The coMET User Guide

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

Edited: July 2015; Compiled: June 14, 2016

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
##
##   Martin, T., Yet, I, Tsai, P-C, Bell, J.T. coMET: visualisation of regional
##   epigenome-wide association scan results and DNA co-methylation patterns BMC
##   Bioinformatics, 2015 (accepted)
```

[*] tiphaine.martin@kcl.ac.uk
[†] thomas.hardiman@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.

The current version of coMET can visualise EWAS results and annotations from a genomic region up to an entire chromosome in the upper and middle panels of the coMET plot. However, the lower panel (co-methylation) is restricted to visualising a maximum of 120 single-CpG or region-based datapoints. This limitation is due to limitations in the size of a standard A4 plot, and may be updated in the near future. However, the user can use the function comet.list to extracts all significant correlations beyond a given threshold in the dataset from either a genomic region or from an entire chromosome if required.

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")
```

CoMET uses the packages called "psych", "corrplot" and "colortools", which are not available from bioconductor. This must be installed before the installation of coMET.

```
install.packages("psych")
install.packages("corrplot")
install.packages("colortools")
```

coMET has a development version on gitHub, go to the section "Install the development version of coMET from Bioconductor".

You can install also on the version R 3.2.2 via the master version of package on gitHub. The same steps must be followed as described in the section "Install the development version of coMET from Bioconductor".

After downloading from Bioconductor or gitHUB, and installing on your computer, CoMET can be loaded into a R session using this command:

```
library("coMET")
```

```
## Loading required package:  grid
## Loading required package:  biomaRt
## Loading required package:  Guiz
## Loading required package:  S4Vectors
## Loading required package:  stats4
## 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 objects are masked from 'package:stats':
##
##    IQR, mad, xtabs

## The following objects are masked from 'package:base':
##
##    Filter, Find, Map, Position, Reduce, anyDuplicated, append, as.data.frame,
##    cbind, colnames, do.call, duplicated, eval, evalq, get, grep, grepl,
##    intersect, is.unsorted, lapply, lengths, mapply, match, mget, order, paste,
##    pmax, pmax.int, pmin, pmin.int, rank, rbind, rownames, sapply, setdiff,
##    sort, table, tapply, union, unique, unsplit

##
## Attaching package:  'S4Vectors'

## The following objects are masked from 'package:base':
##
##    colMeans, colSums, expand.grid, rowMeans, rowSums

## Loading required package:  IRanges

## Loading required package:  GenomicRanges

## Loading required package:  GenomeInfoDb

## Loading required package:  psych

##
## Attaching package:  'psych'

## The following object is masked from 'package:IRanges':
##
##    reflect

## Warning:  replacing previous import 'ggplot2::Position' by 'BiocGenerics::Position' when
loading 'ggbio'
```

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, use standard Bioconductor command line, e.g.

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

3.2 Install the version of coMET from gitHub

Another way to install coMET is to download the master package from gitHUB https://github.com/TiphaineCMartin/coMET or the devel package https://github.com/TiphaineCMartin/coMet/tree/devel. Once downloaded use command line:

```
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 EWAS (or other omic-WAS) results, annotation tracks, and correlations between features. This version is installed in the Shiny web-service. Currently, it is formated only to visualise human data.
2. **comet** is the generic function that allows us to visualise quickly EWAS results, annotation tracks, and correlations between features. Users can visualise more personalised annotation tracks and give multiple extra EWAS/omic-WAS results to plot.
3. **comet.list** is an additional function that allows us to extract the values of correlations, the pvalues, and estimates and confidence intervals for all datapoints that surpass a particular threshold.

The functions can read the data input files, but it is also possible to use data frames within R for all data input except for the configuration file. The latter can be achieved with the two functions **comet** and **comet.list**. The structure of the data frames (number of columns, type, format) follows the same rules as for the data input files (cf. section "File formats").

5 File formats

There are five types of files that can be given by the user to produce the plot:

1. Info file is defined in the option **mydata.file**. This is mandatory and has to be in tabular format with a header.
2. Correlation file is defined in the option **cormatrix.file**. This is mandatory and has to be in tabular format with a header.
3. Extra info files are defined in the option **mydata.file.large**. This is optional, and if provided has to be 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 **biofeat.user.type** and **biofeat.user.type.plot**.
5. Configuration file contains the values of these options instead of defining these by command line. Each line in the file is one option. The name of the option is in capital letters and is separated by 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)

This file is mandatory and has to be in tabular format with a header. The name of features has to start by a letter. Info files can be a list of CpG sites with/without Beta value (for example 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 can be either a numeric value (positive or negative values) or only direction sign ("+", "-"). The number of columns and their types are defined by 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 the info file:

1. **site**: 4 columns with a header:
   (a) Name of omic feature
   (b) Name of chromosome
   (c) Position of omic feature
   (d) P-value of omic feature
2. **region**: 5 columns with a header:
   (a) Name of omic feature
   (b) Name of chromosome
   (c) Start position of omic feature
   (d) End position of omic feature
   (e) P-value of omic feature
3. **site_asso**: 5 columns with a header:
   (a) Name of omic feature
   (b) Name of chromosome
   (c) Position of omic feature
   (d) P-value of omic feature
   (e) Direction of association related to this omic feature. This can be the sign or an actual value of association effect size.
4. **region_asso**: 6 columns with a header:
   (a) Name of omic feature
   (b) Name of chromosome
   (c) Start position of omic feature
   (d) End position of omic feature
   (e) P-value of omic feature
   (f) Direction of association related to this omic feature. This can be the sign or an actual value of association effect size.

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

This file is mandatory and has to be 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)

This file is optional file and if provided has to be in tabular format with an header. The name of features has to start by a letter. The extra info files can be described in the option **mydata.large.file** and their format in **mydata.large.format**. More than one extra info file can be used, each should be 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

Each line in the file is one option. The name of the option is in lowercase letters and is separated by its value by "=" without space. If there are multiple values such as for the option **list.tracks** or options for additional data, these need to be separated them by a "comma" withou space. 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 on a regional Manhattan plot.
   - **pval.threshold**: Significance threshold to be displayed as a red dashed line

- **pval.threshold2**: Another Significance threshold (optional)
- **disp.pvalueplot**: Value can be TRUE or FALSE. Used to either display or hide Manhattan plot.
- **disp.beta.association**: Value can be TRUE or FALSE. Used to show the effect size.
- **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 colour of symbol is the colour of co-methylation pattern between the point and the reference site; if TRUE, the effect direction is shown. If the association is positive, the colour is the one defined with the option **color.list**. On the other hand, if the association is negative, the colour is the inverse to that selected.
- **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 as a continuous line with the colour 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 to be visualised. Tracks currently available: geneENSEMBL, CGI, ChromHMM, DNAse, RegENSEMBL, SNP, transcriptENSEMBL, SNPstoma, SNPstru, SNPstrustoma, ISCA, COSMIC, GAD, ClinVar, GeneReviews, GWAS, ClinVarCNV, GCcontent, genesUCSC, xenogenesUCSC, metQTL, eQTL, BindingMotifsBiomart, chromHMM_RoadM miRNATargetRegionsBiomart, OtherRegulatoryRegionsBiomart, RegulatoryEvidenceBiomart, RegulatorySegmentsBiomart and segmentalDupsUCSC. 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. The integration of plots from ggbio and trackviewer can be sometimes not really perfect. It is better to create plots from Gviz and use tracks.gviz
3. The **lower panel** shows the correlation between selected CpG sites in the genomic region (heatmap).
   - **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 colour schemes (heat, bluewhitered, cm, topo, gray, bluetored)
   - **disp.cormatrixmap** : logical option TRUE or FALSE. TRUE (default), if FALSE correlation matrix is not shown)
   - **cormatrix.conf.level** : Alpha level for the confidence interval. Default value= 0.05. CI will be the alpha/2 lower and upper values.)
   - **cormatrix.sig.level** : Significant level to visualise the correlation. If the correlation has a pvalue under the significant level, the correlation will be colored in "goshwhite", else the color is related to the correlation level and the color scheme choosen.Default value =1.)
   - **cormatrix.adjust** : Indicates which adjustment for multiple tests should be used. "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".Default value="none".)

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
configfile <- file.path(extdata, "config_cyp1b1_zoom_4webserver_Grch38.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=38066017
## 12                                           end=38108036
## 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                                cormatrix.conf.level=0.05
## 24                                    cormatrix.sig.level=1
## 25                                    cormatrix.adjust=none
## 26                                      disp.phys.dist=TRUE
## 27                                      disp.color.bar=TRUE
## 28                                         disp.legend=TRUE
## 29 list.tracks=geneENSEMBL,CGI,ChromHMM,DNAse,RegENSEMBL,SNP
## 30                                     disp.mult.lab.X=FALSE
## 31                                           image.type=pdf
## 32   image.title="Example a-DMR in CYP1B1 in Adipose tissue"
## 33                     image.name=cyp1b1_zoom_plus_name_expr
## 34                                           image.size=3.5
## 35                                              genome=hg38
## 36                       dataset.geneE=hsapiens_gene_ensembl
```

6 Creating a plot like the webservice: comet.web

6.1 coMET plot: usage and plot like in the webservice

The user can create a coMET plot 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 **??**.

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
myinfofile <- file.path(extdata, "cyp1b1_infofile_Grch38.txt")
myexpressfile <- file.path(extdata, "cyp1b1_infofile_exprGene_region_Grch38.txt")
mycorrelation <- file.path(extdata, "cyp1b1_res37_rawMatrix.txt")
configfile <- file.path(extdata, "config_cyp1b1_zoom_4webserver_Grch38.txt")
comet.web(config.file=configfile, mydata.file=myinfofile,
          cormatrix.file=mycorrelation ,mydata.large.file=myexpressfile,
          print.image=FALSE,verbose=FALSE)
```

6.2 Hidden values of comet.web function

Hidden values of **comet.web** function are shown in the section. If these values do not correspond to what you want to visualise, you need to use the function **comet**, as a more generic option.

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
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

Continued on next page

Table 1 – continued from previous page

Option	Value
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

