Correlation Motif Vignette

Hongkai Ji, Yingying Wei

October 13, 2015

1 Introduction

The standard algorithms for detecting differential genes from microarray data are mostly designed for analyzing a single data set. However, with the wide use of microarray technologies in biology and medicine, many different microarray studies are available for the same biological problem. Separately analyzing each data set is not an ideal strategy as it may fail to detect some key genes showing low fold changes consistently in all studies. Jointly modeling all data allows one to borrow information across studies to improve statistical inference. However, the simple concordance model, which assumes that differential expression occurs in either all studies or none of the studies, fails to capture study-specific differentially expressed genes. A more flexible model which considers all possible differential expression patterns faces the problem of exponentially growing parameter space when the number of studies increases. Here the R package *Cormtoif* fits a Bayesian Hierachical model to address this dilemma while improving inference on differential expression. The algorithm automatically searches for a small number of latent probability vectors called *correlation motif* to capture the major correlation patterns among multiple data sets. The motifs provide the basis for sharing information across studies. The approach overcomes the barrier of exponentially growing parameter space and is capable of handling a large number of studies. Missing values are also handled by *Cormtoif* .

2 Data preparation

In order to fit the *correlation motif* model, one needs to call the function *cormotiffit*. The first requirement `exprs` is the matrix containing the gene expression data that needs to be analyzed. Each row of the matrix corresponds to a gene and each column of the matrix corresponds to a sample. The data should be normalized, for example by `RMA`, thus it is in *log2* scale.

The second arguement, `groupid`, identifies the group label of each sample. Here we use data *simudata2* as an illustration. *simudata2* are combined from four studies sharing the same $3,000$ genes, each having two experimental conditions and three samples for each condition.

```
> library(Cormotif)
> data(simudata2)
> colnames(simudata2)
```

```
 [1] "gene" "R1"   "R2"   "R3"   "S1"   "S2"   "S3"   "T1"   "T2"   "T3"
[11] "U1"   "U2"   "U3"   "V1"   "V2"   "V3"   "W1"   "W2"   "W3"   "X1"
[21] "X2"   "X3"   "Y1"   "Y2"   "Y3"
```

```
> exprs.simu2<-as.matrix(simudata2[,2:25])
> data(simu2_groupid)
> simu2_groupid
```

```
  R1 R2 R3 S1 S2 S3 T1 T2 T3 U1 U2 U3 V1 V2 V3 W1 W2 W3 X1 X2 X3 Y1 Y2 Y3
1  1  1  1  2  2  2  3  3  3  4  4  4  5  5  5  6  6  6  7  7  7  8  8  8
```

The third arguement, `compid`, represents the study design and hence the comparison pattern. In *simudata2*, `R1,R2,R3` are samples from condition 1 in study 1 and `S1,S2,S3` are from condition 2 in study 1. Simiarly, `T1,T2,T3` represent condition 1 in study 2 and `U1,U2,U3` represent condition 2 in study 2, and so on so forth. We aim at detecting the differential expression pattern of a gene under two different experimental conditions in each study, so we make up the comparison matrix `simu2_compgroup` as following:

```
> data(simu2_compgroup)
> simu2_compgroup
```

```
  Cond1 Cond2
1     1     2
2     3     4
3     5     6
4     7     8
```

3 Model fitting

3.1 No missing data

Once we have specified the group labels and the study design, we are able to fit the *correlation motif* model. We can fit the data with varying motif numbers and use information criterion, such as AIC or BIC, to select the best model. Here for *simudata2*, we fit 5 models with total motif patterns number varying from 1 to 5. And we can see later from the BIC plot, using BIC criterion, the best model is the one with 3 motifs.

```
> motif.fitted<-cormotiffit(exprs.simu2,simu2_groupid,simu2_compgroup,
+                           K=1:5,max.iter=1000,BIC=TRUE)
```

