Vol.2020-BIO-61 No.4
2020/3/12

共通な部分構造の再利用による高速なタンパク質リガンドドッキング手法の開発

久保田 陸人[1]　柳澤 渓甫[2]　吉川 寧[1]　大上 雅史[1]　秋山 泰[1,a)]

概要：創薬研究では数百万〜数億件というきわめて多数の化合物群を扱う必要があり，ドッキングシミュレーションの高速化が求められている．本報告では，化合物の多くが共通な部分構造を持つことに着目し，部分構造に対する計算結果を再利用することにより 1 つの標的タンパク質に対して大量の化合物を高速に評価することに特化したドッキングツールを開発した．部分構造の計算結果から化合物全体のスコアを効率的に近似計算する手法を提案し，DUD-E Diverse Subset を用いた実験において，よく用いられているドッキングツールの一つである AutoDock Vina に比べ同等に近い精度を保った上で約 8.4 倍の高速化を確認した．

キーワード：バーチャルスクリーニング，タンパク質リガンドドッキング，フラグメント分割

Development of an efficient protein-ligand docking method by reuse of fragments

Rikuto Kubota[1]　Keisuke Yanagisawa[2]　Yasushi Yoshikawa[1]　Masahito Ohue[1]
Yutaka Akiyama[1,a)]

Abstract: In the drug discovery, it is necessary to explore a large compound database composed of several to hundred millions of compounds, thus the acceleration of docking calculation is greatly demanded. Reuse of calculation results is one of the feasible ways to accelerate. In this study, we focused on the fact that many of the compounds have common substructures, called fragments, and developed a fast docking tool specialized for the evaluation of a large number of compounds by reusing the docking calculation results of fragments. We propose a docking method to evaluate compounds efficiently by the calculation results of fragments, and we confirmed that the proposed docking method was approximately 8.4 times faster than AutoDock Vina keeping almost same accuracy.

Keywords: virtual screening, protein-ligand docking, compound fragmentation

1. 序論

創薬研究の初期における候補化合物の絞り込み（スクリーニング）においては，きわめて多数の化合物を扱わなくてはならない．購入可能な化合物の立体構造データベースを公開している ZINC [1] における登録化合物件数を例に挙げると，その数は 1 億 2 千万件にも及ぶ．そこで，初期段階でコンピュータを用い薬剤の候補となる化合物をふるいにかける手法（バーチャルスクリーニング）による創薬コストおよび時間の削減が近年試みられている [2]．その中で，特に標的タンパク質と薬剤候補化合物それぞれの立体構造情報に基づく手法を Structure-Based Virtual Screening (SBVS) と呼ぶ．

タンパク質リガンドドッキングシミュレーション（以下，

[1] 東京工業大学 情報理工学院 情報工学系，
Department of Computer Science, School of Computing, Tokyo Institute of Technology
[2] 東京大学 大学院農学生命科学研究科 応用生命工学専攻，
Department of Biotechnology, Graduate School of Agricultural and Life Sciences, The University of Tokyo
[a)] akiyama@c.titech.ac.jp

ドッキングシミュレーション）はタンパク質と化合物との間の物理的なエネルギーを計算することにより複合体構造及び結合の強さを予測する手法であり，SBVS において広く用いられている [3]．一方で，ドッキングシミュレーションでは化合物の平行移動，回転，また化合物内部の回転等も考慮しながら探索を行う必要があり，計算コストがきわめて高いという問題点が存在する．よって，化合物ライブラリに含まれる膨大な数の化合物を評価することを考えると未だドッキングシミュレーションは十分高速であるとは言えず，高速化は強く求められている．

既存のドッキングシミュレーションツールの多くでは 1 つの標的タンパク質と 1 つの薬剤候補化合物とのドッキングシミュレーションを独立に行うが，バーチャルスクリーニングの用途においては 1 つの標的タンパク質に対し化合物データベースに含まれる多数の化合物のドッキングシミュレーションを行うことにより薬剤候補を絞り込むこととなる．このことに着目した手法として，薬剤候補化合物にはお互いに共通な部分構造を持つものが多いことを利用してドッキングシミュレーション以前の薬剤候補化合物の絞り込み（プレスクリーニング）を高速に行う Spresso [4] という手法が存在する．Spresso は部分構造に対するドッキングシミュレーションの結果を多数の化合物に対して再利用することで，大量の化合物の評価を効率的に行う．部分構造の計算結果を再利用することは，プレスクリーニングだけにではなくドッキングシミュレーションそのものにも十分活用可能であると考えられる．

本研究では，薬剤候補化合物にはお互いに共通な部分構造を持つものが多いことに着目し，部分構造に対するドッキング計算結果を効率的に再利用することで，1 つの標的タンパク質に対して大量の化合物を高速に評価することに特化したドッキングシミュレーション手法の開発を行った．

2. 提案手法

2.1 提案手法の概説

本研究では，薬剤候補化合物に互いに共通する部分構造（**フラグメント**）が多く含まれることに着目し，それらに対する計算結果を保存・再利用することにより，1 つの標的タンパク質に対して大量の化合物を評価することに特化したドッキングシミュレーション手法を提案する．提案手法の大まかな流れを**図 1** に示す．

