

フラグメント分割に基づく 高速な化合物プレドッキング手法の開発

指導教員 秋山 泰 教授
計算工学専攻
14M38400 柳澤 溪甫

コンピュータによる予測を利用し、創薬コストの削減および創薬期間の短縮を行うバーチャルスクリーニングと呼ばれる手法が近年広く用いられている。その中でも、タンパク質や化合物の立体構造情報を用いたドッキングシミュレーションを行う手法（Structure-Based Virtual Screening, SBVS）は標的タンパク質に対する既知の薬剤・阻害剤がなくても行うことのできる手法であり、また既知の薬剤の情報を用いないために既知の薬剤と構造的に大きく異なる新たな化合物の発見につながるなど、創薬の現場で大きく期待されている手法である。

しかし一方で、ドッキングシミュレーションは標的タンパク質と化合物との最適な複合体構造を探索するが、相対的な3次元位置、回転および、化合物内部の結合の回転という多くの探索パラメータが存在するため計算コストが高く、データベース内の候補化合物をすべて評価するのが困難な場合が多い。この問題を解決するため、化合物を何らかの手法を用いてフィルタリングし、個数が減少した化合物サブセットについてドッキングシミュレーションを行う、というプロトコルが多々実践されている。ここで、構造ベースの化合物選別手法の利点である既知の薬剤・阻害剤の情報に依存しないこと、構造的に新規な化合物も発見できることの2点を損なわずにフィルタリングを行うことが重要となるが、従来の手法の多くはこの要件を達成することが出来ていないのが現状である。[@comment ここまでが長い。短くできないか？](#)

そこで本研究では、化合物の部分構造に着目することで、ドッキングに基づきつつ高速にフィルタリングを行う手法（プレドッキング）を提案する。この手法では化合物を内部に回転可能な結合を持たない部分構造（フラグメント）に分割し、標的タンパク質とフラグメントとの間でドッキングシミュレーションを行う。フラグメントへの分割によって化合物内部の回転をなくすことでドッキングシミュレーションの最適化問題の探索パラメータを

減少させることができ、従来のドッキングベースのフィルタリング手法である glide HTVS モードと比べて DUD-E を用いて平均して約9倍の高速化を達成した。この高速化効率化は化合物ライブラリ（データベース）のサイズが大きいほどさらに向上する傾向が示されている。また、フラグメントのドッキングシミュレーション結果のスコアからフィルタリングに用いる化合物のスコアへの算出方法を複数検討し、フィルタリング後の化合物ベースのドッキング計算も含めて評価した場合に従来手法に比べて精度が最大約 25%・速度が最大約 40%改善されるケースが存在することを示した。 [@memo \(1050 文字程度\)](#)

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	3
2.1	SBVS (Structure-Based Virtual Screening) 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	6
2.3	化合物のフィルタリング	7
2.3.1	既存のフィルタリング手法	8
2.3.2	既存手法の問題点	8
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（プレドッキング）手法の開発	10
3.1	提案手法の概説	10
3.1.1	フィルタリングの要件	10
3.1.2	提案手法へのアイデア	11
3.2	提案手法の詳細の説明	11
3.2.1	提案手法のフローチャート	12
3.2.2	化合物のフラグメントへの分割	12
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	15
3.2.4	化合物のフィルタリングスコアの算出	15
第4章	実験	18
4.1	データセット	18
4.2	予測精度の評価指標	18
4.3	計算環境	19

4.4	比較対象	21
4.5	評価実験	21
4.5.1	フラグメント分割	21
4.5.2	ドッキング速度の評価	21
4.5.3	予測精度の評価	22
4.5.4	フィルタリング手法としての性能評価実験	23
第 5 章	考察	26
5.1	総和法におけるフラグメント数に対するペナルティ	26
5.2	提案手法が得意とするケースの調査	29
5.3	提案手法の利用例	30
第 6 章	結論	34
6.1	本研究の結論	34
6.2	今後の課題	35
	謝辞	36
	参考文献	37
付録 A	DUD-E の詳細	40
付録 B	各手法を単体で用いた場合の ROC 曲線	41

目 次

2.1	ドッキングシミュレーションのイメージ	4
2.2	化合物の内部自由度	5
2.3	Glide のワークフロー ¹⁾	6
2.4	eHiTS のクリーク探索	7
2.5	AutoDock の GA で用いる変数群	7
3.1	フラグメント単位でのドッキング結果例	12
3.2	提案手法の手順	13
3.3	化合物のフラグメント分割アルゴリズム ²⁾	14
3.4	ZINC ”drugs now” 10,639,555 化合物を分割した例	15
3.5	複数のドッキング結果の出力例および最良構造の選択	16
3.6	maxsumBS の算出	17
4.1	ROC-AUC 計算例	20
4.2	EF 計算例	20
4.3	DUD-E ターゲットにおける化合物数とフラグメント種類数の関係	21
4.4	EF (1%)、EF (2%) 算出までの流れ	24
5.1	ターゲット fnta の全ての化合物のうち重原子数 32 の化合物の単純加算スコア	26
5.2	ホルモテロール	30
5.3	カンデサルタン	30
B.1	ROC 曲線例	42

表 目 次

2.1	ドッキングシミュレーションソフトウェアの計算速度	7
4.1	DUD-E のターゲットの化合物	18
4.2	利用した計算環境	19
4.3	フラグメント 1 種類あたりの化合物数	22
4.4	ドッキング計算時間の比較	22
4.5	提案手法の予測精度	23
4.6	フィルタリング手法としての提案手法間の精度評価	25
4.7	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	25
5.1	小さなフラグメントを無視することによる score_sum の精度の変化	27
5.2	フラグメント数に対する線形ペナルティによる score_sum の精度の変化	27
5.3	小さなフラグメントを無視することによる maxsumBS の精度の変化	28
5.4	フラグメント数に対する線形ペナルティによる maxsumBS の精度の変化	28
5.5	提案手法が上手く行ったケース	29
5.6	提案手法が得意なターゲットの性質	29
5.7	総化合物数が 1 万以上存在する DUD-E のターゲットに対する評価実験	31
5.8	化合物全体を評価するのに要する時間の比較	32
5.9	提案手法が従来手法に速度・精度ともに勝る例	33
6.1	提案手法の性能	34
6.2	通常ドッキング (glide SP) と組み合わせた速度・精度評価	34
A.1	DUD-E の詳細	40

第1章

序論

1.1 研究背景

近年、創薬の初期段階において、コンピュータによる予測を通して大量の化合物から薬剤候補化合物を選別するバーチャルスクリーニング (Virtual Screening, VS) と呼ばれる手法が用いられ、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。

1. 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)
 - タンパク質-化合物ドッキングシミュレーション^{1),3),4)}
2. 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)
 - 構造活性相関 (Quantitative Structure-Activity Relationship, QSAR) を用いた手法⁵⁾
 - 機械学習による分類手法⁶⁾
 - 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⁷⁾
3. 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)⁸⁾

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が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LBVS等と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,¹⁾ eHiTS,³⁾ Autodock⁴⁾ 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⁹⁾、比較的広く利用されている¹⁰⁾。

一方、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という問題点が存在する。これを解決するためにドッキング手法の高速化の研究^{11)–13)}が行われているが、速度の点で未だ不十分であり、例えば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 ZINC¹⁴⁾ に存在する 22,724,825 件の化合物を一斉にドッキングシミュレーションをしようとする 12 コアの計算機で半年以上を要してしまうのが現状である。

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^{15),16)}。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¹⁷⁾、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。[@comment このあたりの文章を図表にまとめられると良い](#)

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

第 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。第 3 章では提案手法について述べ、第 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、第 5 章では第 4 章で行った実験の結果についての考察を加え、第 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

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 (Structure-Based Virtual Screening) 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

- 既知のリガンド情報を必要とせず
- 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

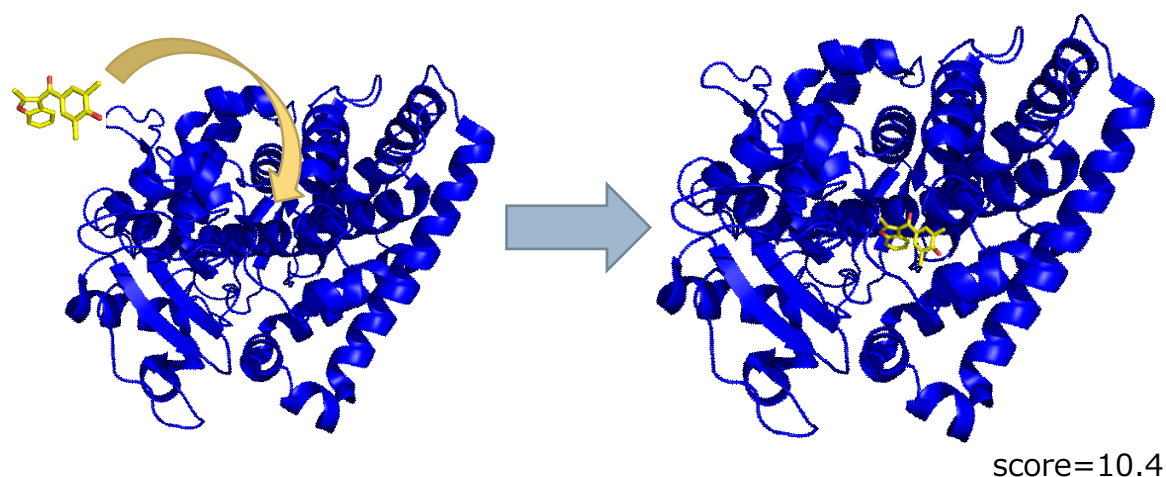


図 2.1 ドッキングシミュレーションのイメージ

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 2.1）。この得られたスコアを直接、あるいは何らかの形で変換を行った評価値を用いて複数の化合物の選別を行う。

