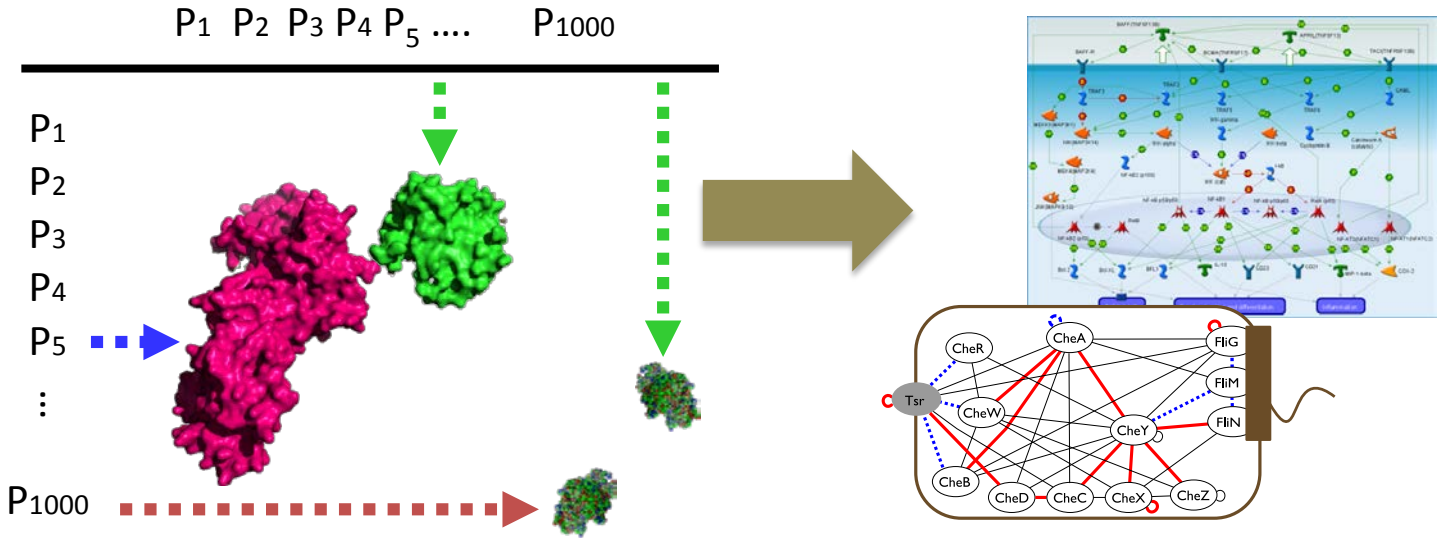
網羅的タンパク質ドッキングプログラム MEGADOCK-K の利用

松崎　由理, 大上　雅史, 内古閑　伸之



■課題代表者　秋山泰(1)
□課題参加者　松崎由理(2)　大上雅史(1)　石田貴士(1)　内古閑伸之(3)

(1)東京工業大学　大学院情報理工学研究科　(2)東京工業大学　情報生命博士教育院
(3)中央大学　理工学部

MEGADOCK-K 講習会の概要

- 開発の背景
- 「京」をはじめとする高性能並列計算機上での実行を想定した並列化と性能評価
- ドッキングスコア関数の開発
- MEGADOCK-K実行の流れ
- MEGADOCK-Kを応用したタンパク質間相互作用予測研究の例

背景

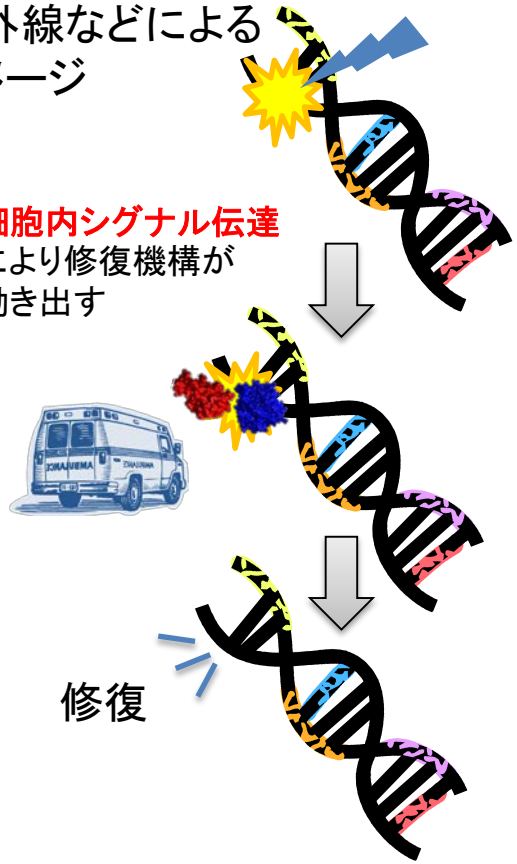
<<膨大なタンパク質構造データを活用した生命システム解析>>

- タンパク質は細胞の機能を担う重要な分子
- タンパク質間の相互作用が生体の機能を構成
- 膨大なタンパク質構造データを相互作用解析に活用したい (2014/12/8現在のPDB登録データは105,383)
- 高性能計算機の活用による網羅的な解析（システム解析）も視野に入る
  - MDなどのシミュレーション
    - 時間解像度の高いシミュレーションで動的な性質を考察
    - 計算量は膨大
  - タンパク質ドッキング
    - 静止画的解析（剛体ドッキング）
    - 網羅的な解析を現実的な時間スケールで実行可能

細胞活動の制御を担う
タンパク質間相互作用（PPI）ネットワーク

例：DNA修復

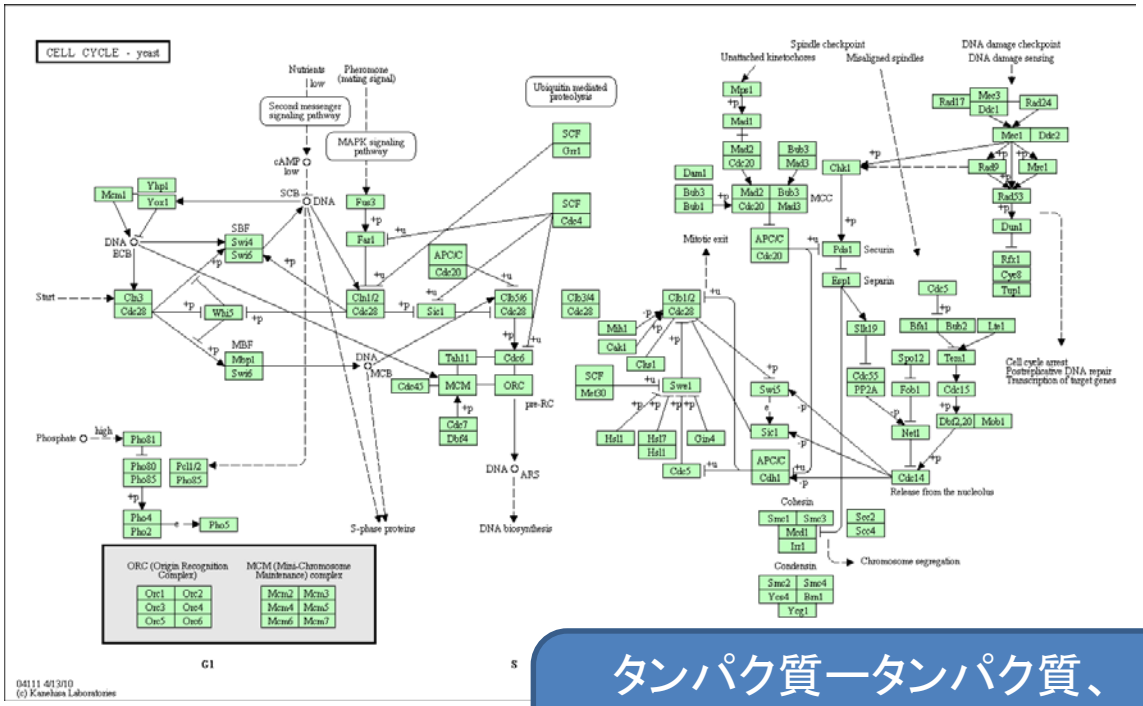
紫外線などによる
ダメージ



細胞内シグナル伝達
により修復機構が
動き出す

修復

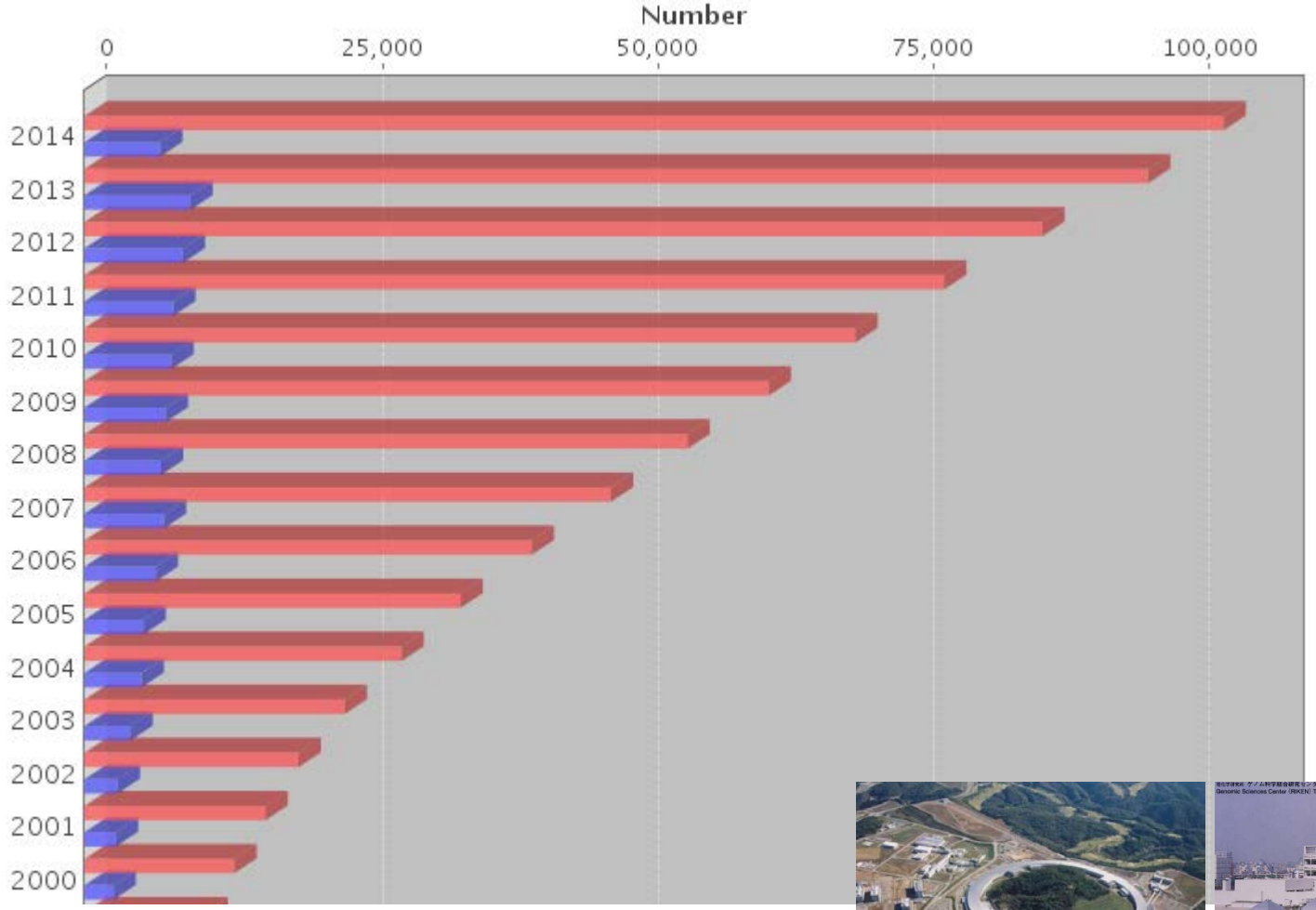
細胞内のシグナル伝達経路



http://www.kegg.jp

タンパク質ータンパク質、
タンパク質ー小分子の
相互作用ネットワーク

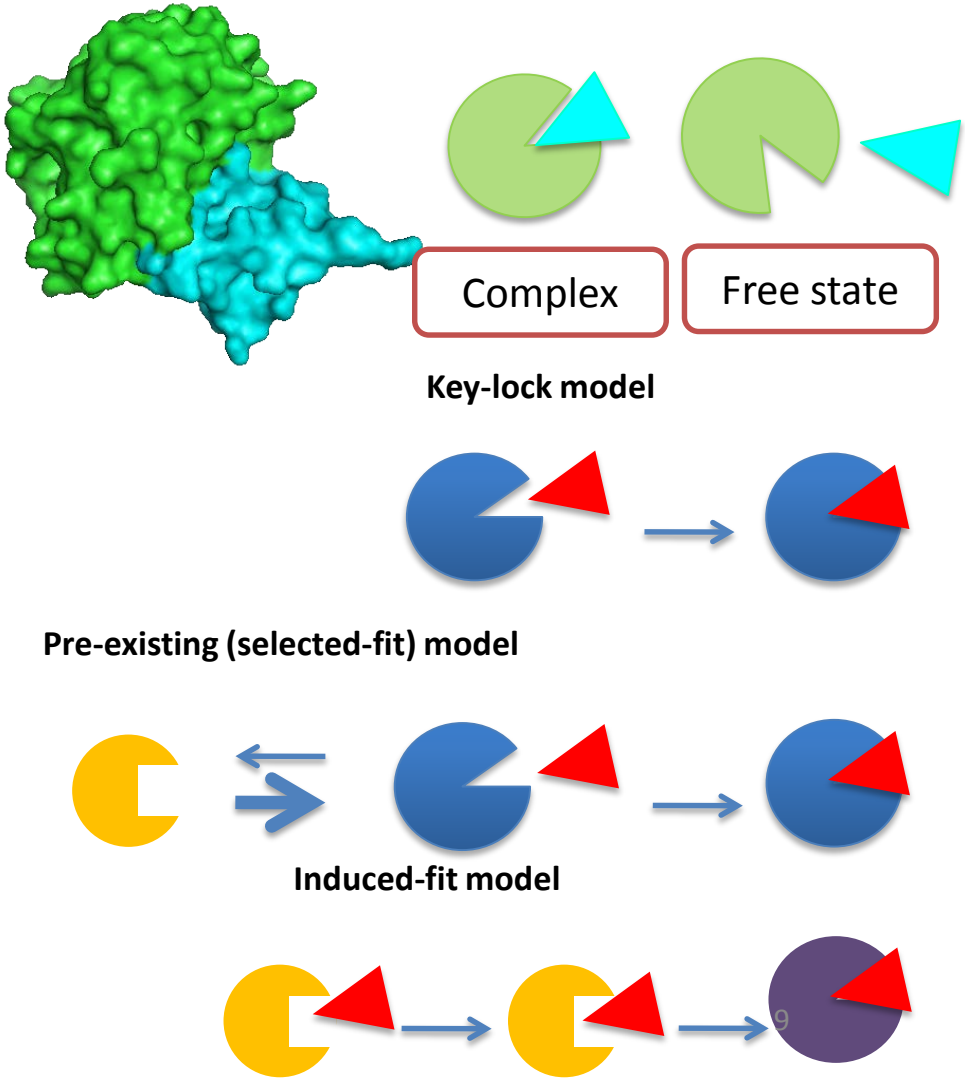
タンパク質立体構造データ数は年々伸びている



RCSB PDB
(as of 2014-09-22)

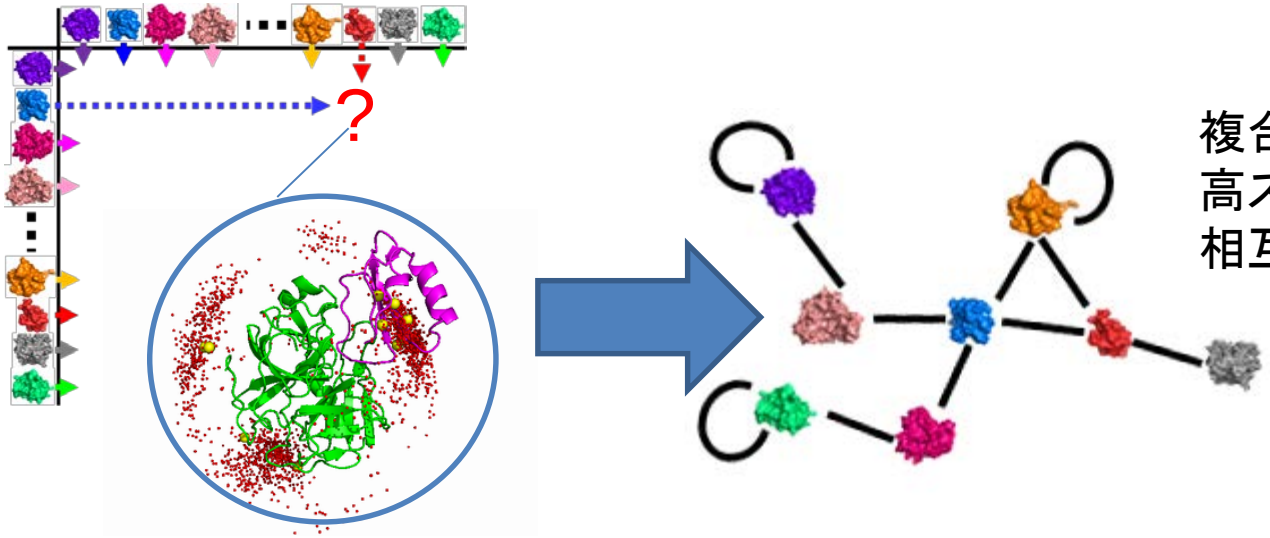
構造のマッチングはタンパク質間相互作用の特異性を決める重要な要素

- 複合体の三次元構造において二つのタンパク質の形状は相補的になっている
- 複合体の状態の形状と独立した状態の形状がよく似ていることが多い（違うものも多くある）



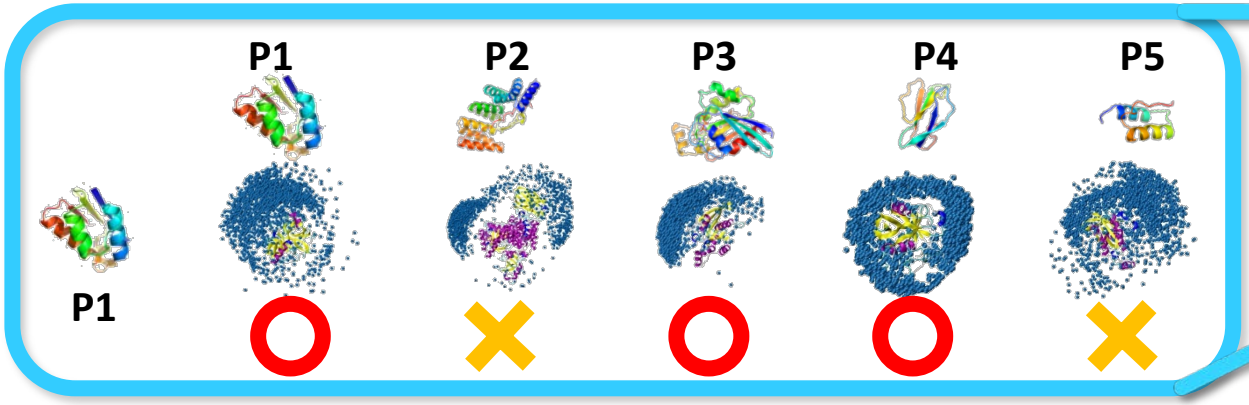
タンパク質立体構造を用いた相互作用予測問題

- 入力: タンパク質の立体構造群
- 方法:
  剛体ドッキングの網羅的計算と
  ポストドッキング解析
- 出力: タンパク質間相互作用ネットワーク



MEGADOCK：タンパク質ドッキングによる網羅的PPIネットワーク推定システム

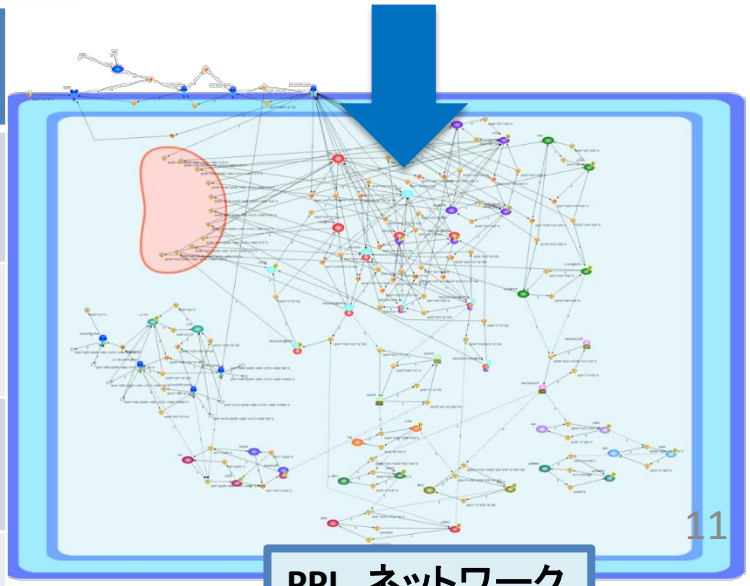
網羅的ドッキングとポストドッキング解析



相互作用予測

	P1	P2	P3	P4	P5
P1	○	×	○	○	×
P2	×	×	×	×	×
P3	○	×	×	○	×
P4	○	×	○	×	○
P5	×	×	×	○	×

Target pathway	Problem size (#dockings)	1ノードで計算すると、、
バクテリアのシグナル伝達系 （chemotaxis）	10,201	約3ヶ月
ヒトのシグナル伝達系（apoptosis）	24,964	約7ヶ月
ヒトのシグナル伝達系（EGFR）	247,009	約6年
肺がん関連タンパク質	3,690,241	約92年