提案手法では，化合物の部分構造に対する計算結果を保存するため，またそれらを効率的に再利用するために以下の 2.1.1 節，2.1.2 節に示す 2 つのアイデアを採用した．

2.1.1 アイデア 1：化合物をフラグメントに分割し，フラグメントのエネルギースコアを計算する

ドッキングシミュレーションでは化合物の並進移動・回転・内部自由度からなるきわめて広い探索空間からより良い解を見つける最適化問題を解く必要がある．その中で探索空間が大きい要因の一つである化合物の内部自由度に対する解決法として，eHiTS [5], FlexX [6] などの構造分割に基づくドッキングシミュレーションツールにおいては，化合物を内部自由度のより少ない**部分構造に分割**し，その**部分構造とタンパク質とのエネルギースコア等を計算し最終的な複合体構造を構成する**，という手法を用いている．

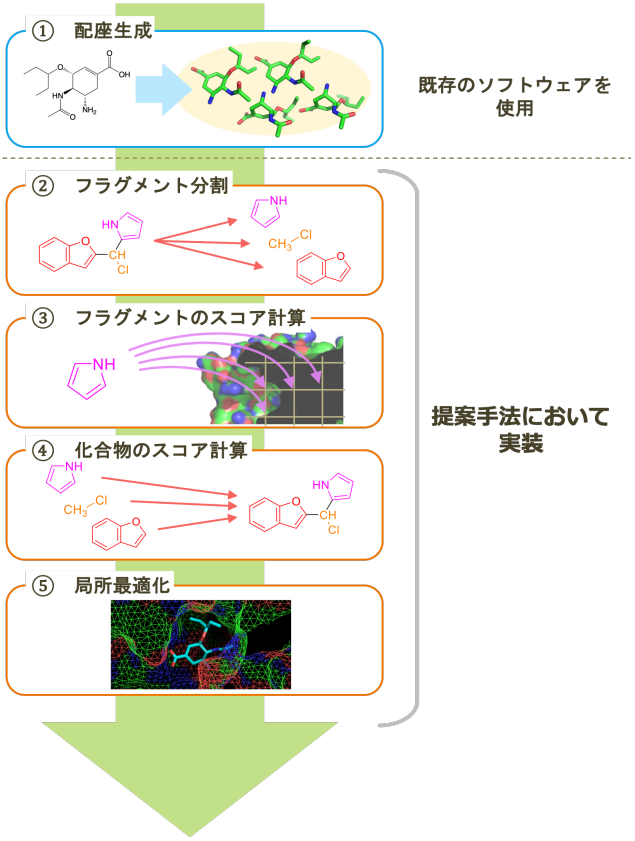

図 1 提案手法の大まかな流れのイメージ図

本研究では，小峰ら [7] の使用した化合物の分割方法を用いて化合物をフラグメントに分割し，フラグメントに対するエネルギースコアを計算する．そして計算したエネルギースコアをメモリに保存し，同じフラグメントを持つような化合物の評価の際にそのデータを再利用することで，計算回数の削減を行う．

2.1.2 アイデア 2：列挙した配座に基いて，フラグメントから化合物の構造を擬似的に組み立て評価する

構造分割に基づくドッキングシミュレーションツールでは，部分構造同士の衝突などが発生しないように最終的な複合体構造を構築する必要がある．しかし，部分構造同士の衝突や距離を考慮すると，厳密な解を求める場合部分構造の数の指数時間かかり，近似的な解を求める場合においても多くの計算時間を要してしまうことが考えられる．

ここで，FRED [8] はあらかじめ化合物のとりうる立体構造（配座）を列挙し，それらを内部自由度のない剛体として扱って探索を行うというドッキングシミュレーション手法を採用している．本研究では，FRED と同様にあらか

じめ配座を列挙し，それらをフラグメント同士の取りうる相対位置の情報と見ることで，フラグメント同士の衝突などを考慮せず高速に最終的な複合体の構造を構成する．

2.2 提案手法の詳説

本研究の提案手法は以下の手順でドッキングシミュレーションを行う．なお，提案手法は並列化はされておらず，全て 1 CPU コアを用いて処理を行う．

(1) 前処理として各化合物に対し配座生成ツールを用いて配座を生成しておく．
(2) 化合物を部分構造（フラグメント）に分割する．
(3) 化合物の各フラグメントに対し，エネルギースコアを網羅的に計算する．このとき，これより前に評価した別の化合物を含め，すでに同じフラグメントに対する計算を行っていた場合はその計算結果を再利用する．
(4) 配座から得られるフラグメントの相対位置情報を用いて元の化合物を復元し，元の化合物に対し擬似的なエネルギースコアを網羅的に計算する．
(5) 擬似的なエネルギースコアが上位であるポーズを複数選択し，正確なエネルギースコアを計算しつつ局所最適化を行う．

