

3点 a,b,c が i1, j1, k1 に対して RMSD が最小に重なり合うようにリガンド分子全体を移動・配置する。4 × 4 の行列対角化を Jacobi 法、Housholder 法等で行う。

2.1.4 末端原子の回転による構造の最適化

リガンド分子をある位置にドッキングした時、分子の端（化学結合が1つしかないもの）を回転させ、エネルギースコアの最適化を行う。

（1）リガンド -OH 基の回転

リガンド -OH 基を回転させて、配座をたくさん発生させる。

機能の ON/OFF は、実行時に指定できるようにする。

（2）蛋白質の側鎖の回転

リガンドと水素結合可能な側鎖を回転させる。

機能の ON/OFF は、実行時に指定できるようにする。

各原子毎のスコアは、マッチングの際に記録しておくようにする。

2.1.5 座標の微調整によるエネルギースコアの最適化

リガンド分子をある位置にドッキングした後、リガンドを少しずつ移動させながらスコアを再評価する。

以下を繰り返しながら、スコアを評価する。

- ・リガンドを各軸方向に移動させる。
- ・リガンドを3次元回転させる。

各軸を移動する範囲は、実行時に指定できるようにする。

2.2 Local search

Global searchによって得られた構造は、スコアの順にソートされ、スコアの良いものから一定個数の複合体が、Local minimum searchの対象となる。

topologyファイルをもとに、Grid ポテンシャル上で、最急降下法ないし共役勾配法を使用して分子構造の最適化を行う。Gridポテンシャルは、ラグランジュ1次補間ないし、より高次のスプライン補間によってgradient計算を行う。

入力ファイルがmol2で、リガンドのtopologyファイルがない場合、初期入力のリガンド構造を元に、Bond/angleに、初期構造の分子構造のbond/angleを平衡値とする適当なトポロジーを自動生成する。

NOTE

1次補間の場合、収束が得られない場合が多いので、収束のパラメーター、補間の回数には注意を要する。

3. 原子タイプ

(1) 原子タイプの割当て

原子タイプは、レセプター側、リガンド側とも、AMBER に準拠した原子タイプの割り当てを行うが、更に H-donor、H-acceptor の区別を付加する。ただし、半径・パラメーターの類似のものを同一とみなして整理し、原子タイプを簡略化したものを用いる。これは、将来的に割り振りが変更できるような仕様で作成する。

sievgene での原子の分類。グリッドポテンシャルの数を減らすため、原子の VDW 半径の類似のものを 1 グループにまとめている。

- 1) H (H0)
- 2) H (NH, SH, PH)
- 3) H (CH), Na, Mg, Ca
- 4) N, O, F
- 5) C, S
- 6) I, Cl, Br
- 7) P

(2) United atom model

「All atom」だけではなく、All atom での入力から "-CH (-CH, -CH₂, -CH₃)" の "H" をやや半径の大きい "C" に入れ込む「United atom model」を内部で自動生成できるようにする。United atom model 化された "C" のパラメータについては、AMBER OPLSなどを参考にする。

具体的な計算の仕方：

- (1) -CH, -CH₂, -CH₃ を通常の C よりやや半径の大きい C' に変換する。この際、水素原子の電荷を -C' の電荷に足しこむ。
- (2) ドッキングおよびリガンド座標の最適化後に -C' に水素原子を付加し、-CH, -CH₂, -CH₃ を復元する。各 C' が -CH, -CH₂, -CH₃ のいずれに該当するかは -C' の電荷から判断する。水素原子付加の際、-CH, -CH₂ は方向が確定するが -CH₃ は確定しないため、二面角が 0°、120°、240° となるように水素原子を付加する。

4. スコア関数系

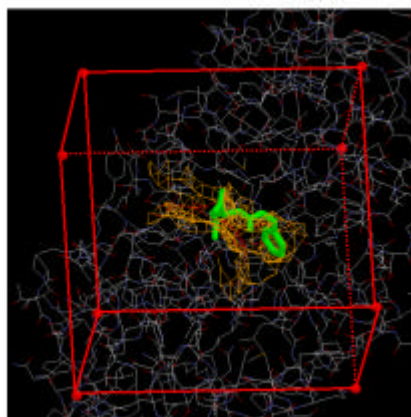
4.1 概要

試行・予測されたリガンド分子配置が妥当かどうか、エネルギーに相当するスコアを計算して判定する。

(1) Grid ポテンシャルの生成

Grid は、mesh 点に対し、x-y-z 方向にそれぞれ 8 マージンを取るような直方体で作成しているが、このマージンは可変として外部入力で与えられるようにする。サイズは x-y-z 軸それぞれ $60 \times 60 \times 60$ で、mesh 幅は約 0.4 であるが、これも可変とし外部入力で与えられるようにする。Grid サイズは 60 から変更できるようにするが、これはコンパイル時に指定する形式でよい。

vdW 相互作用、ASA、クーロン相互作用の情報を Grid に載せるため、vdW に対し約 10~20 種類の相互作用のタイプがあり、さらに同様に ASA のタイプが原子タイプごとにあり、静電場、H-donor、H-acceptor の力場が必要なので、実数単精度で「 $60 \times 60 \times 60 \times (\text{原子タイプ数} \times 2 + \text{正電場} 1 \text{個} + \text{H-donor} \cdot \text{H-acceptor} 2 \text{種類})$ 」の配列を用意する。配置の発生が mesh 幅に対して大まかなので、補間はラグランジュの 1 次補間式ないし、スプライン補間で行う。下図にリガンドの位置 (緑)、mesh 点 (黄色)、Grid の大きさ (赤) を示す。

