Package: GALLO (via r-universe)

September 9, 2024

Title Genomic Annotation in Livestock for Positional Candidate LOci

Version 1.1

Description The accurate annotation of genes and Quantitative Trait Loci (QTLs) located within candidate markers and/or regions (haplotypes, windows, CNVs, etc) is a crucial step the most common genomic analyses performed in livestock, such as Genome-Wide Association Studies or transcriptomics. The Genomic Annotation in Livestock for positional candidate LOci (GALLO) is an R package designed to provide an intuitive and straightforward environment to annotate positional candidate genes and QTLs from high-throughput genetic studies in livestock. Moreover, GALLO allows the graphical visualization of gene and QTL annotation results, data comparison among different grouping factors (e.g., methods, breeds, tissues, statistical models, studies, etc.), and QTL enrichment in different livestock species including cattle, pigs, sheep, and chicken, among others.

URL <https://github.com/pablobio/GALLO>

Depends R (>= 4.0.0)

biocViews Software

Imports circlize, DT, data.table, doParallel, dplyr, dynamicTreeCut, ggplot2, graphics, grDevices, foreach, lattice , magick, parallel, RColorBrewer, rtracklayer, stats, stringr, unbalhaar, utils

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests BiocStyle, Hmisc, knitr, rmarkdown, testthat

VignetteBuilder knitr

Repository https://pablobio.r-universe.dev

RemoteUrl https://github.com/pablobio/gallo

RemoteRef HEAD

RemoteSha 96f8b8f97f9e0a8d73cc8a6dd51d65735dee09ba

Contents

find_genes_qtls_around_markers 2
import_gff_gtf 3
overlapping_among_groups 4
plot_overlapping 4
plot_qtl_info 5
QTLenrich_plot 6
qtl_enrich 7
relationship_plot 9

Index **11**

`find_genes_qtls_around_markers`

Search genes and QTLs around candidate regions

Description

Takes a list of candidate markers and or regions (haplotypes, CNVs, windows, etc.) and search for genes or QTLs in a determined interval

Usage

```
find_genes_qtls_around_markers(
  db_file,
  marker_file,
  method = c("gene", "qtl"),
  marker = c("snp", "haplotype"),
  interval = 0,
  nThreads = NULL,
  verbose = TRUE
)
```

Arguments

`db_file`	The dataframe obtained using the import_gff_gtf() function
`marker_file`	The file with the SNP or haplotype positions. Detail: For SNP files, the columns "CHR" and "BP" with the chromosome and base pair position, respectively, are mandatory. For the haplotype, the following collumns are mandatory: "CHR", "BP1" and "BP2"
`method`	"gene" or "qtl"
`marker`	"snp" or "haplotype"

interval	The interval in base pair which can be included upstream and downstream from the markers or haplotype coordinates.
nThreads	Number of threads to be used
verbose	Logical value defining if messages should of not be printed during the analysis (default=TRUE)

Value

A dataframe with the genes or QTLs mapped within the specified intervals

Examples

```
data(QTLmarkers)
data(gffQTLs)
out.qtls<-find_genes_qtls_around_markers(db_file=gffQTLs, marker_file=QTLmarkers,
method = "qtl", marker = "snp",
interval = 500000, nThreads = 1)
```

---

import_gff_gtf *Import .gtf and .gff files to be used during gene and QTL annotation, respectively*

Description

Takes a .gft or .gff file and import into a dataframe

Usage

```
import_gff_gtf(db_file, file_type)
```

Arguments

db_file	File with the gene mapping or QTL information. For gene mapping, a .gtf file from Ensembl database must be used. For the QTL search, a .gff file from Animal QTlLdb must be used. Both files must use the same reference annotation used in the original study
file_type	"gtf" or "gff"

Value

A dataframe with the gtf or gtf content

Examples

```
gffpath <- system.file("extdata", "example.gff", package="GALLO")

qtl.inp <- import_gff_gtf(db_file=gffpath,file_type="gff")
```

overlapping_among_groups *Overlapping between grouping factors*

Description

Takes a dataframe with a column of genes, QTLs (or other data) and a grouping column and create some matrices with the overlapping information

Usage

```
overlapping_among_groups(file, x, y)
```

Arguments

file	A dataframe with the data and grouping factor
x	The grouping factor to be compared
y	The data to be compared among the levels of the grouping factor

Value

A list with three matrices: 1) A matrix with the number of overlapping data; 2) A matrix with the percentage of overlapping; 3) A matrix with the combination of the two previous one

Examples

```
data(QTLmarkers)
data(gtfGenes)
genes.out <- find_genes_qtls_around_markers(db_file=gtfGenes,
marker_file=QTLmarkers,method="gene",
marker="snp",interval=100000, nThreads=1)
overlapping.out<-overlapping_among_groups(
file=genes.out,x="Reference",y="gene_id")
```

plot_overlapping *Plot overlapping between data and grouping factors*

Description

Takes the output from overlapping_among_groups function and creates a heatmap with the overlapping between groups

