The coMET User Guide

Tiphaine Martin [*], Idil Yet [†], Pei-Chien Tsai [‡], Jordana T. Bell [§]

Edited: September 2014; Compiled: March 9, 2015

1 Citation

```
citation(package='coMET')

##
## To cite 'coMET' in publications use:
##
##   Martin, T., Erte, I, Tsai, P-C, Bell, J.T. coMET: an R plotting package to
##   visualize regional plots of epigenome-wide association scan results QG14, 2014
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {coMET: an R plotting package to visualize regional plots of epigenome-wide associ
##     author = {{Martin} and {T.C.} and {Erte} and {I.} and {Tsai} and {P-C.} and {Bell} and {J.
##     journal = {QG14},
##     year = {2014},
##     month = {May},
##     url = {http://quantgen.soc.srcf.net/qg14/},
##   }
```

[*] tiphaine.martin@kcl.ac.uk

[†] idil.yet@kcl.ac.uk

[‡] peichien.tsai@kcl.ac.uk

[§] jordana.bell@kcl.ac.uk

Contents

2 Introduction

The CoMET package is a web-based plotting tool and R-based package to visualize omic-WAS results in a genomic region of interest, such as EWAS (epigenome-wide association scan). CoMET provides a plot of the EWAS association signal and visualisation of the methylation correlation between CpG sites (co-methylation). The CoMET package also provides the option to annotate the region using functional genomic information, including both user-defined features and pre-selected features based on the Encode project. The plot can be customized with different parameters, such as plot labels, colours, symbols, heatmap colour scheme, significance thresholds, and including reference CpG sites. Finally, the tool can also be applied to display the correlation patterns of other genomic data in any species, e.g. gene expression array data.

coMET generates a multi-panel plot to visualize EWAS results, co-methylation patterns, and annotation tracks in a genomic region of interest. A coMET figure (cf. Fig. 1) includes three components:

1. the upper plot shows the strength and extent of EWAS association signal;
2. the middle panel provides customized annotation tracks;
3. the lower panel shows the correlation between selected CpG sites in the genomic region.

The structure of the plots builds on snp.plotter (Luna et al., 2007), with extensions to incorporate genomic annotation tracks and customized functions. coMET produces plots in PDF and Encapsulated Postscript (EPS) format.

Currently, coMET can visualise the correlation of at maximun 120 omic features. This limitation is due to the visualisation of links between the genomic positions and their poistions in the correlation matrix. Nevertheless, optional input files can be uploaded to include association P-values from the same studies (e.g. from a larger region around the first genomic region of interest) and/or other studies (e.g. from replication of from gene expression analysis) on the upper plot.

User can extract the correlations between all omic features that are visualised via the two main functions (comet and comet.web).

3 Usage

CoMET requires the installation of R, the statistical computing software, freely available for Linux, Windows, or MacOS. CoMET can be downloaded from bioconductor. Packages can be installed using the install.packages command in R. The coMET R package includes two major functions **comet.web** and **comet** to visualise omci-WAS results.

- The function **comet.web** generates output plot with the same settings of genomic annotation tracks as that of the webservice (http://epigen.kcl.ac.uk/comet or direcly http://comet.epigen.kcl.ac.uk:3838/coMET/).
- The function **comet** generates output plots with the customized annotation tracks defined by user.

```
source("http://bioconductor.org/biocLite.R")
biocLite("coMET")
```

Currently, coMET is under the development version of Bioconductor, go to the section "Install he development version of coMET from Bioconductor".

After loading from Bioconductor or gitHUB, CoMET can be loaded into a R session using this command:

```
library("coMET")

## Loading required package:  grid
## Loading required package:  biomaRt
## Loading required package:  Gviz
## Loading required package:  BiocGenerics
## Loading required package:  parallel
##
## Attaching package:  'BiocGenerics'
##
## The following objects are masked from 'package:parallel':
##
##    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ, clusterExport,
##    clusterMap, parApply, parCapply, parLapply, parLapplyLB, parRapply,
##    parSapply, parSapplyLB
##
## The following object is masked from 'package:stats':
##
##    xtabs
##
## The following objects are masked from 'package:base':
##
##    anyDuplicated, append, as.data.frame, as.vector, cbind, colnames, do.call,
##    duplicated, eval, evalq, Filter, Find, get, intersect, is.unsorted, lapply,
##    Map, mapply, match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##    Position, rank, rbind, Reduce, rep.int, rownames, sapply, setdiff, sort,
##    table, tapply, union, unique, unlist, unsplit
##
## Loading required package:  S4Vectors
## Loading required package:  stats4
## Loading required package:  IRanges
## Loading required package:  GenomeInfoDb
## Loading required package:  GenomicRanges
## Loading required package:  psych
##
## Attaching package:  'psych'
##
## The following object is masked from 'package:IRanges':
##
##    reflect
```

The configuration file specifies the options for the coMET plot. Example configuration and input files are also provided on http://epigen.kcl.ac.uk/comet. Information about the package can viewed from within R using this command:

```
?comet
?comet.web
?comet.list
```

3.1 Install the development version of coMET from Bioconductor

To install coMET from the development version of Bioconductor, the user must install R-devel from http://www.bioconductor.org/developers/how-to/useDevel/. Following this installation, you can use classically the Bioconductor's command lines.

```
source("http://bioconductor.org/biocLite.R")
biocLite("coMET")
```

3.2 Install the version of coMET from gitHub

One other way to install coMET is to download the package from gitHUB https://github.com/TiphaineCMartin/coMET and use this command line below:

```
install.packages("YourPath/coMET_YourVersion.tar.gz",repos=NULL,type="source")
##This is an example
install.packages("YourPath/coMET_0.99.9.tar.gz",repos=NULL,type="source")
```

4 Functions in coMET

Currently, there are 3 main functions:

1. **comet.web** is the pre-customized function that allows us to visualise quickly the omic-WAS results, annotation tracks, and correlation between omic features. This version is installed in the Shiny webservice. Currently, it is formated only to visualise the Human data.
2. **comet** is the generic function that allows us to visualise quickly the omic-WAS results, annotation tracks, and correlation between omic features. Users can visualise more personalised annotation tracks and give multiple extra omic-WAS results to plot.
3. **comet.list** is the function that allows us to obtain the values of correlations, the pvalues, confidence interval associated.

All functions were first designed to read the data files. But it is possible to use data frames for all data input except for the configuration file with two functions **comet** and **comet.list**. The structures of data frames (number of columns, type of contains,format) follow the same rules than files (cf. section "File formats").