MEGADOCK-K 講習会の概要

- 開発の背景
- 「京」をはじめとする高性能並列計算機上での実行を想定した並列化と性能評価
- ドッキングスコア関数の開発
- MEGADOCK-K実行の流れ
- MEGADOCK-Kを応用したタンパク質間相互作用予測研究の例

Strategies for the docking challenge

- **MEGADOCK**:
  $1000 \times 1000 = 1\ million$ スケールのドッキングを現実的な時間で計算するためプログラム
  - 新規スコア関数でスコア計算の時間を減じる
  - ソフトウェアの並列実装により「京」のような

  大規模並列計算環境で効率的に実行できるようにする

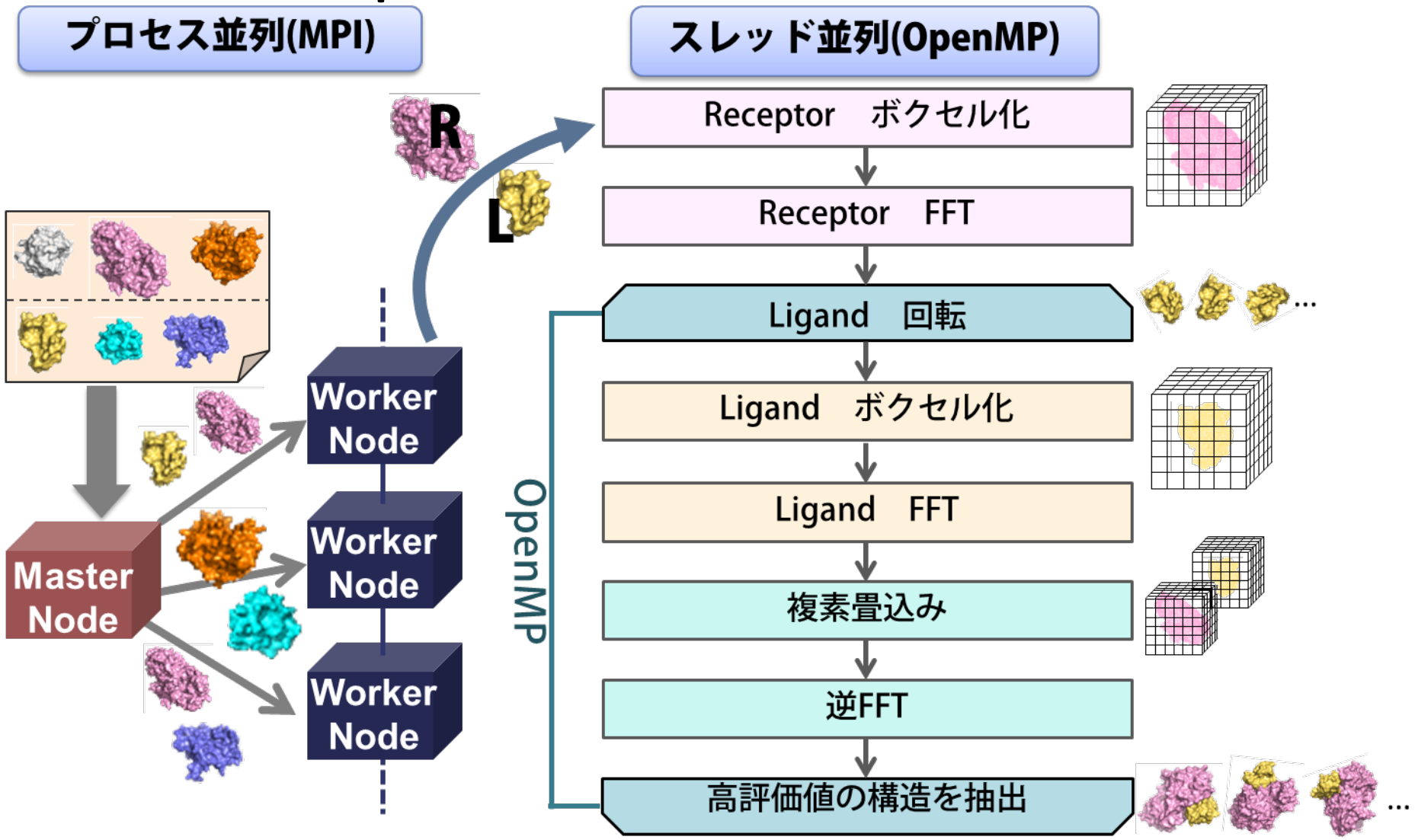
K computer @ RIKEN | MPI/OpenMP

4th in top500
(10.5 Petaflops, Nov. 2014)
705,024 CPU cores
(8 CPU cores ×
88,128 nodes)
Memory: 16GB/node

TSUBAME@Titech (GPU, OpenMP)

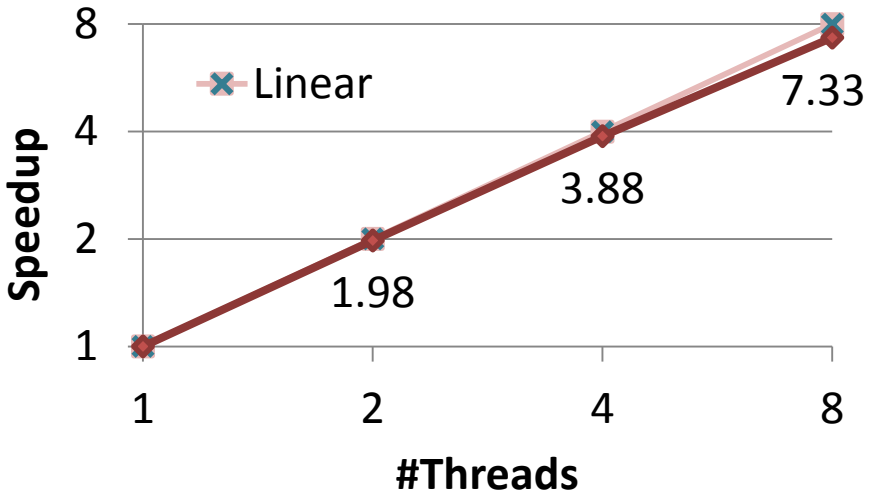
15st in top500
(2.8 Petaflops, Nov. 2014)
17,984 CPU cores
4,258 GPUs (3 GPUs/node)
Memory: 58GB/node

「京」での利用を想定した MPI/OpenMP ハイブリッド並列化



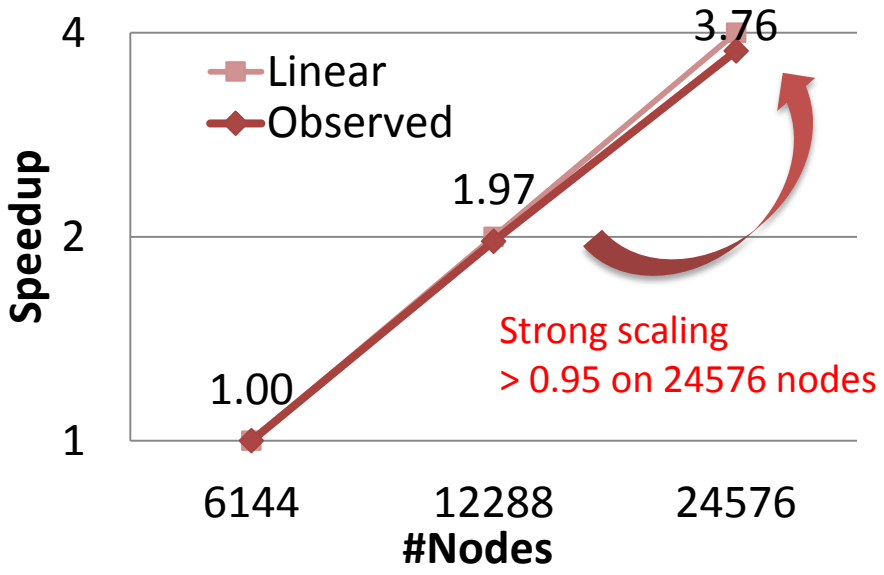
「京」での
並列スケーラビリティ

スレッド並列



ノード間並列

220タンパク質構造間の網羅的ドッキング



- オープンソースライセンスで公開済
- 東工大TSUBAMEでも400ノードの環境でノード間並列のスケーラビリティを確認

Matsuzaki, *et al.*, *Source Code Biol Med*, 2013.

MEGADOCK-K 講習会の概要

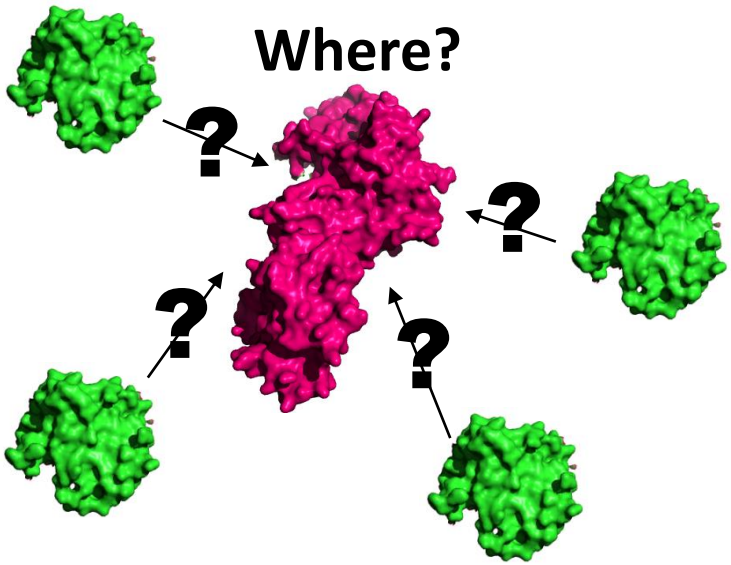
- 開発の背景
- 「京」をはじめとする高性能並列計算機上での実行を想定した並列化と性能評価
- ドッキングスコア関数の開発
- MEGADOCK-K実行の流れ
- MEGADOCK-Kを応用したタンパク質間相互作用予測研究の例

MEGADOCK-K 講習会の概要

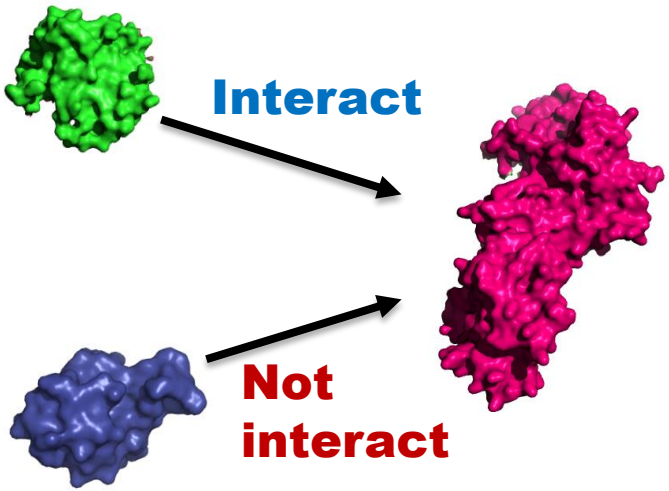
- 開発の背景
- 「京」をはじめとする高性能並列計算機上での実行を想定した並列化と性能評価
- ドッキングスコア関数の開発
- MEGADOCK-K実行の流れ
- MEGADOCK-Kを応用したタンパク質間相互作用予測研究の例

立体構造にもとづくPPI予測

- **タンパク質ドッキング**
  - 従来は結合部位予測などに用いられてきた
  - 複数の相手から結合相手を予測する問題にも適用できないか？



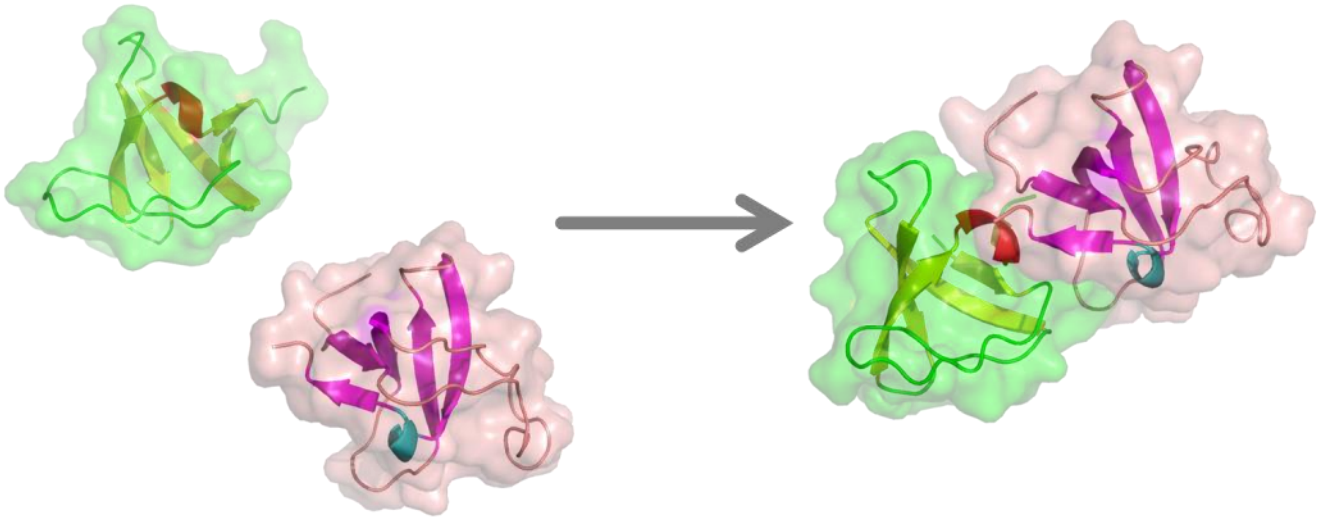
Interacting site prediction



PPI prediction

タンパク質ドッキング計算

- **タンパク質ドッキング計算**
  - 2つのタンパク質からなる複合体の構造を予測する
- **入力**
  - それぞれのタンパク質の立体構造情報
- **出力**
  - 予測されたタンパク質複合体構造



アンフィンゼンのドグマ

タンパク質はエネルギー最小状態に勝手にフォールディングする

タンパク質はアミノ酸配列が決まれば天然構造は一意に決まる

- **結合自由エネルギー差** $\Delta G_{binding}$

$$\Delta G_{binding} = \Delta E_{elec} + \Delta E_{vdW} + \Delta G_{des} + \Delta E_{int} - T\Delta S_{sc} - T\Delta S_{bb}$$

- ΔE_{elec} : 静電相互作用エネルギー
- ΔE_{vdW} : ファンデルワールスエネルギー
- ΔG_{des} : 脱溶媒和自由エネルギー
- ΔE_{int} : 結合時の構造変化に伴う内部エネルギー変化
  （結合角，二面角，結合長に関するエネルギー変化）
- ΔS_{sc} : 側鎖（side chain）エントロピー
- ΔS_{bb} : 主鎖（backbone）エントロピー

基本的な複合体構造予測(ドッキング)の戦略

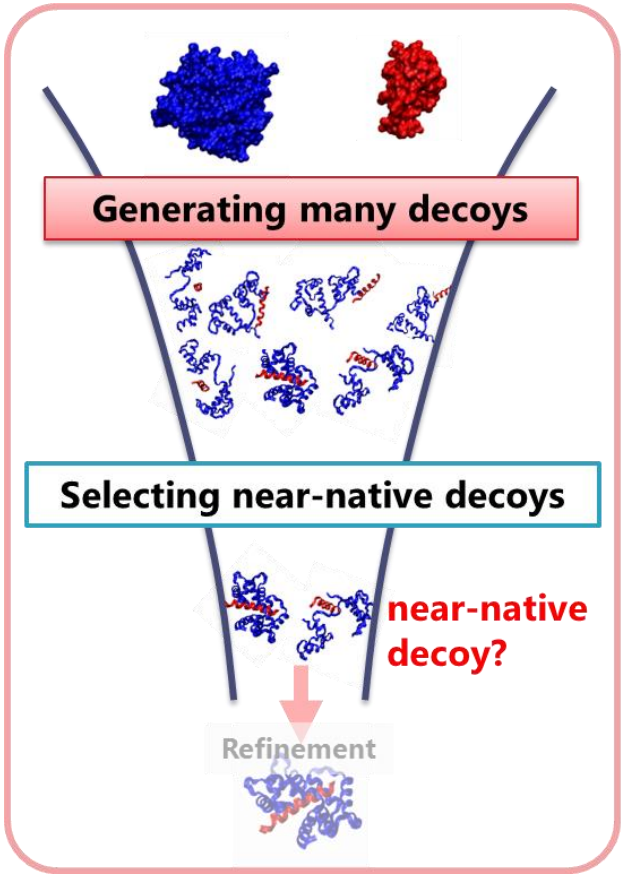
- **単純に攻めると組み合わせ爆発**
  - 膨大な構造の自由度
    - タンパク質全体の構造自由度(並進，回転)
    - タンパク質(アミノ酸)の主鎖構造の自由度
    - 個々のアミノ酸(側鎖)の構造自由度
  - 構造の評価関数

    ファンデルワールス力，静電相互作用，
    疎水性相互作用，水素結合，π-π相互作用，etc.

