

## GaREUS 法を用いた自由エネルギー地形の計算

理化学研究所・生命機能科学研究センター

尾嶋 拓

Calculation of Free-Energy Landscape Using the GaREUS Method  
RIKEN Center for Biosystems Dynamics Research  
Hiraku Oshima

(投稿日 2021/03/04、再投稿日 2021/06/16、受理日 2021/08/23)

キーワード：分子動力学法 (Molecular Dynamics)、自由エネルギー計算法、拡張アンサンブル法、GaREUS 法、GENESIS

### イントロダクション

分子動力学シミュレーション (Molecular Dynamics: MD) は、粒子間に働く相互作用および力を計算し、運動方程式を解くことで、粒子の運動を求める方法である。溶媒を含む全ての相互作用を考慮し、古典力学の範囲内で生体分子の動的構造を計算し、その計算結果を解析することで様々な物理量を求めることができる。タンパク質や核酸などの生体分子の動的構造や相互作用を原子解像度で調べる手法として生命科学において広く用いられている。特に、MD シミュレーションでは各状態での安定性または存在確率を求めることができ、これらは自由エネルギー地形と呼ばれている。自由エネルギー地形を詳細に分析することで、生体分子の構造変化や分子機構を理解することができる。

しかし、生体分子の自由エネルギー地形は多数の局所安定状態を持った複雑な形状をしており、限られた計算時間ではシミュレーションが局所安定状態に捕らわれてしまう。この問題を避けるために、拡張アンサンブル法などの効率的なサンプリング法が用いられている。拡張アンサンブル法では、エネルギーあるいは事前に定義された反応座標における次元ランダムウォークを実現することで、状態空間を広く探索する。

様々な拡張アンサンブル法が提案されているが、計算コストを抑えつつ、タンパク質構造を効率的に探索する計算手法が望ましい。そのような計算手法として、我々は Gaussian accelerated MD (GaMD) 法 (文献 1,2) と Replica-Exchange Umbrella Sampling (REUS) 法 (文献 3,4) の 2 つの拡張アンサンブル法を組み合わせた Gaussian accelerated Replica-Exchange Umbrella Sampling (GaREUS) 法を提案した (文献 5)。GaREUS 法では、GaMD 法によってタンパク質の構造柔軟性を向上させ、同時に REUS 法によって反応座標空間を広く探索する。GaREUS 法で得られたトラジェクトリーから二段階の Reweighting を行うことで、正確な自由エネルギー地形を求めることが出来る。GaREUS 法は、理化学研究所で開発している MD シミュレーションソフトウェア「GENESIS」 (文献 6,7) に実装されており、GPU ワークステーションからスーパーコンピュータまで様々な計算機を用いた計算が可能である。本稿では GaREUS 法を用いた自由エネルギー地形の計算方法を説明する。

## 計算機・ソフトウェア

GaREUS 法の実行には MD シミュレーションソフトウェア GENESIS を用いる。GENESIS はオープンソフトウェア (LGPLv3) として公開しており、無料で利用できる。最新版は GENESIS のウェブサイト (<https://www.r-ccs.riken.jp/labs/cbrt/>) から入手可能である。

GENESIS をコンパイル・実行するには、OS として Linux、C コンパイラおよび Fortran コンパイラとして gcc、gfortran または Intel Fortran/C++ compiler、開発環境として GNU autoconf、並列計算ライブラリーとして OpenMPI、数学ライブラリーとして BLAS と LAPACK が必要である。GPU を利用する場合は CUDA ライブラリーも必要となる。詳細な必要条件やコンパイル方法は GENESIS のウェブサイトかマニュアルを参考にしてほしい。

GaREUS 法では複数のレプリカ (MD シミュレーション) を同時に実行する必要があるため、複数の計算ノードを持ったコンピュータクラスターかスーパーコンピュータを用いる。Intel 社、AMD 社、富士通 (A64FX) の CPU を搭載しており、GPU を利用する場合は、CUDA 対応の nVIDIA 社の GPU を搭載したものをを用いる。例えば、10 レプリカの計算をする場合、8CPU コア以上を搭載した計算ノードが 10 個必要となる。複数ノードを用いた計算にはバッチジョブシステムの利用が必要だが、利用方法はソフトウェアや計算機システムごとに大きく異なるため、各システムのマニュアルを参照するか管理者に問い合わせしてほしい。