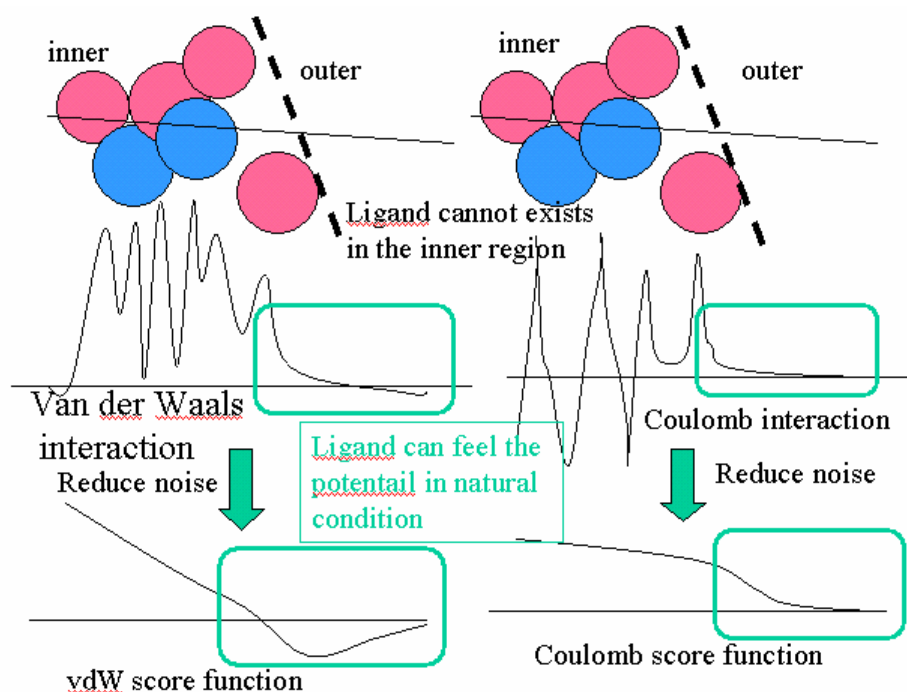


生成した Grid ポテンシャルは、ファイル出力 (アスキー形式、バイナリー形式) できるようにする。また、プログラムは、レセプターの PDB が入力されなくても、ファイルに保存された Grid ポテンシャルの情報で、リスタートできるようにする

(2) スコア関数の作成

スコア(力場)は、科学ではないので、良い結果を与えるのが良い関数形・スコアパラメータである。docking は、多極小での最小化問題なのでポテンシャル面(スコア面)は滑らかなのが良い。リガンド分子の座標は自然界ではたんに質構造と重ならない。したがって、リガンド分子が蛋白質内部に位置した場合に生じるすべての極小は人工的なものであって、重要ではない(下図)。vdW 力では原子の重なる場合に無限大の値が、クーロン力では正負の無限大の値に達する。これらの値に意味はない。

蛋白質外部のポテンシャルがやや正しく構成されれば良く、蛋白質内部のポテンシャルは、リガンドが内部に位置した場合速やかに最寄の外部へ移動できるようなポテンシャルを構成すればよい。

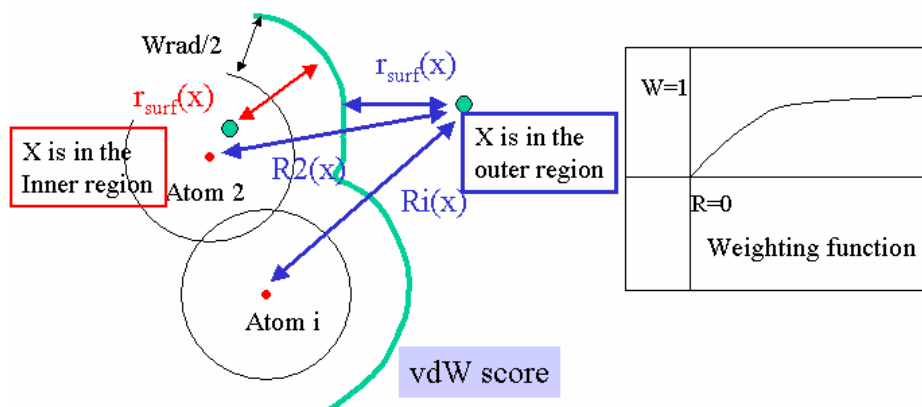


(3) 最後は、データベースとの結果の比較により、扱いやすい関数系と、良い結果を与えるパラメータを選択する。スコア関数は次の4種類を用意する。

- ・ vdW
- ・ ASA
- ・ クーロン
- ・ 水素結合

水素結合はクーロンに含まれるが、静電場・電荷の扱いは困難なので補助的に水素結合を追加(ガウス型)する。

4.2 van der Waals score



$$score(x) = w(r_{surf}(x)) \sum_i \left\{ \frac{A}{R_i(x)^{12}} - \frac{B}{R_i(x)^6} \right\} : x \in outer$$

