

生体分子シミュレーション研究の GPGPUによる加速

松永 康佑 (理研 AICS)
ymatsunaga@riken.jp

理研シンポジウム「理研RICC新搭載GPUと科学へのアプリケーション」
2013年6月27日 理研・鈴木梅太郎ホール

内容

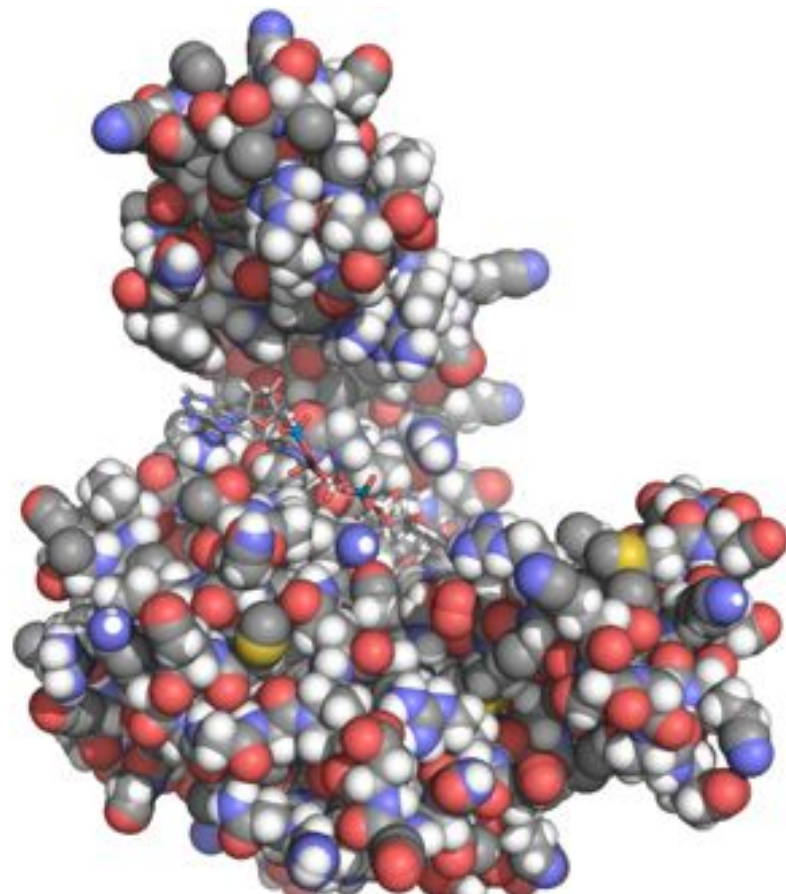
- GPUによる生体分子シミュレーション(ユーザーの視点から)
 - 生体分子シミュレーションについて、時間スケールの問題
 - 主な生体分子シミュレーションパッケージのGPUサポート状況とベンチマーク
 - AMBERを使ったマイクロ秒のシミュレーション
- GPUによるトラジェクトリ解析
 - 最近のシミュレーションの傾向(大規模・網羅的)
 - GPUを使った解析の初歩(MATLAB使用)
- まとめ

GPUによる生体分子シミュレーション ユーザーの視点から

生体分子シミュレーションとは？

- 目的
ニュートン方程式を逐一解いて、分子の挙動を原子レベルの解像度で観察する。マクロな性質を、分子のミクロな性質に基づく理論で調べて理解する
- 現在取り扱える系のサイズとターゲット
数万～数億原子 (周期境界条件)
水分子・脂質を陽に含めて、酵素、膜タンパク質、分子モーター(の一部)などを扱うにはおそろく十分
- 時間スケールの問題
数値積分の時間ステップは**フェムト(10^{-15})秒**
生体分子の特徴的な時間スケールは少なくとも**マイクロ(10^{-6})またはミリ(10^{-3})秒**

リガンド結合に伴うアデニル酸キナーゼの構造変化
(およそ0.1ミリ秒の時間スケール)

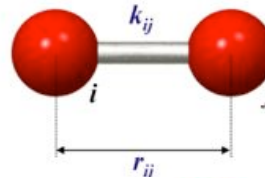


GPUを使ってマイクロ秒以上のシミュレーション
が日常的に可能になるのではと期待

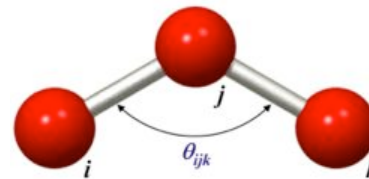
Matsunaga, Y. et al. "Minimum Free Energy Path of Ligand-Induced Transition in Adenylate Kinase."
PLoS Comput. Biol. **8**, e1002555 (2012).

生体分子シミュレーションのボトルネック

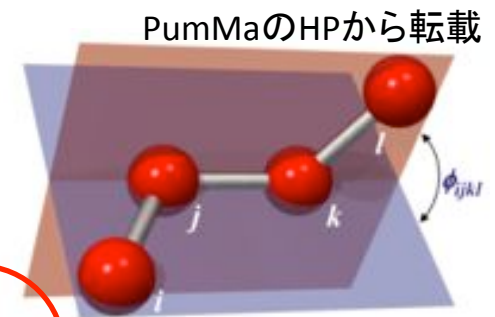
$$V(\mathbf{r}) = \sum_{bonds} k_{ij} (r_{ij} - r_0)^2$$



$$+ \sum_{angles} k_{ijk} (\theta_{ijk} - \theta_0)^2$$



$$+ \sum_{torsions} k_{ijkl} [1 + \cos(n\phi_{ijkl} - \omega)]$$



$$+ \sum_i \sum_{j>i} \epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \text{vdW term}$$

$$+ \sum_i \sum_{j>i} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \text{Electrostatic term}$$

長距離相互作用が
ボトルネック

$O(N^2)$

主な生体分子シミュレーションパッケージの GPUサポート状況 (2013年6月時点)

- CHARMM

最新版(c37b1)でCPUコードの高速化が行われるとともに(領域分割 DOMDEC)、OpenMMを使ったGPU計算をサポート

- AMBER

NVIDIA社とのプロジェクトで、pmemdモジュール(AMBERのMDエンジン)のnativeなGPUサポートが行われてきた。精度を落とす代わりに野心的な高速化が図られ、バグフィクスとともに段々と安定してきている