5 File formats

There are five types of files that the user can give to produce the plot:

1. info file is defined in the option **mydata.file**. It is mandatory and has to be a file in tabular format with a header.
2. correlation file is defined in the option **cormatrix.file**. It is mandatory and has to be a file in tabular format with a header.
3. extra info files are defined in the option **mydata.file.large**. It is not mandatory, but if you add it, it has to be a file in tabular format with a header.

4. Annotation info file is defined in the option **biofeat.user.file**. This option exists only in the function **comet.web** and the user should inform also the format to visualise this data with the options textbf*biofeat.user.type* and textbf*biofeat.user.type.plot*.
5. Configuration file contains the values of these options instead of defining these by command line. It is a file where each line is one option. The name of option is in capital and is separated to its value by "=". If there are multiple values such as for the option **list.tracks** or the options for additional data, you need to separated them by a "comma".

5.1 Format of the info file (for option: mydata.file, mandatory)

It is mandatory and has to be a file in tabular format with an header. Info file can be a list of CpG sites with/without Beta value (DNA methylation level) or direction sign. If it is a site file then it is mandatory to have the 4 columns as shown below with headers in the same order. Beta can be the 5th column(optional) and it can be either a numeric value (positive or negative values) or only direction sign ("+", "-"). The number of columns and their types are defined but the option **mydata.format**.

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
infofile <- file.path(extdata, "cyp1b1_infofile.txt")

data_info <-read.csv(infofile, header = TRUE,
                     sep = "\t", quote = "")

head(data_info)

##     TargetID CHR  MAPINFO         Pval
## 1 cg22248750   2 38294160 2.749858e-01
## 2 cg11656478   2 38297759 7.794549e-01
## 3 cg14407177   2 38298023 2.863869e-01
## 4 cg02162897   2 38300537 3.148201e-07
## 5 cg20408276   2 38300586 1.467739e-06
## 6 cg00565882   2 38300707 7.563132e-03
```

Alternatively, the info file can be region-based and if so, the region-based info file must have the 5 columns (see below) with headers in this order. The beta or direction can be included in the 6th column (optional).

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
infoexp <- file.path(extdata, "cyp1b1_infofile_exprGene_region.txt")

data_infoexp <-read.csv(infoexp, header = TRUE,
                        sep = "\t", quote = "")

head(data_infoexp)

##                              TargetID CHR MAPINFO.START MAPINFO.STOP         Pval BETA
## 1 ENSG00000138061.7_38294652_38298453   2      38294652     38298453 3.064357e-17    +
## 2 ENSG00000138061.7_38301489_38302532   2      38301489     38302532 1.145430e-07    +
## 3 ENSG00000138061.7_38302919_38303323   2      38302919     38303323 1.014050e-08    -
```

In summary, there are 4 possible formats for info file:

1. **site**: the file has only 4 columns with a header:
   (a) Name of omic feature
   (b) Name of chromosome
   (c) the position of omic feature
   (d) the pvalue of omic feature
2. **region**: the file has only 5 columns with a header:
   (a) Name of omic feature
   (b) Name of chromosome
   (c) the start position of omic feature
   (d) the end position of omic feature
   (e) the pvalue of omic feature
3. **site_asso**: the file has only 5 columns with a header:
   (a) Name of omic feature
   (b) Name of chromosome
   (c) the position of omic feature
   (d) the pvalue of omic feature
   (e) the direction of the association related to this omic feature. It can be only the sign or the value of direction
4. **region_asso**: the file has only 6 columns with a header:
   (a) Name of omic feature
   (b) Name of chromosome
   (c) the start position of omic feature
   (d) the end position of omic feature
   (e) the pvalue of omic feature
   (f) the direction of the association related to this omic feature. It can be only the sign or the value of direction.

5.2 Format of correlation matrix (for option: cormatrix.file, mandatory)

It is mandatory and has to be a file in tabular format with an header. The data file used for the correlation matrix is described in the option **cormatrix.file**. This tab-delimited file can take 3 formats described in the option **cormatrix.format**:

1. **cormatrix**: pre-computed correlation matrix provided by the user; Dimension of matrix : CpG_number X CpG_number. Need to put the CpG sites/regions in the ascending order of positions and to have a header with the name of CpG sites/regions;
2. **raw**: Raw data format. Correlations of these can be computed by one of 3 methods Spearman, Pearson, Kendall (option **cormatrix.method**). Dimension of matrix : sample_size X CpG_number. Need to have a header with the name of CpG sites/regions ;
3. **raw_rev**: Raw data format. Correlations of these can be computed by one of 3 methods Spearman, Pearson, Kendall (option **cormatrix.method**). Dimension of matrix : CpG_number X sample_size. Need to have the row names of CpG sites/regions and a header with the name of samples ;

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
corfile <- file.path(extdata, "cyp1b1_res37_rawMatrix.txt")

data_cor <-read.csv(corfile, header = TRUE,
                    sep = "\t", quote = "")
```

```
data_cor[1:6,1:6]