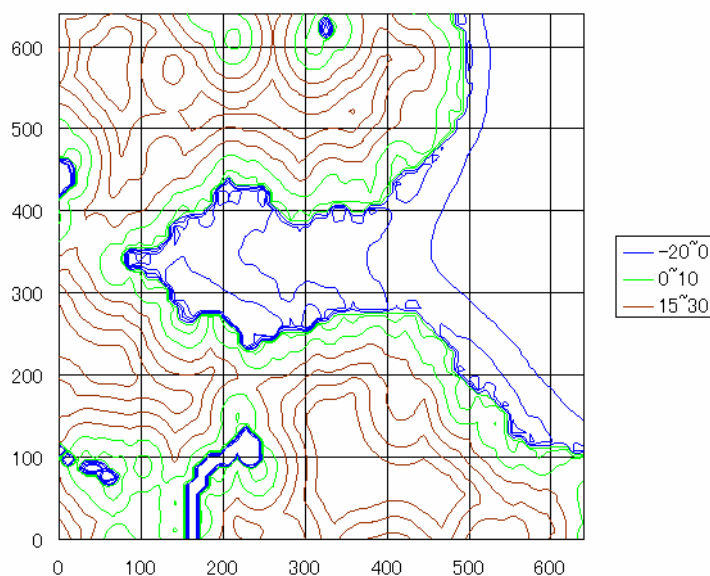
$$score(x) = 10.0 r_{surf}(x) : x \in inner$$

$$w(r) = 1 - e^{-Cr}, C = 2.0$$

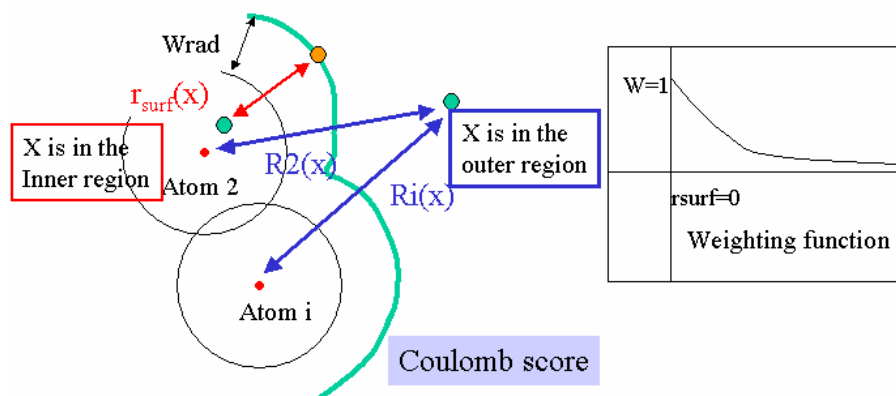
where w is a weight function & $q = -100.0$

プローブ半径 $Wrad$ を使い、レセプター側の outer/inner 領域の界面を通常の accessible surface で定義する。レセプター側の outer 領域では、通常の van der Waals 力の計算を行う。現状では、AMBER タイプの力場計算を行う。重み関数 w は、界面までの最短距離 r_{surf} の関数とし、力場を outer/inner 領域の界面で連続に接続する。

原子タイプは、AMBER のものをそのまま用いると数が多いので、半径・パラメータの類似のものをいくつか集約して簡素化する。



4.3 Coulomb score



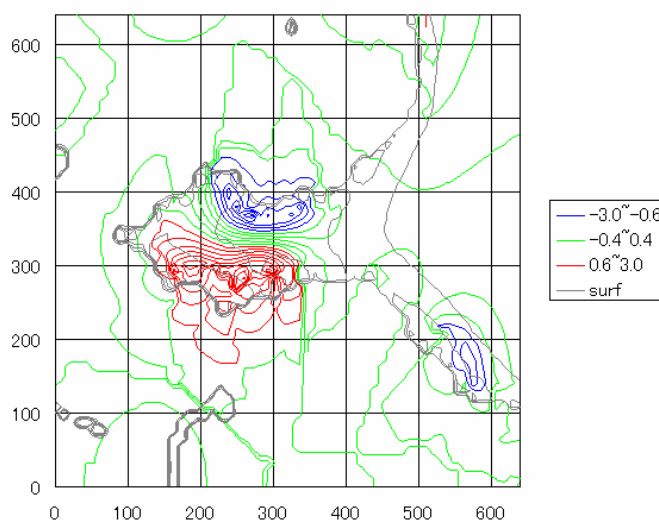
$$score(x) = \sum_i \frac{332q_i}{4R(x)_i^2} : x \in outer$$

$$score(x) = w(r_{ASA}(x)) \sum_i \frac{332q_i}{4R(ASA)_i^2} : x \in inner$$

$$w(R) = \frac{1}{1+R}$$

where q_i is charge of i -th atom of protein.

