Package 'coMET'

September 11, 2024

Type Package

Title coMET: visualisation of regional epigenome-wide association scan (EWAS) results and DNA co-methylation patterns

Version 1.37.0

Date 2022-10-26

Author Tiphaine C. Martin [aut,cre], Thomas Hardiman [aut], Idil Yet [aut], Pei-Chien Tsai [aut], Jordana T. Bell [aut]

Maintainer Tiphaine Martin <tiphaine.martin@mssm.edu>

Description Visualisation of EWAS results in a genomic region. In addition to phenotype-association P-values, coMET also generates plots of co-methylation patterns and provides a series of annotation tracks. It can be used to other omic-wide association scans as lon:g as the data can be translated to genomic level and for any species.

Depends R (>= 4.1.0), grid, utils, biomaRt, Gviz, psych

Suggests BiocStyle, knitr, RUnit, BiocGenerics, showtext

Imports hash,grDevices, gridExtra, rtracklayer, IRanges, S4Vectors, GenomicRanges, stats, corrplot

License GPL (>= 2)

URL http://epigen.kcl.ac.uk/comet

biocViews Software, DifferentialMethylation, Visualization, Sequencing, Genetics, FunctionalGenomics, Microarray, MethylationArray, MethylSeq, ChIPSeq, DNASeq, RiboSeq, RNASeq, ExomeSeq, DNAMethylation, GenomeWideAssociation, MotifAnnotation

VignetteBuilder knitr

NeedsCompilation no

git_url https://git.bioconductor.org/packages/coMET

git_branch devel

git_last_commit d78add3

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-09-11

Contents

coMET-package . . . 4
bindingMotifsBiomart_ENSEMBL . . . 5
check.configVar . . . 8
check.configVar.cometlist . . . 8
check.format.mydata . . . 9
ChIPTF_ENCODE . . . 9
chromatinHMMAll_UCSC . . . 11
chromatinHMMOne_UCSC . . . 13
chromHMM_RoadMap . . . 14
chrUCSC2ENSEMBL . . . 16
clinCNV . . . 17
ClinVarCnv_UCSC . . . 17
ClinVarMain_UCSC . . . 18
col2HSV . . . 20
comet . . . 20
comet.list . . . 27
comet.web . . . 28
complementary . . . 33
compute.cormatrix . . . 34
CoreillCNV_UCSC . . . 35
COSMIC_UCSC . . . 36
cpgIslands_UCSC . . . 37
cpgPvalue . . . 38
create.color.bar . . . 39
create.color.list . . . 40
create.color.list.large . . . 41
create.symbol.list . . . 41
create.symbol.list.large . . . 42
create.tracks.user . . . 43
create.tracks.web . . . 43
createList.trackUser . . . 44
dgfootprints_RoadMap . . . 44
DNaseI_FANTOM . . . 46
DNaseI_RoadMap . . . 47
DNAse_UCSC . . . 49
draw.legend . . . 50
draw.name.genes.web . . . 51
draw.name.tracks.web . . . 51
draw.plot.annotation . . . 52
draw.plot.axis.data . . . 52
draw.plot.comet . . . 53
draw.plot.comet.nopval . . . 53
draw.plot.comet.web . . . 54
draw.plot.cormatrix.plot . . . 55
draw.plot.grid.mydata . . . 55
draw.plot.grid.mydata.large . . . 56

draw.plot.grid.mydata.names . . . 56
draw.plot.grid.setup . . . 57
draw.plot.linesconnection . . . 57
draw.plot.mydata.ggbio . . . 58
eQTL . . . 59
eQTL_GTEx . . . 61
fix.values . . . 63
fix.values.generic . . . 63
fix.values.large . . . 64
GAD_UCSC . . . 64
gcContent_UCSC . . . 65
GeneReviews_UCSC . . . 67
genesName_ENSEMBL . . . 68
genes_ENSEMBL . . . 69
GWAScatalog_UCSC . . . 70
HiCdata2matrix . . . 72
HistoneAll_UCSC . . . 73
HistoneOne_UCSC . . . 74
imprintedGenes_GTEx . . . 76
interestGenes_ENSEMBL . . . 77
interestTranscript_ENSEMBL . . . 79
ISCA_UCSC . . . 80
knownGenes_UCSC . . . 82
metQTL . . . 83
miRNATargetRegionsBiomart_ENSEMBL . . . 85
otherRegulatoryRegionsBiomart_ENSEMBL . . . 87
pizza . . . 89
printPlot.comet . . . 90
printPlot.comet.nopval . . . 90
printPlot.comet.web . . . 91
psiQTL_GTEx . . . 91
read.config . . . 93
read.file.cormatrix . . . 94
read.file.mydata . . . 94
read.file.mydata.large . . . 95
refGenes_UCSC . . . 96
regulationBiomart_ENSEMBL . . . 97
regulatoryEvidenceBiomart_ENSEMBL . . . 98
regulatoryFeaturesBiomart_ENSEMBL . . . 100
regulatorySegmentsBiomart_ENSEMBL . . . 102
repeatMasker_UCSC . . . 105
retrieve.data . . . 106
segmentalDups_UCSC . . . 107
set.image.parameters . . . 108
setColors . . . 108
snpBiomart_ENSEMBL . . . 109
snpLocations_UCSC . . . 110
structureBiomart_ENSEMBL . . . 111

TFBS_FANTOM 113
transcript_ENSEMBL 114
xenorefGenes_UCSC 115

Index **117**

`coMET-package` *visualisation of regional epigenome-wide association scan (EWAS) results and DNA co-methylation patterns (and also for other omic-WAS)*

Description

coMET is an R package for visualising EWAS results in a genomic region. Along with phenotype-association plots, coMET also generates plots of co-methylation patterns and provides a series of annotation tracks. The software is designed for epigenetic data, but can also be applied to genomic and functional genomic datasets (other omic-WAS results) in any species.

Details

Package:	coMET
Type:	Package
Version:	1.11.5
Date:	2018-04-16
License:	GPL (>=2)

coMET is an R package that can generate regional plots of EWAS results, DNA co-methylation patterns, and genomic information. A coMET figure includes 3 panels with a plot of P-values from EWAS, customized annotation tracks, and a triangle heatmap plot which demonstrates the correlation structure of DNA methylation at the CpG sites in the genomic region. Plots are created as PDF or EPS files.

Author(s)

Tiphaine C. Martin, Thomas Hardiman, Idil Yet, Pei-Chien Tsai, Jordana T. Bell

Maintainer: Tiphaine Martin <tiphaine.martin@mssm.edu>

Website: http://www.epigen.kcl.ac.uk/comet

References

Martin, T.C, 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.

Examples

```
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 <- "hg38"

if(interactive()){
    genetrack <-genes_ENSEMBL(gen,chrom,start,end,showId=TRUE)
    snptrack <- snpBiomart_ENSEMBL(gen, chrom, start, end,
                dataset="hsapiens_snp_som",showId=FALSE)
    strutrack <- structureBiomart_ENSEMBL(gen, 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",
      mydata.large.file=myexpressfile, mydata.large.type="listfile",
      tracks.gviz=listgviz, verbose=FALSE, print.image=FALSE,disp.pvalueplot=FALSE)
} else {
    data(geneENSEMBLtrack)
    data(snpBiomarttrack)
    data(ISCAtrack)
    data(strucBiomarttrack)
    data(ClinVarCnvTrack)
    data(clinVarMaintrack)
    data(GWASTrack)
    data(GeneReviewTrack)

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

`bindingMotifsBiomart_ENSEMBL`

Creates a binding motif track from ENSEMBL

Description

Creates a binding motif track from ENSEMBL using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
bindingMotifsBiomart_ENSEMBL(gen, chr, start, end, featureDisplay="all",
datasetEnsembl = NULL, title="Binding Motifs ENSEMBL")
```

Arguments

gen	The name of the genome. Currently only handles human data from either the previous version, GRCh37 (also known as hg19) or the current version, GRCh38 (also known as hg38).
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
featureDisplay	A vector of regulatory features to be displayed, such as Egr1. Spelling and capitalisation of features must be identical to those in the user guide. There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "CTCF"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("Egr1","CTCF")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.
datasetEnsembl	Allows the user to manually set which data set is used if required.
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to ENSEMBLregulation binding motif biomart

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 10000
end <- 50000
featureDisplay <- "CTCF"

if(interactive()){
 bindMotifsBiomartTrackSingle<-bindingMotifsBiomart_ENSEMBL(gen,chr,start,
 end,featureDisplay)
  plotTracks(bindMotifsBiomartTrackSingle, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
} else {
  data(bindMotifsBiomartTrackSingle)
  plotTracks(bindMotifsBiomartTrackSingle, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
}

######

library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 10000
end <- 50000
featureDisplay <- c("CTCF","Egr1")

if(interactive()){
 bindMotifsBiomartTrackMultiple<-bindingMotifsBiomart_ENSEMBL(gen,chr,start,end,featureDisplay)
  plotTracks(bindMotifsBiomartTrackMultiple, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
} else {
  data(bindMotifsBiomartTrackMultiple)
  plotTracks(bindMotifsBiomartTrackMultiple, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
}

######

library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 10000
end <- 50000
featureDisplay <- "all"

if(interactive()){
 bindMotifsBiomartTrackAll<-bindingMotifsBiomart_ENSEMBL(gen,chr,start,end,featureDisplay)
  plotTracks(bindMotifsBiomartTrackAll, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
} else {
```

```
  data(bindMotifsBiomartTrackAll)
  plotTracks(bindMotifsBiomartTrackAll, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
}
```

`check.configVar` *Check if all variables have a value related to the functions comet and comet.web*

Description

Check if all variables have a value related to the functions comet and comet.web

Usage

```
check.configVar(config.var)
```

Arguments

`config.var`	configuration variables

Value

Nothing

Author(s)

Tiphaine Martin

`check.configVar.cometlist` *Check if all variables have a value related to the function comet.list. Comet.list gives the list of correlation between omic features.*

Description

Check if all variables have a value related to the function comet.list. Comet.list gives the list of correlation between omic features.

Usage

```
check.configVar.cometlist(config.var)
```

Arguments

`config.var`	configuration variables

Value

Nothing

Author(s)

Tiphaine Martin

`check.format.mydata` *Check the format of different data*

Description

Check the format of different data

Usage

```
check.format.mydata(gbl.var, option, numfile)
```

Arguments

`gbl.var`	list of internal variables
`option`	option that say if the data to check are main or supplementary data
`numfile`	the order of file to check from the list of main or supplementary data

Value

gbl.var updated or error message

Author(s)

Tiphaine Martin

`ChIPTF_ENCODE` *Creates a TF motif track from ENCODE*

Description

Creates a track of TF motifs from ENCODE using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
ChIPTF_ENCODE(gen="hg19", chr, start, end, bedFilePath,
featureDisplay='all', motifColorFile, type_stacking='dense',
showId=FALSE,just_group="above", title="TF motifs ENCODE")
```

Arguments

gen	the name of the genome. Default value=hg19
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
bedFilePath	The path of the BED file from Kheradpour and Kellis, 2014.
featureDisplay	A vector of regulatory features to be displayed, such as Predicted heterochomatin. Spelling and capitalisation of features must be identical to those in the user guide. There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "Predicted heterochomatin"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("Predicted low activity","Predicted heterochomatin")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.
motifColorFile	The path of the BED file with 2 columns (the first for motif name and the second for the color in hex format without \# in the beginning) with a header.
type_stacking	Object of class"character", the stacking type of overlapping items on the final plot.One in c(hide, dense, squish, pack,full). More information cf the option "stacking" in Gviz
showId	logical. say if we write the name of group
just_group	position. say where we write the name of group (choice in c("above","righ","left"))
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to BindingMotifsBiomart binding motif biomart

Examples

```
library("Gviz")
gen <- "hg19"
chr<-"chr1"
start <- 1000
end <- 329000
```

```
if(interactive()){
    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, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
} else {
    data(chipTFtrack)
    plotTracks(chipTFtrack, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
}
```

`chromatinHMMAll_UCSC` *Creating multiple chromHMM tracks from the UCSC genome browser*

Description

Create multiple chromHMM Broad tracks by connecting to the UCSC genome browser using the GViz bioconductor package

Usage

```
chromatinHMMAll_UCSC(gen, chr, start, end, mySession, color='coMET',
pattern = NULL, table.name = NULL)
```

Arguments

`gen`	the name of the genome
`chr`	the chromosome of interest
`start`	the first position in region of interest (the smallest value)
`end`	the last position in region of interest (the biggest value)
`mySession`	the object session from the function browserSession of rtracklayer
`color`	the colour scheme used for plots. By defult this is set to 'coMET' to allow easy indentifcation of differnent elements. The colour scheme set by UCSC can also be used. Consult userguide for table of colours.
`pattern`	the pattern of the track to visualise
`table.name`	the name of the table from the track

Value

list of AnnotationTrack objects of GViz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=wgl

See Also

chromatinHMMOne_UCSC

Examples

```
  library("Gviz")
  library(rtracklayer)
  gen <- "hg19"
  chr <- "chr2"
  start <- 38290160
  end <- 38313219
  if(interactive()){
      BROWSER.SESSION="UCSC"
      mySession <- browserSession(BROWSER.SESSION)
      genome(mySession) <- gen
      track.name="Broad ChromHMM"
      tablestrack<-ucscTables(gen, track=track.name)
      table.name<-tablestrack[1]
      PATTERN.REGULATION<-"GM12878"

      chromhmmPattern<-chromatinHMMAll_UCSC(gen,chr,start,end,mySession,
      color='coMET',PATTERN.REGULATION)
      plotTracks(chromhmmPattern, from = start, to =end,
fontfamily="sans",fontfamily.title="sans")

      chromhmmNoPattern<-chromatinHMMAll_UCSC(gen,chr,start,end,
      mySession,color='coMET')
      plotTracks(chromhmmNoPattern, from = start, to =end,
fontfamily="sans",fontfamily.title="sans")
  } else {

      data(chromhmmPattern)
      plotTracks(chromhmmPattern, from = start, to =end,
fontfamily="sans",fontfamily.title="sans")

      data(chromhmmNoPattern)
      plotTracks(chromhmmNoPattern, from = start, to =end,
fontfamily="sans",fontfamily.title="sans")
  }
```

chromatinHMMOne_UCSC *Creating one chromHMM track from the UCSC genome browser*

Description

Create one track of only one type of chromHMM Broad element from the UCSC genome browser using the Gviz bioconductor package

Usage

```
chromatinHMMOne_UCSC(gen, chr, start, end, mySession, color="coMET",
title="ENCODE/Broad chromHMM", table.name = NULL)
```

Arguments

gen	the name of the genome. Data is not currently available for GRCh38 (hg38).
chr	the chromosome of interest
start	the first position in region of interest (the smallest value)
end	the last position in region of interest (the biggest value)
mySession	the object session from the function browserSession of rtracklayer
color	the color scheme used for plots. By defult this is set to 'coMET' to allow easy indentifcation of differnent elements. The color scheme set by UCSC can also be used. Consult userguide for table of colors.
title	Name of tracks
table.name	the name of the table from the track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=wgI

See Also

chromatinHMMAll_UCSC

Examples

```
  library("Gviz")
  library("rtracklayer")
  gen <- "hg19"
  chr <- "chr2"
  start <- 38290160
  end <- 38303219
  color <- "coMET"

  if(interactive()) {
      BROWSER.SESSION="UCSC"
      mySession <- browserSession(BROWSER.SESSION)
      genome(mySession) <- gen
      track.name="Broad ChromHMM"
      tablestrack<-ucscTables(gen, track.name)
      table.name<-tablestrack[1]
      chromhmmtrackone<-chromatinHMMOne_UCSC(gen,chr,start,end
      ,mySession,color="coMET",table.name)
      plotTracks(chromhmmtrackone, from = start, to =end,
fontfamily="sans",fontfamily.title="sans")
  }else {
      data(chromhmmtrackone)
      plotTracks(chromhmmtrackone, from = start, to =end,
fontfamily="sans",fontfamily.title="sans")
  }
```

