User's handbook for BayesMD: Flexible Biological Modelling for Motif Discovery

Man-Hung Eric Tang[1*], Anders Krogh[1], Ole Winther[1,2*†]

1 The Bioinformatics Centre, Department of Molecular Biology
& Biotech Research and Innovation Centre, University of Copenhagen,
Ole Maaløes Vej 5, DK-2100 København Ø, Denmark

2 Informatics and Mathematical Modeling, Building 321,
Technical University of Denmark, DK-2800 Lyngby, Denmark

*These authors contributed equally.

†Corresponding author.

Abstract

BayesMD is a flexible, fully Bayesian model for motif discovery consisting of motif, background and alignment modules. Our modular approach builds-in biological knowledge about the statistical properties of binding sites, background sequences, positional preferences and the number of occurences of sites, providing a flexible and comprehensive framework for the investigation of cis-regulatory elements. BayesMD can be customised to different kind of biological applications e.g microarray, ChIP-chip, ditag, CAGE data analysis by integrating appropriately chosen features and functionalities.
This documents provides all useful information for installation and use of the BayesMD software.

1 How to get BayesMD software?

BayesMD can be downloaded from our homepage http://bayesmd.binf.ku.dk/. We provide 2 software packages:

- BayesMD Matlab source code (for matlab 7.0 and higher)
- BayesMD standalone (for both X86 and AMD64 architectures on Linux platform)

2 Using the Matlab version

2.1 Installation

The BayesMD software package is ready to use. Unpack BayesMDpackage.tar.gz. This will create the work directory *BayesMD/* where all files will be located.

2.2 Running BayesMD using the launcher

Use the script called BayesMDlauncher.m to run the program from the matlab shell. The following syntax shall be used:

```
BayesMDlauncher(infile,option1,option1value,option2,option2value,...)
```

2.2.1 List of arguments

Mandatory argument

Only one argument is mandatory to launch BayesMD: the input sequence file in fasta format *infile*.
This runs the software using the default parameter values. It is possible for advanced users to set their own values by passing pairs of arguments: 'argumentname', 'argumentvalue' both in string format. Below we describe the available options

Optional arguments

- Motif width *width* (default 12)
- Number of sought motifs M (default 1)
- Occurrence model *occurmodel* (default 110505)
- Motif model *motifmodel* (default transfac)
- Background model *bgmodel* (default human_order4_K4)
- Reverse complement *reversecomplement*
  set to 1 if we consider motifs on both strands.
- Masking value for N's *pr_posN*:
  relative probability for positions containing N's
  (between 0 and 1, default 0.1)
- Masking value for lower-case letters *pr_possoft*:
  relative probability for positions containing small letters
  (between 0 and 1, default 0.1)

- Positional prior *positionalmodel* (default flat)
- Custom positional prior file *mapfile*
- Baseline value for user-defined positional prior *baselinecst*
- Number of Samples *Nsamp* (default 20000)
- Number of phase shifts per sequence update *phshift_ratio* (default 1e-3)
- Phase shift interval *dshift* (default 10)
- Number of copies (parallel tempering) *Ntemp* (default 3)
- Constant for the maximum temperature *TmaxCst* (default 4000)
- Maximum temperature *Tmax* (default $1 + \frac{TmaxCst}{\sum Lengths}$)
- Minimum temperature *Tmin* (default 1)
- Temperature update frequency *itemp* (default 3 iterations)
- Number of trials for the sampler *Ntrials* (default 1)
- Number of trials for the motif analyser *Ntrials_marginals* (default 1)
- Fraction of the number of samples for the 'burn in' *burnin* (default 0.3)
- Fraction of occurring counts that are included in the analysis *align_threshold* (default 1)
- Number of alignments to display *n_align* (default 3)
- Minimum number of occurrences to define a motif *th_motif* (default 5)
- Number of the most frequently occurring motifs to consider when constructing marginals *top_occur_max* (default 100)
- Maximum shift considered when testing for shifted motifs *shift_th* (default 3)
- Maximum number of marginal motifs to return *return_max* (default 1000)
- Toggle whether only to return the displayed motifs *return_displayed* (default 1)

2.2.2 Occurrence model

This parameter defines the maximum number of occurrences per sequence and the prior occurrence probability of a motif in each sequence. The model is defined by a vector of relative probabilities for each occurrence. In the following example, the probability of 0 occurrences is twice higher than the one for 1 occurrence.

```
case '105'
    pr_occur = [1 0.5] ;
```