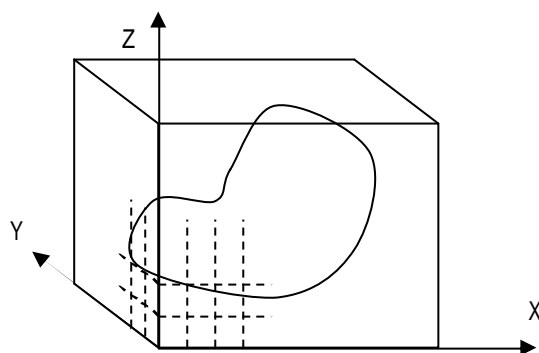
プローブ半径 W_{rad} を使い、レセプター側の outer/inner 領域の界面を通常の accessible surface で定義する。レセプター側の outer 領域では、通常の電場計算を行う。上式は、距離依存誘電率及び定数値の誘電率での電場計算の式であるが、Poisson-Boltzman の連続体近似法での電場計算でも応用できるようにする。重み関数 w は、界面までの最短距離 r_{ASA} の関数とし、電場を outer/inner 領域の界面で連続に接続する。上式は、AMBER タイプの距離依存誘電率の式であるが、より一般的なものを用いる。



Poisson-Boltzman の連続体近似法による電荷計算

距離依存誘電率及び定数値の誘電率での電場計算のほかに、Poisson-Boltzman の連続体近似法での電場計算も応用できるようにする。

(1) 対象分子が入る直方体を定義し、メッシュを生成する。



(2) 各メッシュ点について、分子の内部、外部、境界であるか判定する。

(3) 静電場の計算をSOR法 (Successive Over-relaxation method) で行う。

格子点 i, j, k の電位 $f_{i,j,k}$ を反復解法で以下のように解く。

$$f_{i,j,k}^n = f_{i,j,k}^{n-1} + w \left(\frac{f_{i-1,j,k}^{n-1} + f_{i+1,j,k}^{n-1} + f_{i,j-1,k}^{n-1} + f_{i,j+1,k}^{n-1} + f_{i,j,k-1}^{n-1} + f_{i,j,k+1}^{n-1}}{6} - f_{i,j,k}^{n-1} \right)$$

加速係数 : $w > 1.0$

(4) 各メッシュの情報と、ドッキングスコア計算用の各グリッドとの対応をとる。

それぞれはズレているので、補間して対応させる。

4.4 ASA のスコア計算

ASA の計算は、プローブ球をレセプター表面に転がしたときの ASA のスコアを計算する方法 (Ricimond の方法) とプローブ球とレセプター原子の 2 体ポテンシャルである距離の関数でスコアを計算する方法の 2 つを備える。前者には解析的な計算方法を、後者には関数系は線形の区分多項式を用いる。2 体ポテンシャルで定義する場合は、

$$E_{ASA} = \begin{cases} \sum_{a,b} f(A + CR_{ab}) : R_{ab} \leq \mathbf{s}_a + \mathbf{s}_b + 2r_{prob} \\ 0 : R_{ab} > \mathbf{s}_a + \mathbf{s}_b + 2r_{prob} \end{cases}, \quad \text{Eq. 14}$$

ここで $A = -4\pi\{(\sigma_a + r_{prob})^2 + (\sigma_b + r_{prob})^2\}$ 、 $C = -A/(\sigma_a + \sigma_b + 2r_{prob})$ 。 σ_a と r_{prob} は、a 番目の原子の vdW 半径とプローブ半径である。

4.5 水素結合のスコア計算

レセプターの表面に存在する H-donor と H-acceptor と、リガンド側の H-acceptor と H-donor の引力項を計算する。等方的なガウス関数型ポテンシャルと水素結合の距離と角度を考慮して数え上げる関数形の 2 つを用いる。等方的なガウス関数型ポテンシャルは、

$$score = C \exp(-Dr)$$

ここで C、D は定数で、r はレセプターの表面に存在する H-donor (H-acceptor) と、リガンド側の H-acceptor (H-donor) の間の距離である。

H-donor の作る力場と H-acceptor の作る力場は別に作成し、H-donor とその近傍に存在する H-acceptor の作用が相殺しないようにする。水素結合の距離と角度を考慮して数え上げる関数形は、

$$E_{H-bond} = \sum_{a,b} c \times \exp^{-((R_{ab} - d_{OH})/r_1)^2 - ((\mathbf{q} - \mathbf{q}_0)/r_2)^2}$$

R_{ab} と θ は、a-番目の H ドナーと b-番目のアクセプターの間の距離と角度である。c、 r_1 、 d_{OH} 、 r_2 と θ_0 は定数であり、それぞれ -3.0 kcal/mol, 3.0 Å, 1.2 Å, 40.0 度, と 180.0 度に設定される。

4.6 Grid ポテンシャルのスムージング

4-1 から 4-4 で生成した Grid ポテンシャルは、数値的にスムージングをかける。スムージングの回数は、外部入力で指定できるようにする。

$$V_{a,b,g} = \frac{n \cdot v_{a,b,g} + \sum_{a'=a-1,a+1,b'=b-1,b+1,g'=g-1,g+1} v_{a',b',g'}}{n+6}$$

ここで $v_{\alpha,\beta,\gamma}$ は、グリッド点 (α , β , γ) でのスコアの値である。

5. スコアの計算

【ドッキングの総合スコア】

ラフなドッキングでは、複数配座の rigid docking であるので、Grid 場で計算されるスコアの総計をその分子配座の総合スコアとする。各項には、係数をかけ、重みを調整できるようにする。g は、0.01/kcal/mol である。

$$S_{raw} = g \cdot (c_{vdW} E_{vdW} + c_{elec} E_{elec} + c_{H-bond} E_{H-bond} + c_{ASA} E_{ASA})$$

