

2016年度 ポスト「京」重点課題1

# GENESIS講習会(実習)

理化学研究所 生命システム研究センター  
分子機能シミュレーション研究チーム

李秀栄、尾嶋拓、新津藍、杉田有治

(資料提供: 小林千草)

2016年9月8日 FOCUS(神戸)

# 今日の内容

## ■ 13:30 – 13:50 講義

- 生体分子の分子動力学計算
- GENESISの特徴

## ■ 休憩(10分)

## ■ 14:00 – 16:00 実習

- GENESISのコンパイル
- 実行例(1) 分子動力学法計算と解析
- 実行例(2) レプリカ交換分子動力学計算と解析

# 今日の内容(改定)

## ■ 13:30 – 13:45 講義

- 生体分子の分子動力学計算
- GENESISの特徴

## ■ 13:45 – 14:45 実習(1)

- GENESISのコンパイル
- 分子動力学法計算と解析

## ■ 休憩(15分)

## ■ 15:00 – 16:00 実習(2)

- レプリカ交換分子動力学計算と解析

# 端末操作と利用ソフト

## ■ FOCUSスパコンへのログイン

- Xwin Serverを起動
- Cygwin64 Terminalを起動
- Terminalで「ssh -Y ユーザ名@ff」と入力

## ■ データ転送

- WinSCPを使用

## ■ トラジェクトリーデータの可視化

- VMD1.9.2を使用

(※)端末からログアウトすると、ユーザにより生成された全てのデータは消去されます。

# 実習材料

## ■ 実習で使用する材料は以下のディレクトリにあります

[../share/Lecture\\_0908/](#)

### **GENESIS\_binary:** (GENESISのバイナリファイル)

- spdyn (GENESIS, spdyn)
- prst\_setup (並列I/Oのセットアップ)
- crd\_convert (トラジェクトリコンバータ)

### **Compile:** (GENESISのコンパイル)

genesis-1.1.0 (最新のGENESIS code)

### **Test:** (テスト計算用のファイル)

- Regression\_test/
- test\_common
- test\_atdyn
- test\_spdyn
- test\_remd ...

### **Tutorial\_1:** (Tutorial 1 : タンパク質のMD)

- 1\_setup
- 2\_minimization
- 3\_heating
- 4\_equilibration
- 5\_production
- 6\_analysis

### **Tutorial\_2:** (Tutorial 3: タンパク質のレプリカ交換MD)

- 1\_setup
- 2\_minimization
- 3\_equilibration
- 4\_production
- 5\_analysis

# GENESISのコンパイル (1)

## ■ 推奨されるコンパイラ

インテル、富士通コンパイラ、または、gfortran 4.4.7より新しいもの

※Analysis toolのコンパイルにはLAPACKが必要になります

## GENESIS-1.1.0のソースツリー

|                 |              |                        |                           |
|-----------------|--------------|------------------------|---------------------------|
| ..:             |              | depcomp                |                           |
| COPYING         |              | install-sh             |                           |
| README          | 簡単な説明        | lib/                   | 共通コード                     |
| src/            | ソースコード       | Missing                |                           |
| bin/            | バイナリ(コンパイル後) | spdyn/                 | spdynのコード                 |
| <b>./src:</b>   |              | <b>./src/analysis:</b> |                           |
| GENESIS_VERSION |              | Makefile.am            |                           |
| Makefile.am     |              | Makefile.in            |                           |
| Makefile.in     |              | libana/                | analysis共通コード             |
| aclocal.m4      |              | pcaana/                | PCA解析                     |
| analysis/       | 解析ツールのコード    | rpath_generator/       | rpath生成コード                |
| atdyn/          | atdynのコード    | rstcnv/                | rst_convertのコード           |
| bootstrap       |              | trjana/                | trj_analysisのコード          |
| cleanup         |              | trjcnv/                | crd/pcrd/remd_convertのコード |
| config.h.in     |              |                        |                           |
| configure       |              |                        |                           |
| configure.ac    |              |                        |                           |

# GENESISのコンパイル (2)

- ../share/Lecture\_0908/Compileを自分のディレクトリにコピーし、GENESISのsrcディレクトリに移動してください

```
% cp -r ../share/Lecture_0908/Compile .  
% cd Compile/genesis-1.1.0/src
```

- コンパイルはmakeを使います

```
% ./configure --host=FOCUS (もしくは、FCFLAGS="-axAVX,SSE4.2 -O3 -ip -mkl=parallel" ./configure)  
% make install
```

- 出来上がったバイナリはgenesis-1.1.0/binに作成されます
  - GENESISはMD、トラジェクトリ変換プログラム、解析プログラム合わせて20個のアプリケーションを作成します
  - 今回は、時間の都合上、既にコンパイル済みのアプリケーションを使います

```
実行ファイルの場所: ../share/Lecture_0908/GENESIS_binary
```

# FOCUSスパコンシステムの利用について

## ■ FOCUSスパコンシステム概要

- フロントエンドサーバ(RHEL 6.6)、演算サーバ(OS:CentOS 6.6)
- 開発環境: Glibc、GNU CC、Intel Parallel Studio XE
- ジョブ管理システム: SLURMベース

- 端末利用室に設置の各種端末はインターネットに接続されておられません。このため「FOCUSスパコン利用の手引き」は以下のサイト(FOCUSスパコンネットワーク内)より閲覧してください。

<http://fl01.j-focus.jp/ablog/usermanual/>



# 演算ノードとキュー構成

(<http://www.j-focus.jp>)

|       | ノード数 | 理論演算性能 (GFlops) | コア数    | メモリ (GB) | HDD容量 (GB) | ノード間ネットワーク                | 備考                 |
|-------|------|-----------------|--------|----------|------------|---------------------------|--------------------|
| Aシステム | 224  | 108             | 12     | 48       | 500        | IB-QDR 40Gbps             | ノード間演算並列指向         |
| Bシステム | 2    | 119             | 16     | 512      | 600        | IB-QDR 40Gbps             | プリポスト・大容量メモリ処理用    |
| Cシステム | 22   | 108             | 12     | 48       | 500        | Gigabit Ethernet 1Gbps    | ノード内並列             |
| Dシステム | 80   | 400             | 20     | 64       | 6000       | IB-FDR 56Gbps             | ノード内並列／ノード間高並列演算兼用 |
| Eシステム | 48   | 4444            | 20+240 | 128      | 2000       | IB-FDR 56Gbps             | コプロセッサ(4基)搭載システム   |
| Gシステム | 4    | 1251            | 12+60  | 64       | 1000       | 10Gigabit Ethernet 10Gbps | コプロセッサ(1基)搭載システム   |

| キュー名  | 最大ジョブ数 | 最長実行時間 | 最大ジョブ投入数 | 最大ノード数(※1) | ジョブあたり最大ノード数(※1) | システム | 備考 |
|-------|--------|--------|----------|------------|------------------|------|----|
| a024h | 未設定    | 24時間   | 無制限      | 224        | 224              | A    |    |
| a096h | 未設定    | 96時間   | 5/ユーザ    | 100        | 100              | A    |    |
| b024h | 未設定    | 24時間   | 無制限      | 2          | 2                | B    |    |
| b096h | 未設定    | 96時間   | 5/ユーザ    | 1          | 1                | B    |    |
| c024h | 未設定    | 24時間   | 無制限      | 22         | 22               | C    |    |
| c096h | 未設定    | 96時間   | 5/ユーザ    | 10         | 10               | C    |    |
| c006m | 未設定    | 6分     | 5/ユーザ    | 2          | 2                | C    | ※2 |
| d024h | 未設定    | 24時間   | 無制限      | 80         | 80               | D    |    |
| d072h | 未設定    | 72時間   | 20/ユーザ   | 40         | 40               | D    |    |
| e024h | 未設定    | 24時間   | 無制限      | 48         | 16               | E    |    |
| e072h | 未設定    | 72時間   | 8/ユーザ    | 24         | 8                | E    |    |
| g024h | 未設定    | 24時間   | 無制限      | 4          | 4                | G    |    |