Initialization of this parameter can be done by giving one of the pre-defined values or by self defined occurrence proabilities written as:

```
'[pr_0occ pr_1occ pr_2occ ... ]'
```

Pre-defined values

- 105 : [1 0.5]
- 01 : [0 1]
- 051 : [0.5 1]
- 101 : [1 0.1]
- 11 : [1 1]
- 110505 : [1 1 0.5 0.5]
- 110101 : [1 1 0.1 0.1]
- 1105050101 : [1 1 0.5 0.5 0.1 0.1]

2.2.3 Motif model

This parameter defines which trained mixture model will be used for the analysis. We propose 3 models:

- 'transfac' : mixture of Dirichlet prior distributions trained from the matrices in the TRANSFAC database. The model is trained for 6 mixture components.
- 'jaspar' : mixture of Dirichlet prior distributions trained from the matrices in the JASPAR database. The model is trained for 6 mixture components.
- 'ones' : flat distribution.

User-defined motif models can be trained using the tool that is provided in the BayesMD website. To use this self trained model, provide the following pair of parameter (¡motifModelFilename¿ must have a .mat extension):

```
'motifmodel' '<motifModelFilename>'
```

2.2.4 Background model

This parameter defines which background model will be used for the analysis. We propose several predefined mixture background models for the following organisms: human, mouse, drosophila and yeast. We used promoter sequences from the UCSC genome browser and the SCPD database to train these mixture Dirichlet prior distributions. (see suppl. mat).

- drosophila_order2_K1 : $2nd$ order background, 1 mixture
- drosophila_order4_K1 : $4th$ order background, 1 mixture
- drosophila_order2_K4 : $2nd$ order background, 4 mixtures
- drosophila_order4_K4 : $4nd$ order background, 4 mixtures
- drosophila : default value ($2nd$ order background, 4 mixtures)
- mouse_order2_K1 : $2nd$ order background, 1 mixture
- mouse_order4_K1 : $4nd$ order background, 1 mixture
- mouse_order2_K4 : $2nd$ order background, 4 mixtures
- mouse_order4_K4 : $4nd$ order background, 4 mixtures
- mouse : default value (2^{nd} order background, 4 mixtures)
- human_order2_K1 : $2nd$ order background, 1 mixture
- human_order4_K1 : $4nd$ order background, 1 mixture
- human_order2_K4 : $2nd$ order background, 4 mixtures
- human_order4_K4 : $4nd$ order background, 4 mixtures
- human : default value ($2nd$ order background, 4 mixtures)
- yeast_order2_K1 : $2nd$ order background, 1 mixture
- yeast_order4_K1 : $4nd$ order background, 1 mixture
- yeast_order2_K4 : $2nd$ order background, 4 mixtures
- yeast_order4_K4 : $4nd$ order background, 4 mixtures
- yeast : default value ($2nd$ order background, 4 mixtures)
- ones_order1 : $1st$ order background flat prior
- ones_order2 : $2nd$ order background flat prior

User-defined background models can be trained using the tool that is provided in the BayesMD website. To use this self trained model, provide the following pair of parameter (¡bgModelFilename¿ must have a .mat extension):

```
'motifmodel' '<bgModelFilename>'
```

2.2.5 Positional priors

This parameter defines positional preferences of motif occurrence, also called by the term soft-masking. we distinguish two soft-masking schemes:

- low-complexity: We set a constant value (chosen between 0 and 1) to sequence positions that have been marked as low complexity regions (N's or lower case letters).
  pr_posN and *pr_possoft* are set to 0.1 (10 times less probable) as default.

- user-defined: This option enables users to map input sequences to position specific prior information abour the sequences like conservation or binding preferences. Typically, the sequence map should be loaded with a tab-formated text file, 1 line per sequence, 1 number per nucleotide, using the following format:

  ```
  sequence label<tab>value1<space>value2<space>value3...
  chr4:90958946-90960704  0.1630 0.1587 0.1539 0.1487 ...
  chr8:143893708-143894655        0.2019 0.2016 0.2013 ...
  ```

  In order to avoid missing values or zeros, we add a baseline value to the score which is then normalized. This value can be changed by the user (*baselinecst*).

2.3 Other tools

2.3.1 Tools for training the mixture models

The matlab scripts that are used to train the background mixture models are provided on the BayesMD homepage.

