

Package ‘statgenSTA’

September 23, 2020

Type Package

Title Single Trial Analysis (STA) of Field Trials

Version 1.0.6

Date 2020-09-23

Description Phenotypic analysis of field trials using mixed models with and without spatial components. One of a series of statistical genetic packages for streamlining the analysis of typical plant breeding experiments developed by Biometris.
Some functions have been created to be used in conjunction with the R package 'asreml' for the 'ASReml' software, which can be obtained upon purchase from 'VSN' international (<<http://www.vsn.co.uk/software/asreml-r>>).

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 3.3)

Imports emmeans, ggplot2 (>= 3.3.0), ggrepel, gridExtra, knitr, lme4, mapproj, maps, qtl, scales (>= 1.1.0), SpATS (>= 1.0-10), xtable

Suggests asreml(>= 3.0), rmarkdown, testthat, tibble

SystemRequirements pdflatex

VignetteBuilder knitr

URL <https://biometris.github.io/statgenSTA/index.html>,
<https://github.com/Biometris/statgenSTA/>

BugReports <https://github.com/Biometris/statgenSTA/issues>

NeedsCompilation no

Author Bart-Jan van Rossum [aut, cre],
Fred van Eeuwijk [ctb] (<<https://orcid.org/0000-0003-3672-2921>>),
Martin Boer [ctb],
Marcos Malosetti [ctb] (<<https://orcid.org/0000-0002-8150-1397>>),

Daniela Bustos-Korts [ctb] (<<https://orcid.org/0000-0003-3827-6726>>),
 Emilie J. Millet [ctb] (<<https://orcid.org/0000-0002-2913-4892>>),
 Joao Paulo [ctb] (<<https://orcid.org/0000-0002-4180-0763>>),
 Maikel Verouden [ctb] (<<https://orcid.org/0000-0002-4893-3323>>),
 Willem Kruijer [ctb] (<<https://orcid.org/0000-0001-7179-1733>>),
 Ron Wehrens [ctb] (<<https://orcid.org/0000-0002-8798-5599>>),
 Choazhi Zheng [ctb] (<<https://orcid.org/0000-0001-6030-3933>>)

Maintainer Bart-Jan van Rossum <bart-jan.vanrossum@wur.nl>

Repository CRAN

Date/Publication 2020-09-23 11:50:03 UTC

R topics documented:

as.data.frame.extract	2
extract	3
extractSTA	5
fitTD	7
getMeta	11
outlierSTA	12
plot.STA	14
plot.TD	16
report	19
report.STA	20
STAtoCross	21
STAtoTD	23
summary.STA	24
summary.TD	25
TD	27
TDHeat05	30
TDMaize	31
wheatChl	32

Index	34
--------------	-----------

as.data.frame.extract *Coerce extracted heritabilities to data.frame*

Description

Function for coercing heritabilities extracted from an STA object to a data.frame for nicer printing and easier post processing.

Usage

```
## S3 method for class 'extract'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	An object of class extract.
row.names	An optional character vector of row.names to be added to the data.frame.
optional	Ignored.
...	Ignored.

extract	<i>Extract statistics from Fitted Models</i>
---------	--

Description

This function is deprecated. Use [extractSTA](#) for extracting results from fitted models.

Usage

