```
+~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~+
+                                                                                               +
+     coMOTIF: a mixture model with an EM algorithm for motif and co-factor motif discovery     +
+                                             v1.0                                              +
+                                        December, 2010                                         +
+~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~+
```

Usage: coMOTIF -fseq seqFile -fpwm0 pwmFile -model modelType optional arguments

-fseq string File name for sequence data in FASTA format (case insensitive).
Example:

```
>200C_17C_chr1:91356950-91357349
GTCGGTTTTCTTGCCAACACATACTTTATTTTCTCTTTATGGCTAAATAA
AACTCCTGTGTGTGTATGTACCACATTTCACCCGTTCATCTGCTGCCAGA
CGCTGGGCTGCAGTGAACAGCGATGCGGTAGGCATGAACGAGCAGCATGA
ACGAGCAGGTGTCCCTGCGGTGTGCTTAGGCCTTTGGGTAGATTCCCAGG
TACGGGTGGGTCATGTGATCTTGCCAATTGTTTTAAATTGAAGCCAGGTT
TTTGGTTGACTCATTCTCATCAGCCACCAGTAGATGGAGGGAGTGAGACA
TGCAAACAGAGTGCTGTCCCCACTGCCCGGAGTCTGTGACATCCATCCCT
AAAGATGTGTGTTCATATTGTTCCGTGTGGATGTGCCCGAGTGTGTGTAG
```

-model 1PWM or 2PWM Specify which model to run: 1PWM - one-motif model or 2PWM - two-motif model (default). The 1PWM model consists of background and one motif (two components) whereas the 2PWM model consists of background, motif1, and co-factor motif2 (three components). The one-motif model identifies one motif at a time whereas the two-motif model identifies two motifs simultaneously.

-fpwm0 string File name for the seed PWMs. All PWMs must be placed in a single file (see below). Each PWM must be represented by a) PWM identifier (with or without #); b) number of rows (4) & columns in each PWM; c) PWM matrix either in integer counts OR decimal frequencies. The first PWM is always taken as the PWM for the primary motif. Depending on the model type (1PWM or 2PWM), the remaining PWMs are ether used as the starting PWMs for the co-factor motifs or the starting PWMs for the one-motif model.

In the example below, there are three PWMs. For the two-motif model, coMOTIF automatically carries out two independent runs: HNF4A/HNF1A and HNF4A/Foxa2. For the one-motif model, coMOTIF carries out three independent runs with each of the three PWMs as the starting PWM.

Example1:

```
#HNF4A.mx
4       13
28      2       12      5       3       59      53      56      4       6       3       4       42
7       2       4       23      51      1       2       1       4       2       22      49      7
27      56      35      20      4       3       10      8       58      33      11      5       10
5       7       16      19      9       4       2       2       1       26      31      9       8
#HNF1A.mx
```

```
4       14
5       1       1       1       20      16      1       8       14      2       0       13      8       5
0       0       0       0       0       2       0       2       0       0       4       1       8       13
14      20      0       0       0       1       0       4       1       0       0       3       3       0
2       0       20      20      1       2       20      7       6       19      17      4       2       3
Foxa2
4       10
0.4761 0.5951 0.5951 0.3095 0.0477 0.8807 0.9521 0.9759 0.0001 0.8807
0.0953 0.1429 0.0715 0.0953 0.4999 0.1191 0.0477 0.0001 0.5237 0.0001
0.0477 0.2143 0.1905 0.5475 0.0239 0.0001 0.0001 0.0239 0.0239 0.0715
0.3809 0.0477 0.1429 0.0477 0.4285 0.0001 0.0001 0.0001 0.4523 0.0477
```

If the co-factor motif is not abundant, it may be difficult to identify - meaning that the starting PWM may converge to a different PWM. In that case, one might want to consider: 1) set -full to 0. This allows coMOTIF to run the mixture model with the PWM for the co-factor motif fixed (without being updating) (preferred option); 2) alternatively, list the PWM for the co-factor motif multiple times in the PWM file (see example below) and set -maskS to 1 (see below). This allows coMOTIF iteratively masks the motif instances once found. Masking is not ideal and it does not guarantee finding desired motif.

Example2:

```
#HNF4A.mx
4       13
28      2       12      5       3       59      53      56      4       6       3       4       42
7       2       4       23      51      1       2       1       4       2       22      49      7
27      56      35      20      4       3       10      8       58      33      11      5       10
5       7       16      19      9       4       2       2       1       26      31      9       8
#Foxa2_a
4       10
20      25      25      13      2       37      40      41      0       37
4       6       3       4       21      5       2       0       22      0
2       9       8       23      1       0       0       1       1       3
16      2       6       2       18      0       0       0       19      2
#Foxa2_b
4       10
20      25      25      13      2       37      40      41      0       37
4       6       3       4       21      5       2       0       22      0
2       9       8       23      1       0       0       1       1       3
16      2       6       2       18      0       0       0       19      2
```

-full 1 or 0 Indicator for whether or not to update the PWM for the co-factor motif in EM (1-yes[default], 0-no) This argument allows one to turn on or off the optimization for the co-factor PWM and allow coMOTIF to identify low-abundant co-factor motifs. However, this depends on the quality of the starting PWMs.

PRIOR PARAMETERS FOR PROPORTIONS

A sequence may not contain a binding site for either motif (pure 'noise'). A binding site may be present in plus or reverse complementary strand of a sequence. Thus, there are nine possible states: no motif (only noise), motif 1 on plus strand, motif 1 on minus strand, motif 2 on plus strand motif 2 on minus strand, both motifs 1 and 2 on plus strand, motif 1 on plus and motif 2 on minus, and motif1 on plus and motif 2 on minus strand. coMOTIF estimates the nine probabilities for each sequence. Summing these probabilities across all sequences give the nine proportions in the data.