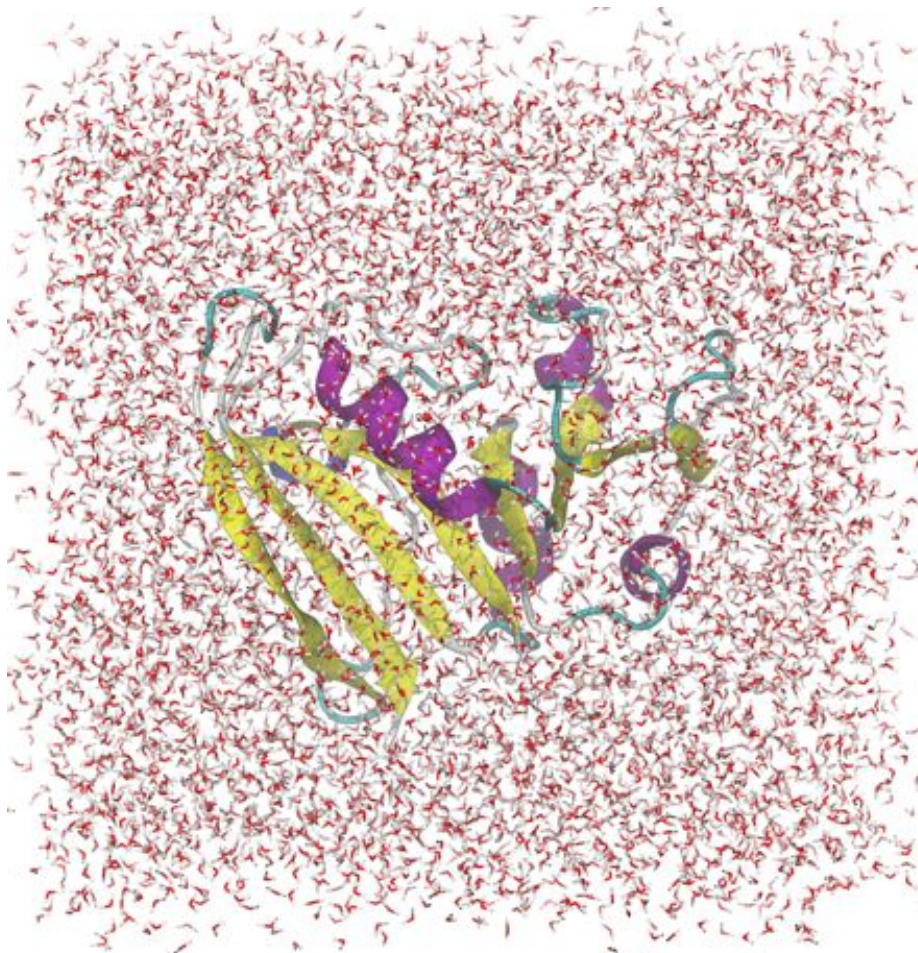
- NAMD

direct part(cutoff計算)のGPU計算をサポート。reciprocal partはCPUで計算

- GROMACS

これまでOpenMMを使ってGPGPUをサポートしていたが、最新版(ver 4.6以降)でnativeにGPGPUをサポート

Benchmark: DHFR



System: DHFR in explicit solvent

Size: **23,558 atoms** (23,569 for GROMACS)

CPU: Intel Xeon E5-2630 x 2 (12 cores)

GPU: NVIDIA GeForce GTX680

CUDA: CUDA Toolkit 4.2, driver 304.54

OS: Linux kernel 3.0.0-27 (Ubuntu 11.10)

Ensemble: NVE and NPT

time step $\Delta t = 2$ fs with SHAKE or LINCS

Electrostatic: PME, cutoff = 9Å

Programs: pmemd.cuda_SPFP of AMBER 12

(bugfix 18 applied), NAMD nightly build

(2013-06-16), GROMACS 4.6.2 single precision

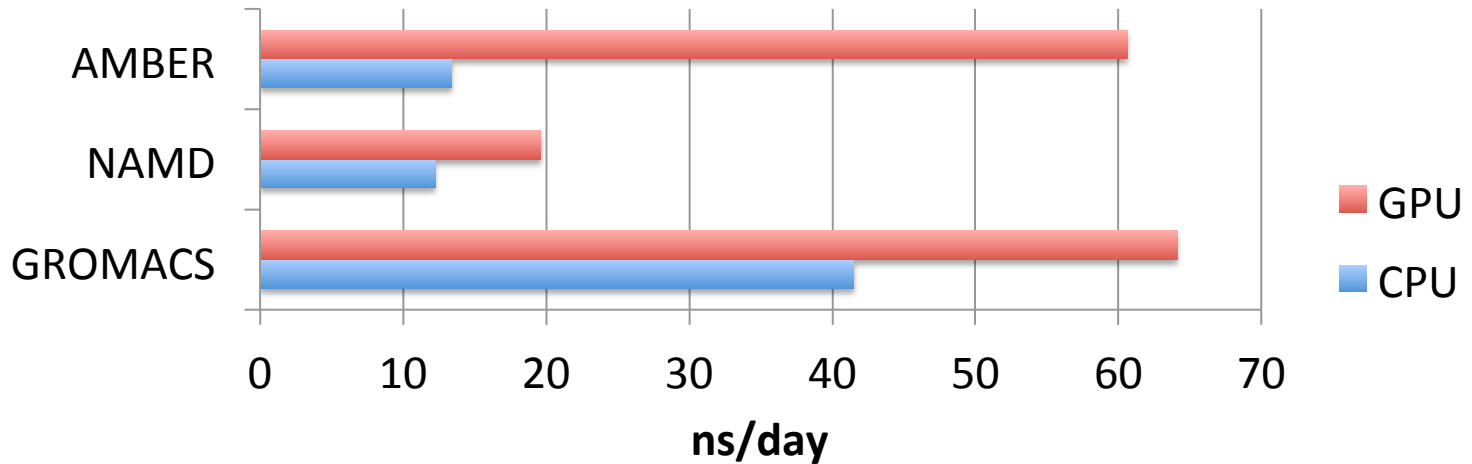
Input files: JAC benchmark files were used for

NAMD and AMBER. gromacs-gpubench-

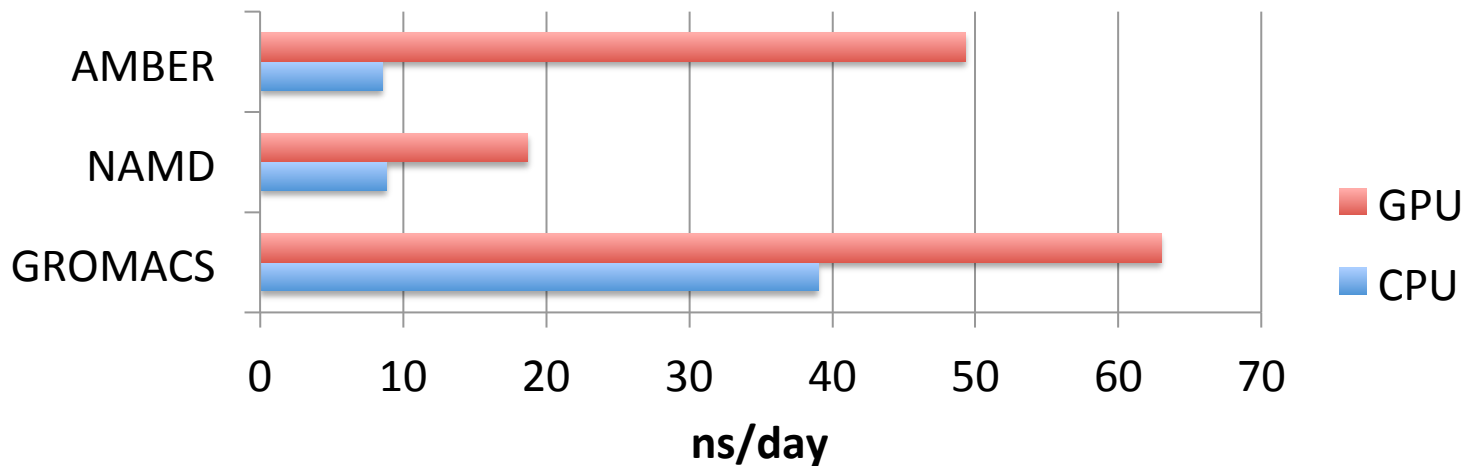
dhfr.tar.gz for GROMACS (modified for NVE and NPT).