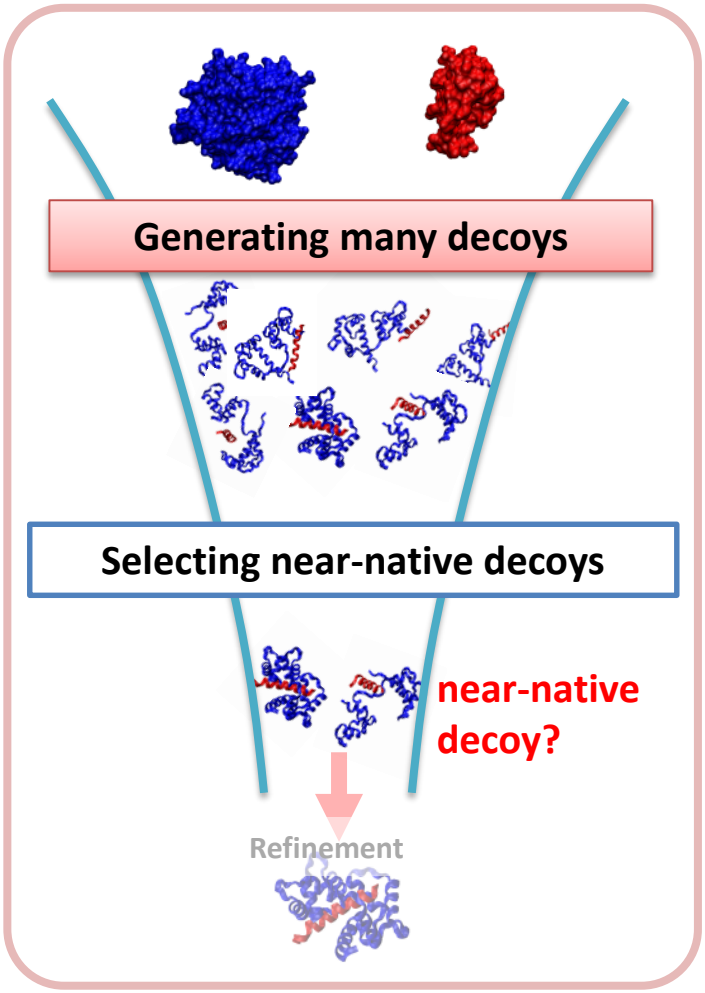
  → **全部を見て最適化は不可能**　**どこから手をつけるか**

- **基本的な戦略：大枠から攻める**

  0. どの辺りにくっつくかアタリをつける
  1. 単体構造を剛体として並進と回転を許した探索によるサンプリング(剛体ドッキング)
  2. 詳細な評価関数によるリスコアリング
  3. 主鎖/側鎖構造を変化させてリファインメント



剛体としてポーズを探索する



(まずは)剛体としてポーズの探索をする

$N \times N \times N$グリッド空間の平行移動とリガンド回転

N=128 と 3,600通りのリガンド回転の場合,

128 × 128 × 128 × 3,600

≒**75億個** の候補となる探索部位がある

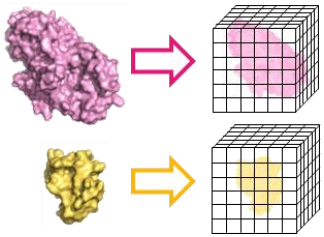
効率的にポーズを探索して
サンプリングする手法が必要

解決策

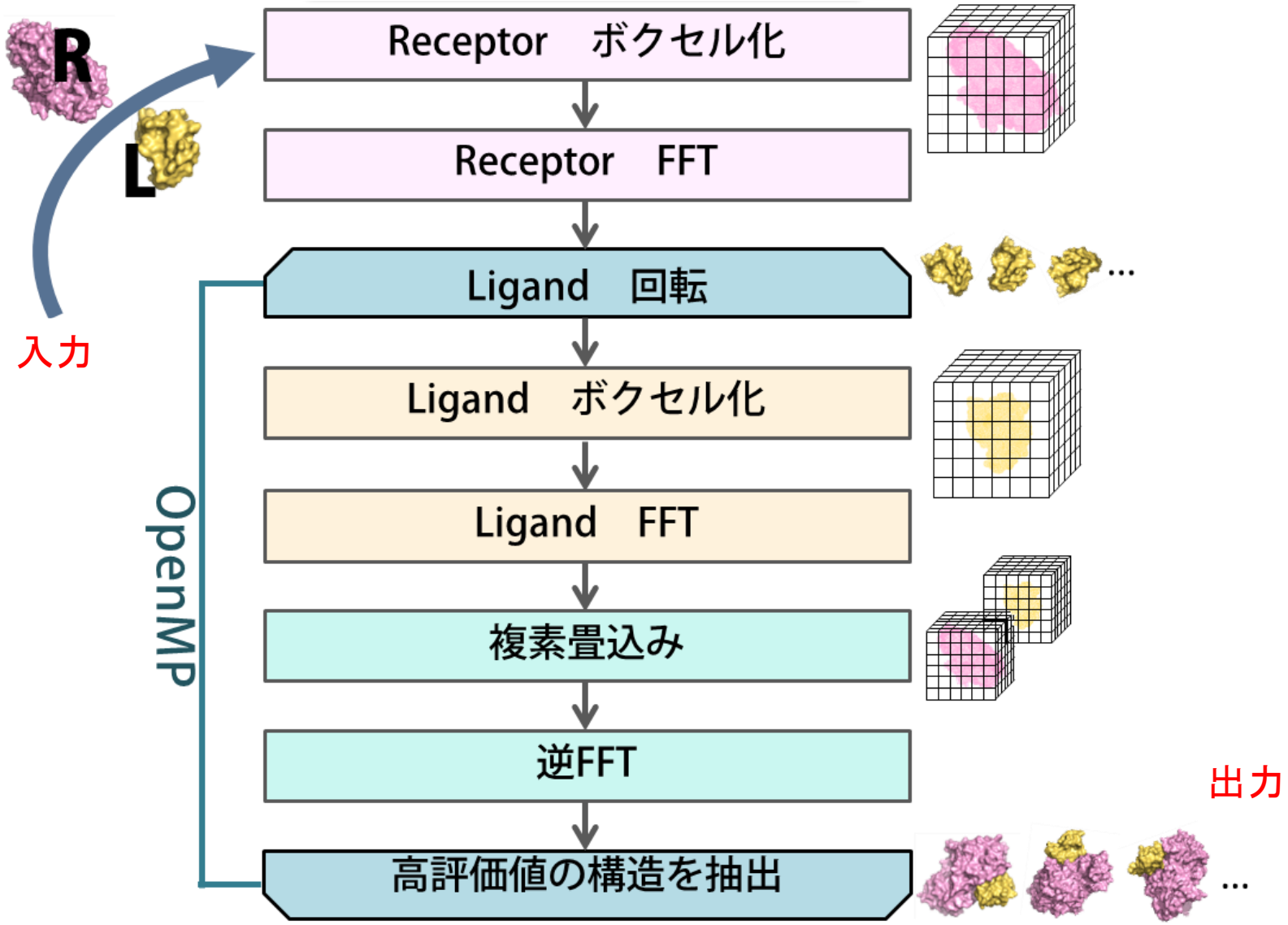
⇨ グリッド空間上のスコア関数を定義して,
サンプリングの基準とする

MEGADOCKのドッキング計算

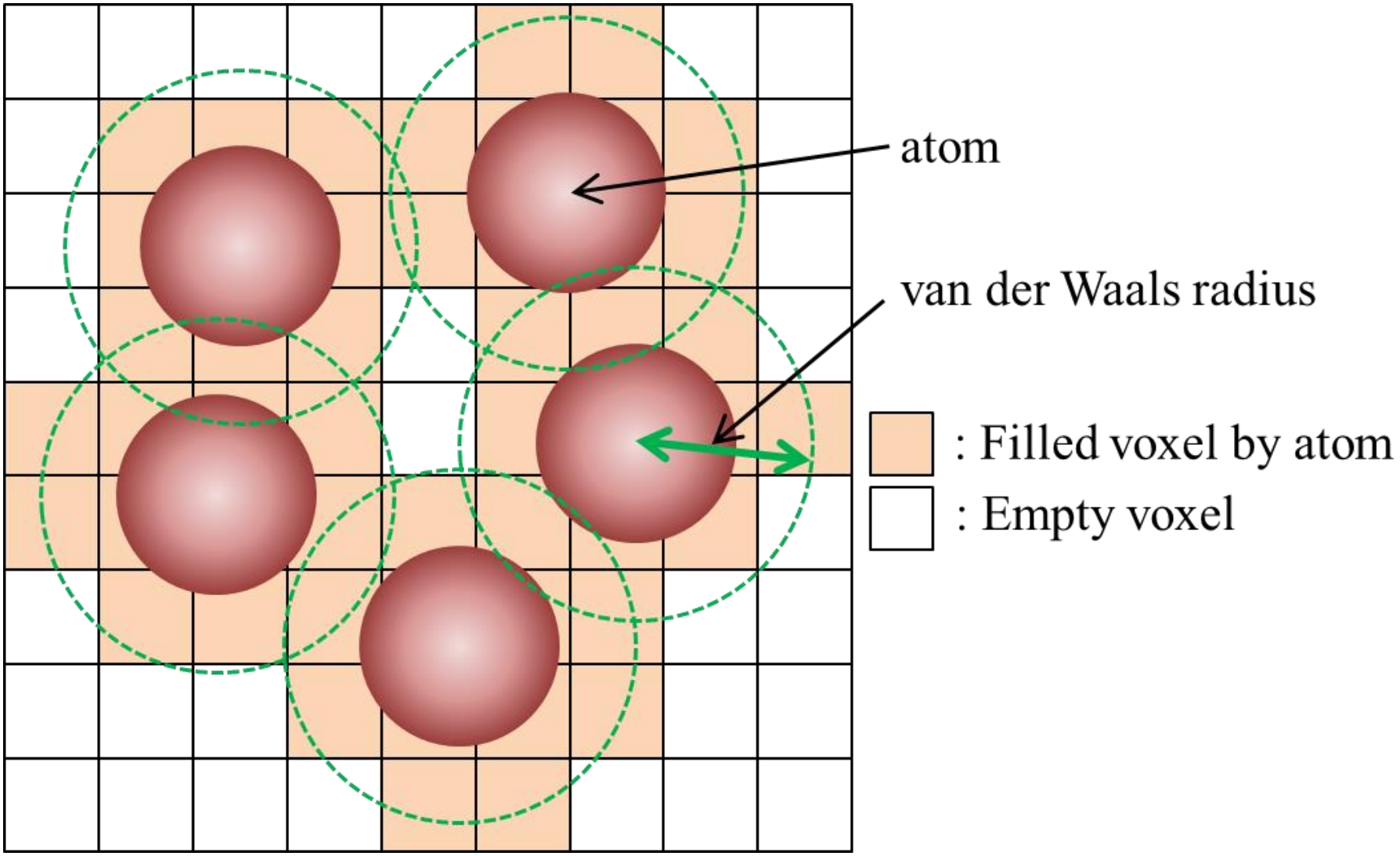
- **剛体探索**
  - タンパク質を剛体とみなして計算
  - 自由度が並進自由度と回転自由度のみ
    - 3次元並進と3次元回転の6次元空間探索
  - タンパク質の柔軟性を(あまり)考慮できないが,
    計算量を削減できる
    - スコア関数の設計である程度の柔軟性は考慮できる
- **グリッドに基づく構造探索**
  - 剛体探索で最も主流
    (Katchalski-Katzir+1997; Chen+2002; Mintseris+2007など)
  - タンパク質を3次元グリッド上で表現し,
    積和の形のスコア関数を計算する(以降のスライドで説明)



ドッキング計算の流れ


R
L
入力
Receptor　ボクセル化
Receptor　FFT
Ligand　回転
Ligand　ボクセル化
Ligand　FFT
複素畳込み
逆FFT
高評価値の構造を抽出
OpenMP
出力

グリッドへの当てはめ



グリッドに基づく剛体ドッキング法

①タンパク質のグリッドに
スコアを付与

レセプター　リガンド

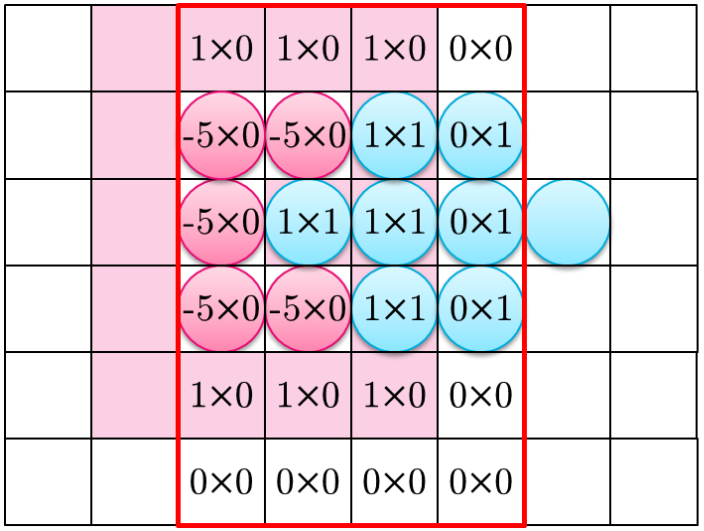
(↓実際は3次元)

0	1	1	1	1	0
0	1	-5	-5	1	0
0	1	-5	1	1	0
0	1	-5	-5	1	0
0	1	1	1	1	0
0	0	0	0	0	0

レセプター

0	0	0	0	0	0
0	0	1	1	0	0
0	1	1	1	1	0
0	0	1	1	0	0
0	0	0	0	0	0
0	0	0	0	0	0

リガンド

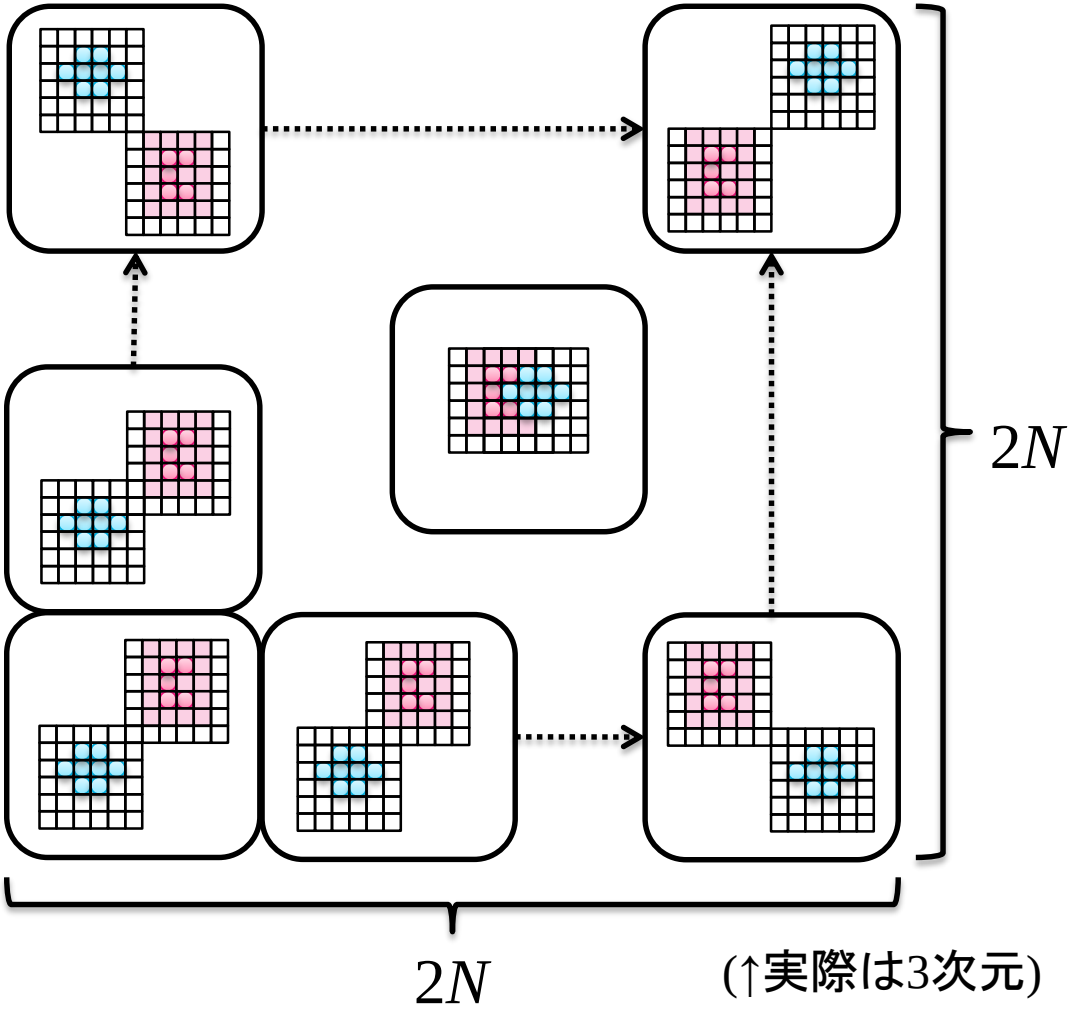


$$
\begin{aligned}
S=\ & 0+0+0+0 \\
& +0+0+1+0 \\
& +0+1+1+0 \\
& +0+0+1+0 \\
& +0+0+0+0 \\
& +0+0+0+0 \\
& =4
\end{aligned}
$$

②評価値の計算
→グリッドを重ねて積和

グリッドに基づく剛体ドッキング法

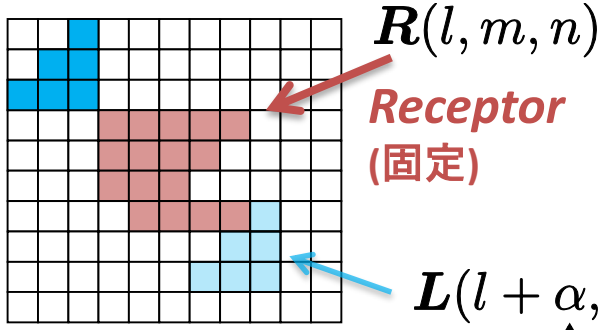
③リガンドを平行移動させて全配置の評価値を計算



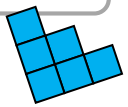
・グリッドの1辺の数:N
・平行移動パターン数:$8N^3$

グリッドに基づく剛体ドッキング法

- **Katchalski-Katzirアルゴリズム** [Katchalski-Katzir+1992]
  - 高速フーリエ変換(FFT)によるグリッド上の評価関数計算



3600通りの回転パターンで
並進探索を繰り返す



1回の並進探索の計算式

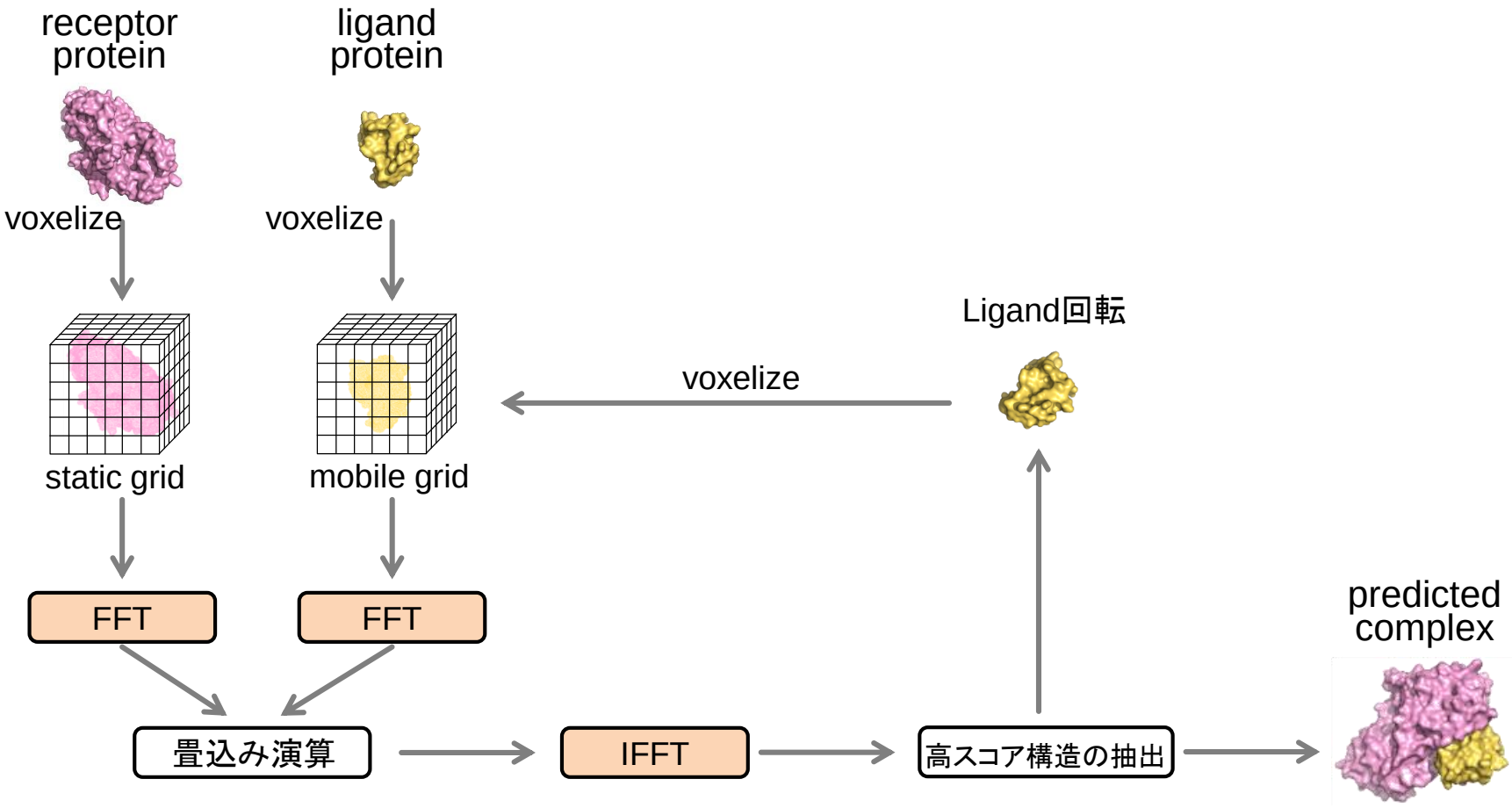
$$\text{評価関数} S(\alpha,\beta,\gamma)=\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\boldsymbol{R}(l,m,n)\boldsymbol{L}(l+\alpha,m+\beta,n+\gamma)$$

畳込み和（相関関数）$O(N^6)$

$$S(\alpha,\beta,\gamma)=\mathrm{IFFT}\left[\mathrm{FFT}[\boldsymbol{R}(l,m,n)]^{*}\mathrm{FFT}\left[\boldsymbol{L}(l,m,n)\right]\right]$$

FFTの利用　$O(N^3\log N)$

ドッキング処理の流れ


receptor protein
ligand protein
voxelize
voxelize
static grid
mobile grid
voxelize
Ligand回転
FFT
FFT
畳込み演算
IFFT
高スコア構造の抽出
predicted complex

スコア関数

- **タンパク質の結合自由エネルギーの要素をモデル化**
  - ファンデルワールス
  - クーロンポテンシャル
  - 溶媒和自由エネルギー　など
- **MEGADOCKのスコア関数**
  - ① 形状相補性 (rPSC関数)
  - ② 静電相互作用
  - ③ 統計的ペアワイズポテンシャル (atomic contact energy)
  - MEGADOCK-K (MEGADOCK 3.0)では①と②のみ
  - 最新版のMEGADOCK 4.0は①～③全て実装済

スコア関数の設計

- **単純に考えると要素の数だけ↓の関数が必要**

ZDOCK 3.0 (Mintseris+2007) の例

$$\mathrm{Shape}(\alpha,\beta,\gamma)=\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\boldsymbol{R}_{Shape}(l,m,n)\boldsymbol{L}_{Shape}(l+\alpha,m+\beta,n+\gamma)$$

$$\mathrm{Elec}(\alpha,\beta,\gamma)=\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\boldsymbol{R}_{Elec}(l,m,n)\boldsymbol{L}_{Elec}(l+\alpha,m+\beta,n+\gamma)$$

$$\mathrm{Desolv1}(\alpha,\beta,\gamma)=\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\boldsymbol{R}_{Desolv1}(l,m,n)\boldsymbol{L}_{Desolv1}(l+\alpha,m+\beta,n+\gamma)$$

$$\mathrm{Desolv2}(\alpha,\beta,\gamma)=\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\boldsymbol{R}_{Desolv2}(l,m,n)\boldsymbol{L}_{Desolv2}(l+\alpha,m+\beta,n+\gamma)$$

$$\vdots$$

$$\mathrm{Desolv6}(\alpha,\beta,\gamma)=\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\boldsymbol{R}_{Desolv6}(l,m,n)\boldsymbol{L}_{Desolv6}(l+\alpha,m+\beta,n+\gamma)$$

これらを最後に（重み付きで）合計した値を「探索部位の良さ」とする.

