Package: ctl (via r-universe)

October 13, 2024

Version 1.0.0-10

Date 2024-01-15

Title Correlated Trait Locus Mapping

Author Danny Arends <Danny.Arends@gmail.com>, Yang Li, Gudrun A Brockmann, Ritsert C Jansen, Robert W Williams, and Pjotr Prins

Maintainer Danny Arends <Danny.Arends@gmail.com>

Depends R (>= 2.10), MASS, parallel, qtl

Description Identification and network inference of genetic loci associated with correlation changes in quantitative traits (called correlated trait loci, CTLs). Arends et al. (2016) <doi:10.21105/joss.00087>.

Encoding UTF-8

ByteCompile true

License GPL-3

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-01-16 12:10:02 UTC

Contents

`ctl-package` *CTL - CTL mapping in experimental crosses*

Description

Analysis of experimental crosses to identify genetic markers associated with correlation changes in quantitative traits (CTL). The additional correlation information obtained can be combined with QTL information to perform de novo reconstruction of interaction networks.

For more background information about the method we refer to the methodology article published in XX (201X).

The R package is a basic iomplementation and it includes the following core functionality:

- `CTLscan` - Main function to scan for CTL.
- `CTLsignificant` - Significant interactions from a `CTLscan`.
- `CTLnetwork` - Create a CTL network from a `CTLscan`.
- `image.CTLobject` - Heatmap overview of a CTLscan.
- `plot.CTLscan` - Plot the CTL curve for a single trait.
- `ctl.circle` - Circle plot CTLs on single and multiple traits.
- `ctl.lineplot` - Line plot CTLs on single and multiple traits.
- `CTLprofiles` - Extract CTL interaction profiles.

For all these functions we also provide examples and demonstrations on real genetical genomics data. We thank all contributors for publishing their data online and will accept submissions of intrestion datasets, currently ctl provides:

- `ath.metabolites` - Metabolite expression data from Arabidopsis Thaliana
- `ath.churchill` - Metabolite expression data from Arabidopsis Thaliana
- `yeast.brem` - Gene expression data from Saccharomyces cerevisiae

Details

More detailed information and/or examples are given per function as needed. Some additional functionality:

- `basic.qc` - Some basic quality checks for phenotype and genotype data
- `CTLscan.cross` - Use an R/qtl cross object with `CTLscan`

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>
Contributions from: Bruno Tesson, Pjotr Prins and Ritsert C. Jansen

References

- TODO

See Also

- `CTLscan` - Scan for CTL
- `CTLscan.cross` - Use an R/qtl cross object with `CTLscan`

`ath.churchill` *Example metabolite expression data from Arabidopsis Thaliana on 9 metabolites.*

Description

Arabidopsis recombinant inbred lines by selfing. There are 403 lines, 9 phenotypes, and 69 markers on 5 chromosomes stored as a list with 3 matrices: genotypes, phenotypes, map

Usage

```
data(ath.churchill)
```

Format

Data stored in a list holding 3 matrices genotypes, phenotypes and map

Details

Arabidopsis recombinant inbred lines by selfing. There are 403 lines, 9 metabolic phenotypes, and 69 markers on 5 chromosomes.

Source

Arabidopsis Bay-0 x Sha metabolite data from XX, senior author: Gary Churchill 2012, Published in: Plos

References

TODO

Examples

```
library(ctl)
data(ath.churchill)           # Arabidopsis thaliana dataset

ath.gary$genotypes[1:5, 1:5]  # ath.gary is the short name
ath.gary$phenotypes[1:5, 1:5]
ath.gary$map[1:5, ]
```

---

ath.metabolites *Example metabolite expression data from Arabidopsis Thaliana on 24 metabolites.*

Description

Arabidopsis recombinant inbred lines by selfing. There are 162 lines, 24 phenotypes, and 117 markers on 5 chromosomes stored as a list with 3 matrices: genotypes, phenotypes, map

Usage

```
data(ath.metabolites)
```

Format

Data stored in a list holding 3 matrices genotypes, phenotypes and map

Details

Arabidopsis recombinant inbred lines by selfing. There are 162 lines, 24 phenotypes, and 117 markers on 5 chromosomes.

Source

Part of the Arabidopsis RIL selfing experiment with Landsberg Erecta (Ler) and Cape Verde Islands (Cvi) with 162 individuals scored (with errors) at 117 markers. Dataset obtained from GBIC - Groningen BioInformatics Centre, University of Groningen.

References

- Keurentjes, J. J. and Fu, J. and de Vos, C. H. and Lommen, A. and Hall, R. D. and Bino, R. J. and van der Plas, L. H. and Jansen, R. C. and Vreugdenhil, D. and Koornneef, M. (2006), The genetics of plant metabolism. *Nature Genetics.* **38**-7, 842–849.
- Alonso-Blanco, C. and Peeters, A. J. and Koornneef, M. and Lister, C. and Dean, C. and van den Bosch, N. and Pot, J. and Kuiper, M. T. (1998), Development of an AFLP based linkage map of Ler, Col and Cvi Arabidopsis thaliana ecotypes and construction of a Ler/Cvi recombinant inbred line population. *Plant J.* **14**(2), 259–271.