`chromHMM_RoadMap` *Creates a ChromHMM track from a file of RoadMap*

Description

Creates a ChromHMM track from a file of RoadMap using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
chromHMM_RoadMap(gen="hg19",chr, start, end, bedFilePath,
featureDisplay = 'all', colorcase='roadmap15',
title=" chromHMM RoadMap")
```

Arguments

`gen`	the name of the genome. Default value=hg19
`chr`	The chromosome of interest
`start`	The starting position in the region of interest (the smallest value)
`end`	The end position in the region of interest (the largest value)
`bedFilePath`	The file path to the .BED file containing the data to be visualised

`featureDisplay` A vector of features to be displayed, such as 1_TssA. Spelling and capitalisation of features must be identical to those in the user guide (in the 'State & Acronym' column). There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "1_TssA"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("1_TssA","2_TssAFlnk")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.

`colorcase` the type of colors used to visualise different elements contained in ROADmap data with 15-,18-,25- states. choice between roadmap15, roadmap18, comet18, roadmap25 and comet25.

`title` The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to RoadMap Epigenome

Examples

```
library("Gviz")
chr <- "chr1"
start <- 4500000
end <- 4600000
featureDisplay <- "7_Enh"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "RoadMap/E063_15_coreMarks_mnemonics.bed")

if(interactive()){
  chromHMM_RoadMapSingle <- chromHMM_RoadMap(gen="hg19",chr,start, end,
  bedFilePath, featureDisplay = featureDisplay, colorcase='roadmap15' )
  plotTracks(chromHMM_RoadMapSingle, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
} else {
  data(chromHMM_RoadMapSingle)
  plotTracks(chromHMM_RoadMapSingle, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
}
```

```
######

library("Gviz")
chr <- "chr22"
start <- 38291000
end <- 38301200
featureDisplay <- c("7_Enh","13_ReprPC")

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "RoadMap/E063_15_coreMarks_mnemonics.bed")

if(interactive()){
  chromHMM_RoadMapMultiple <- chromHMM_RoadMap(gen="hg19",chr,start, end,
  bedFilePath, featureDisplay = featureDisplay, colorcase='roadmap15' )
  plotTracks(chromHMM_RoadMapMultiple, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
} else {
  data(chromHMM_RoadMapMultiple)
  plotTracks(chromHMM_RoadMapMultiple, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
}

#####

library("Gviz")
chr <- "chr22"
start <- 38291000
end <- 38301200
featureDisplay <- "all"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "RoadMap/E063_15_coreMarks_mnemonics.bed")

if(interactive()){
  chromHMM_RoadMapAll <- chromHMM_RoadMap(gen="hg19",chr,start, end,
  bedFilePath, featureDisplay = featureDisplay, colorcase='roadmap15' )
  plotTracks(chromHMM_RoadMapAll, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
} else {
  data(chromHMM_RoadMapAll)
  plotTracks(chromHMM_RoadMapAll, from = start, to = end,
  fontfamily="sans",fontfamily.title="sans")
}
```

---

chrUCSC2ENSEMBL *Removing "chr" to the chromosome number from UCSC to transform it to ENSEMBL chromosome format*

Description

Removing "chr" at the beginning of the chromosome number

Usage

```
chrUCSC2ENSEMBL(chr)
```

Arguments

chr	the chromosome number in UCSC format

Value

the number of chromosome at ENSEMBL format

Author(s)

Tiphaine Martin

Examples

```
chr<-"chr7"
chrUCSC2ENSEMBL(chr)
```

---

clinCNV *Data sets*

Description

Some sample data sets used for the illustrative examples and the vignette.

ClinVarCnv_UCSC *Create one track of the genomic positions of variants from the ClinVar database (CNV only)*

Description

Create one track of the genomic positions of variants from the ClinVar database (CNV only, Variants excluded) using the Gviz bioconductor package

Usage

```
ClinVarCnv_UCSC(gen, chr, start, end, title="ClinVar Variants", showId = FALSE)
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in region of interest (the smallest value)
end	the last position in region of interest (the biggest value)
title	The name of the annotation track
showId	Show the ID of the genetic elements

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=clin

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

snpLocations_UCSC, structureBiomart_ENSEMBL, snpBiomart_ENSEMBL, CoreillCNV_UCSC, COSMIC_UCSC, ClinVarMain_UCSC

Examples

```
library("Gviz")
chrom <- "chr2"
start <- 38290160
end <- 38303219
gen <- "hg38"
if(interactive()){
    clinCNV<-ClinVarCnv_UCSC(gen,chrom,start,end)
    plotTracks(clinCNV, from = start, to =end,
    fontfamily="sans",fontfamily.title="sans")
}else {
    data(ClinVarCnvTrack)
    plotTracks(clinCNV, from = start, to =end,
             fontfamily="sans",fontfamily.title="sans")
}
```

ClinVarMain_UCSC *Create one track of the genomic positions of variants from the ClinVar database (variants only)*

Description

Create one track of the genomic positions of variants from the ClinVar database (Variants only, CNV excluded) using the Gviz bioconductor package

Usage

```
ClinVarMain_UCSC(gen, chr, start, end, title="ClinVar Variants", showId=FALSE)
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in region of interest (the smallest value)
end	the last position in region of interest (the biggest value)
title	The name of the annotation track
showId	Show the ID of the genetic elements

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=clin

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

snpLocations_UCSC, structureBiomart_ENSEMBL, snpBiomart_ENSEMBL, CoreillCNV_UCSC, COSMIC_UCSC, ClinVarCnv_UCSC

Examples

```
library("Gviz")
gen <- "hg38"
chrom <- "chr2"
start <- 100000
end <- 10000000

if(interactive()) {
    clinVariant<-ClinVarMain_UCSC(gen,chrom,start,end)
    plotTracks(clinVariant, from = start, to =end,
    fontfamily="sans",fontfamily.title="sans")
}else{
    data(clinVarMaintrack)
    plotTracks(clinVariant, from = start, to =end,
    fontfamily="sans",fontfamily.title="sans")
}
```

col2HSV *col2HSV: converts a color to HSV in hexadecimal notation*

Description

col2HSV converts an R color (or a set of colors) into an HSV color model, and then returns the color names in hexadeciaml notation

Usage

```
col2HSV(color)
```

Arguments

color	an R color name or a color in hexadecimal notation

Value

A character vector with the color(s) name(s) in hexadecimal notation

Author(s)

Gaston Sanchez

Examples

```
# convert 'tomato'
col2HSV("tomato")
```

comet *Visualize EWAS results in a genomic region of interest*

Description

coMET is an R-based package to visualize EWAS (epigenome-wide association scans) results in a genomic region of interest. The main feature of coMET is to plot the the significance level of EWAS results in the selected region, along with correlation in DNA methylation values between CpG sites in the region. The coMET package generates plots of phenotype-association, co-methylation patterns, and a series of annotation tracks.

Usage

```
comet(mydata.file = NULL, mydata.format = "site", mydata.type = "file",
    mydata.large.file = NULL, mydata.large.format = "site",
    mydata.large.type = "listfile", cormatrix.file = NULL,
    cormatrix.method = "spearman", cormatrix.format = "raw",
    cormatrix.color.scheme = "bluewhitered",cormatrix.conf.level=0.05,
    cormatrix.sig.level= 1, cormatrix.adjust="none",
    cormatrix.type = "listfile", mydata.ref = NULL,
    start = NULL, end = NULL, zoom = FALSE, lab.Y = "log",
    pval.threshold = 1e-05,pval.threshold.2 = 0,disp.pval.threshold = 1,
    disp.association = FALSE, disp.association.large = FALSE,
    disp.region = FALSE, disp.region.large = FALSE,
   disp.beta.association = FALSE, disp.beta.association.large = FALSE, factor.beta = 0.3,
    symbols = "circle-fill",
    symbols.large = NA, sample.labels = NULL, sample.labels.large = NULL,
   use.colors = TRUE , disp.color.ref = TRUE, color.list = NULL, color.list.large = NULL,
    disp.mydata = TRUE, biofeat.user.file = NULL, biofeat.user.type = NULL,
    biofeat.user.type.plot = NULL,
    genome = "hg19", dataset.gene = "hsapiens_gene_ensembl",
    tracks.gviz = NULL,
    disp.mydata.names = TRUE, disp.color.bar = TRUE, disp.phys.dist = TRUE,
    disp.legend = TRUE, disp.marker.lines = TRUE, disp.cormatrixmap = TRUE,
    disp.pvalueplot =TRUE, disp.type = "symbol", disp.mult.lab.X = FALSE,
    disp.connecting.lines = TRUE, palette.file = NULL, image.title = NULL,
    image.name = "coMET", image.type = NULL, image.size = 3.5,
    fontsize.gviz=5, font.factor = 1,
    symbol.factor = NULL, print.image = TRUE, connecting.lines.factor = 1.5,
    connecting.lines.adj = 0.01, connecting.lines.vert.adj = -1,
    connecting.lines.flex = 0, config.file = NULL, verbose = FALSE)
```

Arguments

`mydata.file`	Name of the info file describing the coMET parameters
`mydata.format`	Format of the input data in mydata.file. There are 4 different options: site, region, site_asso, region_asso.
`mydata.type`	Format of mydata.file. There are 2 different options: FILE or MATRIX.
`mydata.large.file`	Name of additional info files describing the coMET parameters. File names should be comma-separated. It is optional, but if you add some, they need to be file(s) in tabular format with a header. Additional 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.large.format.

`mydata.large.format`
: Format of additional data to be visualised in the p-value plot. Format should be comma-separated. There are 4 different options for each file: site, region, site_asso, region_asso.

`mydata.large.type`
: Format of mydata.large.file. There are 2 different options: listfile or listdataframe.

`cormatrix.file`
: Name of the raw data file or the pre-computed correlation matrix file. It is mandatory and has to be a file in tabular format with an header.

`cormatrix.method`
: Options for calculating the correlation matrix: spearman, pearson and kendall

`cormatrix.format`
: Format of the input cormatrix.file. TThere are two options: raw file (raw if CpG sites are by column and samples by row or raw_rev if CpG site are by row and samples by column) and pre-computed correlation matrix (cormatrix)

`cormatrix.color.scheme`
: Color scheme options: heat, bluewhitered, cm, topo, gray, bluetored

`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"

`cormatrix.type`
: Format of cormatrix.file. There are 2 different options: listfile or listdataframe.

`mydata.ref`
: The name of the referenceomic feature (e.g. CpG-site) listed in mydata.file

`start`
: The first nucleotide position to be visualised. It could be bigger or smaller than the first position of our list of omic features.

`end`
: the last nucleotide position to be visualised. It has to be bigger than the value in the option start, but it could be smaller or bigger than the last position of our list of omic features.

`zoom`
: Default=False

`lab.Y`
: Scale of the y-axis. Options: log or ln

`pval.threshold`
: Significance threshold to be displayed as a red dashed line

`pval.threshold.2`
: the second significance threshold to be displayed as a orange dashed line

`disp.pval.threshold`
: Display only the findings that pass the value put in disp.pval.threshold

`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.association.large`
: This logical option works only if mydata.large.file contains the effect direction (mydata.large.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.large. 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.

`disp.region.large`
: This logical option works only if mydata.large.file contains regions (mydata.large.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.

`disp.beta.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 size of symbol is the default size of symbole; if TRUE, the effect direction is shown.

`disp.beta.association.large`
: This logical option works only if mydata.large.file contains the effect direction (mydata.large.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 size of symbol is ththe default size of symbole; if TRUE, the effect direction is shown.

`factor.beta`
: Factor to visualise the size of beta. Default value = 0.3.

`symbols`
: The symbol shown in the p-value plot. Options: circle, square, diamond, triangle. symbols can be filled by appending -fill, e.g. square-fill. Example: circle,diamond-fill,triangle

`symbols.large`
: The symbol to visualise the data defined in mydata.large.file. Options: circle, square, diamond, triangle; symbols can either be filled or not filled by appending -fill e.s., square-fill. Example: circle,diamond-fill,triangle

`sample.labels`
: Labels for the sample described in mydata.file to include in the legend

`sample.labels.large`
: Labels for the sample described in mydata.large.file to include in the legend

`use.colors`
: Use the colors defined or use the grey color scheme

`disp.color.ref` Logical option TRUE or FALSE (TRUE default). if TRUE, the connection line related to the reference probe is in purple, if FALSE if the connection line related to the reference probe stay black.

`color.list` List of colors for displaying the P-value symbols related to the data in mydata.file

`color.list.large` List of colors for displaying the P-value symbols related to the data in mydata.large.file

`disp.mydata` logical option TRUE or FALSE. TRUE (default). If TRUE, the P-value plot is shown; if FALSE the plot will be defined by GViz

`biofeat.user.file` Name of data file to visualise in the tracks. File names should be comma-separated.

`biofeat.user.type` Track type, where multiple tracks can be shown (comma-separated): DataTrack, AnnotationTrack, GeneregionTrack.

`biofeat.user.type.plot` Format of the plot if the data are shown with the Gviz's function called DataTrack (comma-separated)

`genome` The human genome reference file. e.g. "hg19" for Human genome 19 (NCBI 37), "grch37" (GRCh37),"grch38" (GRCh38)

`dataset.gene` The gene names from ENSEMBL. e.g. hsapiens_gene

`tracks.gviz` list of tracks created by Gviz.

`disp.mydata.names` logical option TRUE or FALSE. If True (default), the names of the CpG sites are displayed.

`disp.color.bar` Color legend for the correlation matrix (range -1 to 1). Default: blue-white-red

`disp.phys.dist` logical option (TRUE or FALSE). TRUE (default).Display the bp distance on the plots

`disp.legend` logical option TRUE or FALSE. TRUE (default) Display the sample labels and corresponding symbols on the lower right side

`disp.marker.lines` logical option TRUE or FALSE. TRUE (default), if FALSE the red line for pval.threshold is not shown

`disp.cormatrixmap` logical option TRUE or FALSE. TRUE (default), if FALSE correlation matrix is not shown

`disp.pvalueplot` logical option (TRUE or FALSE). TRUE (default), if FALSE the pvalue plot is not shown

`disp.type` Default: symbol

`disp.mult.lab.X` logical option TRUE or FALSE. FALSE (default).Display evenly spaced X-axis labels; up to 5 labels are shown.

`disp.connecting.lines`
: logical option TRUE or FALSE. TRUE (default) displays connecting lines between p-value plot and correlation matrix

`palette.file`
: File that contains color scheme for the heatmap. Colors are hexidecimal HTML color codes; one color per line; if you do not want to use this option, use the color defined by the option cormatrix.color.scheme

`image.title`
: Title of the plot

`image.name`
: The path and the name of the plot file without extension. The extension will be added by coMET depending on the option image.type.

`image.type`
: Options: pdf or eps

`image.size`
: Default: 3.5 inches. Possible sizes : 3.5 or 7

`fontsize.gviz`
: Font size of writing in annotation track. Default value =5

`font.factor`
: Font size of the sample labels. Range: 0-1

`symbol.factor`
: Size of the symbols. Range: 0-1

`print.image`
: Print image in file or not.

`connecting.lines.factor`
: Length of the connecting lines. Range: 0-2

`connecting.lines.adj`
: Position of the connecting lines horizontally. Negative values shift the connecting lines to the left and positive values shift the lines to the right. Range: (-1;1) option -1 means no connecting lines.

`connecting.lines.vert.adj`
: Position of the connecting lines vertically. Can be used to vertically adjust the position of the connecting lines in relation to the CpG-site names. Negative value shift the connecting lines down. Range: (-0.5 - 0), option -1 mean the default value related to the plot size (-0.5 for 3.5 plot size; -0.7 for 7.5 plot size)

`connecting.lines.flex`
: Adjusts the spread of the connecting lines. Range: 0-2

`config.file`
: 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 and its value are separated 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" and not extra space. (i.e. list.tracks=geneENSEMBL,CGI,ChromHMM,DNAse,RegENSEMBL

`verbose`
: logical option TRUE or FALSE. TRUE (default). If TRUE, shows comments.

Details

The function is limited to visualize 120 omic features.

Value

Create a plot in pdf or eps format depending to some options

Author(s)

Tiphaine Martin

References

http://epigen.kcl.ac.uk/comet/

See Also

comet.web,comet.list

Examples

```
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 <- "hg38"

if(interactive()){
    cat("interactive")
    genetrack <-genes_ENSEMBL(gen,chrom,start,end,showId=TRUE)
    snptrack <- snpBiomart_ENSEMBL(gen, chrom, start, end,
                dataset="hsapiens_snp_som",showId=FALSE)
    strutrack <- structureBiomart_ENSEMBL(gen, 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",
      mydata.large.file=myexpressfile, mydata.large.type="listfile",
      tracks.gviz=listgviz, verbose=FALSE, print.image=FALSE,disp.pvalueplot=FALSE)
} else {
    cat("Non interactive")
    data(geneENSEMBLtrack)
    data(snpBiomarttrack)
    data(ISCAtrack)
    data(strucBiomarttrack)
    data(ClinVarCnvTrack)
    data(clinVarMaintrack)
    data(GWASTrack)
    data(GeneReviewTrack)
    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",
        mydata.large.file=myexpressfile,  mydata.large.type="listfile",
```

