A User Manual for MultiModule

Qing Zhou[*]

November, 2007

This is a beta version of the MultiModule program, which finds modules of motifs in multiple species. You are welcome to use this program in your research. The formal paper that describes the statistical model and method upon which MultiModule is developed is published in

> Zhou, Q. and Wong, W.H. (2007) Coupling hidden Markov models for the discovery of cis-regulatory modules in multiple species. *Annals of Applied Statistics*, 1: 36-65.

1 Input format and options

After putting MultiModule.tar in an OS or Linux system, please unpack the tar file and then type "./MultiModuleOS" or "./MultiModuleU". A simple user manual will appear on the screen as shown in Table 1.

Table 1: A simple screen manual of MultiModule

```
Usage: ./MultiModule -i Seqfile (options)
See Readme.pdf for the format of input sequences

Options:
-o Output file (default output.txt)
-n Maximum number of iterations (default 1000)
-p Cutoff for posterior probabilities (default 0.5)
-N Number of species
-K Number of motifs
-L Expected module length (default: 200)
-w Minimal motif width (default: 8)
-W Maximal motif width (default: 15)
-u Probability of updating alignments at each iteration (default: 0.2)
-c Run in the collapsed sampler mode (recommended)
-s Run in the motif mode
```

---

[*] Department of Statistics, University of California, Los Angeles. Email: zhou@stat.ucla.edu.

The detailed explanations of these input options are given as follows.

- **-i: The input sequence file.** The input data for this program are sequences from ortholog genes in multiple species. Suppose we have n genes from N species. The sequences are all put in fasta format into one file with the following order.

  ```
  >gene 1 in species 1
  .......
  >gene 2 in species 1
  .......
  .
  .
  .
  >gene n in species 1
  .......
  >gene 1 in species 2
  .......
  .
  .
  .
  >gene n in species 2
  .......
  >gene 1 in species 3
  .......
  .
  .
  .
  >gene n in species N
  .......
  ```

  If an ortholog sequence is not available, please still include it in fasta format but use an “N” as the corresponding sequence. For instance,

  ```
  >gene 2 in species 2
  N
  ```

  The name of each sequence does not matter, but please make them unique in your input file.

- **-n Maximum number of iterations.** MultiModule is based on a Gibbs sampler which iteratively samples from conditional distributions to approximate the target joint distribution. “-n” specifies the total number of iterations for MultiModule. We recommend to run 1000 to 2000 iterations. The first 50% of iterations are treated as a burn-in period, which are not used in computing posterior probabilities.

- **-p Cutoff for posterior probabilities.** This option specifies the cutoff value for posterior probabilities to generate predicted modules and motif sites based on their respective posterior probabilities. We recommend to set p between 0.5 and 0.7.

- **-N Number of species.** This specifies the number of species included. Please be aware that the computational complexity of MultiModule is proportion to 2^N. Thus including many species will result in a long running time. In our experience, 3 or 4 species are usually enough.
- **-K Number of motifs.** This gives the number of transcription factors (TFs) assumed in the input sequences. You may want to specify this value a bit larger than your expected number of TFs. For example, if you suspect there are 3 motifs in the input sequences, you can specify this value to be 4 or 5. First, you may find some new motifs. Second, some low complexity sequence patterns, such as "GGGGCGGGG", are usually enriched in mammalian data. Allow some extra motif patterns may decrease the contamination of these low complexity patterns with your true motifs.
- **-L Expected module length.** Input expected module length. This serves as the prior mean of a module length. Thus in the output modules, each may have distinct length and some of them may have much longer or shorter length than this expected value.
- **-w and -W Motif width.** These two options tell MultiModule the range to update motif width, i.e. it updates motif width in [w,W] for each motif.
- **-u Probability of updating alignments at each iteration.** One unique feature of MultiModule is that it updates ortholog alignments dynamically, which takes into account the uncertainty of multiple alignments. However, this step is quite time-consuming. Thus we usually update alignments with a relatively small probability for each iteration. If you set -u 0.2, then on average MultiModule updates alignments every 5 iterations.
- **-c Collapsed sampler version.** If you set "-c", then all the parameters in MultiModule are integrated out, which usually gives faster convergence. We always recommend to use "-c".
- **-s Run in the motif mode.** This option tells the program to run under the motif mode, in which no module structure is assumed. This will be useful if no modules but only individual motifs are contained in the input sequences.

2 Format of output files

MultiModule will output a few files after running. The main output file, the one specified in the "-o" option, contains the detailed summary information, including the module locations and compositions, the score and matrix for each motif, and the detailed information about all the predicted motif sites.

2.1 The output module format

Each module is output in one row with five columns, such as:

```
0	0	265	327	+3(265)-1(300)-2(319)
```

The first two columns give the indices of the species and the sequence within the species. Here it is sequence 0 in species 0 (all index starts from 0). The 3rd and 4th columns give the start and end positions of the module, relative to the left end of the sequence. The last column gives the composition of the module, which is the same as that of CisModule output. Please refer to “http://www.stat.ucla.edu/∼zhou/CisModule/Readme_L.pdf”.

2.2 The output motif format

For each motif, a Bayesian score will be output, which can be used to rank the same motifs output from multiple runs (see Zhou and Wong 2007 for more details). Then an estimated weight matrix (PWM) is given and followed by all the predicted sites from all the species. Each motif site is output in two rows, such as

```
>1    1    312    -1    0.964    1.000
GATAATTGGT
```

The first line (always starting with ‘>’) gives the information about this site and the second line is the actual sequence of the site. The first line contains 6 columns.
Column 1: index for species.
Column 2: index for sequence within the species.
Column 3: start position of the motif site.
Column 4: orientation of the site: ‘1’, forward strand; ‘-1’, backward strand.
Column 5: probability score of this site (between 0 and 1).
Column 6: probability of this site being aligned to other orthologs (between 0 and 1).

2.3 Other output files

The “*_Pm.txt” file gives the number of iterations in which a sequence position is sampled as within a module. These numbers divided by the post burn-in iterations (= maximal number of iterations/2) give the posterior module probability for all the sequence positions. The numbers for a sequence are written in one line with tab delimitated. In addition, MultiModule also generates a “*module.txt” and K (the number of motifs in the input options) “*motif$k$.txt” files ($k = 1, \cdots, K$), which contain the location information of each prediction. These can be extracted from the main output file, but are provided for the convenience of large-scale research. If running under the motif mode, module-related files will not be generated.

When running MultiModule, you may check the file “*.info” for current progress.

3 An example and some tips

There is an example input sequence set, “InputSeq.fa”, in the downloaded package. This data set contains 3×20 sequences, i.e. 20 sequences from each of 3 species. There is also a runexample.sh file, which gives you the parameters for MultiModule on this data set. Expected output files “output_result.txt” and others are also provided. Please note between multiple runs, output results may varies slightly.

You may want to run repeat masker before using MultiModule. As discussed in the paper, it is recommended to run MultiModule multiple times. For each run, you can set only for 500 to 1000 iterations. But you may want to run it 10 or more times independently. Then you can rank all the output motifs (PWMs) by their scores as reported in the output files, and use the combined prediction method discussed in Zhou and Wong (2007). The posterior module probabilities (provided in the “*_Pm.txt” files) will be useful to calculate the average module probabilities over multiple runs, which are needed to determine the combined-predicted modules.