目 次

図 目 次

表 目 次

第1章

序論

1.1　研究背景

- computer-aided drug discovery では、SBDD, LBDD, CGBVS の 3 種類の薬剤候補化合物の選別手法が存在している
- このうち、SBDD は演繹的な手法であり、タンパク質の構造が得られれば阻害剤が存在しなくとも薬剤開発が可能であり、非常に有用。
  もし阻害剤が知られているターゲットだとしても構造が既知の阻害剤とは大きく異なる薬剤候補を見つけられる。
  これは LBDD や CGBVS にはないメリット**@todo CGBVS にはこのメリットはないのか？どんなメリットが主張されているのか？調査が必要。**
  @memo この時点で Pharmacophore の手法は除外している
- SBDD では化合物-タンパク質ドッキングというシミュレーション手法を用いて化合物を評価する。
- 様々な研究がすすんでおり、Glide, Autodock 等といったさまざまなツールが開発されている。
- その中でも、Glide というドッキングツールが良い精度を出すことが知られている @cite 比較論文 ref
- このドッキングツールは、行う計算の内容の関係上、計算コストが高い。
- ドッキング計算手法の高速化研究 @cite GPU 実装, Autodock Vina などは行われているが、不十分である。
- そのため、ドッキングツールを用いて化合物として購入可能な数千万以上の化合物 (ZINC の件数を利用) を一斉に評価することは難しい。

- したがって、フィルタリングの必要性がある。
- しかしフィルタリングは既知の化合物に基づいた手法が殆どであり、SBDD の長所である新規の構造を持つ薬剤候補の発見能力を奪うことになる。
- Glide が高速ドッキングモードを提供しているが、数千万化合物の単位ではまだ計算コストが大きい。

1.2 研究目的

- 研究背景から、より高速に、新規の構造を持つ薬剤候補をフィルタリングする必要がある
- そこで、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この研究では提案する

1.3 本論文の構成

2 章では提案手法のベースとなるドッキングシミュレーションについて詳しく説明、3 章で提案するフィルタリング手法の説明を行う。4 章で実験について述べ、5 章ではこの実験の結果についての考察を加える。最後に 6 章で結論および今後の展望について述べる。

第2章

化合物-タンパク質ドッキングシミュレーション

@todo ドッキングベースのフィルタリング手法を提案するために、ドッキングシミュレーションの説明をする、という流れは遠回り過ぎる？

2.1　ドッキングシミュレーションの要素

2.1.1　探索アルゴリズム

計算コスト削減の鍵となるので、詳しく説明する。

2.1.2　スコア関数

こちらはあまり詳細に踏み込まない。

2.2 現存するドッキングツール

2.2.1 Glide 通常ドッキングモード (Glide SP モード)

2.2.2 GOLD

2.2.3 Autodock

2.3 ドッキングによるフィルタリングツール

2.3.1 Glide 高速ドッキングモード (Glide HTVS モード)

@todo Glide HTVS をフィルタリングツールと呼んでいいのか？内部でどのような計算の簡略化を行っているのか記述。

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

3.1　提案手法へのアイデアの概説

ここでは数値的な話は一切しない。こういうアイデアの元でやりますよーという話。

- まず、これから提案する手法の要件を定義する。
  - フィルタリングにおいて重要なことは計算時間が短いこと
  - 逆に、構造は出せなくてもその後の通常ドッキングがやってくれるから構造を出力することに固執しません
- 要件から、提案手法のフローチャートはこのようにしました。（従来手法と提案手法、それぞれのフローチャートを図で示す）
  - 化合物を分割してからドッキングします。分割によって内部自由度を削減すると、前述の理由によりドッキング計算コストが減少します。
    (background の時点でドッキングの計算量は内部自由度によって増減することを示しておく。)
  - 衝突を考慮するなどの構造の再構成はせず、ドッキングで得られたスコアのみに着目して計算を行います。
    図 3.1 のように化合物数に対して計算が $O(n)$ で済むように計算することで、高速に化合物のスコアを得ます。