```
        tracks.gviz=listgviz, verbose=FALSE, print.image=FALSE,disp.pvalueplot=FALSE)
}
```

`comet.list` *List the correlations between omic features*

Description

coMET is an R-based package to visualize EWAS (epigenome-wide association scans) results in a genomic region of interest. The main feature of coMET is to plot the the significance level of EWAS results in the selected region, along with correlation in DNA methylation values between CpG sites in the region. The coMET package generates plots of phenotype-association, co-methylation patterns, and a series of annotation tracks. In addition, the function comet.list gives the list of correlations between omic features

Usage

```
comet.list(cormatrix.file = NULL, cormatrix.method = "spearman", cormatrix.format = "raw",
    cormatrix.conf.level=0.05, cormatrix.sig.level= 1, cormatrix.adjust="none",
    cormatrix.type = "listdataframe", cormatrix.output="cormatrix_list",
    config.file = NULL, verbose = FALSE)
```

Arguments

`cormatrix.file`	Name of the raw data file or the pre-computed correlation matrix file. It is mandatory and has to be a file in tabular format with an header.
`cormatrix.method`	Options for calculating the correlation matrix: spearman, pearson and kendall. Default value= spearman
`cormatrix.format`	Format of the input cormatrix.file. TThere are two options: raw file (raw if CpG sites are by column and samples by row or raw_rev if CpG site are by row and samples by column) and pre-computed correlation matrix (cormatrix)
`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 below 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"
`cormatrix.type`	Format of cormatrix.file. There are 2 different options: listfile or listdataframe.

cormatrix.output
: The path and the name of the output file without the extension

config.file
: 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 and its value are separated by "=".

verbose
: logical option TRUE or FALSE. TRUE (default). If TRUE, shows comments.

Value

Create a list of correlation between omic features

Author(s)

Tiphaine Martin

References

http://epigen.kcl.ac.uk/comet/

See Also

comet.web,comet

Examples

```
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)
```

---

comet.web *Visualize EWAS results in a genomic region of interest with predefined annotation tracks*

Description

coMET is an R-based package to visualize EWAS (epigenome-wide association scans) results in a genomic region of interest. The main feature of coMET is to plot the the significance level of EWAS results in the selected region, along with correlation in DNA methylation values between CpG sites in the region. The coMET package generates plots of phenotype-association, co-methylation patterns, and a series of annotation tracks.

Usage

```
comet.web(mydata.file = NULL, mydata.format = c("site", "region",
"site_asso", "region_asso"),
        mydata.large.file = NULL,
        mydata.large.format = c("site", "region", "site_asso", "region_asso"),
     cormatrix.file = NULL, cormatrix.method = c("spearman", "pearson", "kendall"),
        cormatrix.format = c("cormatrix", "raw","raw_rev"),
        cormatrix.color.scheme = "heat", cormatrix.conf.level=0.05,
        cormatrix.sig.level= 1, cormatrix.adjust="none",mydata.ref = NULL,
        genome="hg19", start = NULL, end = NULL, zoom = FALSE, lab.Y = "log",
        pval.threshold = 1e-07, pval.threshold.2 = 0,  disp.pval.threshold = 1,
        disp.association= FALSE, disp.association.large = FALSE,
        disp.beta.association = "FALSE", disp.beta.association.large = "FALSE",
        factor.beta = 0.3,
       disp.region = FALSE, disp.region.large = FALSE, symbols = "circle-fill",
        symbols.large = NA, sample.labels = NULL, sample.labels.large = NULL,
        use.colors = TRUE, disp.color.ref = TRUE, color.list = NULL,
        color.list.large = NULL, biofeat.user.file = NULL,
        biofeat.user.type = c("GeneRegion", "Annotation", "Data"),
        biofeat.user.type.plot = NULL,
        list.tracks = "geneENSEMBL,CGI,ChromHMM,DNAse,RegENSEMBL,SNP",
        pattern.regulation = "GM12878",
        image.title = NULL, image.name = "coMET", image.type = c("pdf", "eps"),
        image.size = 3.5, fontsize.gviz=5, font.factor = 1,
        print.image = FALSE, config.file = NULL, verbose = FALSE)
```

Arguments

.

mydata.file	Name of the info file describing the coMET parameters. It is mandatory and has to be a file in tabular format with a 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.
mydata.format	Format of the input data in mydata.file. There are 4 different options: site, region, site_asso, region_asso.
mydata.large.file	Name of additional info files describing the coMET parameters. File names should be comma-separated. It is optional, but if you add some, they need to be file(s) in tabular format with a header. Additional 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.large.format.

`mydata.large.format`
: Format of additional data to be visualised in the p-value plot. Format should be comma-separated.There are 4 different options for each file: site, region, site_asso, region_asso.

`cormatrix.file` Name of the raw data file or the pre-computed correlation matrix file. It is mandatory and has to be a file in tabular format with an header.

`cormatrix.method`
: A character string indicating which correlation coefficient is to be used for the test. One of "pearson", "kendall", or "spearman", can be abbreviated.

`cormatrix.format`
: A character string indicating which format of the input cormatrix.file is to be used. There are three options: raw file (raw if CpG sites are by column and samples by row or row_rev if CpG site are by row and samples by column) and pre-computed correlation matrix (cormatrix)

`cormatrix.color.scheme`
: A character string indicating which Color scheme options is to be used: heat, bluewhitered, cm, topo, gray, bluetored

`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"

`mydata.ref` The name of the reference omic feature (e.g. CpG-site) listed in mydata.file

`genome` The human genome reference file. e.g. "hg19" for Human genome 19 (NCBI 37), "grch37" (GRCh37),"grch38" (GRCh38)

`start` The first nucleotide position to be visualised. It could be bigger or smaller than the first position of our list of omic features.

`end` the last nucleotide position to be visualised. It has to be bigger than the value in the option start, but it could be smaller or bigger than the last position of our list of omic features.

`zoom` logical option TRUE or FALSE. FALSE (default)

`lab.Y` Scale of the y-axis. Options: log or ln

`pval.threshold` Significance threshold to be displayed as a red dashed line. Default value = 1e-7

`pval.threshold.2`
: the second significance threshold to be displayed as a orange dashed line. Default value= 0 (no printed)

`disp.pval.threshold`
: Display only the findings that pass the value put in disp.pval.threshold

`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.association.large`
: This logical option works only if mydata.large.file contains the effect direction (MYDATA.large.FORMA=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.large. On the other hand, if the association is negative, the color is the opposed color.

`disp.beta.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 size of symbol is the default size of symbole; if TRUE, the effect direction is shown.

`disp.beta.association.large`
: This logical option works only if mydata.large.file contains the effect direction (mydata.large.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 size of symbol is ththe default size of symbole; if TRUE, the effect direction is shown.

`factor.beta`
: Factor to visualise the size of beta. Default value = 0.3.

`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.

`disp.region.large`
: This logical option works only if mydata.large.file contains regions (mydata.large.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.

`symbols`
: The symbol shown in the p-value plot. Options: circle, square, diamond, triangle. symbols can be filled by appending -fill, e.g. square-fill. Example: circle,diamond-fill,triangle

`symbols.large`
: The symbol to visualise the data defined in mydata.large.file. Options: circle, square, diamond, triangle; symbols can either be filled or not filled by appending -fill e.s., square-fill. Example: circle,diamond-fill,triangle

`sample.labels`
: Labels for the sample described in mydata.file to include in the legend

`sample.labels.large`
: Labels for the sample described in mydata.large.file to include in the legend

`use.colors`
: Use the colors defined or use the grey color scheme

`disp.color.ref`
: Logical option TRUE or FALSE (TRUE default). if TRUE, the connection line related to the reference probe is in purple, if FALSE if the connection line related to the reference probe stay black.

`color.list`
: List of colors for displaying the P-value symbols related to the data in mydata.file

`color.list.large`
: List of colors for displaying the P-value symbols related to the data in mydata.large.file

`biofeat.user.file`
: Name of data file to visualise in the tracks. File names should be comma-separated.

`biofeat.user.type`
: Track type, where multiple tracks can be shown (comma-separated): DataTrack, AnnotationTrack, GeneRegionTrack.

`biofeat.user.type.plot`
: Format of the plot if the data are shown with the Gviz's function called DataTrack (comma-separated)

`list.tracks`
: List of annotation tracks to visualise. Options include geneENSEMBL, CGI, ChromHMM, DNAse, RegENSEMBL, SNP, transcriptENSEMBL, SNPstoma, SNPstru, SNPstrustoma, BindingMotifENSEMBL, otherRegulatoryENSEMBL, regulatoryEvidenceENSEMBL, regulatoryFeaturesENSEMBL, regulatorySegmeENSEMBL, miRNAENSEMBL, ImprintedtissuesGenes, COSMIC, GAD, ClinVar, GeneReviews, GWAS, ClinVarCNV, GCcontent, genesUCSC, xenogenesUCSC, SegDuplication,RepeatElt.

`pattern.regulation`
: The cell/tissue or the list of cells/tissues to visualise in the regulation region defined by Broad ChromHMM

`image.title`
: Title of the plot

`image.name`
: The path and the name of the plot file without extension. The extension will be added by coMET depending on the option image.type.

`image.type`
: Options: pdf or eps

`image.size`
: Default: 3.5 inches. Possible sizes : 3.5 or 7

`fontsize.gviz`
: Font size of writing in annotation track. Default value =5

`font.factor`
: Font size of the sample labels. Range: 0-1

`print.image`
: Print image in file or not.

`config.file`
: 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 and its value are separated 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" and not extra space. (i.e. list.tracks=geneENSEMBL,CGI,ChromHMM,DNAse,RegENSEMBL

`verbose`
: logical option TRUE or FALSE. TRUE (default). If TRUE, shows comments.

Details

The function is limited to visualize 120 omic features.

Value

Create a plot in pdf or eps format depending to some options

Author(s)

Tiphaine Martin

References

http://epigen.kcl.ac.uk/comet/

See Also

comet,comet.list

Examples

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

comet.web(config.file=configfile, mydata.file=myinfofile, cormatrix.file=mycorrelation,
    mydata.large.file=myexpressfile, print.image=FALSE,verbose=FALSE)
```

---

complementary *Complementary or opposite color*

Description

Complementary or opposite color scheme is formed by colors that are opposite each other on the color wheel (example: red and green). The high contrast of complementary colors creates a vibrant look that must be managed well so it is not jarring.

Usage

```
complementary(color, plot = TRUE, bg = "white",
  labcol = NULL, cex = 0.8, title = TRUE)
```

Arguments

color	an R color name or color in hexadecimal notation
plot	logical value indicating whether to plot a color wheel with the generated scheme
bg	background color of the plot. Used only when plot=TRUE
labcol	color for the labels (i.e. names of the colors). Used only when plot=TRUE
cex	numeric value indicating the character expansion of the labels
title	logical value indicating whether to display a title in the plot. Used ony when plot=TRUE

Details

The complementary color is obtained following a color wheel with 12 colors, each one spaced at 30 degrees from each other. Complementary color schemes are tricky to use in large doses, but work well when you wnat something to stand out. In addition, omplementary colors are really bad for text.

Value

A character vector with the given color and the complementary color in hexadecimal notation

Author(s)

Gaston Sanchez

Examples

```
# complementary color of 'tomato' with no plot
opposite("tomato", plot = FALSE)

# complementary color of 'tomato' with color wheel
opposite("tomato", bg = "gray30")
```

---

compute.cormatrix *Compute the correlation matrix between CpG sites*

Description

Compute the correlation matrix between CpG sites

Usage

```
compute.cormatrix(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

gbl.var updated

Author(s)

Tiphaine Martin

`CoreillCNV_UCSC` *Create one track of the genomic positions of CNV in chromosomal aberration and inherited disorders from the NIGMS Human Genetic Cell Repository data*

Description

Create one track of the genomic positions of copy-number variants (CNVs) in chromosomal aberration and inherited disorder cell lines from the NIGMS Human Genetic Cell Repository using the Gviz bioconductor package.

Usage

```
CoreillCNV_UCSC(gen, chr, start, end,title="Coriell CNVs", showId=FALSE)
```

Arguments

`gen`	the name of the genome. Data is not currently available for GRCh38 (hg38).
`chr`	the chromosome of interest
`start`	the first position in the region of interest (the smallest value)
`end`	the last position in the region of interest (the largest value)
`title`	The name of the annotation track
`showId`	Show the ID of the genetic elements

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=cori

See Also

snpLocations_UCSC, structureBiomart_ENSEMBL, snpBiomart_ENSEMBL, COSMIC_UCSC, ClinVarMain_UCSC, ClinVarCnv_UCSC

Examples

```
library("Gviz")
gen <- "hg19"
chrom <- "chr2"
start <- 38290160
end <- 38303219

if(interactive()){
    coreilVariant<-CoreillCNV_UCSC(gen,chrom,start,end)
    plotTracks(coreilVariant, from = start, to =end,
    fontfamily="sans",fontfamily.title="sans")
} else {
     data(coreilVarianttrack)
    plotTracks(coreilVariant, from = start, to =end,
    fontfamily="sans",fontfamily.title="sans")
}
```

COSMIC_UCSC *Create one track of the genomic positions of variants from COSMIC [obselete]*

Description

[obselete] No more possible to extract COSMIC data from UCSC.

Create one track of the genomic positions of variants from COSMIC, the "Catalogue Of Somatic Mutations In Cancer" in extracting data from UCSC and using the Gviz bioconductor package.

Usage

```
COSMIC_UCSC(gen, chr, start, end,title= "COSMIC", showId=FALSE)
```

Arguments

gen	the name of the genome. Data is not currently available for GRCh38 (hg38)
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
title	The name of the annotation track
showId	Show the ID of the genetic elements

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=cos

See Also

snpLocations_UCSC, structureBiomart_ENSEMBL, snpBiomart_ENSEMBL, CoreillCNV_UCSC, ClinVarMain_UCSC, ClinVarCnv_UCSC,

Examples

```
library("Gviz")
chrom <- "chr2"
start <- 38290160
end <- 38303219
gen <- "hg19"
if(interactive()){
    cosmicVariant<-COSMIC_UCSC(gen,chrom,start,end)
    plotTracks(cosmicVariant, from = start, to =end,
    fontfamily="sans",fontfamily.title="sans")
}else {
    data(cosmicVarianttrack)
    plotTracks(cosmicVariant, from = start, to =end,
    fontfamily="sans",fontfamily.title="sans")
}
```

---

cpgIslands_UCSC *create track CpG Island from UCSC*

Description

create track CpG Island from UCSC using the Gviz bioconductor package

Usage

```
cpgIslands_UCSC(gen, chr, start, end, title="CpG Islands UCSC")
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
title	Name of tracks

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=cpg

Examples

```
library("Gviz")
chrom <- "chr2"
start <- 100000
end <- 1000000
gen <- "hg38"

if(interactive()) {
    cpgIstrack<-cpgIslands_UCSC(gen, chrom, start, end)
    plotTracks(cpgIstrack, from = start, to =end,
    fontfamily="sans",fontfamily.title="sans")
}else {
    data(cpgIslandtrack)
    plotTracks(cpgIstrack, from = start, to =end,
    fontfamily="sans",fontfamily.title="sans")
}
```

---

cpgPvalue *Create a plot of pvalue of CpG with DataTrack of Gviz*

Description

Create a plot of pvalue of CpG with DataTrack of Gviz

Usage

```
cpgPvalue(cprange, data, chr, start, end, typefunction, title="CpG pvalue")
```

Arguments

cprange	Range to visualise
data	data to analyse
chr	the chromosome of interest
start	the first position of region of interest (the smallest value)
end	the last position of region of interest (the biggest value)
typefunction	Type of function to visualise the data
title	Name of tracks

Value

the object DataTrack of Gviz

Author(s)

Tiphaine Martin

---

create.color.bar *Create color bar of heatmap*

Description

Create color legend for the correlation matrix

Usage

```
create.color.bar(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

list of different matrix

`color.cut` the matrix of color related to correlation matrix

,

`color.cut.ref` the vector of color related to the reference CpG

,

`cormatrix.key` the generic panel having different panels associated with the correlation matrix

,

`map.label.ldtype`
the panel with the method of creation of correlation matrix

.

`map.label.distance`
the panel with the legend of distance

Author(s)

Tiphaine Martin

`create.color.list` *Create color list for the main data*

Description

Create color list for the main data

Usage

```
create.color.list(config.var, gbl.var)
```

Arguments

`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables

Value

Return a list called split.tmp.color.list which contains the list of color for the info data

Author(s)

Tiphaine Martin

create.color.list.large *Create list of colors for the supplementary data*

Description

Create list of colors for the supplementary data in the plot of pvalue

Usage

```
create.color.list.large(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Return a list called large.split.tmp.color.list which contains the list of color for the extra data

Author(s)

Tiphaine Martin

create.symbol.list *create symbol list for the upper plot in the grid*

Description

create symbol list for the upper plot in the grid

Usage

```
create.symbol.list(config.var, split.color.list, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
split.color.list	
gbl.var	list of internal variables

Value

return a list of symbole:

`split.symbol.list`
: a list which contains the list of symbole for the info data

,

`split.fill.list`
: a list which contains the list of fill for the info data

Author(s)

Tiphaine Martin

`create.symbol.list.large` *Create a list of symblo for the supplementary data*

Description

Create a list of symblo for the supplementary data

Usage

```
create.symbol.list.large(config.var, large.split.color.list, gbl.var)
```

Arguments

`config.var`
: list of all variables defined in configuration file or via options of comet function

`large.split.color.list`
: list of color

`gbl.var`
: list of internal variables

Value

return a list of symbole:

`large.split.symbol.list`
: a list which contains the list of symbole for the extra data

,

`large.split.fill.list`
: a list which contains the list of fill for the extra data

Author(s)

Tiphaine Martin

create.tracks.user *Create track from the user data*

Description

Create track from the user data

Usage

```
create.tracks.user(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Update the object listtracks_user of gbl.var from the user data if the user gives some data for the annotation tracks.

