ELMER: An R/Bioconductor Tool Inferring Regulatory Element Landscapes and Transcription Factor Networks Using Cancer Methylomes

Lijing Yao

March 13, 2015

need to be done

Contents

1 Introduction

ELMER This document provide an introduction of the *ELMER*, which provides functions for identifying differential DNA methylation probes at distal regulatory regions, genes whose expressions associate with differential DNA methylation

probes and regulatory TF in the regulatory network. It is designed to use a large number of samples' DNA methylation data (HM450K) and gene expression data to inferring regulatory element landscapes and transcription factor network from primary tissues.

Need add more

1.1 Benchmark

need to write

1.2 Example data

Follow the following step to obtain a example data for *ELMER*

```
> #Example file download from URL: https://dl.dropboxusercontent.com/u/61961845/ELMER.example.tar.gz
> URL <- "https://dl.dropboxusercontent.com/u/61961845/ELMER.example.tar.gz"
> download.file(URL,destfile = "ELMER.example",method= "wget",
+               extra = c("--no-check-certificate -a download.log"))
> untar("./ELMER.example")
> library(ELMER)
```

2 Installing and Loading ELMER

need to write.

3 Input data

For running whole pipeline analysis in *ELMER*, at least 4 input files are required. A matrix of DNA methylation, a matrix of gene expression for the same samples, a GRanges object contains information such as names and coordinates for the probe on the HM450K and gene annotation GRanges object. For TCGA data, samples information will be automatically generated. But samples information should be provided when using custom data.

3.1 DNA methylation data

Raw DNA methylation data can be processed by *Methylumi* or *minfi* (ref) generating DNA methylation level for each CpG. The DNA methylation level at each CpG is referred to as a beta (β) value, calculated as (M/(M+U)), where M represents the methylated allele intensity and U the unmethylated allele intensity. Beta values range from 0 to 1, reflecting the fraction of methylated alleles at each CpG in the each tumor; beta values close to 0 indicating low levels of DNA methylation and beta values close to 1 indicating high levels of DNA methylation. Then generate DNA methylation beta value matrix for all the samples (columns) and probe loci (rows) and save matrix as meth.rda

```
> load("./ELMER.example/Result/LUSC/LUSC_meth_refined.rda")
> str(Meth)
```

```
 num [1:1725, 1:268] 0.819 0.842 0.91 0.875 0.333 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:1725] "cg00045114" "cg00050294" "cg00066722" "cg00093522" ...
  ..$ : chr [1:268] "TCGA-43-3394-11A-01D-1551-05" "TCGA-43-3920-11B-01D-1551-05" "TCGA-56-8305-01A-11D-229
```

3.2 Gene expression data

Gene expression value can be generated from different platform such as array or RNA-seq, gene level or transcripts level gene expression. Generate Gene expression matrix for all the samples (columns) and gene (rows) and save matrix as RNA.rda

```
> load("./ELMER.example/Result/LUSC/LUSC_RNA_refined.rda")
> str(GeneExp)
```

```
 num [1:3898, 1:234] 0 0.43 10.08 6.45 8.59 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:3898] "ID126767" "ID343066" "ID26574" "ID24" ...
  ..$ : chr [1:234] "TCGA-22-5472-01A-01R-1635-07" "TCGA-22-5489-01A-01R-1635-07" "TCGA-22-5491-11A-01R-185
```

3.3 Sample information

Sample Information should be stored as data.frame object contains sample ID, groups label (such as tumor, normal) and other description for each samples. For TCGA data, tumor, normal group label will be automatically generated by fetch.mee function by specify option TCGA=TRUE.

```
> mee <- fetch.mee(meth=Meth, exp=GeneExp, TCGA=T)
```

```
~~~ MEE.data: initializator ~~~
```

```
> head(getSample(mee))
```

```
                             ID                      meth.ID                       exp.ID
TCGA-43-3394-11 TCGA-43-3394-11 TCGA-43-3394-11A-01D-1551-05 TCGA-43-3394-11A-01R-1758-07
TCGA-56-8305-01 TCGA-56-8305-01 TCGA-56-8305-01A-11D-2294-05 TCGA-56-8305-01A-11R-2296-07
TCGA-56-8307-01 TCGA-56-8307-01 TCGA-56-8307-01A-11D-2294-05 TCGA-56-8307-01A-11R-2296-07
TCGA-56-8308-01 TCGA-56-8308-01 TCGA-56-8308-01A-11D-2294-05 TCGA-56-8308-01A-11R-2296-07
TCGA-56-8309-01 TCGA-56-8309-01 TCGA-56-8309-01A-11D-2294-05 TCGA-56-8309-01A-11R-2296-07
TCGA-58-8386-01 TCGA-58-8386-01 TCGA-58-8386-01A-11D-2294-05 TCGA-58-8386-01A-11R-2296-07
                    TN
TCGA-43-3394-11 Normal
TCGA-56-8305-01  Tumor
TCGA-56-8307-01  Tumor
TCGA-56-8308-01  Tumor
TCGA-56-8309-01  Tumor
TCGA-58-8386-01  Tumor
```

3.4 Probe information

Probe information should be stored as GRanges object containing coordinate of each probe on the DNA methylation array and names of each probe. The default probe information is for HM450K.