Examples

```
library(ctl)
data(ath.metabolites)           # Arabidopsis thaliana dataset

ath.metab$genotypes[1:5, 1:5]   # ath.metab is the short name
ath.metab$phenotypes[1:5, 1:5]
ath.metab$map[1:5, ]
```

---

ath.result *Output of QCLscan after 5000 permutations on the metabolite expression data from Arabidopsis Thaliana.*

Description

Results from a QCLscan on Arabidopsis recombinant inbred lines by selfing. There are 162 lines, 24 phenotypes, and 117 markers on 5 chromosomes stored as a list with 3 matrices: genotypes, phenotypes, map

Usage

```
data(ath.result)
```

Format

Cross object from R/QTL

Details

Results from a QCLscan on Arabidopsis recombinant inbred lines by selfing. There are 162 lines, 24 phenotypes, and 117 markers on 5 chromosomes. the QCLscan also includes 5000 permutations

Source

Part of the Arabidopsis RIL selfing experiment with Landsberg Erecta (Ler) and Cape Verde Islands (Cvi) with 162 individuals scored (with errors) at 117 markers. Dataset obtained from GBIC - Groningen BioInformatics Centre, University of Groningen.

References

- Keurentjes, J. J. and Fu, J. and de Vos, C. H. and Lommen, A. and Hall, R. D. and Bino, R. J. and van der Plas, L. H. and Jansen, R. C. and Vreugdenhil, D. and Koornneef, M. (2006), The genetics of plant metabolism. *Nature Genetics.* **38**-7, 842–849.
- Alonso-Blanco, C. and Peeters, A. J. and Koornneef, M. and Lister, C. and Dean, C. and van den Bosch, N. and Pot, J. and Kuiper, M. T. (1998), Development of an AFLP based linkage map of Ler, Col and Cvi Arabidopsis thaliana ecotypes and construction of a Ler/Cvi recombinant inbred line population. *Plant J.* **14**(2), 259–271.

Examples

```
data(ath.result)          # Arabidopsis thaliana dataset
ath.result[[1]]           # Print the QCLscan summary of the phenotype 1
```

`basic.qc` *Create quality control plots.*

Description

Create quality control plots, used in the examples of CTL mapping.

Usage

```
basic.qc(genotypes, phenotypes, map_info)
```

Arguments

`genotypes`	Matrix of genotypes. (individuals x markers)
`phenotypes`	Matrix of phenotypes. (individuals x phenotypes)
`map_info`	Matrix of genetic map information

Details

None.

Value

None.

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Scan for CTL
- plot.CTLscan - Plot a CTLscan object

Examples

```
#TODO
```

`ctl.circle` *Circleplot CTL on multiple traits*

Description

Plot the CTL for genome-wide CTL on multiple traits (the output of CTLscan).

Usage

```
ctl.circle(CTLobject, mapinfo, phenocol, significance = 0.05, gap = 50, cex = 1,
verbose = FALSE)
```

Arguments

`CTLobject`	An object of class `"CTLobject"`, as output by CTLscan.
`mapinfo`	The mapinfo matrix with 3 columns: "Chr" - the chromosome number, "cM" - the location of the marker in centiMorgans and the 3rd column "Mbp" - The location of the marker in Mega basepairs. If supplied the marker names (row-names) should match those in the CTLobject.
`phenocol`	Which phenotype results to plot. Defaults to plot all phenotypes.
`significance`	Significance threshold to set a genome wide False Discovery Rate (FDR).
`gap`	Gap between chromosomes in cM.
`cex`	Global magnificantion factor for the image elements.
`verbose`	Be verbose.

Details

None.

Value

None.

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

See Also

- CTLscan - Scan for CTL
- CTLprofiles - Extract CTL interaction profiles
- print.CTLscan - Print a summary of a CTLscan
- par - Plot parameters
- colors - Colors used in plotting

Examples

```
library(ctl)
data(ath.result)       # Arabidopsis Thaliana results
data(ath.metabolites)  # Arabidopsis Thaliana data set

ctl.circle(ath.result, ath.metab$map, sign=0.001)
ctl.circle(ath.result, ath.metab$map, phenocol = 1:6, sign = 0.01)
```

---

ctl.lineplot *Lineplot CTL on multiple traits*

Description

Plot the CTL for genome-wide CTL on multiple traits (the output of CTLscan).

Usage

```
ctl.lineplot(CTLobject, mapinfo, phenocol, significance = 0.05, gap = 50,
col = "orange", bg.col = "lightgray", cex = 1, verbose = FALSE)
```

Arguments

CTLobject	An object of class "CTLobject", as output by CTLscan.
mapinfo	The mapinfo matrix with 3 columns: "Chr" - the chromosome number, "cM" - the location of the marker in centiMorgans and the 3rd column "Mbp" - The location of the marker in Mega basepairs. If supplied the marker names (row-names) should match those in the CTLobject (only significant markers will be annotated).
phenocol	Which phenotype results to plot. Defaults to plot all phenotypes.
significance	Significance threshold to set a genome wide False Discovery Rate (FDR).
gap	The gap between chromosomes in cM.
col	Line color used.
bg.col	Node background color.
cex	Global magnificantion factor for the image elements.
verbose	Be verbose.

Details

None.

Value

None.

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

See Also

- CTLscan - Scan for CTL
- CTLprofiles - Extract CTL interaction profiles
- print.CTLscan - Print a summary of a CTLscan
- par - Plot parameters
- colors - Colors used in plotting

Examples

```
require(ctl)
data(ath.result)       # Arabidopsis Thaliana results
data(ath.metabolites)  # Arabidopsis Thaliana data set

todo <- c(1,3,4,5,6,8,9,10,11,12,14,17,18,19,22,23)
op   <- par(mfrow = c(4,4))
op   <- par(oma = c(0.1,0.1,0.1,0.1))
op   <- par(mai = c(0.1,0.1,0.1,0.1))
for(x in todo){ # Overview of the 16 traits with CTLs
  ctl.lineplot(ath.result, ath.metab$map, phenocol = x, sign=0.1)
}
```

---

ctl.load *ctl.load - Load CTLs calculated by the D2.0 version*

Description

Load CTLs calculated by the D2.0 version

Usage

```
ctl.load(genotypes = "ngenotypes.txt", phenotypes = "nphenotypes.txt",
output = "ctlout", from=1, to, verbose = FALSE)
```

Arguments

genotypes	Original datafile containing the genotypes scanned.
phenotypes	Original datafile containing the phenotypes scanned.
output	Directory containing the output files.
from	Start loading at which phenotype.
to	Continue loading untill this phenotype.
verbose	Be verbose.