```
## Warning in .local(x, ..., na.rm = na.rm):  'na.rm' argument is ignored
```

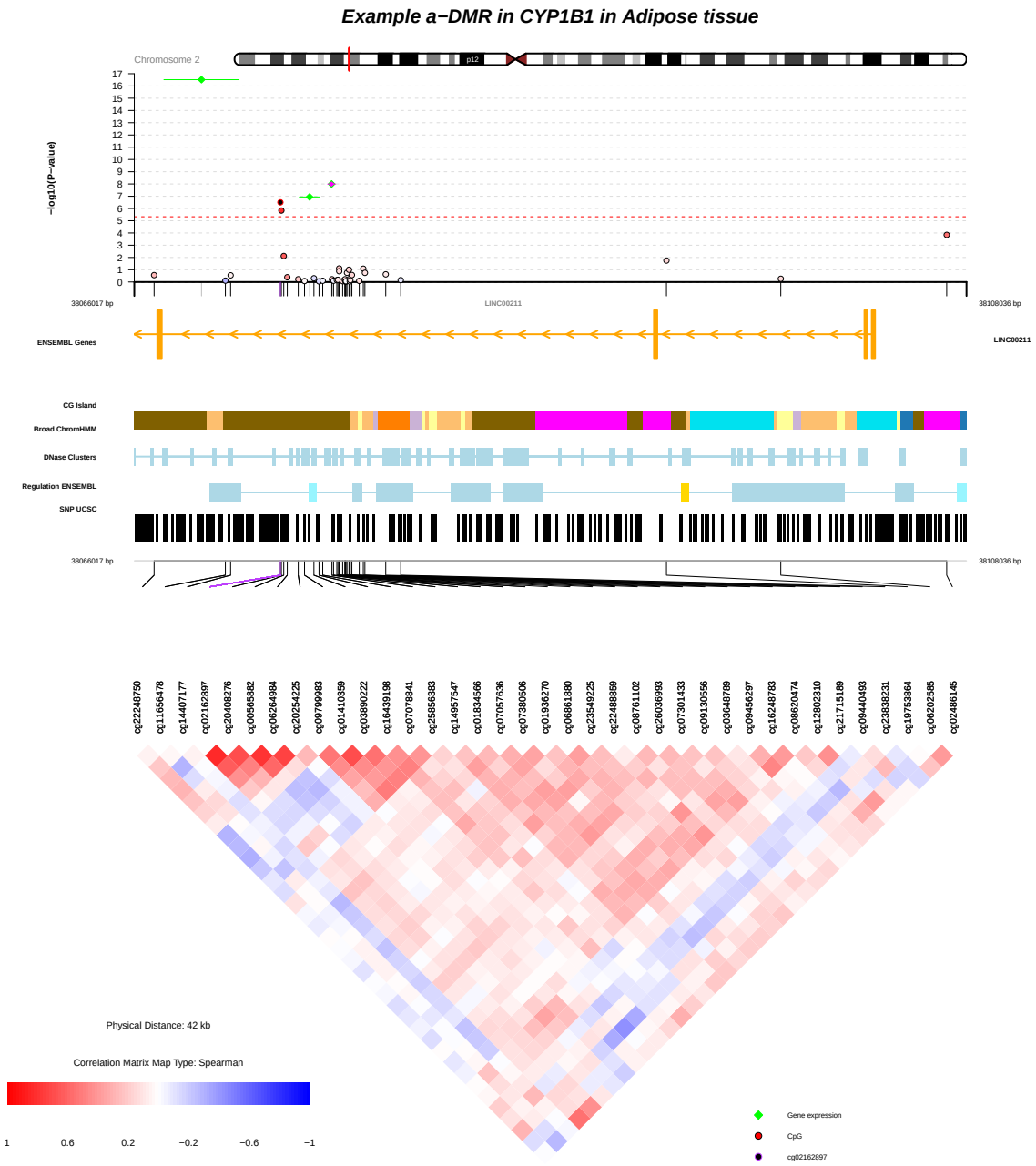

Figure 1: Plot with comet.web function.

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 **??**. 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 Input from data files

In this figure **??**, we create 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 <- "*"

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

genetrack <-genes_ENSEMBL(gen,chrom,start,end,showId=TRUE)
snptrack <- snpBiomart_ENSEMBL(gen,chrom, start, end, dataset="hsapiens_snp_som",showId=FALSE)
cpgIstrack <- cpgIslands_UCSC(gen,chrom,start,end)

prombedFilePath <- file.path(extdata, "/RoadMap/regions_prom_E063.bed")
promRMtrackE063<- DNaseI_RoadMap(gen,chrom,start, end, prombedFilePath,
                            featureDisplay='promotor', stacking_type="squish")

bedFilePath <- file.path(extdata, "RoadMap/E063_15_coreMarks_mnemonics.bed")
chromHMM_RoadMapAllE063 <- chromHMM_RoadMap(gen,chrom,start, end, bedFilePath, featureDisplay = "

listgviz <- list(genetrack,snptrack,cpgIstrack,promRMtrackE063,chromHMM_RoadMapAllE063)

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)
```

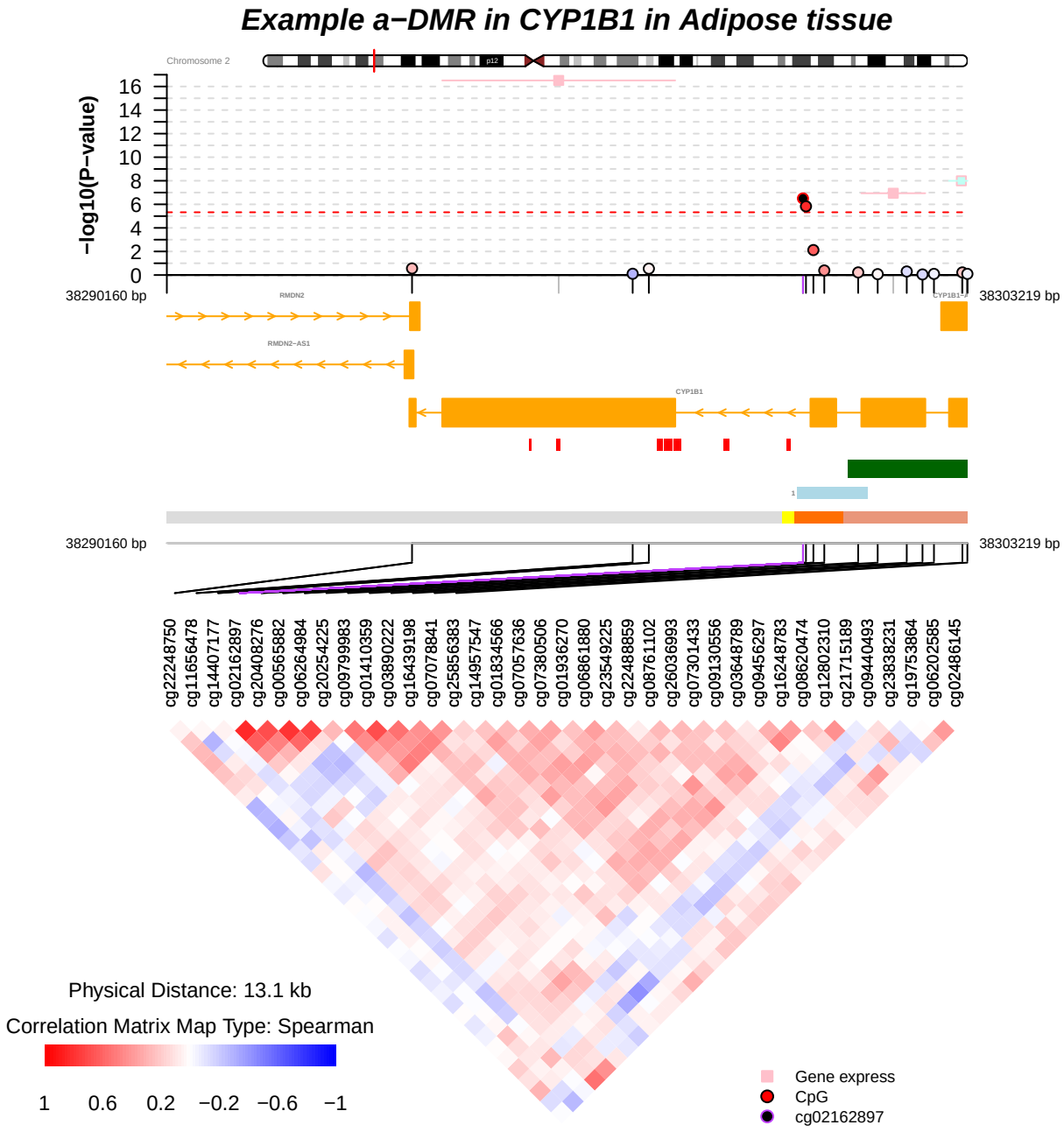

Figure 2: Plot with comet function from files.

7.1.2 coMET plot using input from a data frame

In this figure **??**, we visualize the same data as in figure **??**, but the data is in data frame format and not read in from an input file.

In addition, if the user would like to visualise only the correlations between CpG sites with P-value less than or equal to 0.05 in the upper plot, this option can be included. The correlations with a P-value greater than 0.05 can have the colour "goshwhite" whereas the other correlations will be displayed using a colour related to the correlation level. Conversely, in the P-value plot (upper plot), the points of each omic feature have their colours related to their correlations with the reference omic feature without taking into account the P-value associated with the correlation matrix.

Eventually, we increase the size of font using the option **fontsize.gviz**

```
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 <-genes_ENSEMBL(gen,chrom,start,end,showId=TRUE)
snptrack <- snpBiomart_ENSEMBL(chrom, start, end, dataset="hsapiens_snp_som",showId=FALSE)
iscatrack <-ISCA_UCSC(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",
```

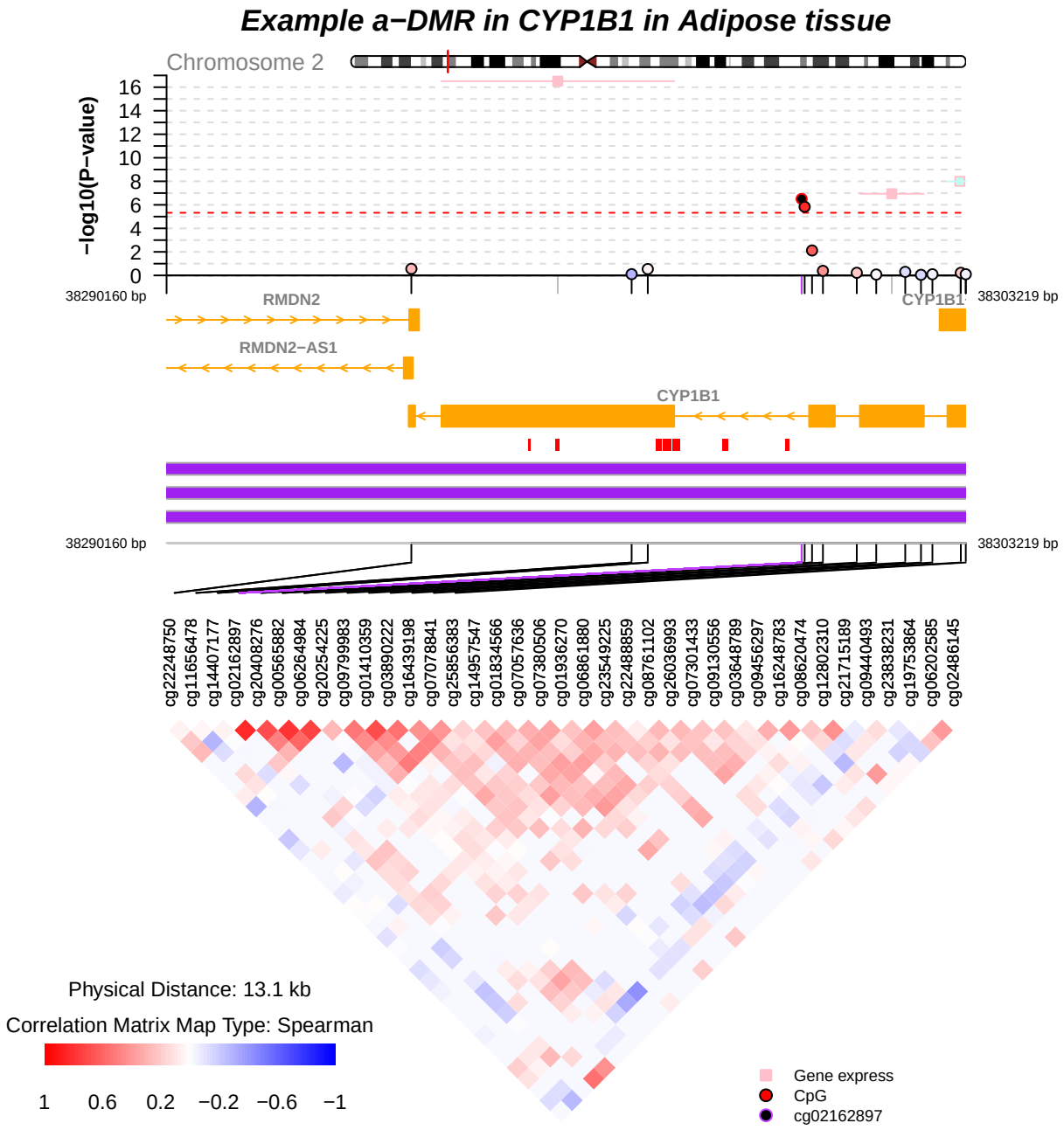

Figure 3: Plot with comet function from matrix data and with a pvalue threshold for the correlation between omics features (here CpG sites).

```
    mydata.large.file=listmatrix.expression, mydata.large.type="listdataframe",
    fontsize.gviz =12,
    tracks.gviz=listgviz,verbose=FALSE, print.image=FALSE)
```

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 **??**.

```
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 <-genes_ENSEMBL(gen,chrom,start,end,showId=FALSE)
snptrack <- snpBiomart_ENSEMBL(chrom, start, end,
                        dataset="hsapiens_snp_som",showId=FALSE)
strutrack <- structureBiomart_ENSEMBL(chrom, start, end,
                               strand, dataset="hsapiens_structvar_som")
clinVariant<-ClinVarMain_UCSC(gen,chrom,start,end)
clinCNV<-ClinVarCnv_UCSC(gen,chrom,start,end)
gwastrack <-GWAScatalog_UCSC(gen,chrom,start,end)
geneRtrack <-GeneReviews_UCSC(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",
      fontsize.gviz =12,
      tracks.gviz=listgviz, verbose=FALSE, print.image=FALSE,disp.pvalueplot=FALSE)
```