```
> probe <- ReadBed(system.file("extdata","Illumina-methyl-450K-manifest.hg19.bed.xz",
+                              package = "ELMER"))
> mee <- fetch.mee(meth=Meth, exp=GeneExp, TCGA=T, probeInfo=probe)
```

```
~~~ MEE.data: initializator ~~~
```

```
> getProbeInfo(mee)
```

```
GRanges object with 1725 ranges and 1 metadata column:
         seqnames                ranges strand   |            name
            <Rle>             <IRanges>  <Rle>   |     <character>
```

```
     [1]     chr1 [172674159, 172674159]      *   |      cg00045114
     [2]     chr1 [  2886818,   2886818]      *   |      cg00050294
     [3]     chr1 [ 43634520,  43634520]      *   |      cg00066722
     [4]     chr1 [  2252019,   2252019]      *   |      cg00093522
     [5]     chr1 [ 16465562,  16465562]      *   |      cg00107046
     ...      ...                   ...    ... ...             ...
  [1721]     chr1 [215147891, 215147891]      *   |      cg27589988
  [1722]     chr1 [ 46632696,  46632696]      *   |      cg27636310
  [1723]     chr1 [  3472204,   3472204]      *   |      cg27637706
  [1724]     chr1 [  3283394,   3283394]      *   |    ch.1.131529R
  [1725]     chr1 [174947362, 174947362]      *   | ch.1.173213985R
  -------
  seqinfo: 24 sequences from an unspecified genome; no seqlengths
```

3.5 Gene information

Gene information should be stored as GRanges object containing coordinates of each gene, gene id, gene symbol and gene isoform id. The default gene information is UCSC gene annotation.

```
> load(system.file("extdata","UCSC_gene_hg19.rda",package = "ELMER"))
> ## For TCGA data number couldn't be as rownames, so "ID" was added to each gene id.
> ## If your geneID is consistent with rownames of gene expression matrix, this step
> ## can be skipped.
> txs$GENEID <- paste0("ID",txs$GENEID)
> geneInfo <- promoters(txs,upstream = 0, downstream = 0)
> save(geneInfo,file="./ELMER.example/Result/LUSC/geneAnnot.rda")
> mee <- fetch.mee(meth=Meth, exp=GeneExp, TCGA=T, geneInfo=txs)
```

```
~~~ MEE.data: initializator ~~~
```

```
> getGeneInfo(mee)
```

```
GRanges object with 13741 ranges and 3 metadata columns:
        seqnames                ranges strand   |     tx_name      GENEID      SYMBOL
           <Rle>             <IRanges>  <Rle>   | <character> <character> <character>
  10000     chr1 [243651535, 244006584]      -   |  uc001hzz.1     ID10000        AKT3
  10000     chr1 [243663021, 244006584]      -   |  uc001iab.2     ID10000        AKT3
  10000     chr1 [243663021, 244006886]      -   |  uc021plu.1     ID10000        AKT3
  10001    chr14 [ 71050957,  71067384]      -   |  uc001xmf.3     ID10001        MED6
  10001    chr14 [ 71050957,  71067384]      -   |  uc010tth.2     ID10001        MED6
    ...      ...                   ...    ... ...         ...         ...         ...
   9988     chr7   [86781677, 86825648]      +   |  uc003uik.3      ID9988       DMTF1
   9988     chr7   [86781677, 86825648]      +   |  uc011khb.2      ID9988       DMTF1
   9988     chr7   [86781870, 86825648]      +   |  uc003uil.3      ID9988       DMTF1
   9988     chr7   [86792198, 86809018]      +   |  uc003uim.1      ID9988       DMTF1
   9988     chr7   [86792198, 86825648]      +   |  uc003uin.3      ID9988       DMTF1
  -------
  seqinfo: 93 sequences (1 circular) from hg19 genome
```

3.6 MEE.data

The above 5 components will form MEE.data object as input for mulitple functions.

```
> mee <- fetch.mee(meth=Meth, exp=GeneExp, TCGA=T, probeInfo=probe, geneInfo=txs)
```