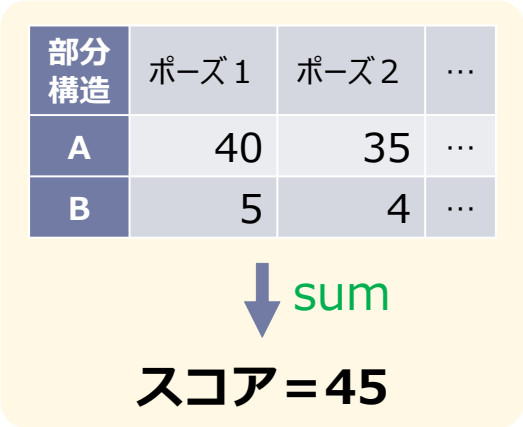

図 3.1 スコア統合イメージ (暫定版)

3.2 提案手法の詳細の説明

すでにフローチャートを示しているので、「化合物の分割」、「部分構造単位でのドッキング」「部分構造のスコアの統合」の 3 つについて説明する。

3.2.1 化合物の分割

- 分割は基本的に小峰による手法を用い 、重原子数が 2 以下のフラグメントは除外することを示す。
- 図 3.2 のような具体例を示し、分割がどのように行われるのか明確にする。必ず「内部自由度を考慮しないドッキングで OK」であることに言及する。

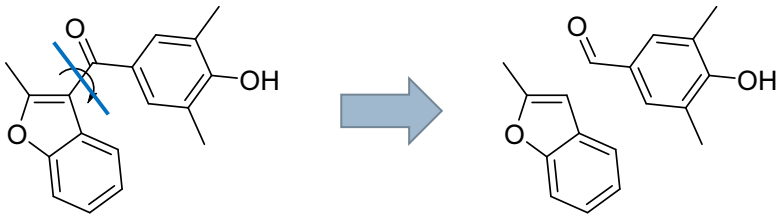

図 3.2 フラグメント分割例

- また、この分割によって多数の化合物で共通部分が発生することも示す。共通部分が発生することも図で示す**@todo 図の作成**
  これについては、1000 万化合物に対して行った実験（図 3.3）を説明、結果を示す。

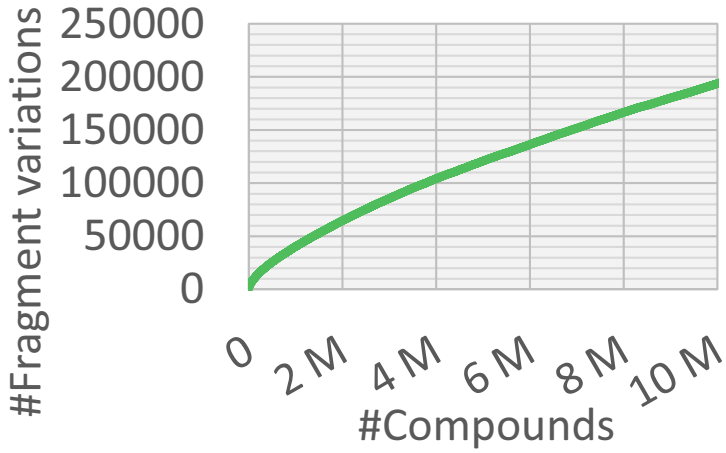

図 3.3 大量の化合物を分割した場合の例

3.2.2 部分構造単位でのドッキング

- 以下の二通りのドッキングで実験を行うことを述べる。
  - glide の通常ドッキングモード (SP) & 内部自由度を無視するオプション
  - glide の高速ドッキングモード (HTVS)
- ドッキング結果として複数の構造が出力される場合があるが、そのときは最良のスコアを取得することを示す (図 3.4 のような)。

以下の二通りのドッキングを行う。

3.2.3 部分構造のスコアの統合

最初に、フラグメントごとのドッキング結果は構造を保持しないことを図で示し、したがってポーズを出力しないことを明記する。**@todo 図 3.5 のようなものを示す。**（なぜか図 3.5 が表示されないが、図 3.4 と同様に手書きなので無視。）