MEGADOCKのスコア関数

- **畳み込みは1回だけ**

$$S(\alpha,\beta,\gamma)=\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\boldsymbol{R}(l,m,n)\boldsymbol{L}(l+\alpha,m+\beta,n+\gamma)$$

- **グリッドに割り当てる値 (*R*()と*L*()) を工夫する**

MEGADOCK-K (MEGADOCK 3.0)

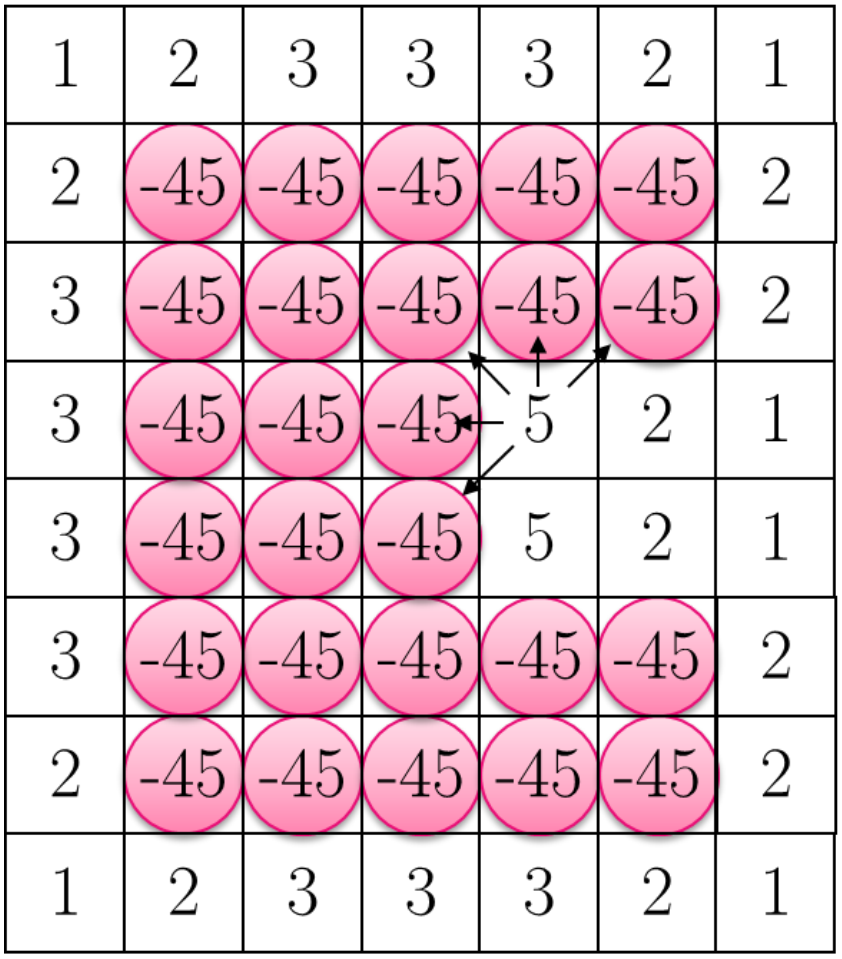
スコア関数 ＝ 形状相補性(rPSC) ＋i クーロン力

MEGADOCK 4.0

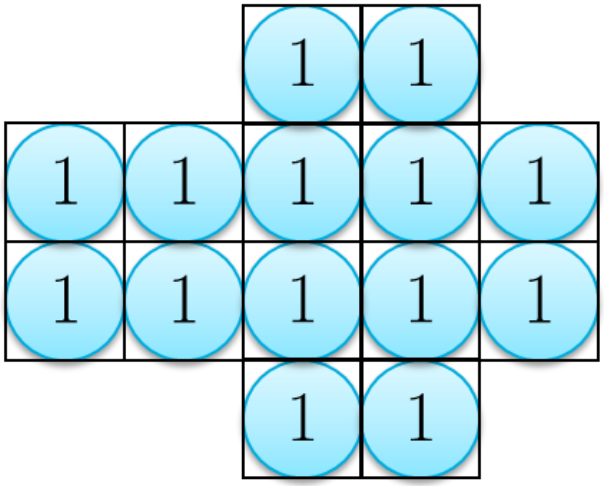
スコア関数 ＝ 形状相補性 (rPSC) | 脱溶媒和項 (RDE) ＋i クーロン力

– 畳み込み式が1つだけになるので，計算時間の削減が可能
– FFTWを用いた比較ではZDOCK3.0の約9.8倍の高速化を達成

形状相補性 (rPSC)



Receptor rPSC $R_{\mathrm{rPSC}}(l,m,n)$



Ligand rPSC $L_{\mathrm{rPSC}}(l,m,n)$

静電相互作用

- **静電相互作用項の計算方法**
  - grid $\boldsymbol{i} = (l, m, n)$の電荷 $q(l, m, n)$を決定する
    - 規則に従ってタンパク質原子の電荷を付与 $q_{\mathrm{atom}} \rightarrow$grid化 $q(l, m, n)$
  - grid $\boldsymbol{i} = (l, m, n)$の電界 $\varphi(\boldsymbol{i})$を計算する

$$\varphi(\boldsymbol{i}) = \sum_{\boldsymbol{j} \in \mathbb{N}^3} \frac{q(\boldsymbol{j})}{\varepsilon(r_{\boldsymbol{ij}}) r_{\boldsymbol{ij}}} \qquad \varepsilon(r) = \begin{cases} 4 & (r \leq 6\text{Å}) \\ 38r - 224 & (6\text{Å} < r < 8\text{Å}) \\ 80 & (r \geq 8\text{Å}) \end{cases}$$

$$E_R(l, m, n) = \begin{cases} \varphi(l, m, n) & \text{entire grid excluding core} \\ 0 & \text{core} \end{cases}$$

$$E_L(l, m, n) = q(l, m, n)$$

$$\mathrm{ELEC}(\alpha, \beta, \gamma) = \sum_{l,m,n} E_R(l, m, n) E_L(l + \alpha, m + \beta, n + \gamma)$$

rPSCと複素FFTの利用

$$\boldsymbol{R}(l,m,n) = R_{\mathrm{rPSC}}(l,m,n) + iR_{\mathrm{P}}(l,m,n)$$
$$\boldsymbol{L}(l,m,n) = L_{\mathrm{rPSC}}(l,m,n) - iL_{\mathrm{P}}(l,m,n)$$

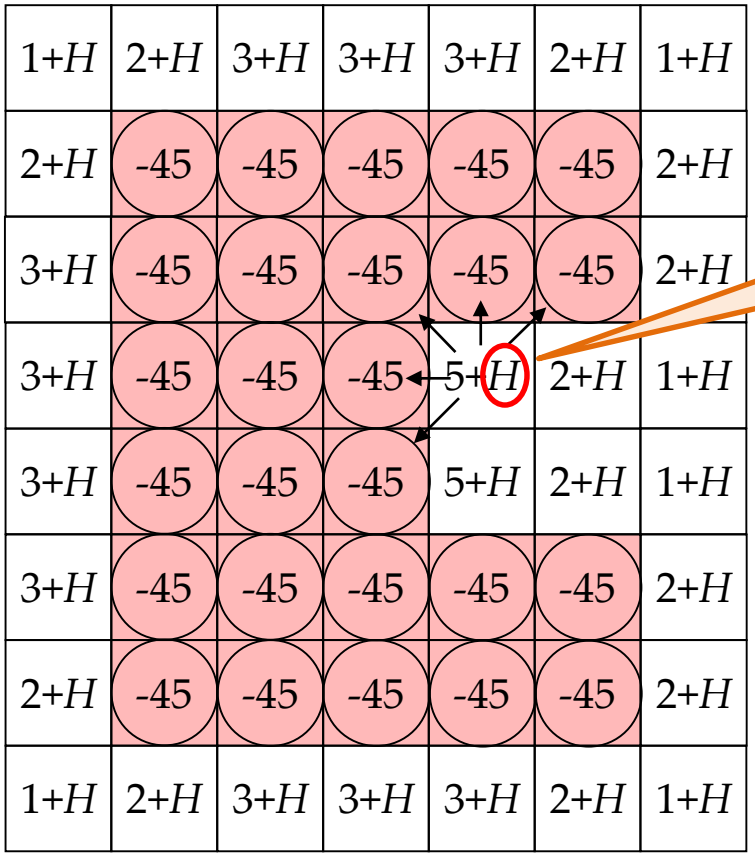
$$\boldsymbol{S}(\alpha,\beta,\gamma) = \Re\left[\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\boldsymbol{R}(l,m,n)\boldsymbol{L}(l+\alpha,m+\beta,n+\gamma)\right]$$
$$= \sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\{R_{\mathrm{rPSC}}(l,m,n)L_{\mathrm{rPSC}}(l+\alpha,m+\beta,n+\gamma)$$
$$+ R_{\mathrm{P}}(l,m,n)L_{\mathrm{P}}(l+\alpha,m+\beta,n+\gamma)\}$$
$$= S_{\mathrm{rPSC}}(\alpha,\beta,\gamma) + S_{\mathrm{P}}(\alpha,\beta,\gamma)$$

他の物理化学的相互作用のスコアを計算時間を増加させずに導入可能
（標準的なFFTライブラリは複素FFTを扱うので計算時間を増加させずに同時に計算可能）

→MEGADOCK-KではPの部分に静電相互作用を導入

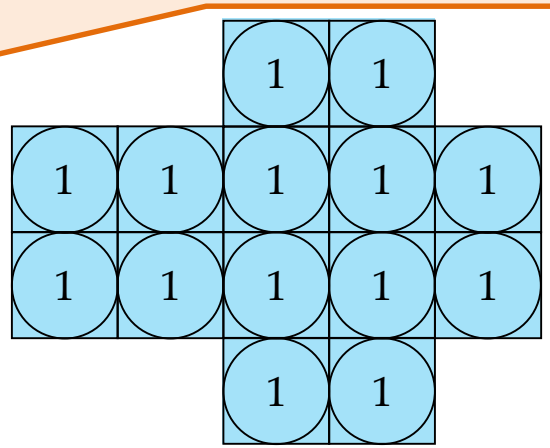
脱溶媒和項 (RDE)

- **Atomic Contact Energy (ACE) による統計的ペアワイズポテンシャルを用いて脱溶媒和自由エネルギーを推定**



Receptor

*H*の部分に周囲の原子のACE値を格納しておく.
Ligandの“1”との積により, Ligand原子との
ペアワイズポテンシャルが畳み込み演算で計算可能.



Ligand

$$H(l,m,n) = \begin{cases} \text{周囲のACE値の和} & \text{(外部)} \\ 0 & \text{(内部)} \end{cases}$$

MEGADOCKのスコア関数

$$R(l,m,n) = R_{\mathrm{rPSC+RDE}}(l,m,n) + iR_{\mathrm{ES}}(l,m,n)$$
$$= R_{\mathrm{rPSC}}(l,m,n) + w_{\mathrm{DE}}R_{\mathrm{RDE}}(l,m,n) + iR_{\mathrm{ES}}(l,m,n)$$
$$L(l,m,n) = L_{\mathrm{rPSC}}(l,m,n) - iw_{\mathrm{ES}}L_{\mathrm{ES}}(l,m,n)$$

$$S(\alpha,\beta,\gamma) = \Re\left[\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N} R(l,m,n)L(l+\alpha,m+\beta,n+\gamma)\right]$$

スコア関数

$$= \sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}\{R_{\mathrm{rPSC}}(l,m,n)L_{\mathrm{rPSC}}(l+\alpha,m+\beta,n+\gamma)$$
$$+ w_{\mathrm{DE}}R_{\mathrm{RDE}}(l,m,n)L_{\mathrm{rPSC}}(l+\alpha,m+\beta,n+\gamma)$$
$$+ w_{\mathrm{ES}}R_{\mathrm{ES}}(l,m,n)L_{\mathrm{ES}}(l+\alpha,m+\beta,n+\gamma)\}$$

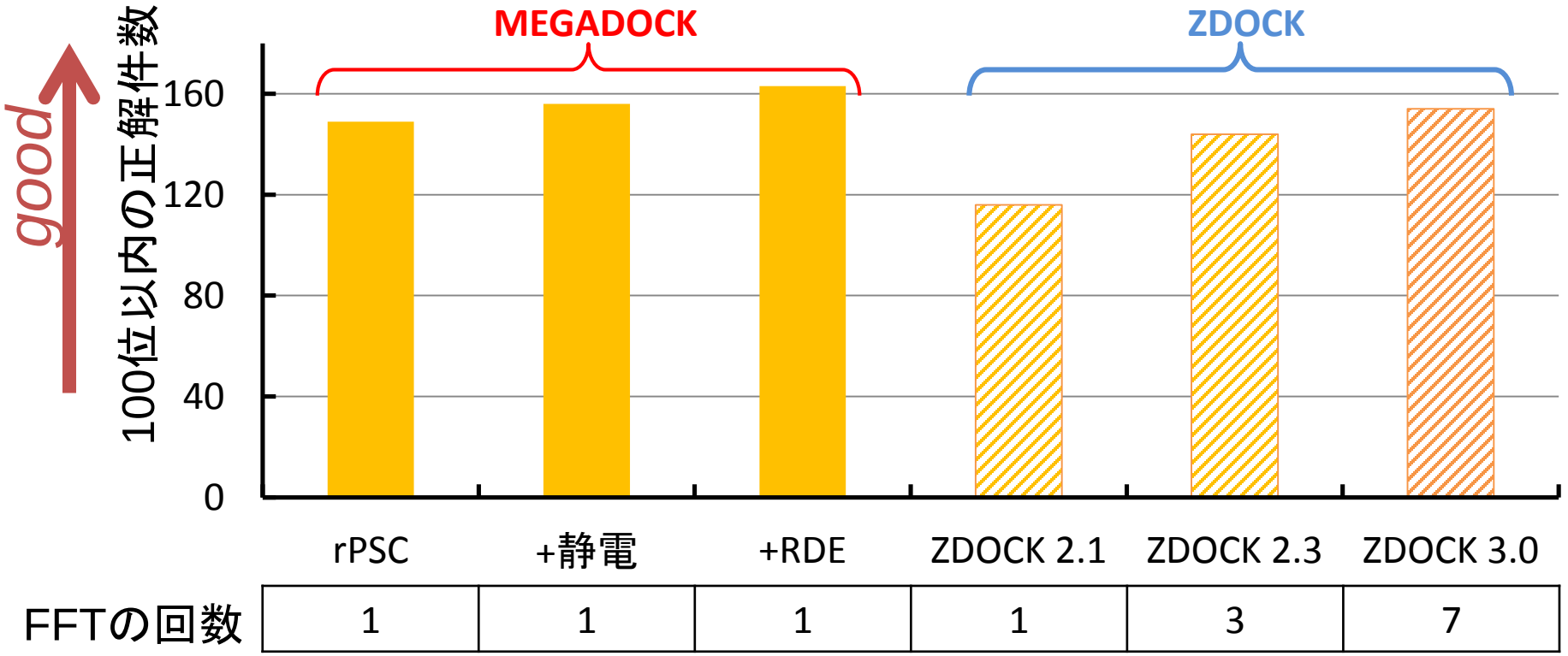
$$= S_{\mathrm{rPSC}}(\alpha,\beta,\gamma) + w_{\mathrm{DE}}S_{\mathrm{RDE}}(\alpha,\beta,\gamma) + w_{\mathrm{ES}}S_{\mathrm{ES}}(\alpha,\beta,\gamma)$$

形状相補性　　脱溶媒和自由エネルギー　　静電相互作用

1回の畳込み演算で3つの要素を同時に計算可能

ドッキング計算の性能

Protein-Protein Docking Benchmark version 4.0 (bound set) を用いた性能比較

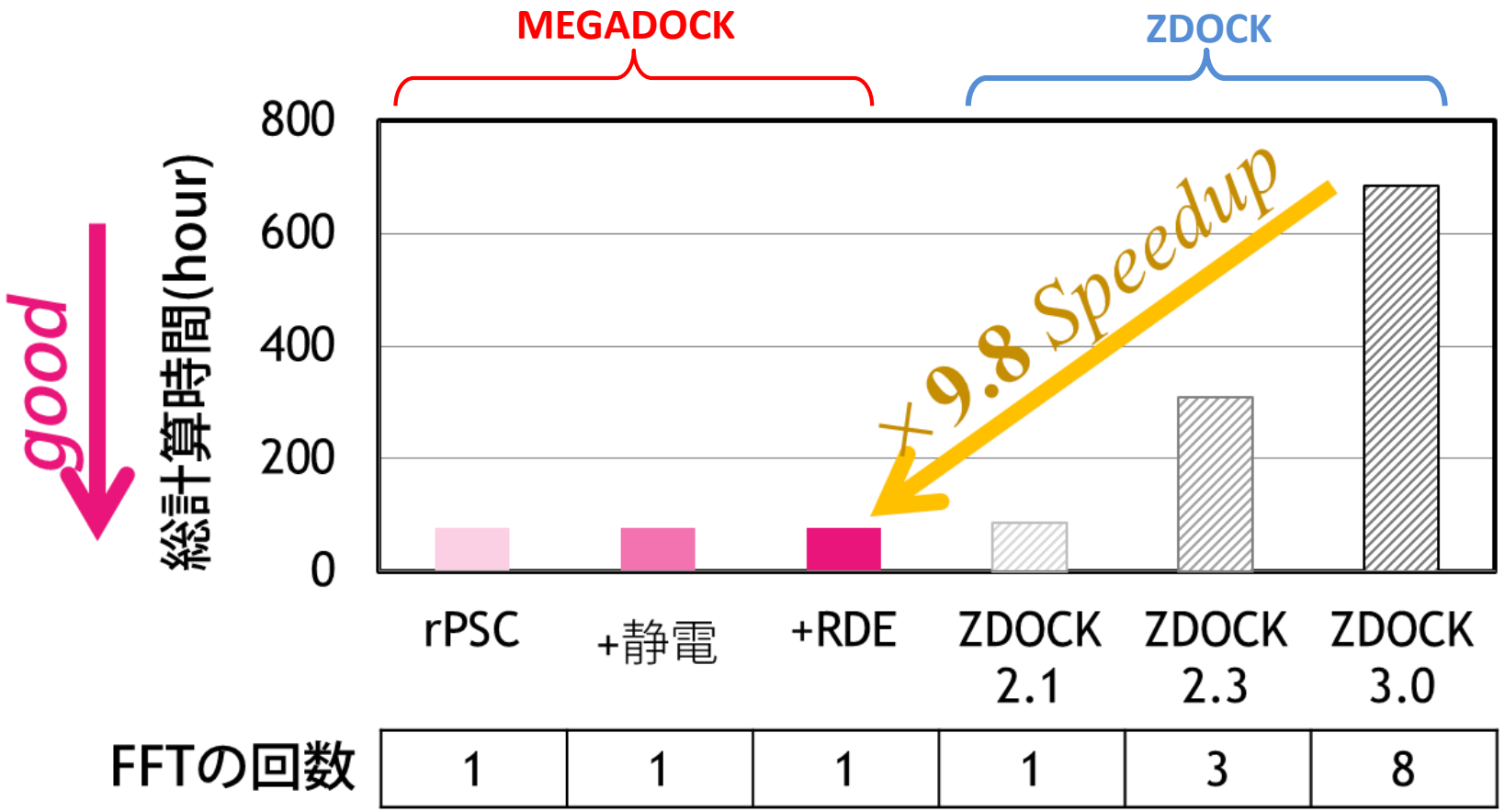


従来のドッキング計算ツール(ZDOCK)と
同程度のドッキング予測精度を実現

* ZDOCK 2.1…形状
ZDOCK 2.3…形状+静電+溶媒和
ZDOCK 3.0…形状+静電+溶媒和

評価実験ー計算時間の比較

計算環境：Intel Xeon X5670 2.93 GHz (1コア使用)
測定対象：Docking Benchmark 4.0のドッキング計算にかかった総計算時間



