

# 目次

第1章	序論	1
1.1	研究背景	1
1.2	研究目的	2
1.3	本論文の構成	2
第2章	ドッキングシミュレーションによる薬物候補化合物の選別 (Structure-based Virtual Screening, SBVS)	3
2.1	SBVSとは	3
2.2	化合物-タンパク質ドッキングシミュレーション	3
2.2.1	ドッキングシミュレーションの要素	4
2.2.2	ドッキングシミュレーションの問題点	5
2.3	化合物のフィルタリング	6
2.3.1	既存のフィルタリング手法	6
2.3.2	既存手法の問題点	6
第3章	提案手法：化合物の部分構造を利用したフィルタリング手法の開発	7
3.1	提案手法の概説	7
3.1.1	フィルタリングの要件	7
3.1.2	提案手法へのアイデア	8
3.2	提案手法の詳細の説明	8
3.2.1	提案手法のフローチャート	9
3.2.2	化合物のフラグメントへの分割	9
3.2.3	フラグメント単位でのドッキングシミュレーション	11
3.2.4	化合物のフィルタリングスコアの算出	11
第4章	実験	14
4.1	データセット	14
4.2	予測精度の評価指標	14
4.3	計算環境	16

---

4.4	比較対象	16
4.5	評価実験	16
4.5.1	フラグメント分割	16
4.5.2	ドッキング速度の評価	17
4.5.3	予測精度の評価	17
4.5.4	フィルタリング手法としての性能評価実験	18
<b>第5章</b>	<b>考察</b>	<b>21</b>
5.1	提案手法の得手・不得手の調査	21
5.1.1	提案手法が得意なターゲット	22
5.1.2	提案手法が不得意なターゲット	22
5.2	提案手法のユースケース	22
<b>第6章</b>	<b>結論</b>	<b>23</b>
6.1	本研究の結論	23
6.2	今後の課題	23
	謝辞	24
	参考文献	25
付録A	DUD-Eの詳細	27
付録B	ROC曲線	28

## 図 目 次

3.1	スコア統合イメージ (暫定版) . . . . .	9
3.2	化合物のフラグメント分割アルゴリズム <sup>1)</sup> . . . . .	10
3.3	大量の化合物を分割した場合の例 . . . . .	11
3.4	フラグメントの結合スコアの取得 . . . . .	12
3.5	maxsumBS の算出 . . . . .	13
4.1	ROC-AUC 計算例 . . . . .	15
4.2	EF 計算例 . . . . .	16
4.3	EF (1%)、EF (2%) 算出までの流れ . . . . .	19

## 表 目 次

4.1	DUD-E のターゲットの化合物 . . . . .	14
4.2	ドッキング計算時間の比較 . . . . .	17
4.3	提案手法の予測精度 . . . . .	17
4.4	フィルタリング手法としての提案手法間の精度評価 . . . . .	20
4.5	フィルタリング手法としての提案手法と従来手法の比較 . . . . .	20
5.1	提案手法が上手く行ったケース . . . . .	21
5.2	提案手法が上手く行かないケース . . . . .	22
A.1	DUD-E の詳細 . . . . .	27

## 第1章

# 序論

### 1.1 研究背景

近年、創薬の初期段階においてバーチャルスクリーニング (Virtual Screening, VS) と呼ばれる、コンピュータによる予測を用いて大量の化合物から薬剤候補化合物を選別する手法を用いることで創薬コストの削減、および創薬にかかる時間の短縮が試みられている。このコンピュータを用いた化合物の選別手法は大きく3つに分けられる。

- タンパク質や化合物の立体構造を用いた手法 (Structure-Based Virtual Screening, SBVS)
  - ー タンパク質-化合物ドッキングシミュレーション @cite Glide, eHiTS, Autodock
- 既知の薬剤・タンパク質の活動を阻害する化合物 (阻害剤) の情報を用いた手法 (Ligand-Based Virtual Screening, LBVS)
  - ー 構造活性相関 (Quantitative Structure-Activity Relationship, QSAR) を用いた手法 @cite
  - ー 機械学習による分類手法 @cite
  - ー 化合物の官能基の性質を用いたファーマコフォアモデルに基づく化合物分類手法 @cite
- タンパク質と薬剤との2部グラフなどのネットワークを構築し、類似度から予測を行う創薬手法 (Chemical Genomics-Based Virtual Screening, CGBVS) @cite Brown & Okuno (2012)

このうち、タンパク質-化合物ドッキングシミュレーションによるSBVSは物理的なエネルギーを計算する演繹的な手法であり、既知の薬剤や阻害剤が存在しない創薬標的であってもタンパク質の構造得られれば薬物候補化合物を選別することができる、非常に有用な方法である。また、既知の薬剤や阻害剤から法則性を見つけ出すなど帰納的な手法であるLB等に比べて既知の薬剤や阻害剤と大きく性質の異なる、「新規の構造を持った」薬剤候補化合物を発見する能力が高いこともドッキングシミュレーションによるSBVSのメリットである。ドッキングシミュレーションはGlide @cite ,

eHiTS @cite , Autodock @cite を始めとして多様なツールが開発されており、その中でも Glide は予測精度が高く @cite 比較論文、比較的広く利用されている。しかし、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結合構造という非常に複雑な探索空間の中での最適化問題を解くため、計算コストが非常に高い。これを解決するためにドッキング手法の高速化の研究 @cite Autodock の GPU 実装, BUDE, Autodock Vina が行われているが、速度的、もしくは精度的に未だ不十分であり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 @cite に存在する xxx 件の化合物を一斉にドッキングシミュレーションで予測することは困難というのが現状である。

以上の理由から、SBVSを用いた創薬研究ではドッキングシミュレーションを行う前に化合物を選別する、フィルタリングが行われることが多い @cite 実例を示す。しかし、このフィルタリング手法の多くはLBVSのように、既知の薬剤などの化合物情報を用いるものであり、前述したSBVSの長所である「既知の薬剤や阻害剤と大きく性質の異なる薬剤候補化合物」をフィルタリングで落としてしまうことが多く、ドッキングシミュレーションとは相性が悪い。また、Glideの簡易ドッキングモードであるHTVSモードを用いてフィルタリングを行うこともあり @cite Glide HTVSをフィルタリング手法に用いている論文、この手法を用いればSBVSの長所を損なうことなくフィルタリングを行うことができるが、前述したような数千万単位の化合物数ではGlide HTVSモードですら計算量が膨大になってしまう。また、Glideは計算に利用するコア数に応じてライセンスを購入しなければならない形式の商用ソフトであり、TSUBAME2.5などのスーパーコンピュータの大規模利用による高速化を行うことができない。

## 1.2 研究目的

1.1節で示したように、SBVSにおけるフィルタリングは未だ研究が不十分であり、高速に、新規の構造を持つ、見込みのある化合物を残すフィルタリング手法を開発する必要がある。本論文では、ドッキングに基づいた、フィルタリングに特化した手法を提案し、ドッキングに基づいたフィルタリングの既存手法であるGlide HTVSと比較、提案手法の有用性を述べる。