```
extract(
  STA,
  trials = names(STA),
  traits = NULL,
  what = "all",
  keep = NULL,
  restoreColNames = FALSE
)
```

Arguments

STA	An object of class STA.
trials	A character vector of trials for which the statistics should be computed. If not supplied, statistics are computed for all trials that have been modeled.
traits	A character vector of traits for which the statistics should be computed. If not supplied, statistics are computed for all traits that have been modeled.
what	A character vector indicating which statistics should be computed. Most statistics are available for all models, some only for models fitted using a certain engine. If this is the case, this is indicated in the list with options in details. If what = "all", all available statistics are computed.
keep	A character vector of column(s) in the object of class TD used for modeling. These columns will be kept as output when computing fitted values, residuals, standardized residuals and rMeans. Columns can also be kept when computing (se)BLUEs and (se)BLUPs but only if the column to keep contains unique values for the modeled variables, i.e. a column repId with several different values per genotype cannot be kept.
restoreColNames	Should the original column names be restored in the output of the extracted data?

Details

Possible options for what are:

- F - BLUEs** Best Linear Unbiased Estimators.
- F - seBLUES** Standard errors of the BLUEs.
- R - BLUPs** Best Linear Unbiased Predictors.
- R - seBLUPs** Standard errors of the BLUPs.
- F - ue** Unit errors - only for lme4 and asreml.
- R - heritability** Heritability.
- F - varCompF** Variance components for model with genotype as fixed component.
- R - varCompR** Variance components for model with genotype as random component.
- R - varGen** Genetic variance component(s).
- R - varErr** Residual variance component(s) - only for lme4 and asreml.
- R - varSpat** Spatial variance components - only for SpATS.
- F - fitted** Fitted values for the model with genotype as fixed component.
- F - residF** Residuals for the model with genotype as fixed component.
- F - stdResF** Standardized residuals for the model with genotype as fixed component
- R - rMeans** Fitted values for the model with genotype as random component.
- R - ranEf** Random genetic effects.
- F - residR** Residuals for the model with genotype as random component.
- F - stdResR** Standardized residuals for the model with genotype as random component
- F - wald** Results of the wald test - only for lme4 and asreml.
- F - CV** Coefficient of variation - only for lme4 and asreml.
- F - rDfF** Residual degrees of freedom for the model with genotype as fixed component.
- R - rDfR** Residual degrees of freedom for the model with genotype as random component.
- R - effDim** Effective dimensions - only for SpATS.
- R - ratEffDim** Ratio's of the effective dimensions - only for SpATS.
- F - sed** Standard error of difference - only for asreml.
- F - lsd** Least significant difference - only for asreml.
- all** All available statistics.

Value

A list with, per trial for which statistics have been extracted, a list of those statistics.

See Also

[fitTD](#)

extractSTA	<i>Extract statistics from fitted models</i>
------------	--

Description

Extract and calculate various results for fitted models such as BLUEs, BLUPs, unit errors and heritabilities. For a full list of results that can be extracted, see Details.

The result(s) to extract are specified in `what`. If a single result is extracted, if possible, this result is returned as a `data.frame`. If this is not possible, the result is returned as a list. If multiple results are extracted at the same time, these are always returned as a list. Where relevant, this behavior can be changed by specifying `asDataFrame`. Results that are returned as `data.frame` are marked as such in the `data.frame` column in the table in Details.

Most results can only be calculated if a model is fitted with genotype as fixed or with genotype as random. E.g. to compute heritabilities a model should be fitted with genotype as random effect. This is indicated in the list in Details with "F" and "R" respectively.

Usage

