

Simulation with Amber

Naf Guo
2025

Download Amber24

<https://ambermd.org/GetAmber.php>

<https://ambermd.org/GetAmber.php>

从2024年的某一天开始，大善人amber学术免费了！我们需要下载amber24和ambertools24

Option 2: Getting source code in tar format

This path is the traditional way AmberTools has been distributed. AmberTools is distributed in source code format, and must be compiled in order to be used. You will need **C**, **C++** and **Fortran90** compilers.

To download version 24 of the AmberTools distribution, please fill in the following form and click the "Download" button. The file size you should get is about **541 MB**, and the [md5sum](#) is **e4232d79e1eb69f679db75c2e6ac2906**.

Note: Installation instructions are in Chapter 2 of the [Amber 2024 Reference Manual](#). More detailed instructions for many systems may be found at the [OS-specific requirements](#) page.

Name:

Institution:

Download

Ambertools24的安装包

Getting Amber24 for non-commercial use

Read the above license: note that it contains restrictions on usage and redistribution.

By filling out this form, you agree to the license, and confirm that all use will be non-commercial. Note that there is no license fee for non-commercial use.

Name:

Institution:

Accept non-commercial license and download

The downloaded file will be about 173 MB, and have an md5sum of **0e9a01f2f2d5f0c1ebc75d2bdd5cde12**.

Please note: You must also download and untar AmberTools24 using Option 2, as described above. Installation instructions are in Chapter 2 of the [Amber 2024 Reference Manual](#). More detailed instructions for many systems may be found at the [OS-specific requirements](#) page.

Amber24的安装包

Install Amber24

<https://ambermd.org/Installation.php>

<https://ambermd.org/Installation.php>



The Amber Project
Chemistry through a Computational Lens

AmberTools24 Amber24 Manuals Tutorials Force Fields Contacts History

Installing Amber

These are specific instructions and shortcuts for getting Amber installed on popular operating systems. This information supplements what is in Chapter 2 of the *Amber 2024 Reference Manual*.

Amber runs on most platforms: Linux, macOS, and Windows. You can find instructions for your favorite platform on the following dedicated pages as well as answers to Frequently Asked Questions:

- [Arch/Manjaro Linux](#)
- [Centos](#)
- [Debian](#)
- [FAQ](#)
- [Fedora](#)
- [Intel oneAPI](#)
- [MacOS](#)
- [OpenSuse](#)
- [Ubuntu Linux](#)
- [Windows](#)

Notes: You can find summary results for various Linux distros [here](#).

"How's that for maxed out?" Last modified: May 1, 2024

Useful links:

- Amber Home
- Download Amber
- Installation**
- Amber Citations
- GPU Support
- Updates
- Mailing Lists
- For Educators
- File Formats
- Contributors
- Workshops

经过gromacs的洗礼以后，我们对怎么安装软件应该有一定的熟练度了，amber官方提供了不同操作系统下的安装教程。

Install Amber24 on Ubuntu

<https://ambermd.org/InstUbuntu.php>

首先使用apt安装依赖

Ubuntu 18.04 - Ubuntu 20.04

The following command should install all necessary packages to compile Amber on any Ubuntu version in the range above:

```
apt -y update
apt -y install tcsh make \
    gcc gfortran \
    flex bison patch bc wget \
    xorg-dev libz-dev libbz2-dev
```

If you want to install Amber in parallel, you can use OpenMPI through:

```
apt -y install openmpi-bin libopenmpi-dev openssh-client
```

对ubuntu24.04

```
sudo apt install tcsh make gcc gfortran flex bison patch bc wget xorg-dev libz-dev libbz2-dev
```

```
tar xvfj AmberTools24.tar.bz2
```

```
tar xvfj Amber24.tar.bz2
```

Assume this is Linux:

```
cmake $AMBER_PREFIX/amber24_src \
  -DCMAKE_INSTALL_PREFIX=$AMBER_PREFIX/amber24 \
  -DCOMPILER=GNU \
  -DMPI=FALSE -DCUDA=FALSE -DINSTALL_TESTS=TRUE \
  -DDOWNLOAD_MINICONDA=TRUE \
  2>&1 | tee cmake.log
```

```
./run_cmake
```

```
make install -j 4
```

```
source /home/path-to-amber-dir/amber24/amber.sh
```

之后使用amber提供的run_cmake脚本安装amber

Building with cmake

We highly recommend that you refer to Chapter 2 of the *Amber 2024 Reference Manual* for detailed instructions on how to install Amber24.

Since Amber20, the build system has move to cmake. A script called run_cmake is available in the amber24_src/build directory. For most users, the options chosen in this script should be OK.