ここでは 3 種類のスコア統合手法について述べる。

部分構造スコアの総和

score_sum について記述。

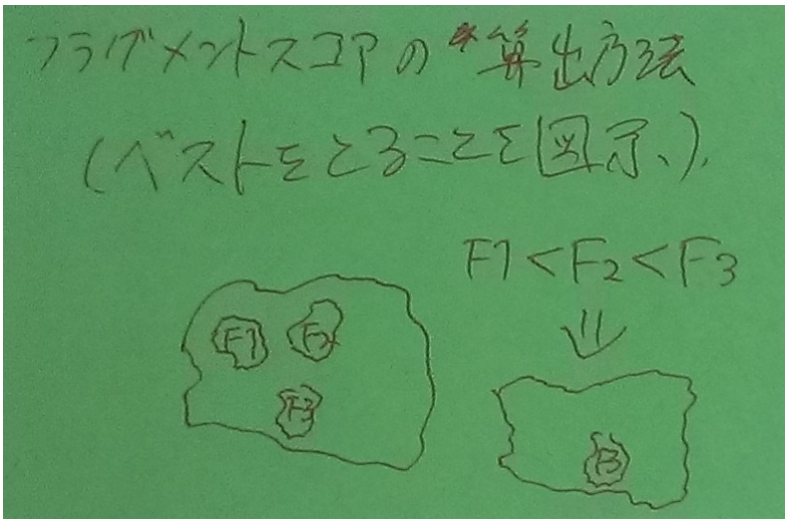

図 3.4 部分構造スコアの取得

図 3.5 統合結果の構造

部分構造スコアの最良値

score_max について記述。

総和法と最良値法の線形和

maxsumBS について記述。

第4章

実験

@comment 現状よりも詳しく記述すべき（大上先生）
ここで話すことは大きく二つ。
どこかで評価軸の説明をする

- それぞれの手法の単独性能について
- それぞれの手法を filtering 手法として用いた場合について

4.1　データセット

DUD-Eref を利用した、だけではなく、input と output を明確にする。

表 4.1　DUD-E diverse subset の詳細

ターゲット名	タンパク質名	正例		負例	
		化合物数	平均分割数	化合物数	平均分割数
akt1					
ampc					
cp3a4					
cxcr4					
gcr					
hivpr					
hivrt					
kif11					

4.2 評価基準

@comment EF および ROC-AUC について示す。できれば数式と算出イメージ図（算出例？）を示す（特に EF)。（大上先生）

4.2.1 ROC-AUC

4.2.2 EF (EnrichmentFactor)

4.3 計算環境

TSUBAME Thin ノード

4.4 実験結果

4.4.1 ドッキングにかかる計算時間

CPU 時間で示す。 @memo ドッキングの計算時間のみを示してよいのか？フラグメント分割やスコア統合の計算時間をどう示すべきか (所要時間：後者 << 前者 << ドッキング計算)

4.4.2 予測精度

ここでは単独手法としての精度を示す。ROC 曲線は Appendix として載せていることを記述@comment ROC 曲線も本文に示す。良い/悪いの評価をする際に必要になるため。（大上先生）

表 4.2 フィルタリング手法の予測精度

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 通常モード					
	glide 高速モード					
最良値 (score_max)	glide 通常モード					
	glide 高速モード					
線形和 (maxsumBS)	glide 通常モード					
	glide 高速モード					
従来手法 (glide 高速モード)						

4.4.3 filteringと通常のドッキングとを合わせた場合の計算時間および精度

filtering時に何%の化合物を残すか、などの議論と合わせて、精度と速度のバランスを示す。@comment トレードオフを示すのについて、フィルタリングのパーセンテージをもっと振るべきでは？5, 10, 20, 30, 40, 50%のように。（大上先生）

