



Amber22 ユーザーマニュアル



目次

1	Amber22 について	2
2	Amber22 インストール概要	3
3	Amber22 コマンド	5
4	Amber22 実行例	7
4.1	Amber22 実行環境の設定	7
4.2	シリアル(1CPU コア)の実行例	10
4.3	MPI 並列の実行例	11
4.4	GPU 版バイナリの実行例 (GPU 版バイナリ同梱時のみ)	12
5	既知の問題点	13
5.1	2 種類の CPU 版バイナリの速度差について	13
A.1	HPC システムズ お問い合わせ先	16

1 Amber22 について

Amber22 はカリフォルニア大学等のグループが開発する分子動力学シミュレーションプログラムのパッケージです。

Amber22 を使用するには開発元との間でライセンス契約が必要です。契約の内容は以下 URL をご参照下さい。

<http://ambermd.org/LicenseAmber22.pdf>

Amber22 のパッケージは ambermd.org のサイトからダウンロードして入手します。パッケージの入手方法等については契約時に開発元より E メール等で案内が届きますので、そちらをご参照下さい。

Amber の最新版である Amber22 のインストールをする際、AmberTools のパッケージが必要です。そこで、2023 年 4 月にリリースされた AmberTools23 も合わせてインストールをしてあります。AmberTools23 は ambermd.org のサイトの下記 URL でユーザー情報を登録することで無償使用することができます。

<http://ambermd.org/GetAmber.php#ambertools>

本マニュアルは Amber22 のインストール概要とジョブの実行方法等をご案内します。Amber22 の詳細については Amber22 の公式サイト（<http://ambermd.org/index.php>）をご覧ください。

2 Amber22 インストール概要

Amber22 と AmberTools23 は以下ディレクトリにインストールをしています。

GPU 版 (Intel Compiler ビルド版)

パッケージ	ディレクトリ
Amber22	/usr/local/amber22
AmberTools23	/usr/local/amber22/AmberTools

GPU 版

GCC ビルド版 (GPU 対応版)

パッケージ	ディレクトリ
Amber22	/usr/local/amber22-gpu
AmberTools23	/usr/local/amber22-gpu/AmberTools

Intel Compiler ビルド版 (CPU 版)

パッケージ	ディレクトリ
Amber22	/usr/local/amber22-cpu
AmberTools23	/usr/local/amber22-cpu/AmberTools

Amber22 の GPU 版については、当初他のパッケージと同様に Intel Compiler を用いたビルドを試みましたが、エラーが発生し正常にビルドが完了しなかったため、バイナリの作成に至りませんでした。このエラーについては、ビルド時の Compiler Option や CUDA の Version によっては解決しなかったため、なんらかのバグの可能性が高いと判断しています。

これより、GPU 版バイナリについては GCC を用いた Amber22 のセットアップを行っていますが、CPU 版バイナリについては、Intel Compiler を用いたバイナリと比較して、GCC 版バイナリはパフォーマンスが低くなることを確認しております。その為、GPU 版パッケージについては、GCC でビルドしたバイナリのほかに、Intel Compiler でビルドした GPU 版バイナリを別途追加しております。上記の詳細については、4.1 節を参照ください。

Amber22 と AmberTools23 はソースコードで配布されています。Amber22 のインストールの際は以下の開発環境を使用してビルドを行っています。

そのため、本文章にて解説している Amber22 の動作は、以下の開発環境の設定が有効になっている必要があります。

Intel Compiler ビルド版

パッケージ	ディレクトリ
Intel oneAPI Base & HPC Toolkit Version 2023.2.0 + Compiler patch 1	/opt/intel/oneapi/compiler/2021.2.0
Intel MPI Version 2021.10.0	/opt/intel/oneapi/mpi/2021.10.0

GCC ビルド版

パッケージ	ディレクトリ
GCC version 8.5.0	(OS 同梱 GCC)
Intel MPI Version 2021.10.0	/opt/intel/oneapi/mpi/2021.10.0
CUDA 12.2	/usr/local/cuda-12.2

開発陣の作成したソース付属のテストをシリアル、MPI 並列で実行し、公式テストの動作に問題が無い事を確認しました。公式テストの結果についての詳細は、それぞれのパッケージに添付しております仕様書を参考下さい。

Amber22 の環境設定は Environment Module と呼ばれる環境設定ユーティリティ (module コマンド) を用いて行います。Amber22 用のモジュール定義ファイル (Environment Module 用設定ファイル) は、Amber22 と同時にインストールされております。動作確認の際に使用したテスト、およびその実行結果については、/opt/hpcs/app_doc 以下に PDF にまとめてありますのでそちらを参照ください。