The installation of Amber is performed in two steps: cmake configuration, then building and install:

```
cd amber24_src/build
# optional: edit the run_cmake script to make any needed changes;
# most users should not need to do this
./run_cmake
# Next, build and install the code:
make install
```

解压缩

之后极其重要的一步：修改build文件夹内的run_cmake脚本，修改DCUDA后为TRUE以编译支持GPU加速的pmemd

Amber Tutorials

<https://ambermd.org/tutorials/>

Amber具有丰富的官方教程

<https://ambermd.org/tutorials/>



几个与模拟强相关的教程

<https://ambermd.org/tutorials/basic/tutorial7/index.php>

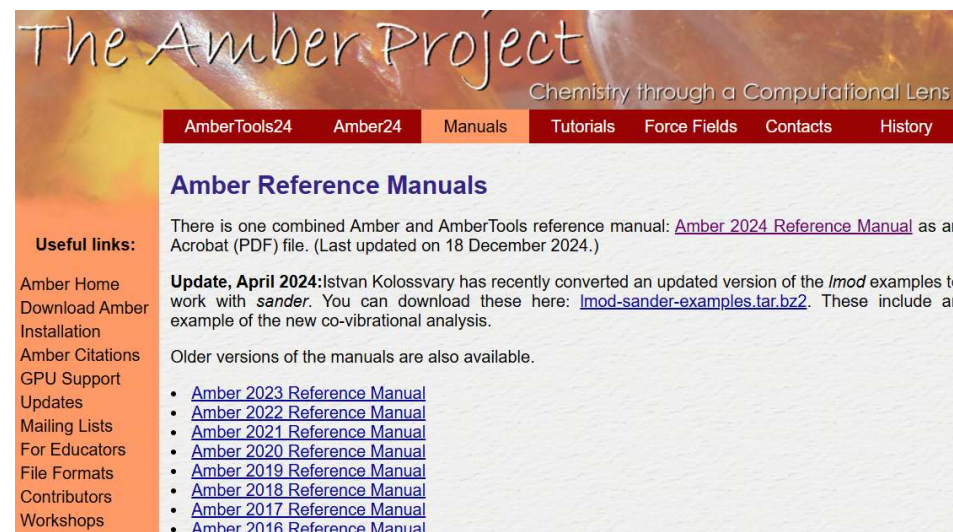
<https://ambermd.org/tutorials/basic/tutorial8/index.php>

<https://ambermd.org/tutorials/basic/tutorial13/index.php>

<https://ambermd.org/tutorials/basic/tutorial14/index.php>

Amber具有详细的文档

<https://ambermd.org/Manuals.php>

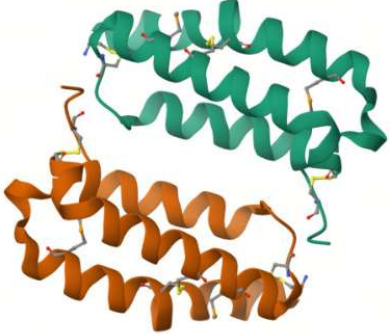


Download PDB

<https://ambermd.org/tutorials/basic/tutorial7/index.php>

我们就按照官方教程，下载2YX8作为测试体系

Biological Assembly 1



Explore in 3D: Structure | Sequence Annotations | Electron Density | Validation Report

Global Symmetry: Cyclic - C2 (Explore in 3D)

Global Stoichiometry: Homo 2-mer - A2

Find Similar Assemblies

Biological assembly 1 generated by PISA (software)

2YX8

Crystal structure of the extracellular domain of human RAMP1

PDB DOI: <https://doi.org/10.2210/pdb2YX8/pdb>

Classification: **PROTEIN TRANSPORT**

Organism(s): *Homo sapiens*

Expression System: *Escherichia coli*

Mutation(s): No

Deposited: 2007-04-24 Released: 2008-04-29

Deposition Author(s): Kusano, S., Kukimoto-Niino, M., Shirouzu, M., Shindo, T., Yokoyama, S., RIKEN Structural Genomics/Proteomics Initiative (RSGI)

Experimental Data Snapshot

Method: X-RAY DIFFRACTION

Resolution: 2.40 Å

R-Value Free: 0.281 (Depositor), 0.276 (DCC)

R-Value Work: 0.221 (Depositor), 0.226 (DCC)

wwPDB Validation

3D Report | Full Report

Metric	Percentile Ranks	Value
Rfree		0.276
Clashscore		8
Ramachandran outliers		0
Sidechain outliers		2.9%
RSRZ outliers		3.8%

