The FEM R package: Identification of Functional Epigenetic Modules

Yinming Jiao and Andrew E. Teschendorff

October 13, 2014

1 Summary

This vignette provides examples of how to use the package FEM to identify interactome hotspots of differential promoter methylation and differential expression, where an inverse association between promoter methylation and gene expression is assumed [1]. By "interactome hotspot" we mean a connected subnetwork of the protein interaction network (PIN) with an exceptionally large average edge-weight density in relation to the rest of the network. The weight edges are constructed from the statistics of association of DNA methylation and gene expression with the phenotype of interest. Thus, the FEM algorithm can be viewed as a functional supervised algorithm, which uses a network of relations between genes (in our case a PPI network) to identify subnetworks where a significant number of genes are associated with a phenotype of interest (POI). We call these "hotspots" also Functional Epigenetic Modules (FEMs). The genome-wide DNA methylation data could have been generated by any platform, but the algorithm has only been tested on Illumina Infinium 27k and 450k data. The FEM algorithm on Illumina 27k data was first presented in [2], with its extension to Illumina 450k data described in [1]. The module detection algorithm used is the spin-glass algorithm of [3]. The PIN used in this vignette includes only protein-protein interactions and derives from Pathway Commons [4], but the user is allowed to specify his own network.
There are three main components to this vignette. These are:

- Application to simulated data.
- Real world example: application to Endometrial Cancer.
- Further details of the algorithm.

2 Application to simulated data

Since FEM is a BioC package, the user first needs to install both R and Bioconductor. R is a free open source software project freely downloadable from the CRAN website *http://cran.r-project.org/*. FEM has several package dependencies, such as igraph, marray, corrplot. So these need to be installed first. For example. to install igraph and corrplot, type *install.packages(c("igraph","corrplot"))*. Since marray is a bioconductor package, we can install it with

source("http://bioconductor.org/biocLite.R");
biocLite("marray").
Load them with

```
> library("igraph");
> library("marray");
> library("corrplot")
```

You can check your version of iGraph by entering *sessionInfo()$otherPkgs$igraph$Version.*
It should be at least version 0.6.
To load the FEM package in your R session,

```
> library("FEM");
```

We demonstrate the functionality of FEM using a simulated toy dataset. To load it into the session, use

```
> data(toydata)#load the toydata
```

In fact, *toydata* is a list and it has four elements.

```
> names(toydata)
```

```
[1] "statM"      "statR"      "adjacency"  "annotation" "tennodes"
```

The first element is *statM*

```
> head(toydata$statM)#"head" function is used to show the first 10 rows of statM
```

```
       t-value      p-value
1 -0.065111202 4.740427e-01
2  0.001161095 4.995368e-01
3  0.195829329 4.223719e-01
4  3.934019816 4.176845e-05
5 -0.025401308 4.898674e-01
6 -0.089637422 4.642877e-01
```

statM is a matrix of t-statistics and p-values of differential methylation (one row for each gene promoter) with rownames annotated with a geneID (here just an index). There are 84 rows because the maximally connected component of the 100-node random graph generated was of size 84. Most of the rows have statistics which have been simulated to be close to zero (between -0.5 and 0.5), meaning that for these there is no association between DNA methylation and the phenotype of interest (POI). There are also ten randomly selected probes whose t-statistics are randomly chosen to lie between 2 and 5, meaning that for these promoters high methylation is associated positively with the POI.
Now lets's check the 10 genes (here just the index) whose t-statistics is larger than 2, meaning that for these 10 genes there is a significant positive association between methylation and the phenotype of interest (POI).

```
> as.vector(which(toydata$statM[,1]>2))->tennodes;
> tennodes;
```

```
 [1]  4 11 19 30 37 49 64 77 79 80
```

The second member is *statR*

```
> head(toydata$statR)
```

```
      t-value     p-value
1 -0.09591858 0.461792617
2 -0.03296763 0.486850201
3  0.17791827 0.429393580
4 -2.57586127 0.004999538
5 -0.12861502 0.448831142
6 -0.03371232 0.486553276
```

statR is a matrix of t-statistics and p-values of differential mRNA expression (same dimension as statM.m and ordered in same way) with rownames annotated with the same index gene ID. The t-statistics of the previously selected 10 genes are set to lie between -2 and -5. Thus, this models the case of ten promoters which are underexpressed due to hypermethylation.