```
extractSTA(
  STA,
  trials = names(STA),
  traits = NULL,
  what = "all",
  asDataFrame = length(what) == 1 && what != "all",
  keep = NULL,
  restoreColNames = FALSE
)
```

Arguments

<code>STA</code>	An object of class <code>STA</code> .
<code>trials</code>	A character vector of trials for which the statistics should be computed. If not supplied, statistics are computed for all trials that have been modeled.
<code>traits</code>	A character vector of traits for which the statistics should be computed. If not supplied, statistics are computed for all traits that have been modeled.
<code>what</code>	A character vector indicating which statistics should be computed. Most statistics are available for all models, some only for models fitted using a certain engine. If this is the case, this is indicated in the list with options in details. If <code>what = "all"</code> , all available statistics are computed.
<code>asDataFrame</code>	Should the output be reshaped to a <code>data.frame</code> . This is only possible if the number of statistics to extract is one.
<code>keep</code>	A character vector of column(s) in the object of class <code>TD</code> used for modeling. These columns will be kept as output when computing fitted values, residuals,

standardized residuals and rMeans. Columns can also be kept when computing (se)BLUEs and (se)BLUPs but only if the column to keep contains unique values for the modeled variables, i.e. a column repId with several different values per genotype cannot be kept.

restoreColNames

Should the original column names be restored in the output of the extracted data?

Details

Possible options for what are:

result	model	description	data.frame
BLUEs	F	Best Linear Unbiased Estimators	yes
seBLUEs	F	standard errors of the BLUEs	yes
ue	F	unit errors - only for lme4 and asreml	yes
varCompF	F	variance components for the model with genotype as fixed component	
fitted	F	fitted values for the model with genotype as fixed component	yes
residF	F	residuals for the model with genotype as fixed component	yes
stdResF	F	standardized residuals for the model with genotype as fixed component	yes
wald	F	results of the wald test - only for lme4 and asreml	
CV	F	Coefficient of Variation - only for lme4 and asreml	yes
rDfF	F	residual degrees of freedom for the model with genotype as fixed component	yes
sed	F	standard error of difference - only for asreml	
lsd	F	least significant difference - only for asreml	
BLUPs	R	Best Linear Unbiased Predictors	yes
seBLUPs	R	standard errors of the BLUPs	yes
heritability	R	broad sense heritability	yes
varCompR	R	variance components for the model with genotype as random component	
varGen	R	genetic variance component	yes
varErr	R	residual variance component	yes
varSpat	R	spatial variance components - only for SpATS	
rMeans	R	fitted values for the model with genotype as random component	yes
ranEf	R	random genetic effects	yes
residR	R	residuals for the model with genotype as random component	yes
stdResR	R	standardized residuals for the model with genotype as random component	yes
rDfR	R	residual degrees of freedom for the model with genotype as random component	yes
effDim	R	effective dimensions - only for SpATS	
ratEffDim	R	ratios of the effective dimensions - only for SpATS	

Value

A list with, per trial for which statistics have been extracted, a list of those statistics.

See Also

[fitTD](#)

Examples

```
## Fit model using SpATS.
myModel <- fitTD(TD = TDHeat05, design = "res.rowcol", traits = "yield")

## Extract all available statistics from the fitted model.
extr <- extractSTA(myModel)

## Extract only the BLUEs from the fitted model.
BLUEs <- extractSTA(myModel, what = "BLUEs")

## Extract only the BLUEs from the fitted model and keep trial as variable in
## the output.
BLUEs2 <- extractSTA(myModel, what = "BLUEs", keep = "trial")
```

fitTD

Fit single trial mixed model

Description

Perform REML analysis given a specific experimental design using either SpATS, lme4 or asreml. SpATS is used as a default method when row coordinates (rowCoord) and column coordinates (colCoord) are present, lme4 otherwise. See details for the exact models fitted.

Usage

