Mutation Monte Carlo Simulation of the Nucleosome

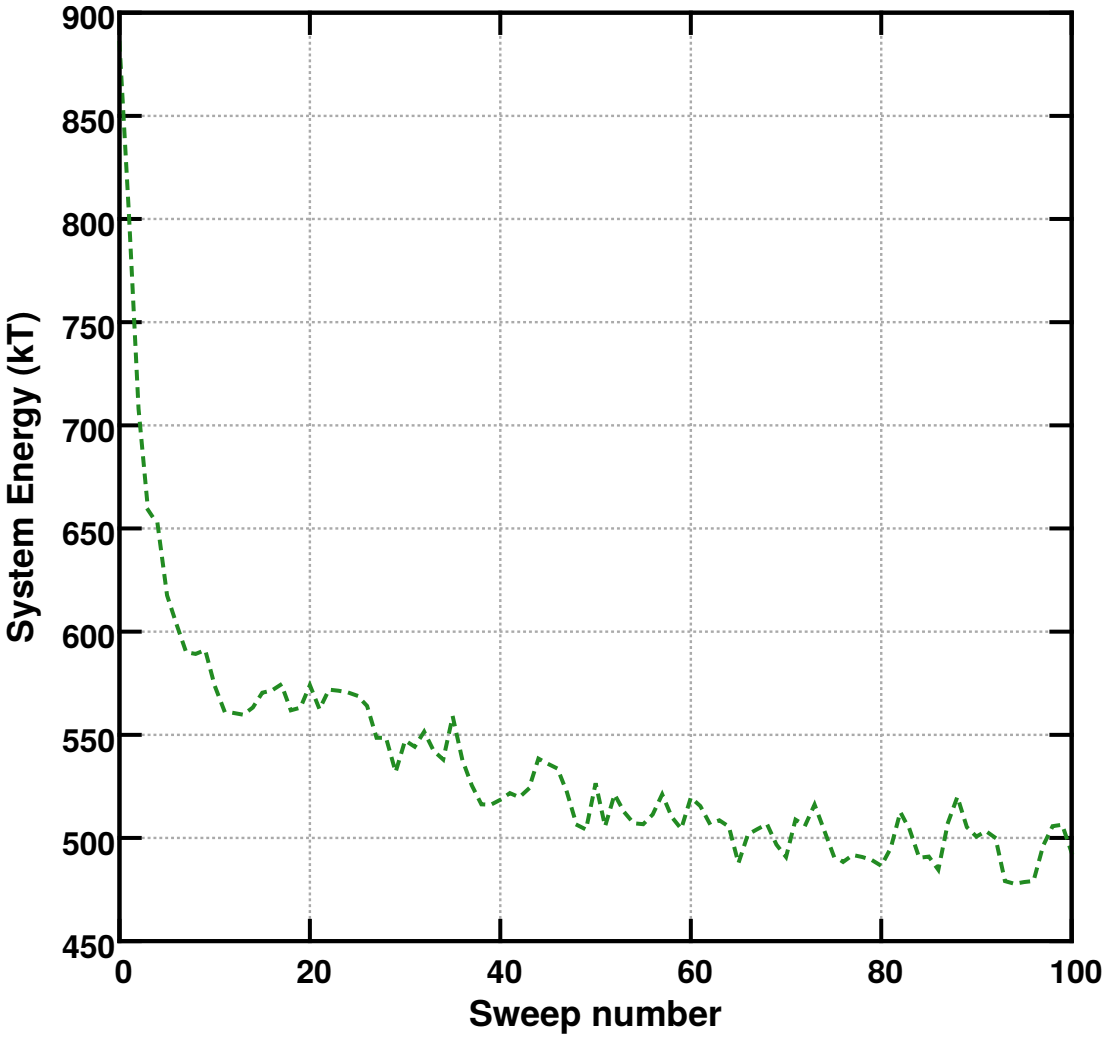
This is a back-to-basics set of C code performing Mutation Monte Carlo (MMC) on the nucleosome simulated as published in Eslami-Mossallam et al. (2016). For in-depth descriptions of the code, see the embedded comments.

