User Manual for MethylHMM Version 6

Fu, A. Q., Genereux, D. P., Stöger, R., Laird, C. D. and Stephens, M.
Contact: Audrey Qiuyan Fu
audreyqyfu@gmail.com

November 10, 2010

Contents

1 Overview

The program `MethylHMM` is a collection of `R` and `C` code that estimates several properties of DNA methyltransferases from double-stranded DNA methylation patterns. The program implements a Bayesian Markov chain Monte Carlo (MCMC) procedure under the hidden Markov model. Details of the model and MCMC procedure are in [3] and its Supplemental Information.

The directory contains three subdirectories: `Code`, `Data` and `Plots`.

- Under the `Code` subdirectory, the user may modify the `main*.R` files to set up customized MCMC runs, as well as the `analysis*.R` files to analyze the MCMC output and to generate plots.
  - `Code/Source` contains the source code in `R` and `C` for the Bayesian MCMC procedure.
  - `Code/Tools` contains `R` functions useful for analyzing the MCMC output.
- The `Data` subdirectory contains the *FMR1* data, the in vitro mouse Dnmt1 data (taken from [6] and [7]), and sample MCMC output based on which [3] is written.
  - The raw data are under `Data/`.
  - `Data/HMM` contains MCMC output under the HMM for the *FMR1* data.
  - `Data/IEM` contains MCMC output under the independent events model (IEM; [1]) for the *FMR1* data.
- The `Plots` subdirectory contains the plots that are generated using the `analysis*.R` files and appear in [3] and its Supplemental Information.

2 Installation and test run

The code under the `Code/Source` directory implements the Bayesian MCMC procedure under the HMM as described in the manuscript and analyzes the *FMR1* data stored in file `FMR1through22SEPrev.txt` under the `Data` directory.

To do a short MCMC run, which takes about 30 seconds on a 2.2 GHz Mac computer, do the following:

1. Compile the `C` code in `Code/Source` to generate a `.so` file to be used in `R`. You need a `cc` or `gcc` compiler for this. You can download it freely if it is not already installed on your computer. In a terminal window, go to the directory `Code`. For example,

   ```
   $ cd MethylHMM_v6/Code/
   ```

   To compile the `C` code, type both lines below:

   ```
   $ R CMD SHLIB ./Source/StrandAssignmentProbs.c -o
   ./Source/StrandAssignmentProbs.so -lm
   $ R CMD SHLIB ./Source/loglikHMM.c -o ./Source/loglikHMM.so -lm
   ```

2. Run `main_test.R` under `Code`. This file sets up a short MCMC run. The user has three options to run it:

   (a) Run `R`. Change the working directory to where the subdirectory `Code` is located. Open `main_test.R` in a text editor. Copy and paste each line into `R`.

(b) Run `R`. Change the working directory to where the subdirectory `Code` is located. Type in the following command line in `R`:

```
> source ("main_test.R")
```

(c) (Batch mode; preferred for long runs of thousands of iterations and longer) In a terminal window, go to where `main_test.R` is located, and type

```
$ R CMD BATCH main_test.R &
```

This command line runs `main_test.R` directly. Here & is to send the job to background, allowing the user to carry out other tasks in the terminal window while the program runs. This option also allows the user to submit the program onto a computer server and leave it running for hours or days. Note that this and only this option generates a `.Rout` file which is a log file containing all the input and output during the running of the program.

3. To view output, go to directory `Data`.

3 Input data format

The user may format the input data file as `FMR1through22SEPrev.txt` under the `Data` directory. For example, the first four lines from `FMR1through22SEPrev.txt` look like the following:

```
F1,1,1,1,1,1,1,0,1,1,1,1,1,0,1,1,1,1,1,1,1,1,0
F1,1,1,1,1,1,1,0,1,1,1,1,1,0,1,1,1,1,1,1,1,1,0
F1,1,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
F1,1,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
```

The input data file should be formatted as follows:

1. It is comma delimited.

2. It contains $2N$ rows, where N is the number of methylation patterns, and $S+1$ columns, where S is the number of CpG sites.