Author(s)

Tiphaine Martin

create.tracks.web *Create tracks for the web page (see cometweb)*

Description

Create tracks for the web page (see cometweb)

Usage

```
create.tracks.web(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Update the object listtracks_gviz of gbl.var with different annotation tracks selected with comet.web

Author(s)

Tiphaine Martin

createList.trackUser *Create list of Gviz's tracks from user's data*

Description

Create list of Gviz's tracks from user's data

Usage

```
createList.trackUser(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Update the object listtracks_user of gbl.var from the user data if the user gives some data for the annotation tracks.

Author(s)

Tiphaine Martin

dgfootprints_RoadMap *Creates a track of DNA motif positional bias in digital genomic Footprinting Sites (DGFP) from a file of RoadMap*

Description

Creates a DGFP track from a file of RoadMap using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
dgfootprints_RoadMap(gen="hg19", chr, start, end, bedFilePath,
tissueGroupDisplay='Blood & T-cell',showId=FALSE, type_stacking="dense",
title= "DGFP RoadMap")
```

Arguments

gen	the name of the genome. Default value=hg19
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
bedFilePath	The file path to the .BED file containing the data to be visualised
tissueGroupDisplay	the group of tissue visualised among list("Neurosph","Epithelial","IMR90","Thymus","Heart","Brain","D & B-cell","Blood & T-cell"="ES-deriv")
showId	logical. say if we write the name of group
type_stacking	Object of class"character", the stacking type of overlapping items on the final plot.One in c(hide, dense, squish, pack,full). More information cf the option "stacking" in Gviz
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to RoadMap Epigenome

Examples

```
library("Gviz")
chr <- "chr1"
start <- 236728
end <- 238778
gen="hg19"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "RoadMap/CD3-DS17198.hg19_subset.bed")

if(interactive()){
  dgfootprints_RoadMapSingle <- dgfootprints_RoadMap(gen,chr,start, end,
  bedFilePath, tissueGroupDisplay='Blood & T-cell' )
  plotTracks(dgfootprints_RoadMapSingle, from = start, to = end,
    fontfamily="sans",fontfamily.title="sans")
} else {
  data(dgfootprints_RoadMapSingle)
```

```
  plotTracks(dgfootprints_RoadMapSingle, from = start, to = end,
    fontfamily="sans",fontfamily.title="sans")
}
```

DNaseI_FANTOM *Creates a enhancer/promoter track from FANTOM*

Description

Creates a track of promoters/enhancers from FANTOM using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
DNaseI_FANTOM(gen="hg19", chr, start, end, bedFilePath,
featureDisplay='enhancer', stacking_type="dense",
title=" DNaseI Fantom")
```

Arguments

gen	the name of the genome. Default value=hg19
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
bedFilePath	The path of the BED file from Kheradpour and Kellis, 2014.
featureDisplay	A vector of regulatory features to be displayed, such as enhancer. Spelling and capitalisation of features must be identical to those in the user guide. There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "Predicted heterochomatin"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("enhancer","promoter")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.
stacking_type	Object of class"character", the stacking type of overlapping items on the final plot.One in c(hide, dense, squish, pack,full). More information cf the option "stacking" in Gviz
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to BindingMotifsBiomart binding motif biomart

Examples

```
library("Gviz")
gen <- "hg19"
chr<- "chr1"
start <- 6000000
end <- 6500000

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
enhFantomFile <- file.path(extdata,
"/FANTOM/human_permissive_enhancers_phase_1_and_2_example970.bed")

if(interactive()){
 enhFANTOMtrack <- DNaseI_FANTOM(gen,chr,start, end,
 enhFantomFile, featureDisplay='enhancer')
  plotTracks(enhFANTOMtrack, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(enhFANTOMtrack)
  plotTracks(enhFANTOMtrack, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}
```

`DNaseI_RoadMap` *Creates a promoter/enhancer regions track from a file of RoadMap*

Description

Creates a track of promoter/enhancer regions from a file of RoadMap using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
DNaseI_RoadMap(gen="hg19", chr, start, end, bedFilePath,
featureDisplay='promotor',showId=TRUE, type_stacking="dense",
title = "DNaseI RoadMap")
```

Arguments

`gen`	the name of the genome. Default value=hg19
`chr`	The chromosome of interest
`start`	The starting position in the region of interest (the smallest value)

end	The end position in the region of interest (the largest value)
bedFilePath	The file path to the .BED file containing the data to be visualised
featureDisplay	A vector of features to be displayed, such as 1_TssA. Spelling and capitalisation of features must be identical to those in the user guide (in the 'State & Acronym' column). There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "1_TssA"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("1_TssA","2_TssAFlnk")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.
showId	Allows to visualise the Id of DNAse group.
type_stacking	Object of class"character", the stacking type of overlapping items on the final plot.One in c(hide, dense, squish, pack,full). More information cf the option "stacking" in Gviz
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to RoadMap Epigenome

Examples

```
library("Gviz")
chr <- "chr2"
start <- 38300049
end <- 38302592
gen="hg19"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "RoadMap/regions_prom_E063.bed")

if(interactive()){
  DNaseI_RoadMapSingle <- DNaseI_RoadMap(gen,chr,start, end,
  bedFilePath, featureDisplay='promotor'  )
  plotTracks(DNaseI_RoadMapSingle, from = start, to = end,
                                   fontfamily="sans",fontfamily.title="sans")
} else {
  data(DNaseI_RoadMapSingle)
```

```
  plotTracks(DNaseI_RoadMapSingle, from = start, to = end,
                                   fontfamily="sans",fontfamily.title="sans")
}
```

DNAse_UCSC *Creation of an UCSC's DNase clusters track - obselete function*

Description

Creation of DNase cluster track from a connection to UCSC genome browser in using the GViz bioconductor package. Obselete function

Usage

```
DNAse_UCSC(gen, chr, start, end, mySession, title="DNA cluster",
track.name = "DNase Clusters", table.name = NULL)
```

Arguments

gen	the name of the genome. Data is not currently available for GRCh38 (hg38).
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
mySession	the object session from the function browserSession of rtracklayer
title	Name of tracks
track.name	the name of the track DNAse_UCSC. "DNase Clusters"(default)
table.name	the name of the table from the track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=wgI

Examples

```
# library("Gviz")
# library("rtracklayer")

# gen <- "hg19"
# chr <- "chr7"
# start <- 38290160
# end <- 38303219
# if(interactive()){
#   BROWSER.SESSION="UCSC"
#   mySession <- browserSession(BROWSER.SESSION)
#   genome(mySession) <- gen
#   track.name="Broad ChromHMM"
#   tablestrack<-tableNames(ucscTableQuery(mySession, track=track.name))
#   table.name<-tablestrack[1]
#   dnasetrack<-DNAse_UCSC(gen,chr,start,end,mySession)
#   plotTracks(dnasetrack, from = start, to =end,
#                                   fontfamily="sans",fontfamily.title="sans")
# }else {
#     data(dnasetrack)
#    plotTracks(dnasetrack, from = start, to =end,
#                                   fontfamily="sans",fontfamily.title="sans")
# }
```

`draw.legend` *Display the legend of the plot*

Description

display the legend of the plot

Usage

```
draw.legend(config.var, gbl.var)
```

Arguments

`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables

Value

Nothing, but the function creates the panel for the legend in the plot

Author(s)

Tiphaine Martin

draw.name.genes.web *display the gene names*

Description

display the gene names for the web page(see cometweb)

Usage

```
draw.name.genes.web(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Updated gbl.var with the list of names of genes found the region of interest. This function is called only in comet.web

Author(s)

Tiphaine Martin

draw.name.tracks.web *Display names of tracks for web page(see cometweb)*

Description

Display names of tracks for web page(see cometweb)

Usage

```
draw.name.tracks.web(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Updated gbl.var with the name of annotation tracks. It is called only in the function comet.web

Author(s)

Tiphaine Martin

draw.plot.annotation *Display the annotation track from ENSEMBL and UCSC*

Description

Display the annotation track from ENSEMBL and UCSC between the plot of pvalue and heatmap

Usage

```
draw.plot.annotation(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Updated the plot with annotation tracks directly.

Author(s)

Tiphaine Martin

draw.plot.axis.data *Display the axis data of plot of pvalue*

Description

Display the axis data of plot of pvalue

Usage

```
draw.plot.axis.data(top.vp, config.var, gbl.var)
```

Arguments

top.vp	The viewport related to top of plot
config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Updated the plot with the different axis of pvalue plot

Author(s)

Tiphaine Martin

`draw.plot.comet` *Display the three plots of coMET*

Description

Display the three plots of coMET according to configuration files : on the upper plot is the plot of pvalue, the middle plot has the annotation tracks, and the lower plot is the heatmap of correlation between CpG sites

Usage

```
draw.plot.comet(config.var, gbl.var, newpage = TRUE)
```

Arguments

`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables
`newpage`	Option to ask if the plot should create on new page or not

Value

Return gbl.var updated with the creation of elements composing the plot produced by the function comet

Author(s)

Tiphaine Martin

`draw.plot.comet.nopval` *Display the three plots of coMET*

Description

Display the three plots of coMET according to configuration files : on the upper plot is the plot of pvalue, the middle plot has the annotation tracks, and the lower plot is the heatmap of correlation between CpG sites

Usage

```
draw.plot.comet.nopval(config.var, gbl.var, newpage = TRUE)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables
newpage	Option to ask if the plot should create on new page or not

Value

Return gbl.var updated with only annotation tracks and correlation matrix visualised

Author(s)

Tiphaine Martin

---

draw.plot.comet.web *Display the three plots of coMET for the web version*

Description

Display the three plots of coMET according to configuration files : on the upper plot is the plot of pvalue, the middle plot has the annotation tracks, and the lower plot is the heatmap of correlation between CpG sites for the web version

Usage

```
draw.plot.comet.web(config.var, gbl.var, newpage = TRUE)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables
newpage	Option to ask if the plot should create on new page or not

Value

Return gbl.var updated with different elements of plot produced by the function comet.web

Author(s)

Tiphaine Martin

draw.plot.cormatrix.plot *Display the correlation plot at the bottom of the grid*

Description

Display the correlation plot at the bottom of the grid

Usage

```
draw.plot.cormatrix.plot(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Return a viewport containing the correlation matrix

Author(s)

Tiphaine Martin

draw.plot.grid.mydata *Display a plot of pvalue of data from MYDATA.FILE*

Description

Display a plot of pvalue of data from MYDATA.FILE

Usage

```
draw.plot.grid.mydata(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Return directly on the upper plot (omic-WAS results) the pvalues related to the primary data

Author(s)

Tiphaine Martin

`draw.plot.grid.mydata.large` *Display the plot of pvalue of the supplementary data*

Description

Display the plot of pvalue of the supplementary data

Usage

```
draw.plot.grid.mydata.large(config.var, gbl.var)
```

Arguments

`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables

Value

Update the upper plot (omic-WAS result plot) with the pvalues from extra data

Author(s)

Tiphaine Martin

`draw.plot.grid.mydata.names` *Display the name of elements defined in DATA.FILE*

Description

Display the name of elements defined in DATA.FILE

Usage

```
draw.plot.grid.mydata.names(config.var, gbl.var)
```

Arguments

`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables

Value

Return the names of omic features vertically between annotation tracks and correlation matrix directly on the plot

Author(s)

Tiphaine Martin

`draw.plot.grid.setup` *Set up the grid of plot*

Description

Set up the grid of plot

Usage

```
draw.plot.grid.setup(config.var, gbl.var)
```

Arguments

`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables

Value

Return gbl.var updated with different layout created

Author(s)

Tiphaine Martin

`draw.plot.linesconnection`
Display the connector lines for the probes

Description

Display the connector lines for the probes

Usage

```
draw.plot.linesconnection(top.vp, config.var, gbl.var)
```

Arguments

`top.vp`	panel of grid to visualise connector lines
`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables

Value

Updated the plot with the connection lines between the genomic position and the position on the correlation matrix

Author(s)

Tiphaine Martin

`draw.plot.mydata.ggbio`

plot tracks created by ggbio that you want to visualise

Description

plot tracks created by ggbio that you want to visualise

Usage

```
draw.plot.mydata.ggbio(config.var, gbl.var, numfile)
```

Arguments

`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables
`numfile`	the number of file to visualise

Value

Return on the plot the annotation tracks created with the functions of ggbio package

Author(s)

Tiphaine Martin

eQTL *Creates a track from a file for eQTL data*

Description

Creates a track from a BED file for eQTL data using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
eQTL(gen,chr, start, end, bedFilePath, featureDisplay, showId=FALSE,
type_stacking="squish",just_group="above", title="eQTL" )
```

Arguments

gen	the name of the genome.
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
bedFilePath	The file path to the .BED file containing the data to be visualised
featureDisplay	A vector of eQTL features to be displayed, such as SNP. Spelling and capitalisation of features must be identical to those in the user guide. There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "CpG"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("SNP","CpG")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.
showId	Allows to visualise the Id of eQTL group.
type_stacking	Object of class"character", the stacking type of overlapping items on the final plot.One in c(hide, dense, squish, pack,full). More information cf the option "stacking" in Gviz
just_group	position. say where we write the name of group (choice in c("above","righ","left"))
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to ENSEMBLregulation binding motif biomart

Examples

```
library("Gviz")
chr <- "chr15"
start <- 74889136
end <-   75018200
featureDisplay <- "SNP"
gen="hg19"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "eQTL.bed")

if(interactive()){
  eQTLTrackSingle <- eQTL(gen,chr,start, end, bedFilePath,
  featureDisplay = featureDisplay )
  plotTracks(eQTLTrackSingle, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(eQTLTrackSingle)
  plotTracks(eQTLTrackSingle, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
}

############

library("Gviz")
chr <- "chr15"
start <- 74889136
end <-   75018200
featureDisplay <- c("SNP","mRNA_pheno")
gen="hg19"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "eQTL.bed")

if(interactive()){
 eQTLTrackMultiple <- eQTL(gen,chr,start, end, bedFilePath, featureDisplay = featureDisplay )
  plotTracks(eQTLTrackMultiple, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(eQTLTrackMultiple)
  plotTracks(eQTLTrackMultiple, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
}