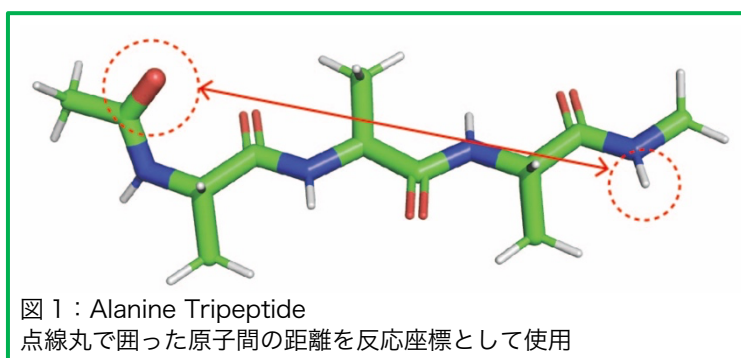
## 計算手順

- 1) 生体分子モデルのセットアップ
- 2) 構造最適化と平衡化
- 3) GaMD パラメータの決定
- 4) 本計算
- 5) 解析

## 実験の詳細

### 1) 生体分子モデルのセットアップ

例として Alanine Tripeptide を用いる (図 1)。GaREUS の反応座標として、残基 1 の酸素と残基 3 の水素の原子間距離を用いる (図 1 の点線丸)。Alanine Tripeptide に水分子を加え、一辺 50 Å 程度のボックスを作る。ここでは力場パラメータとして CHARMM 力場の C36 を用いる。タンパク質のセットアップについては、GENESIS の



ウェブサイト (<https://www.r-ccs.riken.jp/labs/cbrt/>) のチュートリアル (Level 1: Basic tutorials) および文献 8 で説明されているので、そちらを参照していただきたい。

### 2) 構造最適化と平衡化

セットアップで得た構造は原子同士が衝突していたり、不安定な構造になっていたりすることがある。ポテンシャルエネルギーが極端に高すぎるため、そのまま MD 計算をしてしまうとシステムが崩壊してしまう。構造最適化によってポテンシャルエネルギーを下げ、原子の衝突を取り除く。このとき、対象としているタンパク質やリガンドの位置や距離に拘束をかけ、分子が動かないようにする。GENESIS では最急降下法で構造最適化を行う。

構造最適化が終わったシステムは分子に速度が与えられていない絶対零度の状態になっている。分子に初期速度を与え、システムの温度を標的温度までゆっくりと上昇させる。GENESIS では Langevin 熱浴で等温等積 (NVT) の MD 計算を行う。

次に、等温等圧 (NPT) で MD 計算を行い、システムの体積を調整する。体積の時間変化を確認し、体積があまり変化しなくなるまでシミュレーションを行う。GENESIS では Bussi の方法で温度・圧力を調整する。

最後に、本計算と同じ熱力学条件で平衡化を行う。平衡化に必要なステップ数はシステムによって異なるため、温度、体積、ポテンシャルエネルギーの時間変化がドリフトしていないか注意する。構造のトラジェクトリーを見て、構造が壊れていないかも確認する。Root-Mean-Square Displacement (RMSD) や Root-Mean-Square Fluctuation (RMSF) 等で時間変化を確認し、あまり変化しなくなったところで平衡化を終える。構造最適化および平衡化についても GENESIS のウェブサイトのチュートリアル (Level 1: Basic tutorials) および文献 8 で GENESIS のインプットファイルも含めて詳細な説明されているので、そちらを参照していただきたい。

### 3) GaMD パラメータの決定

$$U'(\vec{x}) = U(\vec{x}) + \Delta U^{\text{GaMD}}(U(\vec{x})) \quad (1)$$

$$\Delta U^{\text{GaMD}}(U(\vec{x})) = \begin{cases} \frac{1}{2}k\{E - U(\vec{x})\}^2 & (U(\vec{x}) < E) \\ 0 & (U(\vec{x}) \geq E) \end{cases} \quad (2)$$

$$U_{\max} \leq E \leq U_{\min} + \frac{1}{k} \quad (3)$$

$$k(E - U_{\text{ave}})\sigma_U \leq \sigma_0 \quad (4)$$

$$k_0 = \min \left( 1, \frac{\sigma_0}{\sigma_U} \frac{U_{\max} - U_{\min}}{U_{\max} - U_{\text{ave}}} \right) \quad (5)$$

$$U_i''(\mathbf{x}) = U(\mathbf{x}) + \Delta U^{\text{GaMD}}(U(\mathbf{x})) + \Delta U_i^{\text{REUS}}(\xi(\mathbf{x})) \quad (6)$$

$$\Delta U_i^{\text{REUS}}(\xi(\mathbf{x})) = k(\xi_i - \xi(\mathbf{x}))^2 \quad (7)$$

$$\begin{aligned} p(\xi) &= \frac{\int \delta(\xi - \xi(x)) \exp[-\beta U(x)] dx}{\int \exp[-\beta U(x)] dx} \\ &= \frac{\int \delta(\xi - \xi(x)) \exp[\beta \Delta U^{\text{GaMD}}(x)] \exp[-\beta U'(x)] dx}{\int \delta(\xi - \xi(x)) \exp[-\beta U'(x)] dx} \frac{\int \delta(\xi - \xi(x)) \exp[-\beta U'(x)] dx}{\int \exp[-\beta U'(x)] dx} \\ &= \frac{\int \exp[\beta \Delta U^{\text{GaMD}}(x)] \exp[-\beta U'(x)] dx}{\int \exp[-\beta U'(x)] dx} \frac{p'(\xi)}{\langle \exp[\beta \Delta U^{\text{GaMD}}] \rangle_{\xi}'} \end{aligned} \quad (8)$$