3. Column 1 is the index of the individual from whom the cells were extracted.

4. The $(2i-1)$-st and $2i$-th rows, where $i = 1, \ldots, N$, are a pair of strands from the same double-stranded methylation pattern.

Table 1: Three types of MCMC output files.

`*.out` (without "strandtype" in file name)	MCMC samples of parameters
`*strandtype*.out`	Probabilities of strand assignment over MCMC iterations.
`*.Rout`	Command lines, acceptance rates and run times

4 Output files

Three types of MCMC output files are summarized in Table 1 and explained in detail below.

1. MCMC samples (`*.out` without "strandtype" in file name):
   - The file is tab delimited.
   - It contains R rows, where R is the number of MCMC iterations stored.
   - It contains $S + 13$ columns, where S is the number of CpG sites.
     - Columns 1-3: associating probability of DNMT1, τ_M, of the DNMT3s on the parent strand, τ_{RP}, and of the DNMT3s on the daughter strand, τ_{RD}.
     - Columns 4-6: dissociating probability of DNMT1, ρ_M, of the DNMT3s on the parent strand, ρ_{RP}, and of the DNMT3s on the daughter strand, ρ_{RD}.
     - Columns 7-$(6 + S)$: site-specific methylation probability m.
     - Columns $(S + 7)$-$(S + 8)$: mean r_m and scaled variance g_m of the beta distribution assumed for m.
     - Column $S + 9$: measurement error rate due to inappropriate bisulfite conversion. This rate is assumed to be constant across CpG sites.
     - Columns $(S+10)$-$(S+11)$: de novo activity rate δ_m and maintenance activity rate μ_m of DNMT1 (on the daughter strand).
     - Columns $(S+12)$-$(S+13)$: de novo activity rate δ_{RD} and maintenance activity rate μ_{RD} of the DNMT3s on the daughter strand.
2. Strand assignment probabilities (`*strandtype*.out`). Each probability is the posterior probability for assigning the top strand in a double-stranded pattern to be the parent strand, given the data and estimates of the parameters (also see Section 4.2 in [2]).
   - The file is tab delimited.
   - It contains R rows and N columns, where R is the number of MCMC iterations stored and N is the number of methylation patterns.

5 Analyses of output files

5.1 Basic analyses

File `analysis_FMR1.R` under the `Code` subdirectory contains the `R` code to produce summary statistics and plots from the MCMC output. To use this file, run `R` and set the working directory in `R` to where this file is located. Copy and paste those command lines in this file into `R`.

The analyses include:

1. Generating summary statistics, such as median, 10- and 90-percentiles, of the MCMC samples.

2. Deriving association and nonassociation lengths, as well as hemi-preference ratios, using the MCMC samples.

3. Deriving mean probabilities of maintenance and de novo methylation events, using the MCMC samples.

4. Producing histograms and scatterplots of the original and derived MCMC samples.

5. Producing trace plots of the MCMC samples for diagnosing the performance of the MCMC run.

5.2 Inference for hemimethylated dyads

See `analysis_FMR1Human1through22_hemis.R` for the `R` code. Also see [4] for rationale and results. For each hemimethylated CpG dyad in the data, the code can be used to infer what event (failure of maintenance, de novo on parent CpG, de novo on daughter CpG, or measurement error), with its corresponding probability, could have given rise to the observed dyad. This inference is possible for the HMM and for the independent events model from [1].

5.3 Inference of top two most likely explanations

See `analysis_FMR1Human1through22_paths.R` for the `R` code. Also see [3, 4] for the algorithm and results. The code infers the top two most likely (with ties) explanations for each double-stranded pattern under the HMM.

6 Customization of MCMC runs

To set up your own MCMC run, you may create a new `main.R` file, using the same format as that in `main_test.R`. The command lines you are most likely to change are explained in Section 6.1. To achieve a better performance from the run, you may also want to modify other options, explained in Section 6.2.