# ジョブ管理コマンド

([http://www.j-focus.jp/user\\_guide/ug0000000000.html](http://www.j-focus.jp/user_guide/ug0000000000.html))

表4.2 ジョブ管理コマンド一覧(SLURMコマンド編)

| コマンド      | コマンド用途               | 手順                                  |
|-----------|----------------------|-------------------------------------|
| sinfo -s  | キュー(パーティション)の情報を表示する | <u>4.1.2.キュー情報の確認方法</u>             |
| squeues   | キューのノード実行状況を表示する     |                                     |
| freenodes | 空ノード数を表示する           | <u>4.1.3. 利用可能なノード数の確認方法</u>        |
| sbatch    | ジョブを投入する             | <u>4.2.1.ジョブ投入コマンド(sbatch)</u>      |
| squeue    | ジョブの状態を表示する          | <u>4.2.2.ジョブ情報表示コマンド(squeue)</u>    |
| scancel   | ジョブをキャンセルする          | <u>4.2.3.ジョブのキャンセルコマンド(scancel)</u> |

# GENESISテストセット

- GENESISは、正しくコンパイルされたか確認するために、開発者が実施した計算の入出力ファイルをテストセットとして提供します

```
% cp -r ../share/Lecture_0908/Test .  
% cd Test
```

## GENESIS-1.1.0テストセットのツリー

|                         |               |                   |                  |
|-------------------------|---------------|-------------------|------------------|
| ./Test/regression_test: |               | test_remd.py      |                  |
| build/                  | 計算用の初期ファイル    | test_rpath.py     |                  |
| charmm.py               |               | test_rpath_atdyn/ | rpath(atdyn)のテスト |
| genesis.py              |               | test_rpath_spdyn/ | rpath(spdyn)のテスト |
| param/                  | パラメータファイル     | test_spdyn/       | spdynのテスト        |
| run.sh                  | バッチジョブ用のスクリプト |                   |                  |
| test.py                 |               |                   |                  |
| test_atdyn/             | atdynのテスト     |                   |                  |
| test_common/            | 共通テスト         |                   |                  |
| test_nonstrict.py       |               |                   |                  |
| test_parallel_IO/       | パラレルIOのテスト    |                   |                  |
| test_remd/              | レプリカ交換のテスト    |                   |                  |
| test_remd.csh           |               |                   |                  |

# テストジョブの実施

## ■ SLURM用バッチスクリプト(run.sh)を編集します

```
#!/bin/bash
#SBATCH -p c006m                # キュー名
#SBATCH -N 1                    # ノード数
#SBATCH --ntasks-per-node=8     # ノードあたりのプロセス数
#SBATCH -c 1                    # スレッド数
#SBATCH -J run                  # ジョブ名
#SBATCH -e run.e%J              # 標準エラー出力
#SBATCH -o run.o%J              # 標準出力

module load PrgEnv-intel        # 環境設定読み込み
module load intel/openmpi165

export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK}
bindir=/home1/gleu/share/Lecture_0908/GENESIS_binary
./test.py "mpirun -np 8 ${bindir}/spdyn" # 実行ファイルのディレクトリ名
                                           # テストジョブの実行
```

## ■ ジョブを投入します

```
% sbatch run.sh
% sjstat                # 投入ジョブの確認
```

出力ファイル(run.oXXX)を開き結果を確認します : Passed: 45/45?

# Tutorial 1: タンパク質のMD (1)

## ■ 計算内容

水中のBPTI(牛膵臓トリプシン阻害酵素)のMD計算

参照: <http://www.aics.riken.jp/labs/cbrt/tutorial/>

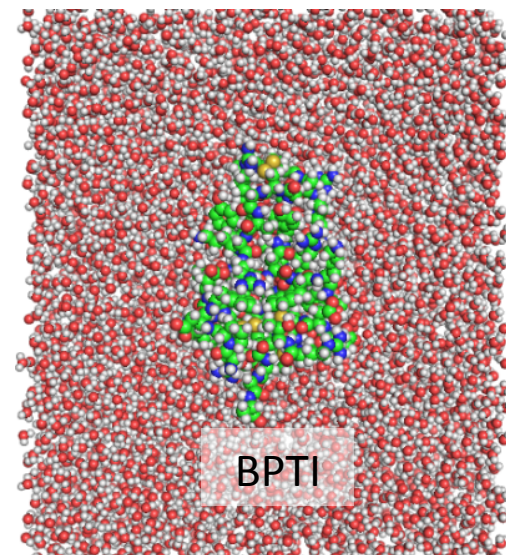
Basic molecular dynamics simulations

## ■ 計算データ

```
% cp -r ../share/Lecture_0908/Tutorial_1 .  
% cd Tutorial_1
```

Tutorial\_1の中身

```
..  
1_setup/           # セットアップ  
2_minimization/    # 最適化  
3_heating/         # 昇温  
4_equilibration/   # 平衡化  
5_production/      # プロダクション  
6_analysis/        # 解析
```



# システムのセットアップ(1\_setup)

- GENESISは、現在インプット用の分子ファイルを作成する部分は存在せず、CHARMMやVMDで作成された分子ファイル(top/par)を読み込むことができます

- 今回は時間の都合により、予め作成した分子ファイル(ionize.pdb、ionize.psf)を用います
- 分子ファイル作成の詳細は、HPをご覧ください

## ■ 初期構造の確認

- WinSCPを使って端末にionize.pdbとionize.psfをダウンロードします
  - ホスト名:「ff」
- VMDを起動し、ionize.psf、ionize.pdbの順で読み込みます
  - 「File」→「New Molecule...」
  - Filename: の「Browse...」で「ionize.psf」を選択して「load」
  - Load files for: が「0: ionize.psf」になっている状態で、Filename:「Browse...」で「ionize.pdb」を選択して「load」

# 構造最適化(2\_minimization)

- 初期構造では、原子間の距離が近すぎたり、不自然な原子結合を持つようなことがあります。不安定なシミュレーションを避けるため、本計算を行う前に「平衡化」と呼ばれる予備計算を十分に行います。
- 構造最適化は、その第1段階でエネルギー値が下がる方向に粒子を動かします。GENESISでは現在、最急降下法(Steepest Decent)を使うことができます

```
./2_minimization:  
run.inp          # GENESISの入力ファイル  
run.sh           # FOCUS用のバッチスクリプト  
output           # 計算結果
```