ここで E_{vdW} , E_{elec} , E_{H-bond} , E_{ASA} は、それぞれ vdW 相互作用、静電相互作用、水素結合、ASA 項であり、 c_{vdW} , c_{elec} , c_{H-bond} , c_{ASA} は、それらの重みである。デフォルトでは、全て 1.0 に設定されている。

また、蛋白質 - 低分子結合自由エネルギーの見積もりを目的とした「G エネルギー」および、in silico スクリーニングにおけるヒット率を最大にすることを目的とした「hit-optimized エネルギー」の計算を行う。

【Local minimum search の総合スコア】

Local minimum search では、リガンド分子内のエネルギーと Grid ポテンシャルから計算されるスコアを別に扱い、

$$\text{総合スコア} = \text{Grid ポテンシャルのスコア} + (\text{係数}) \times \text{リガンド分子内のエネルギー}$$

とする。

係数は、外部入力で与え、調整できるようにする。

6. 結果の表示

【クラスタリング】

各分子の分子配座を総合スコアの良い順番にならべ、予測される複合体構造を簡単にクラスタリングする。

- (1) 最もスコアの良い座標から一定の RMSD 内にある結果を削除する。
- (2) 残った構造の中で一番スコアの良いものを選び、その構造から一定の RMSD 内にある結果を削除する。
- (3) この操作を繰り返すことで簡単なクラスタリングを行う。

【スコア表示】

各予測複合体のスコアとして、次の情報を表示する。

- (1) 総合スコア
- (2) G スコア
- (3) hit-optimized スコア
- (4) MTS スコア
- (5) 回転可能原子数
- (6) 各成分 (vdW、ASA、静電場、H 結合、分子内エネルギー)
- (7) 標的原子からの距離

ドッキングすると予測されるポケット内に指定した原子 (標的) からの距離など。

- (8) 活性スコア

上記の項に係数をかけて合計したものを活性スコアとし、その値をあわせて表示する。係数は外部入力で与え、調整できるようにする。また、GB/SA 法などによるエネルギー見積り等にも対応できるようにする。

- (9) RMSD

各リガンドに対しレファレンス座標が与えられているときは、予測した構造とレファレンス構造との RMSD を計算して表示する。

7. 配座の生成方法

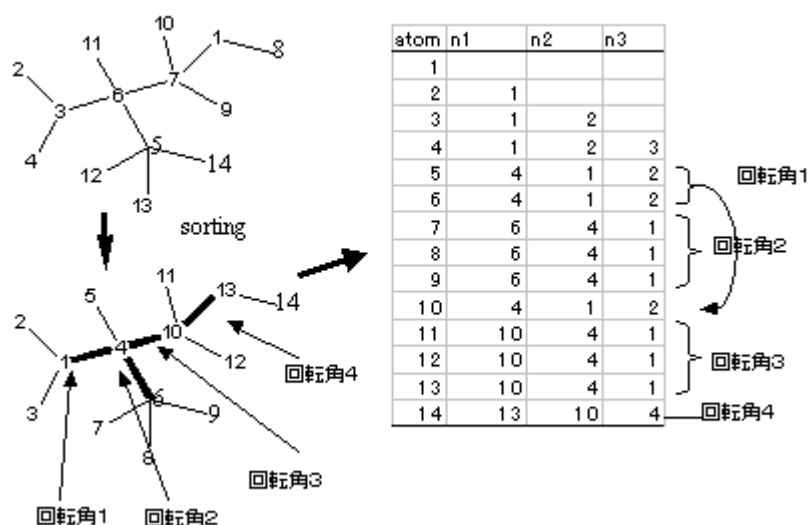
random search の手法を用いて配座を生成する。

与えられた分子のグラフを、tree graph に展開し、tree graph 上の各原子をポーランド記法に準じて原子を訪問するように z-matrix を構成する。回転可能な結合を検出し、ランダムな結合の回転を行う。

7.1 Z-matrix の生成

分子の conformer の生成には、Z-matrix を用いるのが便利である。そこで、グラフ上で化学結合のある原子を辿ることができるように（任意の n （全原子数）に対し、原子 1 ~ 原子 n までのなす分子グラフが常に連結であるように）原子の番号を並べ替えておく。

- (1) 原子 i に対し、結合する原子 $n1 \sim ni$ のうち、原子 $n1 \sim ni$ の先に結合している原子数の一番少ないものから順に訪問する。
- (2) 訪問先がない場合、原子 i に結合する既に訪問した原子へ戻る。



- (3) 原子を並べた後、Z-matrix を以下のフレームに従い構成する。

$n0 \ n1 \ n2 \ n3$: $n1$ は、 $n0$ に連結する一番番号の若い原子
 : $n2$ は、 $n1$ に連結する一番番号の若い原子
 : $n3$ は、 $n2$ に連結する一番番号の若い原子