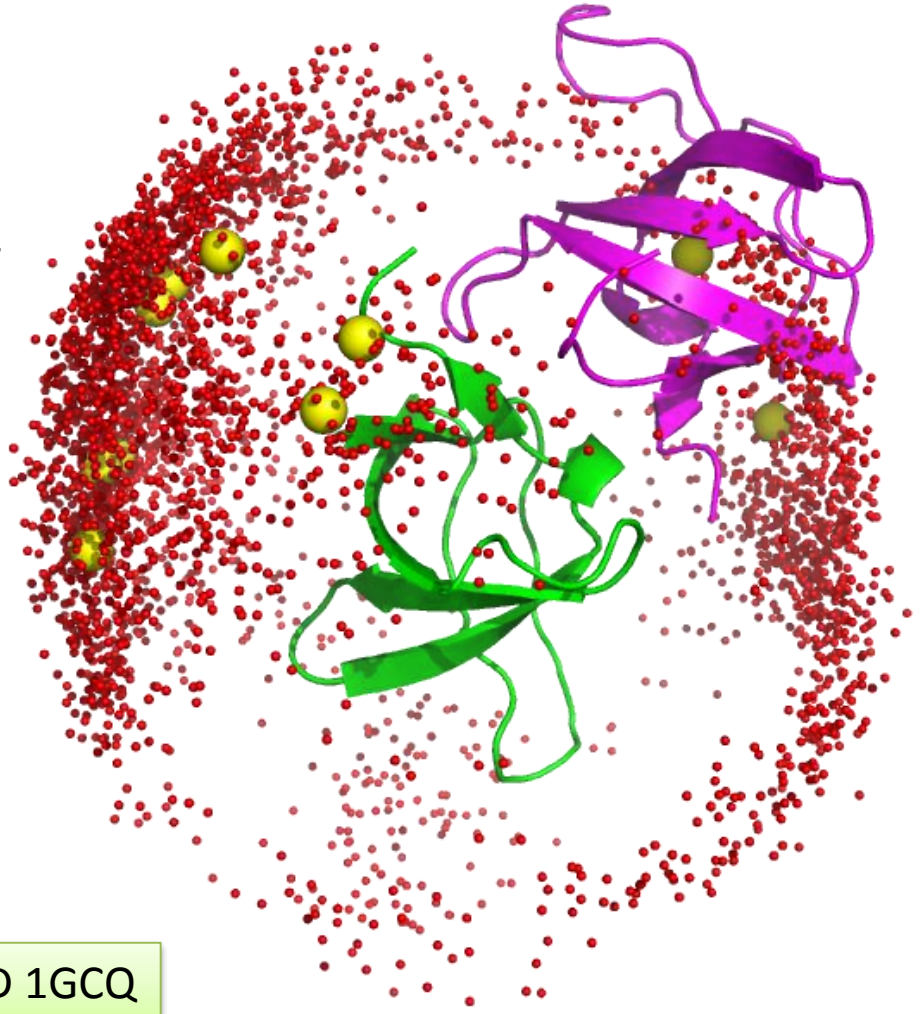
従来手法と同等精度を維持しつつ約9.8倍の高速化を達成

実際に出てくるもの

MEGADOCKの出した答えを可視化

緑：レセプタータンパク質，
マゼンタ：リガンドタンパク質の正しい位置

赤色の点：上位2000個の答えの重心位置
黄色の点：上位10個の答えの重心位置



タンパク質：PDB ID 1GCQ

MEGADOCK-K 講習会の概要

- 開発の背景
- 「京」をはじめとする高性能並列計算機上での実行を想定した並列化と性能評価
- ドッキングスコア関数の開発
- MEGADOCK-K実行の流れ
- MEGADOCK-Kを応用したタンパク質間相互作用予測研究の例

References

- **MEGADOCK評価関数（rPSC形状相補性項）**
  - Ohue M, *et al*., **MEGADOCK: An all-to-all protein-protein interaction prediction system using tertiary structure data**, *Protein Pept Lett*, 21(8): 766-778, 2014.
- **MEGADOCK評価関数（RDE脱溶媒和項）**
  - Ohue M, *et al*., **Improvement of the protein-protein docking prediction by introducing a simple hydrophobic interaction model: an application to interaction pathway analysis**, *Lecture Notes in Comput Sci*, 7632: 178-187, 2012.
- **MEGADOCK for Supercomputers (MEGADOCK-K / MEGADOCK 3.0)**
  - Matsuzaki Y, *et al*., **MEGADOCK 3.0: A high-performance protein-protein interaction prediction software using hybrid parallel computing for petascale supercomputing environments**, *Source Code for Biol Med*, 8(1): 18, 2013.
- **MEGADOCK for NVIDIA GPU Accelerators (MEGADOCK-GPU)**
  - Shimoda T, *et al*., **MEGADOCK-GPU: acceleration of protein-protein docking calculation on GPUs**, In *Proc of ACM-BCB 2013*, 884-890, 2013.
- **MEGADOCK for GPU supercomputers (MEGADOCK 4.0)**
  - Ohue M, *et al*., **MEGADOCK 4.0: an ultra–high-performance protein–protein docking software for heterogeneous supercomputers**, *Bioinformatics*, 30(22): 3281-3283, 2014.
- **MEGADOCK for Intel Xeon Phi Accelerators**
  - Shimoda T, *et al*., **Protein-protein docking on hardware accelerators: comparison of GPU and MIC architectures**, *BMC Systems Biology*. (in press)

MEGADOCKに関する参考資料

- **ホームページ**

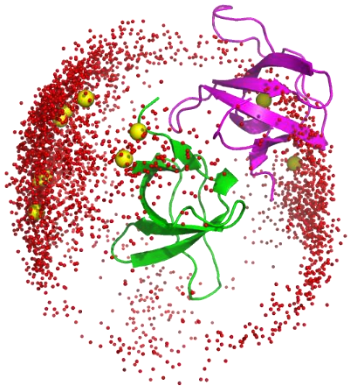
  MEGADOCK 4.0 - Akiyama Lab.
  http://www.bi.cs.titech.ac.jp/megadock/

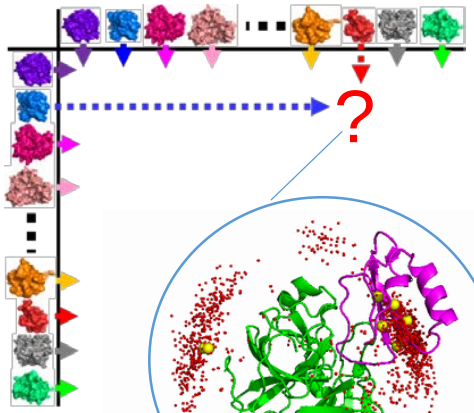
- **MEGADOCKのチュートリアル**

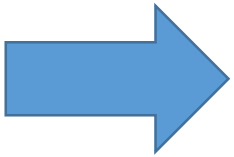
  MEGADOCKでタンパク質ドッキングをやってみる – Qiita
  http://qiita.com/tonets/items/e95eb937f224064e67a3

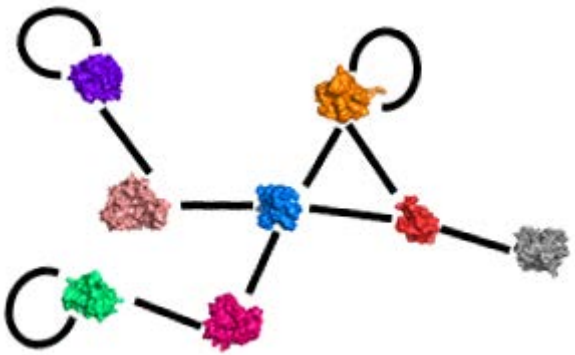
- **右の絵を描く方法**

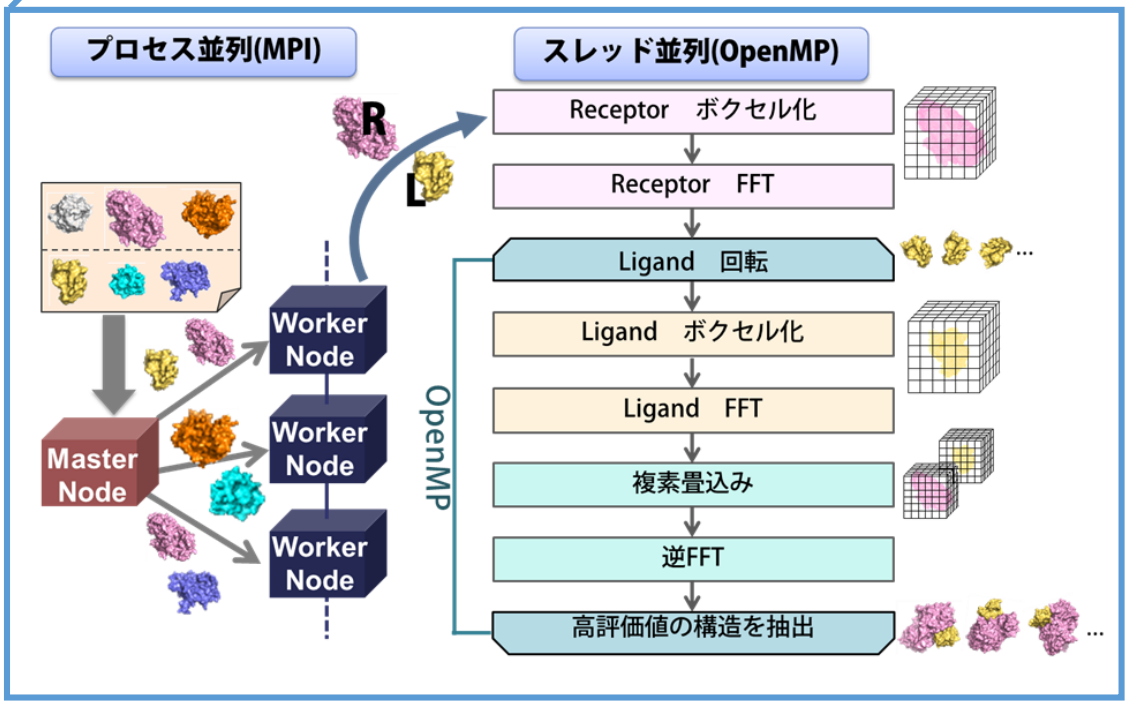
  decoy ligand重心の分布図を作る
  http://goo.gl/p1AR20











休憩後:

- 「京」での実行例
- 実問題への応用例

MEGADOCK-K 講習会の概要

- 開発の背景
- 「京」をはじめとする高性能並列計算機上での実行を想定した並列化と性能評価
- ドッキングスコア関数の開発
- MEGADOCK-K実行の流れ
- MEGADOCK-Kを応用したタンパク質間相互作用予測研究の例

MEGADOCK-K 実行の流れ

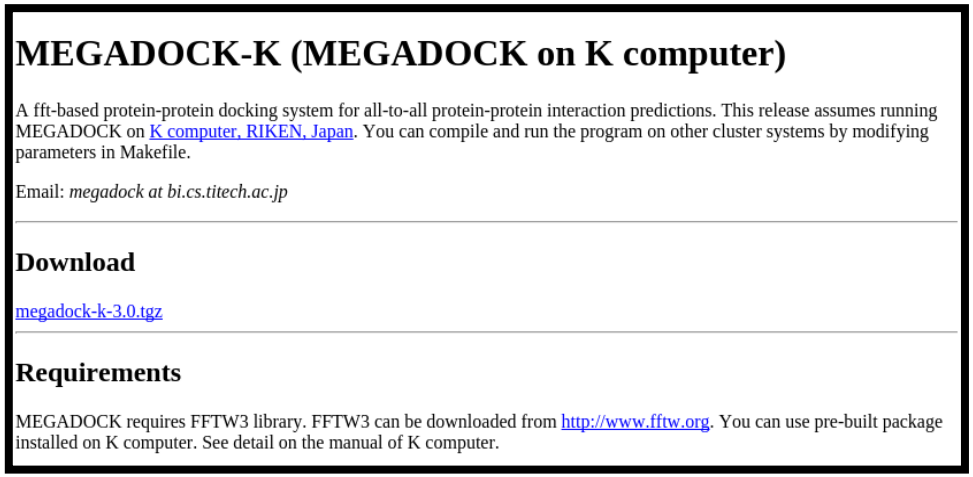
「京」での実行例
（ジョブ実行の流れはSCLSと異なる）

実行の流れ

1. ダウンロード
2. コンパイル
3. スレッド並列版の実行 (interactive job)
4. ハイブリッド並列版の実行
    1. ジョブテーブルの作成
    2. ジョブ投入スクリプトの作成
    3. ジョブ実行
    4. ジョブ実行結果確認
5. ドッキング実行結果
6. ポストドッキング解析
    1. 複合体モデルの作成
    2. 高スコア複合体モデルの分布
    3. 相互作用ペア予測

1. ダウンロード

http://www.bi.cs.titech.ac.jp/megadock/k/



MEGADOCK-K (MEGADOCK on K computer)

A fft-based protein-protein docking system for all-to-all protein-protein interaction predictions. This release assumes running MEGADOCK on K computer, RIKEN, Japan. You can compile and run the program on other cluster systems by modifying parameters in Makefile.

Email: *megadock at bi.cs.titech.ac.jp*

Download

megadock-k-3.0.tgz

Requirements

MEGADOCK requires FFTW3 library. FFTW3 can be downloaded from http://www.fftw.org. You can use pre-built package installed on K computer. See detail on the manual of K computer.

「京」
SCLS
(Fujitsu FX)
FFTW 倍精度利用

- FFTW3 が必要。「京」では /home/apps/fftw/ 以下から利用可能。
- GPUやIntel用のバージョンもあり
  ⇒http://www.bi.cs.titech.ac.jp/megadock/

2. コンパイル

- 環境変数の読みこみ
  ```
  $ source /home/system/Env_base
  ```
- ダウンロードファイルの解凍
  ```
  $ tar xzf megadock-k-3.0.tgz
  ```
- コンパイル
  ```
  $ cd megadock-k-3.0
  $ make
  ```
  - MPI+OpenMP 版が作成される
    - `megadockdp`
  - USE_MPI 変数を定義しないと単一ノードで動作するスレッド並列版が作成される
    - `megadock`
- `-Kfast` を利用する場合オプションに注意(Makefile 参照)

3. スレッド並列版の実行 <1>
（interactive job）

MEGADOCK

- 必須のコマンドライン引数
  - -R [receptor PDBファイル]
  - -L [ligand PDBファイル]
- 主なオプション
  - -N [出力するドッキングポーズ数]
  - -o [出力ファイル名]
  - -t [リガンド一回転あたりのポーズ数]

3. スレッド並列版の実行 <2>（interactive job）

MEGADOCK

- Interactive job のリクエスト

```
$ pjsub  --interact  --rsc-list "node=1"  --rsc-list "node-mem=12Gi" --rsc-list  "elapse=0:10:0"  --sparam "wait-time=3600"
[INFO] PJM 0000 pjsub Job 2625504 submitted.
[INFO] PJM 0081 ..........connected.
[INFO] PJM 0082 pjsub Interactive job 2625504 started.
```

- 環境変数の設定
  - スレッド数

```
$ export OMP_NUM_THREADS=8
```

  - 「京」での実行に必要な環境変数設定

```
$ source /work/system/Env_base
Env_base: K-1.2.0-15
```

3. スレッド並列版の実行 <3> （interactive job）

MEGADOCK

- Docking 実行 (例: receptor.pdb と ligand.pdb)

```
$ ./megadock -R receptor.pdb -L ligand.pdb -N 2000 -t 3
#Using OpenMP parallelization: 8 threads.
#Number of output = 2000
#Set number of scores per one angle = 3
#Receptor = receptor.pdb
#Ligand   = ligand.pdb
#Output file = receptor-ligand.out
(略)
```

- Interactive job の終了

```
$ exit
exit
[INFO] PJM 0083 pjsub Interactive job 262■■■■ completed.
```

3. スレッド並列版の実行 <4>（interactive job）

MEGADOCK

- Docking 結果確認

```
$ ls *.out
receptor-ligand.out
```

  - （例）receptor.pdb と ligand.pdb をドッキングすると receptor-ligand.out という結果ファイルが生成される
  - -o オプションで他の出力ファイル名を指定可能

```
144     1.20
        0.000000        0.000000        0.000000
../DenV//3VWS_A.pdb     23.844501       56.637497       16.135000
../H_sapiens//10GS_A.pdb        18.558500       -0.606000       21.075001
0.523599        1.310100        1.488307        9       115     133     4369.69
0.523599        0.961715        0.014085        136     137     30      4259.43
-3.141593       1.870545        -2.976234       34      133     137     4205.36
0.785398        1.870545        -2.976234       9       125     36      4177.86
0.523599        1.310100        1.488307        9       114     132     4170.33
```

4. ハイブリッド並列版の実行

MEGADOCK-DP

- スレッド並列版と同様のドッキングタスクを多数実行する場合にはハイブリッド並列版を用いる
- 必須のコマンドライン引数
  - –tb [docking job table file]

- 必要なファイル
  - ジョブテーブル (ドッキングジョブのリスト)
  - 「京」の計算ジョブを定義するスクリプト

4. ハイブリッド並列版の実行
ジョブテーブルの作成 <1>

MEGADOCK-DP

(例) ヒトのタンパク質3万構造と
Dengue virus の4つのタンパク質間のドッキング

1. 各タンパク質群の構造ファイルリストを作成

```
12GS_B.pdb
121P_A.pdb
:
:
```

human.txt
(ヒト pdb リスト)

