User Manual for GibbsModule

Dan Xie

March, 2008

GibbsModule is a program for *de novo* discovery of cis-regulatory modules on genome data from multispecies. The program is still in its beta version. May any alteration occurred on the program in the future, this manual is to be changed accordingly. You are welcome to use this program in your research.

1 Input File Format

By typing "./GibbsModule", a brief introduction about the input parameters will be shown as follows:

```
Usage:./a.out -i Inputfile -l Motif_length -o Species_num -t Iteration [-a matching score] [-b
mismatch penalty] [-c open indel penalty] [-d extending penalty]

Parameters:
-i Inputfile: path\filename of input file
-l Motif_length: expectation of motif length
-o Species_num: number of orthologs
-t Iteration: number of iterations for the program to converge
[matching score: matching score for module alingment (default: 1)]
[mismatch penalty: mismatch penalty for module alingment (default: 0.33)]
[open indel penalty: open indel penalty for module alingment (default: 1)]
[extending penalty: extending penalty for module alingment (default: 0.33)]
```

The first 4 input parameters are mandatory for GibbsMoudle program while the last 4 parameters are optional. The details about the parameters are as follow:

-i Inputfile: The input file for GibbsModule consists of sequences from ortholog genes. If user want to input a file consists of N genes from 3 species, the inputfile looks like:

```
>gene1
1.GTTCTTTG……  (← input sequence of the first species in from gene 1)
2.AGCCAAG……  (← input sequence of the second species in from gene 1)
3.GTTGATT……  (← input sequence of the third species in from gene 1)
>gene2
1.TCTGAAG……  (← input sequence of the first species in from gene 2)
2.CGTAAGC……  (← input sequence of the second species in from gene 2)
3.GCTGACC……  (← input sequence of the third species in from gene 2)
.
.
.
>geneN
1.AAGGCAA……  (← input sequence of the first species in from gene N)
2.CAAGGCA……  (← input sequence of the second species in from gene N)
3.CAATGTG……  (← input sequence of the third species in from gene N)
```

-l Motif_length: This parameter tells the program the expected length of the output motifs. While GibbsModule does not allow variation of motif length within one run, the length of out motif will have a fixed length of what user provide with this parameter.

-o Species_num: This parameter tells the program the number of species for the input. Please note that the more species are included in the input, the more time will be consumed. Actually, more species will not necessarily improve the accuracy of module discovery. A reasonable species number should be 2 or 3.

-t Iterations: This parameter tells the program the number of iterations for the Gibbs Sampling process. Gibbs Sampler needs a considerate number of iterations for converging. The actual converging iteration number is different in different situation. We recommend an iteration number of at least 300.

-a Matching Score: Matching score for the alignment algorithm.

-b Mismatch penalty: Mismatch penalty for the alignment algorithm.

-c Open indel penalty: Open indel penalty for the alignment algorithm.

-d Extending penalty: Extending penalty for the alignment algorithm.

Usually the 4 optional parameters do not need to be changed to other values, especially when you do not understand what they are.

2 Output File Format

There will be instant out put be shown on the screen while GibbsModule is running. Two output files will be generated in the same directory with the program, one named “verboseoutput”, the other named “GibbsMoudle_output”.

“verboseoutput” records the PSSM and the sampled positions on each sequence for every iteration as well as the best PSSM by far at every iteration. An example for one iteration is shown as below:

	A	C	G	T	Con	
1	0.097	0.065	0.16	0.71	T	
2	0.097	0.74	0.065	0.13	C	
3	0.32	0.065	0.065	0.58	T	
4	0.097	0.065	0.065	0.81	T	
5	0.13	0.065	0.065	0.77	T	
6	0.097	0.065	0.77	0.097	G	
7	0.29	0.065	0.065	0.61	T	
8	0.097	0.065	0.097	0.77	T	
9	0.61	0.13	0.065	0.23	A	
10	0.13	0.32	0.065	0.52	T	
11	0.19	0.13	0.61	0.097	G	
12	0.52	0.16	0.23	0.13	A	
Seq1	233	TCTTTGATACGT::		233	TCTTTGATACGT->TCTTTGATACGT	
Seq2	900	TTATTGATTTGA::		900	TTATTGATTTGA->TTATTGATTTGA	
Seq3	428	TCTTTGTTATGA::		428	TCTTTGTTATGA->TCTTTGTTATGA	
Seq4	370	TCATTGTTAAGG::		370	TCATTGTTAAGG->TCATTGTTAAGG	
Seq5	56	TCATAGTTTTGG::		56	TCATAGTTTTGG->TCATAGTTTTGG	
Seq6	795	TCTTTGTTTTAA::		815	TCATTGTTATGC->TCATTGTTATGC	
Seq7	297	TCTTTGATACGC::		317	TCTTTGATATGA->TCTTTGATATGA	
Seq8	572	TCTTTGTTATGG::		572	TCTTTGTTATGG->TCTTTGTTATGG	
Seq9	397	TCTTTGTTATGA::		437	TCTTTGATTTAA->TCTTTGATTTAA	
Seq10	731	TCTTTGATACGG::		731	TCTTTGATACGG->TCTTTGATACGG	
Seq11	718	TCATTGTTTCGC::		698	TCATTGTTCTAA->TCATTGTTCTAA	
Seq12	314	TCTTTGTTATGA::		314	TCTTTGTTATGA->TCTTTGTTATGA	
Seq13	714	TCTTTGTTAACC::		694	GCTTTGTTACCA->GCTTTGTTACCA	
Seq14	524	TCTTTGTTACAC::		524	TCTTTGTTACAC->TCTTTGTTACAC	
Seq15	586	TCTTTGTTACGA::		586	TCTTTGTTACGA->TCTTTGTTACGA	
Seq16	549	GCATTGTTCCGA::		569	GCTTTGTGACGA->GCTTTGTGACGA	
Seq17	490	GCTTTGATTCCC::		510	TCTTTGATACCA->TCTTTGATACCA	
Seq18	580	TCATTGTTAGGA::		620	GCATTGTTCCGA->GCATTGTTCCGA	
Seq19	780	TCTTTGTTATGG::		780	TCTTTGTTATGG->TCTTTGTTATGG	
Seq20	349	TCATTGTTATGA::		349	TCATTGTTATGA->TCATTGTTATGA	
Seq21	436	TCTTTGTTTTGC::		436	TCTTTGTTTTGC->TCTTTGTTTTGC	
Seq22	600	TCTTTGTTATGA::		600	TCTTTGTTATGA->TCTTTGTTATGA	