######

library("Gviz")
```

```
chr <- "chr15"
start <- 74889136
end <-   75018200
featureDisplay <- "all"
gen="hg19"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "eQTL.bed")

if(interactive()){
  eQTLTrackAll <- eQTL(gen,chr,start, end, bedFilePath, featureDisplay = featureDisplay )
  plotTracks(eQTLTrackAll, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(eQTLTrackAll)
  plotTracks(eQTLTrackAll, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}
```

---

eQTL_GTEx *Creates a eQTL track from GTEx*

Description

Creates a track of eQTL from GTEx using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
eQTL_GTEx(gen="hg19",chr,start, end, bedFilePath, featureDisplay = 'all',
showId=FALSE, type_stacking="squish",just_group="above",title="eQTL GTEX")
```

Arguments

gen	the name of the genome. Default value=hg19
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
bedFilePath	The path of the BED file from Kheradpour and Kellis, 2014.
featureDisplay	A vector of regulatory features to be displayed, such as Predicted heterochomatin. Spelling and capitalisation of features must be identical to those in the user guide. There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "Predicted heterochomatin"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("Predicted low activity","Predicted heterochomatin")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.

showId	logical. say if we write the name of group
type_stacking	Object of class"character", the stacking type of overlapping items on the final plot.One in c(hide, dense, squish, pack,full). More information cf the option "stacking" in Gviz
just_group	position. say where we write the name of group (choice in c("above","righ","left"))
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to BindingMotifsBiomart binding motif biomart

Examples

```
library("Gviz")
gen <- "hg19"
chr<-"chr3"
start <- 132423172
end <- 132442807
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "/GTEX/eQTL_Uterus_Analysis_extract100.snpgenes")

if(interactive()){
 eGTexTrackall <- eQTL_GTEx(gen,chr,start, end, bedFilePath,
 featureDisplay="all", showId=TRUE,just_group="left")
  plotTracks(eGTexTrackall, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(eGTexTrackall)
  plotTracks(eGTexTrackall, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
}

if(interactive()){
 eGTexTrackSNP <- eQTL_GTEx(gen,chr,start, end, bedFilePath,
 featureDisplay="SNP", showId=TRUE,just_group="left")
  plotTracks(eGTexTrackSNP, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(eGTexTrackSNP)
  plotTracks(eGTexTrackSNP, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

fix.values *Fix and update the values of variables related to main data*

Description

Fix and update the values of variables related to main data

Usage

```
fix.values(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Return the list of config.var and gbl.var updated

Author(s)

Tiphaine Martin

fix.values.generic *Fix and update the values of generic variables*

Description

Fix and update the values of generic variables

Usage

```
fix.values.generic(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Return the list of config.var and gbl.var updated

Author(s)

Tiphaine Martin

`fix.values.large` *Fix and update the values of supplementary data*

Description

Fix and update the values of supplementary data

Usage

```
fix.values.large(config.var, gbl.var)
```

Arguments

`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables

Value

Return the list of config.var and gbl.var updated

Author(s)

Tiphaine Martin

`GAD_UCSC` *Create one track of the genomic positions of variants from the Genetic Association Database (GAD)*

Description

Create one track of the genomic positions of variants from the Genetic Association Database (GAD) (archive of human genetic association studies of complex diseases and disorders) using the Gviz bioconductor package

Usage

```
GAD_UCSC(gen, chr, start, end,title="GAD", showId=FALSE)
```

Arguments

`gen`	the name of the genome. Data is not currently available for GRCh38 (hg38).
`chr`	the chromosome of interest
`start`	the first position in the region of interest (the smallest value)
`end`	the last position in the region of interest (the largest value)
`title`	The name of the annotation track
`showId`	Show the ID of the genetic elements

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=gad

See Also

ISCA_UCSC, GWAScatalog_UCSC, knownGenes_UCSC, genesName_ENSEMBL, GeneReviews_UCSC, genes_ENSEMBL, xenorefGenes_UCSC, transcript_ENSEMBL,

Examples

```
library("Gviz")
gen2 <- "hg19"
chrom2 <- "chr2"
start2 <- 38290160
end2 <- 38303219

if(interactive()) {
  gadtrack<-GAD_UCSC(gen=gen2 ,chr=chrom2 ,start=start2 ,end=end2)
  plotTracks(gadtrack, from = start2, to =end2,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(gadtrack)
   plotTracks(gadtrack, from = start2, to =end2,
                                    fontfamily="sans",fontfamily.title="sans")
}
```

---

gcContent_UCSC *Create one track of GC content from UCSC*

Description

Create a track of GC content from UCSC using the Gviz bioconductor package

Usage

```
gcContent_UCSC(gen, chr, start, end, title="GC Percent")
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
title	Name of tracks

Value

A UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=gc5

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr7"
start <- 38290160
end <- 38303219

if(interactive()){
  gctrack<-gcContent_UCSC(gen,chr,start,end)
  plotTracks(gctrack,from= start, to=end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(gctrack)
  plotTracks(gctrack,from= start, to=end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

GeneReviews_UCSC *Create one track of the genomic positions of variants from GeneReviews*

Description

Create one track of the genomic positions of variants from GeneReviews using the Gviz bioconductor package

Usage

```
GeneReviews_UCSC(gen, chr, start, end,title="GeneReviews", showId=FALSE)
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
title	The name of the annotation track
showId	Show the ID of the genetic elements

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=gen

See Also

ISCA_UCSC, GWAScatalog_UCSC, knownGenes_UCSC, genesName_ENSEMBL, GAD_UCSC, genes_ENSEMBL, xenorefGenes_UCSC, transcript_ENSEMBL,

Examples

```
library("Gviz")
gen <- "hg38"
chrom <- "chr2"
start <- 10000000
end <- 100000000
if(interactive()){
  geneRtrack <-GeneReviews_UCSC(gen,chrom,start,end,showId=TRUE)
  plotTracks(geneRtrack, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(GeneReviewTrack)
  plotTracks(geneRtrack, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

---

genesName_ENSEMBL *Obtain the genes names in the genomic regions of interest from ENSEMBL*

Description

Obtain the genes names in the genomic regions of interest from ENSEMBL

Usage

```
genesName_ENSEMBL(gen, chr, start, end, dataset)
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
dataset	Name of the database to select genes

Details

Can be null

Value

List of name of genes found in this region of interest.

Author(s)

Tiphaine Martin

References

go to ENSEMBL

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

ISCA_UCSC, GWAScatalog_UCSC, knownGenes_UCSC, GeneReviews_UCSC, GAD_UCSC, genes_ENSEMBL, xenorefGenes_UCSC, transcript_ENSEMBL,

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr7"
start <- 38290160
end <- 38303219

if(interactive()){
  dataset<- "hsapiens_gene_ensembl"
  geneNameEnsembl<- genesName_ENSEMBL(gen,chr,start,end,dataset)
  geneNameEnsembl
} else {
  data(geneNameEnsembl)
  geneNameEnsembl
}
```

---

genes_ENSEMBL *Create one track of the genes in the genomic regions of interest from EMSEMBL*

Description

Create one track of the genes in the genomic regions of interest from EMSEMBL using the Gviz bioconductor package

Usage

```
genes_ENSEMBL(gen, chr, start, end, showId=FALSE,title="genes (ENSEMBL)")
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
showId	Show the ID of the genetic elements
title	Name of tracks

Value

A BiomartGeneRegionTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=ens

See Also

ISCA_UCSC, GWAScatalog_UCSC, knownGenes_UCSC, genesName_ENSEMBL, GeneReviews_UCSC, GAD_UCSC, xenorefGenes_UCSC, transcript_ENSEMBL,

Examples

```
library("Gviz")
gen <- "hg19"
chrom <- "chr2"
start <- 38290160
end <- 38303219
if(interactive()) {
  genetrack <-genes_ENSEMBL(gen,chrom,start,end,showId=TRUE)
  plotTracks(genetrack, from = start, to =end,
  fontfamily="sans",fontfamily.title="sans")
} else {
  data(geneENSEMBLtrack)
  plotTracks(genetrack, from = start, to =end,
  fontfamily="sans",fontfamily.title="sans")
}
```

GWAScatalog_UCSC *Create one track of the genomic positions of variants from the GWAS catalog*

Description

Create one track of the genomic positions of variants from the NHGRI Catalog of Published Genome-Wide Association Studies using the Gviz bioconductor package

Usage

```
GWAScatalog_UCSC(gen, chr, start, end, title="GWAS Catalog", showId=FALSE)
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
title	The name of the annotation track
showId	Show the ID of the genetic elements

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=gwa

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

ISCA_UCSC, knownGenes_UCSC, genesName_ENSEMBL, GeneReviews_UCSC, GAD_UCSC, genes_ENSEMBL, xenorefGenes_UCSC, transcript_ENSEMBL,

Examples

```
library("Gviz")
gen <- "hg38"
chrom <- "chr2"
start <- 10000
end <- 100000

if(interactive()) {
  gwastrack <-GWAScatalog_UCSC(gen,chrom,start,end)
  plotTracks(gwastrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(GWASTrack)
  plotTracks(gwastrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

HiCdata2matrix *Creates a HiC matrix from a file (Rao et al., 2014)*

Description

Creates a HiC matrix from Rao et al.,2014.

Usage

```
HiCdata2matrix( chr, start, end, bedFilePath)
```

Arguments

chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
bedFilePath	The path of the BED file from Kheradpour and Kellis, 2014.

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to BindingMotifsBiomart binding motif biomart

Examples

```
library("corrplot")
gen <- "hg19"
chr<-"chr1"
start <- 5000000
end <- 9000000

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "HiC/chr1_1mb.RAWobserved")

if(interactive()){
  matrix_HiC_Rao <- HiCdata2matrix(chr,start, end, bedFilePath)
cor_matrix_HiC <- cor(matrix_HiC_Rao)
diag(cor_matrix_HiC)<-1
corrplot(cor_matrix_HiC, method = "circle")
} else {
```

```
  data(matrix_HiC_Rao)
  cor_matrix_HiC <- cor(matrix_HiC_Rao)
diag(cor_matrix_HiC)<-1
corrplot(cor_matrix_HiC, method = "circle")
}
```

`HistoneAll_UCSC` *Create multiple tracks of histone modifications from the UCSC genome browser*

Description

Create multiple tracks of histone modifications from the UCSC genome browser (ENCODE/Broad) using the Gviz bioconductor package

Usage

```
HistoneAll_UCSC(gen, chr, start, end, mySession, pattern = NULL,
          track.name = "Broad Histone", table.name = NULL)
```

Arguments

`gen`	the name of the genome. Data is not currently available for GRCh38 (hg38).
`chr`	the chromosome of interest
`start`	the first position in the region of interest (the smallest value)
`end`	the last position in the region of interest (the largest value)
`mySession`	the object session from the function browserSession of rtracklayer
`pattern`	The cell type
`track.name`	the name of the track, for example: "Broad Histone"
`table.name`	the name of the table from the track

Value

A list of AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=wgl

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

HistoneOne_UCSC,

Examples

```
library("Gviz")
library("rtracklayer")
gen <- "hg19"
chr <- "chr2"
start <- 38290160
end <- 38313219

if(interactive()){
  BROWSER.SESSION="UCSC"
  mySession <- browserSession(BROWSER.SESSION)
  genome(mySession) <- gen
  pattern1 <- "GM12878"

  histonalltrack<-HistoneAll_UCSC(gen,chr,start,end,mySession,
  pattern=pattern1,track.name="Broad Histone")
  plotTracks(histonalltrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(histonalltrack)
  plotTracks(histonalltrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

HistoneOne_UCSC *Create one track of one histone modification profile from the UCSC genome browser*

Description

Create one track of one histone modification profile from the UCSC genome browser (ENCODE/Broad) using the Gviz bioconductor package

Usage

```
HistoneOne_UCSC(gen, chr, start, end, mySession, title="Broad Histone",
track.name = "Broad Histone", table.name = NULL)
```

Arguments

gen	the name of the genome. Data is not currently available for GRCh38 (hg38).
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)

mySession	the object session from the function browserSession of rtracklayer
title	Name of tracks
track.name	the name of the track, for example: "Broad Histone"
table.name	the name of the table from the track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=wgI

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

HistoneAll_UCSC

Examples

```
library("Gviz")
library("rtracklayer")
gen <- "hg19"
chr <- "chr2"
start <- 38290160
end <- 38303219

if(interactive()) {
  BROWSER.SESSION="UCSC"
  mySession <- browserSession(BROWSER.SESSION)
  genome(mySession) <- gen
  histoneonetrack<-HistoneOne_UCSC(gen,chr,start,end,mySession)
  plotTracks(histoneonetrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(histoneonetrack)
  plotTracks(histoneonetrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

`imprintedGenes_GTEx` *Creates a imprinted genes track from GTEx*

Description

Creates a track of imprinted genes from GTEx using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
imprintedGenes_GTEx(gen="hg19", chr,start, end, tissues="all",
classification="all",showId=FALSE, title="Imprinted genes GTEx")
```

Arguments

`gen`	the name of the genome. Default value=hg19
`chr`	The chromosome of interest
`start`	The starting position in the region of interest (the smallest value)
`end`	The end position in the region of interest (the largest value)
`tissues`	list of tissues among 33 tissues in GTEx
`classification`	list of classification from 5 types (biallelic, consistent with biallelic, consistent with imprinting, imprinted, NC)
`showId`	logical. say if we write the name of group
`title`	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to BindingMotifsBiomart binding motif biomart

Examples

```
library("Gviz")
gen<-"hg19"
chr<- "chr6"
start <- 144251437
end <- 144330541
```

```
if(interactive()){
   allIGtrack <- imprintedGenes_GTEx(gen,chr,start, end,
   tissues="all", classification="imprinted",showId=TRUE)
   allimprintedIGtrack <- imprintedGenes_GTEx(gen,chr,start, end,
   tissues="all", classification="imprinted",showId=TRUE)
   StomachIGtrack <-imprintedGenes_GTEx(gen,chr,start, end,
   tissues="Stomach", classification="all",showId=TRUE)
   PancreasIGtrack <- imprintedGenes_GTEx(gen,chr,start, end,
   tissues="Pancreas", classification="all",showId=TRUE)
   PancreasimprintedIGtrack <- imprintedGenes_GTEx(gen,chr,start, end,
   tissues="Pancreas", classification="biallelic",showId=TRUE)

   imprintinglist <- list(allIGtrack,allimprintedIGtrack,
   StomachIGtrack,PancreasIGtrack,PancreasimprintedIGtrack)

  plotTracks(imprintinglist, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")

} else {

  data(allIGtrack)
  data(allimprintedIGtrack)
  data(StomachIGtrack)
  data(PancreasIGtrack)
  data(PancreasimprintedIGtrack)

   imprintinglist <- list(allIGtrack,allimprintedIGtrack,
   StomachIGtrack,PancreasIGtrack,PancreasimprintedIGtrack)

  plotTracks(imprintinglist, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

`interestGenes_ENSEMBL` *Create one track of the genes in the genomic regions of interest from EMSEMBL*

Description

Create one track of the genes in the genomic regions of interest from EMSEMBL using the Gviz bioconductor package

Usage

```
interestGenes_ENSEMBL(gen, chr, start, end, interestfeatures,interestcolor,
showId=FALSE,title="genes (ENSEMBL)")
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
interestfeatures	A data frame with 3 columns: start of features, end of features, and type of features
interestcolor	A list with the color for each new features defined
showId	Show the ID of the genetic elements
title	Name of tracks

Value

A BiomartGeneRegionTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=ens

See Also

ISCA_UCSC, GWAScatalog_UCSC, knownGenes_UCSC, genesName_ENSEMBL, GeneReviews_UCSC, GAD_UCSC, xenorefGenes_UCSC, transcript_ENSEMBL,

Examples

```
library("Gviz")
gen <- "hg19"
chr <- "chr15"
start <- 75011669
end <- 75019876
interestfeatures <- rbind(c("75011883","75013394","bad"),c("75013932","75014410","good"))
interestcolor <- list("bad"="red", "good"="green")

if(interactive()) {
  interestgenesENSMBLtrack<-interestGenes_ENSEMBL(gen,chr,start,end,
  interestfeatures,interestcolor,showId=TRUE)
  plotTracks(interestgenesENSMBLtrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(interestgenesENSMBLtrack)
  plotTracks(interestgenesENSMBLtrack, from = start, to =end,
```

```
                                    fontfamily="sans",fontfamily.title="sans")
}
```

`interestTranscript_ENSEMBL` *Create a track of transcripts from ENSEMBL*

Description

Create a track to visualize different transcripts from ENSEMBL using the Gviz bioconductor package

Usage

```
interestTranscript_ENSEMBL(gen, chr, start, end,interestfeatures,
interestcolor,showId = FALSE, title="transcripts ENSEMBL")
```

