ChIPModule 1.0 Manual

Computational Systems Biology Lab

EECS,UCF

UNIX version ChIPModule software----------

I. Command line version:

0.Usage example:
python crm.py -i examples/tst -o examples/out_tst -s 10 -c 0.001 -m data/normalized_transfac_matrices -l data/lambda -f xls -d N
This is the command to run the tst example. (tst is given in "example" directory)

1.Prerequisite
You need to install Python. check this website for the installation of the Python, http://python.org/.

2. Function:
The function of this software is to obtain motif combinations using the 522 transfac motifs or a group of motifs provided by users. A motif combination corresponds to a group of transcription factors that coordinately regulate their target genes.

3. Input parameters:
-i <input sequence>
-m <pwm>, 'default': Transfac PWMs, or <your PWMs file>
-s <the required minimum number of sequences that contain the instances of a combination>
-l <lambda>, 'default': default lambda used, or <your lambda file>
-c <the Bonferroni corrected pvalue cutoff for motif combinations>
-o <output folder>
-f <format>: txt,xls
-d <want the fully detailed motif combinations file>: Y (yes) or N (no)

Let's explain these options one by one:
(1) -i <input sequences>
Here you need to input the sequences, which will be used for finding the significant motif combinations. For ChIP-seq data, the sequences are from the ChIP-seq peak regions. We recommend that these sequences are repeat masked by using the repeatmasker at http://www.repeatmasker.org/. If you only have the coordinates of ChIP-seq peak regions (for example bed format file), you can use 'UCSC TABLES' to download the corresponding sequences. You can access the UCSC TABLES at http://genome.ucsc.edu/cgi-bin/hgTables?command=start. The sequence you input must be in fasta-format.e.g:

```
>ENSG00000000457
TGTTTTTCAGTCCAAACCTATTCCTGAATCTGTGTAGCCATCCCTTACTTGCAGTAAATA
GCTTGGTGTGCCTAATTTCAGGGGATCCCTTTCTTAAGTCTTCCTATAGGCTCAACACTT
TCTGGCAAAACTTACTGCGGTCAGCTGAAATACGTTTTCTGTTCATGTCTCCCACCCACA
AATGTAATGCCTTTTCCGTCTTTTGCTGTTGCACCTCAAACTACAAAAGTTACAGCCACA
GTGCATGATAGGTGCTTAGTTAACTAGCACGAATCTTTTT
>ENSG00000000457
ATAATTTGTTCTTATCATAGATTTTGGCAACCTCAGCATATGACTCTTTTTCTTTCCTTA
TTAAGTCCAGAGCTTTCACCTTTTCATTTAAAGGAAGCACTTTACAGCTTCTCCTCGGCA
```

TATCCAAATTGTCAGCACAATTCCTGTGCTTTGCGGCCATTATTAAGTAAAATAAGGGTT
ACCTGAACATAAGCAGGTGGTATCTCTCATTATCTCTGCTATACTACGGGTGCATCTGTT
AACCAAAGCAGCTACTGAGTGGGTAATGAGTGGATACTGTACACAGCGCGGCTACCTGCG
ACAGATGATTCGCGTCAAGGGTGGGACAGAGTGGATGGCCCGAGATTTAATCGGGCTACT
CAGAATACAGGCGGTTTGAAACTTAAGAATTGTTTATTTCTGGAGTTTTCCATTTAACAT
TTCTGAACCAAGGTTTGACCACAGGTAACTGACACTGCAGAAAGCATGGAGGGGGGCAAA
ACTACTGCATTAATATTAAAATTTTCAAATATTACTTTTTGCTAAATGAAATGTGATTCA
GGACCTTCCCTCTCAAAGATCAAGCGAGATCACCACGACCTCCGCCAGCAGCGGCTCTGC
ACGACTCCACCCTCGCAGCCCAGCCAATCAAAGCTACAGGTTGAGTGACGTCACTCTCCT
GAAAGTCCTCGCTAATTCCCGTACTCCTTTTCTCCGCCCT

(2) -m <pwm>,'default' or <your PWMs file>
This specify the PWMs (all motifs should be PWM format). If you choose 'default', then our default 522 Transfac matrices (in the data directory: 'normalized_transfac_matrices') will be used. If you input the your own PWMs file <your PWMs file>, then your own PWMs file will be used. The format of the PWM file should be like:
>M00001   6.6244035486 V$MYOD_01 mouse MyoD
0.211538461538 0.365384615385 0.365384615385 0.0576923076923
0.365384615385 0.211538461538 0.365384615385 0.0576923076923
0.519230769231 0.0576923076923 0.211538461538 0.211538461538
0.0576923076923 0.826923076923 0.0576923076923 0.0576923076923
0.826923076923 0.0576923076923 0.0576923076923 0.0576923076923
0.0576923076923 0.0576923076923 0.673076923077 0.211538461538
0.0576923076923 0.211538461538 0.673076923077 0.0576923076923
0.0576923076923 0.0576923076923 0.0576923076923 0.826923076923
0.0576923076923 0.0576923076923 0.826923076923 0.0576923076923
0.0576923076923 0.211538461538 0.365384615385 0.365384615385
0.0576923076923 0.365384615385 0.0576923076923 0.519230769231
0.211538461538 0.0576923076923 0.519230769231 0.211538461538