Benchmark: DHFR

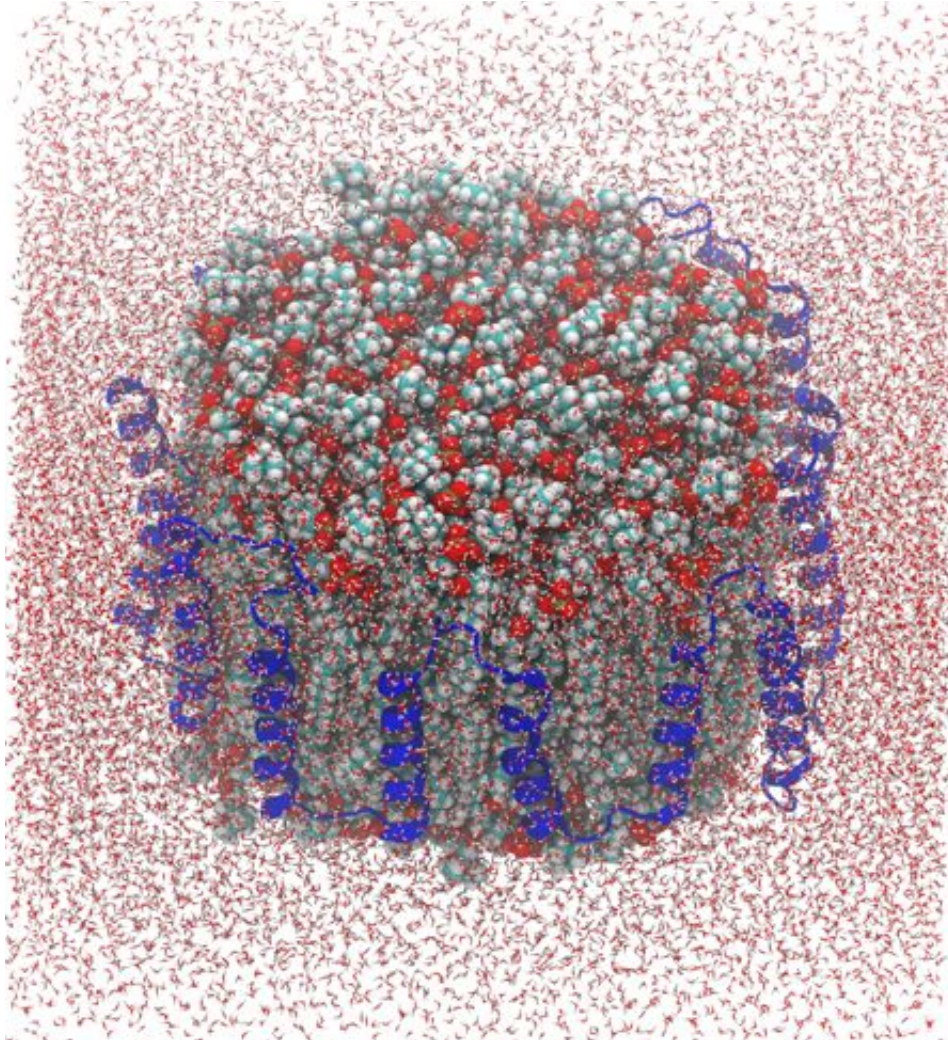
NVE ensemble (定積・定エネルギー条件)



NPT ensemble (定圧・定温条件)



Benchmark: ApoA1



System: ApoA1 in explicit solvent

Size: 92,224 atoms

CPU: Intel Xeon E5-2630 x 2 (12 cores)

GPU: NVIDIA GeForce GTX680

CUDA: CUDA Toolkit 4.2, driver 304.54

OS: Linux kernel 3.0.0-27 (Ubuntu 11.10)

Ensemble: NVE and NPT

time step $\Delta t = 2$ fs with SHAKE

Electrostatic: PME, cutoff = 9Å

Programs: pmemd.cuda_SPFP module

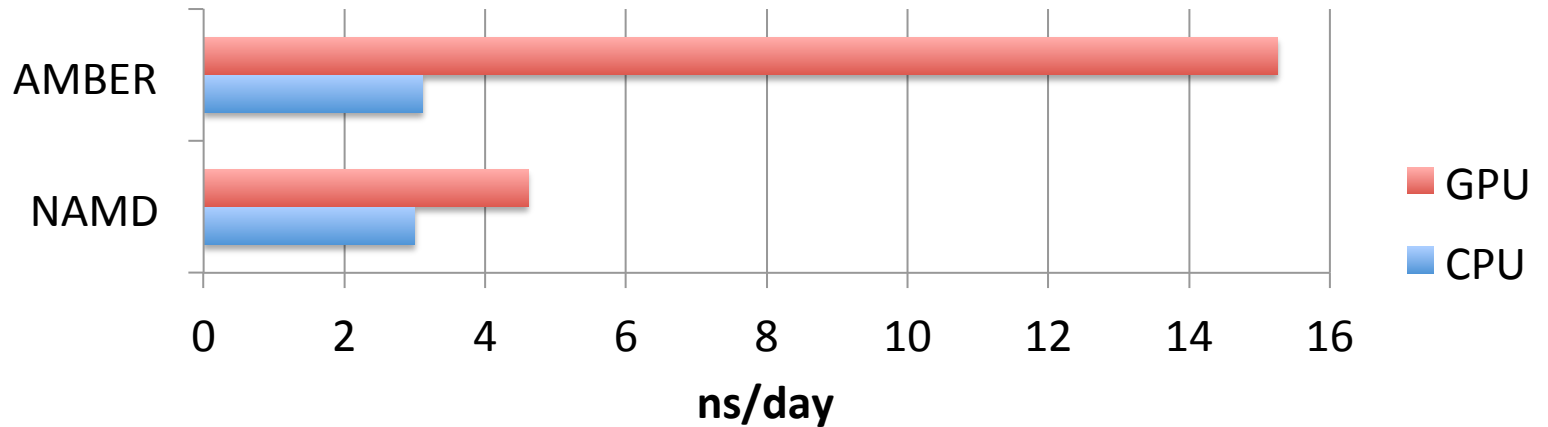
AMBER 12 (bugfix 18 applied), NAMD

nightly build (2013-06-16)

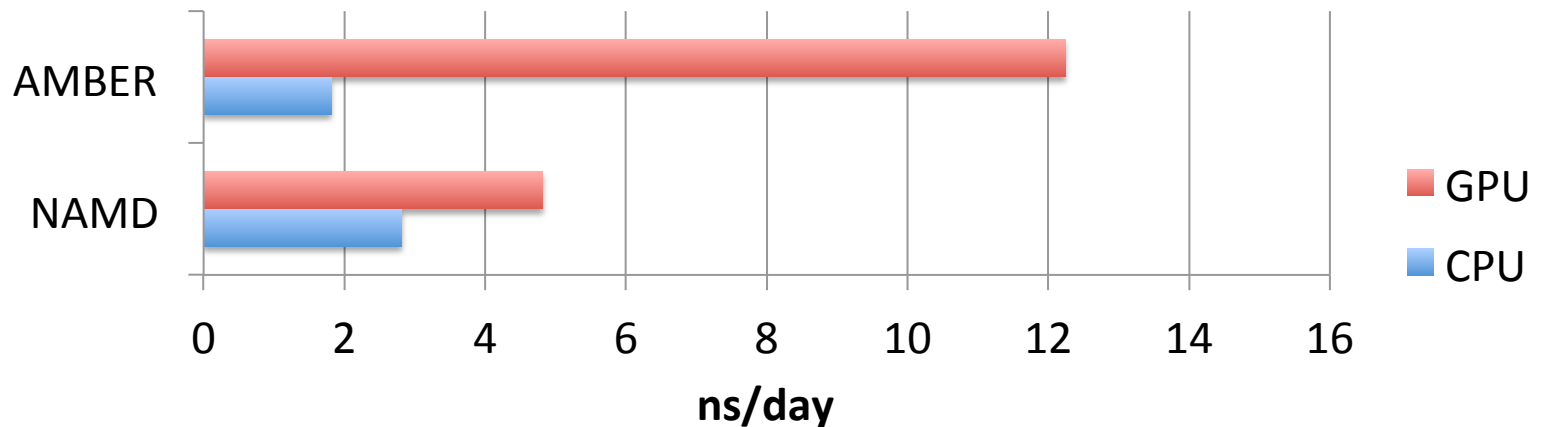
Input files: benchmark file provided by NAMD group was converted to AMBER format by using chamber module of AmberTools.