Worse

Better

Percentile relative to all X-ray structures

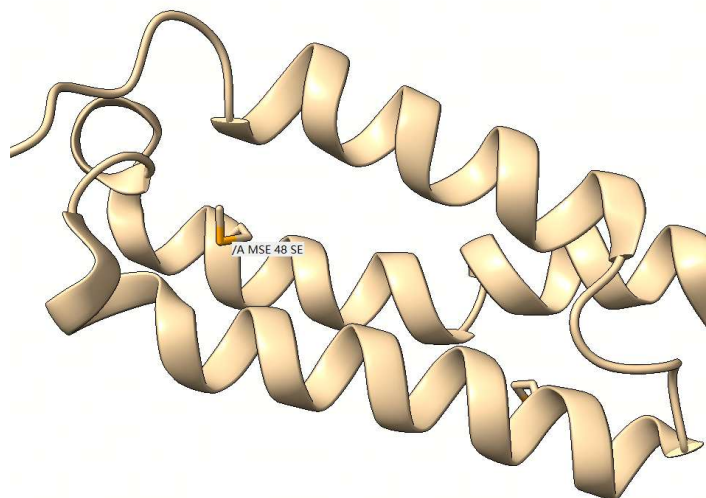
Percentile relative to X-ray structures of similar resolution

This is version 1.2 of the entry. See complete history.

Literature

Download Primary Citation

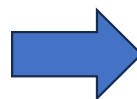
Protein Preparation



此结构有一个有趣的地方，就是它有二个残基是MSE，所谓MSE，就是把甲硫氨酸MET里的硫（S）换成了硒（Se）

需要手动修改回MET

ATOM	167	OD2	ASP	A	47	10.174	7.560	35.935	1.00	54.62	O
HETATM	168	N	MSE	A	48	6.674	9.818	33.156	1.00	45.09	N
HETATM	169	CA	MSE	A	48	6.049	10.959	32.490	1.00	44.83	C
HETATM	170	C	MSE	A	48	4.687	10.621	31.903	1.00	44.85	C
HETATM	171	O	MSE	A	48	3.782	11.457	31.884	1.00	44.46	O
HETATM	172	CB	MSE	A	48	6.968	11.508	31.392	1.00	42.34	C
HETATM	173	CG	MSE	A	48	8.206	12.210	31.932	1.00	37.42	C
HETATM	174	SE	MSE	A	48	7.764	13.636	33.161	1.00	33.09	SE
HETATM	175	CE	MSE	A	48	7.589	15.051	31.873	1.00	30.88	C
ATOM	176	N	GLU	A	49	4.548	9.390	31.429	1.00	45.27	N