2.2.1 配座生成

各化合物に対し，配座生成ツールを使用して取りうる立体構造（配座）を複数生成する．本研究では，配座生成ツールの中で最も性能が良いという報告 [9] のある OMEGA [10] を使用して配座を生成する．OMEGA は OpenEye 社が開発したソフトウェアであり，アカデミックな用途においては無償で使用可能な有償ツールである．

2.2.2 フラグメント分割

小峰ら [7] の使用した化合物の分割方法を用いて，化合物を内部自由度を考慮する必要のないような部分構造（**フラグメント**）に分割する．フラグメントへの分割の具体的なアルゴリズムを以下に示す．

(1) 化合物内部の各結合がそれぞれ回転可能かどうかを判定する．
(2) 回転不可能な結合の両端の 2 原子を同一のフラグメントに含める．
(3) 環構造がある場合は，各環構造に対し環を構成する全ての原子を同一のフラグメントに含める．
(4) 回転可能な結合においても，どちらか片方が 1 原子のみとなる場合は，回転しても構造が変化しないので同一のフラグメントとする．
(5) 1 つの原子のみからなるフラグメントに対し，それがちょうど 1 つのフラグメントと結合している場合にはそれらを同一のフラグメントとする．

2.2.3 フラグメントのスコアを網羅的に計算・保存

各フラグメントに対して，標的タンパク質とフラグメント 1 つとのポケット周辺におけるエネルギースコアを三次元空間内で等間隔（デフォルトでは 0.25 Å 刻み）に計算する．この三次元グリッド状に保存されたフラグメントのエネルギースコアを**フラグメントグリッド**と呼ぶ（**図 2**）．

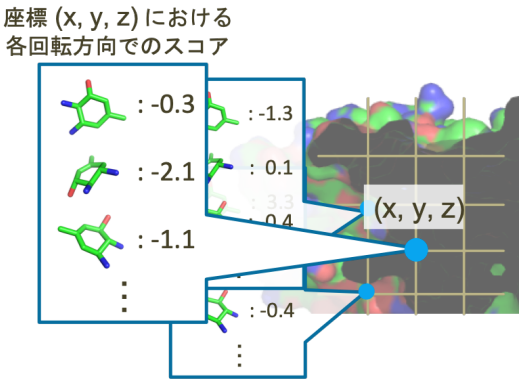

図 2 フラグメントグリッドのイメージ

エネルギースコア関数には AutoDock Vina [11] で用いられているスコア関数を著者が独自に再実装したものを使用している．AutoDock Vina における化合物のエネルギースコアは，化合物内の原子とタンパク質内の原子との全てのペアにおけるエネルギースコアの総和に対し化合物の回転可能結合数にのみ依存する係数を掛けた値で定義される．このスコアは相互作用が存在しないと値は 0 になり，値が小さいほど（負方向に大きいほど）よく相互作用をしていてスコアが良いということを表す．

作成したフラグメントグリッドを再利用するためにはそれらを保存しておく必要がある．しかし，フラグメントの並行移動，回転を考慮しスコアを網羅的に保存すると以下に示す例のようにメモリが大きくなる．よって，数万から数十万種類にもなるフラグメントに対しフラグメントグリッドを同時に保存しておくためにはきわめて膨大なメモリが必要となる．

フラグメントグリッドのメモリサイズの例

一辺 30 Å の立方体グリッド状に 0.25 Å 刻みで，回転角として 60 通り考慮してスコア 1 つあたり 4 byte で保存した場合，フラグメントグリッド 1 つあたりのサイズは $(30/0.25)^3 \times 60 \times 4$ [byte] $\simeq 400$ [MiB] となる．

この問題に対し，本研究では以下のような方法を用いることで効率的な保存，再利用を行うことを考える．

- **各座標において複数ある回転方向のうちエネルギースコアが最良となるもののみ保存**
  上ではフラグメントの回転を考慮し指定した回転方向の通り数分のエネルギースコアを全て保存していたが，**図 3** のように各座標においてフラグメントの回転のうちエネルギースコアが最良となる回転方向のスコアだけ保存し，他の回転方向におけるスコア，及び最

良のスコアを示したのがどの回転方向であるかという情報は保持しない，というようにフラグメントグリッドの定義を変更することを考える．

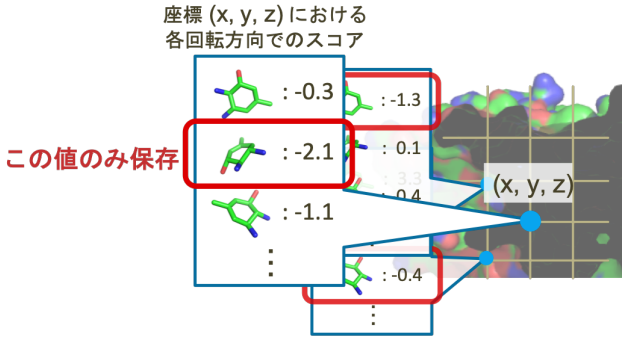

図 3　最良方向のスコアのみ保存するフラグメントグリッドのイメージ

これによりフラグメントグリッドの情報量は下がるが，各座標に格納された値が本来のエネルギースコアより悪い値になってしまうことはなく，この後の局所最適化を行うポーズを「絞り込む」という意味では十分意味をなすものであると考えられる．このデータ削減により，上記のフラグメントグリッドのメモリサイズの例と同様のパラメータでのフラグメントグリッド 1 つあたりのサイズは $(30/0.25)^3 \times 4$ [byte] $\simeq 6.6$ [MiB] になる．