## 1.3 本論文の構成

2章では、ドッキングシミュレーションに基づいたSBVSについての説明を行い、同時に既存のフィルタリング手法について説明する。3章では提案手法について述べ、4章でこの提案手法と既存手法であるGlide HTVSとの比較を行う。また、5章では4章で行った実験の結果についての考察を加え、6章で結論および今後の展望を述べる。

## 第2章

# ドッキングシミュレーションによる薬物候補化合物の選別 (Structure-based Virtual Screening, SBVS)

この章ではドッキングシミュレーションに基づく化合物の選別手法を説明し、既存の化合物フィルタリング手法を紹介する。

## 2.1 SBVSとは

バーチャルスクリーニング (Virtual Screening, VS) とは、コンピュータを用い、データベースに存在する化合物について、創薬標的となっているタンパク質の活性部位への結合のしやすさを仮想的に (Virtual) 評価、選別 (Screening) することを指す。化合物の評価・選別を創薬標的のタンパク質や化合物の立体構造に基づいて行う手法のことを SBVS と呼ぶ。この SBVS は、化合物の評価・選別を既知の創薬標的タンパク質へ結合する化合物 (リガンド, ligand) を用いて行う LBVS (Ligand-based Virtual Screening) と比べて

- 既知のリガンドを必要とせず
- 既知のリガンドにとらわれない、多様な薬剤候補化合物を得ることができる

という長所を持っている。

## 2.2 化合物-タンパク質ドッキングシミュレーション

SBVS における化合物の評価には化合物-タンパク質ドッキングシミュレーションが一般に用いられる。ドッキングシミュレーションは、1つのタンパク質の立体構造と1つの化合物の立体構造を

入力として、化合物がタンパク質中でどのような構造をとるとエネルギー的に最も安定であるかという最適化問題を解き、最安定であると考えられる化合物の構造とその時のスコアを出力する(図??)。@todo docking を説明する図を作成 SBVSにおける複数の化合物の選別にはドッキングシミュレーションによって得られたスコアを直接用いるか、もしくは得られたスコアを何らかの形で変換し、評価値がより良かった化合物を薬物候補化合物として残す。

Glide,<sup>2)</sup> eHiTS @cite などの有償ソフトウェア、Autodock @cite などのオープンソースウェアを始めとして、有償無償問わず様々なドッキングシミュレーションツールが開発されている。

### 2.2.1 ドッキングシミュレーションの要素

SBVSの薬物候補化合物の選別はドッキングシミュレーションによって得られたスコアを基に行われるため算出されるスコアは重要となるが、後述するように探索空間が非常に広く、さらに最適化を行うべきスコア値も一般的に探索空間内で単調ではないため、厳密な最適スコアを求めることは事実上不可能である。そのため、ドッキングシミュレーションにおいては

- 非常に広い探索空間からなる最適化問題で良い準最適解を効率良く見つける探索アルゴリズム
- 適度に高速に計算でき、タンパク質-化合物の結合構造の良し悪しを適切に見積もるスコア関数

の2つは非常に重要であり、これらは1982年に最初のドッキングシミュレーションツールであるDOCK @citeが開発されてより、様々なグループによって研究が進められている。

#### 探索空間

ドッキングシミュレーションでは、タンパク質の位置を固定として、化合物がタンパク質とどのような構造をとると良いかを探索する。この際、探索しなければならない空間は化合物の並進運動および回転運動の6次元に加え、化合物の内部に回転可能な結合を持つため化合物の内部自由度を考慮しなければならない(図??)。この内部自由度はZINC Drug Databaseに登録されている2924個の薬剤化合物で平均 **x.xx** と少なくとも、計算量に大きな影響を与える。

#### 探索アルゴリズム

前述のように探索空間の広さのために大域最適解を求めることは困難であるため、より良い局所最適解を求めるための工夫がツール毎になされている。



- Glide

段階的な全探索を行うことで局所最適解を得る。具体的には、最初の段階では化合物を球体に近似しての位置が良いかどうかの見積もりから始め、徐々に化合物の近似を厳密なものにしていく。それぞれの段階で上位の位置・構造のみを残し次の段階へ進めることで、全探索の空間を現実的な量に制限し、探索を完了させる。@todo イメージ図が論文中にあれば載せる

- eHiTS

化合物を部分構造に分割し、部分構造にとって良い構造をそれぞれ多数記録し、ノードにする。その後、2つの部分構造が構造を構成するのに適度な距離、適度な向きになっているノード間にエッジを張り、作成されたグラフに関して最大クリーク問題を解くことで適切な構造を得る。@todo イメージ図が論文中にあれば載せる

- Autodock

並進運動位置、回転運動位置、化合物の内部回転角を用いた遺伝的アルゴリズム (Genetic algorithm, GA) でより良い局所最適解を得る。@todo イメージ図が論文中にあれば載せる

## スコア関数

探索アルゴリズムがどれほど良く、大域最適なスコアを得たとしても、そのスコアがタンパク質と化合物との物理的な結合エネルギーとの相関がなければ意味がない。しかし、結合エネルギーを厳密に計算するには量子化学計算が必要となり、実用的な時間では計算が完了しないので、近似計算が必要となる。したがって、スコア関数に関しても様々な提案がなされている。@memo スコア関数の内容に関しても Chang<sup>3)</sup> が示しているように Force field, Empirical, Knowledge based で分類しながら詳細すべきでは

### 2.2.2 ドッキングシミュレーションの問題点

2.2.1 節に述べたように、ドッキングシミュレーションツールはそれぞれ高速化のための工夫を凝らしているが、それでも不十分であるのが現状である。例えば、1 コアを用いて 1 つの化合物を評価するのに Glide で 0.2-2.4 分程度 @cite glidel、eHiTS は最速で数秒 @cite eHiTS を要すると述べられている。この速度で 1,000 万化合物を選別しようとする 10 秒で 1 つの化合物を評価できたとしても 1,200 CPU days もの時間を必要とする。このような場合に一般的に用いられる手段である大規模計算化に関しても、Glide や eHiTS はライセンス式の有償ソフトウェアであるために、大量のライセンスを購入する必要がある現実的ではない。

一方、AutoDock はライセンスが必要なく、大規模並列計算が可能であるが、Glide と比べて 250 倍程度も遅いという報告がなされている<sup>4)</sup>。AutoDock はオープンソースウェアであるため、GPU

実装による高速化も提案されているが、遺伝的アルゴリズムやスコア関数の計算が最大 50 倍程度高速になる程度であり<sup>5)</sup>、Glide に及ばない。

## 2.3 化合物のフィルタリング

ドッキングシミュレーションは大きな計算量を必要とするために、1,000 万もの化合物から薬物候補化合物を選別しようとするのが非常に難しいことを 2.2.2 節で述べた。このため、ドッキングシミュレーションを高速化するのではなく、ドッキングシミュレーションの入力とする化合物の数をあらかじめ減ずることで総計算時間を削減するという戦略が創薬研究では良く用いられる。