図2 : GaREUS で用いる数式

GaMD ではシステムの本래のポテンシャル  $U(\mathbf{x})$  にバイアス  $\Delta U^{\text{GaMD}}(U(\mathbf{x}))$  を加えることで、エネルギー障壁を下げる (図2の式(1)および図3)。  $U(\mathbf{x})$  が閾値  $E$  よりも小さいときのみ調和ポテンシャル型のバイアスを加え、それ以外は何も足さない (図2の式(2))。このようにすることで、システムを局所安定な状態に捉えられることなく、状態空間を広くサンプリングすることができるようになる。  $U'(\mathbf{x})$  が連続関数かつ  $U(\mathbf{x})$  における大小関係 ( $U(\mathbf{x}_1) < U(\mathbf{x}_2)$  ならば  $U'(\mathbf{x}_1) < U'(\mathbf{x}_2)$ ) を維持するために、閾値  $E$  はポテンシャルの最大値  $U_{\max}$  と最小値  $U_{\min}$  によって条件づけられる (図2の式(3))。また、正確な Reweighting のために、  $U'(\mathbf{x})$  の分散は図2の式(4)を満たす必要がある。ここで、  $U_{\text{ave}}$  と

$\sigma_U$  はそれぞれポテンシャルの  $U(\mathbf{x})$  の平均値と分散で、 $\sigma_0$  はユーザーが設定する分散の最大値である。 $k_0$  を  $k(U_{\max}-U_{\min})$  と定義すると、閾値  $E$  を下限  $U_{\max}$  に設定する場合、 $k_0$  は図 2 の式(5)で決定する。ここでは最小限の説明しかしていないが、GaMD のパラメータの詳細は Miao の原著論文 (文献 1,2) を参照していただきたい。

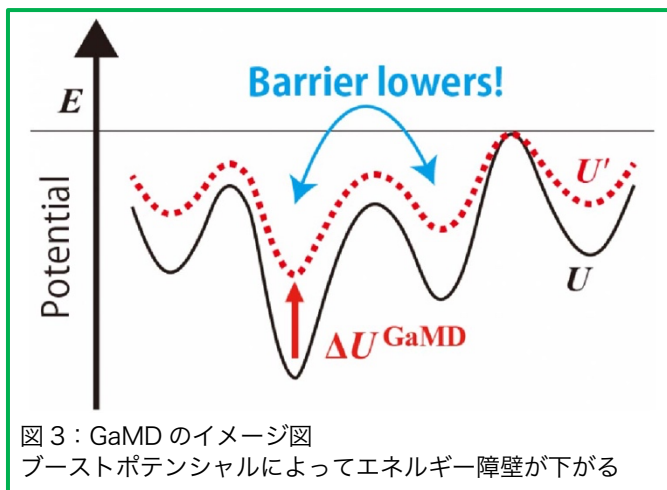


図 3 : GaMD のイメージ図  
ブーストポテンシャルによってエネルギー障壁が下がる

GaMD では二面角とそれ以外の全ポテンシャルエネルギーを別々にブーストする「デュアルブースト」が利用できる (文献 9)。二面角のブーストによってタンパク質の構造柔軟性を向上し、構造探索を加速する。同時に、二面角以外の全ポテンシャルエネルギーをブーストすることで、タンパク質近傍の水分子の運動等を加速し、ケージング効果等を回避する。デュアルブーストの場合は、上記の GaMD パラメータが二面角とそれ以外の全ポテンシャルエネルギーに対してそれぞれ必要となる。

$k$  と  $E$  を決めるには  $U_{\max}$ ,  $U_{\min}$ ,  $U_{\text{ave}}$ ,  $\sigma_U$  が必要となるが、系のポテンシャルエネルギー地形が複雑なため、一般的にはこれらは事前にわからない。まずブーストをかけずに MD 計算を行い、ポテンシャルの形状を探索し、これらのパラメータを決める。通常の MD のインプットに加えて、[GAMD]セクションを追加する (図 4)。それぞれのパラメータの意味は、gamd は GaMD をするか否か、boost=no はブーストしない、boost\_type=DUAL はデュアルブーストを用いる、thresh\_type=LOWER は式(3)の下限  $E_{\max}$  を用いる、sigma0\_dih と sigma0\_pot はそれぞれ二面角とそれ以外のポテンシャルの  $\sigma_0$  という意味である。update\_period は GaMD のパラメータ  $U_{\max}$ ,  $U_{\min}$ ,  $U_{\text{ave}}$ ,  $\sigma_U$  を更新する頻度で、この段階では MD 計算のステップ数を同じにしておく。

```
[OUTPUT]
dcdfile = run.dcd
rstfile = run.rst
gamdfile = run.gamd

[GAMD]
gamd = yes
boost = no
boost_type = DUAL
thresh_type = LOWER
sigma0_pot = 6.0
sigma0_dih = 6.0
update_period = 500000
```

図 4 : 初期パラメータ設定のための GaMD のインプットファイルの一部

計算が終わると [OUTPUT] の gamdfile で指定したファイルに図 5 のような結果が出力される。POT\_MAX, POT\_MIN, POT\_AVE, POT\_DEV が二面角以外のポテンシャルの  $U_{\max}$ ,  $U_{\min}$ ,  $U_{\text{ave}}$ ,  $\sigma_U$  であり、POT\_TH, POT\_KC, POT\_KC0 はそれらから計算した  $E$  と  $k$  となっている。同様に DIH が付いたものは二面角のパラメータである。