Benchmark: ApoA1

NVE ensemble (定積・定エネルギー条件)



NPT ensemble (定圧・定温条件)



各パッケージの個人的な雑感

- CHARMM
 - まだ最新版が手に入っておらず、詳しい情報がない
- AMBER
 - 現時点で最もGPUの恩恵を受けている。Implicit溶媒と精度(SPFPモデル)の実装に関する論文は出たが、Explicit溶媒(PME)の実装論文がまだ出ていない。精度を落としたことによる影響が今後更に検証されていくと思われる
- NAMD
 - もともと多くのノードを使ってスケーリングするように設計されている(領域分割など)ので、GPUを使うよりはたくさんノードを使った計算でパフォーマンスが出る
- GROMACS
 - もともとのCPU向けのチューニング(SIMD最適化)が凄まじく、CPU計算とGPU計算ではまだそれほど差がない

AMBERを使ったGPU計算の準備

自前のGPUマシンで計算する場合

- 最初はGPUのメモリチェック
 - memtestG80 や cuda_memtest を使う
memtestG80 <https://github.com/ihaque/memtestG80>
- AMBERのコンパイル
 - AMBERのGPUサポートページを参考に <http://ambermd.org/gpus/>
 - 必ず最新のbugfixをあてる(インストール時に自動的にパッチされる)
バグフィクス情報 <http://ambermd.org/bugfixes12.html>
5月末に bugfix 18 が公開された。NPTのリスタート問題など重要な修正が施されている
 - テスト
 - 付属のスクリプトを用いる(make test.cuda と ./test_amber_cuda_parallel.sh)
 - 自前のテスト。開発者の一人曰く cellulose NVE を10万ステップ、同じ乱数シードで2回やって、最終結果(エネルギー)が同じかチェックすべきだ、とのこと (SPFPの場合)
 - 参考 Amber ML <http://archive.ambermd.org/201305/0409.html>

RICCを使う場合

インストール済みのバイナリを使用可能

AMBERの実装: SPFP precision model

Single Precision Fixed Precision (SPFP)は、AMBERのGPU計算でデフォルトで採用されている精度モデル

1. 長距離相互作用の各項を単精度(SP)で計算
 2. 各項を足し合わせる際には、固定小数点(FP)で計算(実体は64-bit integerの計算)
- メリット
 - 倍精度で足すよりも計算が速い
 - デメリット
 - 想定した桁数より大きい値は取り扱えない ← 大きい系など
 - 桁数が大きく変わる(dynamic rangeが広い)と不安定 ← minimizationなど

Numerical Stability

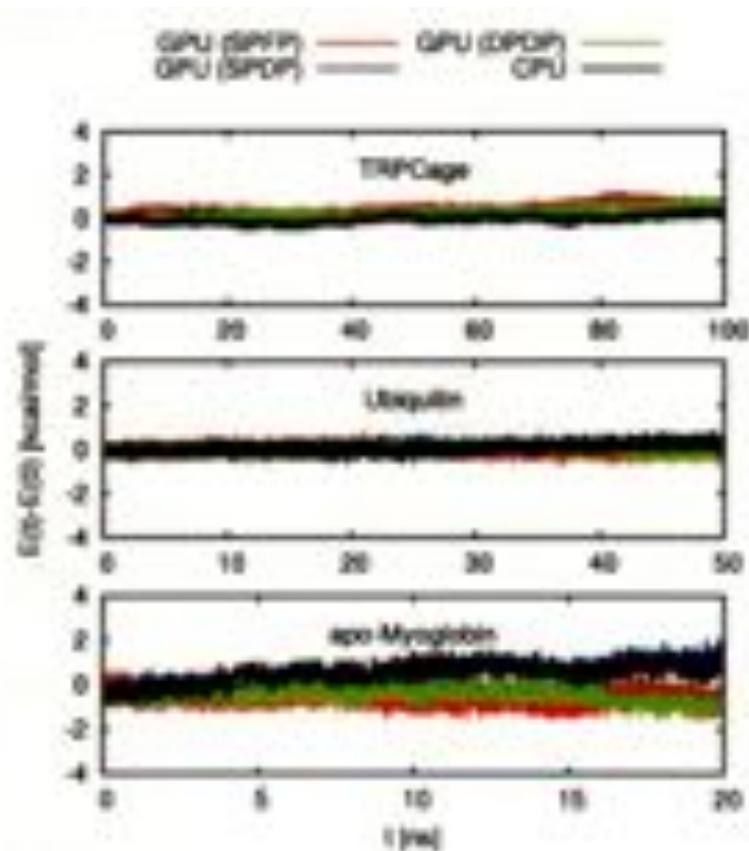
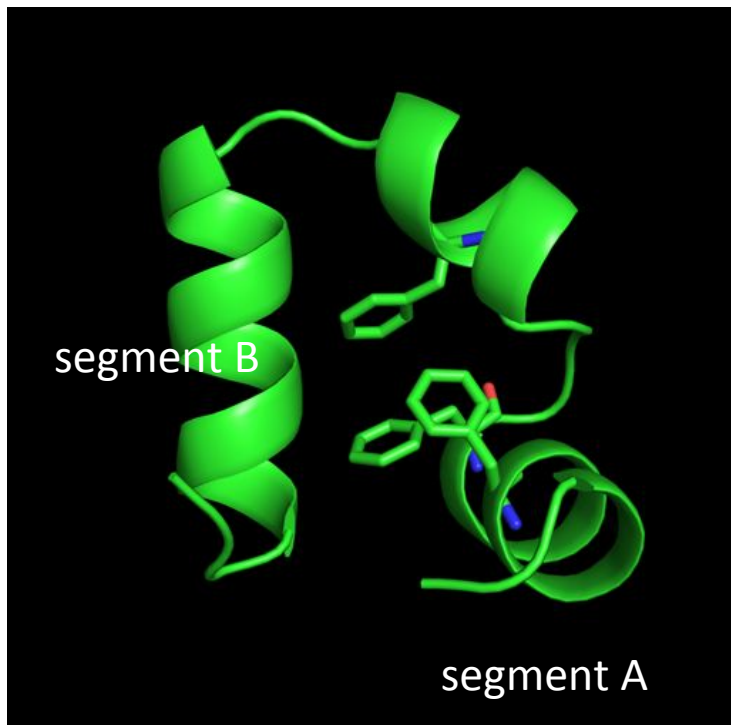


Fig. 3. Total energy (kcal/mol) along constant energy trajectories using a time step of 0.3 fs without constraints. Shown are results for TRPCage (top), ubiquitin (center) and apo-myoglobin (bottom) for different precision models.

Le Grand, S., Götz, A. W. and Walker, R. C. "SPFP: Speed without compromise — A mixed precision model for GPU accelerated molecular dynamics simulations." *Computer Physics Communications* **184**, 374–380 (2013).