Environment Module を用いた Amber22 の設定方法については、4.1 節を参照ください。

3 Amber22 コマンド

Amber22 のコマンドについては Amber22 公式サイト以下の URL にマニュアルがありますので、詳しくはそちらをご覧ください。

<http://ambermd.org/doc12/Amber23.pdf>

以下は主に使用するコマンドと、その説明について Amber22 公式サイトより抜粋したものです。

- **sander**

(part of AmberTools) is the basic energy minimizer and molecular dynamics program. This program relaxes the structure by iteratively moving the atoms down the energy gradient until a sufficiently low average gradient is obtained. The molecular dynamics portion generates configurations of the system by integrating Newtonian equations of motion. MD will sample more configurational space than minimization, and will allow the structure to cross over small potential energy barriers. Configurations may be saved at regular intervals during the simulation for later analysis, and basic free energy calculations using thermodynamic integration may be performed. More elaborate conformational searching and modeling MD studies can also be carried out using the sander module. This allows a variety of constraints to be added to the basic force field, and has been designed especially for the types of calculations involved in NMR, Xray or cryo-EM structure refinement.

- **pmemd**

(part of Amber) is a version of sander that is optimized for speed and for parallel scaling; the pmemd.cuda variant runs on GPUs. The name stands for “Particle Mesh Ewald Molecular Dynamics,” but this code can now also carry out generalized Born simulations. The input and output have only a few changes from sander.

- **nab**

Stands for Nucleic Acid Builder. NAB is really a compiler that provides a convenient molecular programming language loosely based on C.

- **xleap**

A script that calls xaLeap with specific setup command-line arguments.

- **antechamber**

antechamber is the main program to develop force fields for small organic molecules (e. g. , drugs, modified amino acids) using a version of the general Amber force field (GAFF). These can be used directly in LEaP, or can serve as a starting point for further parameter development.

- **cpptraj**

cpptraj is the main trajectory analysis utility (written in C++) for carrying out superpositions, extractions of coordinates, calculation of bond/angle/dihedral values, atomic positional fluctuations, correlation functions, analysis of hydrogen bonds, etc.

- **MMPBSA. py**

MMPBSA. py is a Python script that automates energy analysis of snapshots from a molecular dynamics simulation using ideas generated from continuum solvent models. (There is also an older perl script, called mm_pbsa.pl, that has similar functionality.)

4 Amber22 実行例

4.1 Amber22 実行環境の設定

弊社出荷の RHEL/AlmaLinux 8 系 OS においては、Amber22 の実行環境の設定を、Amber22 を実行したいタイミングで、[Environment Module](#) と呼ばれる環境設定ユーティリティを用いて実施ください。Environment Module ではアプリケーション等のソフトウェア部品を「モジュール」と呼び、`module load`・`module unload` というコマンドによりモジュールの有効化・無効化を実施できます。モジュールが有効化されているとき、そのアプリケーション等は、PATH 等の環境設定が済んで実行可能な状態になっています。

Environment Module で Amber22 を使用可能とするモジュール定義ファイルは次の場所に設置してあります。

CPU 版 (Intel Compiler ビルド版)

```
/home/.common/modulefiles/oneAPI/oneAPI バージョン/Amber22/AmberToolsp5-cpu
```

GPU 版 (GCC ビルド版)

```
/home/.common/modulefiles/oneAPI/oneAPI バージョン/Amber22/AmberToolsp5-gpu
```

GPU 版同梱の CPU 版 (Intel Compiler ビルド版)

```
/home/.common/modulefiles/oneAPI/oneAPI バージョン/Amber22/AmberToolsp5-cpu
```

以下では Environment Module を用いた Amber 実行環境設定の方法を示します。

(1) 使用できるモジュールの一覧表示 : `module avail`

`module avail` コマンドにより、使用可能なモジュールの一覧を表示します。次は Amber22-AmberTools23p5 CPU 版の場合の例です。

```
$ module_ avail
----- /home/.common/modulefiles/oneAPI/2023.2.0 -----
Amber22/Tools23p5-cpu
(後略)
```

赤字で示したように、使用可能なモジュール (ソフトウェア名/バージョン) が表示されます。

(2) モジュールの有効化・無効化 : `module load` ・ `module unload`

モジュールを有効にするコマンドは `module load` です。その後ろにモジュール名を付けて実行します。例えば Amber22-AmberTools23p5 CPU 版のモジュールを有効化する場合、具体的なコマンドは次となります。

```
$ module load Amber22/Tools23p5-cpu
```