# 構造最適化の入力ファイル

## [INPUT]

```
topfile = ../1_setup/top_all27_prot_lipid.rtf # topology file
parfile = ../1_setup/par_all27_prot_lipid.prm # parameter file
psffile = ../1_setup/ionize.psf             # protein structure file
pdbfile = ../1_setup/ionize.pdb             # PDB file
reffile = ../1_setup/ionize.pdb             # reference for restraints
```

## [OUTPUT]

```
dcdfile = run.dcd # DCD trajectory file
rstfile = run.rst # restart file
```

## [ENERGY]

```
electrostatic = PME # [CUTOFF,PME]
switchdist = 10.0 # switch distance
cutoffdist = 12.0 # cutoff distance
pairlistdist = 13.5 # pair-list distance
pme_ngrid_x = 72 # grid size_x in [PME]
pme_ngrid_y = 80 # grid size_y in [PME]
pme_ngrid_z = 72 # grid size_z in [PME]
contact_check = YES
```

## [MINIMIZE]

```
nsteps = 1000 # number of steps
eneout_period = 100 # energy output period
crdout_period = 100 # coordinates output period
```

```
rstout_period = 1000 # restart output period
```

## [BOUNDARY]

```
type = PBC # [PBC,NOBC]
box_size_x = 70.8250 # box size (x) in [PBC]
box_size_y = 83.2579 # box size (y) in [PBC]
box_size_z = 69.0930 # box size (z) in [PBC]
```

## [SELECTION]

```
group1 = an:CA | an:C | an:O | an:N # index of restraint group 1
```

## [RESTRAINTS]

```
nfunctions = 1 # number of functions
function1 = POSI # restraint function 1
constant1 = 10.0 # force constant
select_index1 = 1 # restrained group
```

Coulomb相互作用で  
PMEを用いる

位置拘束  
どの粒子に拘束をかけるのかは  
[SELECTION]で選ぶ  
([SELECTION]に関しては次ページ参照)

長距離相互作用で用いるFFTの  
グリッド数  
(2,3,5の倍数かつ、並列数によって最適値が変わる、一般的にはグリッドサイズが1Å程度になるようにする。詳しくはマニュアルの5.3.1を参照)

原子間距離を確認し「Bad contact」や不自然な距離をレポートする (本計算には使用しない)

ステップ数



# GENESISの「selection」

Table. Available keywords and operators in group.

| expression                    | meaning                   | example    | other available expression |
|-------------------------------|---------------------------|------------|----------------------------|
| <i>an:name</i>                | atom name                 | an:CA      | atomname, atom_name        |
| <i>ai:number[-[number]]</i>   | atom index <sup>*1</sup>  | ai:1-5     | atomindex, atomidx         |
| <i>atno:number[-[number]]</i> | atom number <sup>*1</sup> | atno:6     | atomno                     |
| <i>rnam:name</i>              | residue name              | rnam:GLY   | residuenname, resname      |
| <i>rno:number[-[number]]</i>  | residue number            | rno:1-5    | residueno, resno           |
| <i>mname:name</i>             | molecule name             | mname:molA | moleculename, molname      |
| <i>segid:ID</i>               | segment index             | segid:PROA | segmentid, sid             |
| hydrogen                      | hydrogen atoms            |            | hydrogenatom               |
| heavy                         | heavy atoms               |            | heavyatom                  |
| all                           | all atoms                 |            | *                          |
| and                           | conjunction               |            | &                          |
| or                            | logical add               |            |                            |
| not                           | negation                  |            | !                          |
| ()                            | assemble                  |            |                            |

(GENESIS User Guide 1.1.0)

# 構造最適化の実行

## ■ ジョブの投入

squeuesでキューのノード実行状況を調べて、  
run.shを適宜編集してください

```
% cd 2_minimization
% sbatch run.sh
% sjstat
```

# 投入ジョブの確認

## ■ ジョブ終了後、以下のファイルが出力されているはずです

```
run.dcd      # dcd形式のトラジェクトリファイル (binary)
run.out      # GENESIS出力ファイル (ascii)
run.rst      # GENESISリスタートファイル (binary)
```

例: 出力ファイル(run.out)

```
*****
*
*              GENESIS SPDYN
*
*      A Molecular Dynamics Simulator with
*      Spatial Decomposition Scheme
*
*      Developed by RIKEN AICS
*
*****

[STEP0] Architecture and Compiler Information
```

(※)  
計算しなおすすめ場合は、  
dcdとrstファイルを消し  
て下さい

# 出力ファイルの見方

## ■ 出力ファイルは6段階に分かれている

[STEP 1]: 入力パラメータの確認

[STEP 2]: 並列数(プロセス数、スレッド数)の確認

[STEP 3]: 分子・エネルギー関数情報の確認

[STEP 4]: 初期座標のエネルギー計算結果

[STEP 5]: シミュレーション計算結果

| INFO: | STEP | POTENTIAL_ENE   | RMSG    | BOND       | ANGLE     | UREY-BRADLEY | DIHEDRAL | IMPROPER | CMAP     | VDWAALS    |
|-------|------|-----------------|---------|------------|-----------|--------------|----------|----------|----------|------------|
| ELECT |      | RESTRAINT_TOTAL |         |            |           |              |          |          |          |            |
| INFO: | 0    | -101787.3074    | 30.2325 | 11248.6855 | 2939.6563 | 74.9556      | 260.8343 | 62.6065  | -72.0093 | 11784.2874 |
|       |      | -128086.3236    | 0.0000  |            |           |              |          |          |          |            |
| INFO: | 100  | -114610.3616    | 5.4209  | 3975.9892  | 2354.4594 | 51.4906      | 256.4631 | 22.6729  | -74.3336 | 9727.0726  |
|       |      | -130925.2074    | 1.0316  |            |           |              |          |          |          |            |

[STEP 6]: 終端処理と計測時間

|  |         |         |            |         |              |         |         |        |        |
|--|---------|---------|------------|---------|--------------|---------|---------|--------|--------|
| Output_Time> Averaged timer profile (Min, Max) | nonbond | =       | 83.908 (   | 83.701, | 84.087)      |         |         |        |        |
| total time                                     | =       | 95.076  | pme real   | =       | 72.354 (     | 70.931, | 73.537) |        |        |
| setup  | =       | 2.334   | pme recip  | =       | 11.545 (     | 10.218, | 13.149) |        |        |
| dynamics                                       | =       | 92.742  | restraint  | =       | 0.011 (      | 0.005,  | 0.018)  |        |        |
| energy   | =       | 86.844  | integrator |         |              |         |         |        |        |
| integrator                                     | =       | 0.319   | constraint | =       | 0.000 (      | 0.000,  | 0.000)  |        |        |
| pairlist                                       | =       | 5.858 ( | 5.740,     | 5.991)  | update       | =       | 0.000 ( | 0.000, | 0.000) |
| energy   |         |         |            |         | comm_coord   | =       | 0.000 ( | 0.000, | 0.000) |
| bond   | =       | 0.271 ( | 0.261,     | 0.293)  | comm_force   | =       | 0.000 ( | 0.000, | 0.000) |
| angle  | =       | 0.276 ( | 0.252,     | 0.300)  | comm_migrate | =       | 0.000 ( | 0.000, | 0.000) |
| dihedral                                       | =       | 0.086 ( | 0.057,     | 0.119)  |              |         |         |        |        |