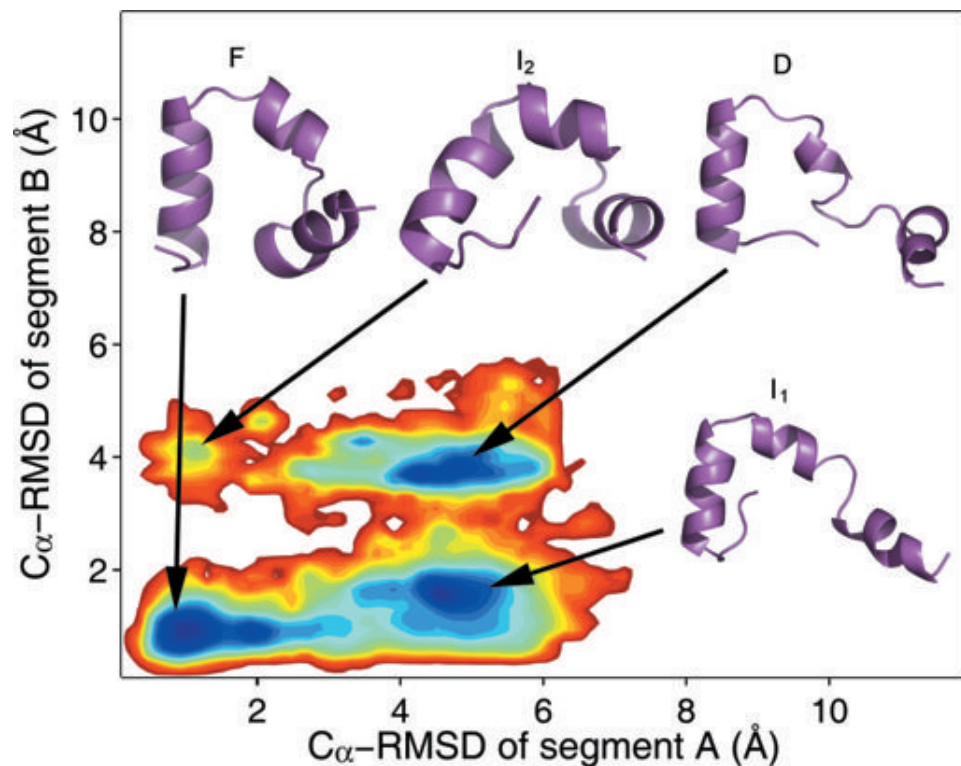
Folding of villin headpiece (implicit solvent)



NMR structure (PDB ID: 1YRF)

System: Villin headpiece (35 residues)
Size: 582 atoms (implicit solvent)

Previous work by Duan et al.

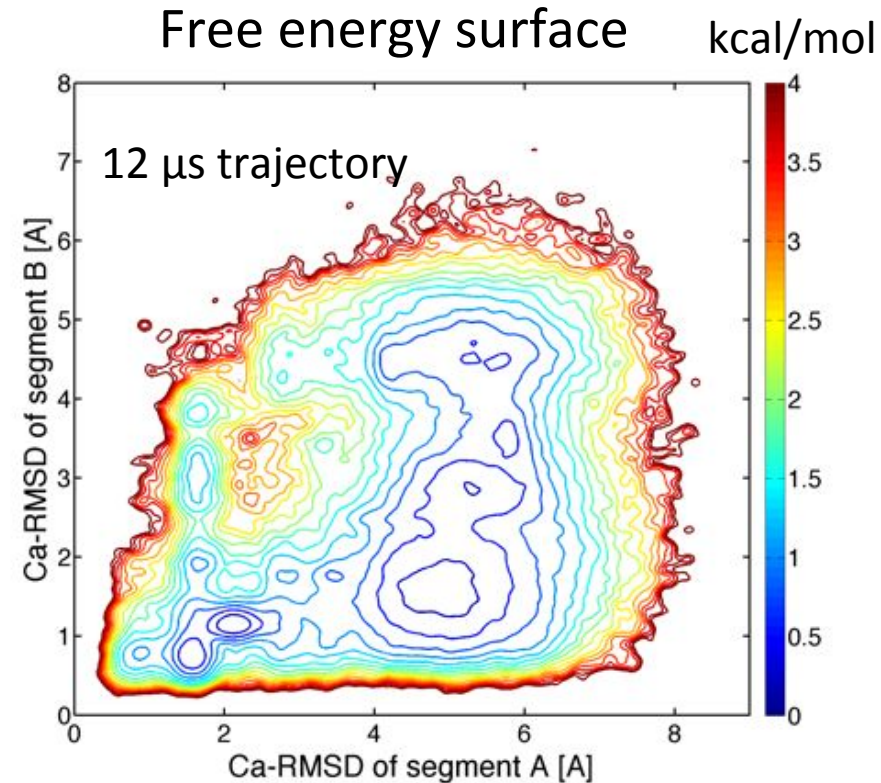
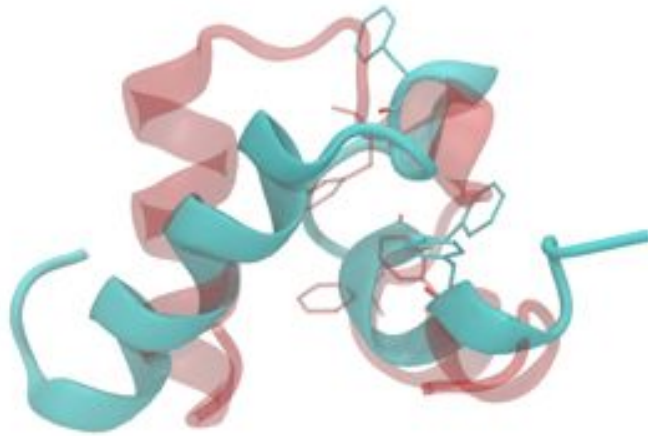


20 replicas were used for Replica Exchange simulation.
200 ns for each REMD replica. Sander module of
AMBER 8 was used

Lei, H., Wu, C., Liu, H. and Duan, Y. Folding free-energy
landscape of villin headpiece subdomain from molecular
dynamics simulations. *PNAS* **104**, 4925–4930 (2007).

Folding of villin headpiece (implicit solvent)

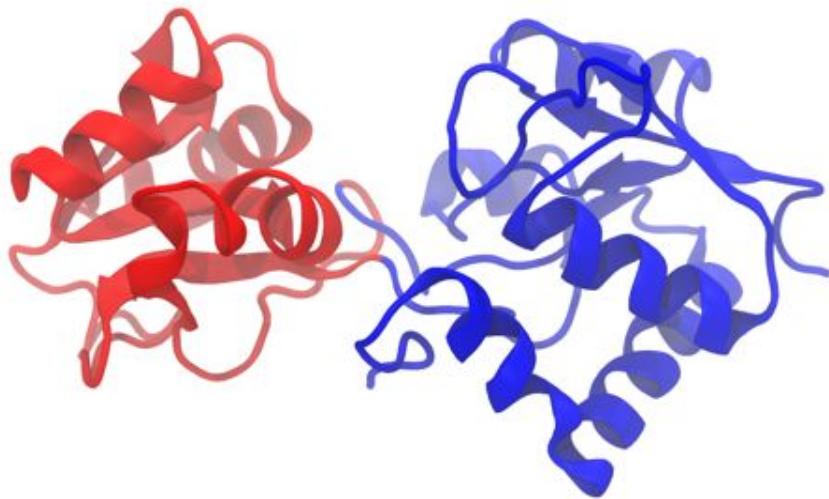
Simulation with GPU (GTX680): Throughput 509ns/day
(CPU: Throughput = 77 ns/day)