The following command identifies the index positions of genes that are significantly underexpressed. They agree with those that are hypermethylated:

```
> as.vector(which(toydata$statR[,1]< -2));
```

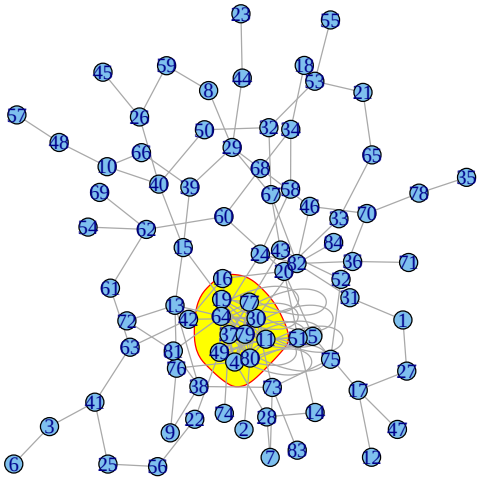
```
 [1]  4 11 19 30 37 49 64 77 79 80
```

The third member is *adjacency*. This is an adjacency matrix with number of rows and columns equal to rows of statR (or statM), ordered in same way and with same gene identifier. The resulting graph is assumed to be connected, and was constructed as the maximally connected subgraph of a 100-node random graph (Erdos-Renyi graph). Specifically, the original random graph was generated with erdos.renyi.game(100, 2/100) and then the solitary nodes were removed, resulting in a connected network of 84 nodes/genes.

Since 10 of these nodes represent differentially methylated and differentially expressed genes, in order to generate a module centred on them, we turned the 10 nodes into a clique, meaning that each of these ten nodes is connected to each other. So they belong to a deliberately created module with high absolute differential methylation and diffential expression t-statistics. They are indicated in the following plot.

Plot this network from *adjacency* with the ten nodes marked as a group.

```
> plot.igraph(graph.adjacency(toydata$adjacency,mod="undirected"),
+ vertex.size=8,mark.groups=toydata$tennodes,mark.col="yellow")
```



In this graph, the ten genes are set as one group, they are connected to each other, their DNA methylation t-statistics values lie between 2 and 5, meaning that for these promoters high methylation is associated positively with the POI and their t-statistics in statR are set to between -2 and -5. Thus, these 10 genes are under-expressed due to hypermethylation. The fourth member of the list is *annotation*, an annotation matrix:

```
> head(toydata$annotation)
```

```
     EentrezID GeneSymbol
[1,] "1"       "Gene1"
[2,] "2"       "Gene2"
[3,] "3"       "Gene3"
[4,] "4"       "Gene4"
[5,] "5"       "Gene5"
[6,] "6"       "Gene6"
```

First column should contain the unique (entrez gene) identifiers as used in adj.m, statM.m and statR.m. Second column should contain the corresponding unique gene symbols. In this toy network case, we use just use integer indices as trivial identifiers.

As mentioned before, FEM is use to detect interactome hotspots of differential promoter methylation and differential expression, where an inverse association between promoter methylation and gene expression is assumed. So let us test whether FEM can detect the simulated module of ten nodes:

We now use DoFEMbi() to find the community structures induced by these phenotype changes in the toydata.

```
> DoFEM_test.o  <- with(toydata,
+                              DoFEMbi(statM,statR,adjacency,
+                              nseeds=1,gamma=0.5,nMC=1000,
+                              sizeR.v=c(1,100),minsizeOUT=10,
+                              writeOUT=TRUE,
+                              nameSTUDY="TEST",ew.v=NULL));
```

The other arguments are :

nseeds: number of seeds/modules to search for. This should be a number such that P-values of significance after multiple testing is less than some reasonable FDR threshold, i.e. < 0.3. Here we set it as one to save time.

gamma: tuning parameter of spin-glass algorithm. Default value generally leads to modules in the desired size range (10-100).

nMC: number of Monte Carlo runs for establishing statistical significance of modularity values under randomisation of the molecular profiles on the network.

You can also use "sizeR.v" to set the desired size range for modules, "minsizeOUT" to set the minimum size of modules to report as interesting and "writeOUT" to indicate whether to write out tables in text format.

