

# CNSS tutorial(jp)

Takanori Saiki

最終更新日：2023 年 2 月 1 日

## 1 はじめに

CNSS(Cell Network Shape Simulation) は分子通信のシミュレーションをおこなう汎用シミュレータです。シミュレーションにおいて基盤となるプログラムは用意されているので、ユーザはパラメータを変更することで、あるいは独自に拡張をおこなうことで自身の考えたシミュレーションモデルを簡単に実装することができます。

## 2 導入

ここでは、CNSS を導入するための手順を説明します。使用環境は Mac OS か Linux を想定していますが、いくつかのツールを揃えることで Windows でも実行可能になります。

### 2-1 ダウンロード

GitHub の CNSS のページ (<https://github.com/saikiRA1011/CellNetworkShapeSimulation>) からソフトウェアをダウンロードします。ダウンロード先はどこでも良いですが、シミュレーションの実行後に画像ファイルや動画ファイルが作成されるため容量の余裕があるドライブに配置しましょう。

### 2-2 セッティング

CNSS ではいくつかの外部ツールをシミュレーションの作成、実行に用いています。以下はそのツールとバージョンの一例です。

- gcc 11.3.0 (C++11 が使用できるバージョン)
- Python 3.9.12
- pip 22.1.2
- make 3.81

Python で用いるライブラリは以下のコマンドから一括でインストールできます。

---

```
1 pip install -r requirements.txt
```

---

## 3 実行

ここではシミュレーションの実行方法を説明します。CNSS では make を用いてシミュレーションの実行から動画ファイルの生成までをおこなうので、make をコンピュータにインストールしておいてください。

シミュレーションをビルドする際には、`make` を実行してください。実行ファイルとして `SimMain` が生成されます。また、シミュレーションを実行する際には `make run` を実行してください。シミュレーション結果は `result` ディレクトリに、ステップごとにテキストファイルとして出力されます。次に、シミュレーション結果から動画ファイルを生成する際には `make convert` を実行してください。最終生成物の `out.mp4` は `video` ディレクトリに、中間生成物の各ステップごとの画像は `image` ディレクトリに出力されます。最後に、生成した動画を確認したい時は

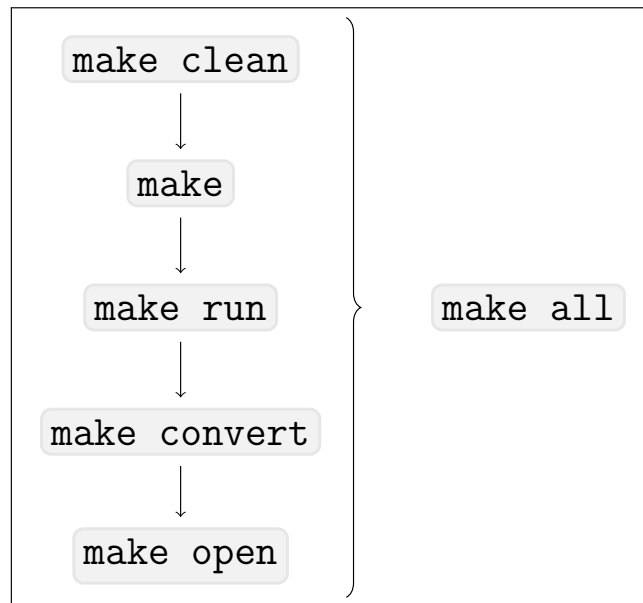


図1 make コマンドによる一連の流れ

`make open` を実行してください。

シミュレーションの設定を変更した際は、そのまま `make` すると修正前のオブジェクトファイルを参照してしまうことがあるので、事前に `make clean` を実行してすべてのオブジェクトファイルを削除してください。