逆に、有効化したモジュールを無効化するコマンドは `module unload` です。同様に、後ろにモジュール名を付けて実行します。

```
$ module unload Amber22/Tools23p5-cpu
```

`module unload` は、有効化したモジュールによる環境設定が別の作業に悪影響を及ぼす際などに使用ください。有効化したモジュールによる環境設定が特に悪影響を及ぼしていない場合には、わざわざ無効化する必要はありません。

正常にモジュールを読み込めたかどうかについては、次の `module list` コマンドで確認できます。

(3) 有効になっているモジュールの表示 : `module list`

`module list` により、Environment Module で有効化されているモジュールの一覧を表示します。

```
$ module list
```

例えば Amber22/Tools23p5-cpu のモジュールが有効になっている場合、次のように Currently Loaded Modulefiles: 行より下に表示されます。

```
# module list
Currently Loaded Modulefiles:
 1) tbb/2021.10.0           4) mkl/2023.2.0
 2) compiler-rt/2023.2.1  5) mpi/2021.10.0
 3) compiler/2023.2.1     6) Amber22/Tools23p5-cpu
```

(4) ユーザーシェルログイン時にモジュールを自動的に有効化する方法

ユーザーログイン時に、自動的にモジュールの有効化を行いたい場合、ユーザーのシェル環境設定ファイルに `module load` コマンドを追記してください。

Amber22-AmberTools23p5 CPU 版の場合の具体的な修正例を次に示します。

- ☞ Bash をお使いの場合
ホームディレクトリの `.bashrc` の最終行に以下の追記を行います。

- ☞ Tcsh をお使いの場合
ホームディレクトリの `.cshrc` の最終行に以下の追記を行います。

```
# ----- 任意のコメント -----  
# module load Amber22/Tools23p5-cpu
```

モジュールの有効化が他のアプリケーションに悪影響を及ぼすようなケースも存在いたしますので、ログイン時のシェル環境における自動有効化については、有効時の影響に十分注意した上で行うようお願いいたします。

4.2 シリアル(1CPU コア)の実行例

以降、Amber22 の機能のうち、sander、pmemd のバイナリ（コマンド）について、実行方法を説明します。

Amber22 のマニュアルに記載されている sander 実行時のオプションは以下の通りです。これらのオプションは pmemd でも有効となります。

```
sander [-help] [-O] [-A] -i mdin -o mdout -p prmtop -c inpcrd -r restrt
-ref refc -mtmd mtmd -x mdcrd -y inptraj -v mdvel -frc mdfrc -e mden
-inf mdinfo -radii radii -cpin cpin -cpout cpout -cprestrt cprestrt
-cein cein -ceout ceout -cerestrtr cerestrtr -evbin evbin -suffix suffix
```

具体的な実行例は以下のようになります。

sander の実行例

```
$ sander -O -i mdin -o mdout -p prmtop -c inpcrd -r restrt
```

pmemd の実行例

```
$ pmemd -O -i mdin -o mdout -p prmtop -c inpcrd -r restrt
```

上記の実行例でのオプションの意味は以下となります。

- -O 出力ファイルを全て上書き
- -i sander のオプションを記述した入力ファイル名を指定。実行例では "mdin"
- -o 出力ファイル名を指定。実行例では、"mdout"
- -p パラメーター/トポロジーのインプットファイル名を指定。実行例では "prmtop"
- -c 初期構造のインプットファイル名を指定。実行例では "inpcrd"
- -r 座標、速度等の最終計算結果の出力ファイル名を指定。実行例では "restrt"

sander、pmemd 以外のコマンドの使用方法や、本文章において解説を行っていない sander、pmemd の実行オプションに関する詳細情報は、Amber22 の公式マニュアルを参照ください。

4.3 MPI 並列の実行例

実行ファイルに `.MPI` の拡張子があるものは MPI での並列実行が可能です。`-np` 以降に並列数を指定します。以下は 16CPU コアでの例です。MPI 並列での実行時のオプションはシリアルと同様です。

以下の実行例内の環境変数 `AMBERHOME` は、Amber22 用モジュール `Amber22/AmberTools23p5` を有効化 (module load) することによって設定されます。

sander.MPI の実行例

```
$ mpirun -np 16 sander.MPI -O -i mdin -o mdout -p prmtop -c inpcrd -r restrt
```

pmemd.MPI の実行例

```
$ mpirun -np 16 pmemd.MPI -O -i mdin -o mdout -p prmtop -c inpcrd -r restrt
```