Details

TODO

Value

CTLobject, a list with at each index a CTLscan object:

- $ctls - Matrix of differential correlation scores for each trait at each marker
- $qtl - Vector of QTL lodscores for each marker (if a QTL scan was perfomed -qtl)
- $p - Vector of maximum scores per marker obtained during permutations
- $l - Matrix of LOD scores for CTL likelihood

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

Examples

```
library(ctl)        # Load CTL library
```

CTLhelper *CTLhelper - Helper functions for CTL mapping*

Description

Helper functions for Correlated Trait Locus (CTL) mapping

Usage

```
ctl.names(CTLobject)
ctl.qtlmatrix(CTLobject)

ctl.name(CTLscan)
ctl.ctlmatrix(CTLscan)
ctl.dcormatrix(CTLscan)
ctl.qtlprofile(CTLscan)
```

Arguments

CTLobject	An object of class "CTLobject", as output by CTLscan.
CTLscan	An object of class "CTLscan". This is a single element from an "CTLobject", as output by CTLscan.

Details

TODO

Value

TODO

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

Examples

```
#TODO
```

CTLmapping *CTLmapping - Scan for correlated trait loci (CTL)*

Description

Scan for correlated trait loci (CTL)

Usage

```
CTLmapping(genotypes, phenotypes, phenocol = 1, nperm = 100, nthreads = 1,
strategy = c("Exact", "Full", "Pairwise"), adjust = TRUE, qtl = TRUE, verbose = FALSE)
```

Arguments

genotypes	Matrix of genotypes. (individuals x markers)
phenotypes	Matrix of phenotypes. (individuals x phenotypes)
phenocol	Which phenotype column(s) should we analyse. Default: Analyse a single phenotype.
nperm	Number of permutations to perform. This parameter is not used when method="Exact".
nthreads	Number of CPU cores to use during the analysis.
strategy	The permutation strategy to use, either • Exact: Uses exact calculations to calculate the likelihood of a difference in correlation: Cor(AA) - Cor(BB). Using a Bonferroni correction. • Full: Most powerful analysis method - Compensate for marker and trait correlation structure (Breitling et al.). • Pairwise: Suitable when we have a lot of markers and only a few traits (< 50) (human GWAS)- Compensates only for marker correlation structure. Note: Exact is the default and fastest option it uses a normal distribution for estimating p-values and uses bonferoni correction. It has however the least power to detect CTLs, the two other methods (Full and Pairwise) perform permutations to assign significance.
adjust	Adjust p-values for multiple testing (only used when strategy = Exact).
qtl	Use the internal slow QTL mapping method to map QTLs.
verbose	Be verbose.

Details

TODO

- NOTE: Main bottleneck of the algorithm is the RAM available to the system

Value

CTLscan, a list of:

- $dcor - Matrix of differential correlation scores for each trait at each marker
- $perms - Vector of maximums per marker obtained during permutations
- $ctls - Matrix of LOD scores for CTL likelihood

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Main function to scan for CTL
- CTLscan.cross - Use an R/qtl cross object with CTLscan
- CTLsignificant - Significant interactions from a CTLscan
- plot.CTLscan - Plot the CTL curve for a single trait

Examples

```
library(ctl)
data(ath.metabolites) # Arabidopsis Thaliana dataset
singlescan <- CTLmapping(ath.metab$genotypes, ath.metab$phenotypes, phenocol = 23)

plot(singlescan)      # Plot the results of the CTL scan for the phenotype

summary <- CTLsignificant(singlescan)
summary               # Get a list of significant CTLs
```

CTLnetwork *CTLnetwork - Interaction network from a genome-wide CTLscan of multiple traits*

Description

Create a file containing the interaction network from a genome-wide CTLscan of multiple traits.

Usage

```
CTLnetwork(CTLobject, mapinfo, significance = 0.05, LODdrop = 2,
what = c("names", "ids"), short = FALSE, add.qtls = FALSE, file = "", verbose = TRUE)
```

Arguments

CTLobject	An object of class "CTLobject", as output by CTLscan.
mapinfo	The mapinfo matrix with 3 columns: "Chr" - the chromosome number, "cM" - the location of the marker in centiMorgans and the 3rd column "Mbp" - The location of the marker in Mega basepairs. If supplied the marker names (row-names) should match those in the CTLobject (only significant markers will be annotated).
significance	Significance threshold for a genome wide false discovery rate (FDR).
LODdrop	Drop in LOD score needed before we assign an edge type.
what	Return trait and marker names or column numbers (for indexing).
short	Edges are markers when TRUE, otherwise markers are nodes (default).
add.qtls	Should marker QTL trait interactions be added to the generated sif network file, QTLs are included when they are above -log10(significance/n.markers).
file	A connection, or a character string naming the file to print to. If "" (the default), CTLnetwork prints to the standard output connection, the console unless redirected by sink.
verbose	Be verbose.

Details

Outputs a sif network file, and a node attribute file:

- ctlnet<FILE>.sif - Shows CTL connections from Trait to Marker with edge descriptions
- ctlnet<FILE>.nodes - Attributes of the nodes (Traits and Genetic markers) nodes to this file can be used to either color chromosomes, or add chromosome locations.