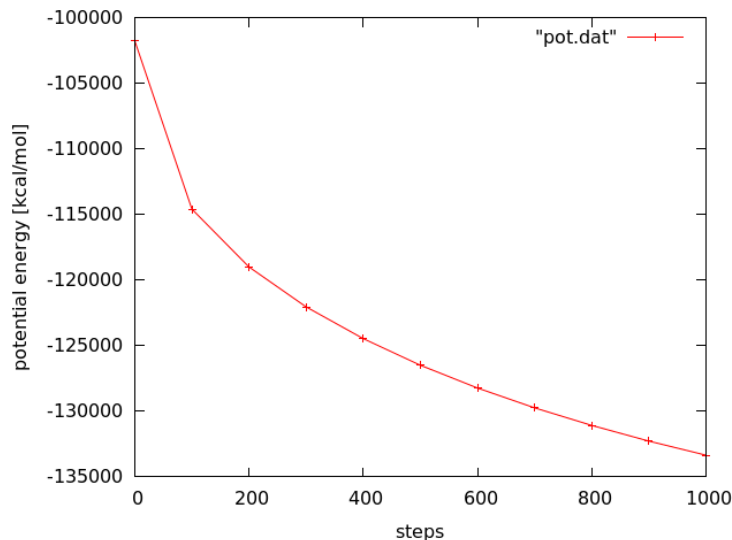
# 計算結果の確認

## ■ ポテンシャルエネルギーの変化を確認します

- GENESIS出力ファイルから各ステップでのエネルギー値を抽出し、gnuplotでプロットします

```
% sh pot.sh  
% gnuplot  
gnuplot> plot "pot.dat" w lp
```

### 実行結果



```
% cat pot.sh  
#!/bin/bash  
  
grep "^INFO" output/run.out | tail -n  
+2 | awk '{print $2, $3}' > pot.dat
```

# 昇温計算 (3\_heating/)

- 構造最適化で、ある程度の安定構造にした後、システムの温度をゆっくり変化させ、本計算で用いる温度まで上昇させる。
- GENESISでは' Annealing' オプションを使ってゆっくり温度を変化させるようなシミュレーションができます

```
./3_heating:  
run.inp          # GENESISの入力ファイル  
run.sh           # FOCUS用のバッチスクリプト  
output           # 計算結果
```

# 昇温計算の入力ファイル抜粋

## [INPUT]

topfile = ../1\_setup/top\_all27\_prot\_lipid.rtf # topology file  
(skip)  
rstfile = ../2\_minimization/run.rst # restart file

## [DYNAMICS]

integrator = LEAP ← # [LEAP,VVER]  
nsteps = 5000 # number of MD steps  
timestep = 0.002 # timestep (ps)  
(skip)  
annealing = YES # simulated annealing  
anneal\_period = 50 # annealing period  
dtemperature = 3 # temperature change at annealing (K)

LEAP: Leapfrog  
VVER: Velocity Verlet

50 stepに1回、3Kずつ  
温度を上げる

## [CONSTRAINTS]

rigid\_bond = YES # constraints all bonds  
# involving hydrogen

## [ENSEMBLE]

ensemble = NVT # [NVE,NVT,NPT]  
tpcontrol = LANGEVIN # thermostat  
temperature = 0.1 # initial temperature (K)

チュートリアルでは時間の関係でこの  
ステップ数にしていますが、  
実際はもっと長い計算を行います

(※) 実際の計算では、計算システムに応じて、標的に温度になるようゆっくり上昇させる

# 昇温計算の実行

## ■ ジョブの投入

```
% cd 3_heating
% sbatch run.sh
% sjstat # 投入ジョブの確認
```

## ■ ジョブ終了後、以下のファイルが出力されているはずです

```
run.dcd # dcd形式のトラジェクトリファイル (binary)
run.out # GENESIS出力ファイル (ascii)
run.rst # GENESISリスタートファイル (binary)
```

### 例: 出力ファイルの[step 5]部分

```
INFO:  STEP      TIME    TOTAL_ENE  POTENTIAL_ENE  KINETIC_ENE      RMSG      BOND      ANGLE
DIHEDRAL  IMPROPER    VDWAALS      ELECT  UREY-BRADLEY      CMAP    RESTRAINT    BOXX    BOXY
BOXZ     VOLUME    TEMPERATURE    PRESSXX    PRESSYY    PRESSZZ    VIRIAL    PRESSURE
(..skip..)
Simulated_Annealing_Leapfrog> Anneal temperature from 210.100 to 213.100
INFO:  3600      7.2000 -121579.0827 -135018.8050  13439.7223    14.8579    126.2918    326.4705
296.5269    18.2507  17470.6626 -153272.5717   43.5707   -75.9996    47.9931    70.8250    83.2579
69.0930  407423.5098    181.9966 -1507.9456 -1629.7733 -1447.1411 -18040.6915 -1528.2867

Simulated_Annealing_Leapfrog> Anneal temperature from 213.100 to 216.100
INFO:  3650      7.3000 -121202.4067 -134760.3492  13557.9425    14.8412    133.4514    322.5176
297.5756    18.2225  17256.3516 -152798.7766   47.1838   -80.8475    43.9725    70.8250    83.2579
69.0930  407423.5098    183.5975 -1553.3990 -1690.9955 -1543.6580 -18521.9527 -1596.0175
```

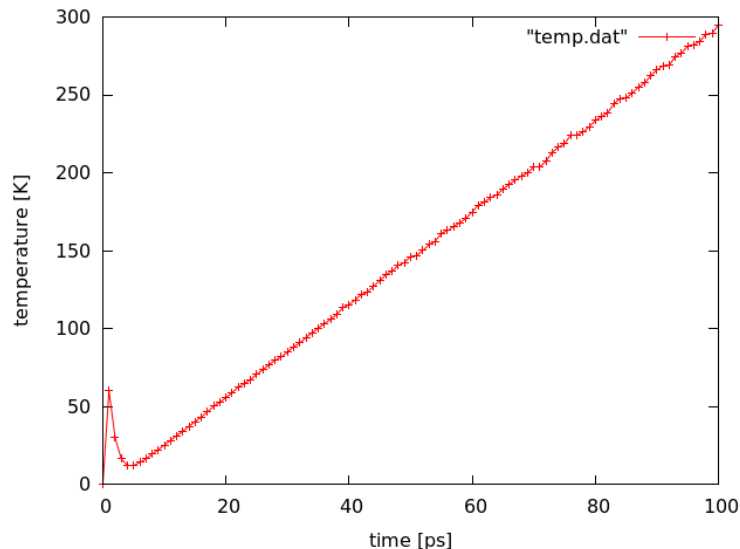
# 計算結果の確認

## ■ 温度の変化を確認します

- GENESIS出力ファイルから各ステップでの温度を抽出し、gnuplotでプロットします

```
% sh temp.sh  
% gnuplot  
gnuplot> plot "temp.dat" w lp
```

### 実行結果



```
$ cat temp.sh  
#!/bin/bash  
  
grep "^INFO" output/run.out | tail -n  
+2 | awk '{print $3, $17}' >temp.dat
```



# 平衡化(4\_equilibration/)

- 本計算と同じ条件の計算を行い、システムを平衡化させる (研究を行う上では、非常に重要なステップ)

```
./4_equilibration:  
run.inp          # GENESISの入力ファイル  
run.sh           # FOCUS用のバッチスクリプト  
output           # 計算結果
```