4.4 GPU 版バイナリの実行例（ GPU 版バイナリ同梱時のみ ）

Amber22 には、GPU を計算コアとして使用することができる機能が存在します。この GPU で計算を行うバイナリは CPU 版バイナリとは異なる名前の、別バイナリとなります。また、GPU バイナリは計算できる条件（対応している計算）が存在しています。そのため、GPU 版バイナリで実行できるすべてのインプットが計算できるわけではありません。詳細については公式マニュアルを参照ください。

Amber22 の GPU 版バイナリは、数値精度が異なる 2 種類のバイナリが存在します。以下の記述は Amber22 公式マニュアルからの抜粋となります。

- ・ SPFP – (Default) Use a combination of single precision for calculation and fixed precision for accumulation. This approach is believed to provide the optimum tradeoff between accuracy and performance and hence at the time of release is the default model invoked when using the executable pmemd.cuda.
- ・ DFPF – Use double precision (and double precision equivalent fixed precision) for the entire calculation. This provides for careful regression testing against the CPU code. It makes no additional approximations above and beyond the CPU implementation and would be the model of choice if performance was not a consideration. On v2.0 NVIDIA hardware (e.g. M2090) the performance is approximately half that of the SPFP model while on v3.0 NVIDIA hardware (e.g. K10) the performance is substantially less than the SPFP model.

GPU 版バイナリの具体的ファイル名は、以下の通りです。

pmemd.cuda、pmemd.cuda_SPFP : Serial 版 SPFP 精度 pmemd バイナリ。
(この2つのバイナリは全く同一となります)

pmemd.cuda_DFPF : Serial 版 DFPF 精度 pmemd バイナリ。

pmemd.cuda.MPI、pmemd.cuda_SPFP.MPI : Parallel 版 SPFP 精度 pmemd バイナリ。

pmemd.cuda_DFPF.MPI : Parallel 版 DFPF 精度 pmemd バイナリ。

上記のバイナリは、4.2 節、4.3 節で解説した CPU 版バイナリ実行例のそれぞれのバイナリ名（Serial 版 : pmemd、Parallel 版 : pmemd.MPI）を、該当するバイナリ名に変更することで実行できます。

5 既知の問題点

5.1 2種類のCPU版バイナリの速度差について

以下に GCC ビルド版バイナリと Intel Compiler ビルド版バイナリにおいて、Amber22 公式ページ内で公開されている¹インプット、Cellulose NVE と Nucleosome について CPU で実行した場合のベンチマーク結果を示します。

ベンチマーク環境：

CPU : Xeon Platinum 8480+ (2.0GHz / 56core, SapphireRapids 世代) x 2

Memory : 512GB : 32GB 4800MT/s DDR5 x 16

OS : AlmaLinux 8.7

Amber : Amber22 patch 5 AmberTools23 patch 5

Input : Amber20_Benchmark_Suite² 内の Cellulose_NVE、nucleosome

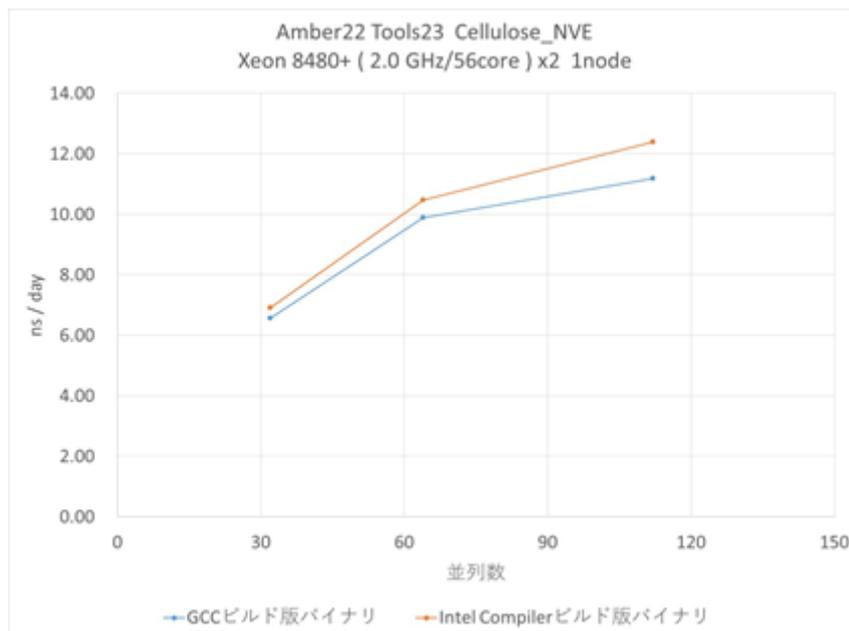


図 1 Cellulose_NVE ベンチマーク結果

¹ <https://ambermd.org/GPUPerformance.php>

² 執筆時点でダウンロードできる Benchmark_Suite の最新は Amber20 です。

表 1 Cellulose_NVE ベンチマーク結果 (ns/day)