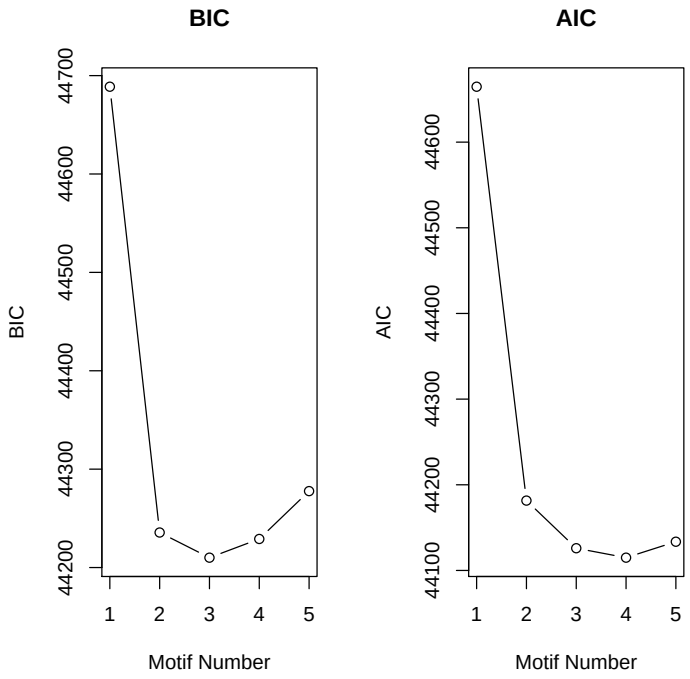
```
[1] "We have run the first 50 iterations for K=2"
[1] "We have run the first 50 iterations for K=3"
[1] "We have run the first 100 iterations for K=3"
[1] "We have run the first 150 iterations for K=3"
[1] "We have run the first 50 iterations for K=4"
[1] "We have run the first 100 iterations for K=4"
[1] "We have run the first 150 iterations for K=4"
[1] "We have run the first 50 iterations for K=5"
[1] "We have run the first 100 iterations for K=5"
[1] "We have run the first 150 iterations for K=5"
[1] "We have run the first 200 iterations for K=5"
[1] "We have run the first 250 iterations for K=5"
[1] "We have run the first 300 iterations for K=5"
```

After fitting the *correlation motif* model, we can check the BIC values obtained by all cluster numbers:

```
> motif.fitted$bic
```

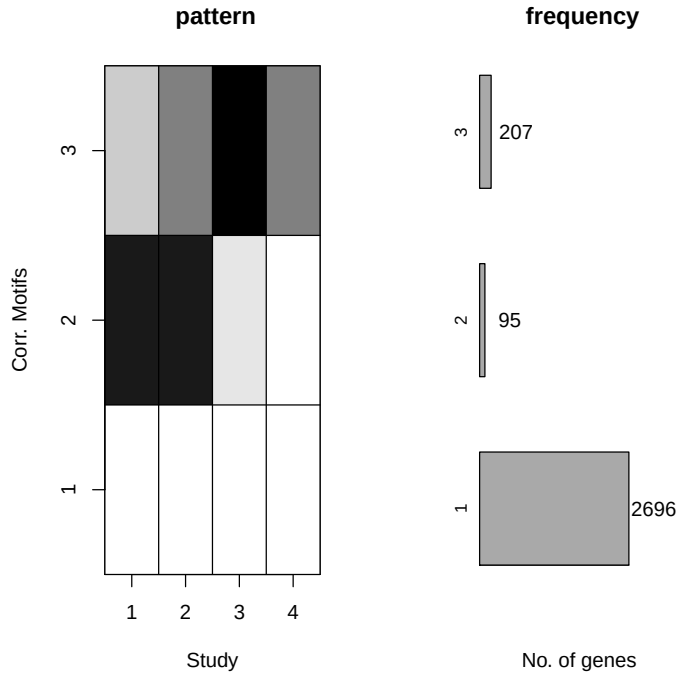
```
     K      bic
[1,] 1 44688.73
[2,] 2 44235.62
[3,] 3 44210.05
[4,] 4 44229.06
[5,] 5 44277.70
```

```
> plotIC(motif.fitted)
```