### 2.3.1 既存のフィルタリング手法

既存のフィルタリング手法は大きく分けて化合物の物理的特徴に基づくフィルタリング、化合物の構造に基づくフィルタリング、ドッキングベースのフィルタリングの 3 種類が存在している。

化合物の物理的特徴に基づくフィルタリング

**@todo Lipinski's rule of five,<sup>6)</sup> QED,<sup>7)</sup> RDL<sup>8)</sup> を紹介。少し方向性は違うが。**

化合物の構造に基づくフィルタリング

fingerprint ベースでのフィルタリングをここに記述。ファーマコフォアもこちらか。

ドッキングベースのフィルタリング

glide HTVS がフィルタリングに利用されるケースを記述。

### 2.3.2 既存手法の問題点

以下の 2 点を示す。

- Glide HTVS をフィルタリングに転用する手法では高速化の度合いが不十分であり、仮想的に化合物空間を広げたようなデータセットに対するドッキングなどを行う条件下では計算時間的に現実的ではない
- 化合物ベースの手法 (含 Pharmacophore) は構造ベースよりも一般に高速だが、既知阻害剤があることが条件になり、さらに得られる化合物は既知の化合物に似てしまうという問題がある (前述した通り～みたいな感じで)

## 第3章

# 提案手法：化合物の部分構造を利用したフィルタリング手法の開発

ここでは、従来手法である Glide HTVS とは異なり、化合物を部分構造に分割することで高速にドッキングを完了させるフィルタリング手法の内容を説明する。

### 3.1 提案手法の概説

前章で述べた通り、ドッキングシミュレーションは時間を要し、その理由は探索空間の広さと化合物の多様性にある。この節では、この2つの問題を解決するアイデア、および高速にフィルタリングを行うために追加する仮定を説明する。

#### 3.1.1 フィルタリングの要件

フィルタリングに求められる要件は2つ存在する。

- 高速に化合物を評価する  
フィルタリングを実用的に行うためには、フィルタリング後に行うドッキングシミュレーションよりも十分に高速である必要がある。
- 予測の精度がある程度保持されている  
一般に計算速度と予測精度はトレードオフの関係にあるが、どれほど高速であってもある程度正例と負例が弁別できなければフィルタリングとして機能しない。したがって、予測精度がある程度保持されていることもフィルタリングには求められる。

一方、フィルタリングはその後に通常のドッキングシミュレーションを行うことを前提とするため、必ずしも「化合物がタンパク質のこの部分に結合する」というドッキングポーズを出力する必要はない。

### 3.1.2 提案手法へのアイデア

前節で示したフィルタリングの要件を満たすために、2つのアイデアを考案した。

#### 化合物を部分構造に分割し、部分構造のドッキングシミュレーションを行う

ドッキングシミュレーションの探索空間のうち、並進運動と回転運動による探索空間の次元は合計6次元と固定されている。一方、化合物の内部自由度は化合物の回転可能な結合の数によって変化し、DUD-E データセットに登録されている化合物の平均は **x.xx** と無視できないほど大きな値である。そこで、本提案手法では、小峰ら [@cite](#) による化合物の分割方法を用いて、化合物を内部自由度を考慮しなくて良い「フラグメント」に分割、これらをドッキングすることで、必要最低限の探索空間でのドッキングシミュレーションを実現する。

#### フラグメントから化合物の構造を再構成せず、フラグメントの結合スコアから化合物のフィルタリングスコアを算出する

化合物をフラグメントに分割した上でドッキングシミュレーションを行うと図??[@todo](#) 図の作成のようにフラグメントごとにタンパク質との結合予測構造が出力され、フラグメントの結合スコアが最も良いポーズを選択したとしても繋がった一つの化合物としては有り得ない構造をとる場合がほとんどである。しかし、矛盾のない化合物の構造をとるようなフラグメントの選択を行うのは  $O(a^n)$  ( $n$  は化合物を構成するフラグメント数) の計算量となり、大きな計算コストを要してしまう。

一方、フィルタリングはその後に通常のドッキングシミュレーションを行うことを前提とするため、必ずしも化合物とタンパク質との結合予測構造を出力する必要はない。そこで、提案手法では構造の矛盾の考慮を行わず、得られたフラグメントの結合スコアのみに着目し、フラグメントの結合スコアから化合物のフィルタリングスコアを算出するのに計算が  $O(n)$  で済むようなスコアの統合を行うことで、高速な化合物の評価を達成する。

## 3.2 提案手法の詳細の説明

前節で用いる2つのアイデアを示したが、それを用いてどのようにフィルタリングを実現しているのかをこの節で詳説する。

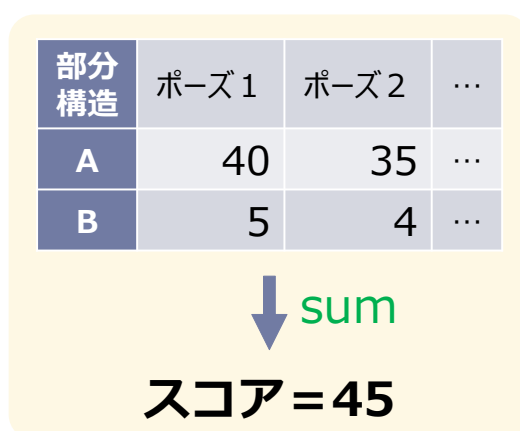


図 3.1 スコア統合イメージ (暫定版)

### 3.2.1 提案手法のフローチャート

提案手法は以下の手順で構成される。

1. 入力された化合物をフラグメントに分割する
2. ドッキングシミュレーションツールを用いてフラグメントの標的タンパク質への結合スコアを算出する
3. フラグメントの結合スコアから化合物のフィルタリングスコアを算出する
4. フィルタリングスコアの上位 N%をフィルタを通過した化合物として出力する

フローチャートを図??に示す。

### 3.2.2 化合物のフラグメントへの分割

化合物の分割は小峰らによる手法<sup>1)</sup>を用い、内部自由度を持たない部分構造であるフラグメントを生成する。実装には C++ を用い、ケモインフォマティクスツールである OpenBabel<sup>9)</sup> および OpenMP、Boost を利用している。フラグメント分割のアルゴリズムを以下に示し、このアルゴリズムによるフラグメント分割の進行を図 3.2 に示す。

1. 元の分子のうち、重原子（水素以外の原子）のみに着目し、原子一つひとつをフラグメントとする。（図 3.2 左から 2 番目）
2. 回転不可能な単結合以外の結合の両端の 2 原子を同一フラグメントとする。

3. 環構造を構成している原子を同一フラグメントとする
4. 回転可能な単結合を構成する原子ペアのうち、片方にそれ以上原子がつながっていない場合には同一フラグメントとする。これは、片方にそれ以上の原子がつながっていない場合、回転可能な単結合を回転させてもその原子がその場で回転するだけとなり、化合物の原子の位置関係には影響を与えないためである。(図 3.2 左から 3 番目)
5. 2 つの単結合の切断により孤立してしまう原子は、切断された先に存在する 2 つのフラグメントのどちらかに併合する。なお、3 つ以上の単結合の切断により孤立してしまう原子に関してはこの操作を行わない。
6. 全ての水素原子について、その原子が結合している重原子の属するフラグメントに含める。(図 3.2 左から 4 番目)