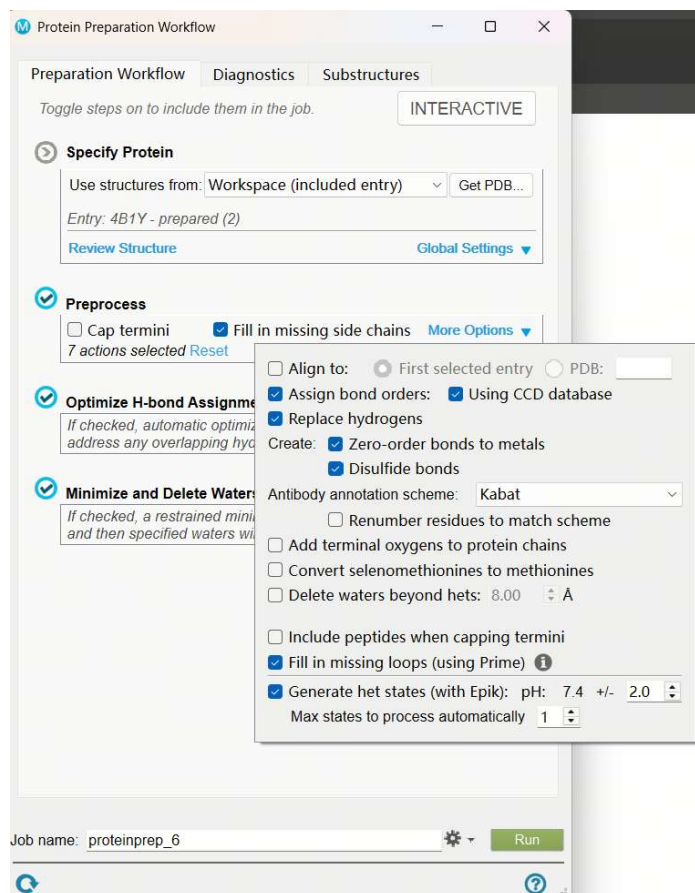


ATOM	167	OD2	ASP	A	47	10.174	7.560	35.935	1.00	54.62	O
ATOM	168	N	MET	A	48	6.674	9.818	33.156	1.00	45.09	N
ATOM	169	CA	MET	A	48	6.049	10.959	32.490	1.00	44.83	C
ATOM	170	C	MET	A	48	4.687	10.621	31.903	1.00	44.85	C
ATOM	171	O	MET	A	48	3.782	11.457	31.884	1.00	44.46	O
ATOM	172	CB	MET	A	48	6.968	11.508	31.392	1.00	42.34	C
ATOM	173	CG	MET	A	48	8.206	12.210	31.932	1.00	37.42	C
ATOM	174	SD	MET	A	48	7.764	13.636	33.161	1.00	33.09	S
ATOM	175	CE	MET	A	48	7.589	15.051	31.873	1.00	30.88	C
ATOM	176	N	GLU	A	49	4.548	9.390	31.429	1.00	45.27	N

<https://ambermd.org/tutorials/basic/tutorial7/index.php>

Protein Preparation

之后就是熟悉的流程，使用maestro或者pdbfixer进行蛋白准备



<https://htmlpreview.github.io/?https://github.com/openmm/pdbfixer/blob/master/Manual.html>

Welcome To PDBFixer!

Select a PDB file to load. It will be analyzed for problems.

☒ Load a local file

PDB File: 未选择任何文件

☐ Download a file from RCSB

PDB Identifier:

```
pdbfixer 4b1y.pdb --output protein.pdb --keep-heterogens=none -  
-add-residues --verbose
```

此处需要记录一下准备好的蛋白带电量，比如-3

Protein Preparation

之后记得处理质子化状态

使用maestro分配的质子化状态

<https://github.com/Sept-naf/gromacs-tutorials/blob/main/maestro2amber.py>

```
python maestro2amber.py maestro_format.pdb amber_readable.pdb
```

使用pdb2pqr分配质子化状态

<https://pdb2pqr.readthedocs.io/en/latest/index.html>

```
pip install pdb2pqr -i https://pypi.tuna.tsinghua.edu.cn/simple
```

对一个已经使用maestro或者pdbfixer处理好了的pdb文件

```
pdb2pqr input.pdb output.pdb --ffout AMBER --with-ph 7.4
```

此时得到的文件就带有amber力场格式的pH7.4下的质子化状态

Build Simulation System

<https://ambermd.org/tutorials/basic/tutorial7/index.php>

到这里蛋白处理好了，可以搭建模拟体系了，amber搭建体系需要用tleap

tleap

使用tleap之后会进入tleap的界面

```
(base) naf@yu-carve-hard-jade:~/Documents/tutorials/amber$ tleap
-I: Adding /home/naf/software/amber24/dat/leap/prep to search path.
-I: Adding /home/naf/software/amber24/dat/leap/lib to search path.
-I: Adding /home/naf/software/amber24/dat/leap/parm to search path.
-I: Adding /home/naf/software/amber24/dat/leap/cmd to search path.

Welcome to LEaP!
(no leaprc in search path)
```

通过左边进入tleap的提示是不是已经可以猜测amber力场参数文件都在哪找了

<https://ambermd.org/tutorials/pengfei/index.php>

Structure File	Library Files	Parameter Files	Topology	Coordinates
Molecules Atom Names Connections Coordinates	Units (Residues) Atom Names Atom Types Charges Connections Coordinates Masses Bonded Params Nonbond Params	Atom Types Masses Bonded Params Nonbond Params	Units (Residues) Atom Names Atom Types Charges Connections Masses Bonded Params Nonbond Params	Coordinates

Amber官方提供了leap的教学，它将PDB与力场参数文件组织起来，变成能进行模拟的拓扑文件和坐标文件

Build Simulation System

<https://ambermd.org/tutorials/basic/tutorial7/index.php>

tleap

在tleap内

```
source leaprc.protein.ff19SB  
source leaprc.water.opc
```

导入蛋白，水和离子的力场参数

```
ramp=loadpdb 2YX8_prepared.pdb
```

导入做好了蛋白质准备的，处理好了质子化状态的蛋白，且将其命名为ramp

```
bond ramp.27.SG ramp.82.SG
```

```
bond ramp.40.SG ramp.72.SG
```