POT_KC0	STEP	POT_MAX	POT_MIN	POT_AVE	POT_DEV	POT_TH	POT_KC	POT_KC0
DIH_MAX	DIH_MIN	DIH_AVE	DIH_DEV	DIH_TH	DIH_KC	DIH_KC0		
0.1215	500000	-38023.6361	-38774.3320	-38363.3642	90.1782	-38023.6361	0.0002	1.000
19.3428		-1.7479	6.6143	2.3350	19.3428	0.0474		

図 5 : GaMD の初期パラメータの出力の例



初期パラメータが決まったら、実際に GaMD のブーストポテンシャルを加えながらパラメータを更新していく。[GAMD] で boost=yes とし、pot\_max や dih\_max 等に先ほどの gamdfile に出力された初期パラメータを入力する (図 6)。ブーストポテンシャルを加えることで探索できるポテンシャル表面の領域が広がり、 $U_{\max}$ ,  $U_{\min}$ ,  $U_{\text{ave}}$ ,  $\sigma_U$  が変化する。update\_period=500 とすると 500 ステップごとに、pot\_max や dih\_max 等を新たな  $U_{\max}$ ,  $U_{\min}$ ,  $U_{\text{ave}}$ ,  $\sigma_U$  を基に更新する。パラメータの更新を繰り返すことで、探索できるポテンシャル表面が徐々に広がっていき、より良い  $U_{\max}$ ,  $U_{\min}$ ,  $U_{\text{ave}}$ ,  $\sigma_U$  が見つかりやすくなる。

更新されたパラメータは update\_period ごとに gamdfile で指定されたファイルに出力される (図 7)。パラメータの時間変化を確認し、あまり変化しなくなったところで計算を終える。出力の最後の行の値で GaMD のパラメータを決定し、以降は変化させない。

```
[OUTPUT]
dcdfile = run.dcd
rstfile = run.rst
gamdfile = run.gamd

[GAMD]
gamd = yes
boost = yes
boost_type = DUAL
thresh_type = LOWER
sigma0_pot = 6.0
sigma0_dih = 6.0
update_period = 500
pot_max = -38023.6361
pot_min = -38774.3320
pot_ave = -38363.3642
pot_dev = 90.1782
dih_max = 19.3428
dih_min = -1.7479
dih_ave = 6.6143
dih_dev = 2.3350
```

図 6: パラメータを更新するための GaMD インプットファイルの一部

POT_KCO	STEP	POT_MAX	POT_MIN	POT_AVE	POT_DEV	POT_TH	POT_KC	POT_KCO	DIH_MAX	DIH_MIN	DIH_AVE	DIH_DEV	DIH_TH	DIH_KC	DIH_KCO
0.1835	500	-38023.6361	-38774.3320	-38192.8674	42.2050	-38023.6361	0.0002	0.1835	19.3428	-1.7479	7.8276	2.2655	19.3428	0.0474	1.0000
0.1789	1000	-38023.6361	-38774.3320	-38176.5333	42.1128	-38023.6361	0.0002	0.1789	19.3428	-1.7479	8.6129	2.4538	19.3428	0.0474	1.0000
0.1686	1500	-38023.6361	-38774.3320	-38164.9018	43.8404	-38023.6361	0.0002	0.1686	19.3428	-1.7479	8.8026	2.3800	19.3428	0.0474	1.0000
0.1590	2000	-38023.6361	-38774.3320	-38154.0682	45.6808	-38023.6361	0.0002	0.1590	19.3428	-1.7479	8.8343	2.3637	19.3428	0.0474	1.0000
0.1530	2500	-38014.2105	-38774.3320	-38145.8525	47.4304	-38014.2105	0.0002	0.1530	19.3428	-1.7479	8.6689	2.3160	19.3428	0.0474	1.0000
0.1505	3000	-37942.4394	-38774.3320	-38135.2229	51.8817	-37942.4394	0.0002	0.1505	19.3428	-1.7479	8.6011	2.2409	19.3428	0.0474	1.0000

図 7: GaMD のパラメータの更新の例

#### 4) 本計算

GaREUS では、GaMD のブーストポテンシャル  $\Delta U^{\text{GaMD}}(U(\mathbf{x}))$  を加えつつ、REUS シミュレーションを行う (図 2 の式(6))。  $\Delta U^{\text{REUS}}(\xi(\mathbf{x}))$  は REUS のバイアスポテンシャル、  $\xi(\mathbf{x})$  は REUS で用いる反応座標、  $i$  はレプリカの番号である。  $\Delta U_i^{\text{REUS}}(\xi(\mathbf{x}))$  は図 2 の式(7)のように調和ポテンシャル型になっており、レプリカごとに異なる反応座標  $\xi_i$  に系を拘束するようになっている。  $\xi_i$  を定期的に交換することで、エネルギー障壁にとらわれず、反応座標空間を広く探索できる。