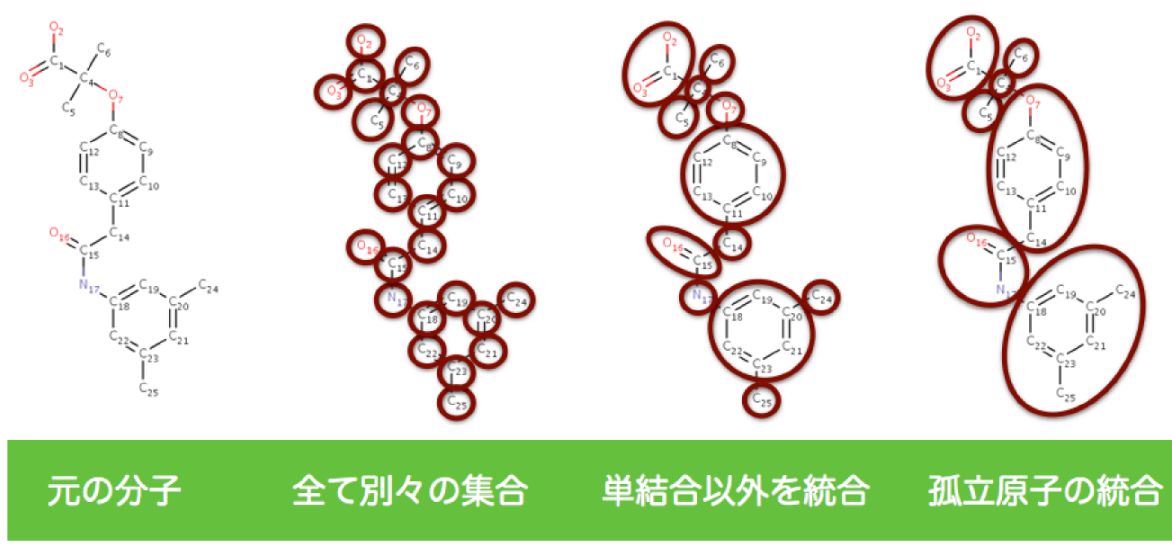


図 3.2 化合物のフラグメント分割アルゴリズム<sup>1)</sup>

この化合物のフラグメントへの分割により、内部自由度を考慮することなくドッキングシミュレーションを行うことができる。また、複数の化合物間で部分構造に共通性が見られることが非常に多く、本研究で用いている分割手法によって得られるものの中にも多数の共通フラグメントが発生する。例えば、ZINC の”drugs now”データセットに含まれている 10,639,555 化合物を順次フラグメント分割した場合のフラグメントの種類数をプロットすると、図 3.3 のようになり、わずか 20 万フラグメントによって 1,000 万化合物が構成されていることが分かる。また、プロットの曲線具合からもわかるように、フラグメント分割を行いドッキングを行うという手法は、化合物数が多いほど化合物単位でドッキングする手法に比べて優位になる。

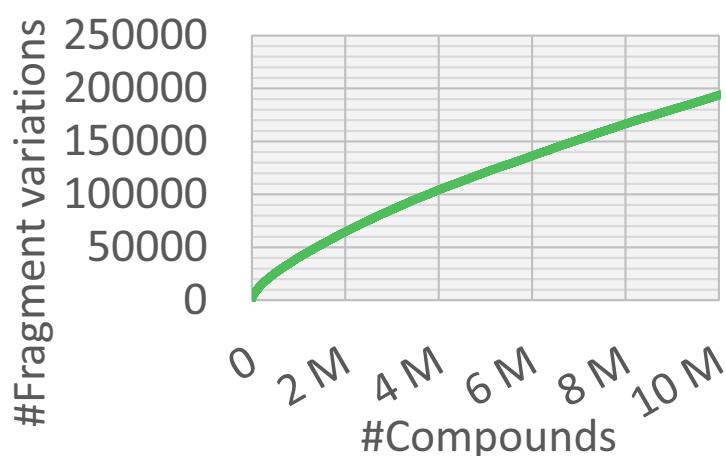


図 3.3 大量の化合物を分割した場合の例

### 3.2.3 フラグメント単位でのドッキングシミュレーション

次に、分割されたフラグメントについて、標的タンパク質との結合スコアを求めるためにドッキングシミュレーションを行う。本研究では、有償ソフトである glide [@cite](#) を用いる。glide には高速 (HTVS) モード、通常 (SP) モード、精密 (XP) モードの 3 種類のモードが存在するが、本研究では SP モードと HTVS モードを利用した場合の評価を行う。SP モードはデフォルト設定では内部自由度を考慮したドッキングを行ってしまうため、内部自由度を無視するオプションを追加している。また、一般的に 1 つのタンパク質と 1 つの化合物とのドッキング結果では複数のタンパク質-フラグメント結合予測構造および結合スコアが出力されるが、この後の化合物のフィルタリングスコアの算出ではこのうち最良の結合スコアを利用する (図 3.4)。

### 3.2.4 化合物のフィルタリングスコアの算出

フラグメント単位でのドッキングシミュレーションによって、フラグメントの結合構造およびその結合スコアを得た。続いて、このフラグメント結合スコアから化合物のフィルタリングに用いるスコアを算出する。本研究では、3 種類のスコアの算出方法の実験を行った。なお、重原子数が 2 以下の小さなフラグメントの結合スコアはフィルタリングスコア算出から除外している。

#### 総和法 (score\_sum)

フラグメント結合スコアの総和をとり、それを化合物のフィルタリングスコアとする。構成する全てのフラグメントがタンパク質と良い結合構造を取れるような化合物が薬物候補化合物として適



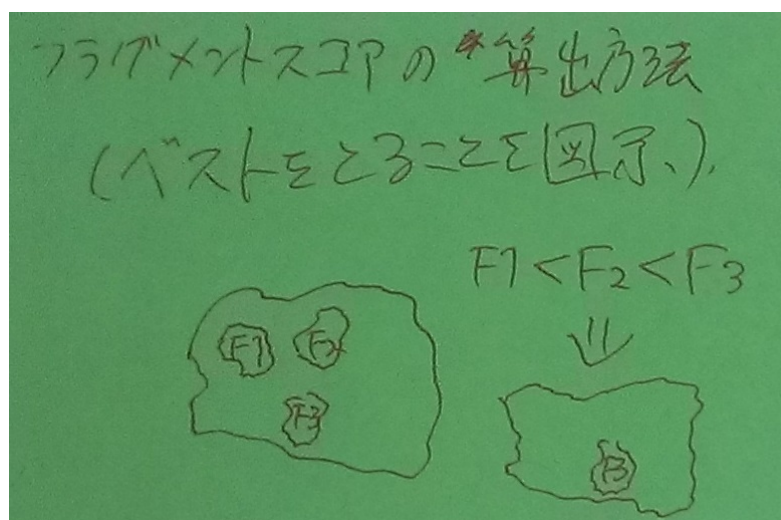


図 3.4 フラグメントの結合スコアの取得

している、として評価を高くする手法である。フラグメント群は化合物に存在する結合という束縛条件を一部緩和したものであるため、一般にこの手法によって得られた化合物フィルタリングスコアは化合物そのものの結合スコアよりも高くなる。