```
fitTD(
  TD,
  trials = names(TD),
  design = NULL,
  traits,
  what = c("fixed", "random"),
  covariates = NULL,
  useCheckId = FALSE,
  spatial = FALSE,
  engine = NA,
  control = NULL,
  progress = FALSE,
  ...
)
```

Arguments

TD An object of class [TD](#).

trials A character vector specifying the trials for which the models should be fitted.

design	A character string specifying the experimental design. Either "ibd" (incomplete block design), "res.ibd" (resolvable incomplete block design), "rcbd" (randomized complete block design), "rowcol" (row column design) or "res.rowcol" (resolvable row column design). Can be ignored when the trial design is specified in the meta data (see setMeta).
traits	A character vector specifying the traits for which the models should be fitted.
what	A character vector specifying whether "genotype" should be fitted as fixed or random effect. If not specified, both models are fitted.
covariates	A character vector specifying covariates to be fitted as extra fixed effects in the model.
useCheckId	Should checkId be used as a fixed effect in the model? If TRUE, TD has to contain a column 'checkId'.
spatial	Should spatial models be tried? Spatial models can only be fitted with SpATS and asreml. If SpATS is used for modeling, only spatial models can be fitted and spatial is always set to TRUE. If asreml is used, fitting spatial models is optional.
engine	A character string specifying the name of the mixed modeling engine to use, either "SpATS", "lme4" or "asreml." For spatial models, "SpaTS" is used as default, for other models "lme4".
control	An optional list with control parameters to be passed to the actual fitting functions. Currently nSeg and nestDiv are valid parameters when fitting a model using SpATS. They pass a value to nseg and nest.div in PSANOVA respectively. For nSeg also a named list can be supplied containing values for nSeg per trial. criterion is a valid parameter when fitting a spatial model using asreml. It may be used to pass a goodness-of-fit criterion for comparing different spatial models. See also in details. Other parameters are ignored.
progress	Should the progress of the modeling be printed. If TRUE, for every trial a line is output indicating the traits fitted for the particular trial.
...	Further arguments to be passed to SpATS, lme4 or asreml.

Details

The actual model fitted depends on the design. For the supported designs, the following models are used:

Design	Code	Model fitted
incomplete block design	ibd	trait = subBlock + genotype + ϵ
resolvable incomplete block design	res.ibd	trait = <i>repId</i> + repId:subBlock + genotype + ϵ
randomized complete block design	rcbd	trait = <i>repId</i> + genotype + ϵ
row column design	rowcol	trait = rowId + colId + genotype + ϵ
resolvable row column design	res.rowcol	trait = <i>repId</i> + repId:rowId + repId:colId + genotype + ϵ

In the models above, fixed effects are indicated in *italics* whereas random effects are indicated in **bold**. genotype can be fitted as fixed or as random effect depending on the value of the parameter what. Extra fixed effects may be fitted using the parameter covariates.

If SpATS is used as modeling engine, an extra spatial term is always included in the model. This term is constructed using the function `PSANOVA` from the SpATS package as `PSANOVA(colCoord, rowCoord, nseg = nSeg, nest.div = 2)` where `nSeg = (number of columns / 2, number of rows / 2)`. `nseg` and `nest.div` can be modified using the `control` parameter.

When `asreml` is used for modeling and `spatial` is TRUE six models are fitted with different random terms and covariance structure. The best model is determined based on a goodness-of-fit criterion, either AIC or BIC. This can be set using the control parameter `criterion`, default is AIC. The fitted random terms depend on the structure of the data. If the design has a regular structure, i.e. all replicates appear the same amount of times in the design, the following combinations of random and spatial terms are fitted:

Random part	Spatial part
random effects based on design	none
random effects based on design	ar1(rowId):colId
random effects based on design	rowId:ar1(colId)
random effects based on design	ar1(rowId):ar1(colId)
random effects based on design + nugget	ar1(rowId):colId
random effects based on design + nugget	rowId:ar1(colId)
random effects based on design + nugget	ar1(rowId):ar1(colId)

If the design is not regular the following combinations of random and spatial terms are fitted:

Random part	Spatial part
random effects based on design	none
random effects based on design	exp(rowCoord):colCoord
random effects based on design	rowCoord:exp(colCoord)
random effects based on design	iexp(rowCoord, colCoord)
random effects based on design + nugget	exp(rowCoord):colCoord
random effects based on design + nugget	rowCoord:exp(colCoord)
random effects based on design + nugget	iexp(rowCoord,colCoord)

Value

An object of class STA, a list containing, per trial that has been analyzed, a list of:

<code>mRand</code>	A list of models with fitted with genotype as random effect.
<code>mFix</code>	A list of models fitted with genotype as fixed effect.
<code>TD</code>	An object of class TD containing the data on which <code>mRand</code> and <code>mFix</code> are based.
<code>traits</code>	A character vector indicating the traits for which the models are fitted.
<code>design</code>	A character string containing the design of the trial. (see <code>fitTD</code> for the possible designs).
<code>spatial</code>	A character string indicating the spatial part of the model. FALSE if no spatial design has been used.

engine	A character string containing the engine used for the analysis.
predicted	A character string indicating the variable that has been predicted.
sumTab	A data.frame with a summary table for the spatial models tried when engine = "asreml" and spatial = TRUE

References

Maria Xose Rodriguez-Alvarez, Martin P. Boer, Fred A. van Eeuwijk, Paul H.C. Eilers (2017). Correcting for spatial heterogeneity in plant breeding experiments with P-splines. *Spatial Statistics* <https://doi.org/10.1016/j.spasta.2017.10.003>

Butler, D. G., et al. (2010). *Analysis of Mixed Models for S language environments: ASReml-R reference manual*. Brisbane, DPI Publications

Douglas Bates, Martin Maechler, Ben Bolker, Steve Walker (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1), 1-48. <https://www.jstatsoft.org/article/view/v067i01/0>.

Examples