The following is the description of the above PWM:
The first line:
>M00001   6.6244035486 V$MYOD_01 mouse MyoD
gives the information of the motif name 'M00001', the score cutoff for this motif (6.6244035486), transcription factor MyoD corresponding to the motif 'V$MYOD_01', and the organism (mouse). They are separated by tab('\t'). If you don't know corresponding transcription factor for your defined motifs, just use 'NA_i'. The score cutoff is chosen such that the score of a random segment will have a score no smaller than the cutoff with the probaility of 0.0001 (lambda). Of course, other lambda values can also be used. see '-l' section for details. Given a DNA segment, we use the PWM to score this segment. If the score > score cutoff, we can claimed the DNA segment as a putative instance of this motif (PWM).

The following lines:
0.211538461538 0.365384615385 0.365384615385 0.0576923076923
0.365384615385 0.211538461538 0.365384615385 0.0576923076923
0.519230769231 0.0576923076923 0.211538461538 0.211538461538
0.0576923076923 0.826923076923 0.0576923076923 0.0576923076923
0.826923076923 0.0576923076923 0.0576923076923 0.0576923076923
0.0576923076923 0.0576923076923 0.673076923077 0.211538461538
0.0576923076923 0.211538461538 0.673076923077 0.0576923076923
0.0576923076923 0.0576923076923 0.0576923076923 0.826923076923
0.0576923076923 0.0576923076923 0.826923076923 0.0576923076923
0.0576923076923 0.211538461538 0.365384615385 0.365384615385
0.0576923076923 0.365384615385 0.0576923076923 0.519230769231
0.211538461538 0.0576923076923 0.519230769231 0.211538461538

represent the PWM. Please note: the probability of (A,C,G,T) in each position should NOT be transformed into log2(p). Within each line, the probabilities of A,C,G,T are separated by space (' ').

(3) -s <the required minimum number of sequences that contain the instances of a motif combination>
Here you need to provide a number as the minimum frequency of a motif combination that we intend to identify. For example, "-s 20" means that all significant motif combination will occur in at least 20 sequences. That is, putative binding sites of each motif in a motif combination will co-occur in at least 20 sequences.

(4) -l 'default' or <your lambda file>
This parameter specify the lambda values (mentioned in '-m' section.) You can use '-l default' to choose the default lambda file. You can also specify your own lambda file by using "-l <your own lambda file>". The following is the format of lambda file:

>M00001 0.0001000334 MyoD
'M00001' denotes the motif ID. "0.0001000334" denotes the lambda obtained. Note that the lambda will not be exactly the same as 0.0001 for different PWMs. You can generate your own lambda and corresponding score cufoff for each motif (PWM) by using the code we provide (get_lambda.py). It is available at 'tool' directory.

usage of 'get_lambda.py':
usage: python.get_lambda.py <input_seq><motif><lambda cutoff>

(5)-c <the Bonferroni corrected pvalue cutoff for motif combinations>
pvalue cutoff specify the significance level of motif combinations. For example, -c 0.01 means that all the output motif combinations will have a corrected pvalue no larger than 0.01.

(6) -o <output folder>
This specify where to save all the results. For example, -o output_folder, then all the output files will be put on the 'output_folder' you defined.

(7) -f <format>: txt,xls
This specify the output format. You can choose 'txt' to get plain text result or you can use 'xls' to obtained excel format result (only use tab <'\t'> as the separator). Note that when opening the xls file, you should only choose tab ('\t') as the separator.

(8) -d <want the fully detailed motif combination file>: Y/N
If you choose Y, then the software will output every details about the motif combination obtained,including motifs included, target genes, the TFBSs for the motifs in combination in their target genes. It is time consuming to generate output this result. So, it is recommended to choose N if you don't need the every details about the motif combinations.

4. Output:
4.1 Motif combination file (.mc file)
This file gives the information about motif combination IDs, motifs (Transcription Factors) in each combination, the number of sequences containing instances of each motif combination, the Bonferroni corrected pvalue of each motif combination, and target sequences of each motif combination. see the following example.
>MOD1 M00108(NRF-2) M00008(Sp1) M00189(AP-2) (10) (0.0)
ENSG00000001461,ENSG00000001036,ENSG00000002016,ENSG00000001617,ENSG00000002586,EN
SG00000003509,ENSG00000002822,ENSG00000004142,ENSG00000004848,ENSG00000004939