#### 最良値法 (score\_max)

フラグメント結合スコアの最良値をとり、それを化合物のフィルタリングスコアとする。構成するフラグメントの内、1つでもタンパク質と非常に良い結合構造をとれるような化合物が薬物候補化合物として適している、として評価を高くする手法である。フラグメント1つの結合スコアが化合物のフィルタリングスコアとなること、ドッキングシミュレーションを行う分子のサイズと結合スコアには正の相関がある<sup>10)</sup>ことから、総和法とは異なりこの手法によって得られた化合物フィルタリングスコアは化合物そのものの結合スコアよりも低くなる。

#### 総和法と最良値法の値の線形和 (maxsumBS)

これまでに示した総和法と最良値法はフラグメント結合スコアの全て、もしくはただ一つを見る手法であり両極端であるため、これらを統合して用いることで、より良い指標となるのではないかと考えた。しかし、総和法の値域が最良値法の値域よりも大きいため単純和では総和法の影響を大きく受けてしまう。そこで、二つの手法を適当なバランスで組み合わせるために、フィルタリングを行いたい化合物の総和法によるスコア、最良値法によるスコアをそれぞれ平均0、分散1にし(すなわちzスコア化し)、変換後のスコアを足し合わせることでバランスよくスコアを統合することを試みた(図3.5)。なお、総和法によるスコアと最良値法によるスコアのバランスをとったスコ



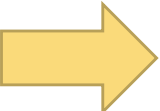
	総和法	最良値法		総和法 z	最良値法 z	和
A	20	8	$z_{score} = \frac{x - \mu}{\sigma}$ 	1.225	0.267	<b>1.492</b>
B	15	4		0	-1.336	<b>-1.336</b>
C	10	10		-1.225	1.069	<b>-0.156</b>

図 3.5 maxsumBS の算出

ア、という意味で maxsumBS (max-sum Balanced Score) として以下記述する。

## 第4章

# 実験

ここでは、提案手法と既存手法との比較実験を行い、提案手法の長所を示す。

### 4.1 データセット

本実験では、データセットとして Directory of Useful Decoys (DUD-E)<sup>11)</sup> を用いた。DUD-E は 102 種類のターゲットについて、それぞれタンパク質・正例化合物・負例化合物を用意している。表 4.1 にターゲットごとの化合物数、正例と負例の比率の最小値、最大値、平均値を、図??に総化合物数の分布を示す@todo ヒストグラム作成。各ターゲットの詳細については付録 A に記載する。なお、DUD-E のターゲットのうち fgfr1 および fa10 は記載されている負例数とデータセットに実際に含まれている負例数が大きく異なっているが、そのまま扱うこととする。

表 4.1 DUD-E のターゲットの化合物

	総化合物数	正/負例の比率
最大値	52,022 (fnta)	1:104 (fnta)
平均値	13,881	1:60
最小値	472 (fgfr1)	1:2.4 (fgfr1)

### 4.2 予測精度の評価指標

バーチャルスクリーニングでは、計算機による選別を通過して活性実験が行われる化合物数が母数に比べて非常に少なくまた化合物ライブラリの中で実際に標的タンパク質に結合し、活動を阻害するのは 1000 個に 1 個などとも言われており @cite これどこで言われてるの?、正例と負例の比が非常に偏っている。そのためこの分野における予測精度の評価指標は以下の 2 種類が多く用いられている。

**ROC-AUC** Receiver Operating Characteristic (ROC) 曲線は、正例/負例の閾値を変化させながら、縦軸に True Positive (TP) 率、横軸に False Positive (FP) 率をとった曲線である。TP 率とはデータセット中の正例の中で正しく正例と判別されたものの割合であり、FP 率とはデータセット中の負例の中で誤って正例と判別されたものの割合である。TP 率、FP 率はそれぞれ以下の式で求められる。

$$\text{TP 率} = \frac{\#TP}{\#TP + \#FN} \quad (4.1)$$

$$\text{FP 率} = \frac{\#FP}{\#FP + \#TN} \quad (4.2)$$

この方法によって描かれた ROC 曲線の曲線下面積 (Area Under the Curve, AUC) を用いた評価指標が ROC-AUC である。具体例を図 4.1 に示す。

**Enrichment Factor** Enrichment Factor (EF) とは、予測結果の上位のみを取り出したときに、元々のデータセットからどれだけ正例が「濃縮されたか」を表す指標である。具体例を図 4.2 に示す。上位どのくらいを取り出すかによって値が異なり、上位 x%取り出したときの集合の正例率を正例率 (x%)、EF を EF (x%) と表記することになると、これらは以下の式で求められる。

$$\text{正例率 (x\%)} = \frac{\text{正例数 (x\%)}}{\text{正例数 (x\%) + 負例数 (x\%)}} \quad (4.3)$$