手动连接所有的二硫键

```
bond ramp.57.SG ramp.104.SG
```

```
addIons2 ramp Na+ 0
```

由于前面知道蛋白带负电，这里用Na⁺中和电荷；对正电体系则用Cl⁻

```
solvateOct ramp OPCBOX 10.0
```

给ramp添加水盒子

```
addIonsRand ramp Na+ 19 Cl- 19
```

添加离子，使体系盐浓度为0.15M，计算方法在下一页

```
SaveAmberParm ramp RAMP.prmtop RAMP.inpcrd
```

存储拓扑文件prmtop和坐标inpcrd

Calculating Salt in an Explicit Water System

<https://ambermd.org/tutorials/basic/tutorial8/index.php>

```
Scaling up box by a factor of 1.413369 to meet diagonal cut criterion
```

```
Solute vdw bounding box:          45.400 31.108 30.346
Total bounding box for atom centers: 73.667 73.667 73.667
(box expansion for 'iso' is 55.9%)
Solvent unit box:                18.865 18.478 19.006
The number of boxes:  x= 4  y= 4  z= 4
Volume: 208141.839 Å3 (oct)
Total mass 110188.212 amu, Density 0.879 g/cc
Added 5569 residues.
```

在加水盒子时能看到整个体系的体积，记录下来

2. Convert the volume of the system in Å³ to liters using the following conversion factors.

Be sure to distribute the exponents when performing the calculations.

$$208141.839 \text{ Å}^3 \times \frac{1 \text{ m}^3}{(10^{10} \text{ Å})^3} \times \frac{(10^2 \text{ cm})^3}{1 \text{ m}^3} \times \frac{1 \text{ mL}}{1 \text{ cm}^3} \times \frac{1 \text{ L}}{10^3 \text{ mL}} = 2.08141839 \times 10^{-22} \text{ L}$$

把体积单位从立方埃换算成升

3. Determine how many chloride ions are present in one liter of solution at a concentration of 150 mM.

$$\frac{150 \text{ mmol}}{1 \text{ L}} \times \frac{1 \text{ mol}}{10^3 \text{ mmol}} \times \frac{6.022 \times 10^{23} \text{ Cl}^- \text{ atoms}}{1 \text{ mol}} = 9.03 \times 10^{22} \text{ Cl}^- \text{ atoms/L}$$

计算对这么多升溶液，需要多少离子能达到0.15M

4. Determine how many chloride ions are needed in the system.

Multiply the volume of the box by the concentration of Cl⁻ ions and round to the nearest whole number.

$$2.08141839 \times 10^{-22} \text{ L} \times 9.03 \times 10^{22} \frac{\text{Cl}^- \text{ atoms}}{\text{L}} = 18.8 \text{ Cl}^- \text{ ions} \approx 19 \text{ Cl}^- \text{ ions}$$

计算具体需要的离子数目

$$208141.839 \times 9.033\text{e-}05$$

总结其实就是Volume乘以一个确定的数值，然后四舍五入（或者四舍六入五成双）

More detail

我们已经做好模拟前的准备了，但是，出于学习的考虑，在更进一步前，我们看几条leap命令

```
> charge ramp
Total unperturbed charge: 0.000000
Total perturbed charge: 0.000000
```

charge命令，检查体系电量

```
> desc ramp.27.SG
ATOM
      Normal      Perturbed
Name:      SG      SG
Type:      S      S
Charge:    -0.0984  0.000
Polarization: 0.0000 0.000
Element: S      (not affected by pert)
Atom flags: (decimal 196608 hex 0x30000)
           posfxd n posblt n posdrwn n selected n
           pert n notdisp n touched Y posknwn Y
           internal n needsmin n needsbuild n
Atom position: 14.367393, -8.553670, -4.209354
Atom velocity: 0.000000, 0.000000, 0.000000
Bonded to .R<NCYX 27>.A<CB 7> by a single bond.
Bonded to .R<CYX 82>.A<SG 8> by a single bond.
```

desc命令，检查体系部分的情况

```
> check ramp
Checking 'ramp'....

Warning: Close contact of 1.169 angstroms between nonbonded atoms HH and HE2
----- .R<TYR 66>.A<HH 15> and .R<HIE 97>.A<HE2 13>
Checking parameters for unit 'ramp'.
Checking for bond parameters.
Checking for angle parameters.
check: Warnings: 20977
Unit is OK.
```