本計算用のインプットファイルでは、[GAMD]に 3) で決定した GaMD のパラメータを入力する (図 8)。GaMD のパラメータを固定するために、update\_period は 0 にしておく。REUS も同時に行うので、[REMD]に REUS の設定を入力する。dimension はレプリカ交換のパラメータの次元で、ここでは 1 とする。exchange\_period は交換の頻度で、ここでは 1000 ステップに一度 REUS のパラメータを交換する。REUS の場合は type1 を RESTRAINT にする。nreplica1 はレプリカの数である。rest\_function1 では REUS で用いるバイアスポテンシャルとして、[RESTRAINTS]の拘束ポテンシャルの番号を指定する。[OUTPUT]で GaREUS で出力されるファイルを指定する。remfile は各レプリカにおけるパラメータ番号が出力され、計算結果をソーティングする際に利用する。

REUS で用いるバイアスポテンシャルは図 9 のように指定する。まず、[SELECTION]でバイアスをかける原子を指定する。ここでは図 1 の点線赤丸が囲った酸素原子と水素原子をそれぞれ group1, group2 で指定している。[RESTRAINTS]ではバイアスの関数を指定する。function が DIST とし、select\_index を 1 2 とすると、[SELECTION]で指定した group1 と group2 の原子間に距離の拘束ポテンシャルがかかる。調和ポテンシャルの係数と平衡値は、constant と reference で指定する。REUS で用いるレプリカ数に対応した数の値を指定する。図 9 のように reference の値をレプリカごとに変えることで、反応座標上のそれぞれの場所に拘束がかかることになる。

```
[OUTPUT]
logfile = rep{}.log
dcdfile = rep{}.dcd
rstfile = rep{}.rst
remfile = rep{}.rem

[GAMD]
gamd = yes
boost = yes
boost_type = DUAL
thresh_type = LOWER
sigma0_pot = 6.0
sigma0_dih = 6.0
update_period = 0
pot_max = -37735.1280
pot_min = -38774.3320
pot_ave = -37971.7854
pot_dev = 58.0212
dih_max = 23.8325
dih_min = -1.7479
dih_ave = 12.1909
dih_dev = 2.3182

[REMD]
dimension = 1
exchange_period = 1000
type1 = RESTRAINT
nreplica1 = 14
cyclic_params1 = NO
rest_function1 = 1
```

図 8 : GaREUS のインプットファイルの一部

```
[SELECTION]
group1 = an:OY and resno:1 # restraint group 1
group2 = an:HNT and resno:3 # restraint group 2

[RESTRAINTS]
nfunctions = 1
function1 = DIST
constant1 = 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
reference1 = 1.80 2.72 3.64 4.56 5.48 6.40 7.32 8.24 9.16 10.08 11.00 11.92 12.84 13.76
select_index1 = 1 2
```

図 9 : GaREUS で指定する反応座標パラメータ例

以上のインプットファイルを基に GENESIS で本計算を実行する。ログファイルに [REMD]の exchange\_period で指定したステップごとにレプリカの交換に関する情報が出力される (図 10)。この例では 4 と 5 が交換しており、AcceptanceRatio が 1 増えている。交換後のパラメータの番号とレプリカの番号 (トラジェクトリーの番号) との対応は ReplIDtoParmID と ParmIDtoReplID からわかる。[RESTRAINTS]で指定した反応座標のパラメータによっては交換率が低すぎたり、高すぎたりすることがあるが、その場

合はパラメータを再調節する。全レプリカで AcceptanceRatio が均一に 20~30%程度になるようにパラメータを選択するのが良いとされている。

```

REMD> Step:      1000 Dimension:  1 ExchangePattern:  2
Replica  ExchangeTrial  AcceptanceRatio  Before  After
  1 > 0 N 0 / 0 1 1
  2 > 3 RR 0 / 1 2 2
  3 > 2 RR 0 / 1 3 3
  4 > 5 AA 1 / 1 4 4
  5 > 4 AA 1 / 1 5 5
  6 > 7 RR 0 / 1 6 6
  7 > 6 RR 0 / 1 7 7
  8 > 9 RR 0 / 1 8 8
  9 > 8 RR 0 / 1 9 9
 10 > 11 RR 0 / 1 10 10
 11 > 10 RR 0 / 1 11 11
 12 > 13 RR 0 / 1 12 12
 13 > 12 RR 0 / 1 13 13
 14 > 0 N 0 / 0 14 14

Parameter : 1 2 3 5 4 6 7 8 9 10 1
12 13 14 2 3 5 4 6 7 8 9 10 1
RepIDtoParmID: 1 2 3 5 4 6 7 8 9 10 1
12 13 14 2 3 5 4 6 7 8 9 10 1
ParmIDtoRepID: 1 2 3 5 4 6 7 8 9 10 1
12 13 14 2 3 5 4 6 7 8 9 10 1
    
```