There are two result files. The colums are

- size: a vector of inferred module sizes for each of the ntop seeds.
- mod: a vector of associated modularities.
- pv: a vector of associated significance P-values (with resolution of nMC runs).
- selmod: index positions of significant modules of size at least minsizeOUT and smaller than the maximum specified in sizeR.v
- fem: a summary matrix of the selected modules.
- topmod: a list of summary matrices for each of the selected module

Let's check the details of the output:

```
> DoFEM_test.o$fem
```

```
     EntrezID(Seed) Symbol(Seed) Size Mod                P
[1,] "11"           "Gene11"     "36" "4.40228923792317" "0"
     Genes
[1,] "Gene11 Gene4 Gene5 Gene19 Gene30 Gene37 Gene49 Gene51 Gene64 Gene77 Gene79 Gene80 Ge
```

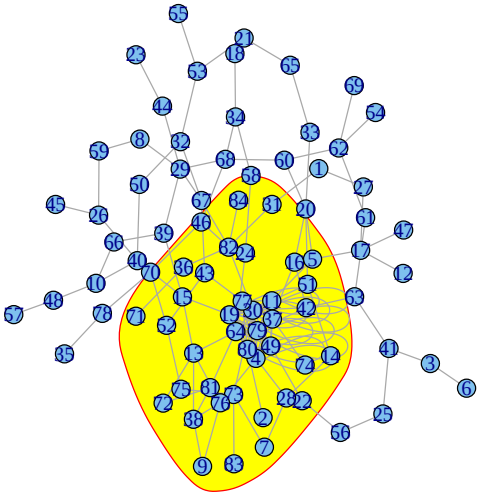
We can see that we get one module with seed "Gene11" (hence its module name "Gene11") and it has 36 members. With the following command, we can have a look at the details of the first five genes in the "Gene11" module:

```
> head(DoFEM_test.o$topmod$Gene11,n=5L)
```

```
   EntrezID Symbol   stat(DNAm)             P(DNAm)
11 "11"     "Gene11" "3.98617629287764"    "3.35733147288641e-05"
4  "4"      "Gene4"  "3.93401981610805"    "4.17684458448805e-05"
5  "5"      "Gene5"  "-0.0254013077355921" "0.489867434011738"
19 "19"     "Gene19" "3.49410701170564"    "0.00023782515959272"
30 "30"     "Gene30" "2.18818748369813"    "0.0143279742210821"
   stat(mRNA)           P(mRNA)                  stat(Int)
11 "-3.05292705958709"  "0.00113310507560836"  "7.29480934183558"
4  "-2.57586126867682"  "0.00499953781149238"  "6.72562916567027"
5  "-0.128615015652031" "0.44883114211167"     "0"
19 "-3.28854257101193"  "0.00050353776488235"  "7.05809017310707"
30 "-3.31781403906643"  "0.000453624323520969" "5.78389382246553"
```

To indicate the "Gene11" module in the whole network, use:

```
> mod.idx<-as.numeric(DoFEM_test.o$topmod$Gene11[,1])
> plot.igraph(graph.adjacency(toydata$adjacency,mod="undirected"),
+ vertex.size=8,mark.groups=mod.idx,mark.col="yellow")
```



In order to check the effectiveness of FEM in detecting the true module, we compute the sensitivity, defined as the fraction of the truly associated genes that are captured by the inferred module:

```
> sensitivity=length(intersect(tennodes,mod.idx))/length(tennodes);
> sensitivity
```

```
[1] 1
```

Thus, we find that the sensitivity is 100%, meaning that FEM inferred a module that contained all 10 genes that were truly differentially methylated and expressed. Although, specificity is not 100%, this is to be expected since a larger module can still exhibit a higher than random average weight density.

3 A real world example: application to Endometrial Cancer

To validate the FEM algorithm on real Illumina 450k data we collected and analyzed 118 endometrial cancers and 17 normal endometrial samples, all with matched RNA-Seq data from the TCGA study [5]. To assign DNA methylation values to a given gene, in the case of Illumina 27k data, we assigned the probe value closest to the transcription start site (TSS). In the case of Illumina 450k data, we assigned to a gene, the average value of probes mapping to within 200bp of the TSS. If no probes map to within 200bp of the TSS, we use the average of probes mapping to the 1st exon of the gene. If such probes are also not present, we use the average of probes mapping to within 1500bp of the TSS. Justification for this procedure is provided in our Bioinformatics paper [1]. For each gene g in the maximally connected subnetwork, we then derive a statistic of association between its DNA methylation profile and the POI (here normal/cancer status), denoted by $t_g^{(D)}$ as well as between its mRNA expression profile and the same POI, which we denote by $t_g^{(R)}$. These statistics have already been computed beforehand using the limma package. We now load them in:

```
> data(realdata);
> attributes(realdata);

$names
[1] "statM"      "statR"      "adjacency"  "annotation"
```