入力ファイル抜粋

トラジェクトリ書き出しを行い、温度・エネルギーだけでなく、中の分子も平衡に達していることも確認する

[OUTPUT]  
dcdfile = run.dcd # DCD trajectory file

[ENERGY]  
[CONSTRAINTS]  
[RESTRAINS (今回はない)]  
[ENSEMBLE]  
これらのパラメータは基本的に本計算と同じものを使う

[ENSEMBLE]  
ensemble = NPT # [NVE,NVT,NPT]  
tpcontrol = LANGEVIN # thermostat and barostat  
temperature = 300.0 # initial temperature (K)  
pressure = 1.0 # target pressure (atm)

チュートリアルでは時間の関係でこのステップ数にしていますが、実際はもっと長い計算を行います(温度・エネルギーの安定性だけでなく、分子の観測したい性質が安定することも確認します)

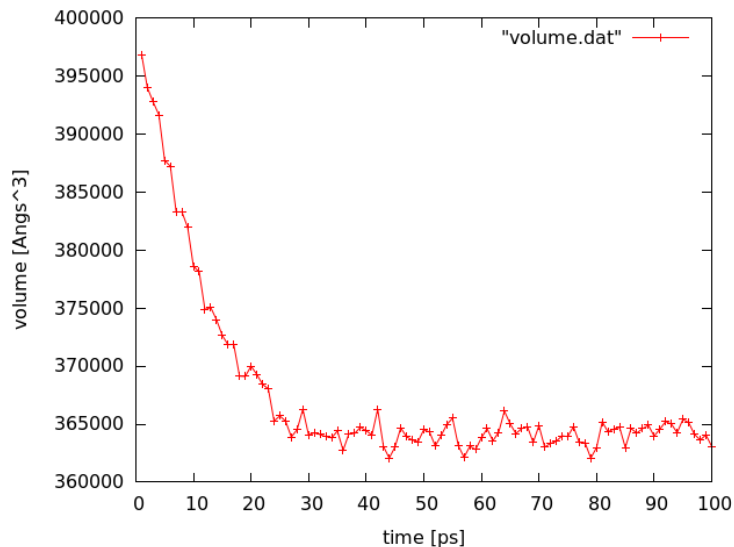
# 計算結果の確認

## ■ 体積の変化を確認します

- GENESIS出力ファイルから各ステップでの体積を抽出し、gnuplotでプロットします

```
$ sh vol.sh  
$ gnuplot  
gnuplot> plot "vol.dat" w lp
```

### 実行結果



```
$ cat vol.sh  
#!/bin/bash  
  
grep "^INFO" output/run.out | tail -n  
+2 | awk '{print $3, $17}' > vol.dat
```

# 本計算(5\_production/)

## ■ 実際にデータを取るための計算

```
./5_production:  
run.inp          # GENESISの入力ファイル  
run.sh           # FOCUS用のバッチスクリプト  
output           # 計算結果
```

入力ファイル抜粋

トラジェクトリ書き出しを行うことで、研究に用いる様々な性質を計算する

[OUTPUT]

dcdfile = run.dcd # DCD trajectory file

チュートリアルでは時間の関係でこのステップ数にしていますが、  
実際はもっと長い計算を行います。

# 解析(6\_analysis/)

## ■ 求めたトラジェクトリから様々な解析を行います

- GENESISは、得られたトジェクトリーを必要に応じて加工し、基本的な解析を行う独自のツールを提供します

```
(例)
./6_analysis:
1_RMSD          # RMSDの計算
2_DIST          # 距離の計算
3_RMSF          # RMSFの計算
4_PCA           # PCA解析
```

- 今回は一例として、「trjana/rmsd\_analysis」ツールを用いて、システムの中からCa原子のみ抜き出しそのRMSD(根二乗平均変位:基準の座標からどれだけずれたかを見る)を計算します

```
./1_RMSD:
run.inp          # GENESISの入力ファイル
run.sh           # FOCUS用のバッチスクリプト
```

# 解析の入力ファイル

- Trajectory、selection、fittingで解析対象と内容を指定します

参照: <http://www.aics.riken.jp/labs/cbrt/tutorial/>

## Trajectory analysis tools

|  |  |
|--|--|
| <b>[INPUT]</b><br>reffile = ../../1_setup/ionize.pdb # PDB file  | trj_natom = 0 # (0:uses reference PDB atom count)  |
| <b>[OUTPUT]</b><br>rmsfile = run.rms # RMSD file   | <b>[SELECTION]</b><br>group1 = an:CA # selection group 1<br><div>Caのみ選択</div>  |
| <b>[TRAJECTORY]</b><br>trjfile1 = ../../5_production/output/run.dcd # trajectory file<br>md_step1 = 500000 # number of MD steps<br>mdout_period1 = 500 # MD output period<br>ana_period1 = 500 # analysis period<br>repeat1 = 1<br>trj_format = DCD # (PDB/DCD)<br>trj_type = COOR # (COOR/COOR+BOX) | <b>[FITTING]</b><br>fitting_method = TR+ROT # method<br>fitting_atom = 1 # atom group<br>mass_weight = YES # mass-weight is applied<br><b>[OPTION]</b><br>check_only = NO # (YES/NO)<br>analysis_atom = 1 # atom group<br><div>基準構造に対する重ね合わせを行う (TR+ROT: 並進と回転で重ね合わせする)</div> <div>YES: 本計算しない</div> |