この化合物-タンパク質ドッキングシミュレーションを行うツールは Glide,¹⁾ eHiTS³⁾ などの有償ソフトウェア、AutoDock⁴⁾ 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開発されている。[@comment 関連ソフトの「名称・作者・論文」などの一覧表をどこかにのせておきたい。](#)

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 つが非常に重要であり、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 DOCK¹⁸⁾ 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。

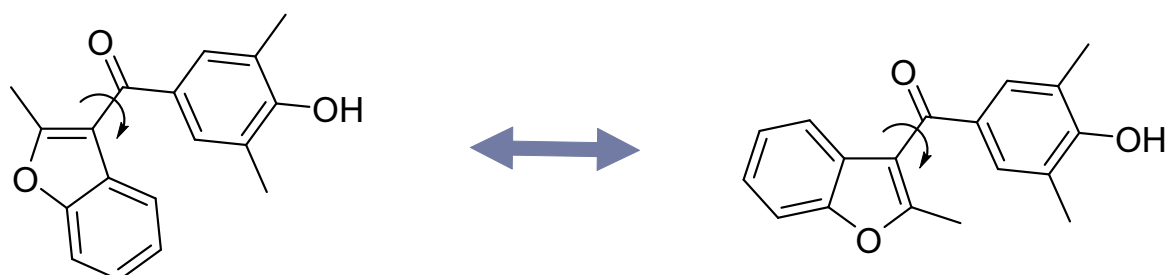


図 2.2 化合物の内部自由度

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。[@comment 図がほしい](#)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2.2)。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4.61であり、これが計算量に大きな影響を及ぼす。

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

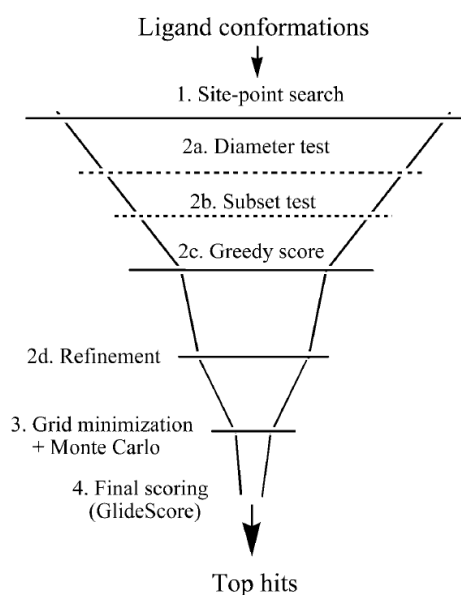
段階的な全探索を行うことで局所最適解を得る。具体的には、最初の段階では化合物を球体に近似しての位置が良いかどうかの見積もりから始め、徐々に化合物の近似を厳密なものにしていく。それぞれの段階で上位の位置・構造のみを残し次の段階へ進めることで、全探索の空間を現実的な量に制限し、探索を完了させる(図2.3)。

- 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(図2.4)。

- 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(図2.5)。

図 2.3 Glide のワークフロー¹⁾

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。[@comment](#) これに関しても Chang が示しているような Force field, Empirical, Knowledge based の分類を表で示す。表にまとめることが大切。

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¹⁾、eHiTS は最速で数秒³⁾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ると 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り現実的ではない。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¹⁹⁾。AutoDock はオープンソー

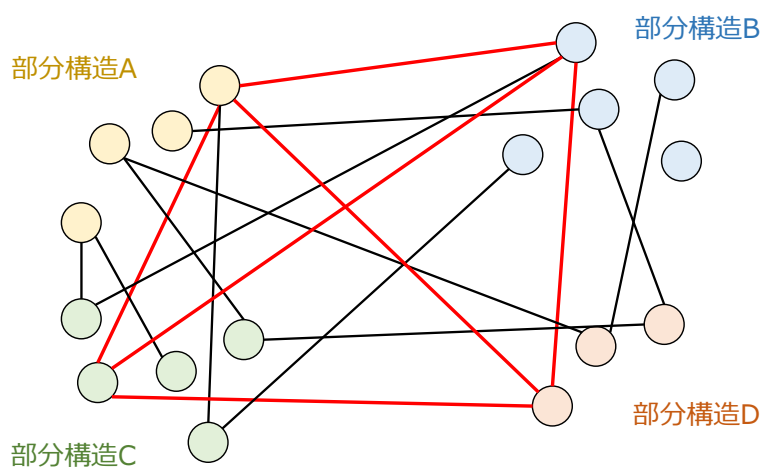


図 2.4 eHiTS のクリーク探索

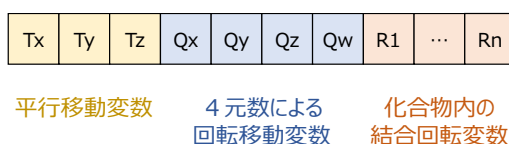


図 2.5 AutoDock の GA で用いる変数群

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¹¹⁾、Glide に及ばない。

表 2.1 ドッキングシミュレーションソフトウェアの計算速度

ソフトウェア名	有償/無償	1 CPU での計算速度
Glide	有償	10-150 sec/compound ¹⁾
eHiTS	有償	<10 sec/compound (最速) ³⁾
AutoDock	無償	Glide の約 250 倍の計算時間 ¹⁹⁾

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 万個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 1. 化合物の物理的特徴に基づくフィルタリング、2. 化合物の構造に基づくフィルタリング、3. 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。

1. 化合物の物理的特徴に基づくフィルタリング

化合物の分子量や水溶性か油溶性かを示す分配係数 (LogP) などの値は創薬において有用な情報である。これらの物理化学的な値を用いて、経口薬として優れた薬物の特徴を 4 つの法則にまとめたリピンスキーの法則²⁰⁾や、これを発展させ既存の薬剤の物理化学的な値からヒストグラムを作成し化合物の薬物らしさ (Druglikeness) のスコアを付ける QED²¹⁾などのフィルタリング手法が存在している。

2. 化合物の構造に基づくフィルタリング

化合物の物理的特徴は物質を巨視的に見ることで得られるパラメータであるが、本来タンパク質の活動の障害はタンパク質や化合物 1 分子単位の非常に微視的なメカニズムによって発生しており、したがって化合物の分子構造はタンパク質との複合体を形成する上で非常に重要な情報である。一般に構造が似ている化合物は同じタンパク質との複合体を形成することが多いため、化合物の分子構造式を数百～数千のあらかじめ定めた局所構造が存在するか否かのバイナリである fingerprint に落とし、これが既知の薬剤やタンパク質の阻害剤にどれほど近いのか、という情報を用いたフィルタリング手法が存在する¹⁵⁾。また、化合物の分子構造式のみでなく、化合物の立体構造を用いて化合物の類似性を評価するファーマコフォアモデリングと呼ばれる手法も存在する¹⁶⁾。

3. ドッキングシミュレーションベースのフィルタリング

2.2.2 節で述べた通り、ドッキングシミュレーションは一般的に計算コストが高くフィルタリングには適していないが、Glide には化合物の構造について強い仮定を置くことで計算を簡易化し、通常ドッキングモード (SP モード) の 10 倍程度の速度²²⁾で計算を完了させる高速ドッキングモード (HTVS モード) が存在する。このモードをフィルタリングとして利用し、フィルタリング後の化合物群に対して SP モードによるドッキングシミュレーションを行うという手法が用いられることがある¹⁷⁾。

@memo これも文章ではなく表にまとめられないか？

2.3.2 既存手法の問題点

2.3.1 節で述べたように、既存のフィルタリング手法は多く存在するものの、以下の 2 点からこれらの手法は改善する余地が残されている。

- 化合物の物理的特徴や化合物の構造に基づくフィルタリング手法は帰納的な手法であり、標的タンパク質を狙った既知の薬剤や阻害剤が必須となる。さらに既知の薬剤や阻害剤を利用できたとしても、化合物の類似性を利用する手法であるために既知の化合物に似てしまうという問題がある。
- Glide の高速ドッキングモードはドッキングシミュレーションとしては高速であるが、それでも 1 化合物 1 秒程度を要する。1,000 万件の化合物のフィルタリングを行う場合 1 CPU 利用で 4 か月程度の期間を要してしまうため、この速度は十分とは言えない。