Arguments

`gen`	the name of the genome
`chr`	the chromosome of interest
`start`	the first position in the region of interest (the smallest value)
`end`	the last position in the region of interest (the largest value)
`interestfeatures`	A data frame with 3 columns: start of features, end of features, and type of features
`interestcolor`	A list with the color for each new features defined
`showId`	Show the ID of the genetic elements
`title`	Name of tracks

Value

A BiomartGeneRegionTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=ens

See Also

ISCA_UCSC, GWAScatalog_UCSC, knownGenes_UCSC, genesName_ENSEMBL, GeneReviews_UCSC, GAD_UCSC, genes_ENSEMBL, xenorefGenes_UCSC,

Examples

```
library("Gviz")
gen <- "hg19"
chr <- "chr15"
start <- 75011669
end <- 75019876
interestfeatures <- rbind(c("75017782","75017835","bad"),c("75013755","75013844","good"))
interestcolor <- list("bad"="red", "good"="green")

if(interactive()){
  interesttransENSMBLtrack<-interestTranscript_ENSEMBL(gen,chr,start,end,
  interestfeatures,interestcolor,showId=TRUE)
  plotTracks(interesttransENSMBLtrack, from=start, to=end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(interesttransENSMBLtrack)
  plotTracks(interesttransENSMBLtrack, from=start, to=end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

ISCA_UCSC *Create one track of the genomic positions of variants from ISCA [obselete database]*

Description

Create one track of the genomic positions of variants from International Standards for Cytogenomic Arrays (ISCA) Consortium using the Gviz bioconductor package (obselete datbase, Impossible to access to data from UCSC from September 2015)

Usage

```
ISCA_UCSC(gen, chr, start, end, mySession, table.name,title="ISCA", showId=FALSE)
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
mySession	the object session from the function browserSession of rtracklayer

table.name	A table of ISCAT classifications: iscaBenign, iscaCuratedBenign, iscaCurated-Pathogenic, iscaLikelyBenign, iscaLikelyPathogenic, iscaPathGainCum, iscaP-athLossCum, iscaPathogenic, iscaUncertain
title	The name of the annotation track
showId	Show the ID of the genetic elements

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=isca

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

GWAScatalog_UCSC, knownGenes_UCSC, genesName_ENSEMBL, GeneReviews_UCSC, GAD_UCSC, genes_ENSEMBL, xenorefGenes_UCSC, transcript_ENSEMBL,

Examples

```
# Oboselet function

library("Gviz")
library("rtracklayer")
gen <- "hg19"
chr <- "chr2"
start <- 38292433
end <- 38305492

if(interactive()){
  BROWSER.SESSION="UCSC"
  mySession <- browserSession(BROWSER.SESSION)
  genome(mySession) <- gen
 iscatrack <-ISCA_UCSC(gen,chrom,start,end,mySession,title="ISCA", table="iscaPathogenic")
  plotTracks(iscatrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(ISCAtrack_Grch38)
  plotTracks(iscatrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

knownGenes_UCSC *Create a track of known genes from the UCSC genome browser*

Description

Create a track of known genes from the UCSC genome browser using the Gviz bioconductor package

Usage

```
knownGenes_UCSC(gen, chr, start, end,title="UCSC known Genes", showId=TRUE)
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
title	Name of tracks
showId	Show the ID of the genetic elements

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=kno

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

ISCA_UCSC, GWAScatalog_UCSC, genesName_ENSEMBL, GeneReviews_UCSC, GAD_UCSC, genes_ENSEMBL, xenorefGenes_UCSC, transcript_ENSEMBL,

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr7"
start <- 38290160
end <- 38303219

if(interactive()) {
  genesUcsctrack<-knownGenes_UCSC(gen,chr,start,end,showId=TRUE)
  plotTracks(genesUcsctrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
}else {
  data(genesUcsctrack)
  plotTracks(genesUcsctrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

`metQTL` *Creates a track from a file for metQTL data*

Description

Creates a track from a BED file for metQTL data using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
metQTL(gen, chr, start, end, bedFilePath, featureDisplay, showId=FALSE,
type_stacking="squish",just_group="above", title="metQTL")
```

Arguments

`gen`	the name of the genome. Default value=hg19
`chr`	The chromosome of interest
`start`	The starting position in the region of interest (the smallest value)
`end`	The end position in the region of interest (the largest value)
`bedFilePath`	The file path to the .BED file containing the data to be visualised
`featureDisplay`	A vector of metQTL features to be displayed, such as SNP. Spelling and capitalisation of features must be identical to those in the user guide. There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "CpG"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("SNP","CpG")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.

showId	Allows the visualization of the Id of metQTL group.
type_stacking	Sets the type of stacking used by Gviz for plots. By default this is set to 'squish'. For more information see Gviz user guide.
just_group	position. say where we write the name of group (choice in c("above","righ","left"))
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to ENSEMBLregulation binding motif biomart

Examples

```
library("Gviz")

gen <- 'hg19'
chr <- "chr15"
start <- 74889136
end <-   75018200
featureDisplay <- "trans_local_metQTL"
type_stacking <- "squish"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
mqtlbedFilePath <- file.path(extdata, "metQTL.bed")

if(interactive()){
  metQTLTrackSingle <- metQTL(gen,chr,start, end,mqtlbedFilePath,
  featureDisplay = featureDisplay )
  plotTracks(metQTLTrackSingle, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(metQTLTrackSingle)
  plotTracks(metQTLTrackSingle, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
}

###

library("Gviz")

gen <- 'hg19'
```

```
chr <- "chr15"
start <- 74889136
end <-   75018200

featureDisplay <- c("trans_local_metQTL","CpG")

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "metQTL.bed")

if(interactive()){
  metQTLTrackMultiple <- metQTL(gen,chr,start, end, bedFilePath,
  featureDisplay = featureDisplay )
  plotTracks(metQTLTrackMultiple, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(metQTLTrackMultiple)
  plotTracks(metQTLTrackMultiple, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
}

#####

library("Gviz")

gen <- 'hg19'
chr <- "chr15"
start <- 74889136
end <-   75018200

featureDisplay <- "all"

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
bedFilePath <- file.path(extdata, "metQTL.bed")

if(interactive()){
  metQTLTrackAll <- metQTL(gen,chr,start, end, bedFilePath,
  featureDisplay = featureDisplay )
  plotTracks(metQTLTrackAll, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(metQTLTrackAll)
  plotTracks(metQTLTrackAll, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

---

miRNATargetRegionsBiomart_ENSEMBL

Creates a track of miRNA target regions from ENSEMBL

Description

Creates a track of miRNA target regions from ENSEMBL using the Gviz bioconductor package.

Usage

```
miRNATargetRegionsBiomart_ENSEMBL(gen, chr, start, end, showId=FALSE,
datasetEnsembl = "hsapiens_mirna_target_feature",
title="miRNA Target Regions ENSEMBL")
```

Arguments

gen	The name of the genome. Currently only handles human data from either the previous version, GRCh37 (also known as hg19) or the current version, GRCh38 (also known as hg38).
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
showId	Show the ID of the genetic elements
datasetEnsembl	Allows the user to manually set which data set is used if required.Default=hsapiens_mirna_target_feature
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to ENSEMBLregulation binding motif biomart

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 1000000
end <- 20000000


if(interactive()){
 miRNATargetRegionsBiomartTrack<-miRNATargetRegionsBiomart_ENSEMBL(gen,chr,start,end,
                                    datasetEnsembl = "hsapiens_mirna_target_feature")
  plotTracks(miRNATargetRegionsBiomartTrack, from = start, to = end,
```

```
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(miRNATargetRegionsBiomartTrack)
  plotTracks(miRNATargetRegionsBiomartTrack, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}
```

`otherRegulatoryRegionsBiomart_ENSEMBL`
Creates a track of other regulatory regions from ENSEMBL

Description

Creates a track from ENSEMBL of other regulatory regions using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
otherRegulatoryRegionsBiomart_ENSEMBL(gen, chr, start, end,
featureDisplay = "all",datasetEnsembl = "hsapiens_external_feature",
title="Other Regulatory Regions ENSEMBL")
```

Arguments

`gen`	The name of the genome. Currently only handles human data from either the previous version, GRCh37 (also known as hg19) or the current version, GRCh38 (also known as hg38).
`chr`	The chromosome of interest
`start`	The starting position in the region of interest (the smallest value)
`end`	The end position in the region of interest (the largest value)
`featureDisplay`	A vector of regulatory features to be displayed, such as Enhancer. Spelling and capitalisation of features must be identical to those in the user guide. There are two possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "Enhancer"), only the name of the specific feature is required. Second, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.
`datasetEnsembl`	Allows the user to manually set which data set is used if required.
`title`	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to ENSEMBLregulation binding motif biomart

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 100000
end <- 5000000
featureDisplay <- "Enhancer"

if(interactive()){
 otherRegulatoryRegionsTrackSingle<-otherRegulatoryRegionsBiomart_ENSEMBL(gen,
 chr,start,end,featureDisplay)
  plotTracks(otherRegulatoryRegionsTrackSingle, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(otherRegulatoryRegionsTrackSingle)
  plotTracks(otherRegulatoryRegionsTrackSingle, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}

########

library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 100000
end <- 5000000
featureDisplay <- "all"

if(interactive()){
 otherRegulatoryRegionsTrackAll<-otherRegulatoryRegionsBiomart_ENSEMBL(gen,
 chr,start,end,featureDisplay)
  plotTracks(otherRegulatoryRegionsTrackAll, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(otherRegulatoryRegionsTrackAll)
  plotTracks(otherRegulatoryRegionsTrackAll, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")

}
```

pizza *Pizza color wheel*

Description

This function displays a color wheel with specified colors

Usage

```
pizza(colors, bg = "gray95", border = NA,
  init.angle = 105, cex = 0.8, lty = 1, labcol = NULL,
  ...)
```

Arguments

colors	a vector with R color names of colors in hexadecimal notation
bg	background color of the plot. Default "gray95"
border	color of the border separating the pizza slices
init.angle	integer value indicating the start angle (in degrees) for the slices
cex	numeric value indicating the character expansion of the labels
lty	argument passed to polygon which draws each slice
labcol	color for the labels (i.e. names of the colors)
...	graphical parameters (par) can be given as argument to pizza

Details

This function is based on the pie function

Author(s)

Gaston Sanchez

Examples

```
# pizza color wheel for rainbow colors
pizza(rainbow(7))

# pizza color wheel for tomato (18 colors)
pizza(setColors("tomato", 18), bg = "gray20", cex = 0.7)
```

printPlot.comet *Create the plot on file from coMet function*

Description

Create the plot on file from coMet function

Usage

```
printPlot.comet(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Return the plot produced by comet function in a file

Author(s)

Tiphaine Martin

printPlot.comet.nopval *Create the plot on file from coMet function*

Description

Create the plot on file from coMet function

Usage

```
printPlot.comet.nopval(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Return the plot produced by comet function without the upper plot (the pvalue plot) in a file

Author(s)

Tiphaine Martin

printPlot.comet.web *Display the plot from cometWeb funciton*

Description

Display the plot from cometWeb funciton

Usage

```
printPlot.comet.web(config.var, gbl.var)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables

Value

Return the plot produced by comet.web function in a file

Author(s)

Tiphaine Martin

psiQTL_GTEx *Creates a psiQTL track from GTEx*

Description

Creates a track of psiQTL from GTEx using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
psiQTL_GTEx(gen,chr,start, end, bedFilePath, featureDisplay = 'all',
showId=FALSE, type_stacking="squish",just_group="above", title="psiQTL GTEX")
```

Arguments

gen	the name of the genome.
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
bedFilePath	The path of the BED file from Kheradpour and Kellis, 2014.

featureDisplay	A vector of regulatory features to be displayed, such as Predicted heterochomatin. Spelling and capitalisation of features must be identical to those in the user guide. There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "Predicted heterochomatin"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("Predicted low activity","Predicted heterochomatin")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.
showId	logical. say if we write the name of group
type_stacking	Object of class"character", the stacking type of overlapping items on the final plot.One in c(hide, dense, squish, pack,full). More information cf the option "stacking" in Gviz
just_group	position. say where we write the name of group (choice in c("above","righ","left"))
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to BindingMotifsBiomart binding motif biomart

Examples

```
library("Gviz")
gen <- "hg19"
chr<-"chr13"
start <- 52713837
end <- 52715894
extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
psiQTLFilePath <- file.path(extdata, "/GTEX/psiQTL_Assoc-total.AdiposeTissue.txt")


if(interactive()){
  psiGTexTrackall<- psiQTL_GTEx(gen,chr,start, end, psiQTLFilePath,
  featureDisplay = 'all', showId=TRUE, type_stacking="squish",
  just_group="above" )
  plotTracks(psiGTexTrackall, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(psiGTexTrackall)
```

```
  plotTracks(psiGTexTrackall, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}

if(interactive()){
 psiGTexTrackSNP<- psiQTL_GTEx(gen,chr,start, end, psiQTLFilePath,
 featureDisplay = 'SNP', showId=TRUE, type_stacking="squish",
 just_group="left")
  plotTracks(psiGTexTrackSNP, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(psiGTexTrackSNP)
  plotTracks(psiGTexTrackSNP, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}
```

`read.config` *Extract the values of variables from configuration file*

Description

Extract the values of variables from configuration file

Usage

```
read.config(config.file, config.var)
```

Arguments

`config.file`	Configuration file
`config.var`	list of all variables defined in configuration file or via options of comet function

Value

Return config.var updated with the different values of options found in the configuration file

Author(s)

Tiphaine Martin

read.file.cormatrix *Read, compute and extract the values from correlation matrix file*

Description

Read, compute and extract the values from correlation matrix file

Usage

```
read.file.cormatrix(config.var, gbl.var, split.cormatrix.file = NULL)
```

Arguments

config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables
split.cormatrix.file	File of correlation matrix

Value

Return gbl.var updated with the raw data of the correlation matrix and the correlation matrix computed

Author(s)

Tiphaine Martin

read.file.mydata *Read the files of main data and extract data*

Description

Read the files of main data and extract data

Usage

```
read.file.mydata(split.mydata.file, config.var, gbl.var, numfile)
```

Arguments

split.mydata.file	List of files to main data
config.var	list of all variables defined in configuration file or via options of comet function
gbl.var	list of internal variables
numfile	The number of file in the list

Value

Return gbl.var updated after reading the primary data

Author(s)

Tiphaine Martin

`read.file.mydata.large` *Read the files of supplemantary data and extract data*

Description

Read the files of supplemantary data and extract data

Usage

```
read.file.mydata.large(large.split.mydata.file, config.var, gbl.var, numfile.large)
```

Arguments

`large.split.mydata.file`	List of supplementary files to check format
`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables
`numfile.large`	The number of file to check in the list

Value

Return gbl.var updated after reading the extra data

Author(s)

Tiphaine Martin

refGenes_UCSC *Create a track of RefSeq genes from the UCSC genome browser*

Description

Create a track of RefSeq genes from the UCSC genome browser using the Gviz bioconductor package

Usage

```
refGenes_UCSC(gen, chr, start, end, title="Ref Genes UCSC", track = "refGene",
IdType="Ref", showId=TRUE)
```

Arguments

gen	The name of the genome
chr	The chromosome of interest
start	The first position in the region of interest (the smallest value)
end	The last position in the region of interest (the largest value)
title	Name of tracks
track	the name of table in UCSC for the group "Genes and Gene Prediction"
IdType	When set to 'ref' shows the gene reference, when set to "name" shows the gene name
showId	Shows the ID or name of the genetic elements

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=kno

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

ISCA_UCSC, GWAScatalog_UCSC, genesName_ENSEMBL, GeneReviews_UCSC, GAD_UCSC, genes_ENSEMBL, xenorefGenes_UCSC, transcript_ENSEMBL, knownGenes_UCSC

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr7"
start <- 38203219
end <- 38303219
IdType <- "name"
track <- "refGene"

if(interactive()) {
  genesUcsctrack<-refGenes_UCSC(gen,chr,start,end,track,IdType)
  plotTracks(genesUcsctrack, from = start, to =end)
}else {
  data(genesUcsctrack)
  plotTracks(genesUcsctrack, from = start, to =end)
}
```

---

regulationBiomart_ENSEMBL

Create a regulation track from ENSEMBL

Description

Create a 'Regulation' track from ENSEMBL using the Gviz bioconductor package

Usage

```
regulationBiomart_ENSEMBL(gen, chr, start, end,title="Regulation ENSEMBL")
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
title	Name of tracks

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to ENSEMBLregulation biomart

Examples