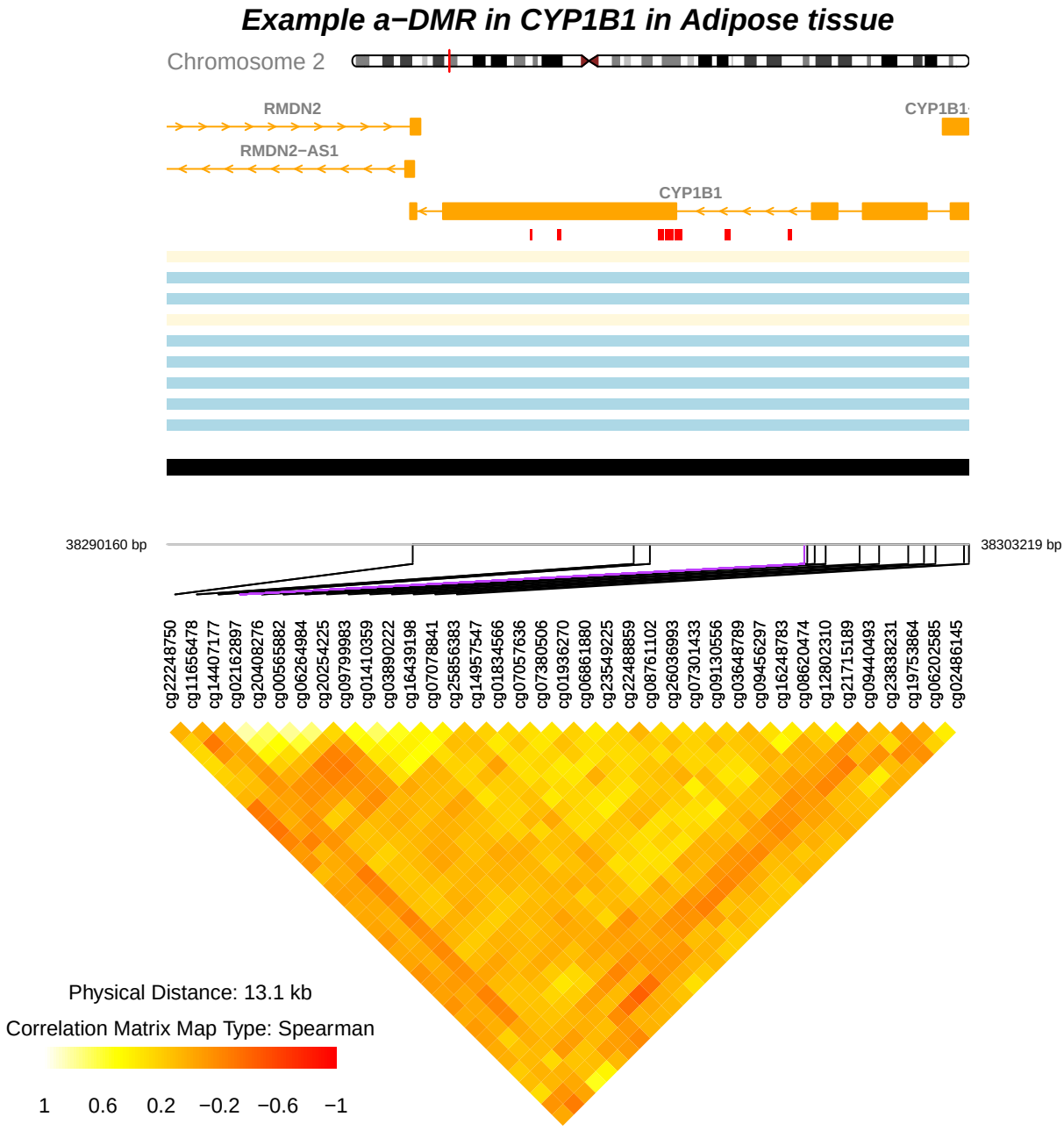

Figure 4: Plot with comet function without pvalue plot.

7.3 coMET plot: Manhattan plot and anonation track

It is possible to visualise only The Manhattan plot and the annotation tracks. In this case, we need to use the option `disp.cormatrixmap = FALSE`, for example see Figure **??**.

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
configfile <- file.path(extdata, "config_cyp1b1_zoom_4nomatrix.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 <-genes_ENSEMBL(gen,chrom,start,end,showId=FALSE)
snptrack <- snpBiomart_ENSEMBL(chrom, start, end,
                        dataset="hsapiens_snp_som",showId=FALSE)
strutrack <- structureBiomart_ENSEMBL(chrom, start, end,
                               strand, dataset="hsapiens_structvar_som")
clinVariant<-ClinVarMain_UCSC(gen,chrom,start,end)
clinCNV<-ClinVarCnv_UCSC(gen,chrom,start,end)
gwastrack <-GWAScatalog_UCSC(gen,chrom,start,end)
geneRtrack <-GeneReviews_UCSC(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",
      fontsize.gviz =12, font.factor=3,
      tracks.gviz=listgviz, verbose=FALSE, print.image=FALSE)
```

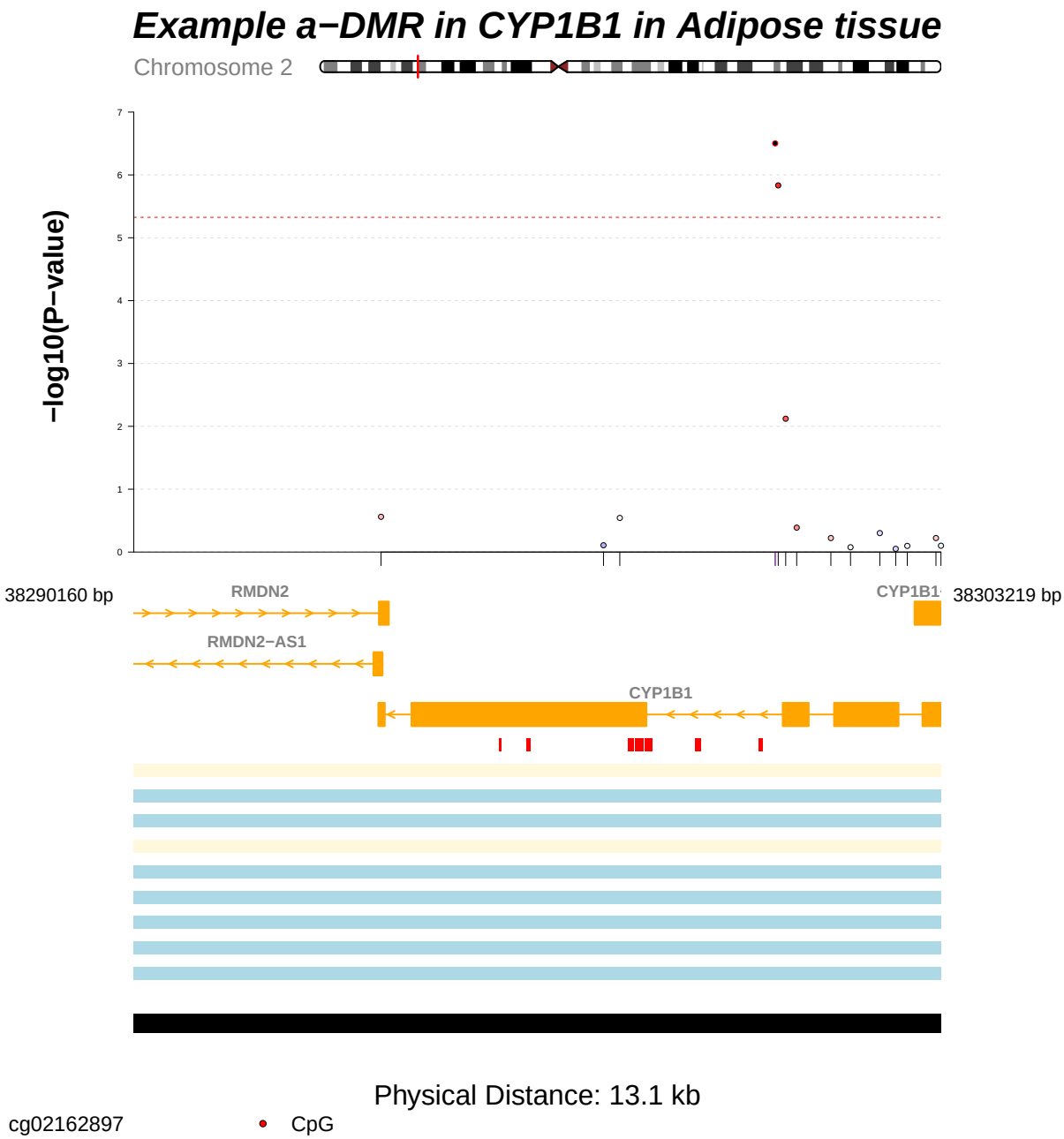

Figure 5: Plot with comet function without the correlation matrix.

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 significance 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 (e.g. 0.05 (**cormatrix.conf.level**))
5. the alpha/2 upper value (e.g. 0.05 (**cormatrix.conf.level**))
6. the pvalue
7. the pvalue adjusted with the method selected (e.g. 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",
           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 Annotation tracks

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

1. **UcscTrack**. Different UCSC tracks can be selected for visualisation from the table Browser of UCSC http://genome-euro.ucsc.edu/cgi-bin/hgTables?hgsid=202842745_DIvit14QO0G6ZPpLoEVABG8aqfrm&clade=mammal&org=Human&db=hg19&hgta_group=varRep&hgta_track=cpgIslandExt&hgta_table=0&hgta_regionType=genome&position=chr6%3A32726553\discretionary{-}{}{}32727053&hgta_outputType=primaryTable&outFileName=
2. **BiomartGeneRegionTrack**. A connection should be established to the Biomart database to visualise the genetic elements.
3. **DataTrack**. This allows the visualisation of numerical data.
4. **AnnotationTrack**. This allows the visualisation of any annotation data.

For more information consult the user guide for Gviz.

9.1 Ensembl

The Ensembl project [**?**] produces genome databases for vertebrates and other eukaryotic species, and makes this information freely available online http://www.ensembl.org/index.html. A set of wrap R functions were created to extract data from Ensembl BioMart for human genome using Ensembl REST [**?**], but they can be extended to other genomes. You can ask help to *tiphaine.martin@kcl.ac.uk*.

This is the list of R functions created in coMET to visualise ENSEMBL data. Below described the colors of tracks and specific characteristics of some annotation tracks.

- **bindingMotifsBiomart_ENSEBML** : Visualise the binding motifs in the genomic region of interest
- **genes_ENSEBML** : Visualise the genes from ENCODE in the genomic region of interest
- **genesName_ENSEBML** : Visualise the name of genes from ENCODE in the genomic region of interest
- **interestGenes_ENSEBML** : Visualise the genes from ENCODE in the genomic region of interest with a specific color for genes of interest
- **interestTranscript_ENSEBML** : Visualise the transcripts from ENCODE in the genomic region of interest with a specific color for exons of interest
- **miRNATargetRegionsBiomart_ENSEBML** : Visualise the miRNA target regions in the genomic region of interest
- **otherRegulatoryRegions_ENSEBML** : Visualise the other regulatory regions in the genomic region of interest
- **regulationBiomart_ENSEBML** (obselet function): Visualise the other regulatory regions in the genomic region of interest
- **regulatoryEvidenceBiomart_ENSEBML** : Visualise the regulatory evidence regions in the genomic region of interest
- **regulatoryFeaturesBiomart_ENSEBML** : Visualise the regulatory features regions in the genomic region of interest
- **regulatorySegmentsBiomart_ENSEBML** : Visualise the regulatory segment regions in the genomic region of interest
- **snpBiomart_ENSEBML** : Visualise the SNPs in the genomic region of interest
- **structureBiomart_ENSEBML** : Visualise the structural variations in the genomic region of interest
- **transcript_ENSEBML** : Visualise the transcripts in the genomic region of interest

Below described the colors of tracks and specific characteristics of some annotation tracks.

9.1.1 Genes and transcripts from Ensembl

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

It is possible to chagne the colour of some exsons by using the function *interestGenesENSEMBL* or *interestTranscriptENSEMBL*. The elements and the colours to be displayed must be given as list. An example is given below:

```
gen <- "hg38"
chr <- "chr15"
start <- 75011669
end <- 75019876
interestfeatures <- rbind(c("75011883","75013394","bad"),c("75013932","75014410","good"))
interestcolor <- list("bad"="red", "good"="green")

interestgenesENSMBLtrack<-interestGenes_ENSEMBL(gen,chr,start,end,interestfeatures,
                                                interestcolor,showId=TRUE)
plotTracks(interestgenesENSMBLtrack, from=start, to=end)
```

9.1.2 Regulatory elements from Ensembl

This function is now obselet in coMET as Ensembl have restructured their databases due to the new version of the genome GRCh38. The same data is now available by using the function 'RegulatoryFeaturesBiomart'.

The colors were :

9.1.3 structureBiomart from Ensembl

Listed below are the colours for somatic structural variation and structural variation.

9.1.4 miRNA Target Regions from Ensembl

The colour of the miRNA target regions is set to Plum4 (hex code: #8B668B)

9.1.5 Binding Motif Biomart from Ensembl

Listed on the next page are the colours used for the different types of binding motifs. The frequency shown is that found in GRCh38 (hg38). Motifs with red text are found only in GRCh37 (hg19), motifs with blue text are found only in GRCh38 (hg38)

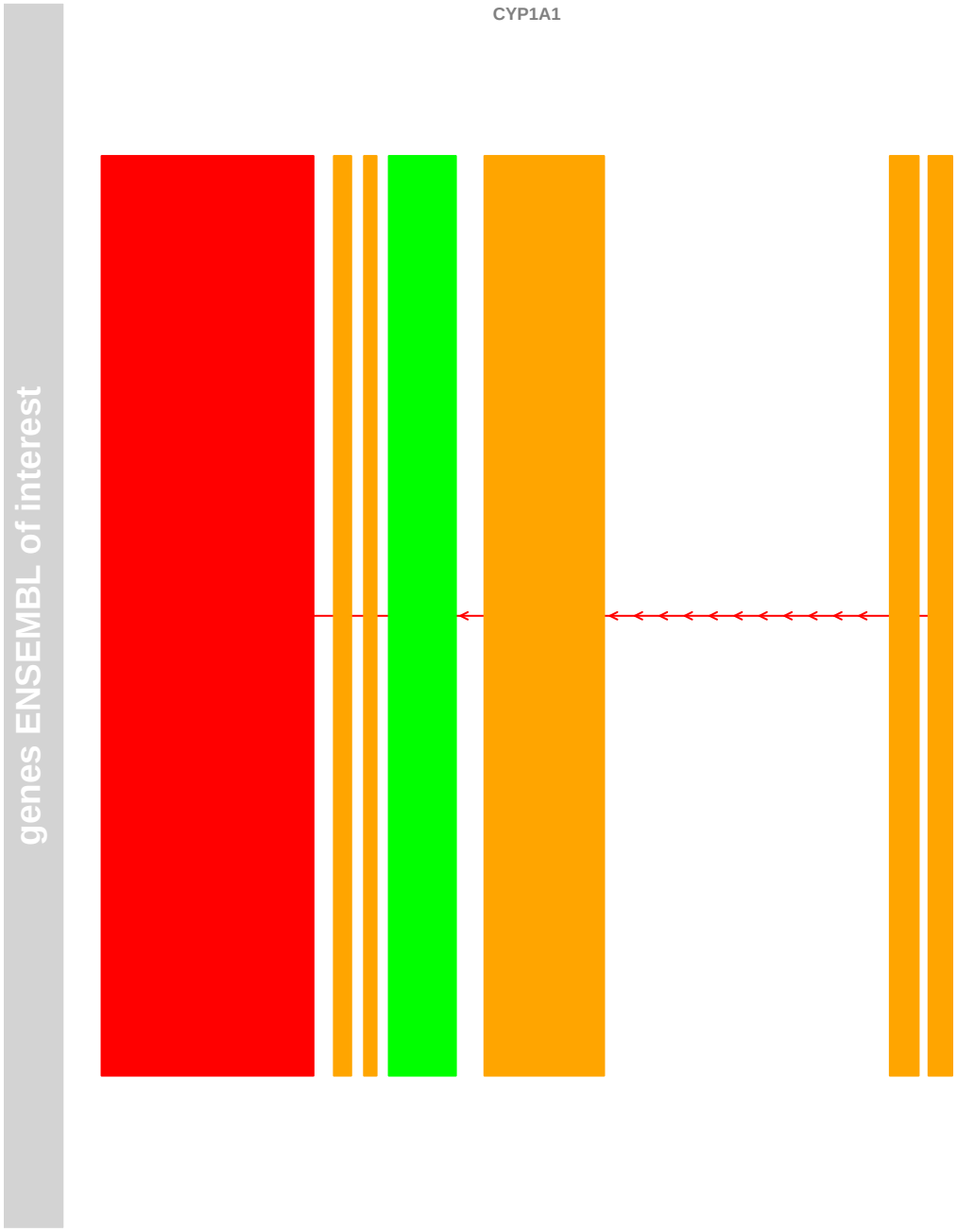

Figure 6: Plot genes with different colors according user's choice.

9.1.6 Other Regulatory Regions Biomart from Ensembl

Listed below are the colours used for the different types of regulatory regions. The frequency shown is that found in GRCh38 (hg38).

9.1.7 Regulatory Features Biomart from Ensembl

Listed below are the colours used for the different types of regulatory features The frequency shown is that found in GRCh38 (hg38).

9.1.8 Other Regulatory Segments Biomart from Ensembl

Listed below are the colours used for the different types of regulatory segments. The frequency shown is that found in GRCh38 (hg38). Segments with red text are found only in GRCh37 (hg19)

9.1.9 Binding Motif Biomart from Ensembl

Listed on the next 3 pages are the colours used for the different types of regulatory evidence elements. The frequency shown is that found in GRCh37 (hg19). At the current time this track has not been optimised for GRCh38 (hg38) meaning any elements found exclusively in GRCh38 do not have an assigned colour and will be displayed in the default track colour of Gviz.

9.2 UCSC

the UCSC Genome Browser [**?**] website http://genome-euro.ucsc.edu/ contains the reference sequence and working draft assemblies for a large collection of genomes.

This is the list of R wrapping functions of some tracks found in UCSC genome browser. Below described the colors of tracks and specific characteristics of some annotation tracks.

- **chromatinHMMAll_UCSC** : Visualise the chromHMM Broad found in UCSC genome browser of all tissues in the genomic region of interest.
- **chromatinHMMOne_UCSC** : Visualise the chromHMM Broad found in UCSC genome browser of the tissue of interest in the genomic region of interest.
- **ClinVarCnv_UCSC** : Visualise clinical CNVs found in ClinVar tracks of UCSC genome browser in the genomic region of interest.
- **ClinVarMain_UCSC** : Visualise clinical SNPs found in ClinVar tracks of UCSC genome browser in the genomic region of interest.
- **CoreillCNV_UCSC** : Visualise CNV found in Coreil tracks of UCSC genome browser in the genomic region of interest.
- **COSMIC_UCSC** : Visualise SNPs found in COSMIIC tracks of UCSC genome browser in the genomic region of interest.
- **cpgIslands_UCSC** : Visualise CpG Island found in CpGIsland tracks of UCSC genome browser in the genomic region of interest.
- **DNAse_UCSC** : Visualise clinical CNV found in ClinVar tracks of UCSC genome browser in the genomic region of interest.
- **GAD_UCSC** : Visualise genes found in GAD tracks of UCSC genome browser in the genomic region of interest.
- **gcContent_UCSC** : Visualise GC content found in UCSC genome browser in the genomic region of interest.
- **GeneReviews_UCSC** : Visualise clinical genes found in GeneReviews tracks of UCSC genome browser in the genomic region of interest.
- **GWAScatalog_UCSC** : Visualise SNPS found in GWAS catalog tracks of UCSC genome browser in the genomic region of interest.
- **HistoneAll_UCSC** : Visualise histone patterns found in UCSC genome browser of all tissues in the genomic region of interest.
- **HistoneOne_UCSC** : Visualise histone patterns found in UCSC genome browser of one tissue of interest in the genomic region of interest.
- **ISCA_UCSC** (obselete) : Visualise clinical CNV found in UCSC genome browser in the genomic region of interest.
- **knownGenes_UCSC** : Visualise known genes found in UCSC genome browser in the genomic region of interest.
- **refGenes_UCSC** : Visualise reference genes found in UCSC genome browser in the genomic region of interest.
- **repeatMasker_UCSC** : Visualise repeat elements found in UCSC genome browser in the genomic region of interest.
- **segmentalDups_UCSC** : Visualise segmental duplcations found in UCSC genome browser in the genomic region of interest.
- **snpLocations_UCSC** : Visualise SNPs found in UCSC genome browser in the genomic region of interest.
- **xenorefGenes_UCSC** : Visualise xeno reference genes found in UCSC genome browser in the genomic region of interest.

9.2.1 ChromHMM from UCSC

For this function there are two possible colour schemes to choose from. The selection between schemes is made with the variable 'colour'. The default scheme is 'coMET', the colours chosen have been selected so that different elements can be easily distinguished. The second scheme is 'UCSC', these are the set colours used by UCSC, in certain plots it may be difficult to distinguish elements apart. These UCSC colours are correct at the time this document was writtern however if these change in the future and this is not reflected here please contact us.

the colours used in both schemes are listed below:

9.2.2 ISCA track (obselete database)

International Standards of Cytogenomic Arrays Consortium defined a set of phenotypes for CNVs. Different colours are defined to represent them. This database is not more accessible from UCSC.

9.2.3 Other potential data from UCSC

You can access to other data via UCSC track hub [**?**] :

- Other tracks and table accessible to UCSC genome browser https://genome.ucsc.edu/cgi-bin/hgTables?hgsid=444062899_lxuSrw4J9exVt1OafMuY4LDbVs1F&clade=mammal&org=Human&db=hg19&hgta_group=allTracks&hgta_track=knownGene&hgta_table=0&hgta_regionType=genome&position=chr21%3A33031597-33041
outputType=primaryTable&hgta_outFileName=
- Track HUB of UCSC genome browser https://genome-euro.ucsc.edu/cgi-bin/hgHubConnect?hubUrl=http%3A%2F%2Ffantom.gsc.riken.jp%2F5%2Fdatahub%2Fhub.txt&hgHubConnect.remakeTrackHub=on&redirect=manual&source=genome.ucsc.edu

and use DataTrack or AnnotationTrack or UCSCTrack of Gviz to visuaslise them.

9.3 ROADMAP epigenomics project

ROADMAP epigenomics projects http://www.roadmapepigenomics.org/ [**?**] aims to produce a public resource of human epigenomic data to catalyze basic biology and disease-oriented research. The project has generated high-quality, genome-wide maps of several key histone modifications, chromatin accessibility, DNA methylation and mRNA expression across 100s of human cell types and tissues (111 consolidated epigenomes from the Roadmap Epigenomics Project and 16 epigenomes from The Encyclopedia of DNA Elements (ENCODE) project).

Release 9 of the compendium contains uniformly pre-processed and mapped data from multiple profiling experiments (technical and biological replicates from multiple individuals and/or datasets from multiple centers) spanning 183 biological samples and 127 consolidated epigenomes.

More information on each type data are on the site of ROADMAP http://egg2.wustl.edu/roadmap/web_portal/index.html and the meta-data on different tissues (more for correspondance between Epigenome ID (EID) and the standartized epigenome name), you need to look at this spreadsheet https://docs.google.com/spreadsheets/d/1yikGx4MsO9Ei36b64yOy9Vb6oPC5IBGlFbYEt-N6gOM/edit#gid=15

The current data are done on Release 9. The data are mapped on the reference genome **hg19**. Below described the colors of tracks and specific characteristics of some annotation tracks.

- **chromHMM_RoadMap** : Visualisation of chromatin states defined in RoadMap project
- **dgfootprints_RoadMap**: Visualisation of DNA motif positional bias in digital genomic Footprinting Sites
- **DNaseI_RoadMap** : Visualisation of promoter/enhancer regions

9.3.1 Chromatin state

There are 3 chromatin states defined in RoadMap project (15 states, 18 states and 25 states). For 18 and 25 states, there are the choice beteen 2 set of colors. First, the colors defined by RoadMap and second, the colors defined by us for a better differentiation between states.

you can use *chromHMM_RoadMap* to visualise chromatin state in :

- 15-states, go to http://egg2.wustl.edu/roadmap/data/byFileType/chromhmmSegmentations/ChmmModels/coreMarks/jointModel/final/ and select the MNEMONICS BED FILES, where bins with the same state label are merged and a label is assigned to the entire merged regions, related to your tissue of interest.
- 18-states, go to http://egg2.wustl.edu/roadmap/data/byFileType/chromhmmSegmentations/ChmmModels/core_K27ac/jointModel/final/ and select the MNEMONICS BED FILES, where bins with the same state label are merged and a label is assigned to the entire merged regions, related to your tissue of interest .
- 25-states, go to http://egg2.wustl.edu/roadmap/data/byFileType/chromhmmSegmentations/ChmmModels/imputed12marks/jointModel/final/ and select your tissue of interest.

You can have more information about these data from ROADMAP website http://egg2.wustl.edu/roadmap/web_portal/chr_state_learning.html#core_15state.

You can visualise this bed using the function *chromHMM_RoadMap* and you can choice the color between *roadmap15*, *roadmap18*, *comet18*, *roadmap25* and *comet25*.

Below you can find the color code for each state depending if 15-,18- or 25-state

Listed below are the colours used for the different elements contained in ROADmap data with 15 states.

Listed below are the colours used for the different elements contained in ROADmap data with 18 states with RoadMap colors.

Listed below are the colours used for the different elements contained in ROADmap data with 18 states with coMET colors.

Listed below are the colours used for the different elements contained in ROADmap data with 25 states with RoadMap colors.

Listed below are the colours used for the different elements contained in ROADmap data with 25 states with coMET colors.

9.3.2 DNA Motif Positional Bias in Digital Genomic Footprinting Sites

The Digital Genomic Footprinting (DGF) sites in each cell type can be visualised using the function *dgfootprints_RoadMap* using the file of DNase/DGF Footprint calls http://egg2.wustl.edu/roadmap/data/byDataType/dgfootprints/

9.3.3 DNaseI-accessible regulatory regions

Using the core 15-state chromatin state model across any of the 111 Roadmap reference epigenomes, and focusing on states TssA, TssAFlnk, and TssBiv for promoters, and EnhG, Enh, and EnhBiv for enhancers, and state BivFlnk (flanking bivalent Enh/Tss) for ambiguous regions, 3 set of data were constructed. The data can be visualised using the function *DNaseI_RoadMap* with the good name of data (variable *featureDisplay*) like in Fig. **??**:

- for **promoter** regions the file of tissue of interest http://egg2.wustl.edu/roadmap/data/byDataType/dnase/BED_files_prom/ or RData files containing matrice of chromatin state call for promoter. Thus, user can select for different tissues.
- for **enhancer** regions the file of tissue of interest http://egg2.wustl.edu/roadmap/data/byDataType/dnase/BED_files_enh/
- for **dyadic** promoter/enhancer region the file of tissue of interest http://egg2.wustl.edu/roadmap/data/byDataType/dnase/BED_files_dyadic/