Value

A matrix with significant CTL interactions and information in 5 Columns:

- TRAIT1 - Trait ID of the origin trait
- MARKER - Marker ID at which the CTL was found
- TRAIT2 - Trait ID of the target trait
- LOD_C - LOD score of the CTL interaction
- CAUSAL - Type of edge determined by QTL LOD-drop:
  - NA - CTL/QTL for TRAIT1 and/or TRAIT2 not found
  - -1 - TRAIT1 is DOWNSTREAM of TRAIT2
  - 0 - UNDETERMINED Edge
  - 1 - TRAIT1 is UPSTREAM of TRAIT2
- LOD_T1 - QTL LOD-score of TRAIT1 at MARKER
- LOD_T2 - QTL LOD-score of TRAIT2 at MARKER

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

Examples

```
library(ctl)
data(ath.result)       # Arabidopsis Thaliana results
data(ath.metabolites)  # Arabidopsis Thaliana data set

ctls <- CTLnetwork(ath.result, significance = 0.1)
op <- par(mfrow = c(2,1))
plot(ctls)
ctl.lineplot(ath.result, ath.metab$map, significance=0.1)
```

CTLprofiles *CTLprofiles - Extract CTL interaction profiles*

Description

Extract the CTL interaction profiles: phenotype x marker (p2m matrix) and phenotype x phenotype (p2p matrix) from a CTLscan.

Usage

```
CTLprofiles(CTLobject, against = c("markers","phenotypes"), significance = 0.05,
verbose=FALSE)
```

Arguments

CTLobject	An object of class "CTLobject", as output by CTLscan.
against	Plot the CTL against either: markers or phenotypes.
significance	Significance threshold to set a genome wide False Discovery Rate (FDR).
verbose	Be verbose.

Details

These matrices can be combined with QTL information to perform de novo reconstruction of interaction networks.

The 'against' parameter is by default set to "markers" which returns a phenotype x markers matrix (p2m matrix), which should be comparible to the QTL profiles of the traits.

When the 'against' parameter is set to "phenotypes" a phenotype x phenotype matrix (p2p matrix) is returned, showing the interactions between the phenotypes.

Value

Matrix: phenotypes x marker or phenotypes x phenotypes

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

Examples

```
library(ctl)         # Load CTL library
data(ath.result)     # Arabidopsis Thaliana results
p2m_matrix <- CTLprofiles(ath.result, against="markers")
p2p_matrix <- CTLprofiles(ath.result, against="phenotypes")
```

`CTLregions` *CTLregions - Get all significant interactions from a genome-wide CTLscan*

Description

Get all significant interactions from a genome-wide CTLscan.

Usage

```
CTLregions(CTLobject, mapinfo, phenocol = 1, significance = 0.05, verbose = TRUE)
```

Arguments

`CTLobject`	An object of class `"CTLobject"`, as output by `CTLscan`.
`mapinfo`	The mapinfo matrix with 3 columns: "Chr" - the chromosome number, "cM" - the location of the marker in centiMorgans and the 3rd column "Mbp" - The location of the marker in Mega basepairs. If supplied the marker names (row-names) should match those in the CTLobject.
`phenocol`	Which phenotype column should we analyse.
`significance`	Significance threshold to set a genome wide False Discovery Rate (FDR).
`verbose`	Be verbose.

Details

TODO

Value

A matrix significant CTL interactions with 4 columns: trait, marker, trait, lod

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

Examples

```
library(ctl)

data(ath.metabolites)                # Arabidopsis Thaliana data set
data(ath.result)                     # Arabidopsis Thaliana CTL results
regions <- CTLregions(ath.result, ath.metab$map)
```

CTLscan *CTLscan - Scan for Correlated Trait Locus (CTL)*

Description

Scan for Correlated Trait Locus (CTL) in populations

Usage

```
CTLscan(genotypes, phenotypes, phenocol, nperm=100, nthreads = 1,
strategy = c("Exact", "Full", "Pairwise"),
parametric = FALSE, adjust=TRUE, qtl = TRUE, verbose = FALSE)
```

Arguments

genotypes	Matrix of genotypes. (individuals x markers)
phenotypes	Matrix of phenotypes. (individuals x phenotypes)
phenocol	Which phenotype column(s) should we analyse. Default: Analyse all phenotypes.
nperm	Number of permutations to perform. This parameter is not used when method="Exact".
nthreads	Number of CPU cores to use during the analysis.
strategy	The permutation strategy to use, either • Exact - Uses exact calculations to calculate the likelihood of a difference in correlation: Cor(AA) - Cor(BB). Using a Bonferroni correction. • Full - Most powerful analysis method - Compensate for marker and trait correlation structure (Breitling et al.). • Pairwise - Suitable when we have a lot of markers and only a few traits (< 50) (human GWAS)- Compensates only for marker correlation structure. Note: Exact is the default and fastest option it uses a normal distribution for estimating p-values and uses bonferoni correction. It has however the least power to detect CTLs, the two other methods (Full and Pairwise) perform permutations to assign significance.

parametric	Use non-parametric testing (Spearman) or parametric testing (Pearson). The DEFAULT is to use non-parametric tests which are less sensitive to outliers in the phenotype data.
adjust	Adjust p-values for multiple testing (only used when strategy = Exact).
qtl	Use the internal slow QTL mapping method to map QTLs.
verbose	Be verbose.

Details

By default the algorithm will not do QTL mapping, the qtl component of the output is an vector of 0 scores for LOD. This is to remove some computational burden, please use the have.qtls parameter to provide QTL data. Some computational bottleneck of the algorithm are:

- RAM available to the system with large number of markers (100K+) and/or phenotypes (100K+).
- Computational time with large sample sizes (5000+) and/or huge amount of phenotype data (100K+).
- Very very huge amounts of genotype markers (1M+)

Some way of avoiding these problems are: CTL mapping using only a single chromosome at a time and / or selecting a smaller subsets of phenotype data for analysis.