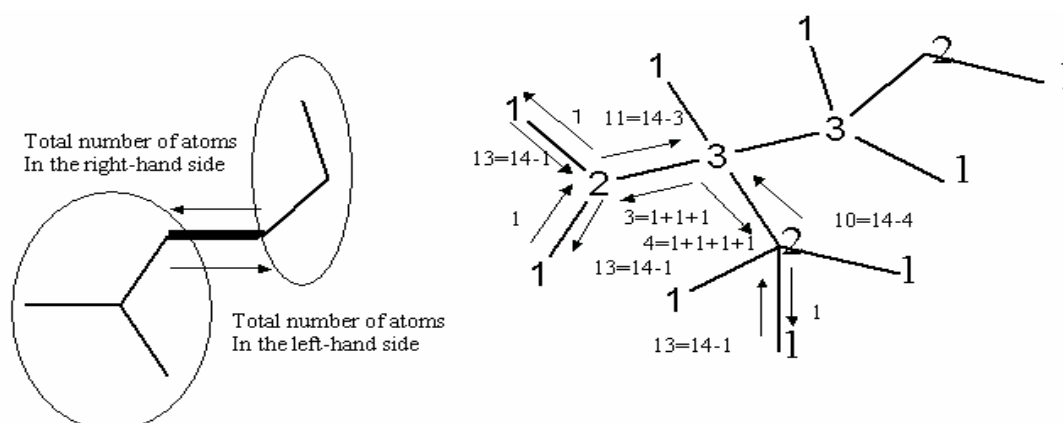
こうして構成した Z-matrix では、 $n_1-n_2-n_3$ が共通の原子に対し、 n_0-n_1 の結合が回転可能ならば、共通の回転角 を 2 面角に加算することで conformer を生成できる。

原子		距離		角度		2 面角
m1	n1	D1	n2	A1	n3	T1
m2	n1	D2	n2	A2	n3	T2
m3	n1	D3	n2	A3	n3	T3
m1	n1	D1	n2	A1	n3	T1+
m2	n1	D2	n2	A2	n3	T2+
m3	n1	D3	n2	A3	n3	T3+

処理方式を「全 Z-matrix 生成後に全原子の座標生成」から「個々の 2 面角を 1 つずつ生成し、少しずつ原子を作成」とすることで配座生成処理の高速化を図る。

具体的な計算の仕方：

- (1) tree graph であれば、必ず 1 本の結合を境に、右と左に graph を分割できる。任意の 1 本の結合に対し、右側に結合する原子数と左側に結合する原子数を計算する。
(ここで、右側原子数 + 左側原子数 = 全原子数)
- (2) 分子グラフに対し、結合が 1 本しかない原子に対し右・左側原子数を計算し、この原子を graph から除く (図で「1」の原子)。
- (3) 新しく構成された graph について同様の操作を行う (図で「2」の原子)。
この操作を原子が削れなくなるまで繰り返す。
- (4) 全ての結合に対して計算した後、任意の原子から出発して、原子数の少ない方向へ結合を辿っていけば、ポーランド記法式に原子を並べ替えることができる。



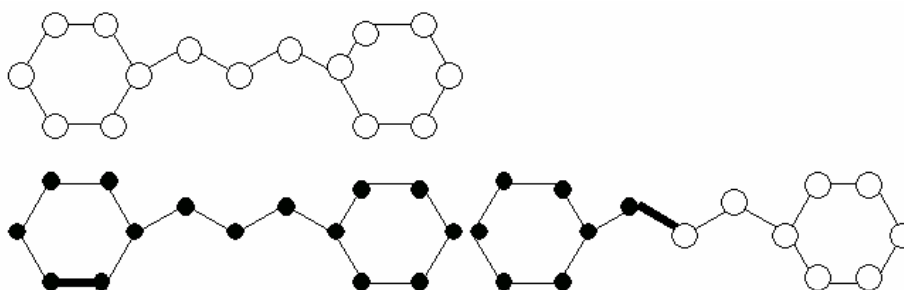
また、入出力ファイルで原子の並びが異なると不便な場合がある。従って、元の原子の順番に戻して出力することができるようにする。原子の順番の並べ替え・元に戻すは、入力で切り替えられるようにする。

7.2 環の検出と開環

上記アルゴリズム (7.1 節) は、環の存在する場合にたどれない原子が残る (環を構成する原子は、任意の 2 原子間に異なる 2 つの経路が存在するため、上記規則で原子を消去できない)。そこで、graph の連結性を保ちながら仮想的に結合を切断し、開環する。

方法：

全原子を全て「white atom」とおく。任意の結合 (太線) を切断した graph を作成し、任意の原子 1 個を選び、これを「black atom」とおく。black atom に結合する全ての原子を black atom にする。この操作を繰り返し、全原子が black atom になれば、最初に切断した結合は、グラフの連結性を保つ切断である。white atom が graph 中に残れば、graph は連結でない。



任意の分子 graph に対し、以下のように (環の配座探索を含まない) conformer を生成する。

(1) 「端原子の刈り込み」 原子が残っていれば「開環」 「端原子の刈り込み」
を繰り返し、右・左側原子数の表を作成する。

(2) ポーランド記法式に原子を並べ替え、Z-matrix を構成する。

(3) 回転可能な固有な結合を検出し、二面角を回転させる。

回転可能な結合とは、環に含まれない結合かつ、一重結合かつ、N を含む場合に N の結合の数が 3 で隣接する原子の結合の数が 3 でない場合、と定義する。

(4) Z-matrix からデカルト座標を計算する。

7.3 複数配座の大量生成

複数配座を短時間に大量に生成する必要があるため、生成した結果の同一性の判断、分子内の原子衝突の判断を高速で行う必要がある。既に生成した配座を記録するなどの工夫を行う。また同じプログラムが再度呼び出されたとき、乱数系列を変えて異なる配座を生成するなどできるようにする。