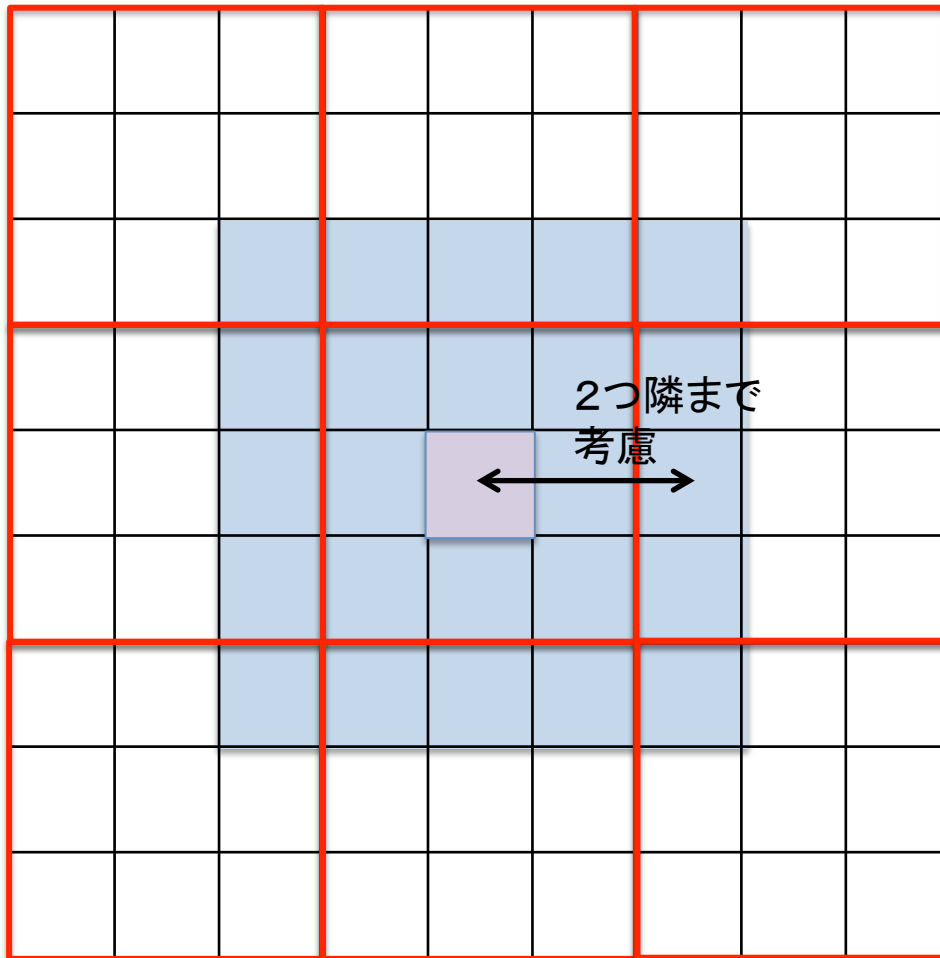
複数trjfileを扱う場合に指定簡略化

- run.shの実行部分を以下の様に変更して実行します

```
${bindir}/rmsd_analysis run.inp > run.out
```

# ドメインとセルの決め方

ドメイン:  、セル: 



- MPIプロセス数に合わせてドメインに分割:

$$\text{ドメイン数} = \text{プロセス数}$$

- セル幅  $d$  は、

$d > (\text{pairlistdist} + \text{buffer})/2$   
を満たしつつ、セル数がドメイン数で割り切れるようにセル幅を決定する。

- 2つ隣のセルまで相互作用を考慮するため、周期境界条件の場合、**最低でも  $5 \times 5 \times 5$  のセルが必要**。

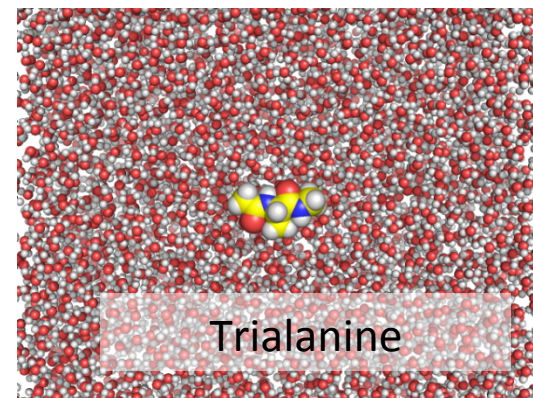
# Tutorial 2: タンパク質のレプリカ交換MD (1)

## ■ 計算内容

水中のTrialanineのレプリカ交換MD計算

参照: <http://www.aics.riken.jp/labs/cbrt/tutorial/>

Replica-exchange molecular dynamics simulations



## ■ 計算データ

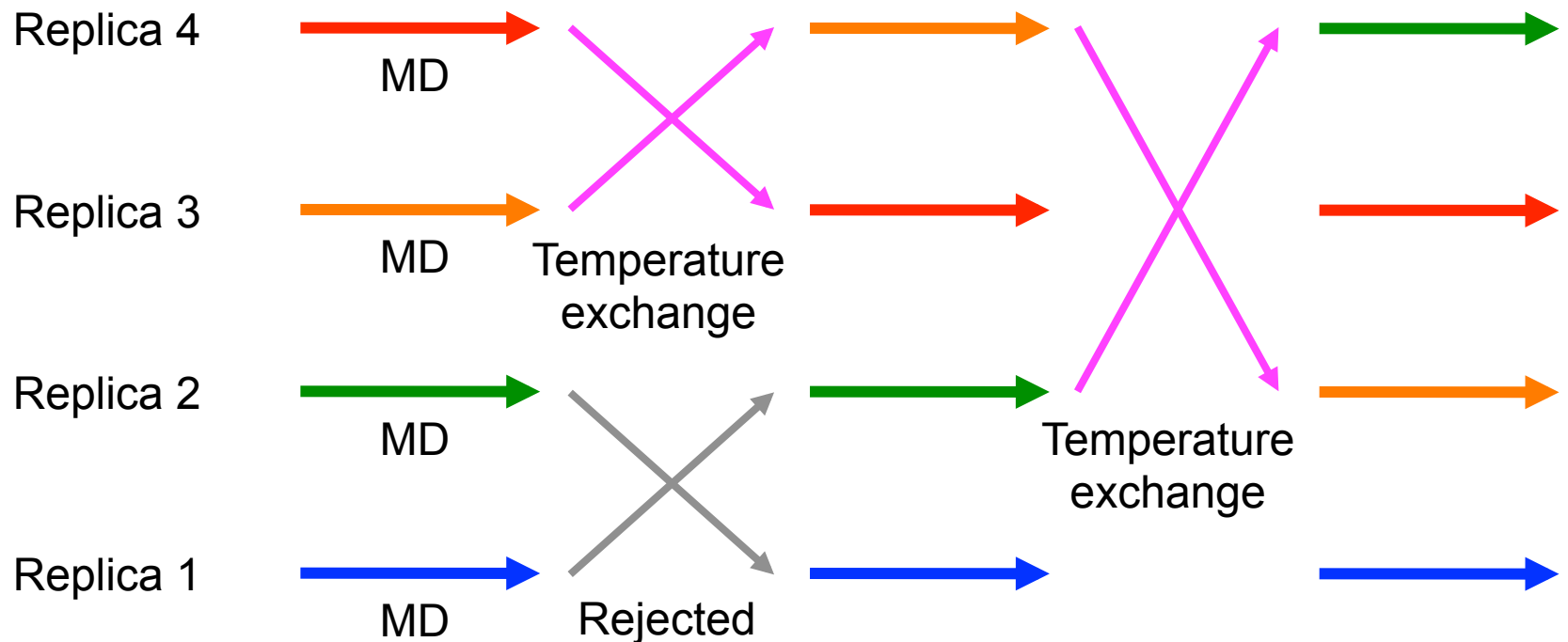
```
% cp -r ../share/Lecture_0908/Tutorial_2 .  
% cd Tutorial_2
```

Tutorial\_2の中身

```
..  
1_setup/           # セットアップ  
2_minimize_pre-equil/ # 最適化と初期平衡化  
3_equilibration/   # 平衡化  
4_production/      # プロダクション  
5_analysis/        # 解析
```

# レプリカ交換MD法

- エネルギー極小状態に捕われず幅広い配位空間を探索  
温度の異なる複数のレプリカ(コピー)を用意し独立にシミュレーションを実行、定期的にレプリカの温度を交換することでサンプリング効率を格段に向上(Y. Sugita and Y. Okamoto. 1999)