##    cg22248750 cg11656478 cg14407177  cg02162897 cg20408276   cg00565882
## 1 -0.08636815 -0.4896557  1.6718967  0.52423342  0.1659252  0.224221521
## 2 -0.00107899 -0.6330666  0.3150612 -0.29820805 -0.4339332 -0.007794883
## 3  0.31656883 -0.2610083 -0.4942691  0.04657351  0.1840397  0.313967471
## 4 -0.40914999  0.6816058 -0.3251337 -0.58656175 -0.2069954  0.150719803
## 5  1.29953262  0.3985525  0.1119045  0.81181511  0.1833470  0.194928273
## 6 -1.11948826  0.3035820 -1.2794597 -0.49785237  0.1076348 -0.876011670
```

5.3 Format of extra info file (for option: mydata.large.file)

It is optional file and if you add one, it has to be a file in tabular format with an header. The extra info files can be described in the option **mydata.large.file** and their format in **mydata.large.format**. Different extra info files are separated by a comma.

This can be another type of info file (e.g expression or replication data) and should follow the same rules as the standard info file.

5.4 Format of annotation file (for option biofeat.user.file)

The file is defined in the option **biofeat.user.file** and the format of file is the format accepted by GViz (BED, GTF, and GFF3).

5.5 Option of config.file

It is a file where each line is one option. The name of option is in capital and is separated to its value by ”=”. If there are multiple values such as for the option **list.tracks** or the options for additional data, you need to separated them by a ”comma”.. If you would like to make your own changes to the plot you can download the configuration file, make changes to it, and upload it into R as shown in the example below.

The important options of a coMET figure include three components:

1. The upper plot shows the strength and extent of EWAS association signal.
   - **pval.threshold**: Significance threshold to be displayed as a red dashed line
   - **disp.association**: This logical option works only if **mydata.file** contains the effect direction (**mydata.format=site_asso** or **region_asso**). The value can be TRUE or FALSE: if FALSE (default), for each point of data in the p-value plot, the color of symbol is the color of co-methylation pattern between the point and the reference site; if TRUE, the effect direction is shown. If the association is positive, the color is the one defined with the option **color.list**. On the other hand, if the association is negative, the color is the opposed color.
   - disp.REGION : This logical option works only if **mydata.file** contains regions (**mydata.format=region** or **region_asso**). The value can be TRUE or FALSE (default). If TRUE, the genomic element will be shown by a continuous line with the color of the element, in addition to the symbol at the center of the region. If FALSE, only the symbol is shown.
2. The middle panel provides customized annotation tracks;

- **list.tracks** (for *comet.web* function): List of annotation tracks that can be visualised: geneENSEMBL, CGI, ChromHMM, DNAse, RegENSEMBL, SNP, transcriptENSEMBL, SNPstoma, SNPstru, SNPstrustoma, ISCA, COSMIC, GAD, ClinVar, GeneReviews, GWAS, ClinVarCNV, GCcontent, genesUCSC, xenogenesUCSC. The elements are separated by a comma.
- tracks.gviz, tracks.ggbio, tracks.trackviewer (for *comet* function): For each option, it is possible to give a list of annotation tracks that is created by the Gviz, GGBio, and TrackViewer bioconductor packages.

3. The lower panel shows the correlation between selected CpG sites in the genomic region.
    - **cormatrix.format** : Format of the input fie **cormatrix.file**: either raw data (option RAW if CpG sites are by column and samples by row or option RAW_REV if CpG site are by row and samples by column) or correlation matrix (option CORMATRIX)
    - **cormatrix.method** : If raw data are provided it will be necessary to produce the correlation matrix using one of 3 methods (spearman, pearson and kendall).
    - **cormatrix.color.scheme** : There are 5 colors (heat, bluewhitered, cm, topo, gray, bluetored)

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
configfile <- file.path(extdata, "config_cyp1b1_zoom_4webserver.txt")

data_config <-read.csv(configfile, quote = "", sep="\t", header=FALSE)
data_config
```

```
##                                                            V1
## 1                                            disp.mydata=TRUE
## 2                                          mydata.format=site
## 3                                          sample.labels=CpG 
## 4                                         symbols=circle-fill
## 5                                                   lab.Y=log
## 6                                         disp.color.ref=TRUE
## 7                                       mydata.ref=cg02162897
## 8                                 pval.threshold=4.720623e-06
## 9                                      disp.association=FALSE
## 10                                          disp.region=FALSE
## 11                                             start=38290160
## 12                                               end=38303219
## 13                            mydata.large.format=region_asso
## 14                               disp.association.large=TRUE
## 15                                     disp.region.large=TRUE
## 16                       sample.labels.large=Gene expression 
## 17                                     color.list.large=green
## 18                                 symbols.large=diamond-fill
## 19                                       cormatrix.format=raw
## 20                                     disp.cormatrixmap=TRUE
## 21                                  cormatrix.method=spearman
## 22                        cormatrix.color.scheme=bluewhitered
## 23                                        disp.phys.dist=TRUE
## 24                                        disp.color.bar=TRUE
## 25                                           disp.legend=TRUE
## 26 list.tracks=geneENSEMBL,CGI,ChromHMM,DNAse,RegENSEMBL,SNP
## 27                                      disp.mult.lab.X=FALSE
## 28                                             image.type=pdf
## 29   image.title="Example a-DMR in CYP1B1 in Adipose tissue"
## 30                      image.name=cyp1b1_zoom_plus_name_expr
## 31                                             image.size=3.5
## 32                                                genome=hg19
## 33                        dataset.geneE=hsapiens_gene_ensembl
```

6 Creating a plot like the webservice: comet.web

User can draw coMET via the coMET website (http://epigen.kcl.ac.uk/comet). It is possible to reproduce the web service plotting defaults by using the function comet.web, for example see Figure 1.

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
myinfofile <- file.path(extdata, "cyp1b1_infofile.txt")
myexpressfile <- file.path(extdata, "cyp1b1_infofile_exprGene_region.txt")
mycorrelation <- file.path(extdata, "cyp1b1_res37_rawMatrix.txt")
configfile <- file.path(extdata, "config_cyp1b1_zoom_4webserver.txt")
comet.web(config.file=configfile, mydata.file=myinfofile,
          cormatrix.file=mycorrelation ,mydata.large.file=myexpressfile,
          print.image=FALSE,verbose=FALSE)
```

6.1 Hidden values of comet.web function

This is the hidden values of **comet.web** function. If these values do not correspond to what you want to visualise, you need to use the function **comet**, more generic.

Option	Value
mydata.type	FILE
mydata.large.type	LISTFILE
cormatrix.type	LISTFILE
disp.cormatrixmap	TRUE
disp.pvalueplot	TRUE
disp.mydata.names	TRUE
disp.connecting.lines	TRUE
disp.mydata	TRUE
disp.type	symbol
biofeat.user.type.plot	histogram
tracks.gviz	NULL
tracks.ggbio	NULL
tracks.trackviewer	NULL
biofeat.user.file	NULL
palette.file	NULL
disp.color.bar	TRUE
disp.phys.dist	TRUE

Continued on next page

Table 1 – continued from previous page

Option	Value
disp.legend	TRUE
disp.marker.lines	TRUE
disp.mult.lab.X	FALSE
connecting.lines.factor	1.5
connecting.lines.adj	0.01
connecting.lines.vert.adj	-1
connecting.lines.flex	0
font.factor	NULL
color.list	red
font.factor	NULL
dataset.gene	hsapiens_gene_ensembl
DATASET.SNP	hsapiens_snp
VERSION.DBSNP	snp142Common
DATASET.SNP.STOMA	hsapiens_snp_som
DATASET.REGULATION	hsapiens_feature_set
DATASET.STRU	hsapiens_structvar
DATASET.STRU.STOMA	hsapiens_structvar_som
BROWSER.SESSION	UCSC

7 Creating a plot with the generic function: comet

It is possible to create the annotation tracks by Gviz, trackviewer or ggbio, for example see Figure 2. Currently, the Gviz option for annotation tracks, in combination with the heatmap of correlation values between genomic elements, provides the most informative and easy approach to visualize graphics.

7.1 coMET plot: pvalue plot, annotation tracks, and correlation matrix

7.1.1 From files

In this figure 2, we create the different tracks outside to coMET with Gviz. The list of annotation tracks and different files are given to the function coMET.

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
configfile <- file.path(extdata, "config_cyp1b1_zoom_4comet.txt")
myinfofile <- file.path(extdata, "cyp1b1_infofile.txt")
myexpressfile <- file.path(extdata, "cyp1b1_infofile_exprGene_region.txt")
mycorrelation <- file.path(extdata, "cyp1b1_res37_rawMatrix.txt")

chrom <- "chr2"
start <- 38290160
end <- 38303219
gen <- "hg19"
strand <- "*"
```