Since running DoFEMbi on large data sets can be lengthy (it takes 23 minutes on a 4 3GHz CPU-core Dell Workstation with 16GB memory), we comment the following line out:

```
> #fembi.o <- DoFEMbi(realdata$statM,realdata$statR,realdata$adjacency,
> #nseeds=100,gamma=0.5,nMC=1000,sizeR.v=c(1,100),minsizeOUT=10,
> #writeOUT=TRUE,nameSTUDY="TCGA-EC",ew.v=NULL);
```

To load in the results use:

```
> data(fembi.o)
```

The algorithm predicts a number of FEM modules. Use the folloing command to display their size and elements.

```
> fembi.o$fem
```

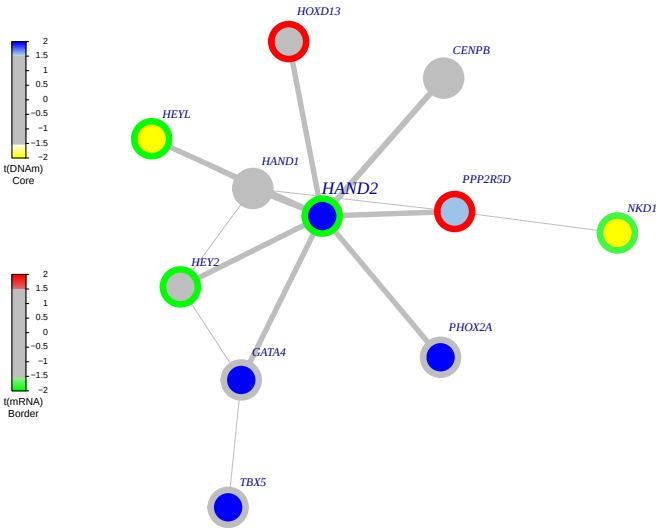
The details of the modules can also be seen using:

```
> fembi.o$topmod
```

In order to illustrate the modules graphically, the user can invoke the function *FemModShow*, which will generate a pdf figure of the module in your working directory and also return an *graphNEL* object which includes methylation and expression color schemes. The *graphNEL* class is defined in the bioconductor graph package, user can use *igraph.from.graphNEL* to convert it to the igraph objects. For instance, the algorithm inferred a module centred around the gene *HAND2*, which has been demonstrated to be causally implicated in the development of endometrial cancer [2]. Thus, given its importance, we generate a detailed network plot of this module:

```
> library("marray");
> library("corrplot");
> HAND2.mod<-fembi.o$topmod$HAND2;
> HAND2.graphNEL.o=FemModShow(fembi.o$topmod$HAND2,name="HAND2",fembi.o$ew,realdata$adjace
```

now the HAND2 module is as following;



Depicted is the functional epigenetic module centred around seed gene HAND2. Edge widths are proportional to the average statistic of the genes making up the edge. Node colours denote the differential DNA methylation statistics as indicated. Border colors denote the differential expression statistics. Observe that despite many nodes exhibiting differential methylation and differential expression, only HAND2, exhibits the expected anticorrelation with hypermethylation (blue) leading to underexpression (green). See Jones et al [2] for the functional, biological and clinical significance of *HAND2* in endometrial cancer.

The user can generate all the modules' graphs by

```
> #for(m in 1:length(names(fembi.o$topmod))){FemModShow(fembi.o$topmod[[m]],
> #name=names(fembi.o$topmod)[m],fembi.o$ew,realdata$adjacency)}
```

In the above examples we provided the statistics and adjacency matrix input objects. However, below we describe a function *DoIntFEM450k* which can generate the integrated statM, statR and adjacency matrices from the actual DNA methylation, gene expression and original network adjacency matrices:
The input arguments are:

- dnaM.m: normalised DNA methylation 450k data matrix, with rownames annotated to 450k probe IDs.
- phenoM.v: phenotype vector corresponding to dnaM.m. So if there are two phenotypes, 1 and 2 say, the contrast is "2 minus 1".
- exp.m: normalized gene expression data matrix with rownames annotated to NCBI Entrez gene IDs. If the mapped Entrez gene IDs are not unique, we use the average value of the same Entrez gene ID as the expresssion value.
- phenoR.v: phenotype vector corresponding to exp.m
- adj.m: adjacency matrix of a network of relations (e.g. PPI network) with rownames/colnames annotated to NCBI Entrez gene IDs.