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（プレドッキング）手法の開発

ここでは本研究で新たに提案する、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（プレドッキング）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するが、その理由は探索空間の広さとドッキング計算を行うべき化合物の数の多さにある。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。[@comment](#) どのくらい高速であれば十分なのか？という例示はあるのか？
- 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予測精度が保持されていること、特に偽負例（False Negative）を出さないように弁別することが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複合体構造を予測する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、必ずしもタンパク質と化合物との複合体構造を出力する必要はなく、偽正例 (False Positive) を発生させることもある程度は許容される。

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

2.2.1 節で述べたように、化合物の内部自由度が及ぼす計算量への影響は大きい。そのため、eHiTS³⁾ や FlexX²³⁾ など一部のドッキングシミュレーションツールでは化合物を内部自由度のより少ない部分構造に分割し、タンパク質と部分構造との結合能力を評価しつつ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。本提案手法では、小峰ら²⁾ 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

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、元の化合物の構造に衝突などの問題が発生しないように部分構造を配置してタンパク質と化合物との複合体構造を形成する。しかし部分構造同士の衝突などの考慮を行うと、単純なアルゴリズムでは計算量が $O(a^n)$ (n は化合物を構成する部分構造数) となってしまう、近似アルゴリズムを用いたとしても時間を要してしまう。

一方本研究で提案するプレドッキング手法は、その後に複合体構造を予測する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タンパク質と化合物との正しい複合体構造を必ずしも出力する必要はない。そこで提案手法ではフラグメントに分割する前の化合物の構造に関する考慮を行わないことにした。こうすることで、図 3.1 のようにフラグメント同士の衝突が許容されてしまうが、化合物のスコアをフラグメント数の線形オーダー $O(n)$ で算出することが可能になり、高速な化合物の評価ができるようになる。

3.2 提案手法の詳細の説明

前節で本研究で用いる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。

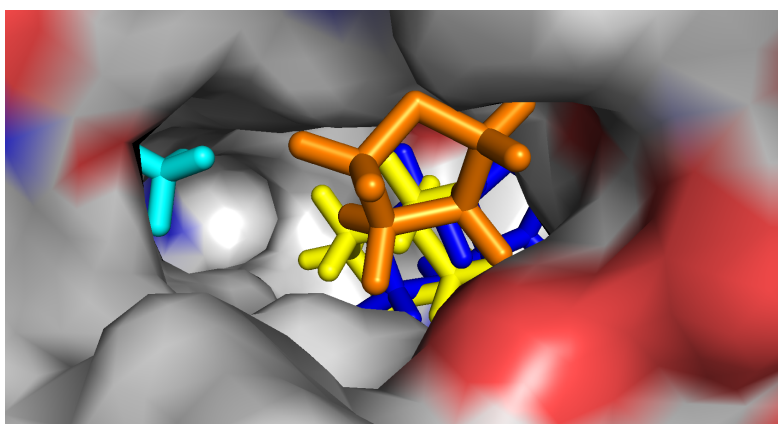


図 3.1 フラグメント単位でのドッキング結果例

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. 入力化合物群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 [@memo](#) すべて「プレドッキングスコア」に名称変更したほうがわかりやすいのでは? の上位 N% をフィルタを通過した化合物として出力する

ワークフローを図 3.2 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²⁾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 OpenBabel²⁴⁾ 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 3.3 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 3.3 左から 2 番目）
2. 回転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。

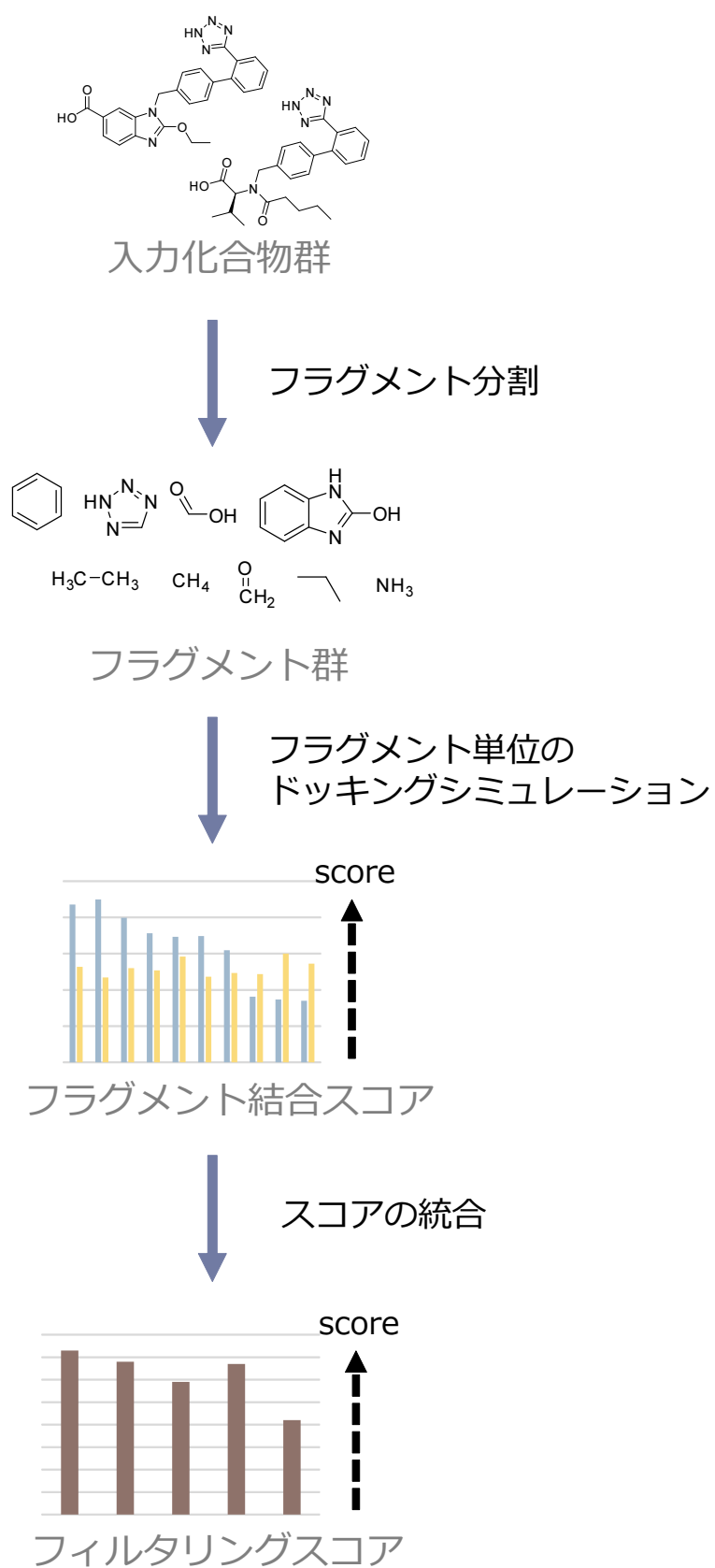


図 3.2 提案手法の手順

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。(図 3.3 左から 3 番目)
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。
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。(図 3.3 左から 4 番目)
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。
7. 切断面に水素原子を付加する。

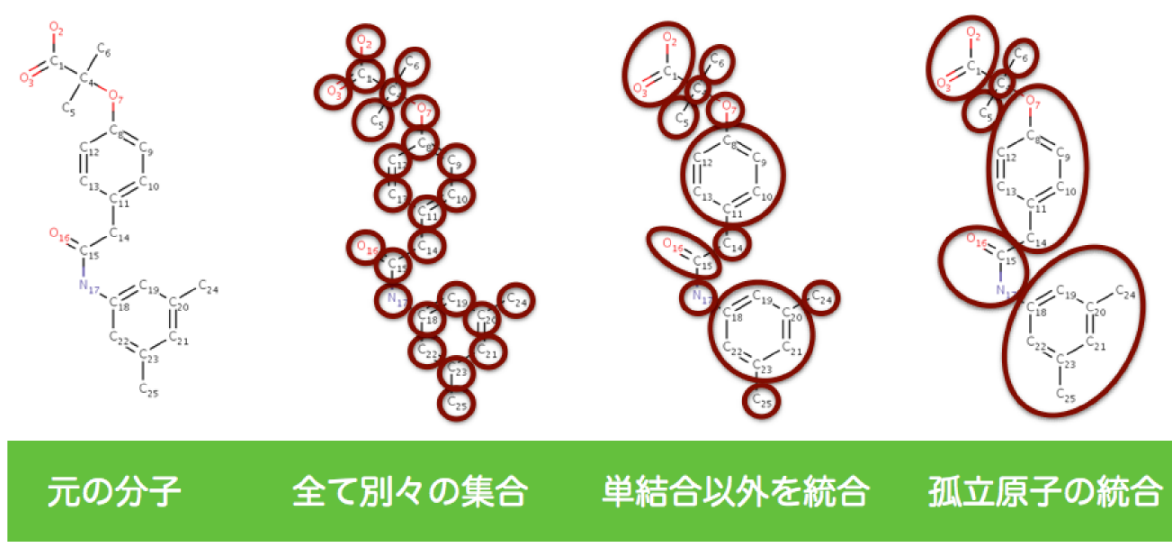


図 3.3 化合物のフラグメント分割アルゴリズム²⁾

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 3.4 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。このフラグメント種類数の増加は化合物数の増加に比べて緩や