$$\text{EF (x\%)} = \frac{\text{正例率 (x\%)}}{\text{正例率 (100\%)}} \quad (4.4)$$

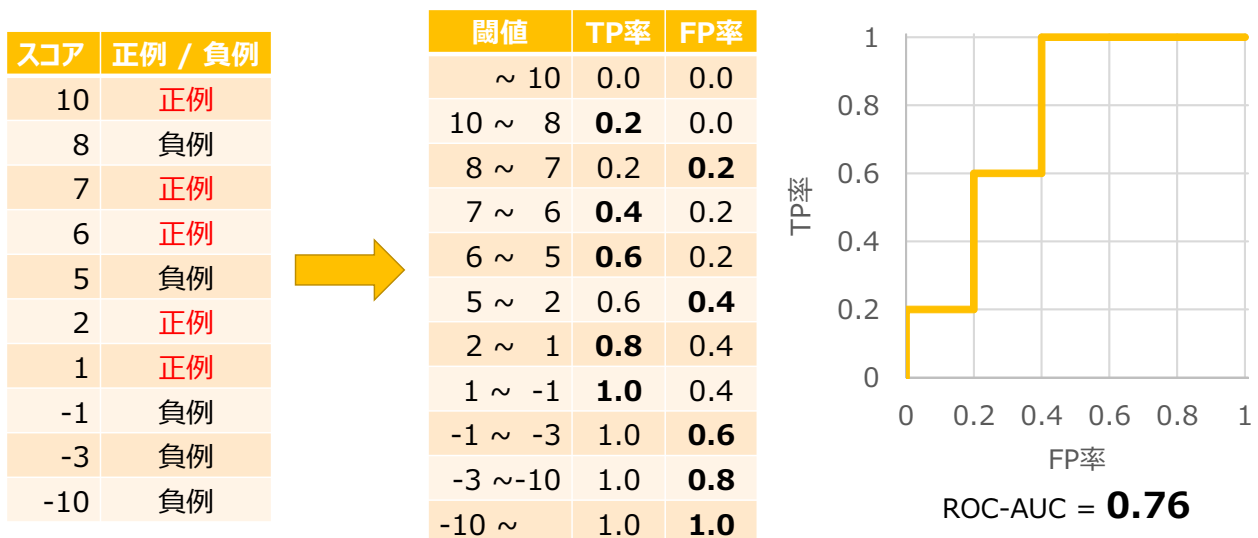


図 4.1 ROC-AUC 計算例

本研究においては、ROC-AUC、EF (1%)、EF (2%)、EF (5%)、EF (10%) の 5 つの指標を用いて手法の評価を行う。

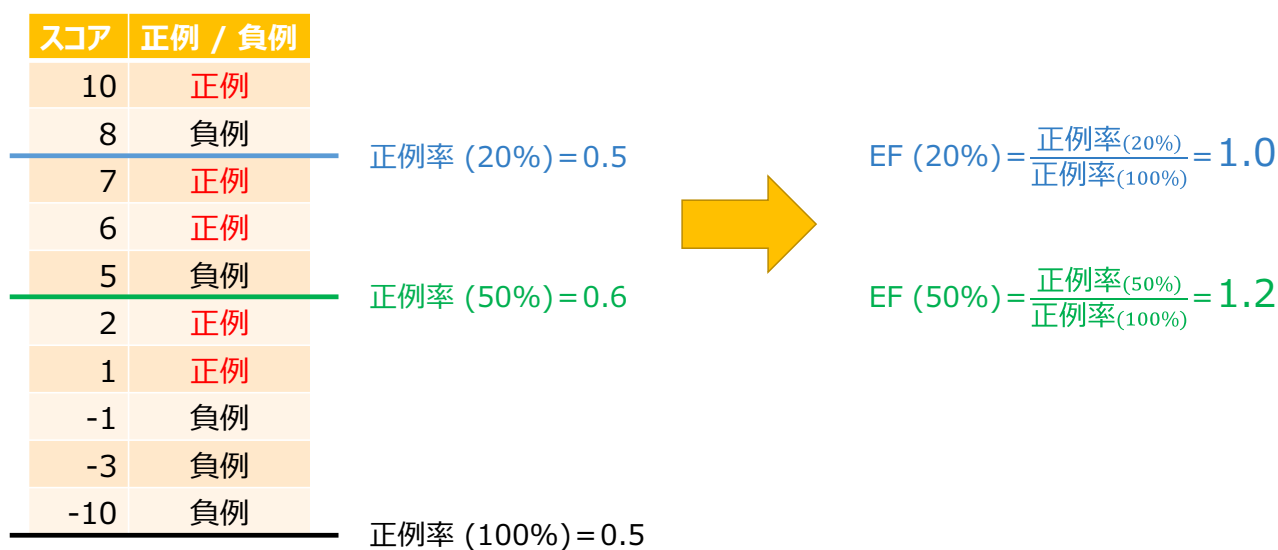


図 4.2 EF 計算例

## 4.3 計算環境

TSUBAME Thin ノード@todo 先輩の修論の記述を参考にする

## 4.4 比較対象

本提案手法はドッキングに基づくフィルタリング手法であるため、同様の用途に用いられている glide HTVS（高速）モードを比較対象として用いる。また、フィルタリングとしての性能を評価するために、glide SP（通常）モードによる化合物ドッキングシミュレーションと組み合わせた評価も行うため、計算時間などの評価に関しては glide SP モードも比較対象とする。

## 4.5 評価実験

### 4.5.1 フラグメント分割

@todo フラグメント分割の結果について述べる節をドッキング速度の評価から独立させる。化合物数とフラグメント種類数の 2 軸の平面でプロットを行うことで化合物数が多いほどフラグメント分割は有用であることを再度確認する。

### 4.5.2 ドッキング速度の評価

はじめに、フィルタリング手法の計算速度を評価する。提案手法は3.2.2節で述べたように化合物数が多ければ多いほど計算コストの削減幅が増幅される手法であるため、DUD-E 102 ターゲット全てでの所要計算時間の平均以外に、総化合物数が最小であるターゲット fgfr1、総化合物数が平均値に最も近いターゲット andr、総化合物数が最大であるターゲット fnta の3種類については独立して結果を示す。

結果は表 4.2 の通りであり、提案手法は既存手法である glide HTVS と比べて平均して約 9 倍（SP モード利用時）から約 15 倍（HTVS モード利用時）の速度向上を達成している。**@todo 表に倍数を載せるべき。どこに置くか考えなければ**

表 4.2 ドッキング計算時間の比較

ターゲット名	総化合物数	フラグメント 種類数	計算時間 [CPU sec.]			
			化合物ドッキング		フラグメントドッキング	
			glide SP	glide HTVS	glide SP	glide HTVS
fgfr1	472	166	3,523	566	164	140
andr	14,612	5,030	42,165	5,498	2,401	1,396
fnta	52,022	7,767	1,770,967	98,665	4,149	2,549
全ての平均	13,881	3,231	236,156	14,813	1,673	987

### 4.5.3 予測精度の評価

次に、提案手法の予測精度の評価を行う。提案手法は2つのドッキングモード（SP モードおよび HTVS モード）、3つのフィルタリングスコア算出方法が存在するため合計 6 通りを示す。

表 4.3 提案手法の予測精度

手法	フラグメント ドッキング	ROC-AUC	Enrichment Factor			
			EF(1%)	EF(2%)	EF(5%)	EF(10%)
総和 (score_sum)	glide SP モード	0.624	5.08	4.14	3.02	2.34
	glide HTVS モード					
最良値 (score_max)	glide SP モード	0.627	6.78	5.47	3.58	2.43
	glide HTVS モード					
線形和 (maxsumBS)	glide SP モード	0.679	6.03	5.03	3.96	3.00
	glide HTVS モード	0.665	5.98	4.84	3.58	2.82
従来手法 (glide HTVS モード)		<b>0.705</b>	<b>16.67</b>	<b>11.18</b>	<b>6.38</b>	<b>4.11</b>

結果は表 4.3 の通りである。なお、各手法を用いた場合のターゲットごとの ROC 曲線は付録 B に記載している。この結果から、単体での予測精度に関しては、どの評価指標においても従来手法が勝っていることが分かる。また、フラグメントドッキングについては glide SP モードを利用した方が精度がよくなっている。速度については 4.5.2 で述べたように glide SP モードを利用したフラグメントドッキングの場合でも 9 倍近く従来手法に比べて高速であるため、予測精度を重要視し以下の実験では glide SP モードを用いることとする。

#### 4.5.4 フィルタリング手法としての性能評価実験

4.5.2 節および 4.5.4 節では、フィルタリング手法を単体で用いた場合の性能を評価し、速度では提案手法が勝っているものの、精度では従来手法に後塵を拝する結果となった。しかし、本研究で提案した手法はフィルタリングを想定したものであり、その次に行われる通常のドッキングシミュレーション手法と組み合わせた場合の速度や精度の評価はより重要となる。