# レプリカ交換MD計算の手順

- 実際の研究においてレプリカ数と温度の幅はシステムの原子数、システム自体の性質によって変わります

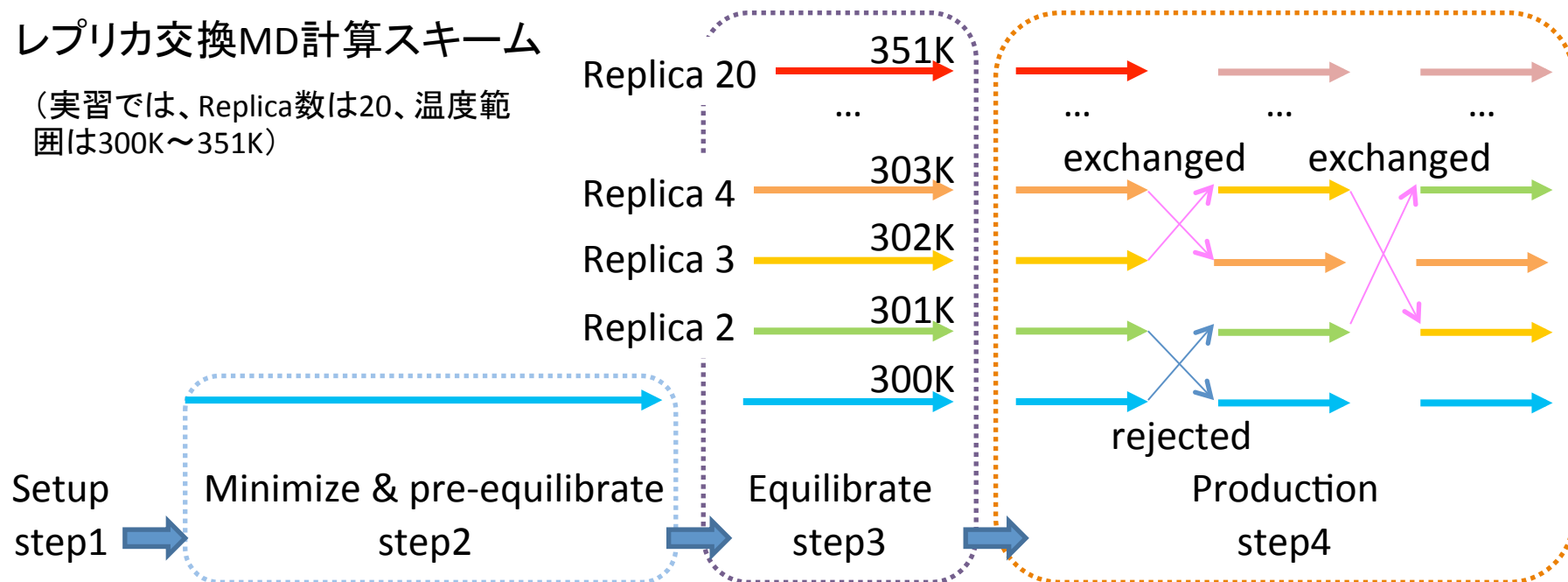
レプリカ数や温度幅の推定に役立つサイト

Temperature generator for REMD-simulations: <http://folding.bmc.uu.se/remd/>

Patriksson and van der Spoel, *Phys. Chem. Chem. Phys.*, **2008**, 10:2073-2077

## レプリカ交換MD計算スキーム

(実習では、Replica数は20、温度範囲は300K～351K)



# 各ステップの説明

- Simulation systemの作成 (1\_setup/)
  - 構造最適化と平衡化 (2\_minimize\_pre-equil/)
    - Step2.1: 構造最適化
    - Step2.2: 昇温計算 (NVT、T=300K) (拘束あり)
    - Step2.3: 平衡化 (NPT、T=300K、P=1atm) (拘束あり)
    - Step2.4: 平衡化 (NPT、T=300K、P=1atm)
  - 各レプリカの平衡化 (3\_equilibrate/)
    - 各レプリカのシステムを該当温度で平衡化 (レプリカの交換なし)
  - 各レプリカの平衡化と本計算 (4\_production/)
  - 解析 (5\_analysis)
- 単独MD、Tutorial\_1と同様

# レプリカ交換MD平衡化計算の入力

## [INPUT]

```
topfile = ../1_setup/top_all36_prot.rtf # topology file
parfile = ../1_setup/par_all36_prot.prm # parameter file
strfile = ../1_setup/toppar_water_ions.str # stream file
psffile = ../1_setup/wbox.psf # protein structure file
pdbfile = ../1_setup/wbox.pdb # PDB file
rstfile = ../2_minimize_pre-equi/step2.4.rst # restart file
```

## [OUTPUT]

```
dcdfile = step3_rep{}.dcd # DCD trajectory file
rstfile = step3_rep{}.rst # restart file
remfile = step3_rep{}.rem # replica exchange ID file
logfile = step3_rep{}.log # log file of each replica
```

## [ENERGY]

```
forcefield = CHARMM # CHARMM force field
electrostatic = PME # Particle Mesh Ewald method
switchdist = 10.0 # switch distance
cutoffdist = 12.0 # cutoff distance
pairlistdist = 13.5 # pair-list distance
vdw_force_switch = YES # force switch option for van der Waals
pme_nspline = 4 # order of B-spline in [PME]
pme_max_spacing = 1.0 # max grid spacing
```

## [DYNAMICS]

```
integrator = LEAP # leapfrog Verlet integrator
nsteps = 1000 # number of MD steps
timestep = 0.002 # timestep (ps)
eneout_period = 100 # energy output period
crdout_period = 100 # coordinates output period
rstout_period = 1000 # restart output period
nupdate_period = 10 # nonbond update period
```

## [CONSTRAINTS]

```
rigid_bond = YES # constraints all bonds involving hydrogen
```

## [ENSEMBLE]

```
ensemble = NVT # NPT ensemble
tpcontrol = LANGEVIN # thermostat and barostat
temperature = 300.00 # initial and target temperature (K)
```

## [BOUNDARY]

```
type = PBC # periodic boundary condition
```

## [REMD]

```
dimension = 1 # number of parameter types
exchange_period = 0 # NO exchange for equilibration
```

```
type1 = temperature # T-REMD
nreplica1 = 20 # number of replicas
parameters1 = 300.00 302.53 305.09 307.65 310.24 \
               312.85 315.47 318.12 320.78 323.46 \
               326.16 328.87 331.61 334.37 337.14 \
               339.94 342.75 345.59 348.44 351.26
```

レプリカ交換MDの指定セクション

レプリカの更新頻度  
(交換をしない)

単独のリスタートファイルからスタート  
レプリカ毎に出力ファイルを生成

(※){}内はレプリカを示す数字で展開される  
(e.g. step3\_rep1.dcd)

# レプリカ交換MD本計算の入力

## [INPUT]

```
topfile = ../1_setup/top_all36_prot.rtf    # topology file
parfile = ../1_setup/par_all36_prot.prm    # parameter file
strfile = ../1_setup/toppar_water_ions.str  # stream file
psffile = ../1_setup/wbox.psf             # protein structure file
pdbfile = ../1_setup/wbox.pdb             # PDB file
rstfile = ../3_equilibrate/step3_rep{}.rst  # restart file
```

## [OUTPUT]

```
dcdfile = step4_rep{}.dcd    # DCD trajectory file
rstfile = step4_rep{}.rst    # restart file
remfile = step4_rep{}.rem    # replica exchange ID file
logfile = step4_rep{}.log    # log file of each replica
```

## [ENERGY]

```
forcefield = CHARMM    # CHARMM force field
electrostatic = PME    # Particle Mesh Ewald method
switchdist = 10.0    # switch distance
cutoffdist = 12.0    # cutoff distance
pairlistdist = 13.5    # pair-list distance
vdw_force_switch = YES    # force switch option for van der Waals
pme_nspline = 4    # order of B-spline in [PME]
pme_max_spacing = 1.0    # max grid spacing
```