Background models

This is done by running Dirichlet_mixture_EM_estimation_run_bg_higher_order.m
We provide training sets (promoter sequences) for 4 organisms: human, mouse, drosophila and yeast.
Below, we describe how to run the script.

1. Run Dirichlet_mixture_EM_estimation_run_bg_higher_order.m
   The command line is:

   ```
   Dirichlet_mixture_EM_estimation_run_bg_higher_order(option1,option1value,...)
   ```

   The training parameters are:

   - the number of mixture components K,
   - background order *order*
   - the sequence set *trainingfile*

If no options are provided, the script runs with its default parameters, K=4, order=2 and trainingfile='upstream1000'.

The available training sets are:
human promoters ('upstream1000'), drosophila promoters ('drosophila-upstream1000'), mouse promoters ('mouse-upstream1000'), yeast promoters ('yeast-upstream1000'), and NestedMICA training set ('nestedMICA').

2. Copy the trained model to the work directory

3. Edit BayesMDlauncher.m and add a new entry to the switch

Users can provide their own background sequences by setting *trainingfile* to 'usr'. The fasta file that contains the training sequences should be named 'user_trainingfile.fa' .

Motif models

This is done by running Dirichlet_mixture_EM_estimation_run.m
We provide training sets that are derived from TRANSFAC and JASPAR matrices.
Below, we describe how to run the script.

1. Run Dirichlet_mixture_EM_estimation_run.m
   The command line is:

   ```
   Dirichlet_mixture_EM_estimation_run(option1,option1value,option2,option2value)
   ```

   The training parameters are:

   - the number of mixture components K
   - training set *dataset* (to be chosen from 'jaspar_all', 'transfac_all, 'usr').

   If no options are provided, the script runs with its default parameters, K=6 and dataset='jaspar_all' , using counts from JASPAR matrices for training.

2. Copy the trained model to the work directory

3. Edit BayesMDlauncher.m and add a new entry to the switch

In order to provide user-specific matrix counts, the option *dataset* must be set to 'usr'. Matrix counts are then provided by a tabbed text-file,named 'training_usr.mat' with 4 columns, each corresponding to A,C,G,T.
Matrices are vertically concatenated. See example below.

```
1 2 2 0
2 1 2 0
3 0 1 1
0 5 0 0
5 0 0 0
0 0 4 1
0 1 4 0
0 0 0 5
```

Figure 1: Example of text formatting for the training file 'training_usr.mat'

2.4 BayesMDrunMarginals : the command line launcher of the motif analyser

The BayesMD application generates a matlab workspace file with the *.mat extension which saves all the sampling data of each run. In order to perform motif analysis with refined parameters and try different threshold values without running the sampling step again, we propose the BayesMDrunMarginals.m program. Parameters that can be used are the following:

- Fraction of the number of samples for the 'burn in' *burnin*
- Fraction of occurring counts that are included in the analysis *align_threshold*
- Number of alignments to display *n_align*
- Minimum number of occurrences to define a motif *th_motif*
- Number of the most frequently occurring motifs to consider when constructing marginals *top_occur_max*
- Maximum shift considered when testing for shifted motifs *shift_th*
- Maximum number of marginal motifs to return *return_max* (default 1000)
- Toggle whether only to return the displayed motifs *return_displayed*

The following syntax shall be used to run the program:

```
BayesMDrunMarginals(workspacefile,option1,option1value,option2,option2value,...)
```

where each parameter is given as a string.

3 Using the standalone version

We propose a standalone binary for non-Matlab users. The available binaries are pre-compiled for both Intel X86 and AMD64 on Linux platform and requires the Matlab Component Runtime (MCR) environment to be installed.

3.1 Installation procedure

1. Download both the binary and the MCR from the BayesMD website.
2. Create a directory, e.g BayesMD and unpack both the binaries into it.

3.2 Using the standalone program

In order to launch the program, execute the shell script run_BayesMDlauncher.sh . It configures the environment variables for the MCR and runs the binary. For example, if the MCR is installed in the folder /home/user/BayesMD/V76, the command-line is the following:

```
>sh ./run_BayesMDlauncher.sh /home/user/BayesMD/V76 infile option1 option1value ...
```

The minimal command-line requires the input file name only. For a detailed description of the options to include to the command-line, please refer to the matlab section of the manual.

3.3 Standalone for BayesMDrunMarginals

A standalone version of the script is available on the webpage. Please refer to the previous section for how to use the motif analyzer.