```
chr<-"chr2"
start <- 38290160
end <- 38303219
```

```
gen<-"hg19"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
prombedFilePath <- file.path(extdata, "/RoadMap/regions_prom_E001.bed")

promRMtrack<- DNaseI_RoadMap(gen,chr,start, end, prombedFilePath,
                             featureDisplay='promotor', type_stacking="squish")

enhbedFilePath <- file.path(extdata, "/RoadMap/regions_enh_E001.bed")

enhRMtrack<- DNaseI_RoadMap(gen,chr,start, end, enhbedFilePath,
                            featureDisplay='enhancer', type_stacking="squish")

dyabedFilePath <- file.path(extdata, "/RoadMap/regions_dyadic_E001.bed")

dyaRMtrack<- DNaseI_RoadMap(gen,chr,start, end, dyabedFilePath,
                            featureDisplay='dyadic', type_stacking="squish")

genetrack <-genes_ENSEMBL(gen,chr,start,end,showId=TRUE)

listRoadMap <- list(genetrack,promRMtrack,enhRMtrack,dyaRMtrack)
plotTracks(listRoadMap, chromosome=chr,from=start,to=end)
```

9.3.4 Processed data and Imputed data

BED and BigWIG file can be visualised with DataTrack objects from files of Gviz package. The data are in http://www.genboree.org/EdaccData/Release-9/sample-experiment/ and http://www.genboree.org/EdaccData/Release-9/experiment-sample/ or go to http://egg2.wustl.edu/roadmap/web_portal/processed_data.html for processed data or to http://egg2.wustl.edu/roadmap/web_portal/imputed.html#imp_sig for imputed data.

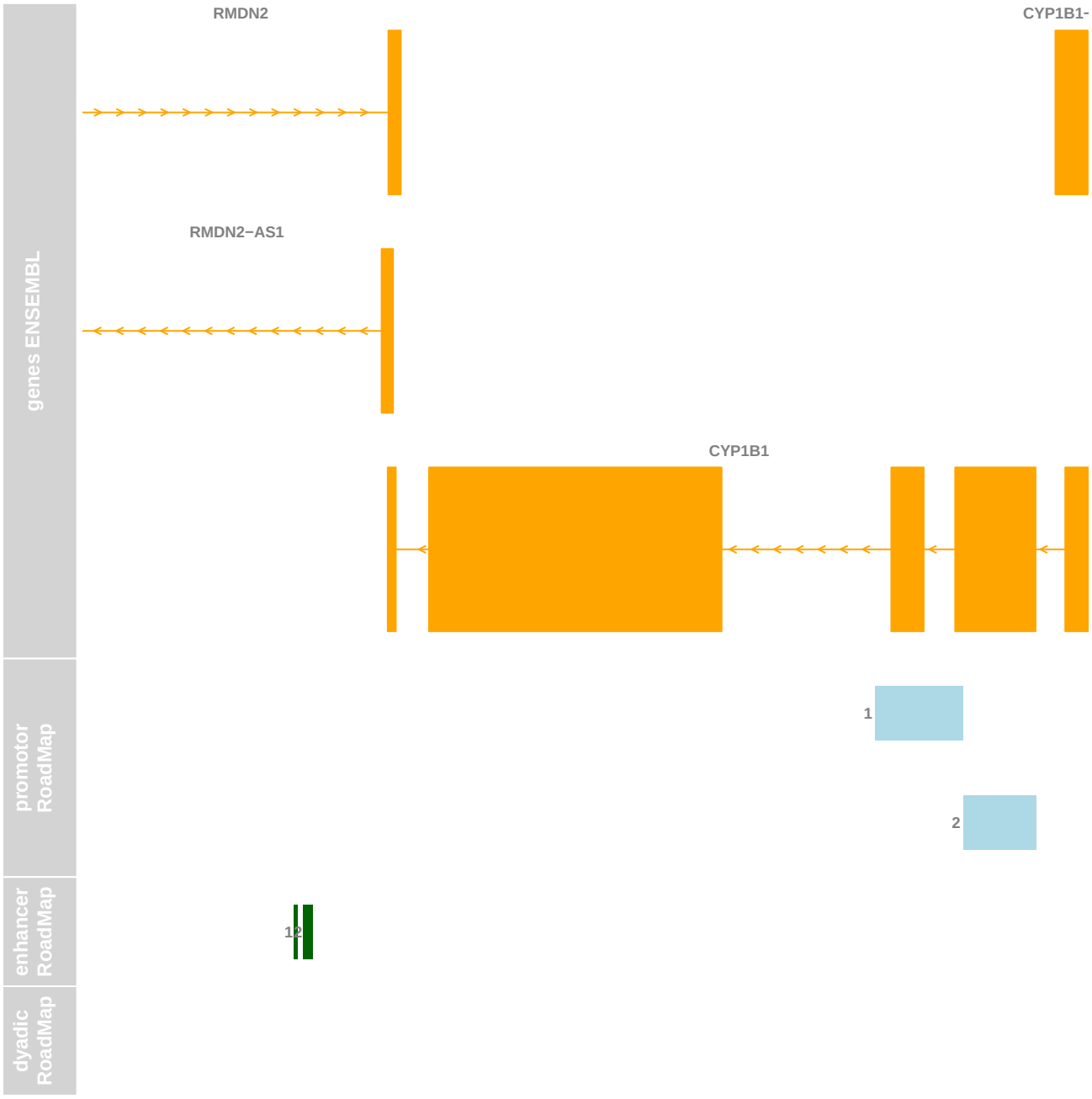

Figure 7: Plot of ROADMAP data.

9.4 ENCODE and GENCODE data

The ENCODE (Encyclopedia of DNA Elements) Consortium is an international collaboration of research groups funded by the National Human Genome Research Institute (NHGRI) https://www.encodeproject.org/. The goal of ENCODE is to build a comprehensive parts list of functional elements in the human genome, including elements that act at the protein and RNA levels, and regulatory elements that control cells and circumstances in which a gene is active.

Genes and transcripts of GENCODE are accessible from ENSEMBL biomart or can be visualised wtith GeneRegionTrack of Gviz. Other data are in BED or BAM format that can be visualised with Gviz tracks.

```
#Genes from GENCODE
chr<-3
start <- 132239976
end <- 132541303
gen<-"hg19"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
gtfFilePath <- file.path(extdata, "/GTEX/gencode.v19.genes.patched_contigs.gtf")
options(ucscChromosomeNames=FALSE)
grtrack <- GeneRegionTrack(range=gtfFilePath ,chromosome = chr, start= start,
                           end= end, name = "Gencode V19",
                           collapseTranscripts=TRUE, showId=TRUE,shape="arrow")
plotTracks(grtrack, chromosome=chr,from=start,to=end)
```

9.4.1 Predicting motifs and active regulators

You can browse known and discovered motifs for the ENCODE TF ChIP-seq datasets. The position of motifs can be visualised using the function **motif_ENCODE** using one of files from http://compbio.mit.edu/encode-motifs/ [**?**] such as http://compbio.mit.edu/encode-motifs/matches.txt.gz

```
#TF Chip-seq data
gen <- "hg19"
chr<-"chr1"
start <- 1000
end <- 329000
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "ENCODE/motifs1000_matches_ENCODE.txt")
motif_color <- file.path(extdata, "ENCODE/TFmotifs_colors.csv")

chipTFtrack <- ChIPTF_ENCODE(gen,chr,start, end, bedFilePath,
                             featureDisplay=c("AHR::ARNT::HIF1A_1",
                                              "AIRE_1","AIRE_2","AHR::ARNT_1"),
                             motif_color,type_stacking="squish",showId=TRUE)

plotTracks(chipTFtrack, chromosome=chr,from=start,to=end)
```

Figure 8: Plot of genes defined by GeneCode.

Figure 9: Plot ENCODE TF ChIP-seq datasets of ENCODE.

9.5 GTEx Portal

The Genotype-Tissue Expression (GTEx) [**?**] project aims to provide to the scientific community a resource with which to study human gene expression and regulation and its relationship to genetic variation. By analyzing global RNA expression within individual tissues and treating the expression levels of genes as quantitative traits, variations in gene expression that are highly correlated with genetic variation can be identified as expression quantitative trait loci, or eQTLs. The data are accessible via http://www.gtexportal.org/. A set of data are downloadable from http://www.gtexportal.org/home/datasets2 (need to have login).

The data were mapped on the reference genome **hg19**. Below described the colors of tracks and specific characteristics of some annotation tracks.

2 functions were created to visualise data from GTEx version 6:

1. **eQTL_GTEx** visualise eGene and significant snp-gene associations based on permutations in a tissue specific. The name of folder in GTEx version 6 is *GTEx_Analysis_V6_eQTLs.tar.gz*.
2. **geneExpression_GTEx** (need to update) visualise fully processed, normalized and filtered gene expression data, which was used as input into Matrix eQTL for eQTL discovery in a tissue specific. The name of folder in GTEx version 6 is *GTEx_Analysis_V6_eQTLInputFiles_geneLevelNormalizedExpression.tar.gz*
3. **GeneRegionTrack** from Gviz can visualise gene level model based on the GENCODE transcript model (cf. example below. Isoforms have been collapsed to single genes. The name of file in GTEx version 6 is *gencode.v19.genes.patched_contigs.gtf*.