```
~~~ MEE.data: initializator ~~~
```

```
> mee
```

```
*** Class MEE.data, method show ***
* meth
 num [1:1725, 1:234] 0.819 0.842 0.91 0.875 0.333 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:1725] "cg00045114" "cg00050294" "cg00066722" "cg00093522" ...
  ..$ : chr [1:234] "TCGA-43-3394-11A-01D-1551-05" "TCGA-56-8305-01A-11D-2294-05" "TCGA-56-8307-01A-11D-229
NULL
* exp
 num [1:3894, 1:234] 0 0.214 10.048 5.007 8.63 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:3894] "ID126767" "ID343066" "ID26574" "ID24" ...
  ..$ : chr [1:234] "TCGA-43-3394-11A-01R-1758-07" "TCGA-56-8305-01A-11R-2296-07" "TCGA-56-8307-01A-11R-229
NULL
* sample
'data.frame':        234 obs. of  4 variables:
 $ ID     : chr  "TCGA-43-3394-11" "TCGA-56-8305-01" "TCGA-56-8307-01" "TCGA-56-8308-01" ...
 $ meth.ID: chr  "TCGA-43-3394-11A-01D-1551-05" "TCGA-56-8305-01A-11D-2294-05" "TCGA-56-8307-01A-11D-2294-0
 $ exp.ID : chr  "TCGA-43-3394-11A-01R-1758-07" "TCGA-56-8305-01A-11R-2296-07" "TCGA-56-8307-01A-11R-2296-0
 $ TN     : chr  "Normal" "Tumor" "Tumor" "Tumor" ...
NULL
* probeInfo
GRanges object with 1725 ranges and 1 metadata column:
         seqnames                ranges strand   |            name
            <Rle>             <IRanges>  <Rle>   |     <character>
     [1]     chr1 [172674159, 172674159]      *   |      cg00045114
     [2]     chr1 [  2886818,   2886818]      *   |      cg00050294
     [3]     chr1 [ 43634520,  43634520]      *   |      cg00066722
     [4]     chr1 [  2252019,   2252019]      *   |      cg00093522
     [5]     chr1 [ 16465562,  16465562]      *   |      cg00107046
     ...      ...                   ...    ... ...             ...
  [1721]     chr1 [215147891, 215147891]      *   |      cg27589988
  [1722]     chr1 [ 46632696,  46632696]      *   |      cg27636310
  [1723]     chr1 [  3472204,   3472204]      *   |      cg27637706
  [1724]     chr1 [  3283394,   3283394]      *   |    ch.1.131529R
  [1725]     chr1 [174947362, 174947362]      *   | ch.1.173213985R
  -------
  seqinfo: 24 sequences from an unspecified genome; no seqlengths
* geneInfo
GRanges object with 13741 ranges and 3 metadata columns:
        seqnames                ranges strand   |     tx_name      GENEID      SYMBOL
           <Rle>             <IRanges>  <Rle>   | <character> <character> <character>
  10000     chr1 [243651535, 244006584]      -   |  uc001hzz.1     ID10000        AKT3
  10000     chr1 [243663021, 244006584]      -   |  uc001iab.2     ID10000        AKT3
  10000     chr1 [243663021, 244006886]      -   |  uc021plu.1     ID10000        AKT3
  10001    chr14 [ 71050957,  71067384]      -   |  uc001xmf.3     ID10001        MED6
  10001    chr14 [ 71050957,  71067384]      -   |  uc010tth.2     ID10001        MED6
    ...      ...                   ...    ... ...         ...         ...         ...
   9988     chr7   [86781677, 86825648]      +   |  uc003uik.3      ID9988       DMTF1
   9988     chr7   [86781677, 86825648]      +   |  uc011khb.2      ID9988       DMTF1
   9988     chr7   [86781870, 86825648]      +   |  uc003uil.3      ID9988       DMTF1
   9988     chr7   [86792198, 86809018]      +   |  uc003uim.1      ID9988       DMTF1
```

```
   9988     chr7   [86792198, 86825648]      +   |  uc003uin.3      ID9988       DMTF1
  -------
  seqinfo: 93 sequences (1 circular) from hg19 genome
******* End Print (MEE.data) *******
```

4 Illustration of ELMER analysis

A subset of chromosome 1 data from TCGA LUSC were used as illustruation.

4.1 Selection of probes within biofeatures

Function, get.feature.probe, is used to select probes that locates within the biofeatures such as H3K27ac ChIP-seq peaks. As default, get.feature.probe function will automatically select distal enhancer probes on HM450K which are at least 2kb away from TSS annotated by GENCODE V15 and UCSC gene and located within the putative comprehensive enhancer from REMC, ENCODE and FANTOM5.

```
> #get distal enhancer probe that 2kb away from TSS and overlap with REMC and FANTOM5
> #enhancers on chromosome 1
> Probe <- get.feature.probe(probe=probe, rm.chr=paste0("chr",c(2:22,"X","Y")))
> save(Probe,file="./ELMER.example/Result/LUSC/probeInfo_feature.rda")
```

4.2 Identifying differentially methylated probes

Function,get.diff.meth, will be used to identify differentially methylated probes among the probes within biofeatures which are selected in the above step.

```
> ## fetch.mee can take path.
> mee <- fetch.mee(meth="./ELMER.example/Result/LUSC/LUSC_meth_refined.rda",
+                  exp="./ELMER.example/Result/LUSC/LUSC_RNA_refined.rda", TCGA=T,
+                  probeInfo="./ELMER.example/Result/LUSC/probeInfo_feature.rda",
+                  geneInfo="./ELMER.example/Result/LUSC/geneAnnot.rda")
```

```
~~~ MEE.data: initializator ~~~
```

```
> sig.diff <- get.diff.meth(mee, cores=detectCores()/2, dir.out ="./ELMER.example/Result/LUSC",
+                           diff.dir="hypo", pvalue = 0.01)
> str(sig.diff$hypo)   ## significant hypomethylated probes
```

```
'data.frame':        742 obs. of  4 variables:
 $ probe         : chr  "cg00045114" "cg00050294" "cg00093522" "cg00163018" ...
 $ pvalue        : num  7.31e-13 4.44e-09 1.14e-23 2.24e-21 2.77e-18 ...
 $ tumorMinNormal: num  -0.35 -0.5 -0.36 -0.384 -0.374 ...
 $ adjust.p      : num  2.42e-12 1.06e-08 1.00e-22 1.56e-20 1.46e-17 ...
```

```
> # get.diff.meth automatic save output file.
> # getMethdiff.hypo.probes.csv contains statistic for all the probes.
> # getMethdiff.hypo.probes.significant.csv contains only the significant probes.
> dir(path = "./ELMER.example/Result/LUSC", pattern = "getMethdiff")
```

```
[1] "getMethdiff.hypo.probes.csv"             "getMethdiff.hypo.probes.significant.csv"
```

4.3 Identifying putative probe-gene pairs

Function, get.pair function, will be used to identify putative target gene for selected probes. This step is the most time consuming step since it contains a large amount calculations for permutation. The more permutation time is, the longer it will take. It is recommended to use multiple core for this step. Default permutation time is 1000 which may need 12 hrs by 4 cores. However 10,000 permutations is recommended which may cost 2 days if high confident results are desired.