## [DYNAMICS]

```
integrator = LEAP    # leapfrog Verlet integrator
nsteps = 2500000    # number of MD steps
timestep = 0.002    # timestep (ps)
eneout_period = 500    # energy output period
crdout_period = 500    # coordinates output period
rstout_period = 5000    # restart output period
nbupdate_period = 10    # nonbond update period
```

## [CONSTRAINTS]

```
rigid_bond = YES    # constraints all bonds involving hydrogen
```

## [ENSEMBLE]

```
ensemble = NVT    # NPT ensemble
tpcontrol = Langevin    # thermostat and barostat
temperature = 300.00    # initial and target temperature (K)
```

## [BOUNDARY]

```
type = PBC    # periodic boundary condition
```

## [REMD]

```
dimension = 1    # number of parameter types
exchange_period = 2500    # NO exchange for equilibration
```

```
type1 = temperature    # T-REMD
nreplica1 = 20    # number of replicas
parameters1 = 300.00 302.53 305.09 307.65 310.24 \
              312.85 315.47 318.12 320.78 323.46 \
              326.16 328.87 331.61 334.37 337.14 \
              339.94 342.75 345.59 348.44 351.26
```

レプリカ交換MDの指定セクション

レプリカの更新頻度

レプリカ毎のリスタートファイルからスタート  
レプリカの数だけ出力ファイルを生成

(※){}内はレプリカを示す数字で展開される  
(e.g. step3\_rep.dcd)

# レプリカ交換MD計算の実行(3\_equilibrate/)

- 今回は、時間の関係上、平衡化計算のみを実施していただきます

```
% cd 3_equilibrate
% sbatch step3.sh
% sjstat # 投入ジョブの確認
```

- バッチスクリプト(step3.sh)の編集

```
#!/bin/bash
#SBATCH -p c006m # キュー名
#SBATCH -N 2 # ノード数
#SBATCH --ntasks-per-node=10 # ノードあたりのプロセス数
#SBATCH -c 1 # スレッド数
#SBATCH -J run
#SBATCH -e run.e%J
#SBATCH -o run.o%J
```

```
module load PrgEnv-intel
module load intel/openmpi165
```

```
export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK}
bindir=/home1/giuo/share/Lecture_0908/GENESIS_binary
mpirun -np 20 ${bindir}/spdyn step3.inp > step3.log
```

squeuesコマンドでキューの空き状況を確認し、適宜変更して下さい

※今の指定は、1プロセス1レプリカの計算で、合計20コアで流れます(2ノード(10コア/ノード))

# レプリカ交換MD計算の出力ファイル

## ■ 各レプリカ毎に出力ファイルが生成されます

- step3.log : 全体の出力ファイル
- step3\_rep[1-20].log : 各レプリカのエネルギーファイル
- step3\_rep[1-20].dcd : 各レプリカのdcd形式のトラジェクトリファイル (binary)
- step3\_rep[1-20].rst : 各レプリカのGENESISリスタートファイル (binary)
- step3\_rep[1-20].rem : レプリカ情報ファイル

各レプリカが何番の温度を計算しているのかを出力  
トラジェクトリの解析に重要！

# レプリカ交換MD出力ファイルの抜粋

## レプリカ交換出力ファイル

## レプリカ情報ファイル

R: 交換失敗(Rejected)  
A: 交換成功(Accepted)

Step数

交換前後の温度

温度の番号

```
REMD> Step:      2500  Dimension:      1  ExchangePattern:      2
Replica      ExchangeTrial      AcceptanceRatio      Before      After
1      1 >      0  N      0 /      0      300.000      300.000
2      2 >      3  A      1 /      1      302.530      305.090
3      3 >      2  A      1 /      1      305.090      302.530
4      4 >      5  A      1 /      1      307.650      310.240
5      5 >      4  A      1 /      1      310.240      307.650
6      6 >      7  A      1 /      1      312.850      315.470
7      7 >      6  A      1 /      1      315.470      312.850
8      8 >      9  R      0 /      1      318.120      318.120
9      9 >      8  R      0 /      1      320.780      320.780
10     10 >     11  R      0 /      1      323.460      323.460
11     11 >     10  R      0 /      1      326.160      326.160
12     12 >     13  A      1 /      1      328.870      331.610
13     13 >     12  A      1 /      1      331.610      328.870
14     14 >     15  A      1 /      1      334.370      337.140
15     15 >     14  A      1 /      1      337.140      334.370
16     16 >     17  A      1 /      1      339.940      342.750
17     17 >     16  A      1 /      1      342.750      339.940
18     18 >     19  R      0 /      1      345.590      345.590
19     19 >     18  R      0 /      1      348.440      348.440
20     20 >      0  N      0 /      0      351.260      351.260
```

```
Parameter      :      300.000      305.090      302.530      310.240      307.650      315.470      312.850      318.120      320.780
323.460      326.160      331.610      328.870      337.140      334.370      342.750      339.940      345.590      348.440      351.260
RepIDtoParmID:      1      3      2      5      4      7      6      8      9
10      11      13      12      15      14      17      16      18      19      20
ParmIDtoRepID:      1      3      2      5      4      7      6      8      9
10      11      13      12      15      14      17      16      18      19      20
```

```
0      1
2500      1
5000      1
7500      1
10000      1
12500      1
15000      2
17500      2
20000      1
22500      1
25000      2
```

# 解析(5\_analysis/)

- remd\_convertを用いて、求めた各レプリカのトラジェクトリから温度毎のトラジェクトリファイルへと変換します

```
% cd 5_analysis
```

## 5\_analysisの中身

```
.:  
1_calc_ratio/      # 交換効率の計算  
2_plot_index/      # 各温度におけるレプリカIDの時間変化  
3_plot_potential/  # 各レプリカのポテンシャルエネルギーの時間変化  
4_sort_dcd/        # 温度毎のトラジェクトリファイルの生成  
5_PCA/             # PCA解析
```



# 解析の入力ファイル

- remd\_convertを用いて、求めた各レプリカのトラジェクトリから温度毎のトラジェクトリファイルへと変換します

## [INPUT]

```
psffile = ../../1_setup/wbox.psf  
reffile = ../../1_setup/wbox.pdb  
dcdfile = ../../4_production/step4_rep{}.dcd  
remfile = ../../4_production/step4_rep{}.rem
```

## [OUTPUT]

```
pdbfile = trialanine.pdb  
trjfile = remd_paramID{}_trialanine.dcd
```

## [SELECTION]

```
group1 = atomno:1-42  
group2 = resno:2 and (an:N or an:CA or an:C or an:O)
```

## [FITTING]

```
fitting_method = TR+ROT  
fitting_atom = 2  
mass_weight = YES
```

## [OPTION]

```
check_only = NO  
convert_type = PARAMETER  
convert_ids = 1 # (empty = all)  
dcd_md_period = 500  
trjout_format = DCD  
trjout_type = COOR+BOX  
trjout_atom = 1  
pbc_correct = NO
```

レプリカの番号からパラメータ(今回は温度)の空間への変換

温度1 (300K) のみの出力  
(デフォルトまたは空欄で  
全てのパラメータを出力)

(※) convert\_ids = 1 3 5と  
すると1 (300K), 3 (304K),  
5 (305K)のファイルを出力