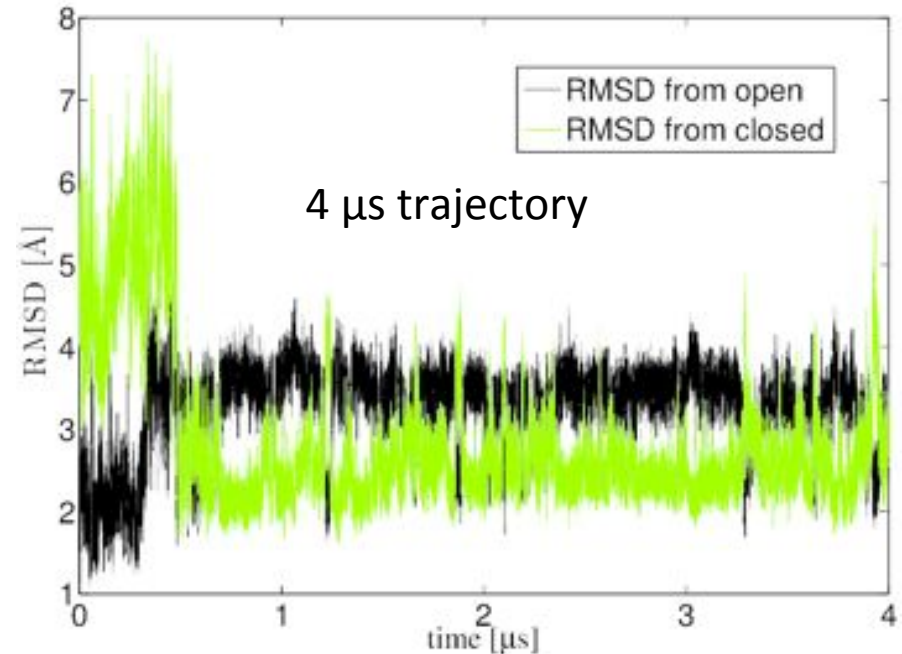
GPU: 24 days for 12 μ s long
(CPU: 156 days)

Functionally relevant motion of LAO-binding protein (explicit solvent)

Simulation with GPU (GTX680): Throughput = 25 ns/day
(CPU: Throughput = 4 ns/day)



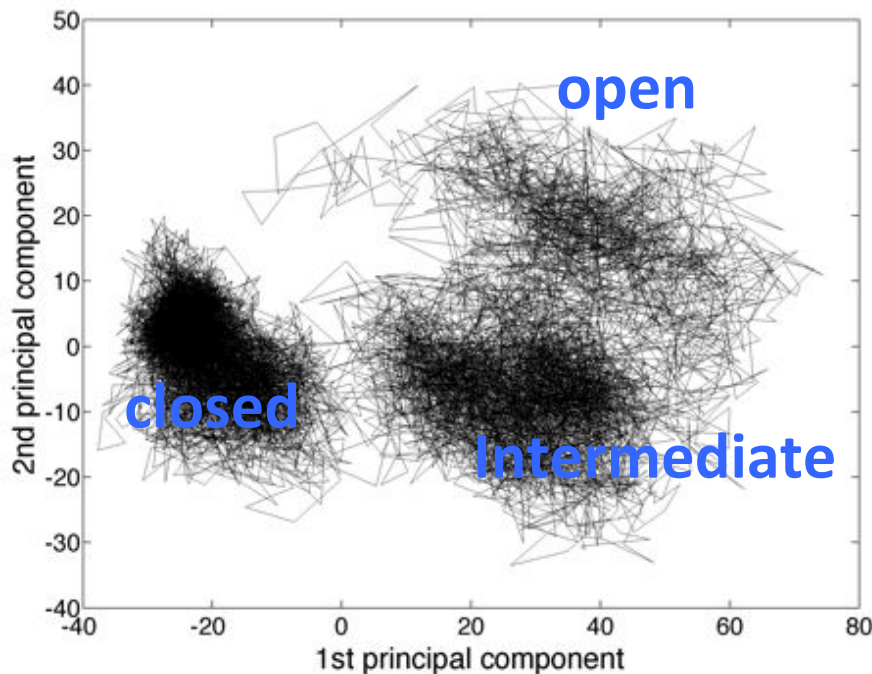
System: LAO-binding protein (238 residues)
Size: 40,978 atoms including TIP3P waters and ions



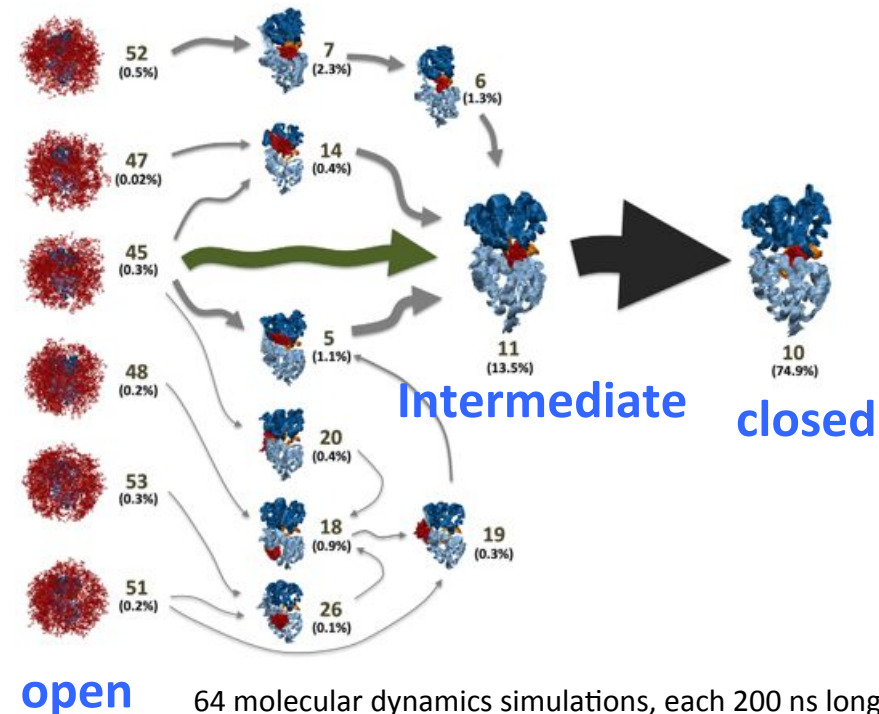
GPU: 156 days for 4 μs long
(CPU: 957 days)

Functionally relevant motion of LAO-binding protein (explicit solvent)

我々の結果 Simulation with GPU



Previous work by Silva et al.



Silva, D.-A., Bowman, G. R., Sosa-Peinado, A. and Huang, X. "A Role for Both Conformational Selection and Induced Fit in Ligand Binding by the LAO Protein." *PLoS Comput. Biol.* **7**, e1002054 (2011).

GPUによるトラジェクトリ解析

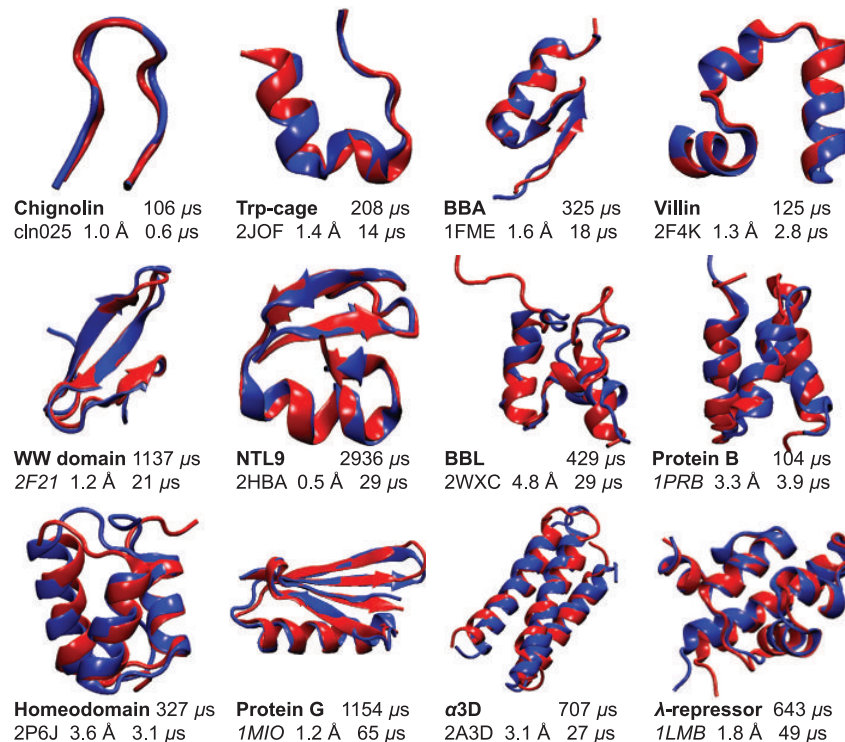
GPU・専用計算機・巨大スパコンの登場 による最近のシミュレーションの傾向

MD専用計算機 Anton@D. E. Shaw Research

Folding simulations by Anton



京@RIKEN AICS 64万コア (次ページで説明)