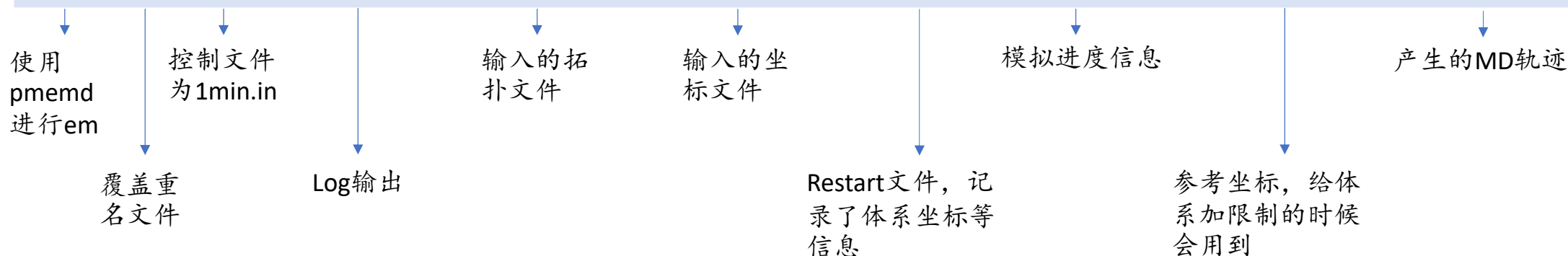
check命令，检查体系力场是否完整

Simulating System: Minimization1

<https://ambermd.org/tutorials/basic/tutorial13/index.php>

体系准备好了，可以退出tleap，正式开始模拟了，首先仍然是能量最小化

```
pmemd -O -i 1min.in -o 1min.out -p RAMP.prmtop -c RAMP.inpcrd -r 1min.rst7 -inf 1min.info -ref RAMP.inpcrd -x mdcrd.1min
```



1min.in

```
minimization of solvent
&cntrl
  imin = 1, maxcyc = 1000,
  ncyc = 20, ntx = 1,
  ntwe = 0, ntwr = 500, ntp = 50,
  ntc = 2, ntf = 2, ntb = 1, ntp = 0,
  cut = 10.0,
  ntr=1, restraintmask = ':1-81',
  restraint_wt = 100.,
  ioutfm=1, ntxo=2,
/
```

- `ntr=1`代表要对体系加限制
- `restraintmask = ':1-81'`代表对残基1-81加限制，注意这是amber重排序过的编号，可以在上一页存出的pdb里确认
- `restraint_wt = 100` 限制大小100 kcal/mol*Ang⁻².

Simulating System: Heat2

<https://ambermd.org/tutorials/basic/tutorial13/index.php>

能量最小化以后是体系加热

```
pmemd.cuda -O -i 2heat.in -o 2heat.out -p RAMP.prmtop -c 1min.rst7 -r 2heat.rst7 -inf 2heat.info -ref 1min.rst7 -x mdcrd.2heat
```

使用pmemd.cuda,
这是GPU加速的版本

上一步得到的坐标,
这一步的输入坐标

```
NSTEP = 648000 TIME(PS) = 648.000 TEMP(K) = 228.63 PRESS = 0.0
Etot = -89860.5401 Ektot = 10302.7109 Eptot = -100163.2511
BOND = 170.7374 ANGLE = 359.9953 DIHED = 349.3371
UB = 0.0000 IMP = 0.0000 CMAP = 82.1253
1-4 NB = 256.8588 1-4 EEL = 2626.6844 VDWAALS = 15575.7389
EELEC = -120003.9513 EHBOND = 0.0000 RESTRAINT = 419.2232
EAMBER (non-restraint) = -100582.4742
NMR restraints: Bond = 0.000 Angle = 0.000 Torsion = 0.000
```

Current Timing Info

Total steps: 1000000 | Completed: 648000 (64.8%) | Remaining: 352000

Average timings for last 213000 steps:

Elapsed(s) = 59.92 Per Step(ms) = 0.28
ns/day = 307.15 seconds/ns = 281.29

Average timings for all steps:

Elapsed(s) = 179.10 Per Step(ms) = 0.28
ns/day = 312.61 seconds/ns = 276.38

Estimated time remaining: 1.6 minutes.