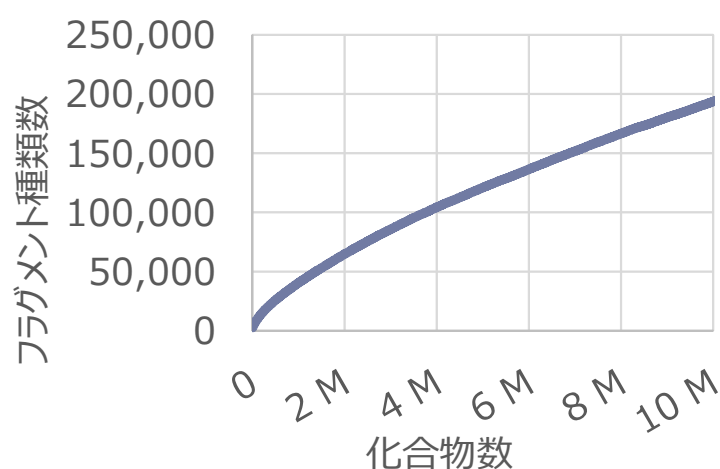


図 3.4 ZINC "drugs now" 10,639,555 化合物を分割した例

かであり、化合物数が多ければ多いほどフラグメント分割による速度への貢献が大きいと考えられる。

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 glide¹⁾ 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 HTVS モードと SP モードを利用した場合の評価を行う。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 3.5)。

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。

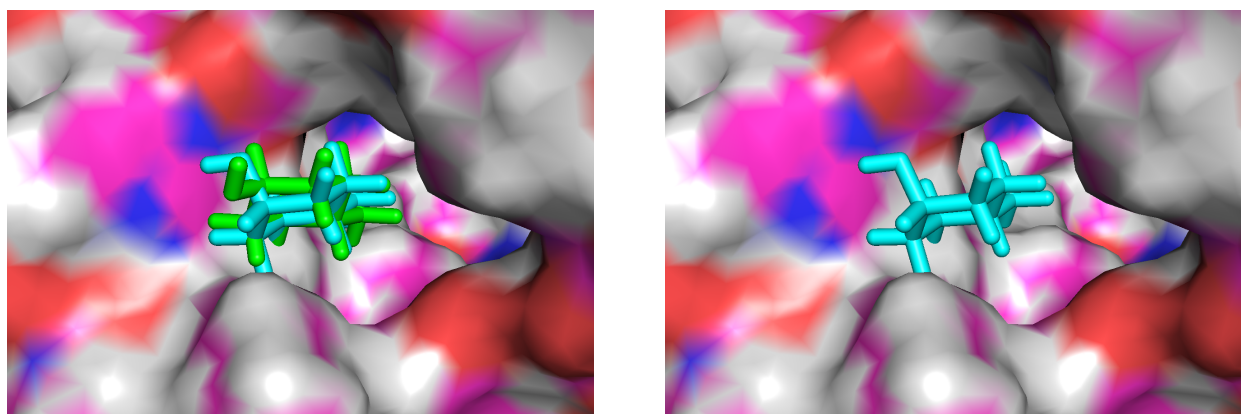


図 3.5 複数のドッキング結果の出力例および最良構造の選択

1. 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。全てのフラグメントが高いスコアでタンパク質と結合できる化合物の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り、特に分割数が多ければ多いほどこの傾向は顕著になる。このため、重原子（水素以外の原子）の数が2個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することでフィルタリングスコアの無意味な向上を抑えている。

$$SCORE_{\text{化合物}} = \sum_{\substack{\text{重原子数} > 2 \text{ の} \\ \text{フラグメント}}} SCORE_{\text{フラグメント}} \quad (3.1)$$

2. 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。1つでもタンパク質との結合スコアが非常に良いフラグメントを持っている化合物の評価が高くなる手法である。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²⁵⁾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。

$$SCORE_{\text{化合物}} = \max_{\substack{\text{すべての} \\ \text{フラグメント}}} SCORE_{\text{フラグメント}} \quad (3.2)$$

3. 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に単純和で

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3.6）。この手法は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アであるので、maxsumBS（max-sum Balanced Score）として以下記述する。

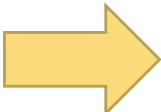
	総和法	最良値法		総和法 z	最良値法 z	和
A	20	8	$z_{score} = \frac{x - \mu}{\sigma}$ 	1.225	0.267	1.492
B	15	4		0	-1.336	-1.336
C	10	10		-1.225	1.069	-0.156

図 3.6 maxsumBS の算出

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)²⁶⁾ 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、平均値を示す。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。

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

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

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。

4.3 計算環境

本研究では、東京工業大学のスーパーコンピュータである TSUBAME2.5 の Thin ノードを利用した。利用した計算環境を表 4.2 に示す。

表 4.2 利用した計算環境

CPU	Intel Xeon X5670, 2.93 GHz (6 cores) ×2
Memory	54 GB RAM

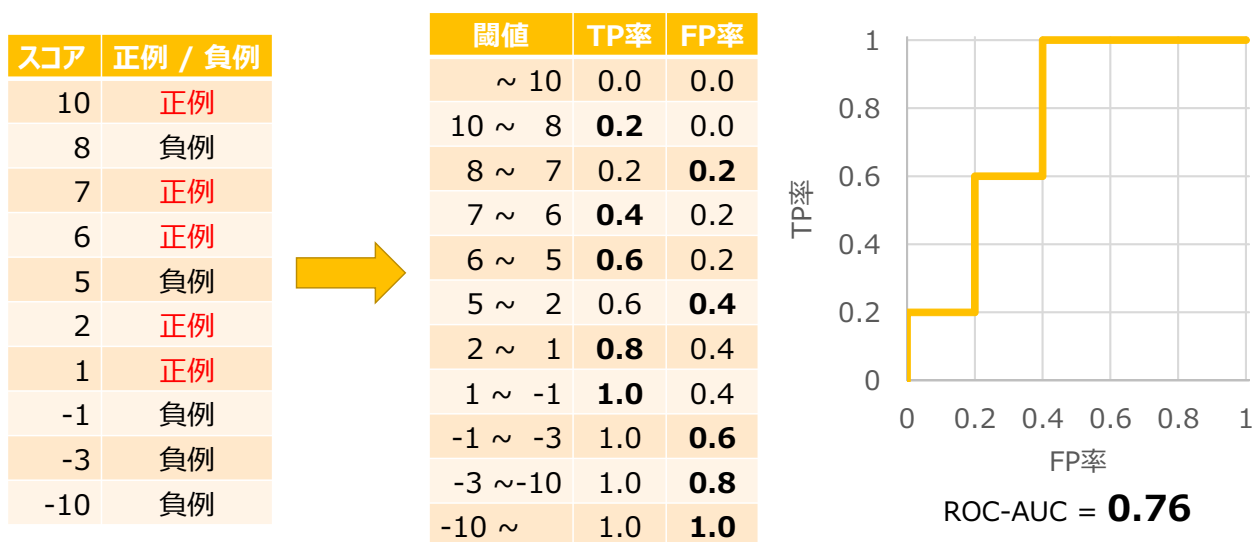


図 4.1 ROC-AUC 計算例

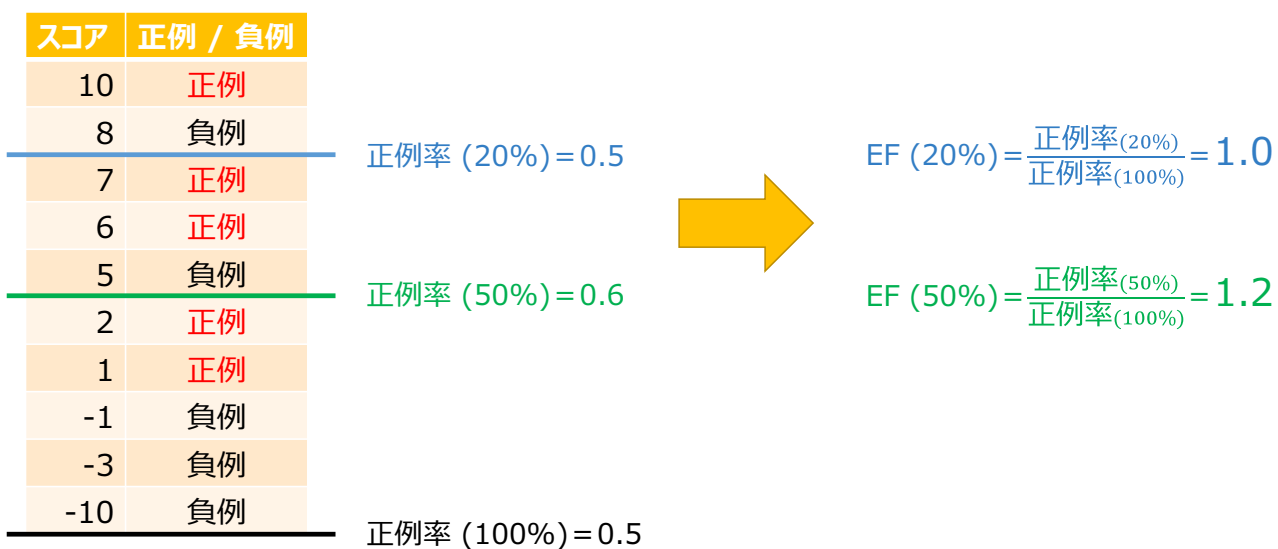


図 4.2 EF 計算例

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

まず、今回用いる複数のターゲットについて、フラグメント分割を行うことでドッキングの必要数をどの程度減らせるのかを示す。それぞれターゲットにフラグメント分割を適用した場合における化合物数とフラグメント種類数の推移は図 4.3 の通りとなり、DUD-E ターゲット全体で平均するとフラグメント種類数は化合物数の約 4 分の 1 に抑えられている。化合物数が多いほど化合物数に対するフラグメント種類数が抑えられる傾向にあることも確認された（表 4.3）。