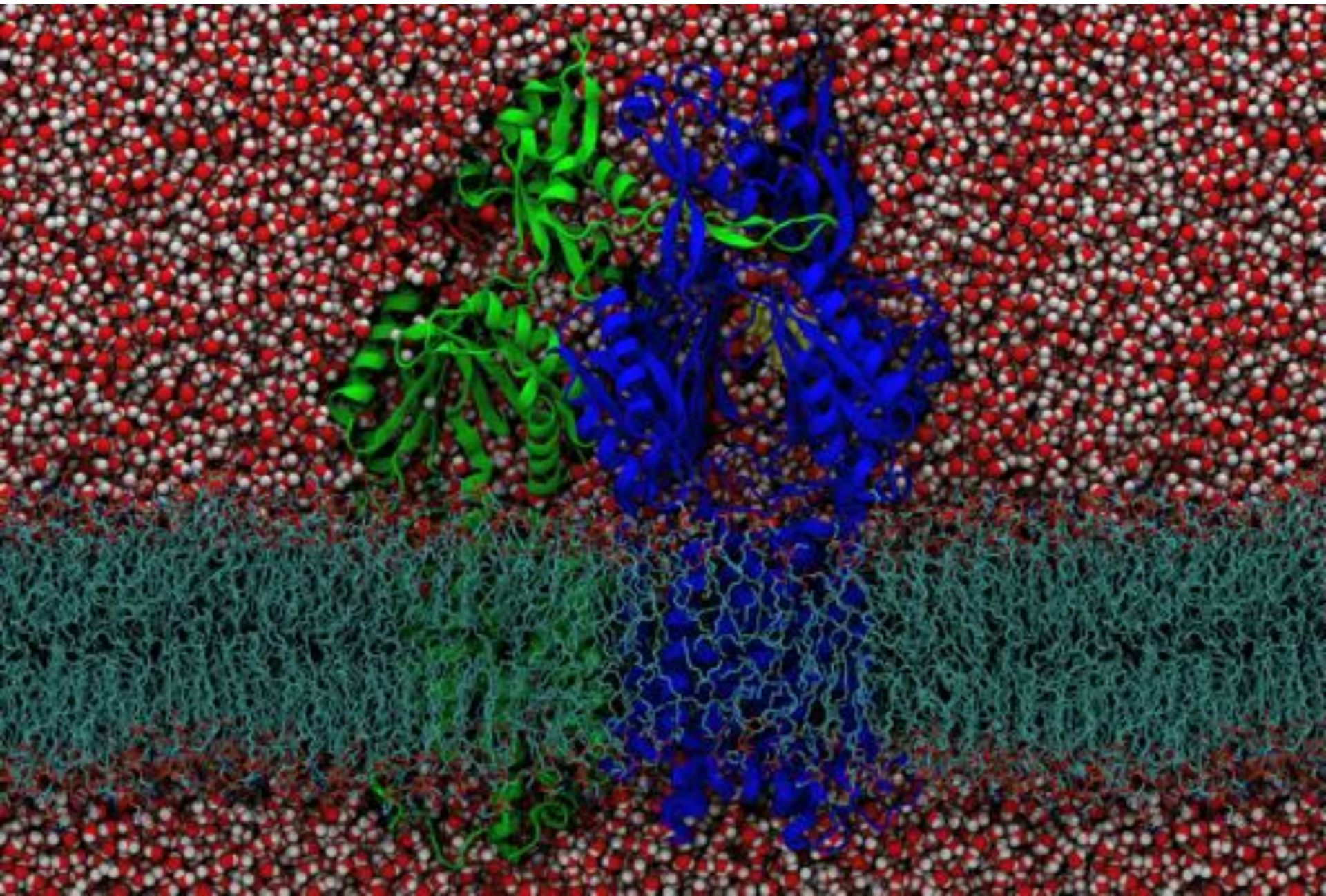


Lindorff-Larsen, K., Piana, S., Dror, R. O. Shaw, D. E.
“How Fast-Folding Proteins Fold.” *Science* **334**, 517 (2011).

長時間・網羅的なシミュレーションが可能になってきている。
それとともにシミュレーション後のデータ解析・データマイニングの比重が大きくなりはじめている

仮説駆動による研究 → データ駆動による研究

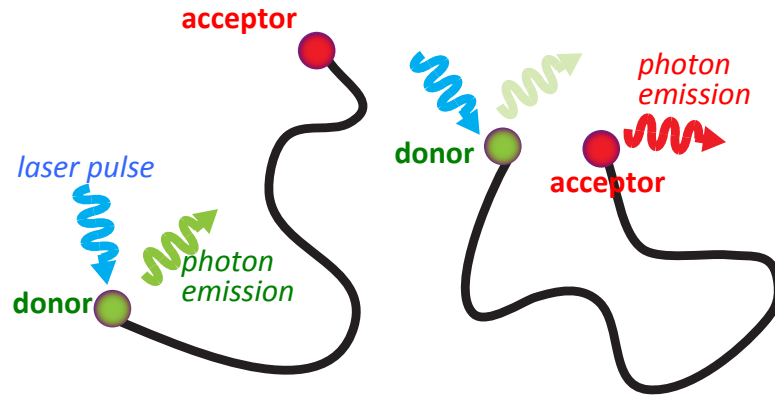
大規模データの例1: 多剤排出トランスポーターAcrBの薬剤排出過程



大規模データの例2: 粒子フィルタによる1分子FRETデータ同化

FRET計測: 分子に2個の色素分子をつけて、蛍光共鳴エネルギー移動効率(FRET)を計測することで、**色素分子間の距離**を計測する方法

粒子フィルタのテスト計算の例

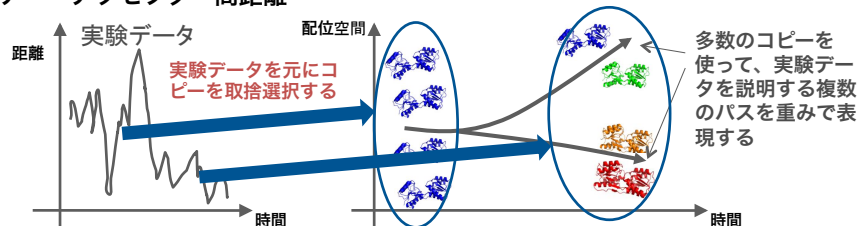


Sisamakakis, E., Valeri, A., Kalinin, S., Rothwell, P. J. & Seidel, C. A. M. *Meth. Enzymol.* **475**, 455–514 (2010).