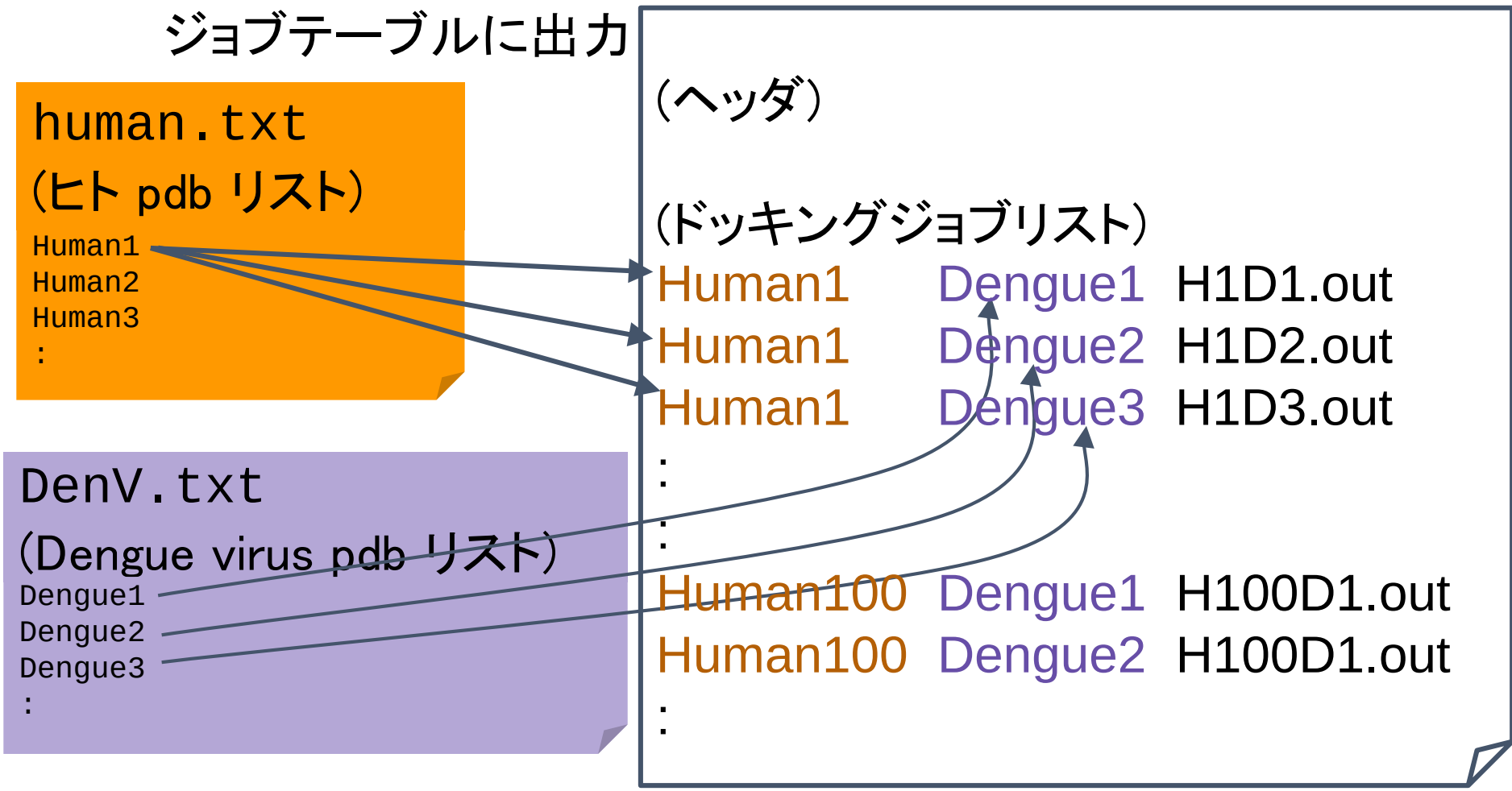
```
3VWS_A.pdb
3VWS_B.pdb
:
:
```

DenV.txt
(Dengue virus pdb リスト)

4. ハイブリッド並列版の実行
ジョブテーブルの作成 <2>

MEGADOCK-DP

2. 2つのファイルにある .pdb の全くみあわせを
   ジョブテーブルに出力



4. ハイブリッド並列版の実行
ジョブテーブルの作成 〈3〉

MEGADOCK-DP

- テーブルヘッダ例

```
##megadock のオプションと続くデータの###
##カラムの対応を定義##
TITLE=Denv-Human Docking
PARAM=-R $1 -L $2 -o $3
```

- ドッキングジョブリスト
  - ヘッダのPARAMに指定した順にタブ区切りで記載
  - 例

```
../rec.pdb../lig.pdb    ../out/rec-lig.out
```

4. ハイブリッド並列版の実行
ジョブテーブルの作成 <4>

MEGADOCK-DP

- ドッキングジョブリスト
  - MEGADCOK-DPはジョブテーブルの上から順番に
    ワーカーノードにドッキング計算を割りふる
  - FFTサイズの大きいものから並べるとよい
  - サイズを調べてテーブルを作成するスクリプトなどを用意するとよい

<例>

```
$ head table.list
TITLE=megadock docking list
PARAM=-R $1 -L $2 -o $3
../Denv/3VWS_A.pdb        ../H_sapiens/134L_A.pdb   ../out/3VWS_A/3VWS_A-134L_A.out
../Denv/3VWS_A.pdb        ../H_sapiens/133L_A.pdb   ../out/3VWS_A/3VWS_A-133L_A.out
../Denv/3VWS_A.pdb        ../H_sapiens/12GS_B.pdb   ../out/3VWS_A/3VWS_A-12GS_B.out
../Denv/3VWS_A.pdb        ../H_sapiens/12GS_A.pdb   ../out/3VWS_A/3VWS_A-12GS_A.out
../Denv/3VWS_A.pdb        ../H_sapiens/12CA_A.pdb   ../out/3VWS_A/3VWS_A-12CA_A.out
../Denv/3VWS_A.pdb        ../H_sapiens/121P_A.pdb   ../out/3VWS_A/3VWS_A-121P_A.out
../Denv/3VWS_A.pdb        ../H_sapiens/11GS_B.pdb   ../out/3VWS_A/3VWS_A-11GS_B.out
../Denv/3VWS_A.pdb        ../H_sapiens/11GS_A.pdb   ../out/3VWS_A/3VWS_A-11GS_A.out
```

4. ハイブリッド並列版の実行
ジョブ投入スクリプトの作成 <1>

MEGADOCK-DP

ジョブスクリプトの構成

- ジョブマネージャ向けの記述
  - ジョブの属性
  - 計算ノードへのファイル転送（Stage-in, Stage-out）の指定
- 計算ノードでの前処理
- megadockdp の実行
- 計算ノードでの後処理

4. ハイブリッド並列版の実行
ジョブ投入スクリプトの作成 〈2〉

MEGADOCK-DP

ジョブマネージャ向けの記述部分

- **ジョブマネージャ向けの記述**
- 計算ノードでの前処理
- megadockdp の実行
- 計算ノードでの後処理

```
#!/bin/bash
###Job の属性###
#PJM --rsc-list "node=128"          ←確保するノード数
#PJM --rsc-list "elapse=1:00:00"    ←実行時間の上限
#PJM --rsc-list "rscgrp=small"      ←リソースグループ
#PJM --gname "hp120■■■"             ←実行者のグループ
#PJM --mpi "use-rankdir"            ←Rank ディレクトリを使用する指定
#PJM --mpi "assign-online-node"     ←稼動中ノードのみ使用

###計算ノードへのファイル転送指定(stgin/stgout): プログラム、入力、出力データ###
#PJM --stg-transfiles all
#PJM --stgin "rank=0 /data/hp120■■■/k0■■■■/megadock/megadockdp 0:../megadockdp"
#PJM --stgin "rank=0 human.tgz 0:../human.tgz"
#PJM --stgin-dir "rank=0 /data/hp120■■■/k0■■■■/PDB/DenV/ 0:../DenV/"
#PJM --stgin "rank=0 ./table.list %r:./table.list"
#PJM --stgin "rank=0 ./pdblist.txt %r:./pdblist.txt"
#PJM --stgout "rank=0 ../out/*.bz2 /data/hp120■■■/k0■■■■/job%j/ stgout=all"
```

4. ハイブリッド並列版の実行
ジョブ投入スクリプトの作成 <3>

MEGADOCK-DP

計算ノードでの前処理部分

```
###環境変数の読み込み###
. /work/system/Env_base

###ジョブテーブル・実行ファイルの指定###
### (stg-in したもの) ###
TABLE=table.list

PROGRAM_NAME=megadockdp
OUT_PATH=../out

###スレッド並列数の指定###
export OMP_NUM_THREADS=8
```

- ジョブマネージャ向けの記述
- **計算ノードでの前処理**
- megadockdpの実行
- 計算ノードでの後処理

```
###結果格納ディレクトリの作成###
for i in `cat pdblist.txt`;
do
   mkdir -p $OUT_PATH/$i
done

###stg-in したファイルの解凍###
pushd ../
tar xzf human.tgz
popd
```

4. ハイブリッド並列版の実行
ジョブ投入スクリプトの作成 <4>

MEGADOCK-DP

アプリケーション実行・後処理部分

- ジョブマネージャ向けの記述
- 計算ノードでの前処理
- **megadockdpの実行**
- **計算ノードでの後処理**

```
###megadockdp の実行###
mpiexec ../$PROGRAM_NAME -rt 0 -tb $TABLE -t 3 -N 2000

###実行終了後、保存された出力ファイルを圧縮###
pushd  ../out
for i in */; do tar cjf ${i/¥//.tar.bz2} $i; fi; done
popd
```

4. ハイブリッド並列版の実行 ジョブ実行

MEGADOCK-DP

pjsub コマンドを使ってジョブを投入する

```
$ pjstgchk run.sh
0 error(s).
14 files (1,160,673,091 bytes) will be
transferred for stage-in.
$ pjsub run.sh
[INFO] PJM 0000 pjsub Job 262[redacted] submitted.
```

```
$ pjstat
```

(実行状況の確認)

4. ハイブリッド並列版の実行 ジョブ実行結果確認

MEGADOCK-DP

- Job 実行結果ファイル例（スクリプト名.[eios]job番号　ファイル）
  - run.sh.e262■■■■ ： 標準エラー出力
  - run.sh.i262■■■■ ： ジョブ統計情報
  - run.sh.o262■■■■ ： 標準出力
  - run.sh.s262■■■■ ： ステージング情報
- Stage out 結果を確認

```
$ ls /data/[Project ID]/[User ID]/[Job ID]
3VWS_A.tar.bz2 ....
$ tar tjf/data/[Project ID]/[User ID]/[Job ID]/3VWS_A.tar.bz2
3VWS_A/
3VWS_A/3VWS_A-121P_A.out …...
```

MEGADOCK-DP

MEGADOCK

5. ドッキング計算結果
＞＞結果ファイル例

3VWS_A-10GS_A.out

```
144    1.20
```
FFT size / Grid size [angstrom]

```
0.000000    0.000000    0.000000
```
初期位置からの回転（x,y,z 方向）

```
../DenV//3VWS_A.pdb    23.844501    56.637497    16.135000
```
Receptor PDB file / Receptor PDB 初期位置の中心座標

```
../H_sapiens//10GS_A.pdb    18.558500    -0.606000    21.075001
```
Ligand PDB file / Ligand PDB 初期位置の中心座標

予測複合体モデル
（スコア降順 / 上位N点）

```
0.523599    1.310100    1.488307    9    115    133    4369.69
```
予測複合体におけるLigandの初期位置からの 回転角（オイラー角）/ Receptor は初期位置 / Ligandの初期位置からの平行移動（グリッド数） / ドッキングスコア

```
0.523599    0.961715    0.014085    136    137    30    4259.43
-3.141593   1.870545    -2.976234   34     133    137   4205.36
0.785398    1.870545    -2.976234   9      125    36    4177.86
0.523599    1.310100    1.488307    9      114    132   4170.33
```

よくあるつまづき

- リソースグループを間違える
- 実行時間内にジョブが終わらない
- 必要なファイルのStage-inを忘れる
- Stage-in したファイルと実際にプログラムが読み込もうとするファイルが違う
- 結果ファイルの Stage-out を忘れる
- 結果ファイルを保存するディレクトリがない
- Stage-in/out 指定したファイルが存在しない
  - pjstgchk run.sh などで確認する

6. ポストドッキング解析

1. 複合体モデルの作成例

MEGADOCK-DP

MEGADOCK

```
$ ./decoygen lig.1.pdb ligand.pdb docking.out 1
```

ドッキングスコア1位の情報をもとに

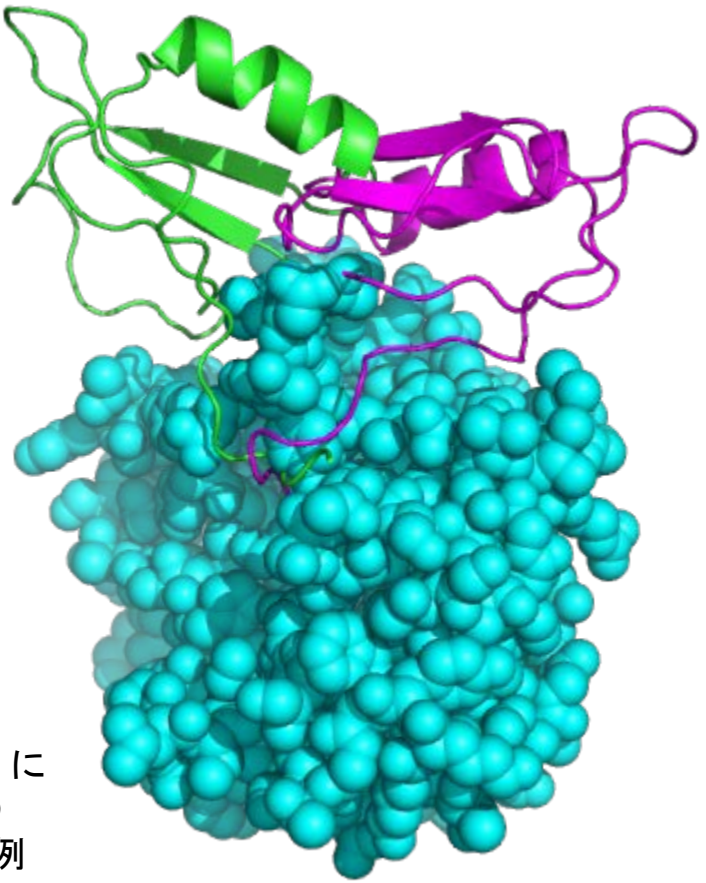
ligand pdb に回転・平行移動を加え

lig.1(順位).pdb を生成

```
$ pymol receptor.pdb lig.1.pdb
```

PDB のビューアを用いて

予測複合体を見る

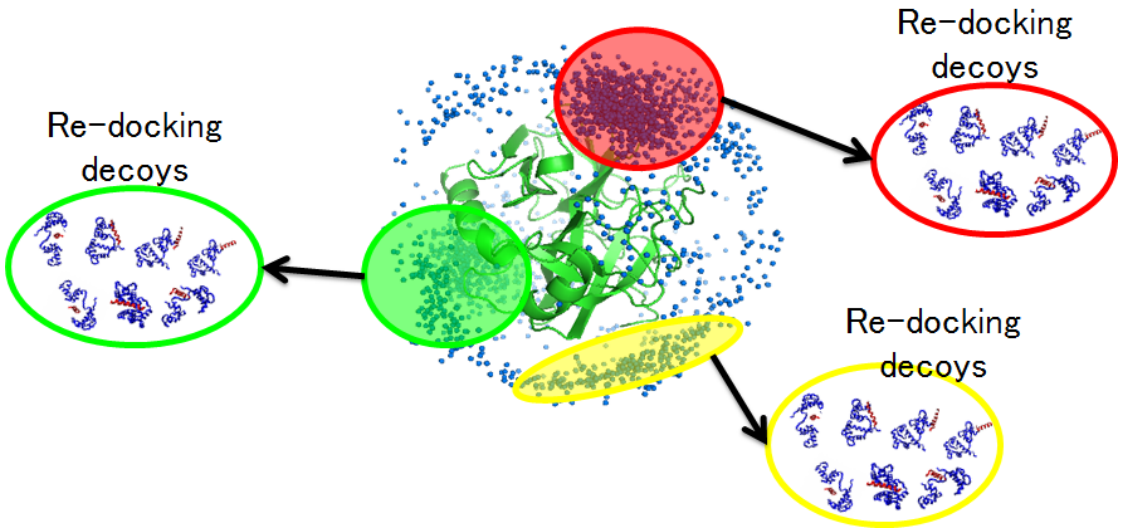


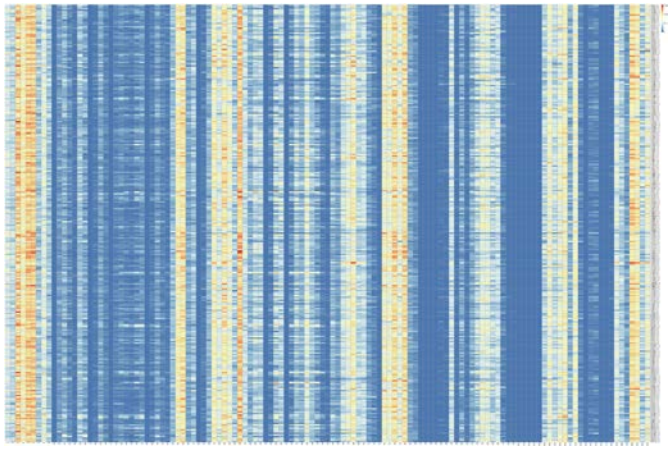
Receptor (sphere 表示) に
対して Ligand の複数の
結合位置候補を示した例

6. ポストドッキング解析

2. 高スコア複合体モデルの分布

- 正しい複合体構造を得るには高いスコアの複合体、モデル2,000程度を見るのが一般的
- 高スコアモデルの分布から情報を得られないか
  - クラスタリングを用いたPPI予測
  - 相互作用プロファイル解析
  - リドッキング





6. ポストドッキング解析

3. ドッキング結果からPPI評価値を計算し相互作用ペアを予測

MEGADOCK Result

Title: Benchmark2.0

Data: 2009/08/19

benchmark	PDB	UniProt	Description
1ACB_r	1ACB_I	P01051	Eglin C;
1AK4_r	1AK4_D	P12497	Gag-Pol polyprotein;AltName: Pr160Gag-P
1ATN_r	1ATN_D	P00639	Deoxyribonuclease-1;EC=3.1.21.1;AltName:
1AVX_r	1AVX_B	P01070	Trypsin inhibitor A;AltName: Kunitz-typ
1AY7_r	1AY7_B	P11540	Barstar;AltName: Ribonuclease inhibitor
1B6C_r	1B6C_B	P36897	TGF-beta receptor type-1;Short=TGFR-1;EC
1BUH_r	1BUH_B	P61024	Cyclin-dependent kinases regulatory subu
1BVN_r	1BVN_T	P01092	Alpha-amylase inhibitor HOE-467A;AltName
1CGI_r	1CGI_I	P00995	Pancreatic secretory trypsin inhibitor;A
1D6R_r	1D6R_I	P01055	Bowman-Birk type proteinase inhibitor;Sh
1DFJ_r	1DFJ_I	P10775	Ribonuclease inhibitor;AltName: Ribonuc
1E6E_r	1E6E_B	P00257	Adrenodoxin, mitochondrial;AltName: Adr
1E96_r	1E96_B	P19878	Neutrophil cytosol factor 2;Short=NCF-2;
1EAW_r	1EAW_B	P00974	Pancreatic trypsin inhibitor;AltName: B
1EWY_r	1EWY_C	P0A3C8	Ferredoxin-1;AltName: Ferredoxin I;
1F34_r	1F34_B	P19400	Major pepsin inhibitor 3;Short=PI-3;Flag
1FC2_r	1FC2_D	P01857	Ig gamma-1 chain C region;
1FQ1_r	1FQ1_B	P24941	Cell division protein kinase 2;EC=2.7.11

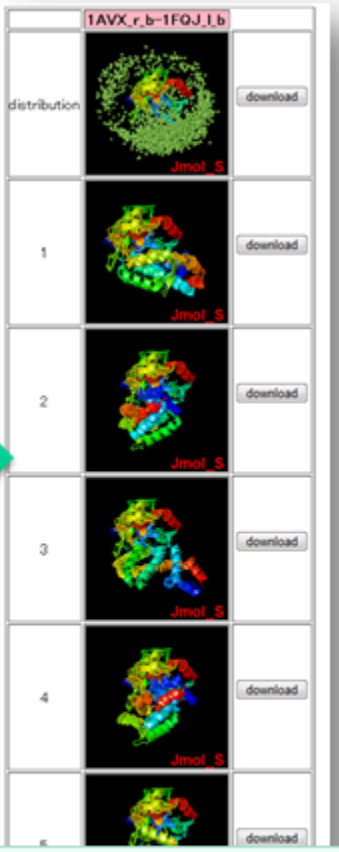
ターゲットタンパク質リスト

receptor: 1AVX_r_b

ligand	score	complex
1FQJ_l_b	8.17	1AVX_r_b-1FQJ_l_b
1AY7_l_b	8.10	1AVX_r_b-1AY7_l_b
1AVX_l_b	7.80	1AVX_r_b-1AVX_l_b
1FQ1_l_b	7.55	1AVX_r_b-1FQ1_l_b
2PCC_l_b	7.37	1AVX_r_b-2PCC_l_b
1DFJ_l_b	7.14	1AVX_r_b-1DFJ_l_b
1PPE_l_b	6.92	1AVX_r_b-1PPE_l_b
1IBR_l_b	6.91	1AVX_r_b-1IBR_l_b
1QA9_l_b	6.25	1AVX_r_b-1QA9_l_b
1D6R_l_b	6.09	1AVX_r_b-1D6R_l_b
1KXQ_l_b	5.82	1AVX_r_b-1KXQ_l_b
1HE8_l_b	5.72	1AVX_r_b-1HE8_l_b
1MAH_l_b	5.69	1AVX_r_b-1MAH_l_b
1CGI_l_b	5.66	1AVX_r_b-1CGI_l_b
1EWY_l_b	5.61	1AVX_r_b-1EWY_l_b
1E96_l_b	5.61	1AVX_r_b-1E96_l_b
1HE1_l_b	5.60	1AVX_r_b-1HE1_l_b
1WQ1_l_b	5.59	1AVX_r_b-1WQ1_l_b
1GRN_l_b	5.55	1AVX_r_b-1GRN_l_b