図 10 : GaREUS のログファイルの例  
レプリカ交換の結果が出力されている

## 5) 解析

GaREUS 法で得られたトラジェクトリーには GaMD のバイアス  $\Delta U^{GaMD}(U(x))$  と REUS のバイアス  $\Delta U^{REUS}(\xi(x))$  がかかっている。トラジェクトリーから直接自由エネルギー地形を計算してしまうと、これらのバイアスがかかった不正確な地形が得られてしまう。正確な自由エネルギー地形を得るために、GaREUS のトラジェクトリーの確率分布を Reweighting することでバイアスを取り除く必要がある。ここでは二段階の Reweighting を行う。まず REUS のバイアスを取り除き、続いて GaMD のバイアスを取り除く。

本計算で得られた各レプリカのトラジェクトリーでは、REUS のパラメータ交換によって1つのレプリカ内でバイアスのパラメータが時間によって変化している。つまり、トラジェクトリー内の各スナップショットには異なるバイアスがかかっている。REUS のバイアスの効果を取り除くにはバイアスパラメータごとにスナップショットがまとめられている方が扱いやすいので、まずトラジェクトリーをパラメータごとにソーティング。GENESIS の remd\_convert に図 11 のようなインプットを入力し、ソーティングする。[INPUT] の dcdfile, remfile, logfile にそれぞれ GaREUS で得た各レプリカのトラジェクトリー (dcd ファイル)、

```

[INPUT]
reffile = ../../1_setup/build/ala3.pdb # PDB file
dcdfile = ../1_remove_water/rep{}.dcd # DCD file
remfile = ../../6_prod/rep{}.rem
logfile = ../../6_prod/rep{}.log

[OUTPUT]
pdbfile = par.pdb # PDB file
trjfile = par{}.dcd # trajectory file
logfile = par{}.log # trajectory file

[SELECTION]
group1 = all # selection group 1

[FITTING]
fitting_method = NO # method

[OPTION]
check_only = NO # (YES/NO)
convert_type = PARAMETER # (REPLICA/PARAMETER)
convert_ids = # (empty = all)(example: 1 2 5-10)
num_replicas = 14
nsteps = 500000
exchange_period = 1000
eneout_period = 50
crdout_period = 50
trjout_format = DCD # (PDB/DCD)
trjout_type = COOR+BOX # (COOR/COOR+BOX)
trjout_atom = 1
pbc_correct = NO # (NO/MOLECULE)
    
```

図 11 : remd\_convert のインプット例



remfile、logfile を指定し、[OPTION] にステップ数や exchange\_period 等の対応する値を指定する。ソーティングされたファイルは[OUTPUT]で指定した trjfile や logfile に出力される。出力された dcd ファイル内では、1 つのバイアスパラメータのみが用いられている（例えば par1.dcd では 1 番目のパラメータのみ）。

ソーティングされたトラジェクトリーから反応座標の値を計算する。図 12 のようなインプットファイルを GENESIS の trj\_analysis に入力する。[TRAJECTORY]の trjfile に先ほどソーティングして得た dcd ファイルを指定し、md\_step, mdout\_period に GaREUS で用いた MD のステップ数と出力頻度を指定する。2 原子間の距離の場合は[OPTION]に distance を指定すると、トラジェクトリーから距離を計算し、[OUTPUT]の disfile で指定したファイルに出力される。二面角を使う場合は distance の代わりに torsion、角度の場合は angle とし、[OUTPUT]ではそれぞれ disfile の代わりに torfile, angfile でファイルを指定する。すべてのトラジェクトリー（つまりすべてのレプリカの dcd ファイル）に対して同様の計算を行う。

```
[INPUT]
reffile      = ../../1_setup/build/ala3.pdb      # PDB file

[OUTPUT]
disfile      = cv1.dat                          # distance file

[TRAJECTORY]
trjfile1     = par1.dcd      # trajectory file
md_step1    = 5000000      # number of MD steps
mdout_period1 = 50        # MD output period
ana_period1 = 50          # analysis period
repeat1     = 1
trj_format  = DCD          # (PDB/DCD)
trj_type    = COOR+BOX    # (COOR/COOR+BOX)
trj_natom   = 0           # (0:uses reference PDB atom count)

[OPTION]
check_only  = NO
distance1   = PROA:1:ALA:OY  PROA:3:ALA:HNT
```

図 12 : trj\_analysis のインプット例

MultiState Bennett Acceptance Ratio (MBAR) を用いて、REUS のバイアスを取り除く（文献 10）。図 13 のようなインプットファイルを GENESIS の mbar\_analysis に入力する。反応座標の結果を[INPUT]の cvfile に指定し、[RESTRAINTS]に REUS で用いた調和ポテンシャルバイアスのパラメータを指定する。[OUTPUT]で weightfile, fenefile, pmffile を指定すると、それぞれ重み、レプリカ間の自由エネルギー変化、反応座

```
[INPUT]
cvfile       = ../3_calc_cv/cv{}.dat

[OUTPUT]
weightfile   = output/weight{}.dat
fenefile     = output/fene.dat
pmffile      = output/pmf.dat

[MBAR]
dimension    = 1
nblocks     = 1
self_iteration = 10
temperature  = 300.0
tolerance    = 10E-08
rest_function1 = 1
grids1      = 0.0 15.0 301

[RESTRAINTS]
constant1    = 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2
reference1   = 1.80 2.72 3.64 4.56 5.48 6.40 7.32 8.24 9.16 10.08 11.00 11.92 12.84 13.76
is_periodic1 = NO
```