Usage

```
plot_overlapping(overlapping_matrix, nmatrix, ntext, group, labelcex = 1)
```

Arguments

overlapping_matrix	The object obtained in overlapping_amoung_groups function
nmatrix	An interger from 1 to 3 indicating which matrix will be used to plot the overlapping, where: 1) A matrix with the number of overllaping data; 2) A matrix with the percentage of overlapping; 3) A matrix with the combination of the two previous one
ntext	An interger from 1 to 3 indicating which matrix will be used as the text matrix for the heatmap, where: 1) A matrix with the number of overllaping data; 2) A matrix with the percentage of overlapping; 3) A matrix with the combination of the two previous one
group	A vector with the size of groups. This vector will be plotted as row and column names in the heatmap
labelcex	A numeric value indicating the size of the row and column labels

Value

A heatmap with the overlapping between groups

Examples

```
data(QTLmarkers)
data(gtfGenes)
genes.out <- find_genes_qtls_around_markers(
db_file=gtfGenes, marker_file=QTLmarkers,
method="gene", marker="snp",interval=100000,
nThreads=1)

overlapping.out<-overlapping_among_groups(
file=genes.out,x="Reference",y="gene_id")
plot_overlapping(overlapping.out,
nmatrix=2,ntext=2,
group=unique(genes.out$Reference))
```

---

plot_qtl_info *Plot QTLs information from the find_genes_qtls_around_markers output*

Description

Takes the output from find_genes_qtls_around_markers and create plots for the frequency of each QTL type and trait

Usage

```
plot_qtl_info(
  qtl_file,
  qtl_plot = c("qtl_type", "qtl_name"),
  n = "all",
  qtl_class = NULL,
  horiz = FALSE,
  ...
)
```

Arguments

qtl_file	The output from find_genes_qtls_around_markers function
qtl_plot	"qtl_type" or"qtl_name"
n	Number of QTLs to be plotted when the qtl_name option is selected
qtl_class	Class of QTLs to be plotted when the qtl_name option is selected
horiz	The legend of the pie plot for the qtl_type should be plotted vertically or horizontally. The default is FALSE. Therefore, the legend is plotted vertically.
...	Arguments to be passed to/from other methods. For the default method these can include further arguments (such as axes, asp and main) and graphical parameters (see par) which are passed to plot.window(), title() and axis.

Value

A plot with the requested information

Examples

```
data(QTLmarkers)
data(gffQTLs)

out.qtls<-find_genes_qtls_around_markers(db_file=gffQTLs,
marker_file=QTLmarkers, method = "qtl",
marker = "snp", interval = 500000,
nThreads = 1)

plot_qtl_info(out.qtls, qtl_plot = "qtl_type", cex=2)
```

QTLenrich_plot *Plot enrichment results for QTL enrichment analysis*

Description

Takes the output from qtl_enrich function and creates a bubble plot with enrichment results

Usage

```
QTLenrich_plot(qtl_enrich, x, pval)
```

Arguments

qtl_enrich	The output from qtl_enrich function
x	Id column to be used from the qtl_enrich output
pval	P-value to be used in the plot. The name informed to this argument must match the p-value column name in the enrichment table

Value

A plot with the QTL enrichment results

Examples

```
data(QTLmarkers)
data(gffQTLs)
out.qtls<-find_genes_qtls_around_markers(db_file=gffQTLs,
marker_file=QTLmarkers, method = "qtl",
marker = "snp", interval = 500000,
nThreads = 1)

out.enrich<-qtl_enrich(qtl_db=gffQTLs,
qtl_file=out.qtls, qtl_type = "Name",
enrich_type = "genome", chr.subset = NULL, padj = "fdr",nThreads = 1)

out.enrich.filtered<-out.enrich[which(out.enrich$adj.pval<0.05),]
QTLenrich_plot(out.enrich.filtered, x="QTL", pval="adj.pval")
```

---

qtl_enrich *Performs a QTL enrichment analysis based in a Bootstrap simulation for each QTL class*

Description

Takes the output from find_genes_qtls_around_markers and run a QTL enrichment analysis

Usage

```
qtl_enrich(
  qtl_db,
  qtl_file,
  qtl_type = c("QTL_type", "Name"),
  enrich_type = c("genome", "chromosome"),
  chr.subset = NULL,
  nThreads = NULL,
```

```
  padj = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"),
  verbose = TRUE
)
```