```
## Fit model using lme4.
myModel1 <- fitTD(TD = TDHeat05, design = "ibd", traits = "yield",
                 what = "fixed", engine = "lme4")

## Summarize results.
summary(myModel1)

## Create base plots of the results.
plot(myModel1)

## Create a pdf report summarizing results.

report(myModel1, outfile = tempfile(fileext = ".pdf"), what = "fixed")

## Fit model using SpATS.
myModel2 <- fitTD(TD = TDHeat05, design = "res.rowcol", traits = "yield",
                 what = "fixed")
summary(myModel2)

## Create spatial plots of the results.

plot(myModel2, plotType = "spatial")

## Create a pdf report summarizing results.

report(myModel2, outfile = tempfile(fileext = ".pdf"), what = "fixed")

## Fit model using asreml.
if (requireNamespace("asreml", quietly = TRUE)) {
```

```

myModel3 <- fitTD(TD = TDHeat05, design = "res.rowcol", traits = "yield",
                 what = "fixed", engine = "asreml")
summary(myModel3)

report(myModel3, outfile = tempfile(fileext = ".pdf"), what = "fixed")
}

```

getMeta

Get and set metadata for TD objects

Description

Functions for extracting and adding metadata for objects of class TD.

getMeta extracts a data.frame with location, date, design, latitude, longitude, plot width and plot length for all trials in TD.

setMeta adds metadata from a data.frame to an object of class TD. See details for the specifications of the data.frame.

The most common use case is extracting metadata from a TD object, modifying the content and then adding it back to the TD object.

Information in the metadata of a TD object is used in plotting functions (e.g. latitude and longitude for a map plot) and when fitting models on the data (the trial design).

Usage

```
getMeta(TD)
```

```
setMeta(TD, meta)
```

Arguments

TD	An object of class TD.
meta	A data.frame containing metadata.

Details

When setting metadata, metadata has to be a data.frame with rownames corresponding to the trials in TD. The data.frame should contain one or more of the following columns:

trLocation The location of the trial. Used as default name when creating plots and summaries.

trDate The date of the trial.

trDesign The design of the trial. One of "none" (no (known) design), "ibd" (incomplete-block design), "res.ibd" (resolvable incomplete-block design), "rcbd" (randomized complete block design), "rowcol" (row-column design) or "res.rowcol" (resolvable row-column design). Used when fitting models.

trLat The latitude of the trial on a scale of -90 to 90. Used when plotting the trials on a map.

trLong The longitude of the trial on a scale of -180 to 180. Used when plotting the trials on a map.

trPIWidth The width of the plot. Used in combination with trPILength to determine the size of the plots in a layout plot of a trial.

trPILength The length of the plot. Used in combination with trPIWidth to determine the size of the plots in a layout plot of a trial.

The values of the metadata of TD will be set to the values in the corresponding column in meta. Existing values will be overwritten, but NA will be ignored so setting a value to NA won't result in accidentally removing it.

See Also

Other functions for TD objects: [TD](#), [plot.TD\(\)](#), [summary.TD\(\)](#)

Examples

```
data("wheatCh1")

## Create a TD object.
wheatTD <- createTD(data = wheatCh1, genotype = "trt", repId = "rep",
                   subBlock = "bl", rowCoord = "row", colCoord = "col")

## Get meta data from wheatTD
(wheatMeta <- getMeta(wheatTD))

## Add location names and latitude/longitude to meta data.
wheatMeta$strLocation <- c("Cauquenes", rep("Santa Rosa", times = 4))
wheatMeta$strLat <- c(-35.58, rep(-36.32, times = 4))
wheatMeta$strLong <- c(-72.17, rep(-71.55, times = 4))