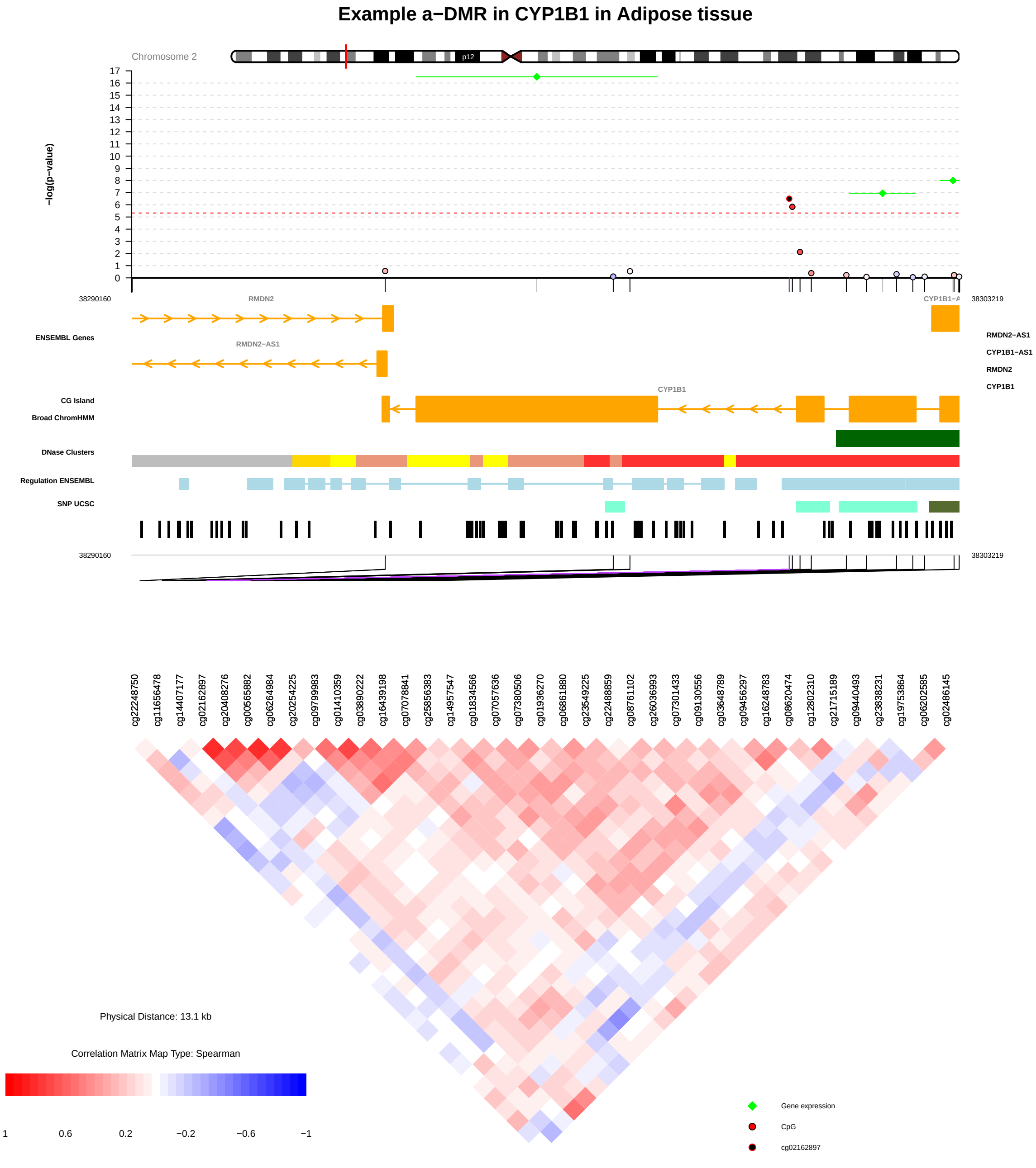

Figure 1: Plot with comet.web function.

```
BROWSER.SESSION="UCSC"
mySession <- browserSession(BROWSER.SESSION)
genome(mySession) <- gen

genetrack <-genesENSEMBL(gen,chrom,start,end,showId=TRUE)
snptrack <- snpBiomart(chrom, start, end, dataset="hsapiens_snp_som",showId=FALSE)
iscatrack <-ISCATrack(gen,chrom,start,end,mySession, table="iscaPathogenic")
```