4 Using the BayesMD webservice

BayesMD webservice offers the possibility to submit jobs to our webserver and retrieve motif prediction results by email and/or html page. The submission page is available at http://servers.binf.ku.dk/bayesmd/.

4.1 Example of a query

4.1.1 Step 1: The query page

1. Provide the dataset of interest by pasting into the window or by providing the location of a fasta file in the box.
2. Change the parameter values in accordance to the dataset
3. Provide an email adress where a notification will be sent once the job is complete.
4. Send the job.

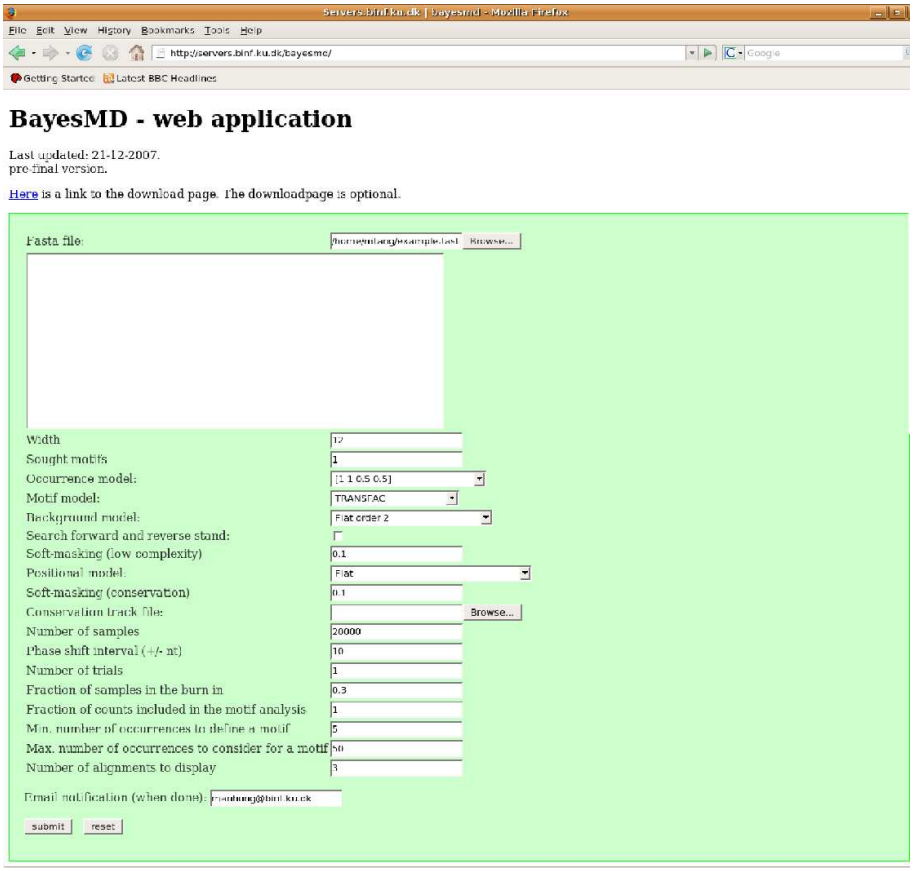


Figure 2: Example of a query page to the BayesMD webserver.

4.1.2 Step 2: Job status

This page shows the status of the query while it is running on the server. It shows the job number (e.g 286-1720276199) that has been assigned and can be closed at any time. To come back to this page paste the url of the page (e.g http://servers.binf.ku.dk/bayesmd/286-1720276199/index.php).