6.1 Basic setups

- Input and output file names. Change the directory in line

  ```
  dataDir = "../Data"
  ```

  Change the input file name in line

  ```
  file.in = paste (dataDir, "/FMR1through22SEPrev.txt", sep="")
  ```

  Change the output file names in lines

  ```
  file.mcmc = paste (dataDir, "FMR1_mcmc.out", sep="")
  file.strandtype = paste (dataDir, "FMR1_strandtype.out", sep="")
  ```

- Nucleotide positions of CpG sites. The positions are specified in `loc`. Its first element is always 0, second element indicates the nucleotide position of the 1st CpG site, and so on.

- Run length. Change the number of iterations in line

  ```
  n.iter = 100
  ```

  You may want to scale up (or down) the value of `step.size` as well. Changing these two arguments together helps control the size of the output files. This is because, after the initial burn-in (number of MCMC iterations for burn-in is specified in `burn.in` as a percentage of the total length), every `step.size`-th MCMC sample is written to the output files. The number of MCMC iterations stored, R, is calculated as

  $$R = \texttt{n.iter} \times (1 - \texttt{burn.in})/\texttt{step.size}.$$

  The total number of iterations should be large, whereas it is good enough to have R around a few hundreds.

- Seed value (`seed.value`). Any positive integer works. This number should be different for different MCMC runs, so that each run can generate a different set of random numbers.

- Disable outputting the MCMC iteration numbers. When running the MCMC for thousands of iterations, the `main.Rout` file will be unnecessarily large. You may disable outputting these numbers onto screen or into the `.Rout` file by setting `PRINT.ITER` to 0.

- Measurement error rate due to failure of bisulfite conversion. This rate is denoted b in [1, 2, 3]. This rate is not estimated by current program. It is estimated by experimental approaches [5].

6.2 Tuning the MCMC run

Since the MCMC algorithm is stochastic and attempts to draw realizations from distributions of interest, its performance can be improved by modifying initial values of the parameters and standard deviations (SDs) used to generate proposals of these parameters. Whereas initial values tell the program where to start sampling, SDs allow the program to move around in large or small steps. The SDs are generally the key tuning parameters.

- Initial values of parameters. There is no need to change initial values for site-specific methylation probability m. All other initial values can be changed. For instance, `tau.curr` specifies initial values for associating probability τ for DNMT1, the DNMT3s on the parent strand and the DNMT3s on the daughter strand. `rho.curr` specifies initial values for dissociating probability ρ in the same order.

- Standard deviations (SDs) that are used to generate proposals of parameters. These values can all be changed based on the acceptance rate of the corresponding parameter. Acceptance rates appear at the end of an MCMC run either in an `R` console or in the `*.Rout` file. An acceptance rate of 20-30% is considered reasonable. If the acceptance rate is much higher, increase the corresponding SD to allow the MCMC to search a larger parameter space. Conversely, if the acceptance rate is much lower, lower the corresponding SD to allow the MCMC to focus searches in a smaller parameter space. You may want to do a few short runs (a few thousands of iterations) to choose the best SD values. `sd.tau` and `sd.rho` specify SDs for τ and ρ, respectively, of DNMT1, the DNMT3s on the parent strand and the DNMT3s on the daughter strand. Note that the acceptance rate may never achieve 20-30% when the data are not informative for a parameter. Experimenting with different values of SD for this parameter can give you a good idea whether this is the case.

6.3 Advanced options

- Fixing measurement error rate c to be constant. The user can achieve this by setting `ESTIMATE.C=FALSE`. The value specified by `c.curr` is used as the fixed value for c throughout the program. Values in `c.ub` and `c.lb` are ignored.

- Prior distributions on ρ. Three options, `uniform`, `logunif` and `jeffreys`, are available for `RHO.PRIOR`.

  - `uniform` assigns a uniform prior to each of the three ρs.
  - `logunif` assigns a uniform prior to ρ_M and a log uniform (uniform on the log scale) prior to either of ρ_{RP} and ρ_{RD}.
  - `jeffreys` assigns a Jeffreys prior (a beta$(1/2, 1/2)$ distribution for proportions) to each of the three ρs.