8. リガンド分子のトポロジー自動生成

リガンド分子が mol2 形式などで与えられていてトポロジーファイルが無い場合、トポロジーファイルを自動生成する。

- (1) 入力となったリガンドの初期構造を安定なものとして、Bond/Angle の平衡距離・角度を計算し、力の定数にはデフォルトで平均的な値を代入しておく。
- (2) 二面角パラメータは、その結合の種類に応じて、標準的なものを代入する。
- (3) 1-4 相互作用は、標準より弱い係数を代入できるようにする。
- (4) 1-5 相互作用は通常の AMBER タイプの計算を行うようにする。
- (5) 原子タイプの割り当ては、AMBER に準拠するが、やや簡素化したものを用いる。

9. 入出力形式

(1) 制御ファイル

myPresto cosgene に準拠した形式とする。

【制御ファイル形式】

```

EXE> INPUT
      TOPOLOGY=  FORM      NAMETO=   Pro.tpl
      COORDINA=  PDB      NAMECO=   Pro.pdb
; Pro.tpl の無い時は、PDB の電荷のカラムから電荷の情報を得る。または mol2 形式。
      LIGANDCO=  PDBX     NAMECO=   Lig.pdbx
      LIGANDCO=  PDBX     NAMERE=   Lig_reference.pdbx
;
; LIGANDCO=  MOL2     NAMECO=   Lig.mol2
;
; LIGANDCO=  MOL2     NAMERE=   Lig_reference.mol2
;
; LIGANDCO=  LIST     NAMELI=   Lig.list
      POINTCO=   PDB      NAMEPO=   point.pdb ; ポケット領域を示すプローブ点
      QUIT

;
; gird generation and Hash table generation
;
EXE> GRID
      RESTART=  YES/NO
      GRIDFORM= ASCII/BINARY
      GRIDNAME= grid.file ; Gridポテンシャルの出力ファイル指定
      PROBDIST= 4.0 ; プローブ点から PROBDIST 内のレセプター原子を対象にする。
      RADVDW= 1.2 ; vDW ポテンシャルの内外領域の界面の設定
      RADELE= 1.2 ; クーロンポテンシャルの内外領域の界面の設定
      RADMESH= 1.6 ; mesh 点の生成のプローブ半径

      DISTMIN= 3.0 ; Hash 表作成時の 3 角形の辺の長さの下限
      DISTMAX= 12.0 ; Hash 表作成時の 3 角形の辺の長さの上限
      DISTSIZ= 0.5 ; Hash 表作成時の 3 角形の辺の長さの刻み

      MARGIN= 8.0 ; Grid 生成時の、探索ポケット周辺にとるマージン
      HYDROHOBICITY= 3 ; Hash 表作成時、疎水性の原子を多く含む 3 角形を排除する。

      SMOOTH= 1 ; Grid ポテンシャルのスモーキングの回数

      RANDOM= 654321
      CUTMET=  RESA      CUTLEN= 16.000
      DIEFUN=  CONS     DIEVAL= 1.000
      QUIT

```

(つづく)

(つづき)

```

;
; comformer generation
;
EXE> CONF
RANDOM=      654321
MAXCONF=    100000 ; 試行する配座の最大数
NOCONF=     100    ; 生成する配座数
ATOMSORT=   YES/NO ; 出力ファイルの原子の並び
CHARGE=     READ/AUTO ; 電荷計算
DMAP=       0.8    ; 衝突判定
PHASE=      6      ; 結合を 3 6 0 /6=60度ごとに回転する。

;
; docking for global search
;
EXE> DOCK
METHOD=     FLEX/RIGID
NOGENERATION= 5 ; 絞込みの回数
LOGFOR=     DETA NOCONF= 30 ; 1 リガンド当たり出力する数
MATCHING=   5   ; 3 角形の原子タイプの適合度
DISTMIN=    6.0 ; Hash 表と比較する 3 原子の作る辺の長さの下限。
DISTMAX=    9.0 ; Hash 表と比較する 3 原子の作る辺の長さの上限。
DISTMIN2=   3.0 ; Hash 表と比較する 3 原子の作る辺の長さの下限。
DISTMAX2=   12.0 ; Hash 表と比較する 3 原子の作る辺の長さの上限。
DISTMIN~ DISTMIN2、DISTMAX~ DISTMAX2 の範囲で、試行回数を見積もって、
リガンドごとに試行回数がある値になるように、上限・下限を調整する。

SCOREWEIGHT_VDW= 1.0 ; 総合スコア計算時の各項の重み
SCOREWEIGHT_ASA= 1.0
SCOREWEIGHT_ELE= 1.0
SCOREWEIGHT_HDA= 1.0

CENTER=     2340 ;
CENTERX=    0.0
CENTERY=    0.0
CENTERZ=    0.0 ; 標的とする原子座標
RADCHECK=   6.0 ; リガンド分子ドッキングの後、
                  標的原子 (座標) から RADCHECK 内に存在する
                  リガンドの原子の数を数える。

```

(つづく)