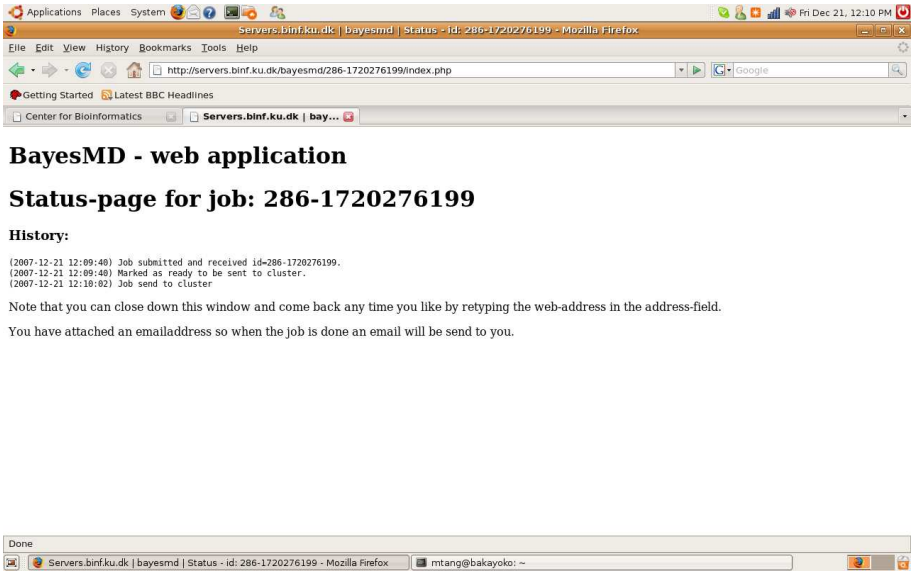


Figure 3: Example of job status page: the result is returned to the window and/or to the email address that has been provided.

4.1.3 Step 3: Result page

This page shows a summary of the motif predictions. It consists of 2 parts:

- A detailed analysis of marginal and inclusive motifs that have been found (see next section)
- Sequence logos for the best marginal and inclusive motifs.