```
> ### identify target gene for significantly hypomethylated probes.
>
> Sig.probes <- read.csv("./ELMER.example/Result/LUSC/getMethdiff.hypo.probes.significant.csv",
+                        stringsAsFactors=F)[,1]
> head(Sig.probes)  # significantly hypomethylated probes
```

```
[1] "cg00045114" "cg00050294" "cg00093522" "cg00163018" "cg00173804" "cg00223046"
```

```
> ## collect nearby 20 gene for Sig.probes
> nearGenes <-GetNearGenes(TRange=getProbeInfo(mee,probe=Sig.probes),
+                          geneAnnot=getGeneInfo(mee),cores=detectCores()/2)
> ## Identify significant probe-gene pairs
> Hypo.pair <-get.pair(mee=mee,probes=Sig.probes,nearGenes=nearGenes,
+                      permu.dir="./ELMER.example/Result/LUSC/permu",permu.size=300,Pe = 0.01,
+                      dir.out="./ELMER.example/Result/LUSC",cores=detectCores()/2,label= "hypo")
> head(Hypo.pair)  ## significant probe-gene pairs
```

	Probe	GeneID	Symbol	Distance	Sides	Raw.p	Pe
cg20701183.ID8543	cg20701183	ID8543	LMO4	2563	L1	7.453984e-14	0.003322259
cg19403323.ID255928	cg19403323	ID255928	SYT14	87477	R1	1.671937e-12	0.003322259
cg12213388.ID84451	cg12213388	ID84451	KIAA1804	993548	L4	2.527644e-12	0.003322259
cg26607897.ID55811	cg26607897	ID55811	ADCY10	292476	R3	4.593610e-12	0.003322259
cg10574861.ID8543	cg10574861	ID8543	LMO4	4715	L1	4.770162e-12	0.003322259
cg26607897.ID23432	cg26607897	ID23432	GPR161	563308	R6	8.048248e-12	0.003322259

```
> # get.pair automatically save output files.
> #getPair.hypo.all.pairs.statistic.csv contains statistic for all the probe-gene pairs.
> #getPair.hypo.pairs.significant.csv contains only the significant probes.
> dir(path = "./ELMER.example/Result/LUSC", pattern = "getPair")
```

```
[1] "getPair.hypo.all.pairs.statistic.csv" "getPair.hypo.pairs.significant.csv"
```

4.4 Motif enrichment analysis on the selected probes

Function, get.enriched.motif, will be used to calculate enrichment for motifs from factorbooks and JASPER among the selected probes. Odds Ratio is used to assess the enrichment level and 95% confidence interval of Odds Ratio is calculated.

```
> ### identify enriched motif for significantly hypomethylated probes which have putative target genes.
>
> Sig.probes.paired <- read.csv("./ELMER.example/Result/LUSC/getPair.hypo.pairs.significant.csv",
+                               stringsAsFactors=F)[,1]
> head(Sig.probes.paired) # significantly hypomethylated probes with putative target gene
```

```
[1] "cg20701183" "cg19403323" "cg12213388" "cg26607897" "cg10574861" "cg26607897"
```

```
> enriched.motif <-get.enriched.motif(probes=Sig.probes.paired,
+                                     dir.out="./ELMER.example/Result/LUSC", label="hypo",
+                                     min.incidence = 10,lower.OR = 1.1)
```

```
6 motifs are enriched.
```

```
> names(enriched.motif)  # enriched motifs
```

```
[1] "AP1"    "BARHL2" "IRF"    "PRDM1"  "TCF7L2" "TP53"
```

```
> # get.enriched.motif automatically save output files.
> # getMotif.hypo.enriched.motifs.rda contains enriched motif and the probes with the motif.
> # getMotif.hypo.motif.enrichment.csv contains summary of enriched motif.
> dir(path = "./ELMER.example/Result/LUSC", pattern = "getMotif")
```

```
[1] "getMotif.hypo.enriched.motifs.rda"  "getMotif.hypo.motif.enrichment.csv"
```

```
> # motif enrichment figure will be automatically generated.
> dir(path = "./ELMER.example/Result/LUSC", pattern = "motif.enrichment.pdf")
```

```
[1] "hypo.motif.enrichment.pdf"
```

4.5 Identifying regulatory TF

Function, get.TFs, will use the anti-correlation of a particular TF and the level of demethylation at its binding site to predict the regulatory TF.

```
> ### identify regulatory TF for the enriched motifs
>
> load("./ELMER.example/Result/LUSC/getMotif.hypo.enriched.motifs.rda")
> TF <- get.TFs(mee=mee, enriched.motif=enriched.motif,dir.out="./ELMER.example/Result/LUSC",
+               cores=detectCores()/2, label= "hypo")
> # get.TFs automatically save output files.
> # getTF.hypo.TFs.with.motif.pvalue.rda contains statistic for all TF with average
> # DNA methylation at sites with enriched motif.
> # getTF.hypo.significant.TFs.with.motif.summary.csv contains only the significant probes.
> dir(path = "./ELMER.example/Result/LUSC", pattern = "getTF")
```

```
[1] "getTF.hypo.significant.TFs.with.motif.summary.csv"
[2] "getTF.hypo.TFs.with.motif.pvalue.rda"
```

```
> # TF ranking based on statistics will be automatically generated.
> dir(path = "./ELMER.example/Result/LUSC/TFrankPlot", pattern = "pdf")
```

```
[1] "AP1.TFrankPlot.pdf"    "BARHL2.TFrankPlot.pdf" "IRF.TFrankPlot.pdf"
[4] "PRDM1.TFrankPlot.pdf"  "TCF7L2.TFrankPlot.pdf" "TP53.TFrankPlot.pdf"
```

5 TCGA.pipe

Function, TCGA.pipe, is easy usage for downloading TCGA data and perform all the analysis in ELMER. For illustration purpose, we skip downloading step. User could use getTCGA function to download TCGA data or use TCGA.pipe specify analysis include "download" step.