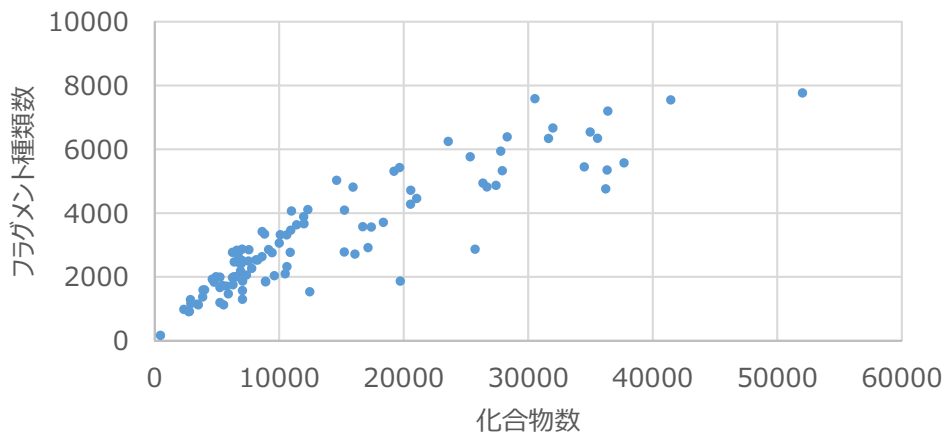


図 4.3 DUD-E ターゲットにおける化合物数とフラグメント種類数の関係

@comment フラグメント種類数：化合物数 = 1 : 4 の線分を補助線として引く

4.5.2 ドッキング速度の評価

つづいて、フィルタリング手法の計算速度を評価する。4.5.1 で述べたように、一つのターゲットに含まれる化合物数が多ければ多いほどフラグメント数は相対的に少なくなり提案手法の計算コストの削減が増幅されるため、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

表 4.3 フラグメント 1 種類あたりの化合物数

	フラグメント 1 種類あたりの化合物数	
	ターゲット数	フラグメント 1 種類あたりの化合物数
全ターゲット	102	4.00
化合物数 1 万未満のターゲット	53	3.17
化合物数 1 万以上のターゲット	49	4.91

数が最小であるターゲット fgfr1、総化合物数が平均値近いターゲット adrb2、総化合物数が最大であるターゲット fnta の 3 種類について独立して結果を示す。

結果は表 4.4 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。

表 4.4 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(x1.0)	164(x3.5)	140(x4.0)
adrb2	15,224	2,779	338,511	17,043(x1.0)	1,481(x11.5)	899(x19.0)
fnta	52,022	7,767	1,770,967	98,665(x1.0)	4,149(x24.0)	2,549(x38.7)
全ての平均	13,881	3,231	236,156	14,813(x1.0)	1,673(x8.9)	987(x15.0)

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。

結果は表 4.5 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高速性を重視した本研究の提案手法よりも勝っていることが分かる。

また、提案手法間の比較を行うことで以下のことが言える。

- タンパク質とフラグメントとのドッキングに glide SP モードを利用した方がどの手法を用いた場合についてもほぼすべての評価指標で予測精度が良くなる。
- ROC-AUC は maxsumBS が他の 2 つの提案手法に比べて良い結果が出ているが、EF (1%) や EF (2%) に関しては score_max が maxsumBS を上回っている。

表 4.5 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	0.618	4.84	3.97	2.99	2.29
最良値 (score_max)	glide SP モード	0.637	6.78	5.65	3.81	2.60
	glide HTVS モード	0.627	6.94	5.55	3.32	2.55
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)		0.705	16.67	11.18	6.38	4.11

4.5.2 で述べたように提案手法の速度は 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。そのため、予測精度を高めるべく以下の実験では glide SP モードを用いることとする。 [@memo 日本語あやしい？](#)

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 4.5.3 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。組み合わせを通した評価は

1. フィルタリング手法で 2%, 5%, 10% まで化合物を削減
2. 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) で再計算
3. 再計算の結果の上位 1% および上位 2% の濃縮率 (EF (1%), EF (2%)) を評価

という手順を用いる。なおフィルタリング手法を用いて 2% まで削減した場合、EF (2%) はフィルタリング手法単体の性能と変わらなくなるため、「-」と表記する。

提案手法間の精度比較

まず提案手法間の精度比較を行い、化合物フィルタリングスコアの算出方法を検討した。結果は表 4.6 のようになり、多くの場合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。

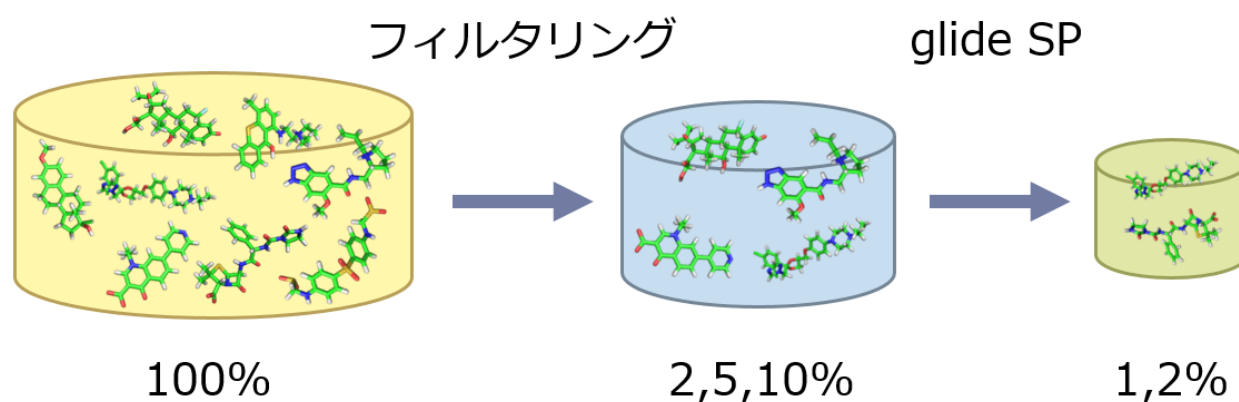


図 4.4 EF (1%)、EF (2%) 算出までの流れ

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) を用いて速度および精度の評価を行う。

表 4.7 の結果より、以下のことが言える。

- 4.5.3 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 4.6 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)		9.09	—	
線形和 (maxsumBS)		8.75	—	
総和 (score_sum)	5%	9.61	5.92	13,481
最良値 (score_max)		10.93	7.49	
線形和 (maxsumBS)		12.92	7.99	
総和 (score_sum)	10%	12.41	7.67	25,289
最良値 (score_max)		11.85	8.24	
線形和 (maxsumBS)		15.45	10.00	

表 4.7 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 総和法におけるフラグメント数に対するペナルティ

もし、フラグメントの結合スコアを単純に全て加算し、それを化合物のフィルタリングスコアとすると、図5.1のように化合物の総原子数が同じであっても分割数が多いほどフィルタリングスコアが向上してしまう。この分割数と総和法のスコアとの相関は最適化問題の条件緩和と考えることで説明できる。すなわち、本来化合物には原子間の結合距離という拘束条件が存在している。フラグメント分割によって切断された原子間の結合は距離を考えずにスコア付けして良いので、分割は原子間の結合という拘束条件を一つずつ緩和することに対応する。このため、フラグメント分割がされればされるほどスコアが良くなってしまふのである。

このような現象を改善するための手法として、以下2つの実験を行った。

1. 小さなフラグメントの無視

重原子（水素以外の原子）の個数に閾値を設け、その閾値を超えているフラグメントの結合スコアのみを総和に用いる。分割が多ければ多いほど小さなフラグメントが発生するため、