この節では通常のドッキングシミュレーションである glide SP モードとの組み合わせを通した評価を行う。なお、ROC 曲線は評価対象全てにスコアがつく必要があるが、フィルタリングの結果 glide SP モードで評価されない化合物にはスコアがつかないため、ROC-AUC を求めることはできない。そのため、評価指標は EF のみを用いることとする。

##### 提案手法間の精度比較

まず、提案手法間の精度比較を行う。フィルタリング手法で 2%、5%、10% の 3 通りまで化合物を削減し、残った化合物を通常のドッキングシミュレーション (glide SP モード) で再評価、その順位に従い EF (1%) および EF (2%) を求める (図 4.3)。なお、フィルタリング手法を用いて 2% まで削減し、そこで EF (2%) を求める場合、再評価の結果に関わらずフィルタリングで残った全ての化合物を用いて Enrichment Factor を計算することになるため、「-」表記とし数値を出さないようにしている。

結果は表 4.5 のようになり、どのようなケースにおいても、フィルタリングスコア算出方法は総和法と最良値法の線形和である maxsumBS を用いるのが最適であることが分かった。

##### 予測精度の従来手法との比較

続いて、提案手法と従来手法との比較を行う。4.5.4 節の実験より、提案手法のフィルタリングスコア算出法は maxsumBS が最も良いことが示されたので、ここでは maxsumBS と従来手法 (glide HTVS モード) を用いて 2%、5%、10% までフィルタリングを行い、フィルタリングを通過した化合物を通常のドッキングシミュレーション (glide SP モード) で再評価した場合の速度および精度の評価を行う。

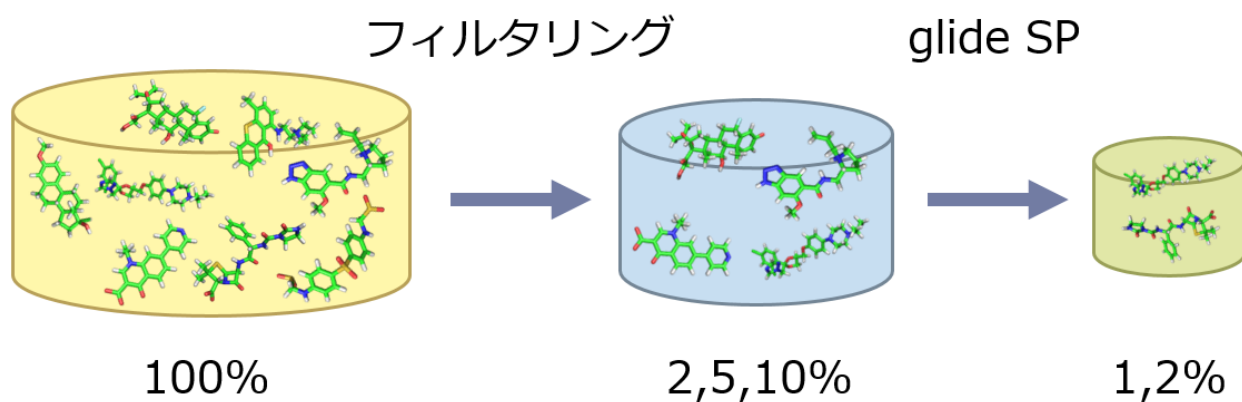


図 4.3 EF (1%)、EF (2%) 算出までの流れ

表 4.5 に結果を示す。節で示した単体での性能評価と同様に、glide HTVS モードをフィルタリングに用いる場合でも提案手法よりも精度が良くなっている。一方、計算速度について、フィルタリングで元の化合物群の 2% を通過させる場合、提案手法と通常ドッキング計算の合計必要時間が従来のフィルタリング手法である glide HTVS モードよりも少なくなっており、従来手法では達成できなかった速度での化合物の選別が可能になっていることがこの結果からわかる。この利点はフィルタを通過させる化合物の割合を高めるほど薄れて行く。これは、通常のドッキングシミュレーションの計算時間が支配的となるため、提案しているフィルタリング手法の計算時間的な利点が押しつぶされてしまうためである。

表 4.4 フィルタリング手法としての提案手法間の精度評価

フィルタリング 手法                      通過率		EF (1%)	EF (2%)	合計計算時間 [CPU sec.]
総和 (score_sum)	2%	6.84	–	6,396
最良値 (score_max)		8.59	–	
線形和 (maxsumBS)		<b>8.75</b>	–	
総和 (score_sum)	5%	9.61	5.92	13,481
最良値 (score_max)		10.36	6.99	
線形和 (maxsumBS)		<b>12.92</b>	<b>7.99</b>	
総和 (score_sum)	10%	12.41	7.67	25,289
最良値 (score_max)		11.58	7.92	
線形和 (maxsumBS)		<b>15.46</b>	<b>10.00</b>	

表 4.5 フィルタリング手法としての提案手法と従来手法の比較

フィルタリング 手法		通過率	EF (1%)	EF (2%)	合計計算時間 [CPU sec.]
提案手法 (maxsumBS)	2%		8.75	—	6,396
従来手法 (glide HTVS モード)			17.85	—	19,536
提案手法 (maxsumBS)	5%		12.92	7.99	13,481
従来手法 (glide HTVS モード)			18.97	12.50	26,621
提案手法 (maxsumBS)	10%		15.46	10.00	25,289
従来手法 (glide HTVS モード)			19.60	12.92	38,429
通常ドッキング (glide SP モード)			21.54	14.68	236,156



## 第5章

## 考察

### 5.1 提案手法の得手・不得手の調査

4.5.4 節の実験結果より、提案手法は従来手法に比べて平均的に見れば精度が低調に終わることが判明した。しかし、一部のターゲットに関しては提案手法が従来手法である glide HTVS モードに優っており（表 5.1）、この理由がわかればどのようなケースにおいて提案手法によるフィルタリングを用いるべきかを明示的にすることができる。同様に提案手法が従来手法より明らかに悪いケース（表 5.2）に関して原因が判明すれば、今後の提案手法の改善につながる。

表 5.1 提案手法が上手く行ったケース

提案手法（score\_sum、score\_max、maxsumBS のいずれか）が従来手法（glide HTVS モード）よりも ROC-AUC で 0.2 以上上回ったケースについて、ROC-AUC の差の降順で示している。

提案手法の種類	ターゲット名	ROC-AUC 差	ROC-AUC	
			従来手法	提案手法
線形和（maxsumBS）	mcr	0.319	0.466	<b>0.785</b>
線形和（maxsumBS）	akt1	0.285	0.539	<b>0.824</b>
最良値（score_max）	kith	0.272	0.615	<b>0.887</b>
最良値（score_max）	akt1	0.265	0.539	<b>0.804</b>
最良値（score_max）	mcr	0.257	0.466	<b>0.723</b>
線形和（maxsumBS）	gcr	0.252	0.528	<b>0.780</b>
最良値（score_max）	gcr	0.242	0.528	<b>0.770</b>
総和（score_sum）	ital	0.212	0.529	<b>0.741</b>
総和（score_sum）	akt1	0.209	0.539	<b>0.748</b>