```
> TCGA.pipe("LUSC",wd="./ELMER.example",cores=detectCores()/2,permu.size=300,
+           analysis = c("distal.enhancer","diffMeth","pair","motif","TF.search"),
+           diff.dir="hypo",rm.chr=paste0("chr",c(1:22,"X","Y")))
```

```
####################
Select distal enhancer probes
####################


####################
Get differential DNA methylation loci
```

```
###################


~~~ MEE.data: initializator ~~~
###################
Predict pairs
###################


~~~ MEE.data: initializator ~~~
###################
Motif search
###################


12 motifs are enriched.
###################
Search responsible TFs
###################


~~~ MEE.data: initializator ~~~
```

6 Generating figures

6.1 Scatter plots

a. Generate scatter plots for one probes' nearby 20 gene expression vs DNA methylation at this probe. Figure11

```
> scatter.plot(mee,byProbe=list(probe=c("cg19403323"),geneNum=20),
+              category="TN", dir.out ="./ELMER.example/Result/LUSC", save=F)
> scatter.plot(mee,byProbe=list(probe=c("cg19403323"),geneNum=20),
+              category="TN", dir.out ="./ELMER.example/Result/LUSC", save=T) ## save to pdf
```

b. Generate scatter plot for one probe-gene pair.Figure22

```
> scatter.plot(mee,byPair=list(probe=c("cg19403323"),gene=c("ID255928")),
+              category="TN", dir.out ="./ELMER.example/Result/LUSC", save=F,lm_line=T)
> scatter.plot(mee,byPair=list(probe=c("cg19403323"),gene=c("ID255928")),
+              category="TN", dir.out ="./ELMER.example/Result/LUSC", save=T,lm_line=T)  ## save to pdf
```

c. Generate scatter plot for TF expression Vs average DNA methylation of the sites with certain motif.Figure33

```
> load("ELMER.example/Result/LUSC/getMotif.hypo.enriched.motifs.rda")
> scatter.plot(mee,byTF=list(TF=c("TP53","TP63","TP73"),
+                            probe=enriched.motif[["TP53"]]), category="TN",
+              dir.out ="./ELMER.example/Result/LUSC", save=TRUE,lm_line=T)
```

6.2 Schematic plot

Schematic plot shows a breif view of linkage between genes and probes. To make schematic plot, Pair object should be generated first.

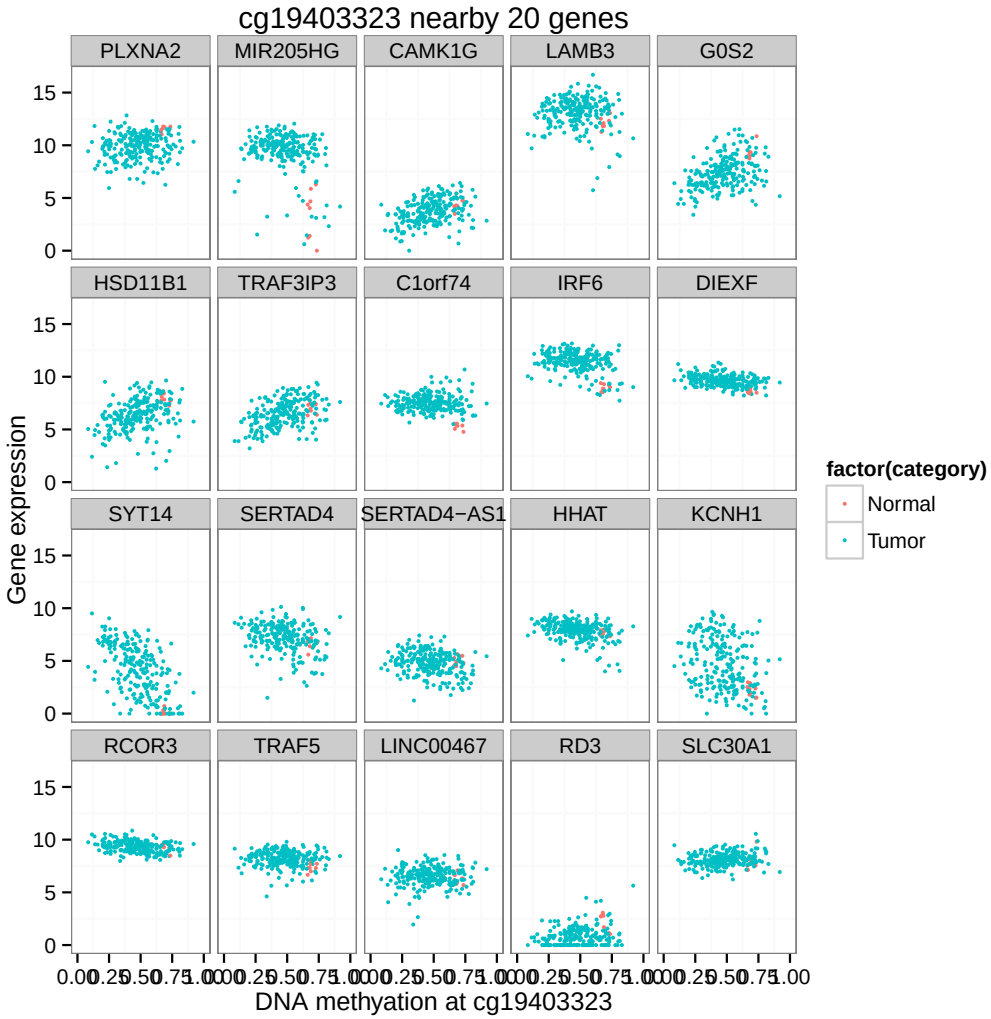

Figure 1: Each scatter plot shows the methylation level of an example probe cg19403323 in all LUSC samples plotted against the expression of one of 20 adjacent genes.

```
> #make pair object for schematic.plot
> pair <- fetch.pair(pair="./ELMER.example/Result/LUSC/getPair.hypo.pairs.significant.withmotif.csv",
+                    probeInfo = "./ELMER.example/Result/LUSC/probeInfo_feature.rda",
+                    geneInfo = "./ELMER.example/Result/LUSC/geneAnnot.rda")
```

```
~~~ Pair: initializator ~~~
```

a. Generate schematic plot for one probe with nearby 20 genes and label the gene significantly linked with the probe. Figure44