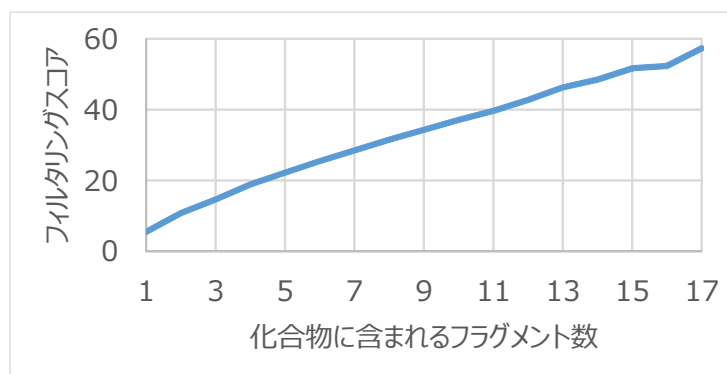


図 5.1 ターゲット fnta の全ての化合物のうち重原子数 32 の化合物の単純加算スコア

小さなフラグメントの結合スコアを無視することで事実上のフラグメント数に対するペナルティとなる。

2. フラグメント数に対する線形ペナルティ

全てのフラグメントの結合スコアを加算した後、化合物が持つフラグメントの個数に応じたペナルティを付与する。図 5.1 を見ると、フィルタリングスコアの平均とフラグメント数との関係は線形に近く、フラグメント数に対して線形なペナルティを課すことでフラグメント数に依存しないフィルタリングスコアとなることが想定される。

この2つの手法を個別に利用した場合の総和法 (score_sum) の精度は表 5.1 および表 5.2 のようになり、重原子数3以下のフラグメントの結合スコアを無視することが最も精度を高めている。

表 5.1 小さなフラグメントを無視することによる score_sum の精度の変化

無視するフラグメントのサイズ	ROC-AUC	Enrichment Factor			
		EF(1%)	EF(2%)	EF(5%)	EF(10%)
全てのフラグメントを利用	0.545	3.46	2.76	2.01	1.63
重原子数1	0.557	2.38	2.16	1.85	1.66
重原子数2以下	0.624	5.08	4.14	3.02	2.34
重原子数3以下	0.634	5.75	4.34	3.03	2.49
重原子数4以下	0.620	4.27	3.43	2.79	2.32
重原子数5以下	0.614	4.43	3.68	2.75	2.13
重原子数6以下	0.537	2.20	1.86	1.53	1.43

表 5.2 フラグメント数に対する線形ペナルティによる score_sum の精度の変化

フラグメント1つあたりの ペナルティ c	ROC-AUC	Enrichment Factor			
		EF(1%)	EF(2%)	EF(5%)	EF(10%)
ペナルティなし	0.545	3.46	2.76	2.01	1.63
$c = 1$	0.559	3.81	2.98	2.18	1.77
$c = 2$	0.586	4.70	3.65	2.66	2.08
$c = 3$	0.622	5.03	4.03	2.86	2.32
$c = 4$	0.588	3.80	3.19	2.51	2.14
$c = 5$	0.549	3.57	2.96	2.20	1.78
$c = 6$	0.530	3.30	2.53	1.91	1.57
$c = 7$	0.520	3.06	2.29	1.72	1.46

一方、同様に総和法のペナルティを変化させながら総和法と最良値法の線形和（maxsumBS）の精度について実験を行うと、重原子数2以下のフラグメントの結合スコアを無視した総和法を用いた場合に最良のROC-AUCとなった（表5.3、表5.4）。maxsumBSの精度はscore_sumよりも良いことから、本研究の提案手法では重原子数2以下のフラグメントの結合スコアを無視した総和法を利用する。

表 5.3 小さなフラグメントを無視することによる maxsumBS の精度の変化

無視するフラグメントのサイズ	ROC-AUC	Enrichment Factor			
		EF(1%)	EF(2%)	EF(5%)	EF(10%)
全てのフラグメントを利用	0.652	5.35	4.56	3.32	2.60
重原子数1	0.652	4.67	4.18	3.25	2.56
重原子数2以下	0.679	6.03	5.03	3.96	3.00
重原子数3以下	0.672	5.57	4.79	3.78	2.85
重原子数4以下	0.653	4.89	4.32	3.46	2.67
重原子数5以下	0.643	4.95	4.28	3.29	2.55
重原子数6以下	0.566	2.76	2.46	2.03	1.79

表 5.4 フラグメント数に対する線形ペナルティによる maxsumBS の精度の変化

フラグメント1つあたりの ペナルティ c	ROC-AUC	Enrichment Factor			
		EF(1%)	EF(2%)	EF(5%)	EF(10%)
ペナルティなし	0.652	5.35	4.56	3.32	2.60
$c = 1$	0.657	5.67	4.78	3.40	2.67
$c = 2$	0.665	6.40	5.02	3.59	2.80
$c = 3$	0.665	5.97	4.84	3.78	2.88
$c = 4$	0.630	6.16	4.69	3.41	2.66
$c = 5$	0.609	6.49	4.71	3.17	2.45
$c = 6$	0.600	6.49	4.69	3.11	2.36
$c = 7$	0.591	6.43	4.62	3.06	2.32

5.2 提案手法が得意とするケースの調査

4.5.3 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ている。しかし、本研究で用いた 102 ターゲット中 46 ターゲットに関しては提案手法が従来手法である glide HTVS モードよりも精度が良く、ROC-AUC で 0.2 以上上回っているケースも表 5.5 に示す通り 3 例存在している。

どのような場合において提案手法が有用であるかを調べるため、この 3 つのターゲットについて化合物の持つフラグメント数の平均、小さなフラグメントを削減した後のフラグメント数の平均、sitemap²⁷⁾ によって計算された各タンパク質の結合部位のサイズを求めた。その結果、結合部位のサイズやデータセット全体を通してのフラグメント数などに傾向は見受けられなかったが、化合物の持つフラグメント数の平均、そのうち重原子数が 2 以下のフラグメント数の平均どちらも正例より負例が上回っているということが判明した（表 5.6）。

表 5.5 提案手法が上手く行ったケース

提案手法（maxsumBS）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	0.785
線形和（maxsumBS）	akt1	0.285	0.539	0.824
線形和（maxsumBS）	gcr	0.252	0.528	0.780

表 5.6 提案手法が得意なターゲットの性質

ターゲット	表 3.16 提案手法が得意なターゲットの性質						結合部位のサイズ [Å ³]
	フラグメント数の平均			重原子数 2 以下の フラグメント数の平均			
	全体	正例	負例	全体	正例	負例	
akt1	7.98	6.94	8.00	4.64	3.08	4.66	637
gcr	5.93	5.33	5.94	2.68	2.00	2.69	471
mcr	5.90	5.43	5.91	2.67	2.06	2.68	179
全 102 ターゲット平均	7.22	7.43	7.21	3.83	3.78	3.83	437

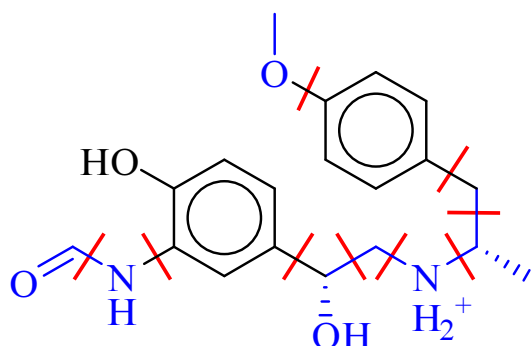


図 5.2 ホルモテロール

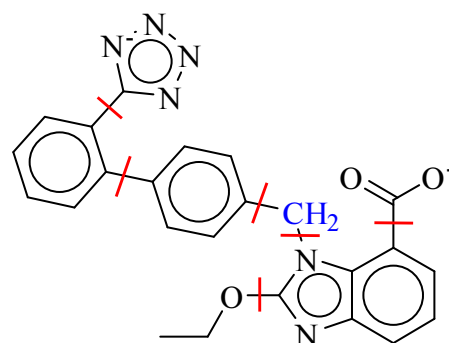


図 5.3 カンデサルタン

どちらも切断面を示し、重原子2以下のフラグメントになってしまう部分を青く表示している

例えば、ホルモテロール（図5.2）とカンデサルタン（図5.3）はそれぞれ薬剤として認められている化合物だが、このうち後者のような化合物が結合するタンパク質を対象としたフィルタリングには提案手法を、前者のような化合物が結合するタンパク質を対象としたフィルタリングには従来手法を用いると良いことが推定される。

一方この性質は5.1節で示したペナルティの影響を受けていると考えられるため、化合物の構造の有利不利が発生しないようなスコア計算手法およびペナルティの考案を引き続き行う必要がある。

5.3 提案手法の利用例

バーチャルスクリーニングでは数百万化合物から数百化合物程度を選別することが多く^{28),29)}、上位0.01%など、非常に小さな比率における Enrichment Factor の計算などが本来必要となる。また、計算時間に関しても数百万化合物を用いた場合に何日を要するのか、という評価が必要である。