To picture the motif patterns learned by the algorithm, we can use function `plotMotif`. Each row in both graphs corresponds to the same one motif pattern. We call the left graph *pattern graph* and the right bar chart *frequency graph* . In the pattern graph, each row indicates a motif pattern and each column represents a study. The grey scale of the cell (k, d) demonstrates the probability of differential expression in study d for pattern k, and the values are stored in `motif.fitted$bestmotif$motif.prior`. Each row of the frequency graph corresponds to the motif pattern in the same row of the left pattern graph. The length of the bar in the frequency graph shows the number of genes of the given pattern in the dataset, which is equal to `motif.fitted$bestmotif$motif.prior` multiplying the number of total genes.

```
> plotMotif(motif.fitted)
```



The posterior probability of differential expression for each gene in each study is saved in `motif.fitted$bestmotif$p.post`

```
> head(motif.fitted$bestmotif$p.post)
```

```
           [,1]       [,2]      [,3]      [,4]
[1,] 0.97793112 0.73599635 0.2691709 0.6931932
[2,] 0.99958237 0.31486006 0.9962635 0.9994624
[3,] 0.98275370 0.12177700 0.6968786 0.9984914
[4,] 0.02162931 0.04543982 0.2869920 0.2329653
[5,] 0.99897992 0.93540737 0.9971760 0.5837514
[6,] 0.04468635 0.93294133 0.9977773 0.1020184
```

And at 0.5 cutoff for the posterior distribution, the differential expression pattern can be obtained as following:

```
> dif.pattern.simu2<-(motif.fitted$bestmotif$p.post>0.5)
> head(dif.pattern.simu2)
```

```
      [,1]  [,2]  [,3]  [,4]
[1,]  TRUE  TRUE FALSE  TRUE
```

```
[2,]  TRUE FALSE  TRUE  TRUE
[3,]  TRUE FALSE  TRUE  TRUE
[4,] FALSE FALSE FALSE FALSE
[5,]  TRUE  TRUE  TRUE  TRUE
[6,] FALSE  TRUE  TRUE FALSE
```

We can aslo order the genes in each study according to their posterior probability of differential expression:

```
> topgenelist<-generank(motif.fitted$bestmotif$p.post)
> head(topgenelist)
```

```
     [,1] [,2] [,3] [,4]
[1,]  117  394   59  221
[2,]  196   23   38  238
[3,]   31   97  330  288
[4,]   73  355  355  249
[5,]  177   63  319  286
[6,]  454   62   96   66
```

3.2 With missing data

Cormtoif can handle data with missing values automatically. Especially here we mimic a situtation where data are merged from studies conducted on different platforms, where different platforms have non-overlapping genes. We set the missing proportion to be 10%.

```
> misprop<-0.10
```

We assume the first two studies are conducted in one platform while the third and fourth studies are conducted on another platform. We randomly set 10% of non-overlapping genes in each platform to be missing. Therefore, 10% missing data actually means that 20% of genes are present in only one of the two platforms.

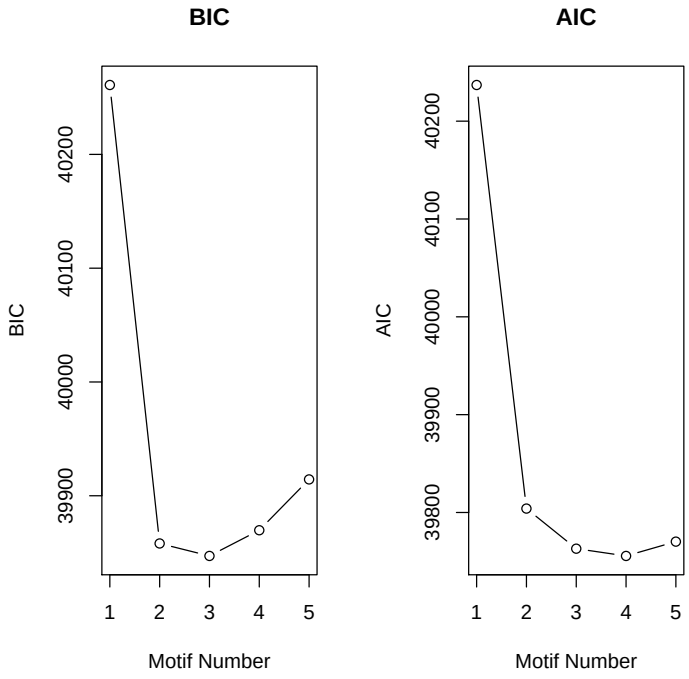
```
> fullindex<-1:nrow(exprs.simu2)
> ##sample index to mimic the merging of studies from different platforms
> mis_index1<-sample(fullindex,misprop*length(fullindex))
> mis_index2<-sample(fullindex[-mis_index1],misprop*length(fullindex))
> exprs.simu2.missing<-exprs.simu2
> exprs.simu2.missing[mis_index1,1:12]<-NA
> exprs.simu2.missing[mis_index2,13:24]<-NA
```