- **オフラインキャッシュ問題として定式化することによるメモリ戦略の最適化**
  上記のデータ削減を行っても，まだ数十万件のフラグメントグリッドを全て保存することは現実的であるとは言えない．この問題に対し，どのフラグメントグリッドを保存しておくべきかという最適なメモリ戦略をオフラインキャッシュ問題に帰着することにより高速に解決するアルゴリズムが提案されている [12]．この手法を用いることで，登場する全てのフラグメントの計算結果を保存するほどのメモリが存在しない場合でも，破棄する計算結果をその都度最適に選択することで効率的にフラグメントの計算結果を再利用することが可能である．本研究では，この手法を利用してフラグメントグリッドのメモリ戦略の最適化を行う．

2.2.4　フラグメントグリッドから化合物のスコアを計算

各化合物に対し，その化合物が持つフラグメントのフラグメントグリッドの値を用いて化合物全体としてのドッキングポーズとスコアを計算する．具体的には，配座の構造から各フラグメントのあるべき相対位置を求め，各フラグメントに対し相対位置情報に合致するような座標を選択し，その座標におけるフラグメントグリッドの値の総和を化合物全体の擬似的なエネルギースコア（以下，グリッドスコアと呼ぶ）とする．上述した化合物全体のグリッドスコアを並行移動・回転に関して網羅的に探索し，複数の位置におけるグリッドスコアを取得する．このとき，ある閾値（デフォルトでは −3.0）よりも良いグリッドスコアが存在しない場合，その化合物についてはドッキングに失敗したとする．

2.2.5　グリッドスコアが上位であるものに対して局所最適化

各化合物のうちグリッドスコアが上位であるような複数のポーズに対し，原子グリッドを用いてより厳密なエネルギースコアを求めつつ局所最適化を行う．局所最適化は山登り法により行い，具体的には化合物の配座を固定させたまま，微小に並進移動させるもしくは回転させるということをランダムに行った際のエネルギースコアを複数通り（デフォルトでは 200 通り）計算し，その中で一番スコアの良い構造を採用する，ということをスコアが改善しなくなるまで行う．そうして局所最適化された各ポーズに対し，エネルギースコアが最も良いものをドッキングポーズとして出力し，そのスコアが化合物のドッキングスコアとなる．

3.　実験

3.1　データセット

本研究では，データセットとして Directory of Useful Decoys, Enhanced (DUD-E) [13] の Diverse Subset（以下，DUD-E Diverse Subset）を用いる．DUD-E Diverse Subset は 8 種類のターゲットからなり，DUD-E に含まれる 102 種類のターゲット全体を代表する部分集合として公式に提供されているデータセットである．

3.2　計算環境

本研究では，計算環境として東京工業大学のスーパーコンピュータである TSUBAME 3.0 の f_node を利用した．f_node 1 基あたりの性能を**表 1** に示す．

表 1　TSUBAME 3.0 f_node の性能

CPU	Intel Xeon E5-2680 v4 2.4GHz (14 cores) × 2
メモリ	256 GiB RAM

3.3　比較対象

本研究では，無償のドッキングシミュレーションツールの中で特に精度の高いツールのひとつである AutoDock Vina [14] を比較手法として用いる．また，参考として有償のドッキングシミュレーションツールである Glide [15] の高速ドッキングモード（HTVS モード，以下では Glide HTVS と表記する）と通常ドッキングモード（SP モード，以下では Glide SP と表記する）の値も併記する．AutoDock Vina は探索空間に関するパラメータのみ後述する方法で決定し，他は全てデフォルトパラメータで実験を行い，Glide は全てデフォルトパラメータで実験を行った．

3.4 化合物に対する前処理及びパラメータの決定

3.4.1 LigPrep による化合物の前処理

DUD-E Diverse Subset の各データセットに含まれる全てのタンパク質及び化合物に対し，LigPrep [16] による前処理を行った．LigPrep は化合物に対し立体異性体，イオン化状態を考慮した構造を生成する商用のツールであり，化合物に対してはデフォルトパラメータで最大で 32 通りの構造異性体及びイオン化状態を生成する．ここで生成された化合物構造は，提案手法，AutoDock Vina 及び Glide で用いる．

3.4.2 データセット内の化合物の分割

今回の実験環境である TSUBAME 3.0 にはジョブの実行時間制限があり，実行時間が 24 時間を超えるジョブは実行することができない．よって，提案手法においては DUD-E Diverse Subset の各データセットに含まれる化合物をそれぞれ 1,000 件単位で分割し，それぞれについて独立にドッキングシミュレーションを行った．提案手法は AutoDock Vina, Glide などの既存手法と異なり化合物群の中に共通なフラグメントに対する計算結果を再利用するため，化合物群を分割してそれぞれについて独立にドッキングシミュレーションを行うことは再利用率の低下に繋がり，提案手法にとって不利な条件での実験となると考えられる．