Value

CTLobject, a list with at each index (i) an CTLscan object:

- $dcor - Matrix of Z scores (method=Exact), or Power/Adjacency Z scores or for each trait at each marker (n.markers x n.phenotypes)
- $perms - Vector of maximum scores obtained during permutations (n.perms)
- $ctl - Matrix of LOD scores for CTL likelihood of phenotype i (n.markers x n.phenotypes)
- $qtl - Vector of LOD scores for QTL likelihood of phenotype i (n.markers)

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan.cross - Use an R/qtl cross object with CTLscan
- CTLregions - Regions with significant CTLs from a CTLscan
- CTLsignificant - Significant interactions from a CTLscan
- CTLnetwork - Create a CTL network from a CTLscan
- image.CTLobject - Heatmap overview of a CTLscan
- plot.CTLscan - Plot the CTL curve for a single trait

Examples

```
library(ctl)
data(ath.metabolites)                # Arabidopsis Thaliana data set

ctlscan <- CTLscan(ath.metab$genotypes, ath.metab$phenotypes, phenocol=1:4)
ctlscan

# Genetic regions with significant CTLs found for the first phenotype
CTLregions(ctlscan, ath.metab$map, phenocol = 1)

summary <- CTLsignificant(ctlscan)    # Matrix of Trait, Marker, Trait interactions
summary                               # Get a list of significant CTLs

nodes <- ctl.lineplot(ctlscan, ath.metab$map)  # Line plot the phenotypes
nodes
```

`CTLscan.cross` *CTLscan.cross - Scan for Correlated Trait Locus (CTL) (R/qtl cross object)*

Description

Scan for Correlated Trait Locus (CTL) in populations (using an R/qtl cross object)

Usage

```
CTLscan.cross(cross, ...)
```

Arguments

`cross`	An object of class `cross`. See `read.cross` for details.
`...`	Passed to `CTLscan` function:

- phenocol - Which phenotype column should we analyse.
- method - We provide 3 ways of mapping correlation differences across the genome:
  - Exact: Uses a Correlation to Z score transformation to calculate the likelihood of a difference in correlation: Cor(AA) - Cor(BB)
  - Power: More powerful analysis method using the squared difference in correlation: (Cor(AA) - Cor(BB))^2
  - Adjacency: Adjacency method which using the squared difference in squared correlation, but keeping the sign of correlation: (sign*Cor(AA)^2 - sign*Cor(BB)^2)^2

Note: Exact is the default and fastest option it uses a normal distribution for estimating p-values and uses bonferoni correction. It has however the least power to detect CTLs, the two other methods (Power and Adjacency) perform permutations to assign significance.

- n.perm - Number of permutations to perform.
- strategy - The permutation strategy to use, either Full (Compensate for marker and trait correlation structure) or Pairwise (Compensate for marker correlation structure). This parameter is not used when method="Exact".
- conditions - A vector of experimental conditions applied during the experiment. These conditions will be used as covariates in the QTL modeling step.
- n.cores - Number of CPU cores to use during the analysis.
- verbose - Be verbose.

Details

TODO

- NOTE: Main bottleneck of the algorithm is the RAM available to the system

Value

CTLscan object, a list with at each index a CTL matrix (Rows: Phenotypes, Columns: Genetic markers) for the phenotype.

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Main function to scan for CTL
- CTLsignificant - Significant interactions from a CTLscan
- CTLnetwork - Create a CTL network from a CTLscan
- image.CTLobject - Heatmap overview of a CTLscan
- plot.CTLscan - Plot the CTL curve for a single trait

Examples

```
library(ctl)
data(multitrait)      # Arabidopsis Thaliana (R/qtl cross object)

mtrait <- calc.genoprob(multitrait)          # Calculate genotype probabilities
qtls   <- scanone(mtrait, pheno.col = 1)     # Scan for QTLS using R/qtl
```

```
ctls   <- CTLscan.cross(mtrait, phenocol = 1, qtl = FALSE)
ctls[[1]]$qtl <- qtls[,3]

ctl.lineplot(ctls, qtls[,1:2], significance = 0.05) # Line plot all the phenotypes

summary <- CTLsignificant(ctls)              # Get a list of significant CTLs
summary
```

`CTLsignificant` *CTLsignificant - Get all significant interactions from a genome-wide CTLscan*

Description

Get all significant interactions from a genome-wide CTLscan.

Usage

```
CTLsignificant(CTLobject, significance = 0.05, what = c("names","ids"))
```

Arguments

`CTLobject`	An object of class `"CTLobject"`, as output by `CTLscan`.
`significance`	Significance threshold to set a genome wide False Discovery Rate (FDR).
`what`	Return trait and marker names or column numbers (for indexing).

Details

TODO

Value

A matrix significant CTL interactions with 4 columns: trait, marker, trait, lod

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

Examples

```
library(ctl)
data(ath.result)
all_interactions <- CTLsignificant(ath.result)
all_interactions[1:10, ]
trait1_interactions <- CTLsignificant(ath.result[[1]])
trait1_interactions
```

`detect.peaks` *detect.peaks - Peak detection algorithm to 'flatten' data above a certain threshold*

Description

Peak detection algorithm to 'flatten' data above a certain threshold

Usage

```
detect.peaks(data, chrEdges = c(1), threshold = 4, verbose = FALSE)
```

Arguments

`data`	A vector of scores per marker/locus.
`chrEdges`	Start positions of the chromosomes.
`threshold`	Threshold to determine regions.
`verbose`	Be verbose.