(つづき)

```
;  
; minimization for local search  
;  
EXE> MINI  
  UPDATE=    1000  
  METHOD=    STEEP/CONJ  
  CPUTIM=    360000.0  
  LOOPLI=   500      UPDATE=    20  
  MONITO=   10      CONVGR=    0.1D0  
  CUTMET=   RESA    CUTLEN=    12.0D0  
  DIEFUN=   DIST    DIEVAL=    4.0D0  
  LOGFOR=   DETA/NO  
  
  ENERGYWEIGHT= 0.0 ; 分子内エネルギーとGridポテンシャルによる  
                      スコアを加算してスコアを作る場合の  
                      分子内エネルギーにかける係数。  
  
  QUIT  
  
EXE> OUTPUT  
  NOCONF=5 ; 1 リガンド当たり出力する構造数  
  
  LIGANDCO= PDBX    NAMECO=    Lig.pdbx  
  LIGANDCO= PDBX    NAMERE=    Lig_reference.pdbx  
;  LIGANDCO= MOL2    NAMECO=    Lig.mol2  
;  LIGANDCO= MOL2    NAMERE=    Lig_reference.mol2  
;  LIGANDCO= LIST    NAMELI=    Lig.list  
  
  QUIT  
  
EXE> END
```

(2) リガンド指定ファイル

リガンド指定ファイルは、1~100000 分子を扱う必要があるため、1 ファイルに複数の分子を記載する。座標も含めて 1 ファイルとする場合と、ファイル名のリストを指定する場合の 2 種類を準備する。また、これらファイルは結果の出力にも用いるため、スコアなどの情報を同時に記載できるようにする。

【PDBX 形式】

PDB ファイル形式と myPresto topology file を、タグで切って並べたもの。

```
<MOLECULE name=ligand1>
  <PDB>
    PDB file for molecule1
  </PDB>
  <TPL>
    topology file for molecule1
  </TPL>
</MOLECULE>
<MOLECULE name=ligand2>
  <PDB>
    PDB file for molecule2
  </PDB>
  <TPL>
    topology file for molecule2
  </TPL>
</MOLECULE>
:
```


【mol2 ファイル形式】

mol2 ファイル形式で、マルチファイル形式のものを許容する。原子情報および結合情報の指定は以下のように行う。

原子情報指定

mol2 ファイルによる原子座標の指定は以下の行で構成される。

```
@<TRIPOS>ATOM
[ 原子情報行 ] × 原子数
```

原子情報行：

```
原子番号  原子名  x-座標  y-座標  z-座標  原子タイプ  残基名  残基番号  電荷
```

構造情報指定

mol2 ファイルによる結合情報の指定は以下の行で構成される。

```
@<TRIPOS>BOND
[ 結合情報行 ] × 結合数
```

結合情報行：

```
結合番号  当該原子番号  対象原子番号  結合タイプ
```

各結合の「結合タイプ」は、以下のいずれかで指定する。

1 : single bond	2 : double bond	3 : triple bond
am : amide bond	ar : aromatic bond	du : dummy bond
un : unknown	nc : not connected	

```
@<TRIPOS>ATOM
  1 H      -0.7330  56.1710  -6.1930 H      0 <1>      0.0646
  2 C      -1.4680  56.3850  -6.9670 C.2    0 <1>      0.0033
  3 C      -1.5020  55.5410  -8.2050 C.3    0 <1>      0.1196
                                     ( 中略 )

@<TRIPOS>BOND
  1   1   2   1
  2   2   3   1
  3   2  27   2
                                     ( 以下略 )
```

【Lig.list 形式】

1 分子の mol2/PDBX を含むファイル名のリストである。

- ・ 1 カラム目：docking に用いる分子構造。
- ・ 2 カラム目：1 カラム目の結果に対する reference 構造。(2 カラム目指定時のみ)

```
"file name 1(molecule1)" "reference file name1(molecule1)"
"file name 2(molecule2)" "reference file name2(molecule2)"
"file name 3(molecule3)" "reference file name3(molecule3)"
"file name 4(molecule4)" "reference file name4(molecule4)"
:
```

(3) スコア表示

「6. 結果の表示」中の「【スコア表示】」に記載されている (1) ~ (9) の項目を標準出力に出力する。出力形式は以下の通りである。

```
INFORMATION> TOP SCORE RANKING          3
      SCORE(/100) dG      HIT      MTS      rotNum      ASA      ELE      HYD      VDW      SURFACE      RMSD
@ 1  -4.10 -11.55 -270.56 -698.34      0  -392.49  -3.84  -8.25  -5.01  86.95  0.88
@ 2  -3.77 -10.72 -248.63 -640.21      0  -358.00  -3.99 -10.20  -5.17  88.04  0.53
@ 3  -3.67 -10.45 -241.42 -622.14      0  -347.69  -3.75 -10.43  -5.22  87.46  0.51
```

10. 制約事項

実現の際には、以下の条件を満たすこと。

- (1) ファイル入出力を極力減らし、中間ファイルの類のファイル I/O を行わないこと。
- (2) 1つの作業ディレクトリ内で、同じ分子を対象とする複数の計算 JOB を同時平行して実行できるようにする（出力名が異なるものは同時実行可能とする）。
- (3) 実数単精度が倍精度より早ければ、精度より速度を優先し単精度を採用すること。
- (4) 移植により、通常の Windows PC のような小型計算機でも動作可能とすること。

以上