Note: The PPI network can be derived from the Pathway Commons resource [4] and follows the procedure described in [6]. The PIN used in previous papers is available at http://sourceforge.net/projects/funepimod. The PPI network consists of 8434 genes annotated to NCBI Entrez identifiers, and is sparse containing 303600 documented interactions (edges). If the user wishes they can use a different PPI network or generate statR using a different method.
The usage of DoIntFEM450k is

```
> #IntFEM450k.o=DoIntFEM450k(dnaM.m,phenoM.v,exp.m,phenoR.v,adj.m)
```

The output IntFEM450k.o has 5 members:

- statR: matrix of gene expression moderated t-statistics and P-values for those genes in the integrated network
- statM: matrix of DNA methylation moderated t-statistics and P-values for those genes in the integrated network
- adj: adjacency matrix of the maximally connected integrated network. At present only maximally connected subnetwork is used.
- avexp: expression data matrix mapped to unique Entrez IDs
- avbeta: DNA methylation data matrix mapped to unique Entrez IDs

4 EpiMod and ExpMod

It may be that we only wish to infer either differential methylation or differential expression interactome hotspots. To this end, we provide specific functions, i.e. *DoIntEpi450K* to do the integration at the DNA methylation level only. Indeed, *DoIntEpi450K* is the same as *DoIntFEM450k*, except that we do not need the *exp.m, phenoR.v* arguments. In this case, the edge weights in the interactome network reflect the combined differential methylation statistics (absolute values) of the genes making up the edge.

DoIntEpi450K input arguments:

- dnaM.m: normalised DNA methylation 450k data matrix, with rownames annotated to 450k probe IDs.
- phenoM.v: phenotype vector corresponding to dnaM.m. So if there are two phenotype, 1 and 2, the contrast is 2 minus 1.
- adj.m: adjacency matrix of a network of relations (e.g. PPI network) with rownames/colnames annotated to NCBI Entrez gene IDs.

The output arguments:

- statM: matrix of DNA methylation moderated t-statistics and P-values for those genes in the integrated network
- adj: adjacency matrix of the maximally connected integrated network. At present only maximally connected subnetwork is used.
- annotation: The annotation data matrix with EntrezID and GeneSymbol
- avbeta: DNA methylation data matrix mapped to unique Entrez IDs

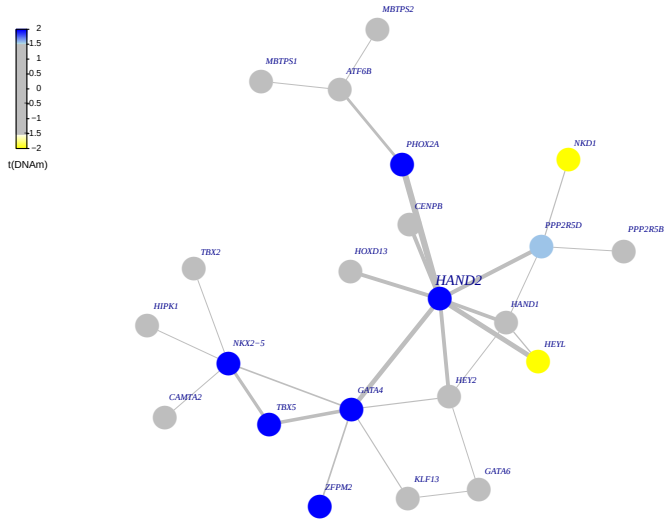
We also provide a special function, *DoEpiMod*, to process the statM, adj, annotation matrix generated by *DoIntEpi450K*. The usage of *DoEpiMod* is same as *DoFEMbi* except that the statR argument is not needed for *DoIntEpi450K*.