By default, coMOTIF uses a flat prior (equal proportion for all nine or three states). If you choose to use different initial values (priors). These probabilities can be specified in the parameter file with the argument -fparm (below).

```
       | backg  motif2+  motif2-
-----------------------------------
backg  |   p00    p01     p02 (background, motif2 in plus, motif2 in minus)
motif1+|   p10    p1l     p12
motif1-|   p20    p21     p22
```

For a one-motif (two-component) model, there are only three such prior probabilities

```
       |
-----------------------------------
backg  |   p00
motif1+|   p10
motif1-|   p20
```

-fparm string File name for the prior parameters.
Examples

```
# flat prior motif proportions for two-motif model:
3  3
0.1111 0.1111 0.1111
0.1111 0.1111 0.1111
0.1111 0.1111 0.1111

# flat prior motif proportions for one-motif model:
3  1
0.3333
0.3333
0.3333
```

BACKGROUND MODEL

To compute the probability of a sequence being generated by the background model, coMOTIF either uses the [A,C,G,T] frequencies in the input data as the parameters for the 0th-order background model or reads in the background model parameters from a user-specified file with -fbackg argument. When a user-specified

background model is used, coMOTIF automatically chooses the highest possible order as the order for the background model. The order of the background model can be changed using -bOrder argument (below).

-fbackg string File name for higher order Markov background model. Up to 9th order (nonamer + 1nt = octamer) is allowed.

```
#monomer frequency
a       0.20850000001660
c       0.29149999998340
g       0.29149999998340
t       0.20850000001660
#dimer frequency
aa      0.04800960194357
ac      0.05151030207800
ag      0.08171634323790
at      0.02720544114470
           ....
ta      0.03460692142891
tc      0.05361072215865
tg      0.07231446287687
tt      0.04800960194357
#trimer frequency
aaa     0.01200480194395
aac     0.01550620248175
aag     0.01420568228200
aat     0.00630252106809
....
tta     0.01190476192858
ttc     0.01180472191322
ttg     0.01230492199004
ttt     0.01200480194395
           ....
```

-bOrder 0-9 The order of the Markov background model (0-9). This argument should only be used with -fbackg.

Other arguments:

-em integer Maximal number of EM steps (default: 1000).
-detail 0 or 1 Print out all nine probabilities for each sequence (0-no[default], 1-yes.
-minF float Minimal fraction of sequences containing a site required for a motif to be reported (default: 0.05).
-posWt 1 or 0 Motif location prior [1-Gaussian(default), 0-uniform]. If you expect central enrichment as in ChIP-seq, use the default. The Gaussian priors are applied to both the primary motif and the co-factor motif, but with difference variances (25 for primary and 75 for co-factor). The joint location prior is uniform (L-w1-w2+1)*(L-w1-w2+2)

-maskS	0 or 1	Indicator for whether to mask motif instances once or not once found [0-no(default), 1-yes]. For a low abundant motif, the initial PWM may converge to a different solution (PWM). Iteratively masking those motifs may lead to the identification of the motif. Note that this argument only applies to the co-factor motif.
-extTrim	0 or 1	Base extension and trimming (0 -no [default], 1 - yes).

Examples:

1. Identify one motif at a time using each of the PWMs in a file as the starting PWM with expected motif central enrichment.

   coMOTIF -fseq input.seq -fpwm0 pwmFileName -model 1PWM

2. Identify one motif at a time using each of the PWMs in a file as the starting PWM without expected motif central enrichment.

   coMOTIF -fseq input.seq -fpwm0 pwmFileName -model 1PWM –posWt 0

3. Identify a primary and cofactor motifs using the first and each of the remaining PWMs as the starting PWMs for the primary and co-factor motifs, respectively.

   coMOTIF -fseq input.seq -fpwm0 pwmFileName -model 2PWM

4. Identify a primary and co-factor motifs in ChIP-seq data with a high-order Markov background model. enrichment

   coMOTIF -fseq input.seq -fpwm0 pwmFileName -model 2PWM -fbackg backgroundFileName (see file in <examples> directory)

Description of Output Files

1) info.txt
   This file contains summary information for the run, e.g., command line, parameters, etc.

2) <PWM1_PWM2>.txt or <PWM1>.txt
   The PWM1 and PWM2 are the names of the primary and co-factor PWMs. Each contains the summary results, estimated PWMs, and the predicted locations of the binding sites.

3) <PWM1_PWM2>.loc or <PWM1>.loc
   This companion file contains the predicted locations of the binding sites. There are four columns in <PWM1_PWM2>.loc. The last two columns list only locations where both motif1 and motif2 are found in the same sequences for plotting the joint distribution.

4) estimatedPWM1.txt
This file contains the estimated PWMs for the primary motif from all runs.

5) estimatedPWM2a.txt, estimatedPWM2b.txt, estimatedPWM2c.txt
The estimated PWMs for the co-factor motifs from all runs are placed in three different files. The PWMs are in STAMP format and the files can be loaded on STAMP (http://www.benoslab.pitt.edu/stamp/) for similarity search.

A) estimatedPWM2a.txt contains the estimated PWMs that are different from the starting PWM and the estimated PWM is also degenerate (less than 1/4 of its position having 1 bit or more information on a 2-bit scale). Presumably the corresponding results (e.g., in files 2,3) might not be interesting.

B) estimatedPWM2b.txt contains the estimated PWMs that are different from the starting PWMs but not degenerate (based on above criterion). This set may be informative.

C) estimatedPWM2c.txt contains the estimated PWMs that did not diverge and are not degenerate. The corresponding results (files 2,3) may be the most interesting.

Software Download
http://www.niehs.nih.gov/research/resources/software/comotif

Contact: li3@niehs.nih.gov