2ns

ntb=1, 恒体积

从0到1000000步将温度从100K升到298K

2heat.in

```
&cntrl
imin = 0, nstlim = 2000000, dt = 0.001,
irest = 0, ntx = 1, ig = -1,
tempi = 100.0, temp0 = 298.0,
ntc = 2, ntf = 2, tol = 0.00001,
ntwx = 10000, ntwe = 0, ntwr = 1000, ntp = 1000,
cut = 8.0, iwrap = 0,
ntt = 3, gamma_ln = 1., ntb = 1, ntp = 0,
nscm = 0,
ntr = 1, restraintmask = ':1-81', restraint_wt = 100.0
nmropt = 1,
ioutfm = 1, ntxo = 2,
/
&wt TYPE = "TEMP0", istep1 = 0, istep2 = 1000000, value1 = 100., value2 = 298., /
&wt TYPE = "END", /
```


Simulating System: NPT平衡3

<https://ambermd.org/tutorials/basic/tutorial13/index.php>

NPT平衡

```
pmemd.cuda -O -i 3md.in -o 3md.out -p RAMP.prmtop -c 2heat.rst7 -r 3md.rst7 -inf 3md.info -ref 2heat.rst7 -x mdcrd.3md
```

3md.in

```
&cntrl  
imin = 0, nstlim = 1000000, dt = 0.001,  
irest = 1, ntx = 5, ig = -1,  
temp0 = 298.0,  
ntc = 2, ntf = 2, tol = 0.00001,  
ntwx = 10000, ntwe = 0, ntwr = 1000, ntpr = 1000,  
cut = 8.0, iwrap = 0,  
ntt = 3, gamma_ln = 1.0, ntb = 2, ntp = 1, barostat = 2,  
nscm = 0,  
ntr = 1, restraintmask = ':1-81', restraint_wt = 100.  
ioutfm = 1, nt xo = 2,  
/  

```

1ns

ntb=2, 恒压
ntp=1, 各向同性压浴
barostat=2, 蒙特卡洛压浴

Simulating System: NPT平衡4

<https://ambermd.org/tutorials/basic/tutorial13/index.php>

```
pmemd.cuda -O -i 4md.in -o 4md.out -p RAMP.prmtop -c 3md.rst7 -r 4md.rst7 -inf 4md.info -ref 3md.rst7 -x mdcrd.4md
```

4md.in

```
&cntrl  
imin = 0, nstlim = 1000000, dt = 0.001,  
irest = 1, ntx = 5, ig = -1,  
temp0 = 298.0,  
ntc = 2, ntf = 2, tol = 0.00001,  
ntwx = 10000, ntwe = 0, ntwr = 1000, ntp = 1000,  
cut = 8.0, iwrap = 0,  
ntt = 3, gamma_ln = 1.0, ntb = 2, ntp = 1,  
nscm = 0, barostat = 2,  
ntr = 1, restraintmask = ':1-81', restraint_wt = 10.  
ioutfm = 1, nt xo = 2,  
/
```

1ns

放松限制到10

Simulating System: Minimization5

<https://ambermd.org/tutorials/basic/tutorial13/index.php>

```
pmemd -O -i 5min.in -o 5min.out -p RAMP.prmtop -c 4md.rst7 -r 5min.rst7 -inf 5min.info -ref 4md.rst7 -x mdcrd.5min
```

5min.in

```
Minimization of everything excluding backbone
&cntrl
imin = 1, maxcyc = 1000,
ncyc = 30, ntx = 1,
ntwe = 0, ntwr = 500, ntp = 50,
ntc = 2, ntf = 2, ntb = 1, ntp = 0,
cut = 8.0,
ntr=1, restraintmask="@CA,N,C", restraint_wt=10.
ioutfm=1, ntxo=2,
/
```

之前约束了残基1-81的所有原子，此时只约束名为CA，N，C的原子，由于我们的体系里除了蛋白就是水和离子，所以这里实际上只约束了蛋白质的主链原子

Simulating System: NPT平衡6

<https://ambermd.org/tutorials/basic/tutorial13/index.php>

```
pmemd.cuda -O -i 6md.in -o 6md.out -p RAMP.prmtop -c 5min.rst7 -r 6md.rst7 -inf 6md.info -ref 5min.rst7 -x mdcrd.6md
```

6md.in

```
&cntrl  
imin = 0, nstlim = 1000000, dt = 0.001,  
irest = 0, ntx = 1, ig = -1,  
tempi = 298.0, temp0 = 298.0,  
ntc = 2, ntf = 2, tol = 0.00001,  
ntwx = 10000, ntwe = 0, ntwr = 1000, ntp = 1000,  
cut = 8.0, iwrap = 0,  
ntt = 3, gamma_ln = 1.0, ntb = 2, ntp = 1,  
nscm = 0, barostat = 2,  
ntr = 1, restraintmask = "@CA,N,C", restraint_wt = 10.  
ioutfm = 1, nt xo = 2,  
/
```

1ns

NPT

Simulating System: NPT平衡7

<https://ambermd.org/tutorials/basic/tutorial13/index.php>

```
pmemd.cuda -O -i 7md.in -o 7md.out -p RAMP.prmtop -c 6md.rst7 -r 7md.rst7 -inf 7md.info -ref 6md.rst7 -x mdcrd.7md
```