閾値

予測されたPPIランキング



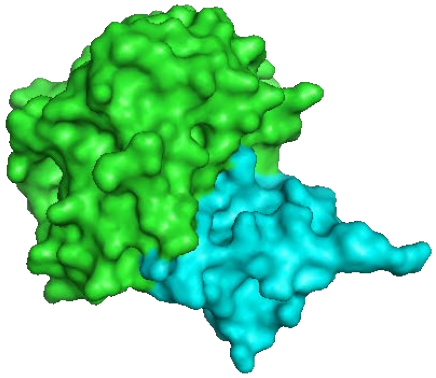
予測された複合体構造

MEGADOCK-K 講習会の概要

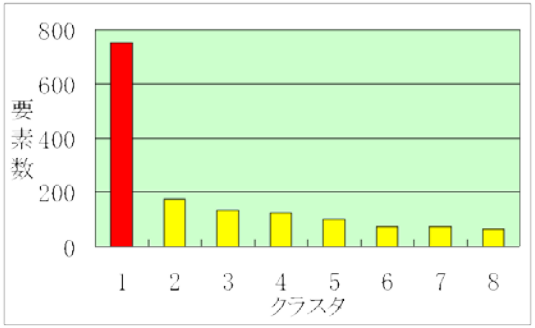
- 開発の背景
- 「京」をはじめとする高性能並列計算機上での実行を想定した並列化と性能評価
- ドッキングスコア関数の開発
- MEGADOCK-K実行の流れ
- MEGADOCK-Kを応用したタンパク質間相互作用予測研究の例

最初のアイディア：高スコア構造のクラスタリングによる相互作用対検出

ドッキング予測

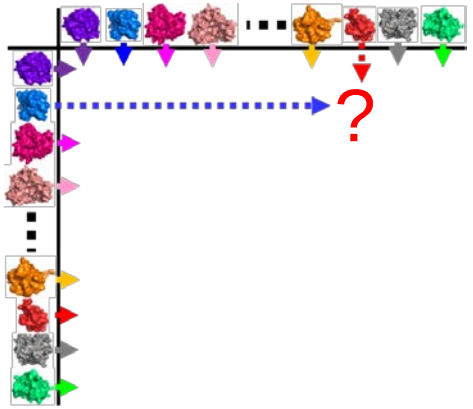


クラスタリング

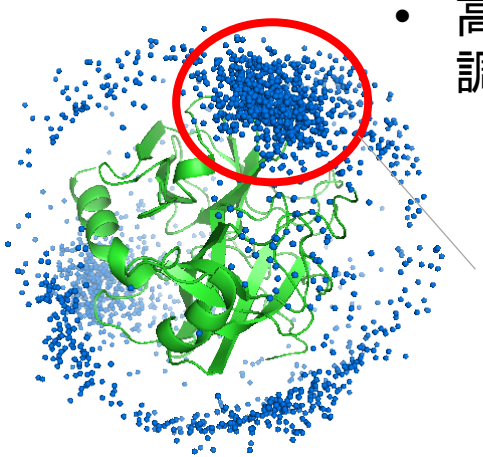


- 類似した構造をまとめ候補を絞り込む
- 高スコア構造の分布を調べる

相互作用対検出



タンパク質群から
相互作用対検出



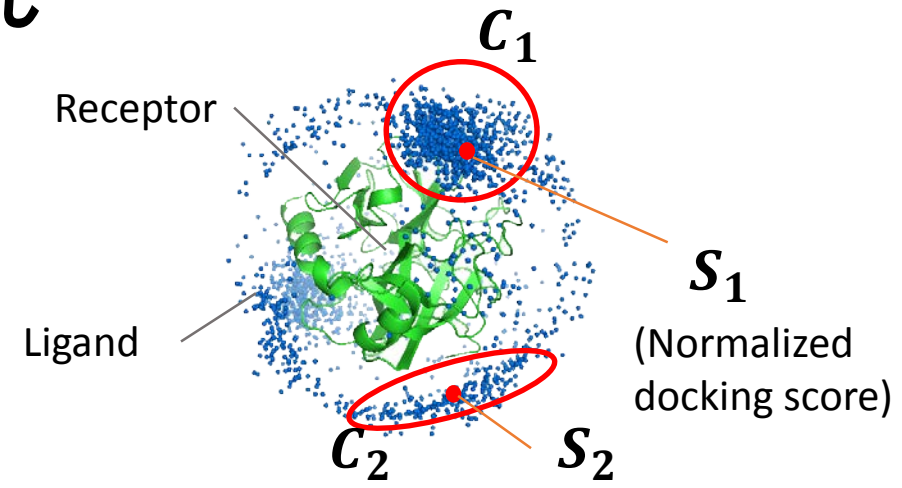
高いスコアの候補が
多く集まるタンパク質対
⇒相互作用しやすい組み合わせ？

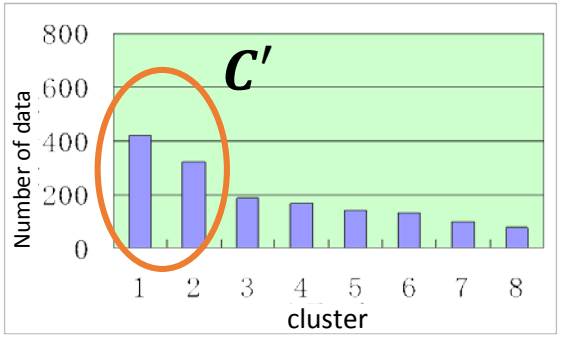
クラスタリングを用いた相互作用予測の流れ

- ドッキングスコア上位2,000モデルを得る
- 三次元座標上の距離をもとにクラスタリング
- クラスタ内のスコア最上位のモデルを代表としスコアを記録 $C_i : s_i$
- 各クラス他のメンバー数 m_i が閾値 $m*$ より多いクラスタ群 C' を選択

$$C' = \{C_i \mid m_i > m^*\}$$

- C' のクラスタのうち代表モデルのスコアを比較
  - 最上位のモデルのスコア（正規化）を親和性評価値 E とする
  - E が閾値以上のペアを相互作用ありと判定



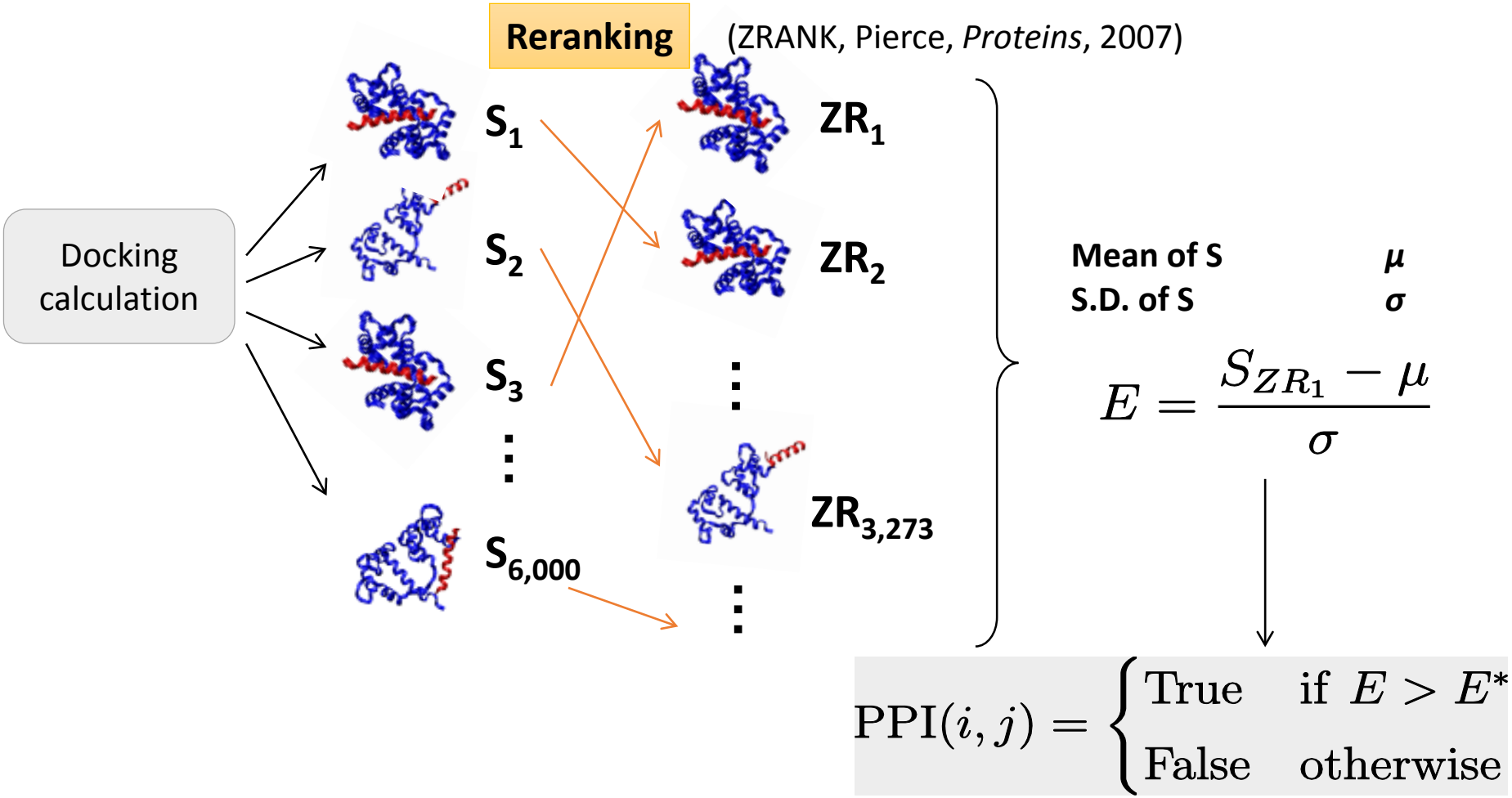


$$E = \begin{cases} max\ s_i,\ i \in C' & \text{if } C' \neq \varnothing \\ 0 & \text{otherwise} \end{cases}$$

$$interaction = \begin{cases} true & \text{if } E > E^* \\ false & \text{otherwise} \end{cases}$$

Matsuzaki , *et al.*, *J Bioinform Comput Biol*, 2009.

リランキングをもとにした相互作用予測 =>現在の MEGADOCKではこちらを利用



Ohue M, *et al.*, *Protein Pept Lett*, 2014.

タンパク質間相互作用予測実験の概要

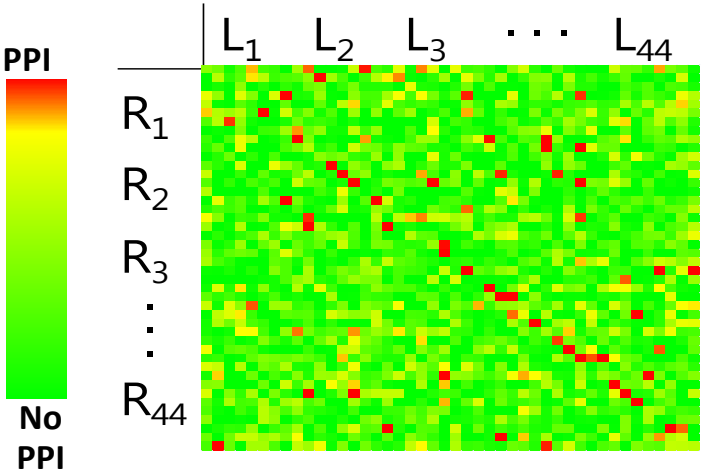
- ベンチマーク
  - 動作テスト用
  - 一般的なドッキング問題の性能評価のために構成されたデータセット
  - 原則として非冗長かつ多くのファミリーを網羅
- 実問題応用
  - 細菌走化性系
  - ヒトアポトーシス系
  - Non-small cell lung cancer パスウェイと薬剤応答タンパク質群

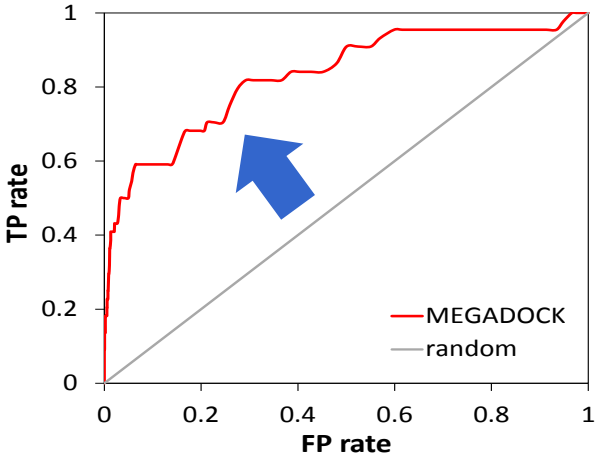
ベンチマークデータ

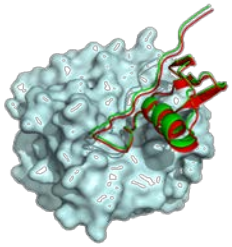
- ZLAB Benchmark 2.0 （現在は4.0）
  ドッキング予測のベンチマークとして提案されたデータセット（PDBデータ，84複合体）
- 半自動的に構築された
- 予測の難しさによる分類
  - Rigid-body 64
  - Medium 13
  - Difficult 8
- この中から44のモノマーペア複合体構造を利用
  - Rigid-body 34
  - Medium 6
  - Difficult 4

Mintseris J, Wiehe K, Pierce B, Anderson R, Chen R, Janin J, Weng Z (2005), Protein-Protein Docking Benchmark 2.0: an update, *Proteins***60**(2), 214-216.

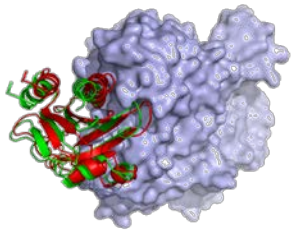
ベンチマークデータへの適用による タンパク質間相互作用予測の検証







Trypsinogen protein and trypsin inhibitor (PDB id : 1CGI)



β-actin and profilin (PDB id : 2BTF)

Predicted docking pose
Crystal structure

44x44=1936 通りのドッキングによる
相互作用ペア予測結果
(対角線上が結晶構造の複合体)

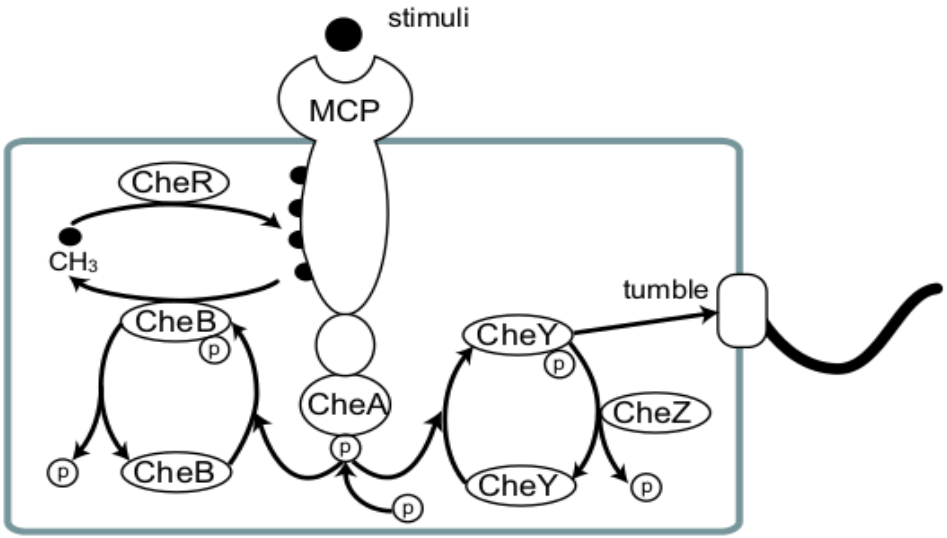
F-measure : 0.44

$$F\text{-measure} = \frac{2 \cdot \mathrm{TP}}{(\mathrm{TP} + \mathrm{FP}) + (\mathrm{TP} + \mathrm{FN})}$$

一般的なベンチマークデータのサブセット (Protein–protein docking benchmark 2.0, Mintseris *et al*, *Proteins*, 2005 より モノマーペアを選択) にMEGADOCKによるPPI予測を適用したところ、ランダムより良い予測精度を得た。

Matsuzaki Y, *et al.*, *J Bioinform Comput Biol*, 2009.
Ohue M, *et al.*, *Protein Pept Lett*, in press.