```
## eQTL data
chr<-"chr3"
start <- 132239976
end <- 132541303
gen<-"hg19"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "/GTEX/eQTL_Uterus_Analysis_extract100.snpgenes")

eGTex<- eQTL_GTEx(gen,chr, start, end, bedFilePath, featureDisplay = 'all',
                  showId=TRUE, type_stacking="squish", just_group="left" )

eGTex_SNP<- eQTL_GTEx(gen,chr, start, end, bedFilePath,
                      featureDisplay = 'SNP', showId=FALSE,
                      type_stacking="dense", just_group="left")

#Genes from
gtfFilePath <- file.path(extdata, "/GTEX/gencode.v19.genes.patched_contigs.gtf")
options(ucscChromosomeNames=FALSE)
grtrack <- GeneRegionTrack(genome="hg19",range=gtfFilePath ,chromosome = chr,
                           start= start, end= end, name = "Gencode V19",
                           collapseTranscripts=TRUE, showId=TRUE,shape="arrow")
eGTexTracklist <- list(grtrack,eGTexTrackSNP)
plotTracks(eGTexTracklist, chromosome=chr,from=start,to=end)
```

2 other functions were created to visualise supplement data from GTEx version 3

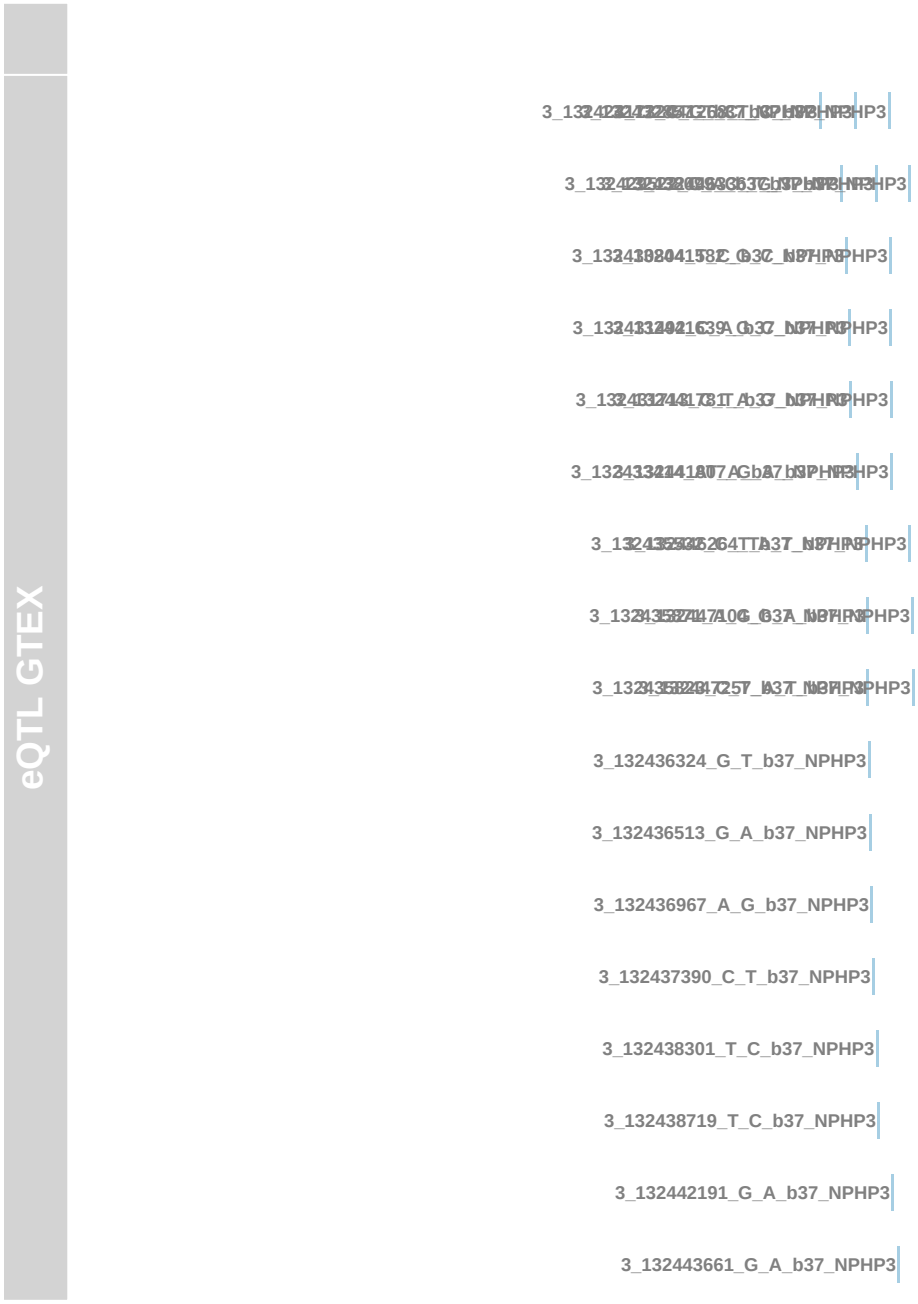

Figure 10: Plot eQTL from GTex.

1. **psiQTL_GTEx** visualise results from the protein truncating variants QTL (psiQTL) analysis for mine main tissues, plus brain, plus multi-tissue that averages the exons where data for three or more tissues is available. The name of file in GTEX version 3 is *gtex_psiqtls.zip*.
2. **imprintedGenes_GTEx** visuaslise gene imprinting genes in different tissues [**?**] via url http://www.gtexportal.org/home/imprintingPage. There are 33 tissues and 5 classification

```
### psiQTL
chr<-"chr13"
```

```
start <- 52713837
end <- 52715894
gen<-"hg19"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
psiQTLFilePath <- file.path(extdata, "/GTEX/psiQTL_Assoc-total.AdiposeTissue.txt")

psiGTex<- psiQTL_GTEx(gen,chr,start, end, psiQTLFilePath, featureDisplay = 'all',
                      showId=TRUE, type_stacking="squish",just_group="above" )

genetrack <-genes_ENSEMBL(gen,chr,start,end,showId=TRUE)

psiTrack <- list(genetrack,psiGTex)
plotTracks(psiTrack, chromosome=chr,from=start,to=end)
```

```
data(imprintedGenesGTEx)
as.character(unique(imprintedGenesGTEx$Tissue.Name))
```

```
##  [1] "Pancreas"                          "Whole_Blood"
##  [3] "Pituitary"                         "Lung"
##  [5] "Cells_EBV-transformed_lymphocytes" "Thyroid"
##  [7] "Adipose_Subcutaneous"              "Artery_Tibial"
##  [9] "Skin_Sun_Exposed_Lower_leg"        "Skin_Not_Sun_Exposed_Suprapubic"
## [11] "Brain"                             "Muscle_Skeletal"
## [13] "Breast_Mammary_Tissue"             "Nerve_Tibial"
## [15] "Adrenal_Gland"                     "Colon_Transverse"
## [17] "Prostate"                          "Artery_Coronary"
## [19] "Heart_Left_Ventricle"              "Heart_Atrial_Appendage"
## [21] "Uterus"                            "Liver"
## [23] "Vagina"                            "Testis"
## [25] "Adipose_Visceral_Omentum"          "Fallopian_Tube"
## [27] "Esophagus_Muscularis"              "Ovary"
## [29] "Cells_Transformed_fibroblasts"     "Esophagus_Mucosa"
## [31] "Kidney_Cortex"                     "Stomach"
## [33] "Artery_Aorta"
```

```
as.character(unique(imprintedGenesGTEx$Classification))
```

```
## [1] "consistent with biallelic"  "imprinted"                  "NC"
## [4] "consistent with imprinting" "biallelic"
```

```
### inprinted genes
chr<- "chr1"
start <- 7895752
end <- 7914572
gen<-"hg19"

genesTrack <- genes_ENSEMBL(gen,chr,start,end,showId=TRUE)
```

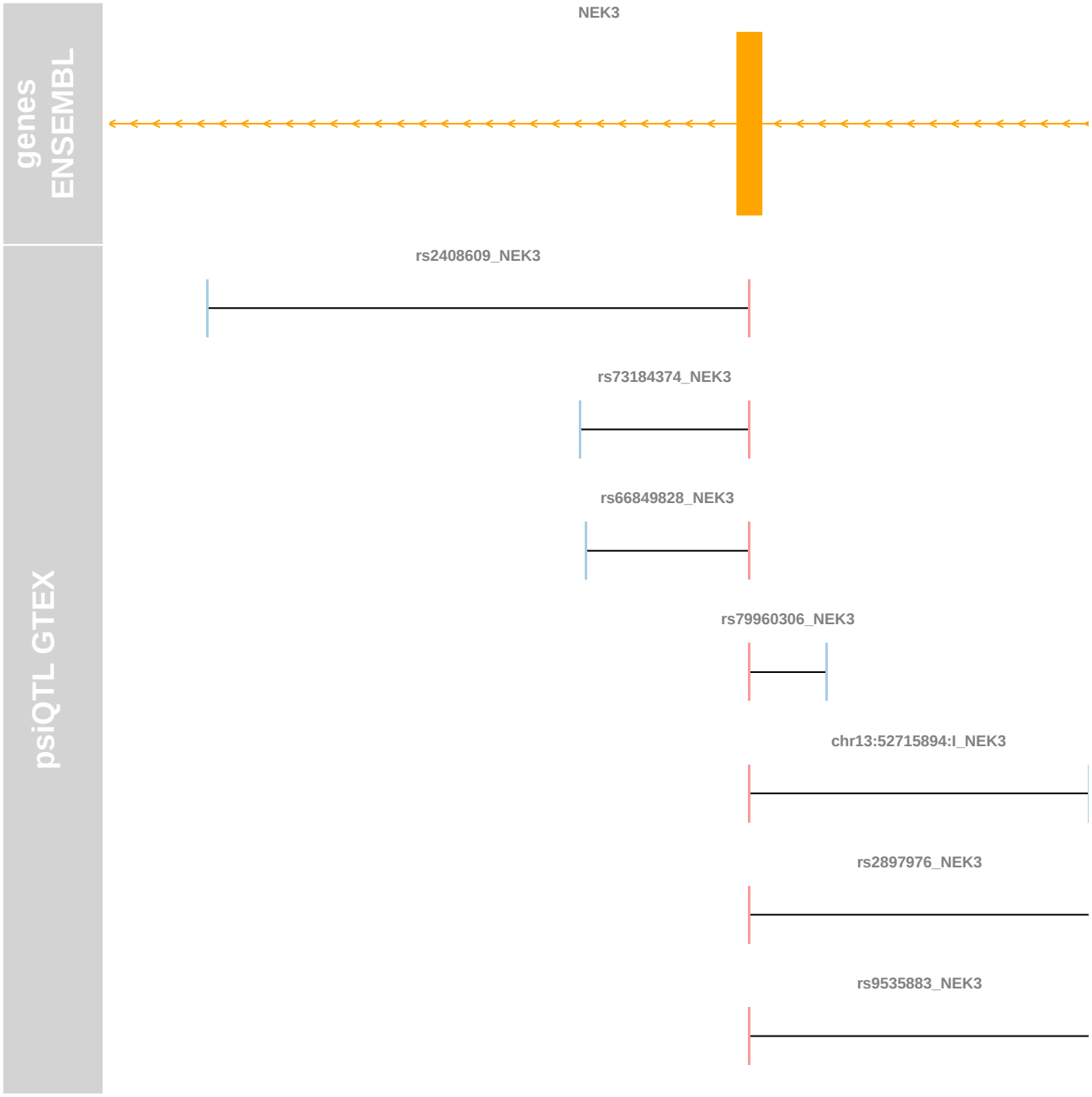

Figure 11: Plot psiQTL from GTex.