3.4.3 OMEGA による配座生成

DUD-E Diverse Subset の各データセットに含まれる化合物から LigPrep により生成された全ての構造に対し，OMEGA [10] により化合物の取りうる立体構造（配座）を生成した．1 つの化合物に対し最大 200 個の配座を生成しており，これは OMEGA のデフォルトパラメータに準拠している．ここで生成された配座構造情報を提案手法で用いる．

3.4.4 探索空間の決定

提案手法及び AutoDock Vina では，実行時に探索空間を三次元空間内の直方体として明示的に指定する必要がある．本研究では，探索空間の中心座標及び探索空間の大きさを Feinstein らの手法 [17] を用いて，標的タンパク質の複合体結晶構造のリガンドの位置および大きさに基づいて，探索空間の直方体を決定した．ここで計算した探索空間の中心座標及び探索空間の大きさの値を提案手法及び AutoDock Vina で用いる．Glide にはデフォルトで独自に探索空間を決定する機構があるため，この値は使用しない．

3.5 評価実験

3.5.1 フラグメント分割

まず，今回用いる DUD-E Diverse Subset に含まれる各データセットについて化合物のフラグメントの分割を行い，各データセットにおけるフラグメント種類数を調べた．さらには DUD-E Diverse Subset 全体でのフラグメント分割実験も行い，化合物件数とフラグメント種類数との関係を調べた．各データセット，及び DUD-E Diverse Subset 全体におけるフラグメント総数，種類数，化合物件数との比率を**表 2** に示す．

表 2 DUD-E Diverse Subset の各データセット及び全体におけるフラグメント総数と種類数（括弧内は化合物件数との比率）

データセット	化合物件数	フラグメント総数	フラグメント種類数
AKT1	16,743	97,119 (×5.8)	8,536 (×0.51)
AMPC	2,898	11,848 (×4.1)	2,843 (×0.98)
CP3A4	11,970	69,612 (×5.8)	8,348 (×0.70)
CXCR4	3,446	18,801 (×5.5)	2,848 (×0.83)
GCR	15,258	73,662 (×4.8)	8,378 (×0.55)
HIVPR	36,286	247,065 (×6.8)	11,098 (×0.31)
HIVRT	19,229	88,673 (×4.6)	11,129 (×0.58)
KIF11	6,966	33,601 (×4.8)	5,415 (×0.78)
全体	112,796	640,381 (×5.7)	30,535 (×0.27)

また，3.4.2 節で述べたように実行環境の制約により提案手法は各データセットの化合物を 1,000 件単位で分割して実験を行った．各データセットに対し，1,000 件単位で分割された化合物の各サブセットにおけるフラグメント種類数の平均を**表 3** に示す．

表 3 DUD-E Diverse Subset の各データセットの化合物 1,000 件単位における平均フラグメント種類数

データセット	化合物 1,000 件単位での平均フラグメント種類数
AKT1	1,044.6
AMPC	1,076.2
CP3A4	1,232.9
CXCR4	919.1
GCR	1,158.2
HIVPR	990.5
HIVRT	1,250.5
KIF11	1,154.7
全体	1,093.5

表 2 に示した通り，今回用いる全てのデータセットにおいてフラグメント種類数が化合物全体の件数よりも少なく，共通なフラグメントが多く登場していることが確認された．また，化合物件数の多いデータセットの方が化合物件数に対するフラグメント種類数の比率が小さい傾向があり，DUD-E Diverse Subset 全体におけるフラグメント種類数，及び各データセットを分割した際のフラグメント種類数においても同様の傾向が見られた．

3.5.2 ドッキング精度の評価

続いて，DUD-E Diverse Subset を用いて提案手法，AutoDock Vina, Glide HTVS, Glide SP のドッキングシミュレーションを行い，各手法の予測精度を比較した．各手法の各データセットにおける ROC-AUC の値及び全体に

おける平均を**表 4** に示す．また，DUD-E Diverse Subset の各データセットにおける Enrichment Factor の値の平均を**表 5** に示す．提案手法は ROC-AUC の値が平均で 0.657 であり，AutoDock Vina の 0.644 に比べほぼ同等かやや高めであると言える．その反面，Enrichment Factor の値においては AutoDock Vina に比べ低めである傾向が見られた．

表 4 各手法の DUD-E Diverse Subset の各データセットにおける ROC-AUC の値

データセット	ROC-AUC			
	提案手法	AutoDock Vina	Glide HTVS	Glide SP
AKT1	**0.765**	0.604	0.533	0.564
AMPC	**0.590**	0.566	0.790	0.832
CP3A4	**0.603**	0.587	0.654	0.672
CXCR4	**0.632**	0.573	0.729	0.666
GCR	**0.605**	0.575	0.509	0.617
HIVPR	0.646	**0.742**	0.683	0.818
HIVRT	0.638	**0.666**	0.638	0.769
KIF11	0.779	**0.841**	0.801	0.861
平均	**0.657**	0.644	0.667	0.725