```
> grid::grid.newpage()
> schematic.plot(pair=pair, byProbe="cg19403323" ,dir.out="./ELMER.example/Result/LUSC",save=F)
> schematic.plot(pair=pair, byProbe="cg19403323" ,dir.out="./ELMER.example/Result/LUSC",save=TRUE)
```

b. Generate schematic plot for one gene with the probes which the gene significanlty linked to. Figure5**??**

```
> grid::grid.newpage()
> schematic.plot(pair=pair, byGene="ID255928" ,dir.out="./ELMER.example/Result/LUSC",save=F)
> schematic.plot(pair=pair, byGene="ID255928" ,dir.out="./ELMER.example/Result/LUSC",save=TRUE)
```

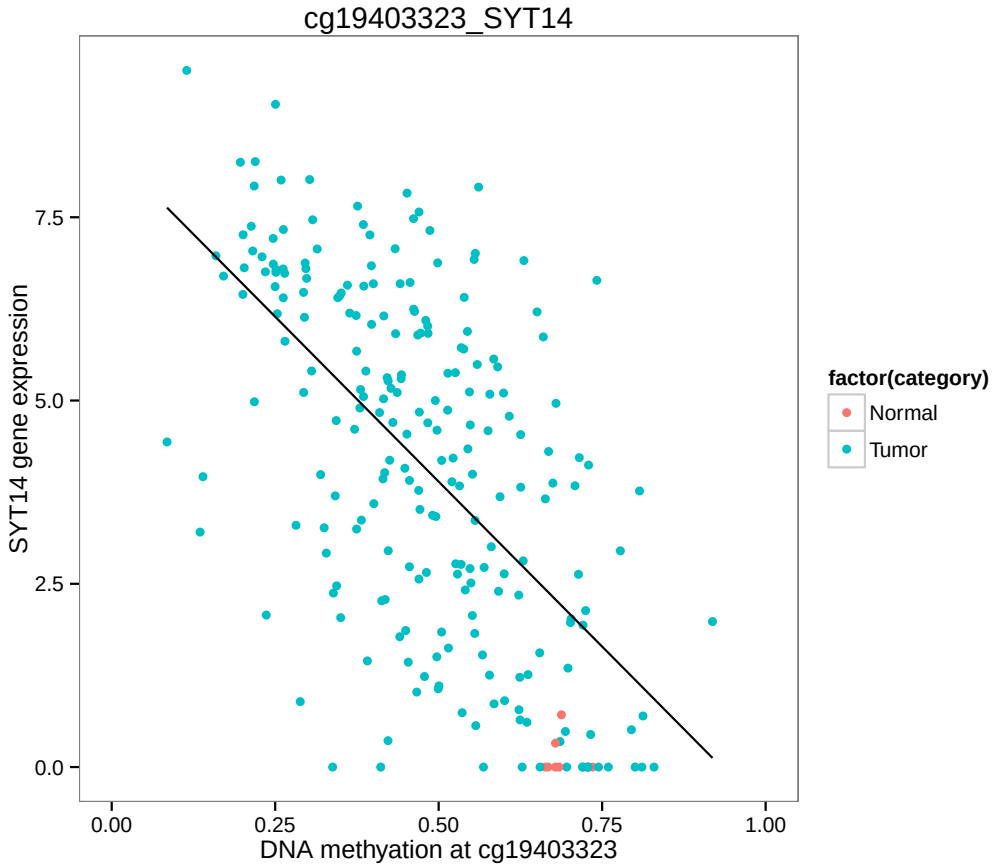

Figure 2: Scatter plot shows the methylation level of an example probe cg19403323 in all LUSC samples plotted against the expression of putative target gene SYT14.

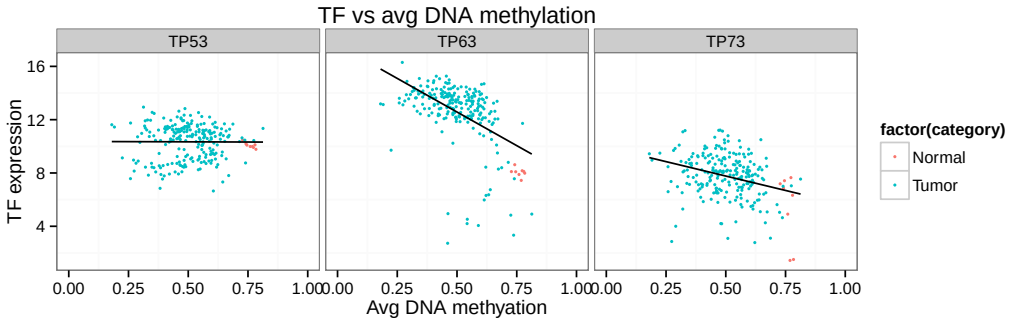

Figure 3: Each scatter plot shows the average methylation level of sites with TP53 motif in all LUSC samples plotted against the expression of transcription factor TP53, TP63, TP73 respectively.

6.3 Motif enrichment plot

Motif enrichment plot shows the enrichment levels for the selected motifs. Figure6**??**