並列数	32	64	112
GCC ビルド版バイナリ	6.56	9.89	11.17
Intel Compiler ビルド版バイナリ	6.91	10.47	12.38

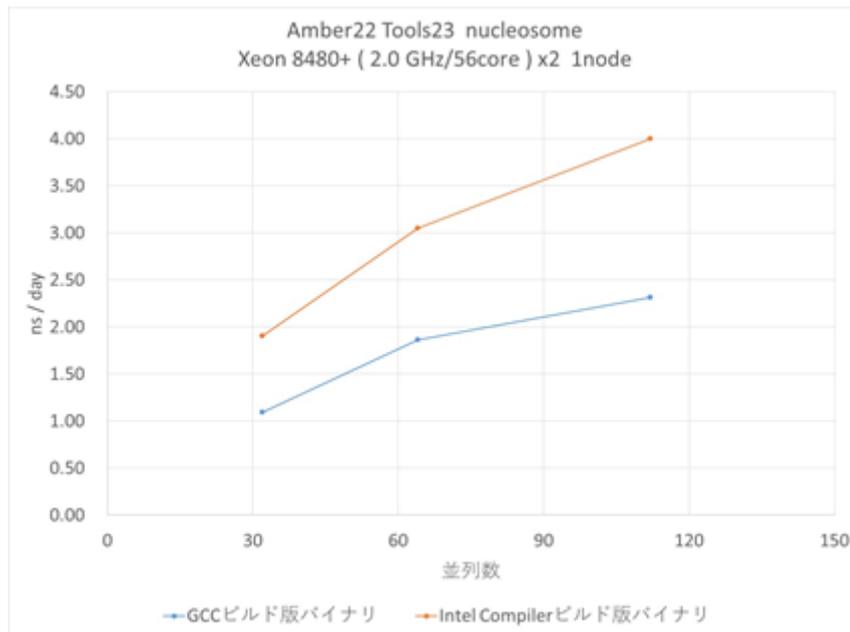


図 2 nucleosome ベンチマーク結果

表 2 nucleosome ベンチマーク結果 (ns/day)

並列数	32	64	112
GCC ビルド版バイナリ	1.09	1.86	2.31
Intel Compiler ビルド版バイナリ	1.90	3.05	4.00

上記の結果のとおり、どちらも Intel Compiler ビルド版バイナリの性能が高く、特に nucleosome においては、1.7 倍近い差がついています。2つのインプットで何故パフォーマンスの差がここまでつくのかは不明ですが、要因の1つとして計算手法の違いが影響しているのではないかと推測しています。

上記のベンチマーク結果より、当初は GPU バイナリについても Intel Compiler を用いたビルドを試みましたが、エラーにより正常にビルドが完了せず、バイナリの作成に至りませんでした。このエラーについてはビルド時の Compiler Option や CUDA の Version 変更によっては解決しなかったため、何らかのバグの可能性が非常に高いと判断しています。

GPU 版バイナリの都合により、GCC ビルド版 Amber22 をセットアップいたしました。GPU を用い

た計算を行う場合の事を考慮し、Intel Compiler を用いてビルドした CPU 版バイナリも同梱いたしました。GPU を用いた計算を行う場合には、必要に応じて Intel Compiler 版を使用してください。

なお、Amber22 の仕様により、GCC ビルド版バイナリと Intel Compiler ビルド版バイナリの設定は共存できません。4.1 節に記載の通り、環境設定を切り替えて使用ください。

付録 A

A.1 HPC システムズ お問い合わせ先



弊社ホームページ http://www.hpc.co.jp/support_index.html

サポート案内やお問い合わせの多い内容など様々な情報を掲載しております。
是非ご活用ください。

HPC システムズ株式会社

〒108-0022 東京都港区海岸 3-9-15 LOOP-X 8 階

HPC 事業部



【営業】 03-5446-5531 【サポート】 03-5446-5532

お電話によるサポート受付は祝日、弊社指定休日を除く月曜日から金曜日の 9:30～17:30
とさせていただきます。



【FAX】 03-5446-5550



【電子メール】 hpcs_support@hpc.co.jp