表 5 各手法の DUD-E Diverse Subset 全体における Enrichment Factor の値の平均

	Enrichment Factor			
	1%	2%	5%	10%
提案手法	5.3	4.5	3.3	2.6
AutoDock Vina	**7.8**	**5.5**	**3.6**	**2.7**
Glide HTVS	8.6	6.4	4.2	3.1
Glide SP	16.5	10.7	6.3	4.3

3.5.3 ドッキング速度の評価

続いて，DUD-E Diverse Subset を用いて提案手法，AutoDock Vina, Glide HTVS, Glide SP のドッキングシミュレーションを行い，各手法の 1 化合物あたりの平均実行時間を比較した．各手法の各データセットにおける 1 化合物あたりの平均実行時間及び全体における平均を**表 6** に示す．全てのデータセットにおいて AutoDock Vina を上回る速度を示し，平均で 8.4 倍高速であることが確認された．

表 6 各手法の DUD-E Diverse Subset の各データセットにおける 1 化合物あたりの平均実行時間（提案手法列の括弧内は AutoDock Vina との速度比）

データセット	平均実行時間［CPU 秒］			
	提案手法	AutoDock Vina	Glide HTVS	Glide SP
AKT1	**31.56** (×19.6)	619.87	0.46	12.04
AMPC	**34.51** (×4.9)	168.39	0.57	5.93
CP3A4	**55.31** (×5.5)	304.86	1.23	21.83
CXCR4	**47.83** (×8.4)	404.03	1.82	32.91
GCR	**40.92** (×5.8)	236.56	0.39	8.69
HIVPR	**45.39** (×8.3)	376.09	0.68	17.47
HIVRT	**28.87** (×6.3)	180.75	0.27	5.51
KIF11	**45.44** (×6.1)	276.24	0.66	13.63
平均	**40.77** (×8.4)	341.89	0.63	13.84

4. 考察

4.1 提案手法と AutoDock Vina との違い

提案手法は AutoDock Vina と同じエネルギースコア関数を再実装して用いているため，精度・速度における差異の大部分は探索空間や探索アルゴリズムの違いに依存していると考えられる．探索空間・探索アルゴリズムにおける提案手法と AutoDock Vina との差異を以下に示す．

- **化合物の内部自由度**
  AutoDock Vina は単結合の回転など化合物の内部自由度も考慮した上で探索を行うが，提案手法は内部自由度として配座生成により得られたフラグメントの相対位置情報のみを考慮する．
  よって，この観点では AutoDock Vina の方が提案手法に比べてより細かく探索をすることができると考えられる．
- **物理的な探索空間**
  パラメータで与えられた探索空間に対して，AutoDock Vina では与えられた探索空間に化合物の全ての原子が含まれるような座標及び回転について探索を行う．対して提案手法では，与えられた探索空間に化合物の重心が含まれ，かつ化合物の全ての原子について探索空間からの距離が一定の値（デフォルトでは 5 Å）以下であるような座標及び回転について探索を行う．
  よって，提案手法は AutoDock Vina に比べて少し物理的に広い範囲を探索することとなる．
- **探索アルゴリズム**
  AutoDock Vina では探索アルゴリズムに反復局所探索法を用いているが，提案手法ではフラグメントの計算結果を再利用することにより空間を網羅的に探索する．よって，AutoDock Vina は良い解が見つかりそうな局所的な空間を重点的に探索し，提案手法は最初に空間全体を一定の粒度で探索してから局所的な探索に移るという違いが存在する．

4.2 提案手法の探索アルゴリズムに対する考察

提案手法のドッキング精度を同じエネルギースコア関数を用いている AutoDock Vina と比較することにより，提案手法の探索空間及び探索アルゴリズムについて評価及び考察を行う．ここでは特に，提案手法が比較的高い精度を示した AKT1 と，比較的低い精度を示した HIVPR について考察を行う．

4.2.1 提案手法が高い精度を示した AKT1 データセットにおける比較

表 7 に示す通り，提案手法は AKT1 データセットにおいて AutoDock Vina に比べて高い精度を示した．

表 7 提案手法及び AutoDock Vina の AKT1 における ROC-AUC, Enrichment Factor の値

	ROC-AUC	Enrichment Factor			
		1%	2%	5%	10%
提案手法	**0.765**	**3.8**	**3.2**	**2.5**	**3.3**
AutoDock Vina	0.604	1.4	1.4	1.8	2.1

AKT1 において提案手法でドッキングスコアが最も良い正例化合物であった CHEMBL359864 について，提案手法及び AutoDock Vina によるドッキングポーズを**図 4** に示す．図 4 において，ピンク色の構造が提案手法が出力したドッキングポーズ（スコア：−11.6）であり，黄色の構造が AutoDock Vina が出力したドッキングポーズ（スコア：−3.9）である．ポケットの全体を使ってドッキングができている提案手法に比べ AutoDock Vina ではポケットの一部しか使えておらず，実際に AutoDock Vina における探索空間はポケット全体を覆えていないことが確認できた．