```
> motif.enrichment.plot(motif.enrichment="./ELMER.example/Result/LUSC/getMotif.hypo.motif.enrichment.csv",
+                       significant=list(OR=1.3), dir.out ="ELMER.example/Result/LUSC",
```

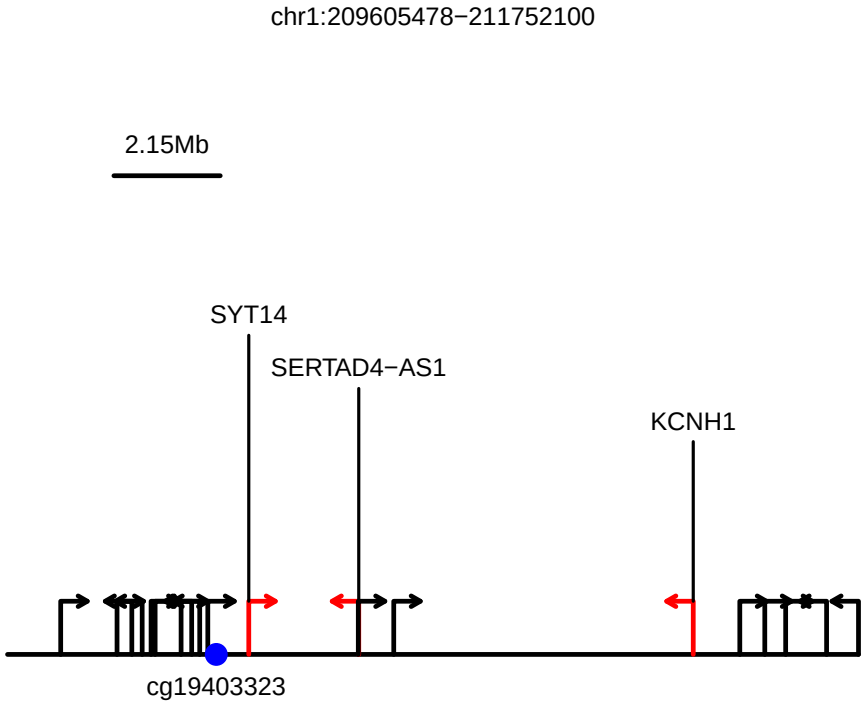

Figure 4: The schematic plot shows probe colored by blue and nearby 20 genes location. The gene significantly linked to the probe were shown in red.

```
+                       label="hypo", save=T)
> motif.enrichment.plot(motif.enrichment="./ELMER.example/Result/LUSC/getMotif.hypo.motif.enrichment.csv",
+                       significant=list(OR=1.3,lowerOR=1.3), dir.out ="ELMER.example/Result/LUSC",
+                       label="hypo", save=)  ## different signficant cut off.
```

6.4 TF ranking plot

TF ranking plot shows statistic -log10(P value) assessing the anti-correlation level of TFs expression level with average DNA methylation level at sites with a given motif. Figure7**??**

```
> load("./ELMER.example/Result/LUSC/getTF.hypo.TFs.with.motif.pvalue.rda")
> TF.rank.plot(motif.pvalue=TF.meth.cor, motif="TP53", TF.label=list(TP53=c("TP53","TP63","TP73")),
+              dir.out="./ELMER.example/Result/LUSC/TFrankPlot", save=F)
$TP53
```

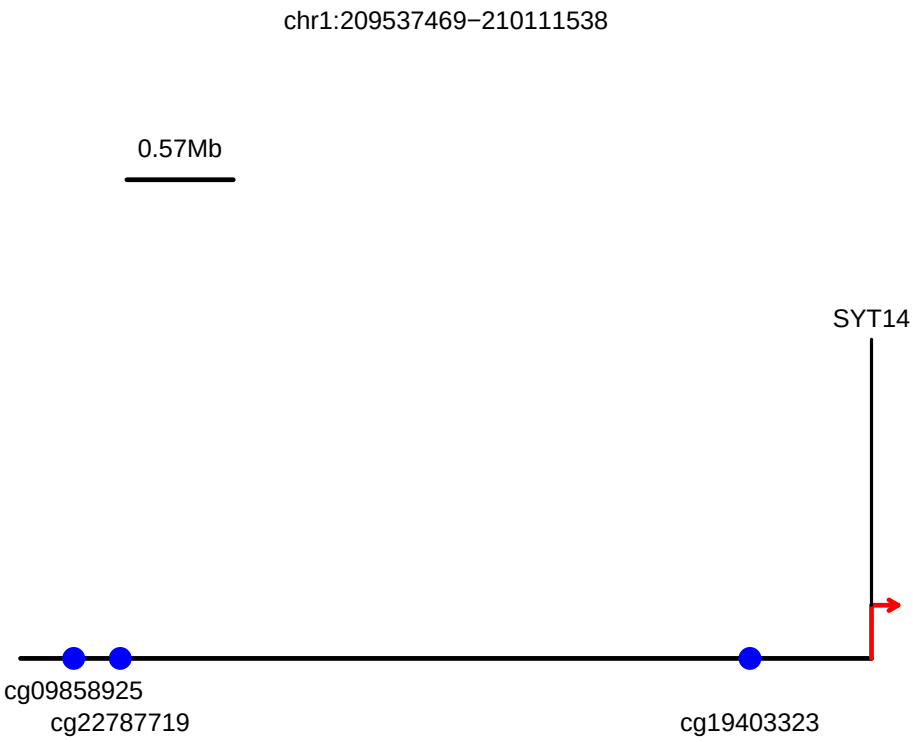

Figure 5: The schematic plot shows the gene coloredin red and all probes in blue which are significantly linked to the expression of this gene.