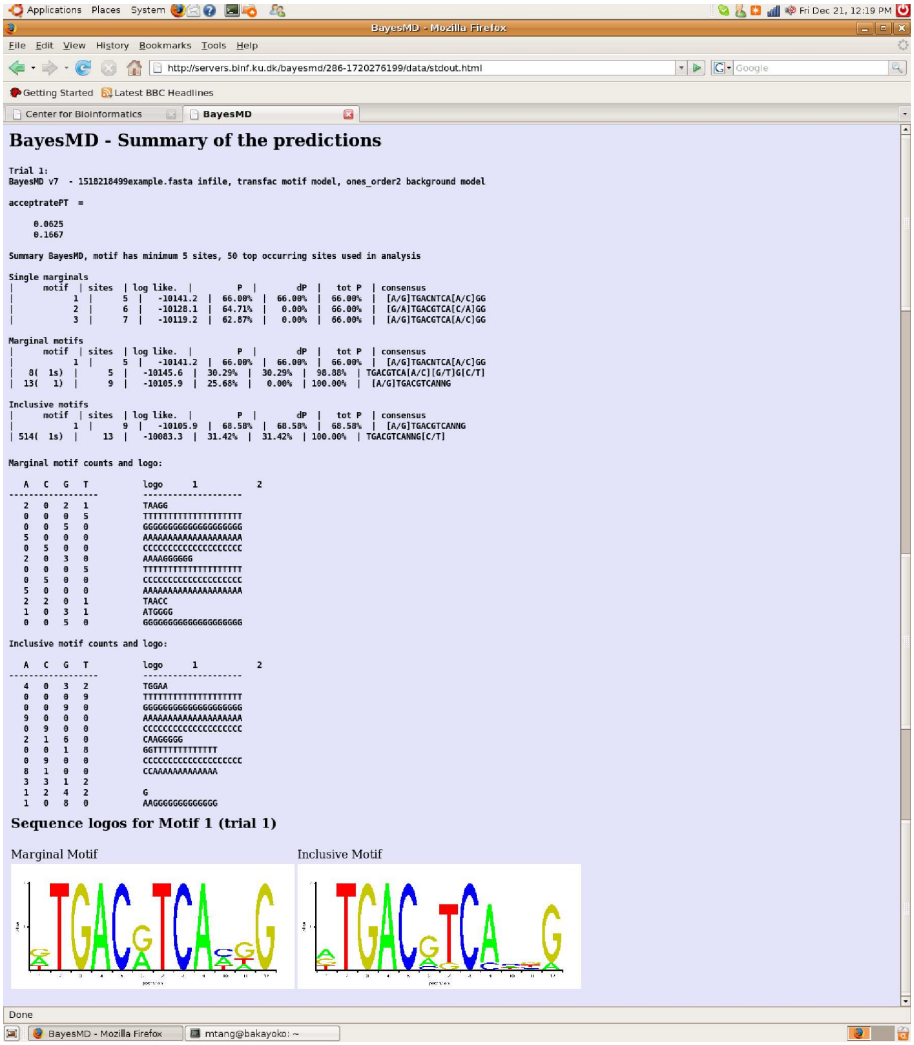


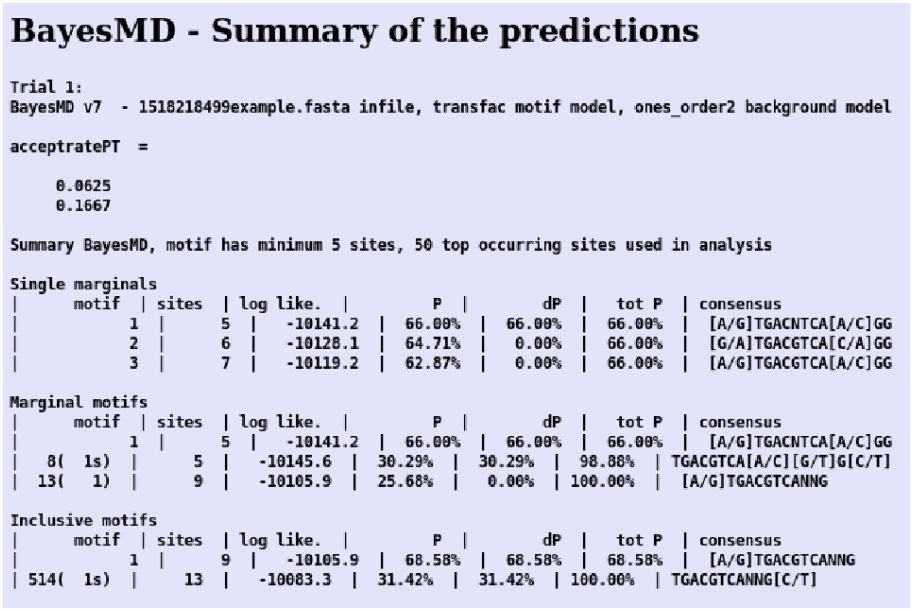
Figure 4: Example of result page.

5 Description of the BayesMD output

This result page is the output of the BayesMD command-line application. Here is a description of each field.

- Trial : current trial number.
- acceptratePT : parallel tempering acceptance rate. This value should not be too close to 0 or 1.
- motif : label for the best occurring motifs of each type (single marginal, marginal, inclusive), the number of hits depends on the parameter *n_align*. If a motif is a shifted version it is labelled with the original motif number in brackets with a 's'.
- sites: number of sites.
- log like. : log-likelihood value of the motif.
- P : probability of the motif
- dP: contribution of the motif to the total value.
- tot : total probability value of the motif.
- consensus: consensus of the motif

The second part of the test output contains the count matrix of the best marginal and inclusive motif as well as a sequence logo in text format.



BayesMD - Summary of the predictions

```
Trial 1:
BayesMD v7  - 1518218499example.fasta infile, transfac motif model, ones_order2 background model

acceptratePT  =

     0.0625
     0.1667

Summary BayesMD, motif has minimum 5 sites, 50 top occurring sites used in analysis

Single marginals
|      motif  | sites  | log like.  |         P  |        dP  |   tot P  | consensus
|            1  |      5  |   -10141.2  |  66.00%  |  66.00%  |  66.00%  |  [A/G]TGACNTCA[A/C]GG
|            2  |      6  |   -10128.1  |  64.71%  |   0.00%  |  66.00%  |  [G/A]TGACGTCA[C/A]GG
|            3  |      7  |   -10119.2  |  62.87%  |   0.00%  |  66.00%  |  [A/G]TGACGTCA[A/C]GG

Marginal motifs
|      motif  | sites  | log like.  |         P  |        dP  |   tot P  | consensus
|            1  |      5  |   -10141.2  |  66.00%  |  66.00%  |  66.00%  |  [A/G]TGACNTCA[A/C]GG
|   8(  1s)  |      5  |   -10145.6  |  30.29%  |  30.29%  |  98.88%  | TGACGTCA[A/C][G/T]G[C/T]
|  13(   1)  |      9  |   -10105.9  |  25.68%  |   0.00%  | 100.00%  |  [A/G]TGACGTCANNG

Inclusive motifs
|      motif  | sites  | log like.  |         P  |        dP  |   tot P  | consensus
|            1  |      9  |   -10105.9  |  68.58%  |  68.58%  |  68.58%  |  [A/G]TGACGTCANNG
| 514(  1s)  |     13  |   -10083.3  |  31.42%  |  31.42%  | 100.00%  | TGACGTCANNG[C/T]
```

Figure 5: Detailed output of the BayesMDmarginal program.

6 Contact

Questions and comments about BayesMD should be sent to:
Man-Hung Eric Tang (manhung@binf.ku.dk)
Ole Winther (owi@imm.dtu.dk)