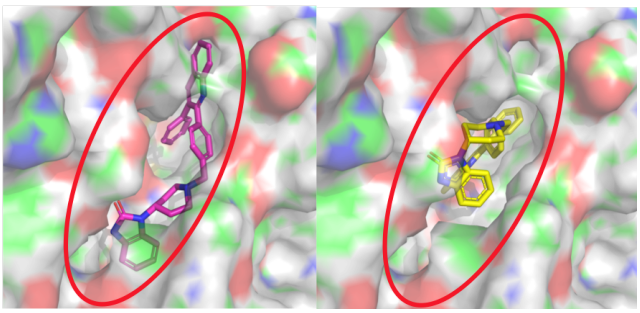

図 4 提案手法及び AutoDock Vina における正例化合物 CHEMBL359864 のドッキングポーズ

提案手法はフラグメントの計算結果を再利用することにより，近似的なスコアながらも AutoDock Vina に比べて物理的に広い範囲を高速に網羅的に探索することができる．AKT1 データセットにおいては結晶リガンドがポケットの奥深くに位置していることもあり，提案手法の物理的な探索空間の広さが有利に働いたと考えられる．また，Glide においても探索空間が小さく推定されていることが確認できたため，Glide HTVS, SP 両モードにおける精度低下の原因も同様であると考えられる．

4.2.2 提案手法が低い精度を示した HIVPR データセットにおける比較

表 8 に示す通り，提案手法は HIVPR データセットにおいて AutoDock Vina に比べて低い精度を示した．

ここで，DUD-E Diverse Subset の各データセットに対し 1 化合物あたりの平均原子数と平均回転可能結合数を調べたところ，HIVPR データセットが平均原子数と平均回転可能結合数どちらにおいても最も多いことが確認できた．よって，提案手法は**化合物の大きさ及び内部自由度が比較的大きい際にドッキング精度が低くなってしまうことが推測される**．その要因としては以下が考察される．

表 8 提案手法及び AutoDock Vina の HIVPR における ROC-AUC, Enrichment Factor の値

	ROC-AUC	Enrichment Factor			
		1%	2%	5%	10%
提案手法	0.646	2.1	2.8	3.0	2.8
AutoDock Vina	**0.742**	**5.2**	**4.4**	**3.7**	**3.2**

- **内部自由度の制限**
  提案手法は化合物の内部自由度を配座生成により得られたものだけに制限し，かつ最終的な局所最適化においても化合物内部の構造を変化させることは行っていない．よって，本来化合物の内部自由度により大きな部分構造をより良い位置に配置できたところを，化合物内部の構造を変化させる最適化を行っていないために相応の高いドッキングスコアを出せず，結果として内部自由度の大きい正例化合物のドッキングスコアがあまり高くならないという可能性が考えられる．
  この点に関しては，**局所最適化において化合物の内部自由度を考慮して探索を行うことで精度が改善する可能性があると考えられる**．
- **回転角のサンプリング数**
  提案手法は探索空間全体を網羅的に探索するが，探索の粒度（並進移動の間隔，回転角のサンプリング数）はその座標でのスコアや化合物の大きさに関わらず一定となっている．特に回転角のサンプリング数については，化合物が大きくなるほど回転による化合物内の各原子の移動距離が大きくなるため影響が大きくなる．
  この点に関しては，**化合物の大きさに合わせて回転角のサンプリング数を多くすることで精度が向上する可能性があると考えられる**．

4.3 提案手法の速度に関する考察

4.3.1 化合物件数の増加に伴う速度向上の可能性

本研究で使用したデータセットにおいても，**化合物件数の増加に対してフラグメント種類数の増加が緩やかであることが確認された**．フラグメントの種類数が相対的に少なくなることでフラグメントの計算結果の再利用性が向上するため，さらに化合物件数を増やした場合により高速に化合物を評価できる可能性が高い．具体的には，今回の実験では化合物群を 1,000 件単位で分割したため表 3 に示すように化合物件数とフラグメント種類数の比はおおよそ 1:1 であるが，一千万件を超える化合物群では化合物件数とフラグメント種類数の比が 100:1 程度になるため [4]，理論的には 100 倍近い高速化が見込める可能性があると考えら

れる．しかし実際には，現在の提案手法においては局所最適化等の入力化合物件数に比例した時間がかかる処理が全体の 3 割ほどの時間を要しており，十分に高い高速化率を達成するにはこれらの処理の改良が必要であると考えられる．

5. 結論

5.1 本研究の結論

本研究では，薬剤候補化合物の多くが共通な部分構造を持つことに着目し，部分構造に対する相互作用スコアの計算結果を再利用することにより，1 つの標的タンパク質に対して大量の化合物を高速に評価することに特化したドッキングシミュレーションツールを開発した．

DUD-E Diverse Subset に含まれる 8 件のデータセットを用いて提案手法のドッキング精度及び速度を評価したところ，ドッキング精度は AutoDock Vina とほぼ同等に近い性能を示し，化合物群を 1,000 件単位で分割した際におけるドッキング速度では AutoDock Vina に比べ約 8.4 倍高速であることを確認した．