The upper part is the PSSM sampled for this iteration; the lower left part is the best positions sampled by far; the lower right part is the positions sampled this iteration.

We choose to output the verbose version for every iteration because from our experience, the outputs recorded during the sampling process usually are informative to us to decide more accurately the results.

For each run of the program, totally three motifs will be reported by GibbsModule, which means the Gibbs sampler will run three times separately, and each time the Gibbs sampler will run the same iteration numbers provided by users. The three separate Gibbs sampler will report

different motifs(or module core in the notion of GibbsModule). In the output file, the ending of each Gibbs sampling process are marked with

```
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ The 1st motif ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ The 2nd motif ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
or
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ The 3rd motif ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

“GibbsModule_output” records the final prediction of our algorithm, including the predicted PSSM, the locations of the predicted module cores, and the fasta format predicted module sequences that are ready to be used as inputs for other motif discovering algorithms. The output module sequences are 150 base pairs long. An example is shown as below:

```
      A      C      G      T      Con
1     0.28   0.21   0.31   0.17   G
2     0.38   0.069  0.28   0.24   A
3     0.21   0.24   0.28   0.24   G
4     0.41   0.17   0.17   0.21   A
5     0.34   0.14   0.14   0.34   A
6     0.28   0.069  0.31   0.31   G
7     0.34   0.24   0.21   0.17   A
8     0.38   0.14   0.24   0.24   A
Seq1   663    GTAAATAT
Seq2   165    GGAGTATA
Seq3   933    GTGGTAGA
Seq4   356    TGCAAAAT
Seq5   0     AGGAAGCA
Seq6   834    AGCTGGGA
Seq7   594    GGAAGGGC
Seq8   937    TAGTAGAG
Seq9   108    GTGGCACA
Seq10  691    CAGGATTG

Modules predicted by GibbsModule:
>Seq1
ATCACCAACTAGTGTTGCCAAAATGAACACTGAATTTCATAGAGAAATGCTAAGCTTGAAATTTAAAGCATCCTT
GTAAATATGCCTCATTATGTATGTACATACATATGTGTGTATGTGTACACATATAAACATTGCACATACATTATG
>Seq2
TTTAAGCTGCTGTTCCAAATTTGCCAGGAAAAATCTTCACCTTCACACTCAAGTAATTAGTACCACATTTAGCAT
GGAGTATACTTTGTCACAAGAATGTGATAAAATTCTTTTATAAGAATTTACTCTGCTAAATACATGTGATGAAGT
>Seq3
GTTGAATGAAGTTGTTGAACACCAAGTAAATGAATGGTGGAAAACTCTTGCCAAAAGCAGGGACTGGGGTTTT
AGAGTGGCCGTGGTAGATTGACTGCACTATTGTCCTCAACTTACGTGCCTCCATGTATCTACAGCCTTTGATAC
TG
>Seq4
AAATATTATTTAAATGGAATCTGGATCTCAAGTGTGAATGATAAGAACTATATTTGAATTTAGATAGATGTTATAT
GCAAAATTGTACTATATGTTATTTATTTTTGTACTATGTGAACATAGTCTCATGTACTGAATTCTACCAACATA
>Seq5
GGAAGCATTTCCTGATCTTCCCACTACCATCAAGTTGATTTTTGCACCTCTTCTCTGTAAACAAGAGCACCCTG
CCCATGTCTCTGACAGCAGTTTTCACATTGTTTGACCATCACAGGTTTATCTCTGCAGGGTCACTATGGATCCA
GA
>Seq6
CAGTGGCACGATCTCGGCTCACTGCAAGCTCCACCTCCAGGGTTCATGCCATTCTCCTGCCTCAGCCTCCTG
AGTAGCTGGGACTACAGGTGCCCGCCACCATGCCCGGCTAATTTTTTTTTTTTTTGTATTTTTAGTAGAGAAGGG
GTTT
>Seq7
AAAGTCTCTGATATGCAGAAATAATGGCGCAAGCTGTCTCTCTCTCTCCCCCTCTCTGCCTCGGCTGCCAGGC
AGGGAAGGGCCCCCTGTCCAGTGGATACGTGACCCACGTGACCCTACCTATCACTGGAGATGATTCACACTC
TTTAC
>Seq8
GGGTTCAAGCCATTCTCCTGCCTCAGCCTCCTCAGTAGCTGGGATTACAGGCACCCGCCACCATGCCTGGCT
AATTTTTGTATTTTTAGTAGAGATGGGGTTTCACCATGTTTGCCAAGCTGGTCTCGAACTCCTGACCTTGTGAT
CCA
>Seq9
TAAGTGGTTTGGGATGAGATTTTTTTTTCCTTTTTATTAGATAGAGTCTCGCTCTGTTGCCCAGGCTAGAGTGC
AGTGGCACAATCTCGACTCACTGCAACCTCCACCTCCCAGGTTCAAGCAGTTCTCCTGCCCCAGCCTCCCAAG
TAG
>Seq10
```