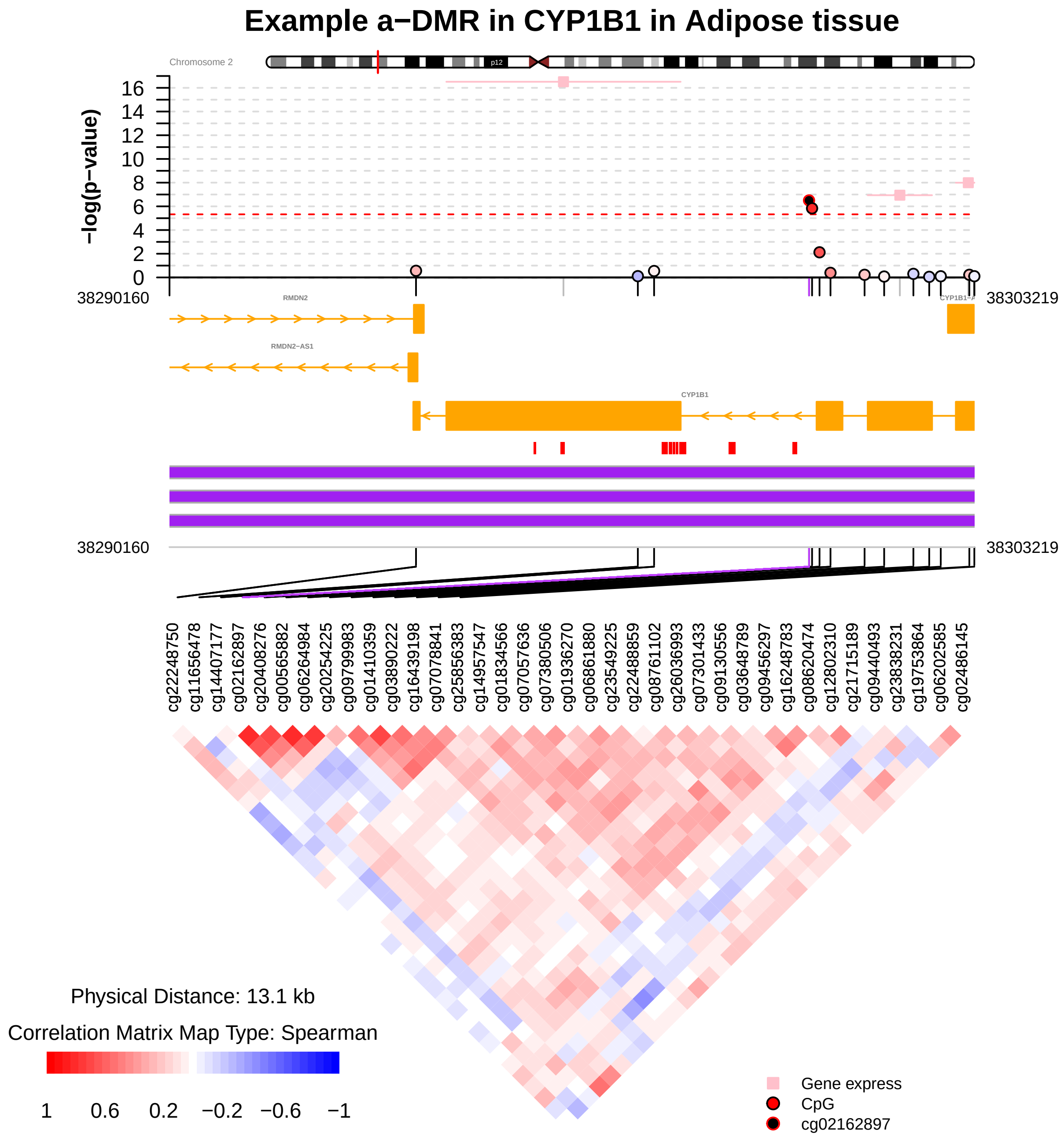

Figure 2: Plot with comet function from files.

```
listgviz <- list(genetrack,snptrack,iscatrack)

comet(config.file=configfile, mydata.file=myinfofile, mydata.type="file",
      cormatrix.file=mycorrelation,  cormatrix.type="listfile",
      mydata.large.file=myexpressfile, mydata.large.type="listfile",
      tracks.gviz=listgviz, verbose=FALSE, print.image=FALSE)
```

7.1.2 From data frame

In this figure 3, we visualize the same data than in the figure 2, but they are in data frame and not from file.

In addition, we would like to visualise only the correlations between CpG sites having a pvalue under or equal 0.05 (alpha of confidence level: 0.05, ajustment with Benjamini and Hochberg). The correlations having a pvalue above 0.05 have a color "goshwhite" whereas the other correlations have the color related to the correlation level. In the pvalue plot (upper plot), the points of each omic feature have their colors related to their correlations with the reference omic feature without taking account the significant level of pvalue associated with the correlation matrix.

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
configfile <- file.path(extdata, "config_cyp1b1_zoom_4comet.txt")
myinfofile <- file.path(extdata, "cyp1b1_infofile.txt")
myexpressfile <- file.path(extdata, "cyp1b1_infofile_exprGene_region.txt")
mycorrelation <- file.path(extdata, "cyp1b1_res37_rawMatrix.txt")

chrom <- "chr2"
start <- 38290160
end <- 38303219
gen <- "hg19"
strand <- "*"

BROWSER.SESSION="UCSC"
mySession <- browserSession(BROWSER.SESSION)
genome(mySession) <- gen

genetrack <-genesENSEMBL(gen,chrom,start,end,showId=TRUE)
snptrack <- snpBiomart(chrom, start, end, dataset="hsapiens_snp_som",showId=FALSE)
iscatrack <-ISCATrack(gen,chrom,start,end,mySession, table="iscaPathogenic")

listgviz <- list(genetrack,snptrack,iscatrack)

matrix.dnamethylation <- read.delim(myinfofile, header=TRUE, sep="\t", as.is=TRUE,
                                    blank.lines.skip = TRUE, fill=TRUE)
matrix.expression <- read.delim(myexpressfile, header=TRUE, sep="\t", as.is=TRUE,
                                blank.lines.skip = TRUE, fill=TRUE)
cormatrix.data.raw <- read.delim(mycorrelation, sep="\t", header=TRUE, as.is=TRUE,
                                 blank.lines.skip = TRUE, fill=TRUE)