```
library("Gviz")
gen <- "hg19"
chr <- "chr7"
start <- 38290160
end <- 38303219

if(interactive()){
  regulationENSEMBLtrack<-regulationBiomart_ENSEMBL(gen,chr,start,end)
  plotTracks(regulationENSEMBLtrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(regulationENSEMBLtrack)
  plotTracks(regulationENSEMBLtrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

---

regulatoryEvidenceBiomart_ENSEMBL

Creates a regulatory feature track from ENSEMBL

Description

Creates a regulatory feature track from ENSEMBL using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
regulatoryEvidenceBiomart_ENSEMBL (gen, chr, start, end,
featureDisplay = "all", datasetEnsembl = "hsapiens_annotated_feature",
title="Other Regulatory Regions ENSEMBL")
```

Arguments

gen	The name of the genome. Currently only handles human data from either the previous version, GRCh37 (also known as hg19) or the current version, GRCh38 (also known as hg38).
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)

featureDisplay	A vector of regulatory features to be displayed, such as DNase1. Spelling and capitalisation of features must be identical to those in the user guide. There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "DNase1"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("CTCF","DNase1")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.
datasetEnsembl	Allows the user to manually set which data set is used if required.
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to ENSEMBLregulation binding motif biomart

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 40000
end <- 50000
featureDisplay <- "H3K27me3"

if(interactive()){
 regulatoryEvidenceBiomartTrackSingle <- regulatoryEvidenceBiomart_ENSEMBL(gen,
 chr,start,end,featureDisplay)
  plotTracks(regulatoryEvidenceBiomartTrackSingle, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(regulatoryEvidenceBiomartTrackSingle)
  plotTracks(regulatoryEvidenceBiomartTrackSingle, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
}

#####

library("Gviz")
gen <- "hg38"
```

```
chr <- "chr1"
start <- 40000
end <- 100000
featureDisplay <- c("H3K27me3","H3K36me3")

if(interactive()){
 regulatoryEvidenceBiomartTrackMultiple<-regulatoryEvidenceBiomart_ENSEMBL (gen,
 chr,start,end,featureDisplay)
  plotTracks(regulatoryEvidenceBiomartTrackMultiple, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(regulatoryEvidenceBiomartTrackMultiple)
  plotTracks(regulatoryEvidenceBiomartTrackMultiple, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}

#####

library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 50000
end <- 100000
featureDisplay <- "all"
if(interactive()){
 regulatoryEvidenceBiomartTrackAll<-regulatoryEvidenceBiomart_ENSEMBL (gen,
 chr,start,end,featureDisplay)
  plotTracks(regulatoryEvidenceBiomartTrackAll, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(regulatoryEvidenceBiomartTrackAll)
  plotTracks(regulatoryEvidenceBiomartTrackAll, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")

}
```

`regulatoryFeaturesBiomart_ENSEMBL`
Creates a regulatory feature track from ENSEMBL

Description

Creates a regulatory feature track from ENSEMBL using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
regulatoryFeaturesBiomart_ENSEMBL(gen, chr, start, end, featureDisplay = "all",
datasetEnsembl = "hsapiens_regulatory_feature",
title="Regulatory Features ENSEMBL")
```

Arguments

gen	The name of the genome. Currently only handles human data from either the previous version, GRCh37 (also known as hg19) or the current version, GRCh38 (also known as hg38).
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
featureDisplay	A vector of regulatory features to be displayed, such as Promoter. Spelling and capitalisation of features must be identical to those in the user guide. There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "Promoter"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("TF binding site","Promoter")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.
datasetEnsembl	Allows the user to manually set which data set is used if required.Default=hsapiens_regulatory_feature
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to ENSEMBLregulation binding motif biomart

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 10000
end <- 500000
featureDisplay <- "Enhancer"

if(interactive()){
 regulatoryFeaturesBiomartTrackSingle<-regulatoryFeaturesBiomart_ENSEMBL(gen,
 chr,start,end,featureDisplay)
  plotTracks(regulatoryFeaturesBiomartTrackSingle, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
```

```
} else {
  data(regulatoryFeaturesBiomartTrackSingle)
  plotTracks(regulatoryFeaturesBiomartTrackSingle, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}

#####

library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 10000
end <- 100000
featureDisplay <- c("CTCF Binding Site","Enhancer")

if(interactive()){
 regulatoryFeaturesBiomartTrackMultiple<-regulatoryFeaturesBiomart_ENSEMBL(gen,
 chr,start,end,featureDisplay)
  plotTracks(regulatoryFeaturesBiomartTrackMultiple, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(regulatoryFeaturesBiomartTrackMultiple)
  plotTracks(regulatoryFeaturesBiomartTrackMultiple, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}

#####

library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 10000
end <- 50000
featureDisplay <- "all"
if(interactive()){
 regulatoryFeaturesBiomartTrackAll<-regulatoryFeaturesBiomart_ENSEMBL(gen,
 chr,start,end,featureDisplay)
  plotTracks(regulatoryFeaturesBiomartTrackAll, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(regulatoryFeaturesBiomartTrackAll)
  plotTracks(regulatoryFeaturesBiomartTrackAll, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")

}
```

---

regulatorySegmentsBiomart_ENSEMBL

Creates a binding motif track from ENSEMBL [obselete]

Description

[obselete] Creates a track of regulatory segments from ENSEMBL using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
regulatorySegmentsBiomart_ENSEMBL(gen, chr, start, end, featureDisplay = 'all',
datasetEnsembl = "hsapiens_external_feature",
title="External Regulatory ENSEMBL")
```

Arguments

gen	The name of the genome. Currently only handles human data from either the previous version, GRCh37 (also known as hg19) or the current version, GRCh38 (also known as hg38).
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
featureDisplay	A vector of regulatory features to be displayed, such as Predicted heterochomatin. Spelling and capitalisation of features must be identical to those in the user guide. There are three possibilities. First, the visualisation of only one feature (e.g. featureDisplay <- "Predicted heterochomatin"), only the name of the specific feature is required. Second, visualisation of a set of features, for this a vector of features is required (e.g. featureDisplay <- c("Predicted low activity","Predicted heterochomatin")). Finally, visualison all features in the genomic region, achived by using the word "all" (e.g. featureDisplay <- "all"), "all" is set by default. You can find the complete list of features and their associated colours in the user guide.
datasetEnsembl	Allows the user to manually set which data set is used if required.Default=hsapiens_segmentation_feature
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to ENSEMBLregulation binding motif biomart

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 10000
end <- 50000
featureDisplay <- "CTCF enriched"

if(interactive()){
 regulatorySegmentsBiomartTrackSingle<-regulatorySegmentsBiomart_ENSEMBL(gen,
 chr,start,end,featureDisplay)
  plotTracks(regulatorySegmentsBiomartTrackSingle, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(regulatorySegmentsBiomartTrackSingle)
  plotTracks(regulatorySegmentsBiomartTrackSingle, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}

####

library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 10000
end <- 50000
featureDisplay <- c("CTCF enriched","Predicted Promoter Flank")

if(interactive()){
 regulatorySegmentsBiomartTrackMultiple<-regulatorySegmentsBiomart_ENSEMBL(gen,
 chr,start,end,featureDisplay)
  plotTracks(regulatorySegmentsBiomartTrackMultiple, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(regulatorySegmentsBiomartTrackMultiple)
  plotTracks(regulatorySegmentsBiomartTrackMultiple, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}

####

library("Gviz")
gen <- "hg38"
chr <- "chr1"
start <- 10000
end <- 50000
featureDisplay <- "all"
if(interactive()){
 regulatorySegmentsBiomartTrackAll<-regulatorySegmentsBiomart_ENSEMBL(gen,
 chr,start,end,featureDisplay)
  plotTracks(regulatorySegmentsBiomartTrackAll, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
```

```
} else {
  data(regulatorySegmentsBiomartTrackAll)
  plotTracks(regulatorySegmentsBiomartTrackAll, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")

}
```

`repeatMasker_UCSC` *Create one track of the genomic positions of regions from repeatMasker_UCSC*

Description

Create one track of the genomic positions of regions from repeatMasker_UCSC using the Gviz bioconductor package

Usage

```
repeatMasker_UCSC(gen, chr, start, end, title="RepeatMasker",
showId=FALSE,type_stacking="full")
```

Arguments

`gen`	the name of the genome
`chr`	the chromosome of interest
`start`	the first position in the region of interest (the smallest value)
`end`	the last position in the region of interest (the largest value)
`title`	The name of the annotation track
`showId`	Show the ID of the genetic elements
`type_stacking`	the type of stacking data for this track. More information go to Gviz (the option "stacking")

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=rms

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr2"
start <- 38290160
end <- 38303219

if(interactive()){
  rmtrack <-repeatMasker_UCSC(gen,chr,start,end,showId=TRUE)
  plotTracks(rmtrack, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
} else {
  data(repeatMaskerTrack)
  plotTracks(rmtrack, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}
```

`retrieve.data` *Retrieve the data from configuration file and data files*

Description

Retrieve the data from configuration file and data files

Usage

```
retrieve.data(config.var, gbl.var)
```

Arguments

`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables

Value

Return config.var and gbl.var updated after reading all data files given

Author(s)

Tiphaine Martin

segmentalDups_UCSC *Create one track of the genomic positions of regions from segmentalDups_UCSC*

Description

Create one track of the genomic positions of regions from segmentalDups_UCSC using the Gviz bioconductor package

Usage

```
segmentalDups_UCSC(gen, chr, start, end,title="Segmental Dups UCSC")
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
title	The name of the annotation track

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

Tom Hardiman

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=rms

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr2"
start <- 100000
end <- 200000

if(interactive()){
  DupTrack <-segmentalDups_UCSC(gen,chr,start,end)
  plotTracks(DupTrack, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
```

```
} else {
  data(DupTrack)
  plotTracks(DupTrack, from = start, to = end,
                                    fontfamily="sans",fontfamily.title="sans")
}
```

`set.image.parameters` *Set up the parameters of image*

Description

Set up the parameters of image

Usage

```
set.image.parameters(config.var, gbl.var)
```

Arguments

`config.var`	list of all variables defined in configuration file or via options of comet function
`gbl.var`	list of internal variables

Value

Return a list of parameter for the production of image

Author(s)

Tiphaine Martin

`setColors` *Set Colors for a color wheel*

Description

This function set a given number of colors to create a color wheel

Usage

```
setColors(color, num)
```

Arguments

`color`	an R color name or a color in hexadecimal notation
`num`	integer value indicating how many colors to be added to the wheel

Value

A character vector with the given color and the set of colors to create a wheel color

Author(s)

Gaston Sanchez

See Also

col2HSV

Examples

```
# create a color wheel based on 'tomato'
setColors("tomato", 12)

# set 7 colors for '#3D6DCC'
setColors("#3D6DCC", 7)
```

---

snpBiomart_ENSEMBL *Create a short variation track from ENSEMBL*

Description

Create a 'Short Variation' track from ENSEMBL using the Gviz bioconductor package

Usage

```
snpBiomart_ENSEMBL(gen,chr, start, end, dataset, showId=FALSE, title = "SNPs ENSEMBL")
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
dataset	The name of the database. Example "hsapiens_snp_som"
showId	Show the the ID of element or not
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

Go to ENSEMBL Biomart

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

snpLocations_UCSC, structureBiomart_ENSEMBL, COSMIC_UCSC, CoreillCNV_UCSC, ClinVarMain_UCSC, ClinVarCnv_UCSC,

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr2"
start <- 38290160
end <- 38303219

if(interactive()){
  snptrack <- snpBiomart_ENSEMBL(gen,chr, start, end,
                       dataset="hsapiens_snp",showId=FALSE)
  plotTracks(snptrack, from=start, to=end,
                                   fontfamily="sans",fontfamily.title="sans")
} else {
  data(snpBiomarttrack)
  plotTracks(snptrack, from=start, to=end,
                                   fontfamily="sans",fontfamily.title="sans")
}
```

---

snpLocations_UCSC *Create a SNP track from UCSC*

Description

Create a SNP track from UCSC using the Gviz bioconductor package

Usage

```
snpLocations_UCSC(gen, chr, start, end,title= "SNPs UCSC", track="All SNPs(142)")
```

Arguments

gen	the name of the genome. Data is not currently available for GRCh38 (hg38).
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
title	Name of tracks
track	The name of the database. Default "All SNPs(142)"

Value

An UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=snp

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

snpLocations_UCSC, structureBiomart_ENSEMBL, COSMIC_UCSC, CoreillCNV_UCSC, ClinVarMain_UCSC, ClinVarCnv_UCSC,

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr7"
start <- 38290160
end <- 38303219

if(interactive()) {
  snpUCSCtrack<-snpLocations_UCSC(gen,chr,start,end,"All SNPs(142)")
  plotTracks(snpUCSCtrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(snpUCSCtrack)
  plotTracks(snpUCSCtrack, from = start, to =end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

---

structureBiomart_ENSEMBL

Create a structural variation track from ENSEMBL

Description

Create a 'Structural Variation' track from ENSEMBL using the Gviz bioconductor package

Usage

```
structureBiomart_ENSEMBL(gen, chr, start, end, strand, dataset,
showId=FALSE, title = "Structural variation")
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
strand	the strand to extract structure data for
dataset	The name of the database. Example "hsapiens_structvar_som"
showId	Show the the ID of the element
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

Go to ENSEMBL Biomart

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

snpLocations_UCSC, snpBiomart_ENSEMBL, COSMIC_UCSC, CoreillCNV_UCSC, ClinVarMain_UCSC, ClinVarCnv_UCSC,

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr2"
start <- 38290160
end <- 38303219

if(interactive()){
  strutrack <- structureBiomart_ENSEMBL(chr, start, end,
                              strand, dataset="hsapiens_structvar_som")
  plotTracks(strutrack, from=start, to=end,
                                  fontfamily="sans",fontfamily.title="sans")
}else {
  data(strucBiomarttrack)
  plotTracks(strutrack, from=start, to=end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

TFBS_FANTOM *Creates a TFBS motif track from FANTOM*

Description

Creates a track of TFBS motifs from FANTOM using the Gviz bioconductor package. A complete list of features and their associated colours can be found in the user guide.