## Add back meta data to wheatTD.
wheatTD <- setMeta(wheatTD, wheatMeta)
```

Description

Function to identify observations with standardized residuals exceeding `rLimit`. If not provided `rLimit` is computed as $qnorm(1 - 0.5 / rDf)$ where `rDf` is the residual degrees of freedom for the model. This value is then restricted to the interval 2..4. Alternatively a custom limit may be provided.

If `verbose = TRUE` a summary is printed of outliers and observations that have the same value for `commonFactors`. The column `outlier` in the output can be used to distinguish real outliers from observations included because of their `commonFactors`.

Usage

```
outlierSTA(
  STA,
  trials = NULL,
  traits = NULL,
  what = NULL,
  rLimit = NULL,
  commonFactors = NULL,
  verbose = TRUE
)
```

Arguments

<code>STA</code>	An object of class <code>STA</code> .
<code>trials</code>	A character vector specifying the trials for which outliers should be identified. If <code>trials = NULL</code> , all trials are included.
<code>traits</code>	A character vector specifying the traits for which outliers should be identified.
<code>what</code>	A character string indicating whether the outliers should be identified for the fitted model with genotype as fixed (<code>what = "fixed"</code>) or genotype as random (<code>what = "random"</code>) factor. If <code>STA</code> contains only one model this model is chosen automatically.
<code>rLimit</code>	A numerical value used for determining when a value is considered an outlier. All observations with standardized residuals exceeding <code>rLimit</code> will be marked as outliers.
<code>commonFactors</code>	A character vector specifying the names of columns in <code>TD</code> used for selecting observations that are similar to the outliers. If <code>commonFactors = NULL</code> , only outliers are reported and no similar observations.
<code>verbose</code>	Should the outliers be printed to the console?

Value

A list with two components:

- `indicator` - a list of numeric vectors indicating the location of the outliers in the data
- `outliers` - a `data.frame` containing the outliers and observations similar to the outliers as defined by `commonFactors`

Examples

```
## Fit a model using lme4.
myModel <- fitTD(TD = TDHeat05, traits = "yield", design = "res.rowcol",
  engine = "lme4")
```

```
## Detect outliers in the standardized residuals of the fitted model.
outliers <- outlierSTA(STA = myModel, traits = "yield")
```

plot.STA

Plot function for class STA

Description

This function draws either four base plots:

- A histogram of the residuals
- A normal Q-Q plot
- A residuals vs fitted values plot
- An absolute residuals vs fitted values plot

or five or six spatial plots:

- A spatial plot of the raw data
- A spatial plot of the fitted data
- A spatial plot of the residuals
- A spatial plot of the estimated spatial trend (SpATS only)
- A spatial plot of the BLUEs or BLUPs
- A histogram of the BLUEs or BLUPs

Spatial plots can only be made if the data contains both row and column information.

Usage

```
## S3 method for class 'STA'
plot(
  x,
  ...,
  trials = NULL,
  traits = NULL,
  what = NULL,
  plotType = c("base", "spatial"),
  spaTrend = c("raw", "percentage"),
  outCols = ifelse(plotType == "base", 2, 3),
  title = NULL,
  output = TRUE
)
```

Arguments

x	An object of class STA.
...	Further graphical parameters.
trials	A character vector indicating the trials to plot. If trials = NULL, all trials are plotted.
traits	A character vector indicating the traits to plot. If traits = NULL, all traits are plotted.
what	A character string indicating whether the fitted model with genotype as fixed (what = "fixed") or genotype as random (what = "random") factor should be plotted. If x contains only one model this model is chosen automatically.
plotType	A character string indicating whether base plots or spatial plots should be made.
spaTrend	A character string indicating how the spatial trend should be displayed. Either "raw" (original scale), or "percentage". If "percentage", the estimated spatial trend is scaled (i.e., divided by the average of the observed response variable of interest across the field) and results are shown as a percentage.
outCols	An integer indicating the number of columns to use for displaying the plots. Usually the default of 2 for base plots and 3 for spatial plots will be fine, but decreasing the numbers may help for nicer printing.
title	A character string used a title for the plot. Note that when a title is specified and multiple plots are created, all plots will get the same title.
output	Should the plot be output to the current device? If FALSE only a list of ggplot objects is invisibly returned.

Value

A list containing ggplot objects for the selected plots.

See Also

Other functions for STA objects: [STAtoCross\(\)](#), [STAtoTD\(\)](#), [report.STA\(\)](#), [summary.STA\(\)](#)

Examples