しかし本研究で評価に利用したデータセットである DUD-E は表4.1で示したように472化合物しか存在しないターゲットも存在しており、このようなターゲットは上位0.01%を計算することは不可能である。そこで、ここではDUD-Eのターゲットのうち総化合物数が10,000以上である49ターゲットを用いることで、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、EF (0.1%)などの小さな割合におけるEFの評価（表5.7）を用い、実際のバーチャルスクリーニングでの提案手法の利用例を示す。計算速度に関しては線形に計算量が増大すると仮定することで数百万化合物を評価した場合の計算時間を見積もる。

なお、総化合物数が10,000以上である49ターゲットの平均化合物数は22,259、平均フラグメント種類数は4,588である。

表 5.7 総化合物数が1万以上存在するDUD-Eのターゲットに対する評価実験

フィルタリング 手法	通過率	EF (0.1%) EF (0.2%) EF (0.5%) EF (1%) EF (2%)					合計計算時間 [CPU sec.]
		EF (0.1%)	EF (0.2%)	EF (0.5%)	EF (1%)	EF (2%)	
提案手法 (maxsumBS)	0.5%	14.33	9.39	—	—	—	3,280
従来手法 (glide HTVS モード)		35.16	29.97	—	—	—	25,452
提案手法 (maxsumBS)	1%	20.26	13.91	7.14	—	—	5,180
従来手法 (glide HTVS モード)		35.36	31.17	22.19	—	—	27,352
提案手法 (maxsumBS)	2%	24.87	19.14	10.57	6.32	—	8,979
従来手法 (glide HTVS モード)		35.54	31.69	23.10	15.50	—	31,151
提案手法 (maxsumBS)	5%	29.24	25.04	16.80	10.51	6.29	20,378
従来手法 (glide HTVS モード)		35.54	31.40	23.56	16.28	10.59	42,550
提案手法 (maxsumBS)	10%	31.94	27.48	19.68	13.58	8.36	39,377
従来手法 (glide HTVS モード)		35.70	31.78	23.80	16.89	11.07	61,549
通常ドッキング (glide SP モード)		35.98	32.82	25.57	18.96	12.82	379,965

超高速な化合物全体の評価 表 5.7 によると、提案手法で 0.5% の化合物をフィルタリングし、それらを通常のドッキングシミュレーションで再評価することで glide HTVS モードを用いる場合の約 8 分の 1 の計算時間で評価を完了させることができる。例えば 1,000 万化合物を評価する場合、今回のケースの 450 倍程度の化合物数となるので、glide HTVS モードは 1 CPU 換算で 4 か月程度を要してしまう。一方、提案手法と通常ドッキングである glide SP モードの組み合わせでは 1 CPU でも半月程度で済む計算となる。この差は非常に大きく、提案手法は有用であると言える。 @memo 計算時間について、図（積み上げ棒グラフ？）で表現

表 5.8 化合物全体を評価するのに要する時間の比較

	合計計算時間 1,000 万化合物評価の	
	[CPU sec.]	推定時間 [CPU days]
提案手法で 0.5% フィルタリング	3,280	17.1
glide HTVS モード単独性能	23,552	122.5

従来手法以下の所要時間の中での予測精度の向上 表 5.7 に示されている通り、提案手法と従来手法とで単純に比較を行うと精度は従来手法に分がある。しかしいくつかのケースについては、化合物ライブラリのサイズを変えることで同程度の所要時間の中で精度を高めることができる。例えば、100 万化合物を従来手法で 10% にフィルタリングし、glideSP でリランキングした場合、上位 1 万化合物の濃縮率 (EF 1% に相当する) は 16.89、この時の推定必要計算時間は 32.0 CPU days となる。一方、1,000 万化合物を提案手法で 1% にフィルタリングし、glide SP でリランキングした場合、上位 1 万化合物の濃縮率 (EF 0.1% に相当する) は 20.26、この時の推定必要計算時間は 26.9 CPU days となり、速度を向上させつつ、予測精度を高めることができる。このようなケースは複数存在しており（表 5.9）、これらの場合においては提案手法を利用すべきであると言える。

表 5.9 提案手法が従来手法に速度・精度ともに勝る例

	化合物ライブラリサイズ	上位 1 万化合物の 濃縮率 (EF)	推定計算時間 [CPU days]
提案手法で 1%フィルタリング	1,000 万	20.26 (EF 0.1%)	26.9
従来手法で 10%フィルタリング	100 万	16.89 (EF 1%)	32.0
提案手法で 1%フィルタリング	500 万	13.91 (EF 0.2%)	13.5
従来手法で 10%フィルタリング	50 万	11.07 (EF 2%)	16.0
提案手法で 2%フィルタリング	1,000 万	24.87 (EF 0.1%)	46.7
従来手法で 10%フィルタリング	200 万	23.80 (EF 0.5%)	64.0
提案手法で 5%フィルタリング	200 万	16.80 (EF 0.5%)	21.2
従来手法で 5%フィルタリング	100 万	16.28 (EF 1%)	22.1

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。提案手法を DUD-E の全ターゲットである 102 種のデータセットを用いて評価すると、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ることが示された。

表 6.1 提案手法の性能

手法	ROC-AUC	Enrichment Factor				平均計算時間 [CPU sec.]
		EF(1%)	EF(2%)	EF(5%)	EF(10%)	
提案手法 (maxsumBS)	0.679	6.03	5.03	3.96	3.00	1,673
従来手法 (glide HTVS モード)	0.705	16.67	11.18	6.38	4.11	14,813

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。

表 6.2 通常ドッキング (glide SP) と組み合わせた速度・精度評価

	化合物ライブラリサイズ	上位 1 万化合物の濃縮率 (EF)	推定計算時間 [CPU days]
提案手法で 1% フィルタリング	1,000 万	20.26 (EF 0.1%)	26.9
従来手法で 10% フィルタリング	100 万	16.89 (EF 1%)	32.0

6.2 今後の課題

今後に向けて、以下の課題が考えられる。

- 速度をなるべく維持しつつの精度の向上
 - － フラグメントをドッキングする際のスコア関数の改善
 - － 通常ドッキングシミュレーションの化合物の結合スコアへの、フィルタリングスコアのフィッティング
 - － 化合物のスコア算出時の非現実的なフラグメント配置に対するペナルティの付与
- どのようなケースで提案手法を用いるのが好ましいかのさらなる調査
- 数百万～数千万化合物程度の、より現実のバーチャルスクリーニングに即した化合物データセットを用いた速度評価
- 提案手法と従来手法の2段階フィルタリングを行った場合の性能・速度評価

謝辞

本研究を進めるにあたり、貴重な時間を割いてご指導を賜り、本論文をまとめる際においても細やかなご助言をいただきました秋山 泰教授に深く感謝申し上げます。

また、研究内容の方向性のディスカッションや発表資料のとりまとめなど、多くの事柄に対して丁寧なご指導をいただきました石田 貴士准教授、ならびに大上 雅史助教に感謝の意を表します。

さらに、本研究を進めるにあたり秋山研究室・関嶋研究室・石田研究室合同ゼミを通して物理化学的な背景を含めた細かく的確なアドバイスを頂いた関嶋 政和准教授に御礼申し上げます。

最後に、本研究を行うにあたり秋山研究室・関嶋研究室・石田研究室の皆様には多大なるご協力を賜りました。暖かく、時には厳しいご指摘を通してご支援いただきましたことを心より感謝いたします。

参考文献

- [1] 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.
- [2] 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.
- [3] Zsolt Zsoldos, Darryl Reid, Aniko Simon, Sayyed Bashir Sadjad, and a. Peter Johnson. eHiTS: A new fast, exhaustive flexible ligand docking system. *Journal of Molecular Graphics and Modelling*, Vol. 26, No. 1, pp. 198–212, 2007.
- [4] Garrett M. Morris, Huey Ruth, William Lindstrom, Michel F. Sanner, Richard K. Belew, David S. Goodsell, and Arthur J. Olson. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of Computational Chemistry*, Vol. 30, No. 16, pp. 2785–2791, 2009.
- [5] Corwin Hansch and Toshio Fujita. ρ - σ - π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. *Journal of the American Chemical Society*, Vol. 86, No. 8, pp. 1616–1626, 1964.
- [6] Ovidiu Ivanciuc. Applications of support vector machines in chemistry. *Reviews in computational chemistry*, Vol. 23, pp. 291–400, 2007.
- [7] Gerhard Wolber, Thomas Seidel, Fabian Bendix, and Thierry Langer. Molecule-pharmacophore superpositioning and pattern matching in computational drug design. Vol. 13, No. January, pp. 23–29, 2008.
- [8] J B Brown and Yasushi Okuno. Minireview Systems Biology and Systems Chemistry : New Directions for Drug Discovery Minireview. *Chemistry & Biology*, Vol. 19, No. 1, pp. 23–28, 2012.