表 5.2 提案手法が上手く行かないケース

提案手法 (score\_sum、score\_max、maxsumBS のいずれか) が従来手法 (glide HTVS モード) よりも ROC-AUC で -0.45 以上下回ったケースについて、ROC-AUC の差の昇順で示している。

提案手法の種類	ターゲット名	ROC-AUC 差	ROC-AUC	
			従来手法	提案手法
最良値 (score_max)	lkha4	-0.572	<b>0.880</b>	0.308
最良値 (score_max)	xiap	-0.506	<b>0.802</b>	0.296
総和 (score_sum)	def	-0.495	<b>0.733</b>	0.238
総和 (score_sum)	weel	-0.487	<b>0.933</b>	0.446
総和 (score_sum)	hs90a	-0.464	<b>0.761</b>	0.297
線形和 (maxsumBS)	hs90a	-0.460	<b>0.761</b>	0.301
線形和 (maxsumBS)	xiap	-0.460	<b>0.802</b>	0.342

### 5.1.1 提案手法が得意なターゲット

**@todo akt1, gcr, ital, kith, mcr の 5 種類について、ROC 曲線を示し、フラグメント数などからの説明を試みる。**

### 5.1.2 提案手法が不得意なターゲット

**@todo def, hs90a, lkha4, weel, xiap の 5 種類について、ROC 曲線を示し、フラグメント数などからの説明を試みる。タンパク質のポケットが大きいと苦手とか、そういう可能性はあるのかな ..?**

## 5.2 提案手法のユースケース

4.5.4 節の結果より、提案手法は精度より速度を重視したいケースにおいて有用であることは先に述べた。ここでは DUD-E のターゲットの内、総化合物数が 25,000 以上である 20 ターゲットを用いて、フィルタリングにおける化合物の通過率がより少ない場合や、さらに小さな割合における EF の評価を行い、大規模化合物ライブラリを利用する場合のユースケースを示す。**@todo score\_sum, score\_max, maxsumBS, glide HTVS を用いたフィルタリングで 0.5, 1, 2, 5% に化合物を削減した場合について、計算時間および EF0.1, 0.2, 0.5, 1, 2% を求め、結果を示し、この結果をもとに提案手法をどのように用いるのが有意義か、というユースケースを示す。**

## 第6章

# 結論

### 6.1 本研究の結論

本研究では、ドッキングに基づいた超高速なフィルタリング手法を提案した。この手法の予測精度はドッキングに基づいたフィルタリングの既存手法である glide HTVS モードに比べ劣っているが、計算速度は既存手法では実現不可能なほど高速であり、さらにデータセットが大きくなるほど相対的に速度が向上していく。**@todo 提案手法 (maxsumBS) と従来手法 (glide HTVS) についての速度と精度に関する簡単な表を作成**

また、フィルタリング後に行う通常のドッキングシミュレーションと組み合わせた場合の速度・精度の評価を行い、提案手法をフィルタリング手法として用いるべきユースケースを示した。**@todo 提案手法が有利になるケースに関する簡単な表を作成**

### 6.2 今後の課題

6.1 節で述べた通り、提案手法の精度は未だ不十分であるため、なるべく速度を維持しつつも精度を高める必要がある。また、本研究では最大でも数万化合物程度のデータセットを用いて評価を行ったが、数百万～数千万程度の、より大規模な化合物データセットを用いた速度評価を行う必要がある。

## 謝辞

ほげほげ。

## 参考文献

- [1] Komine Shunta, Ishida Takashi, and Akiyama Yutaka. 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, Vol. 2015-BIO-4, No. 62, pp. 1–8, 2015.
- [2] Richard a. Friesner, Jay L. Banks, Robert B. Murphy, Thomas a. Halgren, Jasna J. Klicic, Daniel T. Mainz, Matthew P. Repasky, Eric H. Knoll, Mee Shelley, Jason K. Perry, David E. Shaw, Perry Francis, and Peter S. Shenkin. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *Journal of Medicinal Chemistry*, Vol. 47, No. 7, pp. 1739–1749, mar 2004.
- [3] Tiejun Cheng, Qingliang Li, Zhigang Zhou, Yanli Wang, and Stephen H. Bryant. Structure-Based Virtual Screening for Drug Discovery: a Problem-Centric Review. *The AAPS Journal*, Vol. 14, No. 1, pp. 133–141, 2012.
- [4] Tiziano Tuccinardi, Maurizio Botta, Antonio Giordano, and Adriano Martinelli. Protein kinases: Docking and homology modeling reliability. *Journal of Chemical Information and Modeling*, Vol. 50, No. 8, pp. 1432–1441, 2010.
- [5] S Kannan and R Ganji. Porting Autodock to CUDA. *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pp. 1–8, 2010.
- [6] C A Lipinski, F Lombardo, B W Dominy, and P J Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, Vol. 23, No. 1-3, pp. 3–25, 1997.
- [7] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, Vol. 4, No. 2, pp. 90–8, 2012.
- [8] Iskander Yusof and Matthew D. Segall. Considering the impact drug-like properties have on the chance of success. *Drug Discovery Today*, Vol. 18, No. 13-14, pp. 659–666, 2013.
- [9] Noel M O’Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open Babel: An open chemical toolbox. *Journal of cheminformatics*, Vol. 3,

No. 1, p. 33, 2011.

- [10] Marcel L. Verdonk, Valerio Berdini, Michael J. Hartshorn, Wijnand T M Mooij, Christopher W. Murray, Richard D. Taylor, and Paul Watson. Virtual screening using protein-ligand docking: Avoiding artificial enrichment. *Journal of Chemical Information and Computer Sciences*, Vol. 44, No. 3, pp. 793–806, 2004.
- [11] Michael M. Mysinger, Michael Carchia, John J. Irwin, and Brian K. Shoichet. Directory of useful decoys, enhanced (DUD-E): Better ligands and decoys for better benchmarking. *Journal of Medicinal Chemistry*, Vol. 55, No. 14, pp. 6582–6594, 2012.

@todo mendeley の出力した bibtex そのままなので形式がまだバラバラ気味。出力情報を修正する必要あり。

## 付録 A

# DUD-E の詳細

表 A.1 DUD-E の詳細

ターゲット名	PDBID	タンパク質詳細	正例		負例	
			化合物数	平均分割数	化合物数	平均分割数
aa2ar kif11	3EML	Adenosine A2a receptor	nnn	nnn	nnn	nnn

## 付録 B

# ROC 曲線

DUD-E の 102 ターゲットそれぞれについて、7通りの手法の ROC 曲線を記載する。1つの figure に 7つの ROC 曲線が描かれるイメージで、それが 102 個並ぶ。 $3 \times 4 = 12$  が 1 ページで、それが 8 ページ半続く形になることを想像している。