4.2 putative TFBS (transcription factor binding sites) and TFBS locations (.ptfbs file)
This file gives the information about the TFBS and TFBS locations within each input sequences.
The following is the format of this file:
gene_id
TFBS (corresponding motif IDs)
TFBS_location (denotes the distance to the 5' start of sequence). See the following example:
>ENSG00000000457
M00001 M00005 M00017 M00039 M00040 M00041 M00051 M00053 M00066 M00071 M00084
M00101 M00101 M00101 M00113 M00122 M00128 M00129 M00130 M00135 M00137 M00162
M00162 M00175 M00175 M00176 M00177 M00178 M00179 M00184 M00185 M00205 M00206
M00209 M00242 M00250 M00264 M00271 M00287 M00294 M00327 M00338 M00340 M00373
M00378 M00413 M00414 M00415 M00416 M00437 M00446 M00447 M00451 M00474 M00489
M00491 M00493 M00493 M00494 M00496 M00499 M00500 M00513 M00531 M00619 M00621
M00623 M00623 M00631 M00632 M00632 M00641 M00641 M00644 M00665 M00684 M00687
M00687 M00690 M00692 M00692 M00693 M00694 M00698 M00706 M00712 M00720 M00720
M00726 M00731 M00744 M00770 M00775 M00789 M00800 M00801 M00802 M00805 M00805
M00806 M00916 M00917 M00921 M00927 M00931 M00933 M00933 M00941 M00955 M00957
M00967 M00981 M00982 M00984 M01010 M01023 M01033 M01035
490 880 942 945 945 945 80 701 596 488 766 695 718 788 941 489 242 687 687 415 559
287 821 139 581 581 943 943 943 530 920 297 560 553 615 917 973 135 918 687 942
944 388 609 901 373 529 671 969 969 767 568 460 686 50 766 437 972 972 190 717 842
942 387 481 717 68 968 936 661 758 364 697 404 762 77 548 920 422 276 311 491 944
883 767 491 616 768 667 737 296 319 917 245 879 948 783 852 927 912 943 942 304
139 617 169 904 454 291 291 361 944 905 170 698 363 772 452

4.3 Detailed motif combination file (.dmc file)
This file gives the detailed information about motif combination. Obtaining this file is time consuming, if you don't need all the details about each motif combiantion. Then, you can use "-d N" option to quit producing this file. It starts with a motif combination ID, motifs (Transcription Factors) in this combination, the number of sequences containing instances of this motif combinatnion, the Bonferroni corrected pvalue of this motif combiantion. It then follows with each of the target sequences of this motif combination, an instance of each motif in this combination, and the location of this instance. Note that a motif may have multiple instances in one sequences and only one of these instances is chosen in this file. The information of all instances of a motif can be found in the second output file mentioned above. See the following example.
>MOD1 M00108(NRF-2) M00008(Sp1) M00189(AP-2) (10) (0.0)
ENSG00000001461 ACGTACT(459) AGTACTG(338) AAAAAA(367)
ENSG00000001036 ACGTACT(459) AGTACTG(338) AAAAAA(367)
ENSG00000002016 ACGTACT(459) AGTACTG(338) AAAAAA(367)

4.4 Motif frequency file (.mf file)
This file gives the list of motifs that are contained in the output motif combinations. The motifs are sorted according to the number of sequences containing their instances. The first column is the motif ID, the second column is the corresponding transcription factors, the third column is the number of sequences containing instances of this motif, the last column is the percent of input sequences containing the instances of this motif.
an example

5. This software depends on files in code and data directories.
To run the sofeware sucessfully, you should always keep the code and date directories.

II. GUI version

(1) You need python installed.
you can check this website to see how to install python.(http://python.org/)
Note:
Both python2.7 and python3.2 are supported. However, it is recommendated to use python2.7.(It is easier to get tkinter package).

(2) You also need to install python Tkinter package.
If you download python from http://python.org and install it, then Tkinter package is already included by default.
If not, you can check http://www.tkdocs.com/tutorial/install.html to see how to install it.
For common linux distro users, you can also check here (http://tkinter.unpythonic.net/wiki/How_to_install_Tkinter) to see the steps of installing tkinter package.

(3)After (1) and (2) have been finished, you can just double click "run_crm_gui.sh" to get the gui.
All the parameters are the same with the command line version.

//
//
//
Windows version ChIPModule
software--

I. Command line version

0.Usage example:
python crm.py -i tst -o out_tst -s 10 -c 0.001 -m data/normalized_transfac_matrices -l data/lambda -f xls -d N
This is the command to run the tst example. (tst is given in "example" directory).
Please replace the 'python' with your detailed python path. If you don't know the exect path to the python, please
double click the 'check_python_path.py' in the 'tools' directory to get it. For example, my python path is :'C:\Python27\python.exe'
Then, in order to run the program, the command should be changed to:
C:\Python27\python.exe crm.py -i examples\tst -o examples\out_tst -s 10 -c 0.001 -m data\normalized_transfac_matrices -l data\lambda -f xls -d N

1.Prerequisite
You need to install Python. check this website for the installation of the Python, http://python.org/.

2. Input and output
The input parameters and output files are exactly the same with the Unix version.

3. This software depends on files in code and data directories.
To run the sofeware sucessfully, you should always keep the code and date directories

II. GUI version

(1) You need python installed.

you can check this website to see how to install python.(http://python.org/)

(2) Double click the "ChIPModule.gui.py" to get the gui.
All the parameters are the same with the command line version.