  When the first two options are used, the user also needs to specify the lower and upper bound in the prior distribution in `rho.lower` and `rho.upper`. Elements in either vector correspond to values for DNMT1, the DNMT3s on the parent strand and the DNMT3s on the daughter strand. When `RHO.PRIOR="jeffreys"`, values in `rho.lower` and `rho.upper` are ignored.

- Prior distributions on τ. Similar to the above. The only exception is that the log uniform prior, if selected, is assigned to all three τs.

- Estimating the maintenance and de novo activity rates of a class of enzymes.

  - Setting `DNMT1.EST=1`, the user may estimate the two rates for DNMT1 (on the daughter strand).
  - Setting `DNMT3.EST=1`, the user may estimate the two rates for the DNMT3s on the daughter strand.
  - If `DNMT1.EST=1` and `DNMT3.EST=1`, the program estimates the maintenance and de novo activity probabilities for both DNMT1 and the DNMT3s on the daughter strand. These two processes then are modelled identically. To distinguish the two processes in estimation, the user should set additional constraints, such as constraining τ_M and τ_{RD} to take on values in different ranges. For example, the user can set

    ```
    TAU.PRIOR = "unif"
    tau.lower = c(0.05, 0, 0)
    tau.upper = c(1, 1, 0.05)
    ```

    The user may want to run the program with `DNMT3.EST=0` first to get an idea what characteristics help distinguish the two processes.

7 Running the program on in vitro data

For a test run, use `main_test_invitro.R` and refer to Sec 2 in this manual for different ways of running it.

In vitro mouse Dnmt1 data taken from [6] and [7] contain only one strand from each double-stranded pattern. This is because the other strand in each pattern is the template strand, which is fully methylated. Not all template strands are fully methylated in [7], but double-stranded data are not provided there. Use `main_test_invitro.R` as the template R file. Note the following settings:

- Reformatting of the data to create double-stranded patterns.

- Setting the last two values in `tau.curr` to 0 and the last two values in `rho.curr` to 1. This sets the activity of the DNMT3s to 0.

- Setting the last two values in `sd.tau` and `sd.rho` to 0. This means that the program does not update τ_{RP} and τ_{RD}. In other words, activities of the DNMT3s are fixed to 0 throughout the estimation procedure.

- Setting `DNMT1.EST=0`. The program then assumes DNMT1 methylates only hemimethylated CpG sites. This is because the CpG sites in the in vitro data are all hemimethylated, or at least assumed so.

References

[1] Fu, AQ, Genereux, DP, Stöger, R, Laird, CD and Stephens, M (2010) Statistical inference of transmission fidelity of DNA methylation patterns over somatic cell divisions in mammals. *The Annals of Applied Statistics* **4** (2) 871-92.

[2] Fu, AQ, Genereux, DP, Stöger, R, Laird, CD and Stephens, M (2010) Supplement to "Statistical inference of transmission fidelity of DNA methylation patterns over somatic cell divisions in mammals." *The Annals of Applied Statistics* DOI: 10.1214/09-AOAS297SUPPA, DOI: 10.1214/09-AOAS297SUPPB.

[3] Fu, AQ, Genereux, DP, Stöger, R, Laird, CD and Stephens, M. In vivo properties of human DNA methyltransferases inferred from methylation patterns.

[4] Fu, AQ, Genereux, DP, Stöger, R, Laird, CD and Stephens, M. Supplementary Information to "In vivo properties of human DNA methyltransferases inferred from methylation patterns".

[5] Genereux, DP, Johnson, WC, Burden, AF, Stöger, R and Laird, CD (2008) Errors in the bisulfite conversion of DNA: modulating inappropriate- and failed-conversion frequencies. *Nucleic Acids Res.* **36** (22), e150.

[6] Goyal, R, Reinhardt, R and Jeltsch, A. (2006). Accuracy of DNA methylation pattern perservation by the Dnmt1 methyltransferase. *Nucleic Acids Res.* **34** (4), 1182-1188.

[7] Vilkaitis, G, Suetake, I, Klimašauskas, S and Tajima, S (2005). Processive methylation of hemimethylated CpG sites by mouse Dnmt1 DNA methyltransferase. *J. Biol. Chem.* **280** (1), 64-72.