さらに，化合物件数が多くなるとその分共通なフラグメントが増え，フラグメントの計算結果の再利用効率が向上するため，数千万件の化合物のドッキングシミュレーションをまとめて行う場合にさらなる高速化が期待できる．しかし実際には局所最適化等の化合物件数に比例した時間を要する処理が存在するため，高い高速化率の達成にはそれらの処理の改良も必要である．

5.2 今後の課題

本研究の今後の課題として，以下の事項が考えられる．

5.2.1 より大きな化合物群を用いた評価

本研究では，TSUBAME 3.0 の連続 24 時間以下という実行時間制限により，DUD-E Diverse Subset の各データセットを 1,000 件ずつに分割した上で実験を行った．しかし，共通なフラグメントの増加に伴い実際に速度向上することを示すためにも，より大きな化合物件数における速度評価実験が必要である．

5.2.2 各種パラメータの最適化

提案手法には並進・回転移動の粒度，局所最適化を行うポーズ数等の様々なパラメータが存在するが，これらのパラメータはまだ十分に最適化されているとは言えない．よって，様々なデータセットについて実験を行いつつ各種パラメータの探索を行っていくことで精度または速度が向上することが考えられる．特に回転移動の粒度に関しては，HIVPR データセットに対する実験の結果より，化合物の大きさに応じて回転移動の粒度を変更することにより精度が向上する可能性がある．

謝辞 We thank to OpenEye Scientific Software Inc. for providing us with academic licenses for their software.

参考文献

[1] T. Sterling and J. J. Irwin. ZINC 15 – ligand discovery for everyone. *J. Chem. Inf. Model.*, 55(11):2324–2337, 2015.
[2] G. Sliwoski, S. Kothiwale, J. Meiler, and E. W. Lowe. Computational methods in drug discovery. *Pharmacol. Rev.*, 66(1):334–395, 2014.
[3] X.-Y. Meng, H.-X. Zhang, M. Mezei, and M. Cui. Molecular docking: A powerful approach for structure-based drug discovery. *Curr. Comput. Aided Drug Des.*, 7(2):146–157, 2011.
[4] K. Yanagisawa, S. Komine, S. D. Suzuki, *et al.* Spresso: an ultrafast compound pre-screening method based on compound decomposition. *Bioinformatics*, 33(23):3836–3843, 2017.
[5] Z. Zsoldos, D. Reid, A. Simon, S. B. Sadjad, and A. P. Johnson. eHiTS: A new fast, exhaustive flexible ligand docking system. *J. Mol. Graph. Model.*, 26(1):198–212, 2007.
[6] M. Rarey, B. Kramer, T. Lengauer, and G. Klebe. A fast flexible docking method using an incremental construction algorithm. *J. Mol. Biol.*, 261(3):470–489, 1996.
[7] 小峰駿汰, 石田貴士, 秋山泰. フラグメント伸長型タンパク質-化合物ドッキングのビームサーチによる高速化. **情報処理学会研究報告 バイオ情報学** *(BIO)*, 2015-BIO-42(62):1–8, 2015.
[8] M. McGann. FRED pose prediction and virtual screening accuracy. *J. Chem. Inf. Model.*, 51(3):578–596, 2011.
[9] N.-O. Friedrich, C. d. Bruyn Kops, F. Flachsenberg, *et al.* Benchmarking commercial conformer ensemble generators. *J. Chem. Inf. Model.*, 57(11):2719–2728, 2017.
[10] P. C. D. Hawkins, A. G. Skillman, G. L. Warren, B. A. Ellingson, and M. T. Stahl. Conformer generation with OMEGA: Algorithm and validation using high quality structures from the protein databank and cambridge structural database. *J. Chem. Inf. Model.*, 50(4):572–584, 2010.
[11] O. Trott and A. J. Olson. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.*, 31(2):455–461, 2010.
[12] K. Yanagisawa, S. Komine, R. Kubota, M. Ohue, and Y. Akiyama. Optimization of memory use of fragment extension-based protein-ligand docking with an original fast minimum cost flow algorithm. *Comput. Biol. Chem.*, 74:399–406, 2018.
[13] M. M. Mysinger, M. Carchia, J. J. Irwin, and B. K. Shoichet. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.*, 55(14):6582–6594, 2012.
[14] N. S. Pagadala, K. Syed, and J. Tuszynski. Software for molecular docking: a review. *Biophys. Rev.*, 9(2):91–102, 2017.
[15] R. A. Friesner, J. L. Banks, R. B. Murphy, *et al.* Glide: A new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *J. Med. Chem.*, 47(7):1739–1749, 2004.
[16] S. Release 2018-3. LigPrep, Schrödinger, LLC, New York, NY, 2018.
[17] W. P. Feinstein and M. Brylinski. Calculating an optimal box size for ligand docking and virtual screening against experimental and predicted binding pockets. *J. Cheminformatics*, 7(1):18, 2015.