```
allIG <- imprintedGenes_GTEx(gen,chr,start, end, tissues="all",
                             classification="imprinted",showId=TRUE)

allimprintedIG <- imprintedGenes_GTEx(chr,start, end, tissues="all",
                                      classification="imprinted",showId=TRUE)

StomachIG <-imprintedGenes_GTEx(gen,chr,start, end, tissues="Stomach",
```

```
                                classification="all",showId=TRUE)

PancreasIG <- imprintedGenes_GTEx(gen,chr,start, end,
                                  tissues="Pancreas",
                                  classification="all",showId=TRUE)
PancreasimprintedIG <- imprintedGenes_GTEx(gen,chr,start, end, tissues="Pancreas",
                                           classification="imprinted",showId=TRUE)

plotTracks(list(genesTrack, allIG, allimprintedIG,
                StomachIG,PancreasIG,PancreasimprintedIG),
           chromosome=chr, from=start, to=end)
```

```
## Warning in .local(x, ..., na.rm = na.rm):  'na.rm' argument is ignored
```

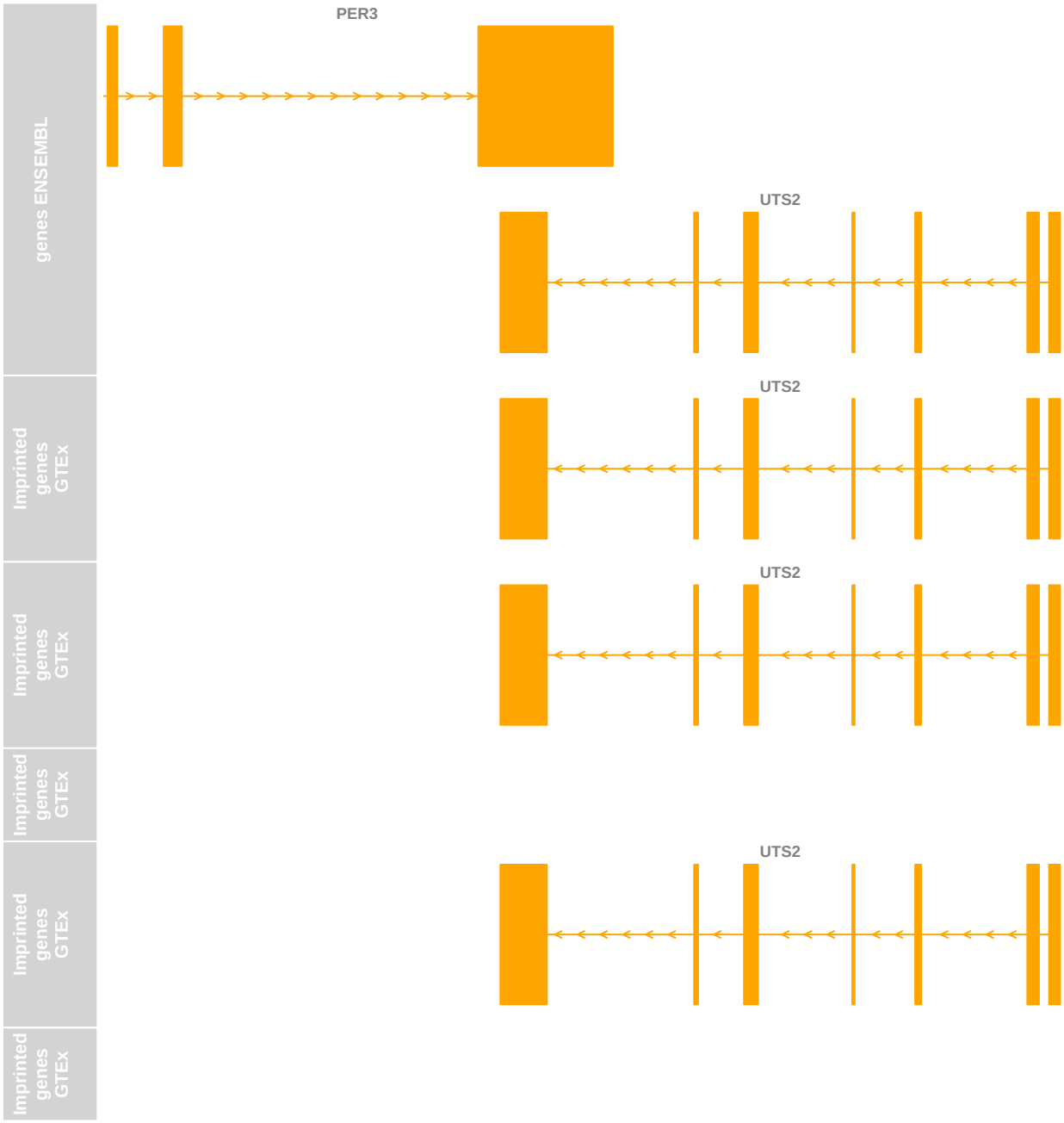

Figure 12: Plot imprinted genes from GTex.

9.6 Hi-C data

Below are examples of Hi-C data available for different tissues.

9.6.1 Hi-C data at 1kb resolution at Lieberman Aiden lab

They [**?**] used in situ Hi-C to probe the three-dimensional architecture of genomes, constructing haploid and diploid maps of nine cell types. The densest, in human lymphoblastoid cells, contains 4.9 billion contacts, achieving 1-kilobase resolution.The data were mapped on **hg19** reference genome.

You can download intrachromosomal matrice from http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525 for the cell-type of interest.

```
library('corrplot')
#Hi-C data
gen <- "hg19"
chr<-"chr1"
start <- 5000000
end <- 9000000
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "HiC/chr1_1mb.RAWobserved")

matrix_HiC <- HiCdata2matrix(chr,start, end, bedFilePath)
cor_matrix_HiC <- cor(matrix_HiC)
diag(cor_matrix_HiC)<-1
corrplot(cor_matrix_HiC, method = "circle")
```

You can quick visualise this data using this HiC-interaction tool http://promoter.bx.psu.edu/hi-c/view.php?species=human&assembly=hg19&source=inside&tissue=GM12878&resolution=1&c_url=&gene=CTXN1&sessionID=

9.6.2 Hi-C Data Browser

You can download heatmap of your region of interest from two cell-line GM06690 (immortalized lymphoblast) or K562 (leukemia) using their website http://hic.umassmed.edu/heatmap/heatmap.php. This data was produced by [**?**]. The region that you want to visualise with this data need to large more than either 100Kb or 1Mb as Heatmaps were generated by dividing the chromosome up into 100 Kb or 1 Mb windows. The data were mapped on **hg19** reference genome.

You need to create info file to define the position of each bin composing your interaction matrice in using the row name of matrice as the name of bin contain the start and end of bin.

9.6.3 Hi-C project at Ren Lab

Interaction matrices for each of the four cell types analysis (mouse ES cell, mouse cortex, human ES cell (H1), and IMR90 fibroblasts) by Ren Lab (to cite them, you need to select the publication for this url http://promoter.bx.psu.edu/hi-c/publications.html) are accessible via url http://chromosome.sdsc.edu/mouse/hi-c/download.html. The interaction matrices are created using either a 40kb bin size throughout the genome. So

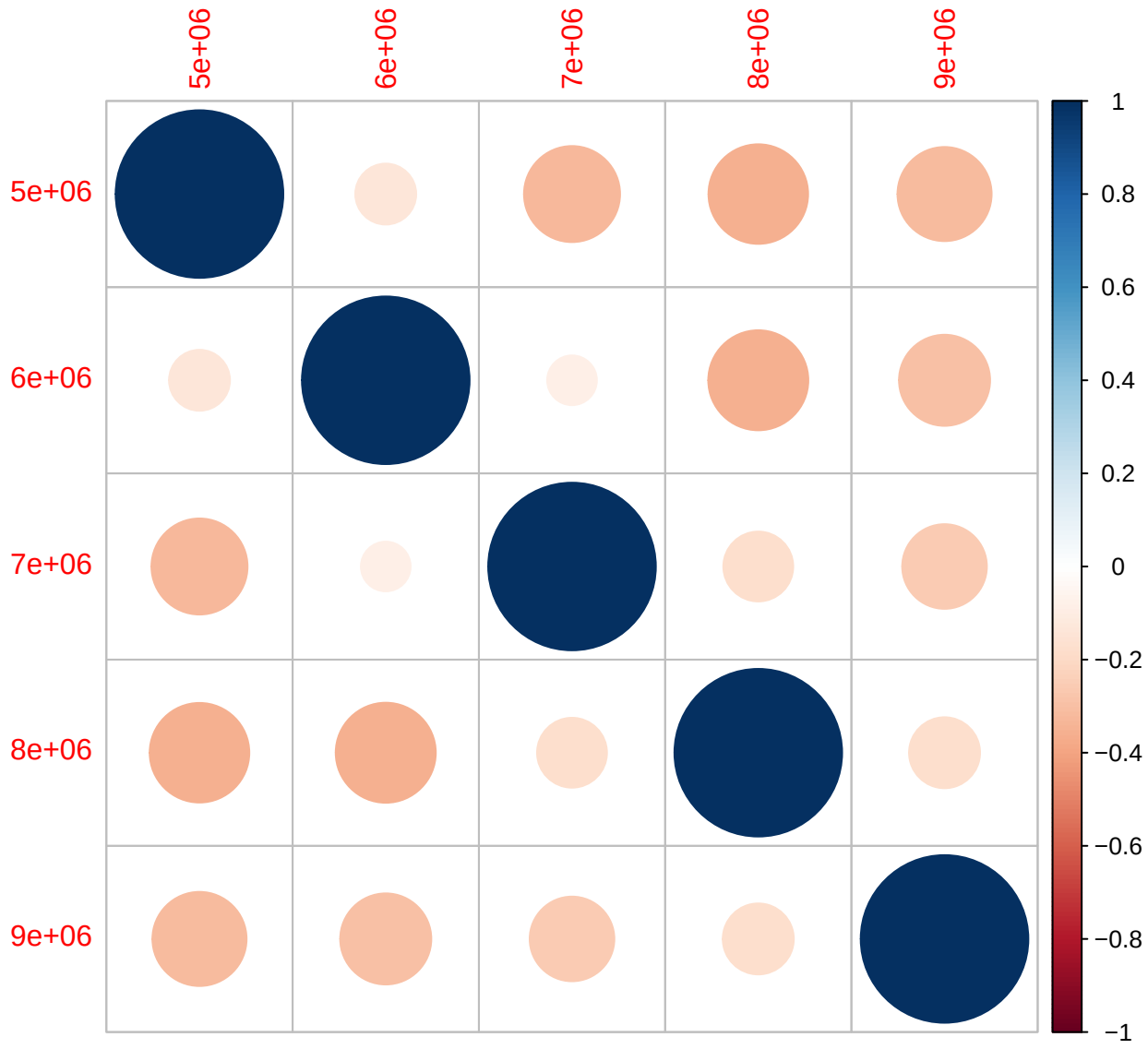

Figure 13: plot HiC data.

the region that you want to visualise with this data need to large more than 40Kb. The data were mapped on **hg19** reference genome.

You need to :

1. Extract from the BED file that contains the locations of each of the topological domains the region of interest
2. Extract in either raw or normalised matrice only the sub-matrice of interest

```
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
info_HiC <- file.path(extdata, "Human_IMR90_Fibroblast_topological_domains.txt")
data_info_HiC <-read.csv(info_HiC, header = FALSE, sep = "\t", quote = "")

intrachr_HiC <- file.path(extdata, "Human_IMR90_Fibroblast_Normalized_Matrices.txt")
data_intrachr_HiC <- read.csv(intrachr_HiC, header = TRUE, sep = "\t", quote = "")

chr_interest <- "chr2"
start_interest <- "1"
end_interest <- "160000"
list_bins <- which(data_info_HiC[,1] == chr_interest &
                     data_info_HiC[,2] >= start_interest &
                     data_info_HiC[,2] <= end_interest )

subdata_info_Hic <- data_info_HiC[list_bins,]
subdata_intrachr_HiC <- data_intrachr_HiC[list_bins,list_bins]
```

9.7 FANTOM5 database

FANTOM http://fantom.gsc.riken.jp/ established the FANTOM database (transcripts, transcription factors, promoters and enhancers active,TSS) and the FANTOM full-length cDNA clone bank, which are available worldwide for about 400 distinct cell types. Currently, FANTOM is in version FANTOM5 phase 2 where data were mapped on reference genome **hg19** for human or **mm9** for mouse [**?**].

To extract data

- from http://fantom.gsc.riken.jp/5/
- from http://fantom.gsc.riken.jp/data/ or http://fantom.gsc.riken.jp/views/
- from BED file used by UCSC HUB http://fantom.gsc.riken.jp/5/datahub/, more information here http://fantom.gsc.riken.jp/5/datahub/description.html

As the data are in classical format such as BED file, you can use easily Gviz's DataTrack function to visuaslise them. However, there are some comment lines that you need to remove in the top of files.

2 functions were created :

- **DNaseI_FANTOM** helps to visualise enhancer regions defined by FANTOM5
- **TFBS_FANTOM** helps to visualise TFBS regions defined by FANTOM5

```
gen <- "hg19"
chr<- "chr1"
start <- 6000000
end <- 6500000

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)

##Enhancer
enhFantomFile <- file.path(extdata, "/FANTOM/human_permissive_enhancers_phase_1_and_2.bed")
enhFANTOMtrack <-DNaseI_FANTOM(gen,chr,start, end, enhFantomFile, featureDisplay='enhancer')