Now we fit the model again on the dataset with missing values and check the learned motifs.

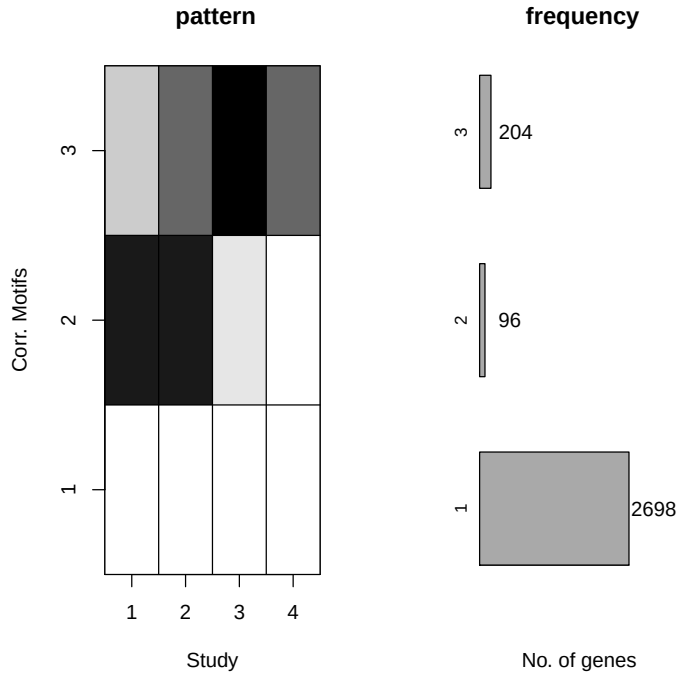
```
> motif.fitted.missing<-cormotiffit(exprs.simu2.missing,simu2_groupid,simu2_compgroup,
+                                   K=1:5,max.iter=1000,BIC=TRUE)
```

```
[1] "We have run the first 50 iterations for K=2"
[1] "We have run the first 50 iterations for K=3"
[1] "We have run the first 100 iterations for K=3"
[1] "We have run the first 50 iterations for K=4"
[1] "We have run the first 100 iterations for K=4"
[1] "We have run the first 150 iterations for K=4"
[1] "We have run the first 200 iterations for K=4"
[1] "We have run the first 50 iterations for K=5"
[1] "We have run the first 100 iterations for K=5"
```

```
> plotIC(motif.fitted.missing)
> plotMotif(motif.fitted.missing)
```



We can see that under 10% missingness our learned motif `motif.fitted.missing` behaves similar to the original `motif.fitted`



From this example, we see that *Cormtoif* is able to deal with data merged from different platforms with non-overlapping genes.

3.3 Other correlation motif fit

The *all motif* method applies a Bayesian model assuming that genes are either differentially expressed in all studies or differentially expressed in none of the studies.

```
> motif.fitted.all<-cormotiffitall(exprs.simu2,simu2_groupid,simu2_compgroup,max.iter=1)
```

The *separate motif* fits the mixture model to each study separately.

```
> motif.fitted.sep<-cormotiffitsep(exprs.simu2,simu2_groupid,simu2_compgroup,max.iter=1)
```

The *full motif* fits all 2^D possible 0-1 motif patterns.

```
> motif.fitted.full<-cormotiffitfull(exprs.simu2,simu2_groupid,simu2_compgroup,max.iter=1)
```

References

[Ji(2011)] Ji, H., Wei, Y. (2011). Correlation Motif. *Unpublished*.

[Smyth 2004] Smyth, G.K. (2004), Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology* 3, Art. 3.