図 13 : mbar\_analysis のインプット例

標に沿った自由エネルギー地形 (PMF) を出力できる。[MBAR]で MBAR の設定を指定する。temperature は計算で用いた温度にし、rest\_functon で[RESTRAINTS]の関数を指定する。grids で PMF のグリッドを指定でき、0.0 15.0 301 とすると、反応座標が 0.0 から 15.0 の範囲で等間隔で 301 個のグリッドごとに PMF を出力する。mbar\_analysis で特に重要なものは、weightfile で出力される重みで、これは GaREUS で得た各スナップショットごとの確率に対応している。この重みを用いることで、GaREUS で指定していない反応座標空間にも射影することで、REUS のバイアスがかかっていない自由エネルギー地形を求めることができる。ただ、mbar\_analysis で取り除けるバイアスは REUS のもののみで、GaMD のバイアスは取り除けていない。

最後に、GaMD のバイアスを取り除く。バイアスがかかっていない確率分布  $p(\xi)$  は図 2 の式(8)のように変形できる。ここでダッシュ記号がついているものは GaMD のバイアスがかかった分布や平均値である。MBAR によって REUS のバイアスは取り除かれているので、mbar\_analysis で得られた重みから  $p'(\xi)$  は計算できる。 $\exp[\beta\Delta U^{\text{GaMD}}]$  の平均は統計的に収束が非常に悪く、そのまま計算すると誤差が非常に大きくなる。そのため、この項はキュムラント展開をし、二次の項までで近似する (文献 11)。これは  $\Delta U^{\text{GaMD}}$  の分布を正規分布 (Gaussian) で近似していることに対応する。ソーティングした logfile に出力されている GaMD のブーストポテンシャル  $\Delta U^{\text{GaMD}}$  と mbar\_analysis で得た重みから、 $p(\xi)$  を計算する。現在 GENESIS の Analysis Tools には GaMD バイアスの Reweighting 用のプログラムは用意されていない。今後 GENESIS のウェブサイトで公開予定の GaREUS のチュートリアルで Reweighting 用の Python スクリプトを配布する予定である。

図 14 は反応座標に沿った自由エネルギー地形の結果である。GaMD のバイアスがかかったもの (つまり MBAR で REUS のバイアスだけ取り除いたもの) ではエネルギー障壁が低い、全バイアスを取り除いたもの (Unbiased) ではエネルギー障壁が高くなっている。GaMD によって小さくなった障壁が Reweighting によって正しく戻されていることがわかる。

$p(\xi)$  での反応座標  $\xi$  は、GaREUS で用いた反応座標と同じである必要はなく、任意の反応座標に対して計算することができる。複数の反応座標にも対応している。例えば、Alanine Tripeptide の 2 つの二面角  $\phi$  (Ala1C-Ala2N-Ala2C $\alpha$ -Ala2C) と  $\psi$  (Ala2N-Ala2C $\alpha$ -Ala2C-Ala3N) を用いて二次元の自由エネルギー地形を計算すると図 15 のようになる。GaMD のバイアスがかかっている状態 (GaMD biased) だと全体的に浅い地形になっており、ベイソン間を移動しやすくなっている。バイアスを取り除いた地形 (Unbiased) はエネルギー障壁が元通りに高くなっている。このように GaMD のバイアスがかかることで、REUS で用いた反応座