Details

TODO

Value

TODO

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Main function to scan for CTL
- CTLsignificant - Significant interactions from a CTLscan
- CTLnetwork - Create a CTL network from a CTLscan
- image.CTLobject - Heatmap overview of a CTLscan
- plot.CTLscan - Plot the CTL curve for a single trait

Examples

```
#TODO
```

---

hist.CTLobject *Plot histogram of CTL permutations*

Description

Plot histogram of CTL permutations (the output of CTLscan).

Usage

```
## S3 method for class 'CTLobject'
hist(x, phenocol=1, ...)
```

Arguments

x	An object of class "CTLscan", as output by CTLscan.
phenocol	Which phenotype column(s) should we analyse. Defaults to analyse all phenotype columns
...	Passed to the function image when it is called.

Details

None.

Value

For a detailed description, see CTLprofiles

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

See Also

- CTLscan - Scan for CTL
- CTLprofiles - Extract CTL interaction profiles
- print.CTLscan - Print a summary of a CTLscan
- par - Plot parameters
- colors - Colors used in plotting

Examples

```
library(ctl)                      # Load CTL library
data(ath.result)
hist(ath.result, phenocol = 1:3)  # Compare the results of the first 3 scans
```

---

image.CTLobject *Plot genome-wide CTL on multiple traits*

Description

Plot the CTL for genome-wide CTL on multiple traits (the output of CTLscan).

Usage

```
## S3 method for class 'CTLobject'
image(x, marker_info, against = c("markers","phenotypes"), significance = 0.05,
col=whiteblack, do.grid=TRUE, grid.col = "white", verbose = FALSE, add=FALSE,
breaks = c(0, 1, 2, 3, 10, 10000), ...)
```

Arguments

x	An object of class "CTLscan", as output by CTLscan.
marker_info	Information used to plot chromosome lines.
against	Plot which interaction matrice, options are: markers: the phenotype*marker or phenotypes: the phenotype*phenotypes matrix.
significance	Significance threshold to set a genome wide False Discovery Rate (FDR).
col	Color-range used in plotting.
do.grid	When TRUE, grid lines are added to the plot.
grid.col	Color used for the grid lines, only used when do.grid = TRUE.
verbose	Be verbose.
add	Add this plot to a previously opened plot window.
breaks	See par.
...	Passed to the function image when it is called.

Details

None.

Value

For a detailed description, see CTLprofiles

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

See Also

- CTLscan - Scan for CTL
- CTLprofiles - Extract CTL interaction profiles
- print.CTLscan - Print a summary of a CTLscan
- par - Plot parameters
- colors - Colors used in plotting

Examples

```
library(ctl)
data(ath.result)       # Arabidopsis Thaliana results
#Phenotype to phenotype matrix
p2p_matrix <- image(ath.result, against="phenotypes")
#Phenotype to marker matrix
p2m_matrix <- image(ath.result, against="markers")
```

---

plot.CTLobject *Plot CTL curves or heatmaps*

Description

Plot the CTL curve or heatmaps for a genome scan (the output of CTLscan).

Usage

```
## S3 method for class 'CTLobject'
plot(x, phenocol = 1:length(x), ...)
```

Arguments

x	An object of class "CTLobject", as output by CTLscan.
phenocol	Which phenotype column(s) should we plot. Defaults to creating an image of all phenotype columns
...	Passed to the function image.CTLobject plot.CTLscan when it is called.

Details

None.

Value

None.

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

See Also

- CTLscan - Scan for CTL
- print.CTLscan - Print a summary of a CTLscan
- par - Plot parameters
- colors - Colors used in plotting

Examples

```
library(ctl)
data(ath.result) # Arabidopsis Thaliana dataset
plot(ath.result)
```

---

plot.CTLpermute *Differential correlation versus likelihood plotted in curves*

Description

Differential correlation versus likelihood plot curves.

Usage

```
## S3 method for class 'CTLpermute'
plot(x, type="s", ...)
```

Arguments

x	An object of class "CTLscan".
type	What type of plot should be drawn. for possible options see plot.
...	Passed to the function plot when it is called.

Details

None.

Value

None.

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

See Also

- CTLscan - Scan for CTL
- print.CTLscan - Print a summary of a CTLscan
- par - Plot parameters
- colors - Colors used in plotting

Examples

```
library(ctl)
data(ath.result) # Arabidopsis Thaliana dataset
plot(ath.result[[1]]$perms)
```

---

plot.CTLscan *Plot CTL results as bar, line or GWAS plot.*

Description

Plot the CTL results for a genome scan (the output of CTLscan) as a barplot, curved line or GWAS plot.

Usage

```
## S3 method for class 'CTLscan'
plot(x, mapinfo = NULL, type = c("barplot","gwas","line"),
onlySignificant = TRUE, significance = 0.05, gap = 25, plot.cutoff = FALSE,
do.legend=TRUE, legend.pos = "topleft", cex.legend=1.0, ydim=NULL,
ylab="-log10(P-value)", ...)
```

Arguments

x	An object of class "CTLscan", as output by CTLscan.
mapinfo	The mapinfo matrix with 3 columns: "Chr" - the chromosome number, "cM" - the location of the marker in centiMorgans and the 3rd column "Mbp" - The location of the marker in Mega basepairs. If supplied the marker names (row-names) should match those in the CTLobject.
type	Type of plot: Summed barplot, GWAS style plot or a basic line plot.

onlySignificant	Plot only the significant contributions to the CTL profile.
significance	Significance threshold for setting a genomewide FDR.
gap	Gap in Cm between chromosomes.
plot.cutoff	Adds a line at -log10(significance) and adds a legend showing the significance level.
do.legend	Adds a legend showing which phenotypes contribute to the CTL profile.
legend.pos	Position of the legend in the plot window.
cex.legend	Maginification of the text in the legend.
ydim	Dimension of the y-axis, if NULL then it will be calculated.
ylab	Label for the y-axis.
...	Passed to the function plot when it is called.