```
## Run a single trait analysis using SpATS.
myModel <- fitTD(TD = TDHeat05, design = "res.rowcol", traits = "yield")

## Create base plots.
plot(myModel, what = "fixed", plotType = "base")

## Create spatial plots.
plot(myModel, what = "fixed", plotType = "spatial")

## Create spatial plots showing the spatial trend as percentage.
plot(myModel, what = "fixed", plotType = "spatial", spaTrend = "percentage")
```

plot.TD

Plot function for class TD

Description

Plotting function for objects of class TD. Plots either the layout of the different trials within the TD object or locates the trials on a map. Also a boxplot can be made for selected traits and trials, a plot of correlations between trials and a scatter plot matrix. A detailed description and optional extra parameters of the different plots is given in the sections below.

Usage

```
## S3 method for class 'TD'
plot(
  x,
  ...,
  plotType = c("layout", "map", "box", "cor", "scatter"),
  trials = names(x),
  traits = NULL,
  title = NULL,
  output = TRUE
)
```

Arguments

x	An object of class TD.
...	Extra plot options. Described per plotType in their respective section.
plotType	A single character string indicating which plot should be made. See the sections below for a detailed explanation of the plots.
trials	A character vector indicating which trials to include in the plot.
traits	A character vector indicating for which traits a plot should be made. Only used if plotType = "box", "cor", and "scatter".
title	A character string used a title for the plot. Note that when a title is specified and multiple plots are created, all plots will get the same title.
output	Should the plot be output to the current device? If FALSE only a list of ggplot objects is invisibly returned.

Layout Plot

Plots the layout of the selected trials. This plot can only be made for trials that contain both row (rowCoord) and column (colCoord) information. If either one of those is missing the trial is skipped with a warning. If blocks (subBlock) are available for a trial these can be colored in different colors per block by setting colorSubBlock = TRUE. If replicates (repId) are available a black line is plotted between different replicates. Missing plots are indicated in white. These can either be single plots in a trial or complete missing columns or rows.

Extra parameter options:

- showGeno** Should individual genotypes be indicated as text in the plot? Defaults to FALSE
- highlight** A character vector of genotypes to be highlighted in the plot.
- colorSubBlock** Should blocks be colored with a different color per subBlock? Defaults to FALSE. colorSubBlock is ignored when highlight is used to highlight genotypes.

Map Plot

A map is plotted with the locations of the trials in the TD object. Mapping the trials is done based on latitude and longitude that can be added when creating an object of class TD. Trials for which either latitude or longitude is not available are skipped with a warning message. The countries in which the trials are located will be plotted on a single map and the location of the trials will be indicated on this map. The actual plot is made using ggplot, but for getting the data for the borders of the countries the maps package is needed.

Extra parameter options:

- colorTrialBy** A character string indicating a column in TD by which the trials on the map are colored.
- colTrial** A character vector with plot colors for the trials. A single color when colorTrialBy = NULL, a vector of colors otherwise.
- printTrialsNames** Should trial names be printed. Defaults to TRUE. Setting this to FALSE can be useful if there are many trials.
- minLatRange** A positive numerical value indicating the minimum range (in degrees) for the latitude on the plotted map. Defaults to 10.
- minLongRange** A positive numerical value indicating the minimum range (in degrees) for the longitude on the plotted map. Defaults to 5.

Box Plot

Creates a boxplot per selected trait grouped by trial. Extra parameter options:

- groupBy** A character string indicating a column in TD by which the boxes in the plot should be grouped. By default the boxes are grouped per trial.
- colorTrialBy** A character string indicating a column in TD by which the boxes are colored. Coloring will be done within the groups indicated by the groupBy parameter.
- colTrial** A character vector with plot colors for the trials. A single color when colorTrialBy = NULL, a vector of colors otherwise.
- orderBy** A character string indicating the way the boxes should be ordered. Either "alphabetic" for alphabetical ordering of the groups, "ascending" for ordering by ascending mean, or "descending" for ordering by descending mean. Default boxes are ordered alphabetically.

Correlation Plot

Draws a heat map of correlations between trials per selected trait. If genotypes are replicated within trials genotypic means are taken before computing correlations. The order of the trials in the heat map is determined by clustering them. Closely related trials will be plotted close to each other.

Scatter Plot

Draws a scatter plot matrix per selected trait. If genotypes are replicated within trials genotypic means are taken before plotting. The lower left of the matrix contains scatter plots between trials. The diagonal contains histograms of the data per trial.

Extra parameter options:

colorGenoBy A character string indicating a column in TD by which the genotypes in the scatter plots are colored.

colGeno A character vector with plot colors for the genotypes. A single color when colorGenoBy = NULL, a vector of colors otherwise.

colorTrialBy A character string indicating a column in TD by which the trials in the histograms are colored.

colTrial A character vector with plot colors for the trials. A single color when colorTrialBy = NULL, a vector of colors otherwise.

trialOrder A character vector indicating the order of the trials in the plot matrix (left to right and top to bottom). This vector should be a permutation of all trials plotted.

addCorr A character string indicating the position of the correlation between trials displayed in each plot, either "tl" for top left, "tr", for top right, "bl" for bottom left or "br" for bottom right. If NULL, the default, then no correlation is added to the plot.

See Also

Other functions for TD objects: [TD](#), [getMeta\(\)](#), [summary.TD\(\)](#)

Examples