listmatrix.expression <- list(matrix.expression)
listcormatrix.data.raw <- list(cormatrix.data.raw)
comet(config.file=configfile, mydata.file=matrix.dnamethylation,
      mydata.type="dataframe",cormatrix.file=listcormatrix.data.raw,
      cormatrix.type="listdataframe",cormatrix.sig.level=0.05,
      cormatrix.conf.level=0.05, cormatrix.adjust="BH",
      mydata.large.file=listmatrix.expression, mydata.large.type="listdataframe",
      tracks.gviz=listgviz,verbose=FALSE, print.image=FALSE)
```

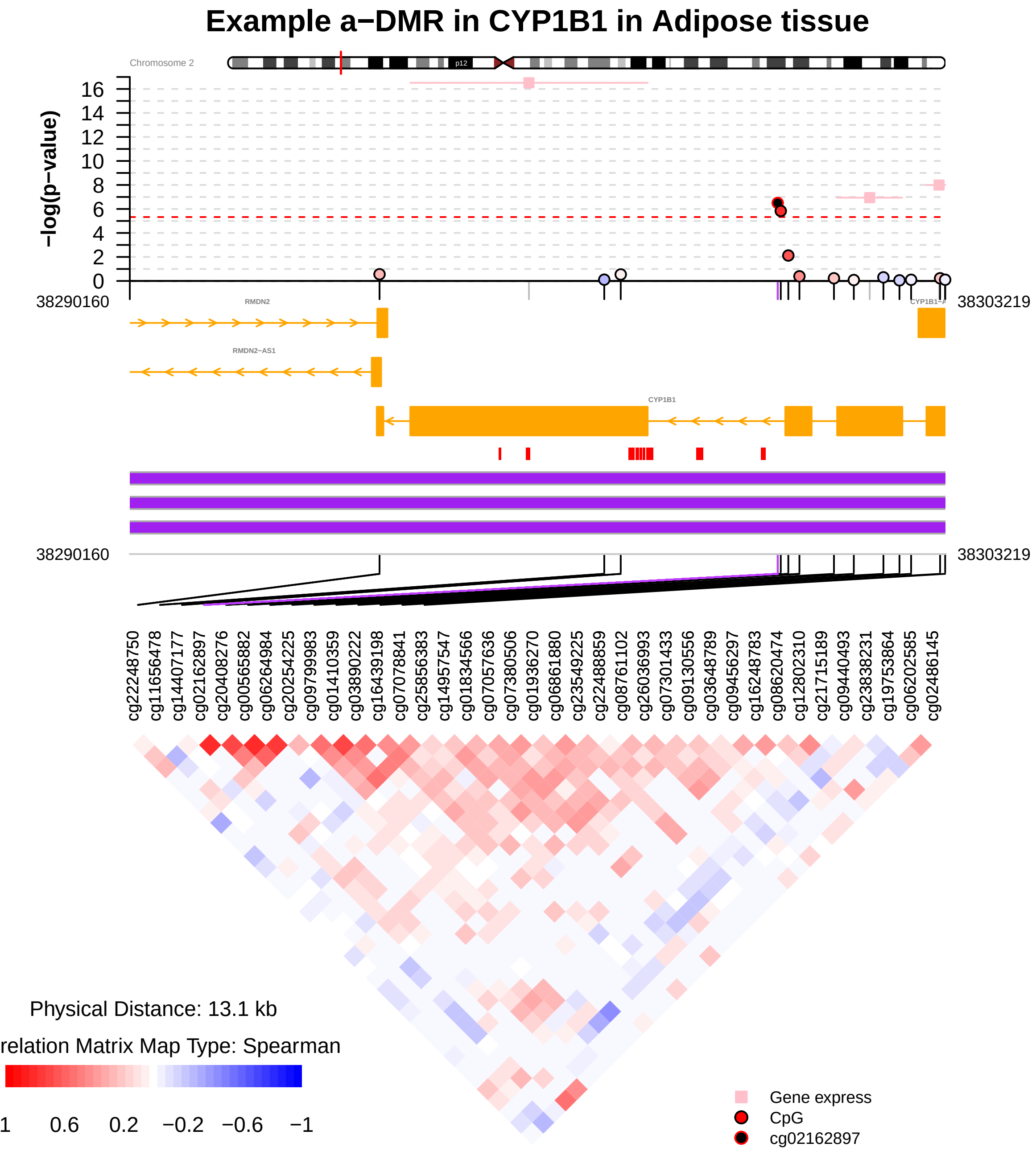

Figure 3: Plot with comet function from matrix data and with a pvalue threshold for the correlation between omics features (here CpG sites).

7.2 coMET plot: annotation tracks and correlation matrix

It is possible to visualise only annotation tracks and the correlation between genetic elements. In this case, we need to use the option `disp.pvalueplot=FALSE`, for example see Figure 4.

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
configfile <- file.path(extdata, "config_cyp1b1_zoom_4cometnopval.txt")
```

```
myinfofile <- file.path(extdata, "cyp1b1_infofile.txt")
mycorrelation <- file.path(extdata, "cyp1b1_res37_rawMatrix.txt")

chrom <- "chr2"
start <- 38290160
end <- 38303219
gen <- "hg19"
strand <- "*"

genetrack <-genesENSEMBL(gen,chrom,start,end,showId=FALSE)
snptrack <- snpBiomart(chrom, start, end,
                       dataset="hsapiens_snp_som",showId=FALSE)
strutrack <- structureBiomart(chrom, start, end,
                              strand, dataset="hsapiens_structvar_som")
clinVariant<-ClinVarMainTrack(gen,chrom,start,end)
clinCNV<-ClinVarCnvTrack(gen,chrom,start,end)
gwastrack <-GWASTrack(gen,chrom,start,end)
geneRtrack <-GeneReviewsTrack(gen,chrom,start,end)

listgviz <- list(genetrack,snptrack,strutrack,clinVariant,
                 clinCNV,gwastrack,geneRtrack)
comet(config.file=configfile, mydata.file=myinfofile, mydata.type="file",
      cormatrix.file=mycorrelation, cormatrix.type="listfile",
      tracks.gviz=listgviz, verbose=FALSE, print.image=FALSE,disp.pvalueplot=FALSE)
```

8 Extract the significant correlations between omic features

CoMET can help to visualise the correlations between omic features with EWAS results and other omic data. In addition, a function **comet.list** can extract the significant correlations according the method (**cormatrix.method**) and the significant level (**cormatrix.sig.level**).

The output file has 7 columns:

1. the name of the first omic feature
2. the name of the second omic feature
3. the correlation between the omic features
4. the alpha/2 lower value. here the alpha is 0.05 (**cormatrix.conf.level**)
5. the alpha/2 upper value. here the alpha is 0.05 (**cormatrix.conf.level**)
6. the pvalue
7. the pvalue adjusted with the method selected. here is Benjamin and Hochberg (**cormatrix.adjust**)

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
mycorrelation <- file.path(extdata, "cyp1b1_res37_rawMatrix.txt")
myoutput <- file.path(extdata, "cyp1b1_res37_cormatrix_list_BH05.txt")