Once we have run *DoEpiMod*, we can use *FemModShow* to show the top modules. The usage and arguments of *FemModShow* for EpiMod is same as previous decribed. You just need to add an argument "mode" and set "mode" = "Epi" as *FemModShow* has three modes, the default one being "integration", which is the one to use with *DoFEMbi*. The "Epi" mode means *FemModShow* will render the Epi-modules generated by *DoEpiMod*.

The workflow applying DoIntEpi450k,DoEpiMod and FemModShow is:

```
> #IntEpi450k.o=DoIntEpi450k(dnaM.m,phenoM.v,adj.m)
> #EpiMod.o=DoEpiMod(IntEpi450k.o$statM,IntEpi450k.o$adjacency,
> #                   nseeds=100,gamma=0.5,nMC=1000,
> #                   sizeR.v=c(1,100),minsizeOUT=10,
> #                  writeOUT=TRUE,nameSTUDY="TCGA-EC",ew.v=NULL);)
> #
> #HAND2.mod<-EpiMod.o$topmod$HAND2; # if there is also HAND2 module
> #HAND2.mod.igraph.o=FemModShow(EpiMod.o$topmod$HAND2,name="HAND2",
> #EpiMod.o$ew,IntEpi450k.o$adjacency,mode="Epi")
```

The DoIntEpi450k example data will be available later in an experimental package. In this case, application to the 450K methylation data of 118 endometrial cancers and 17 normal endometrials from TCGA, results in a module, also centred around *HAND2*:



Depicted is the HAND2 Epi-module which contains many interacting members, most of which are hypermethylated in cancer compared to normal tissue:

If we are interested in inferring differential mRNA expression hotspots, you should run *DoExpMod*. As before, first you should run *DoIntExp* function to integrate the gene expression matrix with the network adjacency matrix and to compute the statistics of differential expression: The input arguments to *DoIntExp* would be:

- exp.m: normalized gene expression data matrix with rownames annotated to NCBI Entrez gene IDs. If the mapped Entrez gene IDs are not unique, we use the average value of the same Entrez gene ID as the expresssion value.
- phenoR.v: phenotype vector corresponding to exp.m. So if there are two phenotype, 1 and 2, the contrast is 2 minus 1.
- adj.m: adjacency matrix of a network of relations (e.g. PPI network) with rownames/colnames annotated to NCBI Entrez gene IDs.

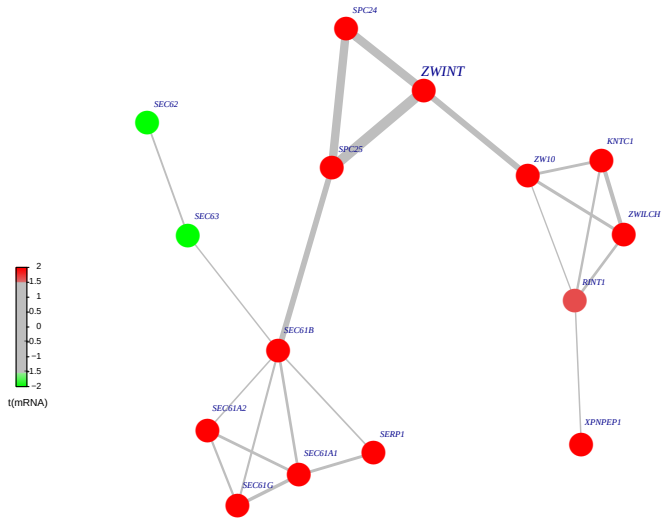
The output arguments are:

- statR: matrix of gene expression moderated t-statistics and P-values for those genes in the integrated network
- adj: adjacency matrix of the maximally connected integrated network. At present only maximally connected subnetwork is used.
- annotation: The annotation data matrix with EntrezID and GeneSymbol
- avexp: expression data matrix mapped to unique Entrez IDs

Subsequently, we can apply *DoExpMod* function to infer modules of differential expression. The usage *DoExpMod* is same as *DoFEMbi* except that *DoExpMod* doesn't need the *statM* argument.
Finally, if you want to render the top differential expression modules, you run *FemModShow* with the argument "mode" set as "Exp".