Details

None.

Value

None.

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

See Also

- CTLscan - Scan for CTL
- print.CTLscan - Print a summary of a CTLscan
- par - Plot parameters
- colors - Colors used in plotting

Examples

```
library(ctl)
data(ath.result)       # Arabidopsis thaliana results
data(ath.metabolites)  # Arabidopsis thaliana data (phenotypes, genotypes and mapinfo

plot(ath.result[[3]])
plot(ath.result[[2]], mapinfo = ath.metab[[3]])
plot(ath.result[[1]], mapinfo = ath.metab[[3]])
plot(ath.result[[3]], mapinfo = ath.metab[[3]])
plot(ath.result[[3]], mapinfo = ath.metab[[3]], type="gwas")
plot(ath.result[[3]], mapinfo = ath.metab[[3]], type="line")
```

plotTraits *plotTraits - Trait vs Trait scatterplot, colored by the selected genetic locus*

Description

Trait vs Trait scatterplot, colored by the selected genetic locus

Usage

```
plotTraits(genotypes, phenotypes, phenocol = c(1, 2), marker = 1, doRank = FALSE)
```

Arguments

genotypes	Matrix of genotypes. (individuals x markers)
phenotypes	Matrix of phenotypes. (individuals x phenotypes)
phenocol	Which phenotype column(s) should be plotted against each other, Default: phenotype 1 versus 2
marker	Which marker (column in genotypes) should be used to add genotype as a color of the dots.
doRank	Transform quantitative data into ranked data before analyzing the slope.

Details

TODO

Value

TODO

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Main function to scan for CTL
- CTLsignificant - Significant interactions from a CTLscan
- CTLnetwork - Create a CTL network from a CTLscan
- image.CTLobject - Heatmap overview of a CTLscan
- plot.CTLscan - Plot the CTL curve for a single trait

Examples

```
library(ctl)
data(ath.metabolites)                # Arabidopsis Thaliana data set

plotTraits(ath.metab$genotypes, ath.metab$phenotypes, marker=75, doRank = TRUE)
```

---

print.CTLobject *Print the results of a CTL genome scan*

Description

Print the results of a multiple phenotype CTL genome scan produced by CTLscan.

Usage

```
  ## S3 method for class 'CTLobject'
print(x, ...)
```

Arguments

x	An object of class "CTLobject", as output by CTLscan.
...	Ignored.

Details

None.

Value

None.

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Scan for CTL
- plot.CTLscan - Plot a CTLscan object

Examples

```
#TODO
```

---

print.CTLscan *Print the results of a single phenotype CTL scan*

Description

Print the results of a single phenotype CTL scan produced by either CTLmapping (Single phenotype scan) or CTLscan (Multi phenotype scan).

Usage

```
## S3 method for class 'CTLscan'
print(x, ...)
```

Arguments

x	An object of class "CTLscan". This is a single element from an "CTLobject", as output by CTLscan.
...	Ignored.

Details

None.

Value

None.

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Scan for CTL
- plot.CTLscan - Plot a CTLscan object

Examples

```
#TODO
```

`qtlimage` *Plot a QTL heatmap of the phenotypes scanned by CTLscan*

Description

Plots the QTL heatmap of a genome wide QTL scan (part of the output of `CTLscan`).

Usage

```
qtlimage(x, marker_info, do.grid=TRUE, grid.col="white", verbose=FALSE, ...)
```

Arguments

`x`	An object of class `"CTLobject"`, as output by `CTLscan`.
`marker_info`	Information used to plot chromosome lines.
`do.grid`	When TRUE, grid lines are added to the plot.
`grid.col`	Color used for the grid lines, only used when do.grid = TRUE.
`verbose`	Be verbose.
`...`	Passed to the function `plot` when it is called.

Details

None.

Value

None.

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

See Also

- `CTLscan` - Scan for CTL
- `print.CTLscan` - Print a summary of a CTLscan
- `par` - Plot parameters
- `colors` - Colors used in plotting

Examples

```
library(ctl)          # Load CTL library
data(ath.metabolites) # Arabidopsis Thaliana data
data(ath.result)      # Arabidopsis Thaliana results
qtlimage(ath.result, ath.metab$map)  # Plot only the qtls
```

`QTLmapping` *QTLmapping - QTL mapping method for CTL analysis*

Description

Internal QTL mapping method used by the CTL analysis, associates every column in the genotypes with a single phenotype

Usage

```
QTLmapping(genotypes, phenotypes, phenocol = 1, verbose = TRUE)
```

Arguments

`genotypes`	Matrix of genotypes. (individuals x markers)
`phenotypes`	Matrix of phenotypes. (individuals x phenotypes)
`phenocol`	Which phenotype column(s) should we analyse. Default: Analyse a single phenotype.
`verbose`	Be verbose.

Details

TODO

- NOTE: Slow approach, it is adviced to use your own QTL mapping data

Value

vector of LOD scores for each genotype column, for phenotype column phenocol

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Main function to scan for CTL
- CTLscan.cross - Use an R/qtl cross object with CTLscan
- CTLsignificant - Significant interactions from a CTLscan
- plot.CTLscan - Plot the CTL curve for a single trait

Examples

```
library(ctl)
data(ath.metabolites) # Arabidopsis Thaliana dataset
qtldata <- QTLmapping(ath.metab$genotypes, ath.metab$phenotypes, phenocol = 23)
plot(qtldata)         # Plot the results of the QTL scan for the phenotype
```

---

scanSD — *scanSD - Analyze the differences in Standard Deviation between genotypes between two traits*

Description

Analyze the differences in Standard Deviation between genotypes between two traits

Usage

```
scanSD(genotypes, phenotypes, phenocol=c(1,2), doRank = FALSE)
```

Arguments

genotypes	Matrix of genotypes. (individuals x markers)
phenotypes	Matrix of phenotypes. (individuals x phenotypes)
phenocol	Which phenotype column(s) should be plotted against each other, Default: phenotype 1 versus 2
doRank	Transform quantitative data into ranked data before analyzing the slope.