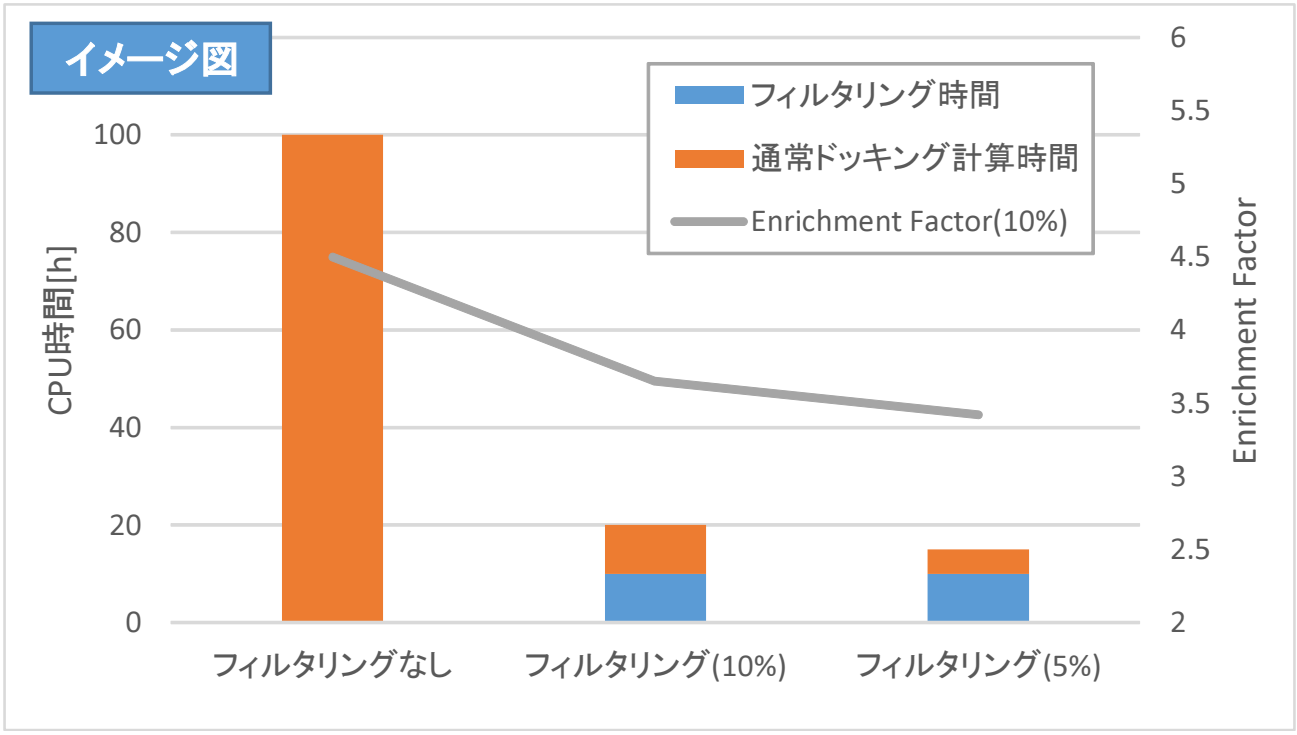

図 4.1 計算時間と精度のトレードオフ

第5章

考察

5.1　提案手法の得手・不得手

ターゲットの active ligand について、何らかの考察を加えることで提案手法が向くタイプのターゲット、向かないタイプのターゲットを評価する。

5.2　フラグメントスコア計算式の改善

5.2.1　各種エネルギースコアの寄与率の評価

hbond, evdw, ecoul など、それぞれ別々に用いた場合の評価がどうなっているのか。
ターゲットの正解化合物の傾向との関係が見つけられればうれしいが。

5.2.2　Glide GScore のフラグメント向け修正

フラグメントドッキングにおいては gscore をそのまま使うのではなく、少しパラメータを調整した方が良い。
公正な評価のために、通常のドッキングに関しても同様のパラメータチューニングをかける。

- 大きな DB に適用した場合の話をする（？　その場合は実験の段において「一致率」による評価を行う）

第6章

結論

6.1　本研究の結論

6.2　今後の課題

謝辞

ほげほげ。

参考文献

[1] ほげほげ

付録 A

ROC曲線

求められた ROC 曲線をひたすらかく。