comet.list(cormatrix.file=mycorrelation,cormatrix.method = "spearman",
```

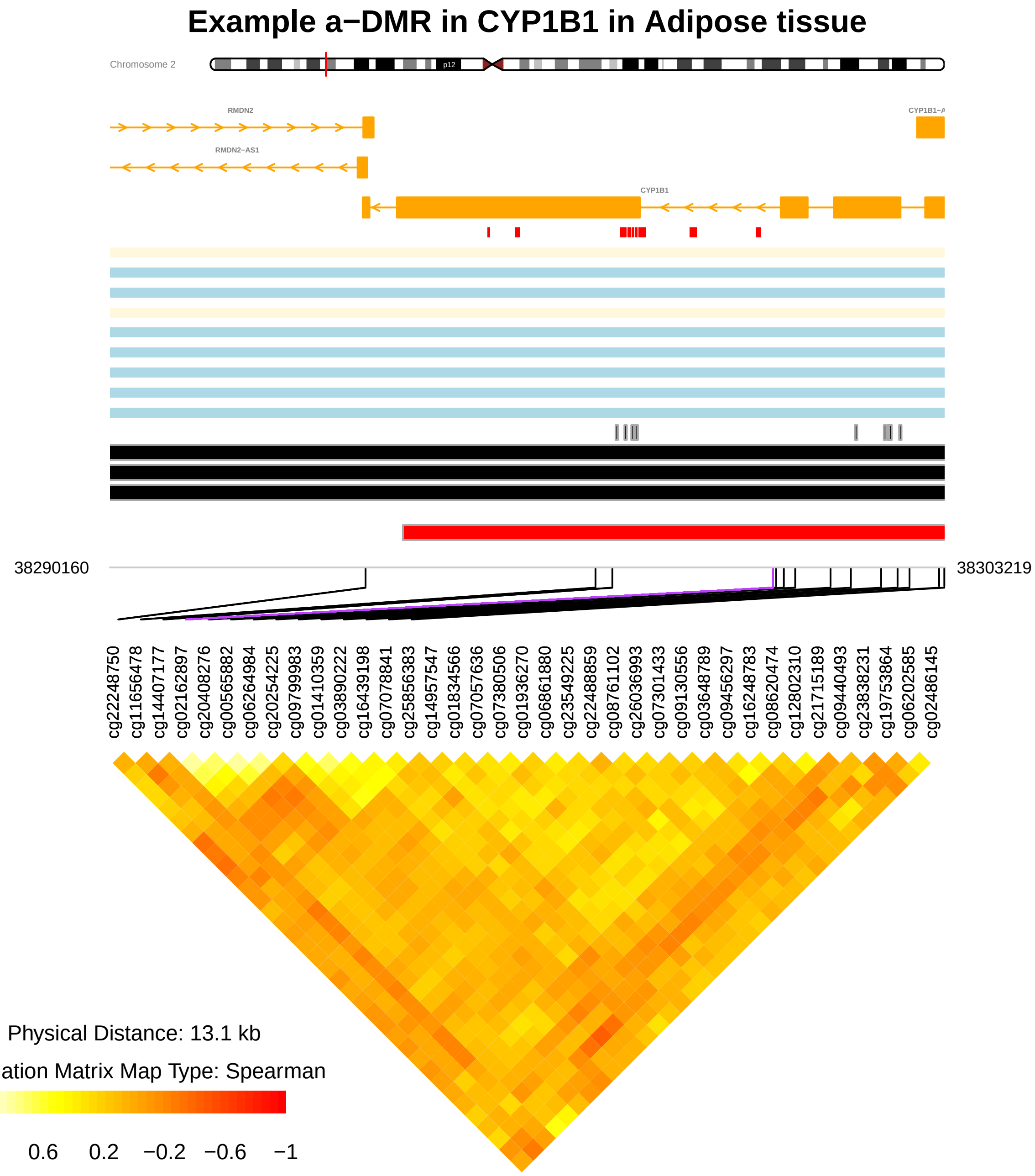

Figure 4: Plot with comet function without pvalue plot.

```
            cormatrix.format= "raw", cormatrix.conf.level=0.05,
            cormatrix.sig.level= 0.05, cormatrix.adjust="BH",
            cormatrix.type = "listfile", cormatrix.output=myoutput,
            verbose=FALSE)

listcorr <- read.csv(myoutput, header = TRUE,
                     sep = "\t", quote = "")
```

```
dim(listcorr)

## [1] 336   7

head(listcorr)

##   omicFeature1 omicFeature2 correlation     lowerCI    upperCI           pvalue
## 1   cg22248750   cg14407177   0.2153743  0.11294792  0.3132713 0.00004975019784
## 2   cg22248750   cg02162897   0.2761912  0.17632357  0.3704308 0.00000015755195
## 3   cg22248750   cg20408276   0.2807258  0.18108231  0.3746643 0.00000009649818
## 4   cg22248750   cg00565882   0.2345897  0.13288218  0.3314082 0.00000947899249
## 5   cg22248750   cg06264984   0.1793832  0.07583111  0.2791072 0.00076134404131
## 6   cg22248750   cg09799983  -0.2979454 -0.39070492 -0.1991959 0.00000001382644
##   pvalue.adjusted
## 1 0.0002029592392
## 2 0.0000011789842
## 3 0.0000007472999
## 4 0.0000447731135
## 5 0.0024145482453
## 6 0.0000001261426
```

9 Extra information about annotation tracks

Annotation tracks can be created with Gviz in using four different functions:

1. UcscTrack. To know the different possible tracks to visualise from UCSC, you need to look at the table Browser of UCSC http://genome-euro.ucsc.edu/cgi-bin/hgTables?hgsid=202842745_Dlvit14QOOG6ZPpLo
   mammal&org=Human&db=hg19&hgta_group=varRep&hgta_track=cpgIslandExt&hgta_table=0&hgta_regionType=genome&position=chr6%3A32726553-32727053&hgta_outputType=primaryTable&hgta_outFileName=
2. BiomartGeneRegionTrack. You need to create a connection to Biomart database and it visualises Genes data
3. DataTrack. Need to be numerical data
4. AnnotationTrack. It allows to visualize any annotation data.

To have more information, it is better to read the Gviz's vignette.

9.1 Genes and transcripts from ENSEMBL and UCSC

The color of genetic elements are defined by the R package Gviz.

9.2 Regulatory elements from ENSEMBL

The color of regulatory elements from ENSEMBL are defined from the same criteria of ENSEMBL done in 2014. The colors and the list of features can be updated in ENSEMBL and not yet in coMET. Please to contact us if you see some difference.

Currently the colors are :

Omic feature	Color
Promoter Associated	darkolivegreen
CTCF Binding Site	cadetblue1
Gene Associated	coral
Non-Gene Associated	darkgoldenrod1
Predicted Transcribed Region	greenyellow
PolIII Transcription Associated	purple
Enhancer	gold
Transcription Factor Binding Site	darkorchid1
Predicted Weak enhancer/Cis-reg element	yellow
Heterochromatin	wheat4
Open Chromatin	snow3
Promoter Flank	tomato
Repressed/Low Activity	snow4
Unclassified	aquamarine

9.3 ChromHMM from UCSC

The color of regulatory regions from UCSC are defined from the same criteria of UCSC done in 2014. The colors and the list of features can be updated in UCSC and not yet in coMET. Please to contact us if you see some difference.

Currently the colors are :

Omic feature	Color
1_Active_Promoter	firebrick1
2_Weak_Promoter	darksalmon
3_Poised_Promoter	blueviolet
4_Strong_Enhancer	Orange
5_Strong_Enhancer	coral
6_Weak_Enhancer	yellow
7_Weak_Enhancer	gold
8_Insulator	cornflowerblue
9_Txn_Transition	darkolivegreen
10_Txn_Elongation	forestgreen
11_Weak_Txn	darkseagreen1
12_Repressed	gainsboro
13_Heterochrom/lo	gray74
14_Repetitive/CNV	gray77
15_Repetitive/CNV	gray86

9.4 structureBiomart from Ensembl

These colors are for somatic structural variation and structural variation for any species.

Omic feature	Color
copy_number_variation	cornsilk
inversion	darkolivegreen
translocation	cyan
sequence_alteration	coral
snp	red
insertion	blueviolet
deletion	orange
indel	darkgoldenrod1
substitution	dodgerblue2

9.5 ISCA track

International Standards of Cytogenomic Arrays Consortium defined a set of phenotypes for CNVs. Different colors are defined to represent them.

Omic feature	Color
iscaPathogeni	purple
iscaPathGainCum	red
iscaPathLossCum	blue
iscaCuratedPathogeni	purple
iscaLikelyPathogeni	lightpurple
iscaUncertain	lightgrey
iscaBenign	black
iscaCuratedBenign	black
iscaLikelyBenign	black

10 coMET: Shiny web-service

10.1 How to use coMET's Shiny web-service

If you want to use coMET via its webservice, you will go to http://epigen.kcl.ac.uk/comet and select one of different instances or direcly one of instance such as http://comet.epigen.kcl.ac.uk:3838/coMET/. We have created different instances of coMET because we did not have the pro version of Shiny. Nevertheless, it uses the same version of coMET.

If you use coMET from a Shiny webservice, you do not need to install coMET package on your computer and to know the command lines to run on R. It is easy to use it, you have just to load different files and configurate your plot. The creation of coMET's plot can take a time because it makes a connexion to UCSC or/and ENSEMBL for the annotation tracks. First, the plot is created on the webpage, you can save the picture. But if you want to have better quality, you need to push the button download and the plot would be recreated in a file at pdf or eps format.

10.2 How to install coMET's Shiny web-service

These are different steps to install coMET on your Shiny web-service. You need to be root of your server to install it.

1. You can install on your server an instance of Shiny http://shiny.rstudio.com/.
2. You need also to install R, Bioconductor and, of course, the package coMET.
3. In the Shiny's folder (e.g. /var/shiny-server/www), you can create a folder called "COMET".
4. Following this, you can install the two scripts of coMET that you find in the folder www of coMET package in the new folder.
5. You need to change owner and the permission to access this folder. Only the user called Shiny can access.
   `chmod 755 /var/shiny-server/www/COMET`
   `chown shiny:shiny /var/shiny-server/www/COMET`
6. You need now to update the configuration file of Shiny (e.g. /etc/shiny-server/shiny-server.conf).
7. You need to change owner and the permission to access this file
   `chmod 744 /etc/shiny-server/shiny-server.conf`
   `chown shiny:shiny /etc/shiny-server/shiny-server.conf`
8. In this end, you can restart the service Shiny via the command line: sudo restart shiny-server

Your Shiny's configuration file:

```
run_as shiny;
# Define a top-level server which will listen on a port
server {
# Instruct this server to listen on port 3838
listen 3838;
# Define the location available at the base URL
location / {
# Run this location in 'site_dir' mode, which hosts the entire directory
# tree at '/srv/shiny-server'
site_dir /var/shiny-server/www;

# Define where we should put the log files for this location
log_dir /var/shiny-server/log;

# Should we list the contents of a (non-Shiny-App) directory when the user
# visits the corresponding URL?
directory_index off;

#   app_init_timeout 3600;
#   app_idle_timeout 3600;
}

}
```

11 SessionInfo

The following is the session info that generated this vignette:

```
toLatex(sessionInfo())
```

- R version 3.1.2 (2014-10-31), x86_64-pc-linux-gnu
- Locale: LC_CTYPE=en_GB.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=en_GB.UTF-8, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_GB.UTF-8, LC_PAPER=en_US.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Base packages: base, datasets, graphics, grDevices, grid, methods, parallel, stats, stats4, utils
- Other packages: BiocGenerics 0.12.1, biomaRt 2.22.0, coMET 0.99.9, GenomeInfoDb 1.2.4, GenomicRanges 1.18.4, Gviz 1.10.9, IRanges 2.0.1, knitr 1.9, psych 1.5.1, S4Vectors 0.4.0, XVector 0.6.0
- Loaded via a namespace (and not attached): acepack 1.3-3.3, AnnotationDbi 1.28.1, base64enc 0.1-2, BatchJobs 1.5, BBmisc 1.9, Biobase 2.26.0, BiocParallel 1.0.3, BiocStyle 1.4.1, Biostrings 2.34.1, biovizBase 1.14.1, bitops 1.0-6, brew 1.0-6, BSgenome 1.34.1, checkmate 1.5.1, cluster 2.0.1, codetools 0.2-10, colorspace 1.2-4, colortools 0.1.5, DBI 0.3.1, dichromat 2.0-0, digest 0.6.8, evaluate 0.5.5, fail 1.2, foreach 1.4.2, foreign 0.8-63, formatR 1.0, Formula 1.2-0, GenomicAlignments 1.2.1, GenomicFeatures 1.18.3, GGally 0.5.0, ggbio 1.14.0, ggplot2 1.0.0, graph 1.44.1, gridExtra 0.9.1, gtable 0.1.2, gWidgets 0.0-54, gWidgetstcltk 0.0-55, hash 2.2.6, highr 0.4, Hmisc 3.14-6, iterators 1.0.7, lattice 0.20-30, latticeExtra 0.6-26, MASS 7.3-37, matrixStats 0.14.0, mnormt 1.5-1, munsell 0.4.2, nnet 7.3-9, OrganismDbi 1.8.0, pbapply 1.1-1, plyr 1.8.1, proto 0.3-10, RBGL 1.42.0, RColorBrewer 1.1-2, Rcpp 0.11.4, RCurl 1.95-4.5, reshape 0.8.5, reshape2 1.4.1, rpart 4.1-9, Rsamtools 1.18.2, RSQLite 1.0.0, rtracklayer 1.26.2, scales 0.2.4, sendmailR 1.2-1, splines 3.1.2, stringr 0.6.2, survival 2.38-1, tcltk 3.1.2, tools 3.1.2, trackViewer 1.2.0, VariantAnnotation 1.12.9, XML 3.98-1.1, zlibbioc 1.12.0