Details

TODO

Value

TODO

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Main function to scan for CTL
- CTLsignificant - Significant interactions from a CTLscan
- CTLnetwork - Create a CTL network from a CTLscan
- image.CTLobject - Heatmap overview of a CTLscan
- plot.CTLscan - Plot the CTL curve for a single trait

Examples

```
library(ctl)
data(multitrait)      # Arabidopsis Thaliana (R/qtl cross object)

sds <- scanSD(pull.geno(multitrait),pull.pheno(multitrait))
```

---

scanSD.cross — *scanSD.cross - Analyze the differences in standard deviation between two traits at a certain genetic marker (R/qtl cross object)*

Description

Analyze the differences in standard deviation between two traits at a certain genetic marker

Usage

```
scanSD.cross(cross, phenocol = c(1,2), doRank = FALSE)
```

Arguments

cross	An object of class cross. See read.cross for details.
phenocol	Which phenotype column(s) should be plotted against each other, Default: phenotype 1 versus 2
doRank	Transform quantitative data into ranked data before analyzing the slope.

Details

TODO

Value

TODO

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Main function to scan for CTL
- CTLsignificant - Significant interactions from a CTLscan
- CTLnetwork - Create a CTL network from a CTLscan
- image.CTLobject - Heatmap overview of a CTLscan
- plot.CTLscan - Plot the CTL curve for a single trait

Examples

```
library(ctl)
data(multitrait)      # Arabidopsis Thaliana (R/qtl cross object)

sds <- scanSD.cross(multitrait)
```

---

scanSlopes *scanSlopes - Create a slope difference profile between two traits*

Description

Create a slope difference profile between two traits

Usage

```
scanSlopes(genotypes, phenotypes, phenocol = 1, doRank = FALSE, verbose = FALSE)
```

Arguments

genotypes	Matrix of genotypes. (individuals x markers)
phenotypes	Matrix of phenotypes. (individuals x phenotypes)
phenocol	Which phenotype column(s) should we analyse. Default: Analyse phenotype 1.
doRank	Transform quantitative data into ranked data before analyzing the slope.
verbose	Be verbose.

Details

TODO

Value

TODO

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Main function to scan for CTL
- CTLsignificant - Significant interactions from a CTLscan
- CTLnetwork - Create a CTL network from a CTLscan
- image.CTLobject - Heatmap overview of a CTLscan
- plot.CTLscan - Plot the CTL curve for a single trait

Examples

```
library(ctl)
data(ath.metabolites)                # Arabidopsis Thaliana data set

slopes <- scanSlopes(ath.metab$genotypes, ath.metab$phenotypes[,1:4], phenocol = 2)
image(1:nrow(slopes), 1:ncol(slopes), -log10(slopes))
```

scanSlopes.cross *scanSlopes.cross - Create a slope difference profile between two traits (R/qtl cross object)*

Description

Create a slope difference profile between two traits (using an R/qtl cross object)

Usage

```
scanSlopes.cross(cross, phenocol = 1, doRank = FALSE, verbose = FALSE)
```

Arguments

cross	An object of class cross. See read.cross for details.
phenocol	Which phenotype column(s) should we analyse. Default: Analyse phenotype 1
doRank	Transform quantitative data into ranked data before analyzing the slope.
verbose	Be verbose.

Details

TODO

Value

TODO

Note

TODO

Author(s)

Danny Arends <Danny.Arends@gmail.com>
Maintainer: Danny Arends <Danny.Arends@gmail.com>

References

TODO

See Also

- CTLscan - Main function to scan for CTL
- CTLsignificant - Significant interactions from a CTLscan
- CTLnetwork - Create a CTL network from a CTLscan
- image.CTLobject - Heatmap overview of a CTLscan
- plot.CTLscan - Plot the CTL curve for a single trait

Examples

```
library(ctl)
data(multitrait)      # Arabidopsis Thaliana (R/qtl cross object)
multitrait$pheno <- multitrait$pheno[,1:4]
slopes <- scanSlopes.cross(multitrait)
image(1:nrow(slopes), 1:ncol(slopes), -log10(slopes))
```

`yeast.brem` *Example gene expression data from Saccharomyces cerevisiae on 301 RNA expressions.*

Description

Saccharomyces recombinant inbred lines. There are 109 lines, 301 phenotypes, genotyped at 282 markers on 16 chromosomes stored as a list with 3 matrices: genotypes, phenotypes and map

Usage

```
data(yeast.brem)
```

Format

Data stored in a list holding 3 matrices genotypes, phenotypes and map

Details

Saccharomyces recombinant inbred lines. There are 109 lines, 301 RNA expression phenotypes. The individuals are genotyped at 282 markers on 16 chromosomes.

Source

Saccharomyces cerevisiae RNA expression data from XX, senior author: Rachel Brem 20XX, Published in: Plos

References

TODO

Examples

```
library(ctl)
data(yeast.brem)              # Yeast data set

yeast.brem$genotypes[1:5, 1:5]
yeast.brem$phenotypes[1:5, 1:5]
yeast.brem$map[1:5, ]
```

Index