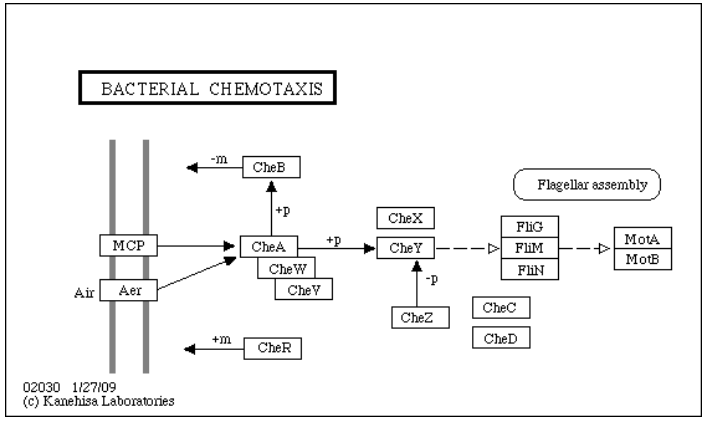
大腸菌の走化性



- 栄養状態などがよい環境に移動するためのシグナル伝達システム
  - 飢餓になると発現
- 二つのシステム的性質
  - 信号増幅
  - 信号に対する完全順応性が信号物質の量に対してロバスト
- シグナル伝達研究のモデルシステムのひとつ
  - 原核生物のシグナル伝達のほとんどを構成する　Two-component system (TCS)
  - タンパク質の量や反応機構、速度定数など実験値がそろっている
  - 経験した濃度を受容体の状態で記憶する原始的な学習機構

走化性系データセット構築

KEGG pathway



Link DB
UniProt

Structural data
（PDB）

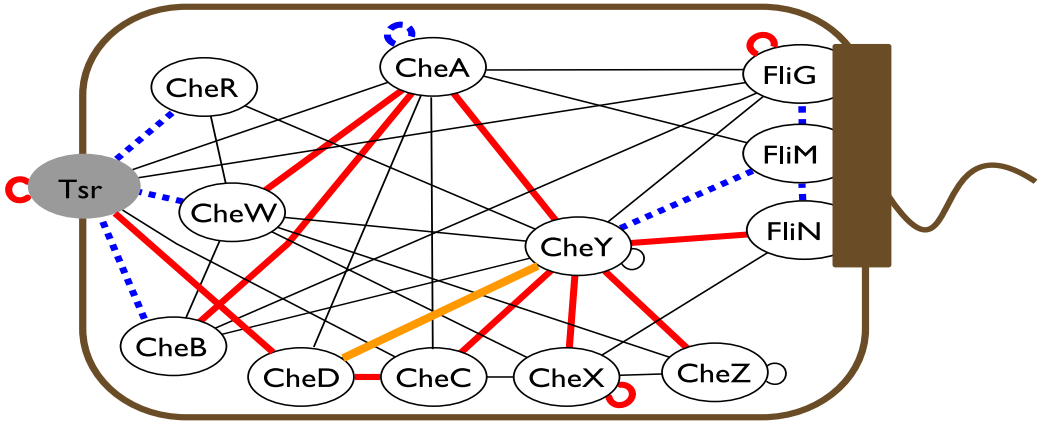
- 対象とする構造データの選択基準
  - X線結晶構造解析のデータ
  - 解像度 < 3.25 Å
  - アミノ酸残基数 > 30
  - 合成されたものは対象外
  - Mutant は除外

ベンチマークデータの
構築基準に準拠

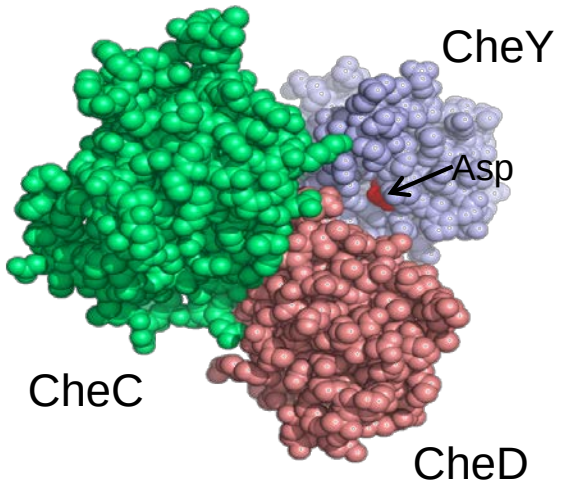
走化性データセット内容

- 13種のタンパク質について合計103のデータ
  同一タンパク質について複数の構造データが有る場合は全て利用
  - *E. coli* 44data
  - *T. maritima* 26data
  - *S. typhimurium* 33data
- データ数はCheA,CheYに偏っていた
- *T. maritima* は同一ラボによるデータが多く偏りも小さい

細菌走化性パスウェイへの応用結果



13 proteins
(101 structures)



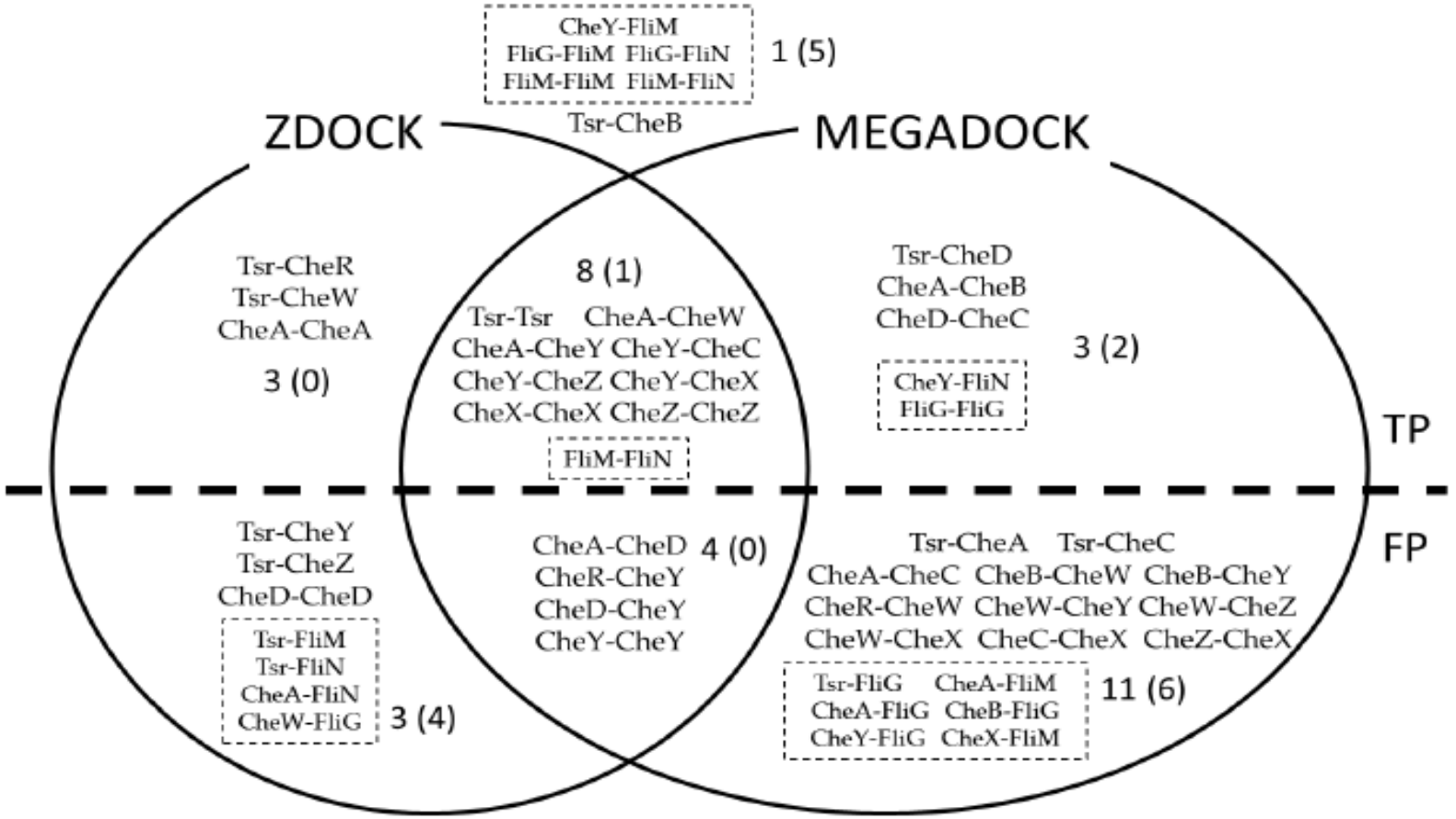
現時点では知られていないがドッキングスコアの高い相互作用(CheY-CheD) について、
その先の三量体予測を試みた。

F-measure : 0.49

小規模な生物系での検証においても、既存知識に照合して良い精度で予測を得られた。

Matsuzaki, *et al.*, *J Bioinform Comput Biol*, 2009.

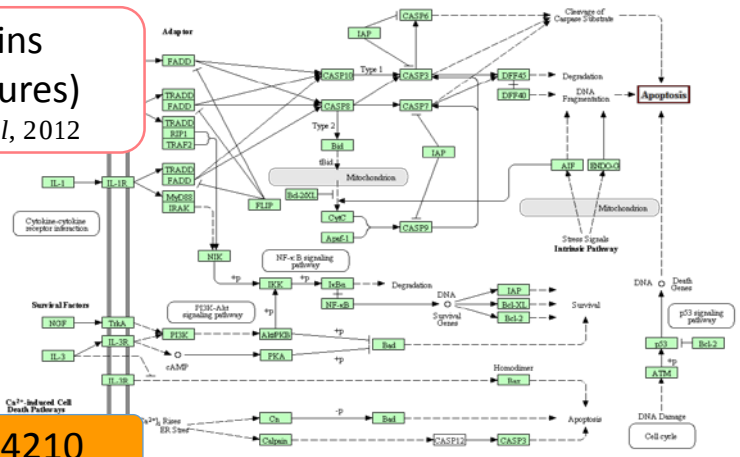
ドッキングスコア関数によって予測される相互作用が異なる



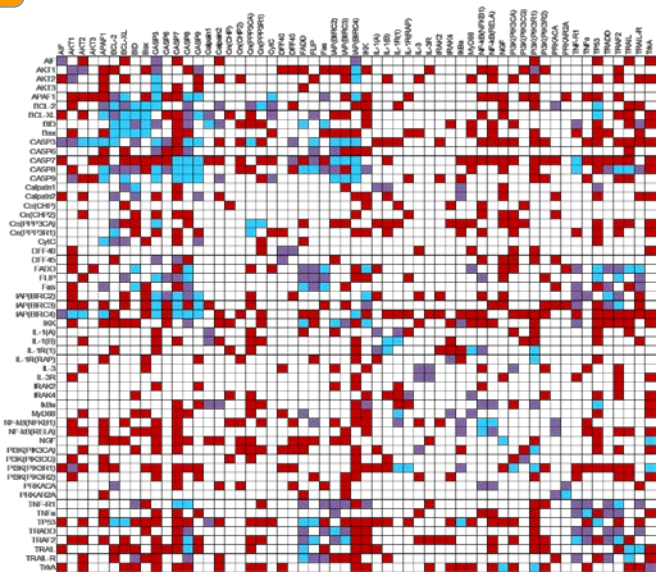
Matsuzaki, *et al.*, *Protein Pept Lett*, 2014.

ヒトアポトーシスパスウェイへの応用結果

57 proteins
(158 structures)
Ozbabacan *et al*, 2012



hsa04210



True Positive

False Positive

False Negative

F-measure : 0.28

タンパク質立体構造以外に事前知識を用いないにもかかわらず、テンプレートベースの探索 (F-measure 0.30, Ozbabacan *et al.*, *J Struct Biol*, 2012) に劣らない精度を得た。

Prediction	Interacting	No Interaction
Positive	88	364
Negative	96	1105

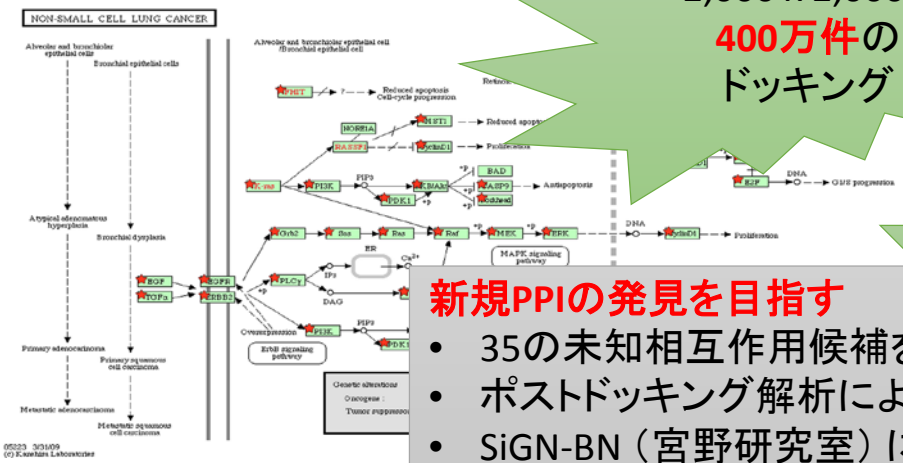
Ohue, Matsuzaki, *et al.*,
BMC Proceedings, 2013.

肺がん薬関連タンパク質間相互作用の探索

肺がんに関連の深い
EGFR シグナル伝達系の
タンパク質

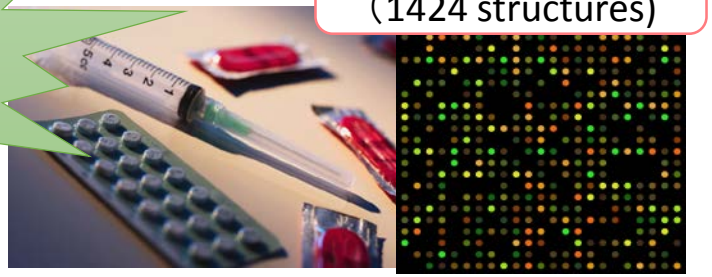
肺がん薬 **Gefitinib** に関連すると推定されたタンパク質 (東京大学医科学研究所 宮野研究室提供)

44 proteins
(497 structures)



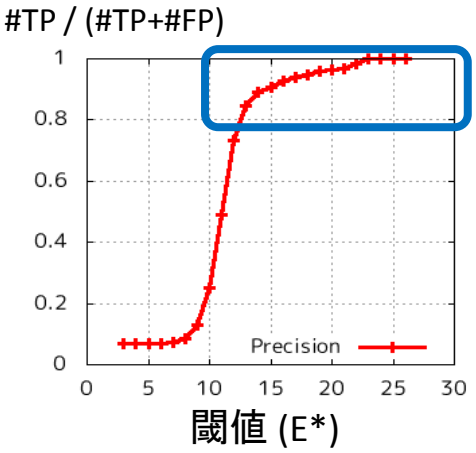
2,000 x 2,000 =
400万件の
ドッキング

294 proteins
(1424 structures)



新規PPIの発見を目指す

- 35の未知相互作用候補を予測済
- ポストドッキング解析によりさらに検討中
- SiGN-BN（宮野研究室）により「京」上で推定された、がん関連遺伝子ネットワーク256件と照合し、統計的な見地から7件の検証実験候補を選定
- 受託解析により実験



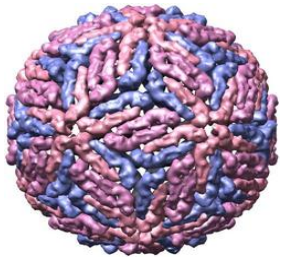
相互作用未知タンパク質ペアの親和性評価

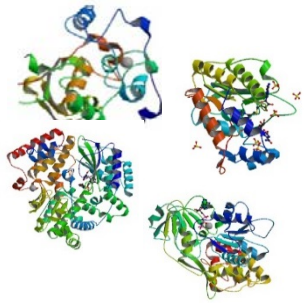
- 表面プラズモン共鳴法による親和性評価を受託解析で依頼
  - Reference Biolabs Inc., Korea
  - Device: Reichert SR7500DC sys
- 7ペアの候補を提出し6ペアについて解離定数が測定された

Reference Biolobs Inc.

[解析中] ウィルス―ヒトタンパク質間の新規相互作用探索

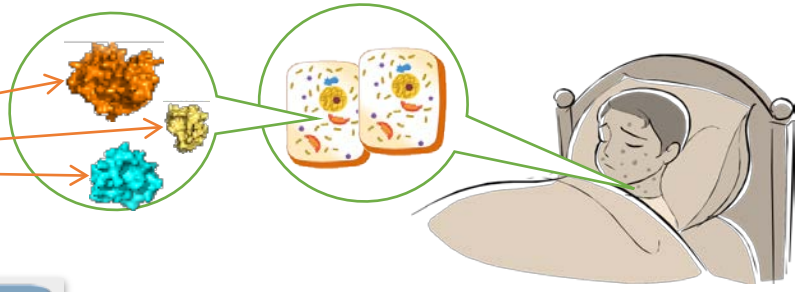
デング熱ウィルスの主要な酵素





?

ヒトのタンパク質群



ヒトのタンパク質をPDBから収集し下記の基準で選択:

- ✓ >25 residues
- ✓ X-ray resolution better than 3.25 Å
- ✓ No mutation

タンパク質名
Protease
Methyltransferase
Polymerase
Helicase

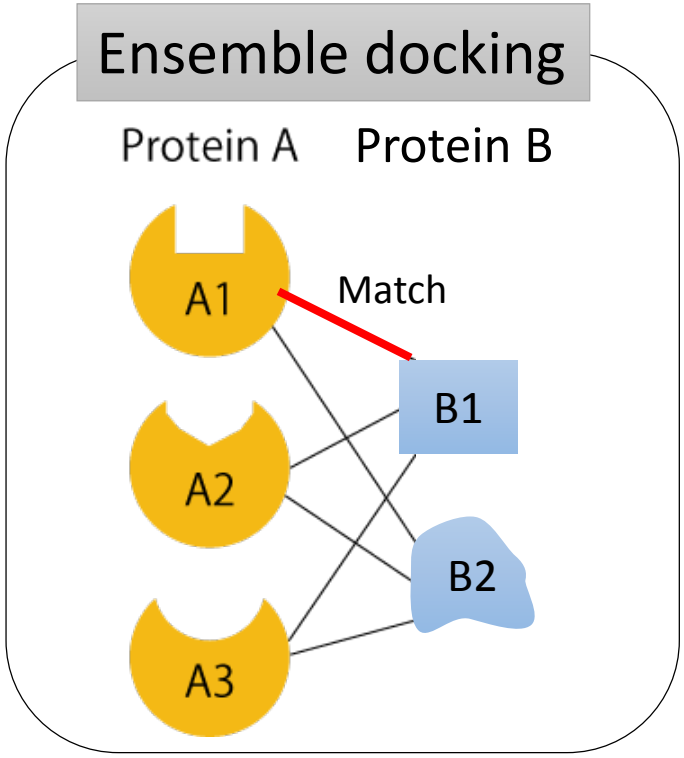
タンパク質数 (UniProt IDs)	3,353
収集した構造数(PDB-chains)	30,544

$4 \times 30{,}544 = 122{,}176$ の
ドッキングを実行し28の新規相互作用候補を得た

2013年6月15日現在

今後の展開

- タンパク質の柔軟性を考慮するためのアンサンブルドッキング
- ドッキング結果の分布とタンパク質間相互作用の種類との関連を解析
- PDB全体を対象にしたドッキングデータベースの構築



関連研究

- バイオインフォマティクス分野でも構造データの利用が試みられ始めている
  - Hue ら, 2010
    - 機械学習で構造データと既知PPIの関係を学習し、構造データからPPIネットワークを予測
- 相互作用表面のテンプレートを利用したタンパク質間相互作用予測
  - Ozbabacanら, 2012 (PRISM)
- 剛体ドッキングを大規模に応用する例も増えている
  - Mosca ら, 2009
    - 酵母既知PPIを対象としたリジッドドッキングによる網羅的複合体予測
    - 約6,000件の計算
    - 構造未知のタンパク質についてはホモロジーモデリングで構造予測してからドッキング
  - Kastritis ら, 2010
    - ドッキングスコアと親和性の関連を議論
  - Wassら, 2011
    - ネイティブな相互作用ペアと相互作用不明なペアのドッキングスコア分布は異なる

References

(Papers contributed by MEGADOCK group)

- Ohue M, Shimoda T, Suzuki S, Matsuzaki Y, Ishida T, Akiyama Y, MEGADOCK 4.0: an ultra–high-performance protein–protein docking software for heterogeneous supercomputers., *Bioinformatics,* **30**:3281-3283, 2014..
- Matsuzaki Y, Ohue M, Uchikoga N, Akiyama Y, Protein-protein interaction network prediction by using rigid-body docking tools: application to bacterial chemotaxis., *Protein and Peptide Letters, **21**:790-798, 2014*.
- Ohue M, Matsuzaki Y, Uchikoga N, Ishida T, Akiyama Y, MEGADOCK: An all-to-all protein-protein interaction prediction system using tertiary structure data., *Protein and Peptide Letters, **21**:766-778, 2014*.
- Matsuzaki Y, Uchikoga N, Ohue M, Shimoda T, Sato T, Ishida T, Akiyama Y, MEGADOCK3.0: A high-performance protein-protein interaction prediction software using hybrid parallel computing for petascale supercomputing environments., *Source Code for Biology and Medicine,* **8**:18, 2013.
- Uchikoga N, Matsuzaki Y, Ohue M, Hirokawa T, Akiyama Y, Improved post-processing of protein-protein docking data using profiles of interaction fingerprints., *PLoS ONE, **8**:e69365, 2013.*
- Ohue M, Matsuzaki Y, Shimoda T, Ishida T, Akiyama Y, Highly precise protein-protein interaction prediction based on consensus between template-based and *de novo* docking methods., *BMC Proceedings, **7***(Suppl. 7):S6, 2013.
- Ohue M, Matsuzaki Y, Ishida T, Akiyama Y, Improvement of the protein-protein docking prediction by introducing a simple hydrophobic interaction model: an application to interaction pathway analysis., *Lecture Note in Bioinformatics,* **7632**:178-187, 2012.
- Ohue M, Matsuzaki Y, Akiyama Y, Docking-calculation-based method for predicting protein-RNA interactions*., Genome Informatics*, **25**:25-39, 2011.
- Fleishman SJ, *et al*., Community-wide assessment of protein-interface modeling suggests improvements to design methodology., *Journal of Molecular Biology*, **414**:289-302, 2011.
- Uchikoga N, Hirokawa T, Analysis of protein-protein docking decoys using interaction fingerprints: application to the reconstruction of CaM-ligand complexes., *BMC Bioinformatics*, **11**:236, 2010.
- Ohue M, Matsuzaki Y, Matsuzaki Y, Sato T, Akiyama Y, MEGADOCK: an all-to-all protein-protein interaction prediction system using tertiary structure data and its application to systems biology study., *IPSJ Transactions on Mathematical Modeling and Its Applications*, **3**: 91-106, 2010.
- Matsuzaki Y, Matsuzaki Y, Sato T, Akiyama Y, *In silico* screening of protein-protein interactions with all-to-all rigid docking and clustering: an application to pathway analysis., *Journal of Bioinformatics and Computational Biology,* **7**:991-1012, 2009.