The workflow applying DoIntExp, DoExpMod and FemModShow is:

```
> #IntExp.o=DoIntExp(dnaR.m,phenoR.v,adj.m)
> #ExpMod.o=DoEpiMod(IntExp.o$statM,IntExp.o$adjacency,
> #                    nseeds=100,gamma=0.5,nMC=1000,
> #                    sizeR.v=c(1,100),minsizeOUT=10,
> #                   writeOUT=TRUE,nameSTUDY="TCGA-EC",ew.v=NULL);)
> #
> #ExpMod1.mod<-ExpMod.o$topmod$ExpMod1; # if there is a ExpMod1 module
> #Exp1Mod1.mod.graphNEL.o=FemModShow(Exp1Mod1.o$topmod$ExpMod1,name="ExpMod1",
> #EpiMod.o$ew,IntExp.o$adjacency,mode="Exp")
```



There is one ExpMod example, a ZWINT-Centered Interactome Hotspot. In this case, application to the expression data of 118 endometrial cancers and 17 normal endometrials from TCGA, results in a module, centered around ZWINT, which is a known component of the kinetochore complex required for the mitotic spindle checkpoint and thus regulates the cell proliferation.

5 Integration with "minfi" package

The functions DoIntFEM450k or DoIntEpi450k use the normalised DNA methylation 450k data matrix with rownames annotated to 450k probe IDs. This kind of normalized 450k data matrix can be generated from raw data by many different tools. For instance, one could use *minfi*, an existing Bioconductor package [7]. The simplified workflow would be something like:

```
> #library(minfi);
> #require(IlluminaHumanMethylation450kmanifest);
>
> #baseDIR <- getwd();# the base dir of the Rawdata
> #setwd(baseDIR);
> #targets <- read.450k.sheet(baseDIR);#read the csv file.
> #RGset <- read.450k.exp(baseDIR); #Reads an entire 450k experiment
> #                                 using a sample sheet
> #MSet.raw <- preprocessRaw(RGset);#Converts the Red/Green channel for an Illumina
> #                                 methylation array into methylation signal,
> #                                 without using any normalization
> #beta.m <- getBeta(MSet.raw,type = "Illumina");# get normalized beta
> #pval.m <- detectionP(RGset,type ="m+u")
```

Before passing on the beta.m object to DoIntFEM450k or DoIntEpi450k, we recommend adjusting the data for the type-2 probe bias, using for instance BMIQ [8]. BMIQ is freely available from either http://code.google.com/p/bmiq/, or the ChAMP Bioconductor package [9].

References

[1] Jiao Y, Widschwendter M, Teschendorff AE (2014) A systems-level integrative framework for genome-wide dna methylation and gene expression data identifies functional gene modules under epigenetic control. Bioinformatics (2014)doi: 101093/bioinformatics/btu316 .

[2] Jones A, Teschendorff AE, Li Q, Hayward JD, Kannan A, et al. (2013) Role of dna methylation and epigenetic silencing of hand2 in endometrial cancer development. PLoS Med 10:e1001551.

[3] Reichardt J, Bornholdt S (2006) Statistical mechanics of community detection. Phys Rev E 74:016110. doi:10.1103/PhysRevE.74.016110. URL http://link.aps.org/doi/10.1103/PhysRevE.74.016110.

[4] Cerami, G E, Gross, E B, Demir, et al. (2011) Pathway commons, a web resource for biological pathway data. Pathway commons, a web resource for biological pathway data 39(Database):D685–D690.

[5] Kandoth C, Schultz N, Cherniack AD, Akbani R, Liu Y, et al. (2013) Integrated genomic characterization of endometrial carcinoma. Nature 497:67–73.

[6] West J, Beck S, Wang X, Teschendorff AE (2013) An integrative network algorithm identifies age-associated differential methylation interactome hotspots targeting stem-cell differentiation pathways. Sci Rep 3:1630.

[7] Aryee MJ, Jaffe AE, Corrada-Bravo H, Ladd-Acosta C, Feinberg AP, et al. (2014) Minfi: a flexible and comprehensive bioconductor package for the analysis of infinium dna methylation microarrays. Bioinformatics 30:1363–9.

[8] Teschendorff AE, Marabita F, Lechner M, Bartlett T, Tegner J, et al. (2013) A beta-mixture quantile normalization method for correcting probe design bias in illumina infinium 450 k dna methylation data. Bioinfomatics 29:189–196.

[9] Morris TJ, Butcher LM, Feber A, Teschendorff AE, Chakravarthy AR, et al. (2014) Champ: 450k chip analysis methylation pipeline. Bioinformatics 30:428–430.