```
> TF.rank.plot(motif.pvalue=TF.meth.cor, motif="TP53", TF.label=list(TP53=c("TP53","TP63","TP73")),
+              dir.out="./ELMER.example/Result/LUSC/TFrankPlot", save=T) # save to pdf
```

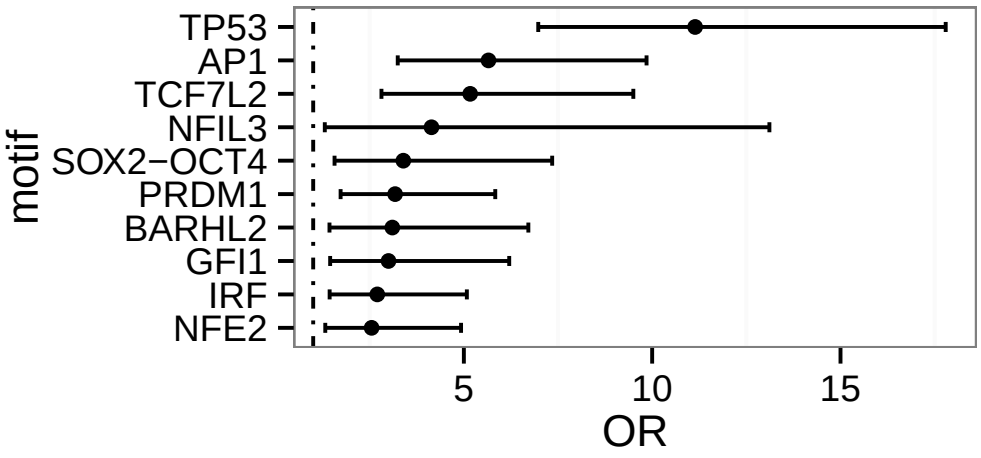

Figure 6: The plot shows the Odds Ratio (x axis) for the selected motifs with OR above 1.3. The range show the 95% confidence interval for each Odds Ratio.

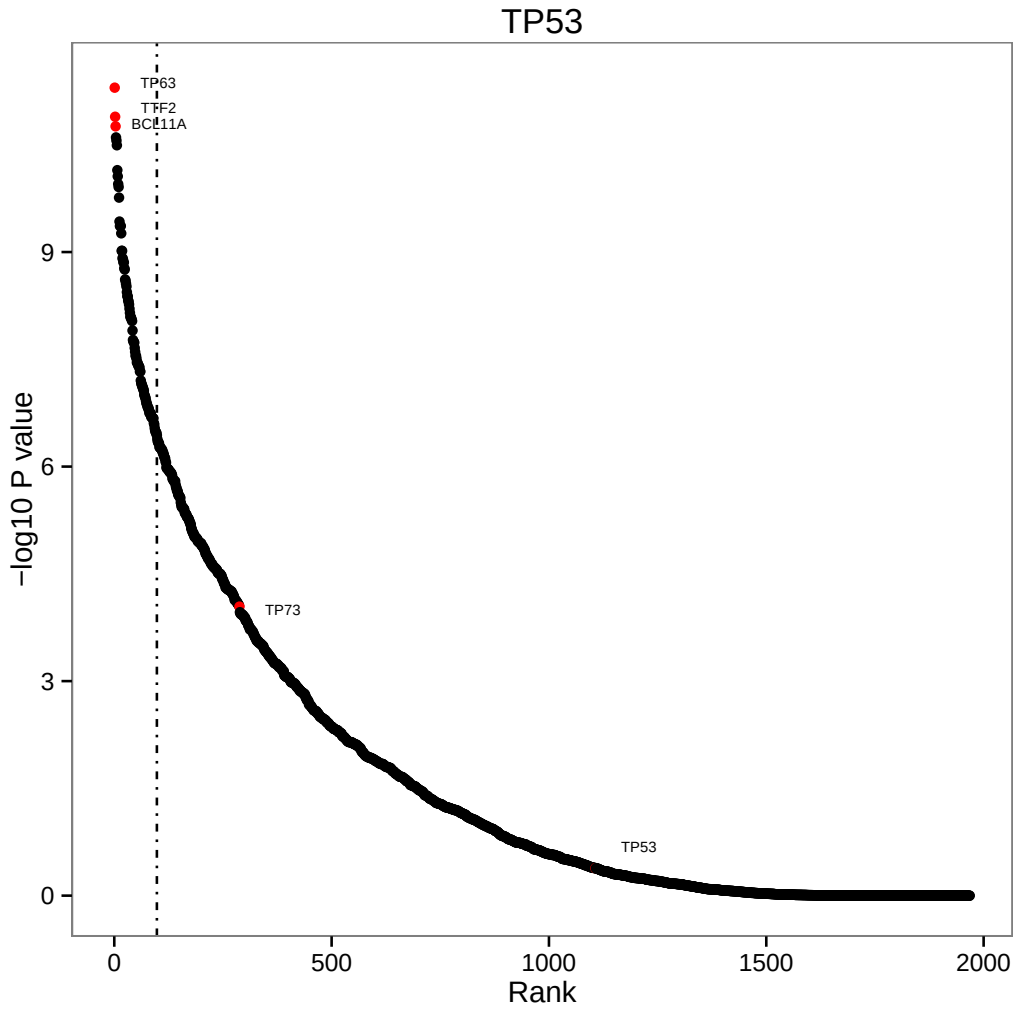

Figure 7: Shown are TF ranking plots based on the score (-log(Pr)) of association between TF expression and DNA methylation of the TP53 motif in the LUSC cancer type . The dashed line indicates the boundary of the top 5% association score. The top 3 associated TFs and the TF family members (dots in red) that are associated with that specific motif are labeled in the plot.