-
- [9] Dennis M. Krüger and Andreas Evers. Comparison of structure- and ligand-based virtual screening protocols considering hit list complementarity and enrichment factors. *ChemMedChem*, Vol. 5, No. 1, pp. 148–158, 2010.
- [10] Elizabeth Yuriev and Paul a. Ramsland. Latest developments in molecular docking: 2010-2011 in review. *Journal of Molecular Recognition*, Vol. 26, No. October 2012, pp. 215–239, 2013.
- [11] S Kannan and R Ganji. Porting Autodock to CUDA. *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pp. 1–8, 2010.
- [12] S. McIntosh-Smith, J. Price, R. B. Sessions, and a. a. Ibarra. High performance in silico virtual drug screening on many-core processors. *International Journal of High Performance Computing Applications*, pp. 1094342014528252–, apr 2014.
- [13] Oleg Trott and Arthur J. Olson. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, Vol. 31, No. 2, pp. 455–461, jan 2010.
- [14] John J Irwin and Brian K Shoichet. ZINC – A Free Database of Commercially Available Compounds for Virtual Screening ZINC - A Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Model*, Vol. 45, pp. 177–182, 2005.
- [15] R Nilakantan, N Bauman, and R Venkataraghavan. New method for rapid characterization of molecular shapes: applications in drug design. *Journal of chemical information and computer sciences*, Vol. 33, No. 1, pp. 79–85, 1993.
- [16] Marco Daniele Parenti, Anna Maria Ferrari, and Via Campi. Docking and Database Screening Reveal New Classes of. *Journal of Medicinal Chemistry*, pp. 2834–2845, 2003.
- [17] Taku Fujimoto, Yasuo Matsushita, Hiroaki Gouda, Noriyuki Yamaotsu, and Shuichi Hirono. In silico multi-filter screening approaches for developing novel beta-secretase inhibitors. *Bioorganic & medicinal chemistry letters*, Vol. 18, No. 9, pp. 2771–5, 2008.
- [18] I D Kuntz, J M Blaney, S J Oatley, R Langridge, and T E Ferrin. A geometric approach to macromolecule-ligand interactions. *Journal of molecular biology*, Vol. 161, No. 2, pp. 269–288, 1982.
- [19] 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.

- [20] 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.
- [21] G Richard Bickerton, Gaia V Paolini, J  r  my Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, Vol. 4, No. 2, pp. 90–8, 2012.
- [22] How long does it take to screen 10,000 compounds with glide?
- [23] Matthias Rarey, Bernd Kramer, Thomas Lengauer, and Gerhard Klebe. A fast flexible docking method using an incremental construction algorithm. *Journal of molecular biology*, Vol. 261, No. 3, pp. 470–89, 1996.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] Thomas a. Halgren. Identifying and characterizing binding sites and assessing druggability. *Journal of Chemical Information and Modeling*, Vol. 49, No. 2, pp. 377–389, 2009.
- [28] Venkat K Pulla, Dinavahi Saketh Sriram, Srikant Viswanadha, Dharmarajan Sriram, and Perumal Yogeeswari. Energy Based Pharmacophore and 3D QSAR Modeling Combined with Virtual Screening to Identify Novel Small Molecule Inhibitors of SIRT1. *Journal of Chemical Information and Modeling*, Vol. 12, p. acs.jcim.5b00220, 2015.
- [29] Faezeh Shirgahi Talari, Kowsar Bagherzadeh, Sahand Golestanian, Michael Jarstfer, and Masoud Amanlou. Potent Human Telomerase Inhibitors: Molecular Dynamic Simulations, Multiple Pharmacophore-Based Virtual Screening, and Biochemical Assays. *Journal of Chemical Information and Modeling*, p. acs.jcim.5b00336, 2015.

@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

DUD-E (A Database of Useful Decoys: Enhanced) は Mysinger らによって作成されたドッキングシミュレーションツールを評価するためのデータセットである²⁶⁾。ターゲットは多様性を考慮して 102 種類が選択されており、それぞれに対して

- 代表タンパク質構造
- 正例となる既知の薬剤・阻害剤
- 負例となるデコイ（実験は行われていないが、既知の薬剤・阻害剤と構造が似ていないためターゲットを阻害しないと考えられる化合物）および実験によってターゲットを阻害しないことが知られている化合物

が用意されている。@todo 以下作成中。スクリプト書いて完成させねば

表 A.1 DUD-E の詳細

ターゲット名	PDBID	タンパク質詳細	化合物数		平均フラグメント数	
			正例	負例	正例	負例
aa2ar	3EML	Adenosine A2a receptor	482	31,498	6.26	7.04
abl1	2HZI	Tyrosine-protein kinase ABL	182	10,746	7.08	7.33

付録 B

各手法を単体で用いた場合の ROC 曲線

総和法 (score_sum)、最良値法 (score_max)、総和法と最良値法の値の線形和 (maxsumBS) の 3 つの提案手法および glide HTVS モードをそれぞれ単体で用いた場合の ROC 曲線を示す。3 つの提案手法についてはフラグメントの結合スコア算出を glide SP モード / glide HTVS モードで行った場合がそれぞれ示されている。例えば、「score_max_SP」とはフラグメントの結合スコアを glide SP モードで求め、そのフラグメントスコアの最良値をとった場合の精度が ROC 曲線で示されている。

@memo 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。図 B.1 のようなものがターゲットを変えながら 102 個並ぶ。 $2 \times 3 = 6$ が 1 ページで、それが 18 ページ続く形になることを想像している。

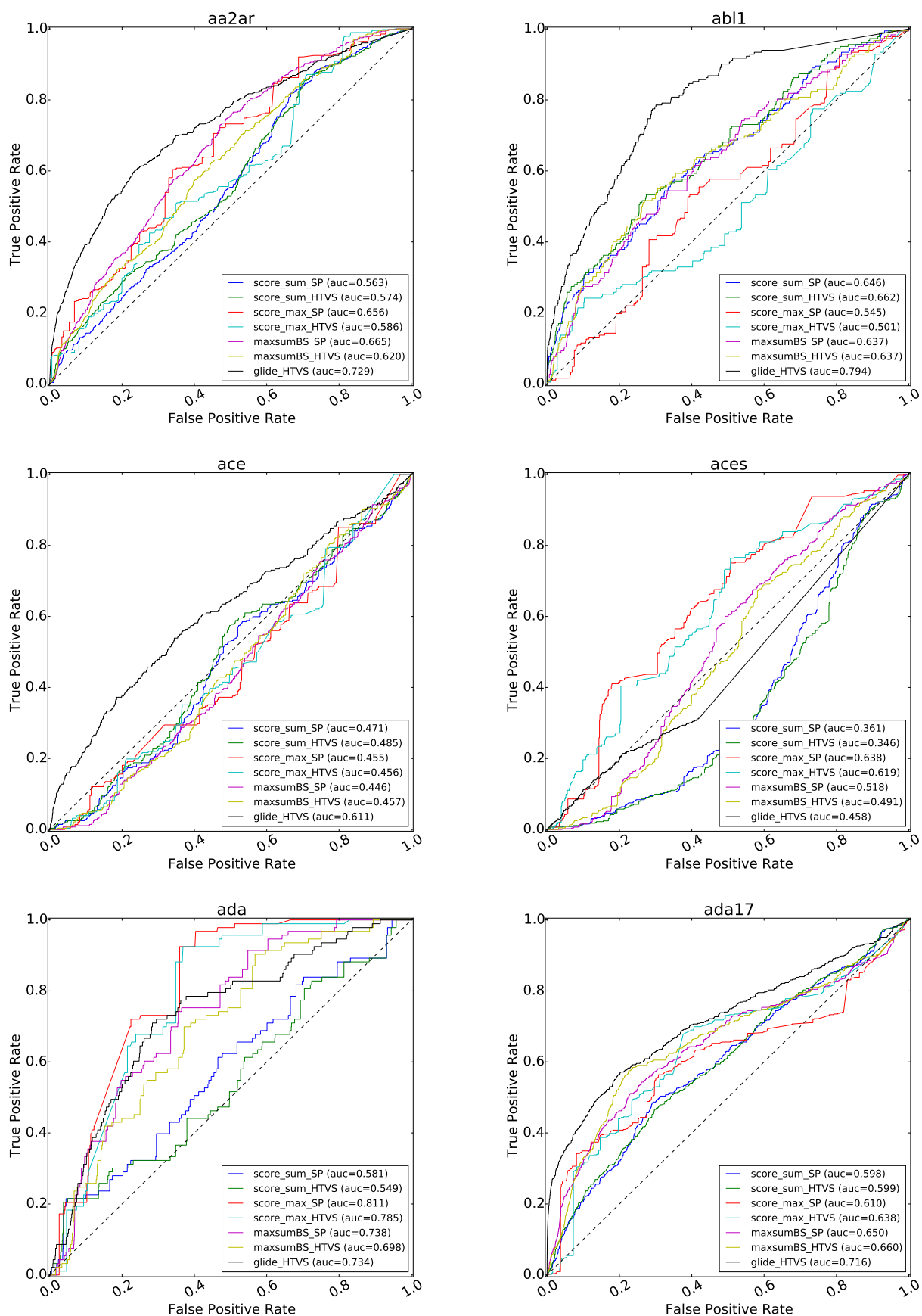


図 B.1 ROC 曲線例