Usage

```
TFBS_FANTOM(gen, chr, start, end, bedFilePath,title="TF motif FANTOM5")
```

Arguments

gen	the name of the genome.
chr	The chromosome of interest
start	The starting position in the region of interest (the smallest value)
end	The end position in the region of interest (the largest value)
bedFilePath	The path of the BED file from Kheradpour and Kellis, 2014.
title	The name of the annotation track

Value

An AnnotationTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

Got to BindingMotifsBiomart binding motif biomart

Examples

```
library("Gviz")
gen <- "hg19"
chr<- "chr1"
start <- 6000000
end <- 6500000

extdata <- system.file("extdata", package="coMET",mustWork=TRUE)
AP1FantomFile <- file.path(extdata, "FANTOM/Fantom_hg19.AP1_MA0099.2.sites_example970.txt")

if(interactive()){
  tfbsFANTOMtrack <- TFBS_FANTOM(gen,chr,start, end, AP1FantomFile)
```

```
  plotTracks(tfbsFANTOMtrack, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(tfbsFANTOMtrack)
  plotTracks(tfbsFANTOMtrack, from = start, to = end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

`transcript_ENSEMBL` *Create a track of transcripts from ENSEMBL*

Description

Create a track to visualize different transcripts from ENSEMBL using the Gviz bioconductor package

Usage

```
transcript_ENSEMBL(gen, chr, start, end,showId = FALSE, title="transcripts ENSEMBL")
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
showId	Show the ID of the genetic elements
title	Name of tracks

Value

A BiomartGeneRegionTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

http://genome-euro.ucsc.edu/cgi-bin/hgTrackUi?hgsid=202839739_2hYQ1BAOuBMAR620GjrtdrFAy6dn&c=chr6&g=ens

See Also

ISCA_UCSC, GWAScatalog_UCSC, knownGenes_UCSC, genesName_ENSEMBL, GeneReviews_UCSC, GAD_UCSC, genes_ENSEMBL, xenorefGenes_UCSC,

Examples

```
library("Gviz")
gen <- "hg19"
chr <- "chr2"
start <- 32290160
end <- 33303219

if(interactive()){
  transENSMBLtrack<-transcript_ENSEMBL(gen,chr,start,end,showId=TRUE)
  plotTracks(transENSMBLtrack, from=start, to=end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(transENSMBLtrack)
  plotTracks(transENSMBLtrack, from=start, to=end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

---

xenorefGenes_UCSC *Create a track for xeno-reference genes from the UCSC genome browser*

Description

Create a track for xeno-reference genes from the UCSC genome browser using the Gviz bioconductor package

Usage

```
xenorefGenes_UCSC(gen, chr, start, end,title="Other RefSeq", showId=FALSE)
```

Arguments

gen	the name of the genome
chr	the chromosome of interest
start	the first position in the region of interest (the smallest value)
end	the last position in the region of interest (the largest value)
title	Name of tracks
showId	Show the ID of the genetic elements

Value

A UcscTrack object of Gviz

Author(s)

Tiphaine Martin

References

http://bioconductor.org/packages/release/bioc/html/Gviz.html

See Also

ISCA_UCSC, GWAScatalog_UCSC, knownGenes_UCSC, genesName_ENSEMBL, GeneReviews_UCSC, GAD_UCSC, genes_ENSEMBL,transcript_ENSEMBL,

Examples

```
library("Gviz")
gen <- "hg38"
chr <- "chr7"
start <- 38290160
end <- 38303219

if(interactive()){
  xenogenestrack<-xenorefGenes_UCSC(gen,chr,start,end,showId=TRUE)
  plotTracks(xenogenestrack, from=start, to=end,
                                  fontfamily="sans",fontfamily.title="sans")
} else {
  data(xenogenestrack)
  plotTracks(xenogenestrack, from=start, to=end,
                                  fontfamily="sans",fontfamily.title="sans")
}
```

Index

∗ **datasets**
clinCNV, 17
∗ **dplot**
chrUCSC2ENSEMBL, 16
create.tracks.web, 43
createList.trackUser, 44
fix.values, 63
fix.values.generic, 63
fix.values.large, 64
printPlot.comet, 90
read.config, 93
read.file.cormatrix, 94
read.file.mydata, 94
read.file.mydata.large, 95
retrieve.data, 106
set.image.parameters, 108
∗ **hplot**
bindingMotifsBiomart_ENSEMBL, 5
ChIPTF_ENCODE, 9
chromatinHMMAll_UCSC, 11
chromatinHMMOne_UCSC, 13
chromHMM_RoadMap, 14
ClinVarCnv_UCSC, 17
ClinVarMain_UCSC, 18
comet, 20
comet.list, 27
comet.web, 28
CoreillCNV_UCSC, 35
COSMIC_UCSC, 36
cpgIslands_UCSC, 37
dgfootprints_RoadMap, 44
DNAse_UCSC, 49
DNaseI_FANTOM, 46
DNaseI_RoadMap, 47
eQTL, 59
eQTL_GTEx, 61
GAD_UCSC, 64
gcContent_UCSC, 65
GeneReviews_UCSC, 67
genes_ENSEMBL, 69
GWAScatalog_UCSC, 70
HiCdata2matrix, 72
HistoneAll_UCSC, 73
HistoneOne_UCSC, 74
imprintedGenes_GTEx, 76
interestGenes_ENSEMBL, 77
interestTranscript_ENSEMBL, 79
ISCA_UCSC, 80
knownGenes_UCSC, 82
metQTL, 83
miRNATargetRegionsBiomart_ENSEMBL, 85
otherRegulatoryRegionsBiomart_ENSEMBL, 87
psiQTL_GTEx, 91
refGenes_UCSC, 96
regulationBiomart_ENSEMBL, 97
regulatoryEvidenceBiomart_ENSEMBL, 98
regulatoryFeaturesBiomart_ENSEMBL, 100
regulatorySegmentsBiomart_ENSEMBL, 102
repeatMasker_UCSC, 105
segmentalDups_UCSC, 107
snpBiomart_ENSEMBL, 109
snpLocations_UCSC, 110
structureBiomart_ENSEMBL, 111
TFBS_FANTOM, 113
transcript_ENSEMBL, 114
xenorefGenes_UCSC, 115
∗ **internal**
check.configVar, 8
check.configVar.cometlist, 8
check.format.mydata, 9
clinCNV, 17
compute.cormatrix, 34
cpgPvalue, 38

create.color.bar, 39
create.color.list, 40
create.color.list.large, 41
create.symbol.list, 41
create.symbol.list.large, 42
create.tracks.user, 43
create.tracks.web, 43
createList.trackUser, 44
draw.legend, 50
draw.name.genes.web, 51
draw.name.tracks.web, 51
draw.plot.annotation, 52
draw.plot.axis.data, 52
draw.plot.comet, 53
draw.plot.comet.nopval, 53
draw.plot.comet.web, 54
draw.plot.cormatrix.plot, 55
draw.plot.grid.mydata, 55
draw.plot.grid.mydata.large, 56
draw.plot.grid.mydata.names, 56
draw.plot.grid.setup, 57
draw.plot.linesconnection, 57
draw.plot.mydata.ggbio, 58
fix.values, 63
fix.values.generic, 63
fix.values.large, 64
printPlot.comet, 90
printPlot.comet.nopval, 90
printPlot.comet.web, 91
read.config, 93
read.file.cormatrix, 94
read.file.mydata, 94
read.file.mydata.large, 95
retrieve.data, 106
set.image.parameters, 108
xenorefGenes_UCSC, 115

∗ **misc**

genesName_ENSEMBL, 68

∗ **package**

coMET-package, 4

∗ **print**

draw.name.genes.web, 51
draw.name.tracks.web, 51
draw.plot.annotation, 52
draw.plot.axis.data, 52
draw.plot.comet, 53
draw.plot.comet.nopval, 53
draw.plot.comet.web, 54
draw.plot.cormatrix.plot, 55
draw.plot.grid.mydata, 55
draw.plot.grid.mydata.large, 56
draw.plot.grid.mydata.names, 56
draw.plot.grid.setup, 57
draw.plot.linesconnection, 57
printPlot.comet.nopval, 90
printPlot.comet.web, 91

allIG *(clinCNV)*, 17
allIGtrack *(clinCNV)*, 17
allimprintedIG *(clinCNV)*, 17
allimprintedIGtrack *(clinCNV)*, 17

BindingMotifAll *(clinCNV)*, 17
BindingMotifMultiple *(clinCNV)*, 17
bindingMotifsBiomart_ENSEMBL, 5
BindingMotifSingle *(clinCNV)*, 17
bindMotifsBiomartTrackAll *(clinCNV)*, 17
bindMotifsBiomartTrackDouble *(clinCNV)*, 17
bindMotifsBiomartTrackMultiple *(clinCNV)*, 17
bindMotifsBiomartTrackSingle *(clinCNV)*, 17

check.configVar, 8
check.configVar.cometlist, 8
check.configVarcometlist *(check.configVar.cometlist)*, 8
check.format.mydata, 9
ChIPTF_ENCODE, 9
chipTFtrack *(clinCNV)*, 17
chromatinHMMAll_UCSC, 11, *13*
chromatinHMMOne_UCSC, *12*, 13
chromatinHMMRoadMapAll *(clinCNV)*, 17
chromatinHMMRoadMapMultiple *(clinCNV)*, 17
chromatinHMMRoadMapSingle *(clinCNV)*, 17
chromHMM_RoadMap, 14
chromHMM_RoadMapAll *(clinCNV)*, 17
chromHMM_RoadMapAllE063 *(clinCNV)*, 17
chromHMM_RoadMapMultiple *(clinCNV)*, 17
chromHMM_RoadMapSingle *(clinCNV)*, 17
chromhmmNoPattern *(clinCNV)*, 17
chromhmmPattern *(clinCNV)*, 17
chromhmmtrackone *(clinCNV)*, 17
chrUCSC2ENSEMBL, 16
clinCNV, 17

ClinVarCnv_UCSC, 17, *19*, *36, 37*, *110–112*
clinVariant *(clinCNV)*, 17
ClinVarMain_UCSC, *18*, 18, *36, 37*, *110–112*
col2HSV, 20, *109*
coMET *(coMET-package)*, 4
comet, 20, *28*, *33*
coMET-package, 4
comet.list, *26*, 27, *33*
comet.web, *26*, *28*, 28
complementary, 33
compute.cormatrix, 34
CoreillCNV_UCSC, *18, 19*, 35, *37*, *110–112*
coreilVariant *(clinCNV)*, 17
COSMIC_UCSC, *18, 19*, *36*, 36, *110–112*
cosmicVariant *(clinCNV)*, 17
cpgIslands_UCSC, 37
cpgIslandtrack *(clinCNV)*, 17
cpgIstrack *(clinCNV)*, 17
cpgPvalue, 38
create.color.bar, 39
create.color.list, 40
create.color.list.large, 41
create.symbol.list, 41
create.symbol.list.large, 42
create.tracks.user, 43
create.tracks.web, 43
createList.trackUser, 44

dgfootprints_RoadMap, 44
dgfootprints_RoadMapSingle *(clinCNV)*, 17
DNAse_UCSC, 49
DNaseI_FANTOM, 46
DNaseI_RoadMap, 47
DNaseI_RoadMapSingle *(clinCNV)*, 17
dnasetrack *(clinCNV)*, 17
draw.legend, 50
draw.name.genes.web, 51
draw.name.tracks.web, 51
draw.plot.annotation, 52
draw.plot.axis.data, 52
draw.plot.comet, 53
draw.plot.comet.nopval, 53
draw.plot.comet.web, 54
draw.plot.cormatrix.plot, 55
draw.plot.grid.mydata, 55
draw.plot.grid.mydata.large, 56
draw.plot.grid.mydata.names, 56
draw.plot.grid.setup, 57
draw.plot.linesconnection, 57
draw.plot.mydata.ggbio, 58
DupTrack *(clinCNV)*, 17
dyaRM *(clinCNV)*, 17
dyaRMtrack *(clinCNV)*, 17

eGTexTrackall *(clinCNV)*, 17
eGTexTrackSNP *(clinCNV)*, 17
enhFANTOMtrack *(clinCNV)*, 17
enhRM *(clinCNV)*, 17
enhRMtrack *(clinCNV)*, 17
eQTL, 59
eQTL_GTEx, 61
eQTLTrackAll *(clinCNV)*, 17
eQTLTrackMultiple *(clinCNV)*, 17
eQTLTrackSingle *(clinCNV)*, 17

fix.values, 63
fix.values.generic, 63
fix.values.large, 64

GAD_UCSC, 64, *67*, *69–71*, *78*, *80–82*, *96*, *114*, *116*
gadtrack *(clinCNV)*, 17
gcContent_UCSC, 65
gctrack *(clinCNV)*, 17
geneNameEnsembl *(clinCNV)*, 17
GeneReviews_UCSC, *65*, 67, *69–71*, *78*, *80–82*, *96*, *114*, *116*
geneRtrack *(clinCNV)*, 17
genes_ENSEMBL, *65*, *67*, *69*, 69, *71*, *80–82*, *96*, *114*, *116*
genesName_ENSEMBL, *65*, *67*, 68, *70, 71*, *78*, *80–82*, *96*, *114*, *116*
genesUcsctrack *(clinCNV)*, 17
genetrack *(clinCNV)*, 17
grtrack *(clinCNV)*, 17
grtrack4eGTex *(clinCNV)*, 17
GWAScatalog_UCSC, *65*, *67*, *69, 70*, 70, *78*, *80–82*, *96*, *114*, *116*
gwastrack *(clinCNV)*, 17

HiCdata2matrix, 72
histonalltrack *(clinCNV)*, 17
HistoneAll_UCSC, 73, *75*
HistoneOne_UCSC, *74*, 74
histoneonetrack *(clinCNV)*, 17

imprintedGenes_GTEx, 76
imprintedGenesGTEx *(clinCNV)*, 17

interestGenes_ENSEMBL, 77
interestgenesENSMBLtrack (clinCNV), 17
interestTranscript_ENSEMBL, 79
interesttransENSMBLtrack (clinCNV), 17
ISCA_UCSC, *65*, *67*, *69–71*, *78*, *80*, 80, *82*, *96*, *114*, *116*
iscatrack (clinCNV), 17

knownGenes_UCSC, *65*, *67*, *69–71*, *78*, *80*, *81*, 82, *96*, *114*, *116*

matrix_HiC_Rao (clinCNV), 17
metQTL, 83
metQTLTrackAll (clinCNV), 17
metQTLTrackMultiple (clinCNV), 17
metQTLTrackSingle (clinCNV), 17
miRNATargetRegionsBiomart_ENSEMBL, 85
miRNATargetRegionsBiomartTrack (clinCNV), 17

opposite (complementary), 33
otherRegulatoryRegionsBiomart_ENSEMBL, 87
OtherRegulatoryRegionsTrackAll (clinCNV), 17
otherRegulatoryRegionsTrackAll (clinCNV), 17
OtherRegulatoryRegionsTrackSingle (clinCNV), 17
otherRegulatoryRegionsTrackSingle (clinCNV), 17

PancreasIG (clinCNV), 17
PancreasIGtrack (clinCNV), 17
PancreasimprintedIG (clinCNV), 17
PancreasimprintedIGtrack (clinCNV), 17
par, *89*
pie, *89*
pizza, 89
polygon, *89*
printPlot.comet, 90
printPlot.comet.nopval, 90
printPlot.comet.web, 91
promRMtrack (clinCNV), 17
promRMtrackE063 (clinCNV), 17
psiGTex (clinCNV), 17
psiGTexTrackall (clinCNV), 17
psiGTexTrackSNP (clinCNV), 17
psiQTL_GTEx, 91

read.config, 93
read.file.cormatrix, 94
read.file.mydata, 94
read.file.mydata.large, 95
refGenes_UCSC, 96
regulationBiomart_ENSEMBL, 97
regulationENSEMBLtrack (clinCNV), 17
regulatoryEvidenceBiomart_ENSEMBL, 98
RegulatoryEvidenceBiomartTrackAll (clinCNV), 17
regulatoryEvidenceBiomartTrackAll (clinCNV), 17
RegulatoryEvidenceBiomartTrackMultiple (clinCNV), 17
regulatoryEvidenceBiomartTrackMultiple (clinCNV), 17
RegulatoryEvidenceBiomartTrackSingle (clinCNV), 17
regulatoryEvidenceBiomartTrackSingle (clinCNV), 17
regulatoryFeaturesBiomart_ENSEMBL, 100
RegulatoryFeaturesBiomartTrackAll (clinCNV), 17
regulatoryFeaturesBiomartTrackAll (clinCNV), 17
RegulatoryFeaturesBiomartTrackMultiple (clinCNV), 17
regulatoryFeaturesBiomartTrackMultiple (clinCNV), 17
RegulatoryFeaturesBiomartTrackSingle (clinCNV), 17
regulatoryFeaturesBiomartTrackSingle (clinCNV), 17
regulatorySegmentsBiomart_ENSEMBL, 102
RegulatorySegmentsBiomartTrackAll (clinCNV), 17
regulatorySegmentsBiomartTrackAll (clinCNV), 17
RegulatorySegmentsBiomartTrackMultiple (clinCNV), 17
regulatorySegmentsBiomartTrackMultiple (clinCNV), 17
RegulatorySegmentsBiomartTrackSingle (clinCNV), 17
regulatorySegmentsBiomartTrackSingle (clinCNV), 17
repeatMasker_UCSC, 105
repeatMaskerTrack (clinCNV), 17

retrieve.data, 106
rmtrack (clinCNV), 17

segmentalDups_UCSC, 107
set.image.parameters, 108
setColors, 108
snpBiomart_ENSEMBL, *18, 19*, *36, 37*, 109, *112*
snpLocations_UCSC, *18, 19*, *36, 37*, *110*, 110, *111, 112*
snptrack (clinCNV), 17
snpUCSCtrack (clinCNV), 17
StomachIG (clinCNV), 17
StomachIGtrack (clinCNV), 17
structureBiomart_ENSEMBL, *18, 19*, *36, 37*, *110, 111*, 111
strutrack (clinCNV), 17

TFBS_FANTOM, 113
tfbsFANTOMtrack (clinCNV), 17
transcript_ENSEMBL, *65*, *67*, *69–71*, *78*, *81, 82*, *96*, 114, *116*
transENSMBLtrack (clinCNV), 17

xenogenestrack (clinCNV), 17
xenorefGenes_UCSC, *65*, *67*, *69–71*, *78*, *80–82*, *96*, *114*, 115