File Descriptions:

- `Nucleosome.c`: Main C file containing the Monte Carlo loop.
- `DNA_Moves.c`: Contains the Monte Carlo move functions.
- `DNA_Energy.c`: Contains everything related to energy calculations.
- `DNA_Aux.c`: Miscellaneous auxiliary functions.
- `dnaMC2.h`: Header file.
- `State`: Directory containing example input.
  - `Nucleosome.state`: Valid starting nucleosome configuration.
  - `Ctract.seq`: Sequence file containing a sequence of 147 C's.
- `Parameterization`: Directory containing model parameterization files.
  - `Crystallography`: Parameters for the Rigid Base Pair model derived from crystallographic experiments (Olson et al. 1998).
  - `MoleculateDynamics`: Parameters for the Rigid Base Pair model derived from molecular dynamics simulations (Lankas et al. 2003).
- `Example_Output`: Some example output of the program, as described below.

Example:

As an example, some output is provided, run at beta = 1 (inverse temperature) for 1000 sweeps. The simulation started at a sequence with 147 C nucleotides, and was allowed to mutate. It quickly lowers its energy:



It does so primarily by changing its sequence away from the C-tract and into something with higher affinity for the nucleosome:

```
Iter.  Sequence

01     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
02     CCGCACCCGCCCCCCCCCCCCCGCCCCCCCCCCCGCCCACCCCCCCCCCCCCC
03     CTGCACACGCCCCTCTCCCGCGCACTCGACCCGCCCACCCCCCCCCCCCCCC
04     TTGCAAACGCCCCTCTGCCGTGCACGCAACCTGCCCACCCCCCCCCCCCCC
05     TTGCAAACGCCCCTCTGCCGTGCAAGCACCCCGTCCACACCCCCCCCCCCC
06     TTGCAAACGCCCCACTGCCGTGCAAGCACGCCGTTCACAGCCGCCCCCCCC
07     TTGAAAACGCCCCACTGCCGGGTAAGCACGACGTTCACAGCCGCCCCCCC
08     CGGAAAACGCCCCTACGACGGGTAAGGACGACGTTCCCAGCCGCCCCCCC
09     CGGAAAACGCCCCTACGACGGACAAGGACGACGTTCCCAGCCGCCCCCGC
10     CGGAAAACGCCCCTACGACGGACAAGGACGACGTTCCCAGCTGCCCCCGC
```

How to compile:

Here is an example command that one might use to compile using `gcc`.

```
gcc -O3 -ffast-math -march=native Nucleosome.c DNA_Energy.c DNA_Moves.c DNA_Aux.c \
-o Nucleosome -lm -lgsl -lblas
```

The exact flags will depend on your system. The first three `-O3, -ffast-math, -march-native` are recommended optimization flags, but do come with potential risks. While debugging code, it may be beneficial to remove these flags.

The code requires three libraries to be installed and linked:

- Math library, should be installed on most systems, linked using `-lm`.
- Gnu Scientific Libraries, found at https://www.gnu.org/software/gsl/ and linked with `-lgsl`.
- A BLAS (Basic Linear Algebra Subprograms) implementation. There are various options out there, the right choice may depend on your system. Usually linked with either `-lblas` or `-lopenblas`.

Running the program:

Here is an example command to run the compiled program:

```
./Nucleosome 0 1 10089
```

The command above runs the program (assuming it is called `Nucleosome`) with the following arguments:

- 0 - Starting position along the sequence. (Only relevant if the sequence provided is larger than the size of the system being simulated.)
- 1 - Number of positions along the sequence for which to run MC. (As above.)
- 10089 - Seed for the random number generator.

All these arguments are optional, but if the starting position is defined, the code also expects the number of positions. The example values are the defaults.

Expected file structure:

The code works with two types of data files and expects very simple formatting. Output is in the same format.

1. Sequence files: text files containing sequences formatted as a string of characters from the set {A, T, C, G, a, t, c, g}. Any characters not part of this set (apart from whitespace, which is skipped) are treated as unidentified DNA to which a standard homogeneous set of elasticity parameters is assigned. When outputting, sequences are printed using the set {A, T, C, G, X}, X representing homogeneous DNA.
2. Configuration files: text files containing the positions and orientations of all base pairs. There should be 12 numbers for every base pair in the system, and all numbers should be tab-separated. The expected order of the values is `x, y, z, R00, R01, R02`, etc. for all base pairs consecutively. Output is in the same format.