7md.in

```
&cntrl  
imin = 0, nstlim = 1000000, dt = 0.001,  
irest = 1, ntx = 5, ig = -1,  
temp0 = 298.0,  
ntc = 2, ntf = 2, tol = 0.00001,  
ntwx = 10000, ntwe = 0, ntwr = 1000, ntpr = 1000,  
cut = 8.0, iwrap = 0,  
ntt = 3, gamma_ln = 1.0, ntb = 2, ntp = 1,  
nscm = 0, barostat = 2,  
ntr = 1, restraintmask = "@CA,N,C", restraint_wt = 1.  
ioutfm = 1, nt xo = 2,  
/
```

1ns

放松约束到1

Simulating System: NPT平衡8

<https://ambermd.org/tutorials/basic/tutorial13/index.php>

```
pmemd.cuda -O -i 8md.in -o 8md.out -p RAMP.prmtop -c 7md.rst7 -r 8md.rst7 -inf 8md.info -ref 7md.rst7 -x mdcrd.8md
```

8md.in

```
&cntrl  
imin = 0, nstlim = 1000000, dt = 0.001,  
irest = 1, ntx = 5, ig = -1,  
temp0 = 298.0,  
ntc = 2, ntf = 2, tol = 0.00001,  
ntwx = 10000, ntwe = 0, ntwr = 1000, ntp = 1000,  
cut = 8.0, iwrap = 0,  
ntt = 3, gamma_ln = 1.0, ntb = 2, ntp = 1,  
nscm = 0, barostat = 2,  
ntr = 1, restraintmask = "@CA,N,C", restraint_wt = 0.1  
ioutfm = 1, ntxo = 2,  
/
```

1ns

放松约束到0.1

Simulating System: NPT平衡9

<https://ambermd.org/tutorials/basic/tutorial13/index.php>

```
pmemd.cuda -O -i 9md.in -o 9md.out -p RAMP.prmtop -c 8md.rst7 -r 9md.rst7 -inf 9md.info -ref 8md.rst7 -x mdcrd.9md
```

9md.in

```
&cntrl  
imin = 0, nstlim = 1000000, dt = 0.001,  
irest = 1, ntx = 5, ig = -1,  
temp0 = 298.0,  
ntc = 2, ntf = 2, tol = 0.00001,  
ntwx = 10000, ntwe = 0, ntwr = 1000, ntp = 1000,  
cut = 8.0, iwrap = 0,  
ntt = 3, gamma_ln = 1.0, ntb = 2, ntp = 1,  
nscm = 1000, barostat = 2,  
ioutfm = 1, ntxo = 2,  
/
```

1ns

无约束平衡

Simulating System: Running MD

<https://ambermd.org/tutorials/basic/tutorial14/index.php>

```
pmemd.cuda -O -i md.in -o md.out -p RAMP.prmtop -c 9md.rst7 -r md.rst7 -inf md.info -ref 9md.rst7 -x mdcrd.md
```

md.in

```
Explicit solvent molecular dynamics constant pressure 50 ns MD
&cntrl
  imin=0, irest=1, ntx=5,
  ntp=50000, ntwx=50000, ntwr=50000, nstlim=25000000,
  dt=0.002, ntt=3, tempi=300,
  temp0=300, gamma_ln=1.0, ig=-1,
  ntp=1, ntc=2, ntf=2, cut=9,
  ntb=2, iwrap=1, ioutfm=1,
/
```


Membrane Protein Simulation?

Charmm-gui

Force Field Options:

AMBER

CHARMM36m

CHARMM36

AMBER

DNA

RNA

Glycan

Lipid

Water

Ligand

FF14SB

OL15

OL3

GLYCAM_06j

Lipid21

TIP3P

GAFF2

☐ Hydrogen mass repartitioning

☐ 12-6-4 ion

Glycolipids and lipoglycans are not supported in current CHARMM-GUI Amber FF implementation.

Input Generation Options:

☒ AMBER

☒ GROMACS

☒ OpenMM

☐ NAMD

☐ GENESIS

☐ Tinker

Equilibration Options:

☐ P21 image transformation (only available for CHARMM)

☐ CHARMM DOMDEC (only available for CHARMM)

☒ Generate grid information for PME FFT automatically

☐ Explicit grid information for PME FFT

X

Y

Z

☒ NPT ensemble

☐ NPAT ensemble

☐ NPgT ensemble

Surface Tension

0

(dyne/cm)

Temperature:

303.15

K

Next Step:

Generate Equilibration and Dynamics Inputs