また、これらの一連の処理は `make all` でまとめて実行可能です。図1に一連の流れを示します。

## 4 シミュレーションの自作

ユーザは自身でプログラムを書くことで (あるいはパラメータを変更することで)、シミュレーションを自作することができます。また、ファイルの内容をデフォルトに戻したい場合は、5-2で解説している `make reset` を実行することで初期化できます。ここでは、シミュレーションの自作に必要な説明をユーザが編集するファイルごとにおこなっていきます。

### 4-1 SimulationSettings.hpp

`SimulationSettings.hpp` にはシミュレーションに必要となるパラメータを定義します。例えば、シミュレーションのステップ数や、Cell(バイオナノマシン)の個数、シミュレーションの step あたりの単位時間です。それぞれのパラメータの説明は当該ファイルにコメントとして記述されているので、それらを参考にしてください。

また、必要であればパラメータを増やすことも可能です。自身で必要だと思ったシミュレーションのパラメータ (例えば Cell の種類が複数ある場合はそれぞれの Cell の個数など) をここに定義すると簡単に拡張できます。

ただし、デフォルトで用意されているパラメータを削除することはできません。

## 4-2 UserSimulation.hpp(.cpp)

UserSimulation.hpp(.cpp) にはシミュレーションのモデルや初期条件 (Cell の分布など) を定義することができます。現在定義可能な関数は、Cell の初期化 (initCells)、ステップ実行前処理 (stepPreprocess)、細胞間に働く力学モデル (calcCellCellForce)、ステップ実行後処理 (stepEndProcess) のみです。

このファイルも SimulationSettings.hpp と同様、ユーザが独自に変数・関数を定義することができるので、Cell の種類ごとに力学モデルを定義したい場合はそれぞれのモデルを別の関数に分けて記述することができます。

## 5 付録

### 5-1 シミュレーションの結果のファイル

シミュレーションを実行すると result ディレクトリに cells\_step 番号 のファイル名で複数のテキストファイルが出力されます。これがシミュレーションの結果を出力したものです。

テキストファイルの中身はタブ文字区切りで、以下の形式でシミュレーション結果が記されています。

---

1	ヘッダ行
2	
3	ID typeID X Y Z Vx Vy Vz R N_contact Contact_ID_1 Contact_ID_2 ...

---

それぞれのデータの説明は以下の通りです。

- ID : Cell の識別子。唯一無二のものが割り当てられる
- typeID : Cell の種類の識別子。種類ごとに異なる動作を定義できる
- X : Cell の X 座標の位置
- Y : Cell の Y 座標の位置
- Z : Cell の Z 座標の位置
- Vx : Cell の X 方向の速度
- Vy : Cell の Y 方向の速度
- Vz : Cell の Z 方向の速度
- R : Cell の半径
- N\_contact : 接触している Cell の数
- Contact\_ID : 接触している Cell の ID

### 5-2 make コマンド

CNSS ではシミュレーションの実行から結果の確認に関わる make コマンド以外にも、いくつかのコマンドを用意しています。以下はそれらの解説です。

- `make` : プログラムをビルドする。
- `make run` : シミュレーションを実行する。
- `make convert` : シミュレーション結果を動画ファイルに変換する。
- `make open` : シミュレーション結果の動画ファイルを確認する。
- `make clean` : オブジェクトファイルをすべて削除する。
- `make all` : オブジェクトファイルをすべて削除してから、シミュレーションのビルド、実行、動画の生成、確認を一通りおこなう。
- `make reset` : シミュレーションのユーザ設定ファイルをデフォルトに戻す。

- `make data-cleanup` : シミュレーションの生成物 (テキスト、画像、動画) をすべて削除する。
- `make data-archive` : 現在保存されてあるシミュレーション結果を zip アーカイブする。
- `make archive-restore date=YYYYMMDD_HHmm` : アーカイブファイルの日時を指定することで、アーカイブの内容を復元する。
- `make archive-cleanup` : すべてのアーカイブファイルを削除する。
- `make help` : Makefile に登録されているコマンドの使用方法和説明を表示する。