粒子フィルタ: 逐次的にベイズモデル(例えば1分子FRETデータのような低次元情報を条件とする隠れた構造変数の事後分布)を推定する逐次モンテカルロ法のアルゴリズムのひとつ。



1分子FRET実験で観測される
ドナー・アクセプター間距離



A. Doucet, N. De Freitas, N. Gordon, "Sequential Monte Carlo methods in practice", Springer Verlag (2001).

1万～10万個の粗視化タンパク質モデルを使って粒子フィルタの計算中

MATLABによるトラジェクトリ解析

MATLABによるシミュレーションのトラジェクトリ解析

- 高レベル言語なので様々なアルゴリズムに気軽にチャレンジできる
- Muti-thread化された関数が多くある
- 統計関連の関数が豊富
- I/Oや構造解析ツール(フィッティングなど)からなる基本フレームワークを整備済み

MDToolbox: <https://github.com/ymatsunaga/mdtoolbox>

MATLABにCUDAプログラミング

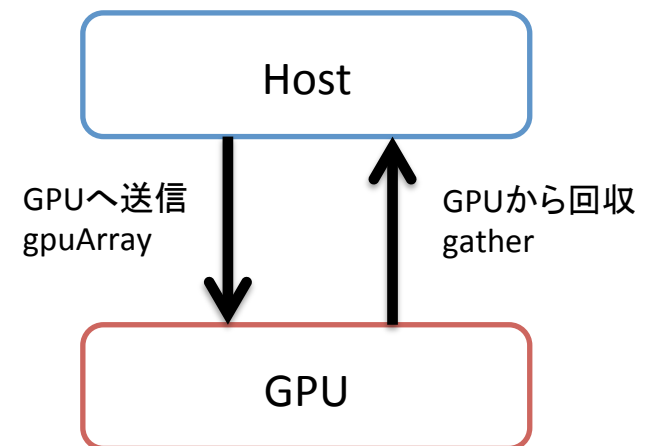
オプションで提供される Parallel Computing Toolbox を使用

```
% 行列を作成
>> A1 = rand(3000, 3000);

% GPUへ送信
>> A2 = gpuArray(A1);

% fftをGPU上で実行 (自動的にCUDAのfftが呼ばれる)
>> B2 = fft(A2);

% GPUからメインメモリへ回収
>> B2 = gather(B2);
```



<http://www.mathworks.co.jp/discovery/matlab-gpu.html>

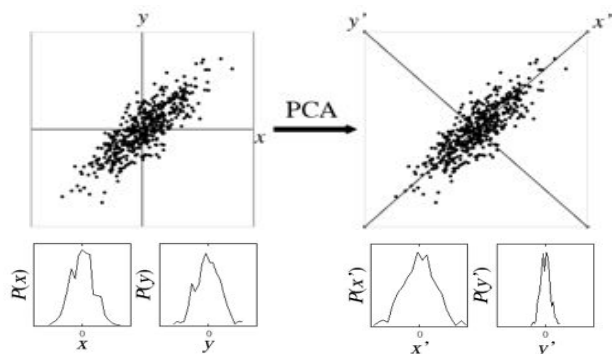
<http://tsuame.gsic.titech.ac.jp/docs/guides/isv-apps/MATLAB/html/matlab4.html>

Parallel Computing Toolboxによる解析の高速化

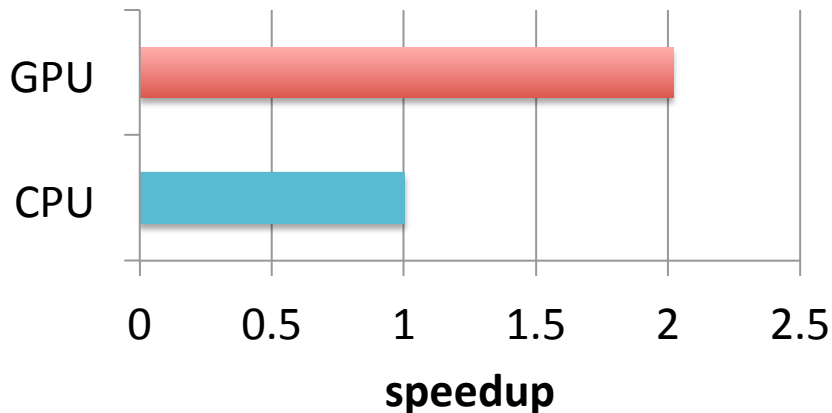
主成分解析(PCA)

高次元データを低次元に縮約する手法の一つ

1. トラジェクトリから共分散行列の計算
2. 共分散行列の対角化



10,000 snapshots, 300 atoms のトラジェクトリを使ったベンチマーク



Weighted Histogram Analysis Method (WHAM)

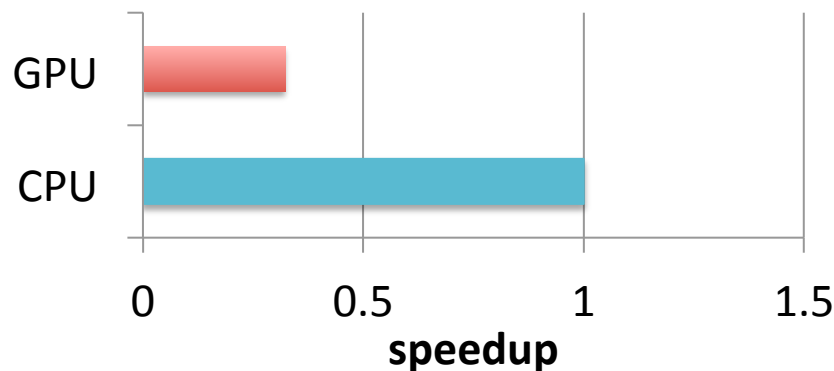
条件の異なる独立したシミュレーションのトラジェクトリをまとめて使って統計量を計算する際の重みを評価する方法

1. エネルギー空間などでヒストグラムを作成
2. ヒストグラムからWHAM方程式を自己無撞着に解く

wham方程式

$$P(\xi) = \frac{\sum_{i=1}^{N_w} g_i^{-1} h_i(\xi)}{\sum_{j=1}^{N_w} n_j g_j^{-1} \exp[-\beta(w_j(\xi) - f_j)]} \quad \exp(-\beta f_j) = \int d\xi \exp[-\beta w_j(\xi)] P(\xi)$$

tri-alanineのumbrella samplingを使ったベンチマーク



今後、低レベル言語を使ったチューニングが必要

まとめ

- GPUによりマイクロ秒以上の生体分子シミュレーションが日常的に可能になってきており、AMBERを使った例を紹介した
- AMBERに関しては、今後応用研究を通して精度落としたことによる影響が更に検証されると思われる
- シミュレーションの速度向上や大規模化により、将来的にデータマイニングの比重が大きくなると予想されるが、解析もGPUで加速できれば理想的

- 共同研究者

杉田有治チームリーダー (理研 AICS, ASI, QBiC)

- 謝辞

理研 AICS 粒子系生物物理研究チームメンバー

理研 QBiC 分子機能シミュレーション研究チームメンバー

木寺詔紀氏 (横浜市大) 寺田透氏 (東大) 山根努氏 (横浜市大) 池口満典氏 (横浜市大)

原田隆平氏 (理研 AICS)

理研 情報基盤センター 中田さん 検崎さん

多剤排出トランポーターAcrB: HPCI「京」一般利用 hp120027

粒子フィルタによるFRETデータ同化: AICS高度化調整枠, HPCI「京」若手人材育成利用 hp120060

トラジェクトリ解析ツール: 科学研究費補助金(若手研究(B)) 24770159

理研 基礎科学特別研究員制度