### TFBS motif
AP1FantomFile <- file.path(extdata, "/FANTOM/Fantom_hg19.AP1_MA0099.2.sites.txt")
tfbsFANTOMtrack <- TFBS_FANTOM(gen,chr,start, end, AP1FantomFile)
```

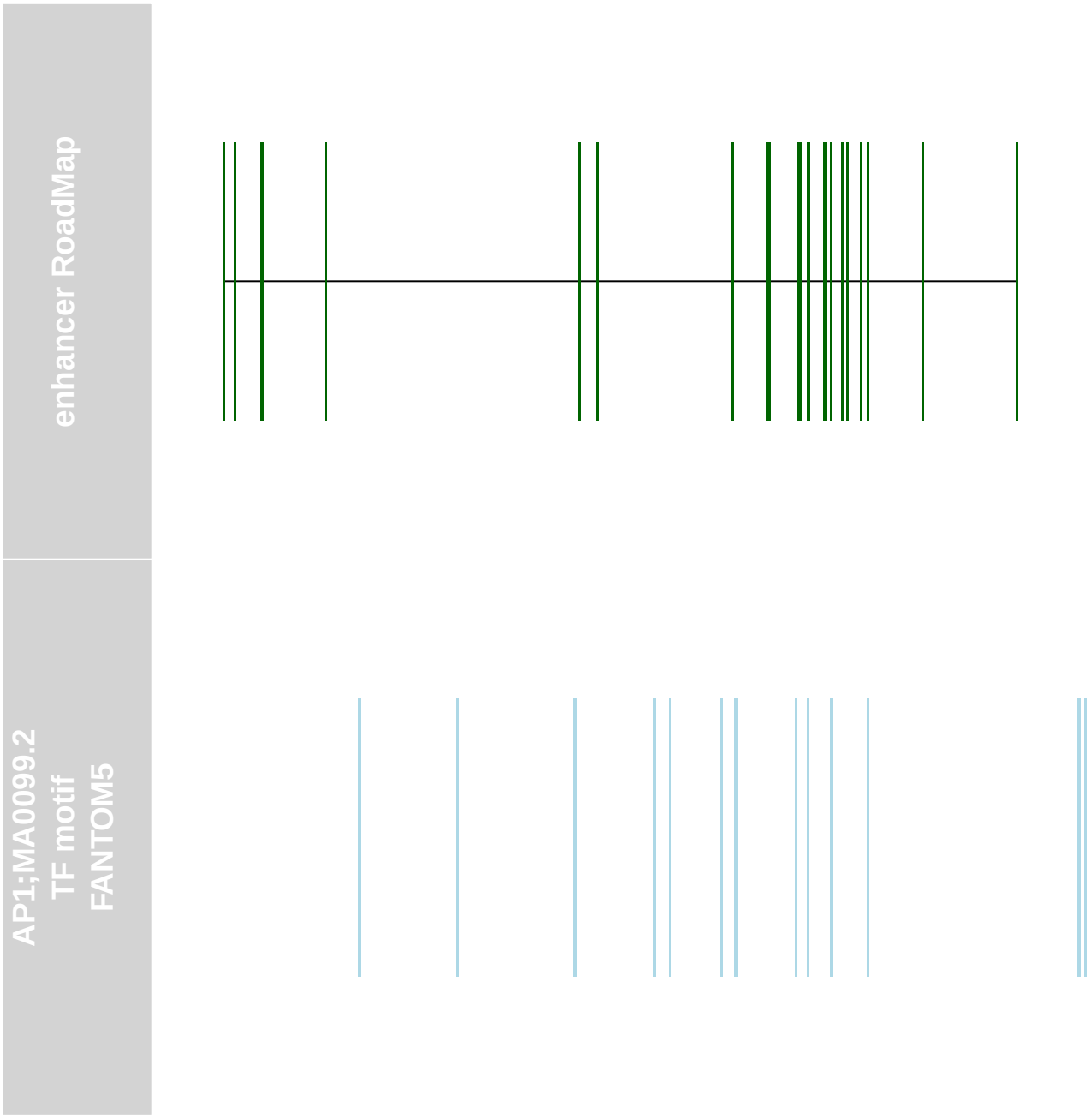

Figure 14: plot FANTOM5 data.

9.8 BLUEprint project

BLUEprint http://www.blueprint-epigenome.eu/ aims to further the understanding of how genes are activated or repressed in both healthy and diseased human cells. BLUEPRINT will focus on distinct types of haematopoietic cells from healthy individuals and on their malignant leukaemic counterparts.

the data were mapped on reference genome partially on **GRCh37** and all on **GRCh38**.

As the data are in classical format such as BED file, BigWig of GTF, you can use easily DataTrack or AnnotationTrack of Gviz to visuaslise them.

9.9 Our data

9.9.1 eQTL data

You can visualise our eQTL using **eQTL** function. Listed below are the colours used for the different elements contained in eQTL data.

9.9.2 metQTL data

You can visualise our eQTL using **metQTL** function.

Listed below are the colours used for the different elements contained in metQTL data.

10 coMET: Shiny web-service

10.1 How to use the coMET web-service

If you want to use coMET via its webservice, please go to http://epigen.kcl.ac.uk/comet and select one of different instances or direcly access one of the instances, for example http://comet.epigen.kcl.ac.uk:3838/coMET/. We have created different instances of coMET because we did not have access to the pro version of Shiny. All instances use the same version of coMET.

If you use coMET from a Shiny webservice, you do not need to install the coMET package on your computer. The web service is user friendly and requires input files and configuration of the plot. The creation of the coMET plot can take some time because it makes a live connection to UCSC or/and ENSEMBL for the annotation tracks. First, the plot is created on the webpage, and then it can be saved as an output file. For better quality plots please use the download option and the plot will be recreated in a file in pdf or eps format.

10.2 How to install the coMET web-service

These are different steps to install coMET on your Shiny web-service and you need to be root to install it.

1. You need to install R, Bioconductor and the coMET package.
2. You need first to install the shiny R package before Shiny Server.
   ```
   sudo su - -c "R -e \"install.packages('shiny', repos='http://cran.rstudio.com/')\""
   ```
3. You can install Shiny Server http://shiny.rstudio.com/, go to https://www.rstudio.com/products/shiny/download-server/.
   ```
   sudo apt-get install gdebi-core
   wget https://download3.rstudio.org/ubuntu-12.04/x86_64/shiny-server-1.4.2.786-amd64.deb
   sudo gdebi shiny-server-1.4.2.786-amd64.deb
   ```
4. Shiny Server should now be installed and running on port 3838. You should be able to see a default welcome screen at http://your_server_ip:3838/. You can make sure your Shiny Server is working properly by going to http://your_server_ip:3838/sample-apps/hello/.
5. You now have a functioning Shiny Server that can host Shiny applications or interactive documents. The configuration file for Shiny Server is at /etc/shiny-server/shiny-server.conf. By default it is configured to serve applications in the /srv/shiny-server/ directory. This means that any Shiny application that is placed at /srv/shiny-server/app_name will be available to the public at http://your_server_ip:3838/app_name/.
6. In a Shiny's folder (e.g. /var/shiny-server/www), you can create a folder called "COMET".
7. Following this, you can install the two coMET scripts in www of the coMET package, within this new folder.
8. You need to change owner and permissions to access this folder. Only the user called Shiny can access it.
   ```
   mkdir -p /var/shiny-server/www/COMET

   chmod 755 /var/shiny-server/www/COMET
   chown shiny:shiny /var/shiny-server/www/COMET

   mkdir -p /var/shiny-server/log

   chmod 755 /var/shiny-server/log
   ```

```
chown shiny:shiny /var/shiny-server/log
```

9. You need now to update the configuration file of Shiny (e.g. /etc/shiny-server/shiny-server.conf).
10. 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
    ```
11. At the end, you should 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 FAQs

- **I cannot see my plot after running comet or comet.web. What should I do?**
  If the previous time comet or comet.web ran and error was produced it prevents the plot from being closed. to fix this use the command '*dev.off()*' as many times as necessary.

- **How do we know if my track has data? and what the data is?**
  Type the name of your track, visualise the track with plotTrack or read different parameters with *str* function.

```
genetrack <-genesENSEMBL(gen,chrom,start,end,showId=TRUE)

plotTracks(genetrack)

str(genetrack)
```

- **How do you increase the size of the font of the name of an object**?
  To enlarge the name of gene, as the object is Gviz object, you can use the option from Gviz
  You can see the value of different parameters via this command line:

```
genetrack <-genesENSEMBL(gen,chrom,start,end,showId=TRUE)

displayPars(genetrack)
```

  So if you want to enlarge the name of gene, you need to do use the option fontsize.gviz in the coMET function, an example is given below:

```
comet(config.file = configfile, mydata.file = myinfofile, mydata.format = "file",
      cormatrix.file = mycorrelation, cormatrix.type = "listfile",
      mydata.large.file = mylargedata,mydata.large.type = "listfile",
      tracks.gviz = listGviz, verbose = TRUE,
      print.image=TRUE,fontsize.gviz=10)
```

- **Can I make a selection of which genes or transcripts to display**?
  To make a selection of genes to display first create the track like you would if you were displaying all genes. From this track create another with only the genes you want to display like in the example below. Please note it is not possible to select genes based on their names unless the option to display gene names instead of gene reference is used, in other cases it is possible to make a selection based on the genes reference number.

```
geneTrack <- refGenesUCSC(gen, chr, start, end, IdType ="name", showId = TRUE)

geneTrackShow <- geneTrack[gene(geneTrack) %in% c("AHRR")]
```

- **How can I better understand where the comet function stopped**?
  Use option *VERBOSE=TRUE* in the function coMET or coMET.web
  If this does not help resolve the issue, please to send your command line with *VERBOSE=TRUE* and its error message to tiphaine.martin@kcl.ac.uk. Do not forget to give alsoinformation about the session by using sessionInfo().

12 SessionInfo

The following is the session info that generated this vignette:

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

- R version 3.3.0 (2016-05-03), `x86_64-apple-darwin13.4.0`
- Locale: `C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8`
- Base packages: base, datasets, grDevices, graphics, grid, methods, parallel, stats, stats4, utils
- Other packages: BiocGenerics 0.18.0, GenomeInfoDb 1.8.2, GenomicRanges 1.24.2, Gviz 1.16.1, IRanges 2.6.0, S4Vectors 0.10.1, biomaRt 2.28.0, coMET 1.4.4, corrplot 0.77, knitr 1.13, psych 1.6.4
- Loaded via a namespace (and not attached): AnnotationDbi 1.34.3, AnnotationHub 2.4.2, BSgenome 1.40.0, Biobase 2.32.0, BiocInstaller 1.22.2, BiocParallel 1.6.2, BiocStyle 2.0.2, Biostrings 2.40.2, DBI 0.4-1, Formula 1.2-1, GGally 1.1.0, GenomicAlignments 1.8.2, GenomicFeatures 1.24.2, Hmisc 3.17-4, Matrix 1.2-6, OrganismDbi 1.14.1, R6 2.1.2, RBGL 1.48.1, RColorBrewer 1.1-2, RCurl 1.95-4.8, RSQLite 1.0.0, Rcpp 0.12.5, Rsamtools 1.24.0, SummarizedExperiment 1.2.3, VariantAnnotation 1.18.1, XML 3.98-1.4, XVector 0.12.0, acepack 1.3-3.3, biovizBase 1.20.0, bitops 1.0-6, chron 2.3-47, cluster 2.0.4, colorspace 1.2-6, colortools 0.1.5, data.table 1.9.6, dichromat 2.0-0, digest 0.6.9, ensembldb 1.4.6, evaluate 0.9, foreign 0.8-66, formatR 1.4, ggbio 1.20.1, ggplot2 2.1.0, grImport 0.9-0, graph 1.50.0, gridExtra 2.2.1, gtable 0.2.0, hash 2.2.6, highr 0.6, htmltools 0.3.5, httpuv 1.3.3, httr 1.1.0, interactiveDisplayBase 1.10.3, lattice 0.20-33, latticeExtra 0.6-28, magrittr 1.5, matrixStats 0.50.2, mime 0.4, mnormt 1.5-4, munsell 0.4.3, nnet 7.3-12, pbapply 1.2-1, plyr 1.8.4, reshape 0.8.5, reshape2 1.4.1, rpart 4.1-10, rtracklayer 1.32.1, scales 0.4.0, shiny 0.13.2, splines 3.3.0, stringi 1.1.1, stringr 1.0.0, survival 2.39-4, tools 3.3.0, trackViewer 1.8.3, xtable 1.8-2, zlibbioc 1.18.0