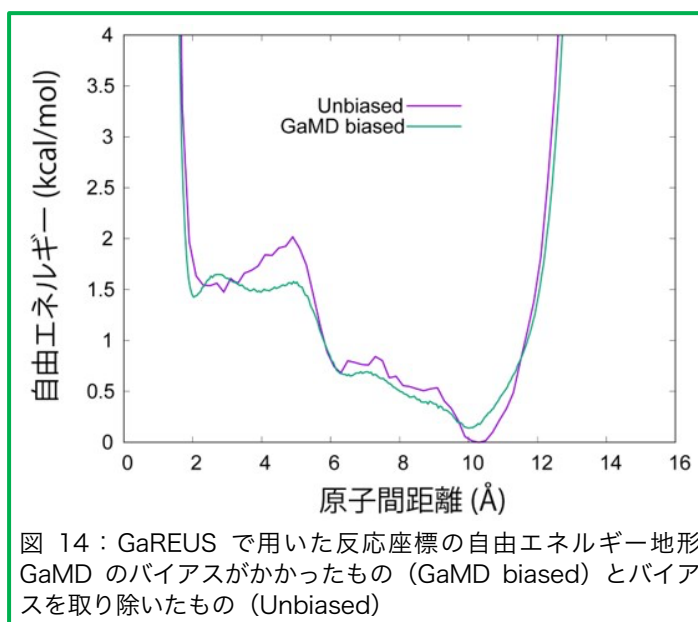
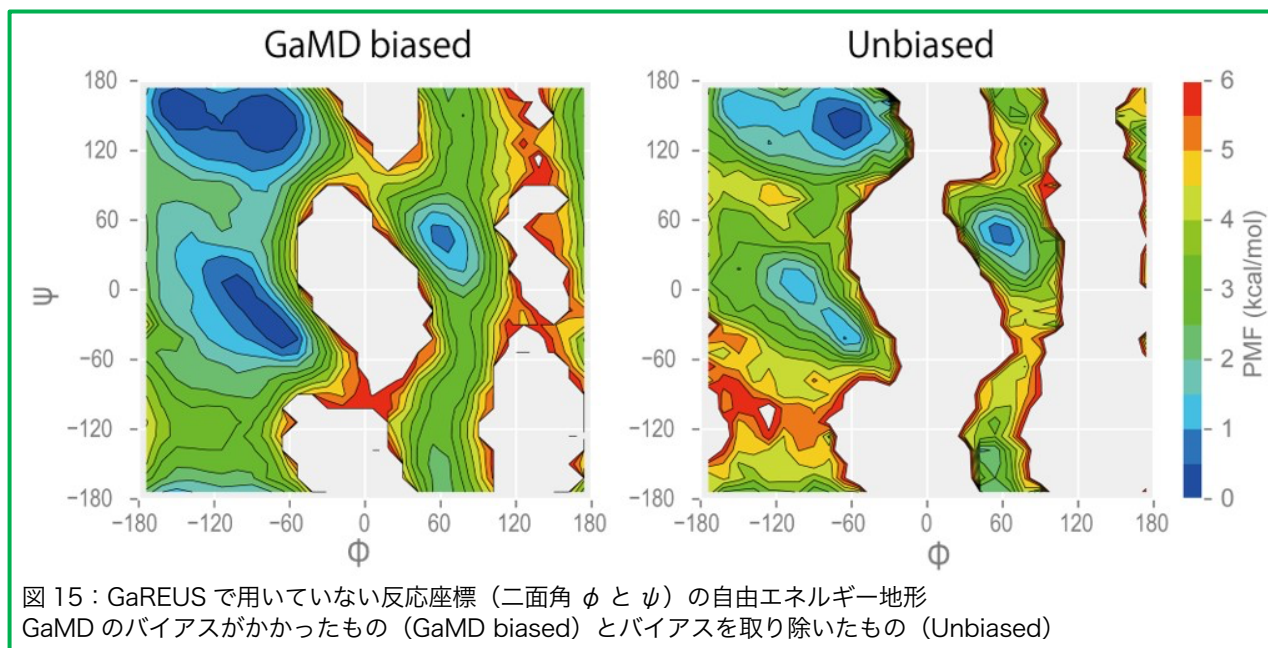


図 14 : GaREUS で用いた反応座標の自由エネルギー地形 GaMD のバイアスがかかったもの (GaMD biased) とバイアスを取り除いたもの (Unbiased)

標以外の自由度に対しても動きやすくなっており、広く構造探索ができるようになり、Reweighting によって正しい自由エネルギー地形に戻ることができる。



以上が GaREUS による自由エネルギー計算のプロトコルであり、通常の REUS シミュレーションと比較して効率的に自由エネルギー地形を求めることができる。計算時間は系に含まれる原子数に大きく依存するため、一概に目安は述べることは難しい。対象とする系で通常の MD 計算を行い、計算時間を見積もってから GPU クラスタやスーパーコンピュータで計算することをお勧めする。また、GaREUS 法で用いた入力パラメータ等の詳細や一覧については、GENESIS のマニュアルにまとめられているので、そちらを参照してほしい。

## 文献

- [1] Miao, Y. et al., *J. Chem. Theory Comput.*, **11**, 3584–3595 (2015)
- [2] Pang, Y. T. et al., *J. Chem. Theory Comput.*, **13**, 9–19 (2017)
- [3] Sugita, Y. et al., *J. Chem. Phys.*, **113**, 6042–6051 (2000)
- [4] Fukunishi, H. et al., *J. Chem. Phys.*, **116**, 9058–9067 (2002)
- [5] Oshima, H. et al., *J. Chem. Theory Comput.*, **15**, 5199–5208 (2019)
- [6] Jung, J. et al., *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, **5**, 310–323 (2015)
- [7] Kobayashi, C. et al., *J. Comput. Chem.*, **38**, 2193–2206 (2017)
- [8] 小林千草, *蛋白質科学会アーカイブ*, **11**, e091 (2018)
- [9] Hamelberg, D. et al., *J. Chem. Phys.*, **120**, 11919–11929 (2004)
- [10] Shirts, M. R. & Chodera, J. D., *J. Chem. Phys.*, **129**, 124105 (2008)
- [11] Miao, Y. et al., *J. Chem. Theory Comput.*, **10**, 2677–2689 (2014)



このコンテンツは [クリエイティブ・コモンズ 表示 - 非営利 - 改変禁止 4.0 国際 ライセンス](https://creativecommons.org/licenses/by-nc-nd/4.0/) の下に提供されています。

© 日本蛋白質科学会 2021 Licensed under [クリエイティブ・コモンズ 表示 - 非営利 - 改変禁止 4.0 国際 ライセンス](https://creativecommons.org/licenses/by-nc-nd/4.0/)