Arguments

qtl_db	The object obtained using the import_gff_gtf() function
qtl_file	The output from find_genes_qtls_around_markers function
qtl_type	A character indicating which type of enrichment will be performed. QTL_type indicates that the enrichment processes will be performed for the QTL classes, while Name indicates that the enrichment analysis will be performed for each trait individually
enrich_type	A character indicating if the enrichment analysis will be performed for all the chromosomes ("genome") or for a subset of chromosomes ("chromosome). If the "genome" option is selected, the results reported are the merge of all chromosomes
chr.subset	If enrich_type equal "chromosome", it is possible to define a subset of chromosomes to be analyzed. The default is equal NULL. Therefore, all the chromosomes will be analyzed
nThreads	The number of threads to be used.
padj	The algorithm for multiple testing correction to be adopted ("holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none")
verbose	Logical value defining if messages should of not be printed during the analysis (default=TRUE)

Details

The simple bias of investigation for some traits (such as milk production related traits in the QTL database for cattle) may result in a larger proportion of records in the database. Consequently, the simple investigation of the proportion of each QTL type might not be totally useful. In order to reduce the impact of this bias, a QTL enrichment analysis can be performed. The QTL enrichment analysis performed by GALLO package is based in a hypergeometric test using the number of annoatted QTLs within the candidate regions and the total number of the same QTL in the QTL database.

Value

A data frame with the p-value for the enrichment result

Examples

```
data(QTLmarkers)
data(gffQTLs)
out.qtls<-find_genes_qtls_around_markers(
db_file=gffQTLs,marker_file=QTLmarkers,
method = "qtl",marker = "snp",
interval = 500000, nThreads = 1)
```

```
out.enrich<-qtl_enrich(qtl_db=gffQTLs,
qtl_file=out.qtls, qtl_type = "Name",
enrich_type = "chromosome",chr.subset = NULL,
padj = "fdr",nThreads = 1)
```

`relationship_plot` *Plot relationship between data and grouping factors*

Description

Takes the output from find_genes_qtls_around_markers function and creates a chord plot with the relationship between groups

Usage

```
relationship_plot(
  qtl_file,
  x,
  y,
  grid.col = "gray60",
  degree = 90,
  canvas.xlim = c(-2, 2),
  canvas.ylim = c(-2, 2),
  cex,
  gap
)
```

Arguments

`qtl_file`	The output from find_genes_qtls_around_markers function
`x`	The first grouping factor, to be plotted in the left hand side of the chord plot
`y`	The second grouping factor, to be plotted in the left hand side of the chord plot
`grid.col`	A character with the grid color for the chord plot or a vector with different colors to be used in the grid colors. Note that when a color vector is provided, the length of this vector must be equal the number of sectors in the chord plot
`degree`	A numeric value corresponding to the starting degree from which the circle begins to draw. Note this degree is always reverse-clockwise
`canvas.xlim`	The coordinate for the canvas in the x-axis. By default is c(-1,1)
`canvas.ylim`	The coordinate for the canvas in the y-axis. By default is c(-1,1)
`cex`	The size of the labels to be printed in the plot
`gap`	A numeric value corresponding to the gap between the chord sectors

Value

A chords relating x and y

Examples

```
data(QTLmarkers)
data(gffQTLs)
out.qtls<-find_genes_qtls_around_markers(
db_file=gffQTLs, marker_file=QTLmarkers,
method = "qtl", marker = "snp",
interval = 500000, nThreads = 1)

out.enrich<-qtl_enrich(qtl_db=gffQTLs,
qtl_file=out.qtls, qtl_type = "Name",
enrich_type = "chromosome",
chr.subset = NULL, padj = "fdr",nThreads = 1)

out.enrich$ID<-paste(out.enrich$QTL," - ",
"CHR",out.enrich$CHR,sep="")

out.enrich.filtered<-out.enrich[which(out.enrich$adj.pval<0.05),]

out.qtls$ID<-paste(out.qtls$Name," - ",
"CHR",out.qtls$CHR,sep="")

out.enrich.filtered<-out.enrich.filtered[order(out.enrich.filtered$adj.pval),]

out.qtls.filtered<-out.qtls[which(out.qtls$ID%in%out.enrich.filtered$ID[1:10]),]

out.qtls.filtered[which(out.qtls.filtered$Reference==
"Feugang et al. (2010)"), "color_ref"]<-"purple"

out.qtls.filtered[which(out.qtls.filtered$Reference==
"Buzanskas et al. (2017)"),"color_ref"]<-"pink"

color.grid<-c(rep("black",length(unique(out.qtls.filtered$Abbrev))),
unique(out.qtls.filtered$color_ref))

names(color.grid)<-c(unique(out.qtls.filtered$Abbrev),
unique(out.qtls.filtered$Reference))

relationship_plot(qtl_file=out.qtls.filtered,
x="Abbrev", y="Reference",cex=1,gap=5,
degree = 90, canvas.xlim = c(-5, 5),
canvas.ylim = c(-3, 3), grid.col = color.grid)
```

Index

find_genes_qtls_around_markers, 2

import_gff_gtf, 3

overlapping_among_groups, 4

plot_overlapping, 4
plot_qtl_info, 5

qtl_enrich, 7
QTLenrich_plot, 6

relationship_plot, 9