```
data("wheatCh1")

## Create a TD object.
wheatTD <- createTD(data = wheatCh1, genotype = "trt", repId = "rep",
                   subBlock = "bl", rowCoord = "row", colCoord = "col")

## Add meta data to be able to plot locations on a map.
wheatMeta <- getMeta(wheatTD)
wheatMeta$trLocation <- c("Cauquenes", rep("Santa Rosa", times = 4))
wheatMeta$trLat <- c(-35.58, rep(-36.32, times = 4))
wheatMeta$trLong <- c(-72.17, rep(-71.55, times = 4))
wheatTD <- setMeta(wheatTD, wheatMeta)

### Layout plot.

## Plot the layout of one of the trials.
plot(wheatTD, trials = "C_SWS_12")

## Highlight some of the genotypes in the layout.
plot(wheatTD, trials = "C_SWS_12", highlight = c("G001", "G002"))

### Map plot.
```

```
## Plot the location of the trials on the map.
plot(wheatTD, plotType = "map")

### Box plot.

## Create a box plot for GY.
plot(wheatTD, plotType = "box", traits = "GY")

## Add coloring by repId to the boxes.
plot(wheatTD, plotType = "box", traits = "GY", colorTrialBy = "repId")

## Sort the boxes in descending order.
plot(wheatTD, plotType = "box", traits = "GY", orderBy = "descending")

### Correlation plot.

## Plot the correlations between trials for GY.
plot(wheatTD, plotType = "cor", traits = "GY")

### Scatter plot

## Plot scatter plot for GY.
plot(wheatTD, plotType = "scatter", traits = "GY")

## Add correlations to top left corner of plots.
plot(wheatTD, plotType = "scatter", traits = "GY", addCorr = "t1")
```

report

Base method for creating a report

Description

Base method for creating a .pdf and .tex report from an R object.

Usage

```
report(x, ...)
```

Arguments

x	An R object
...	Further arguments to be passed on to specific report functions.

See Also

[report.STA](#)

report.STA	<i>Report method for class STA</i>
------------	------------------------------------

Description

pdf reports will be created containing a summary of the results of the fitted model(s). For all selected trails and traits a separate pdf file will be generated. Also a .tex file and a folder containing figures will be created for each report to enable easy modifying of the report.

Usage

```
## S3 method for class 'STA'
report(
  x,
  ...,
  trials = NULL,
  traits = NULL,
  descending = TRUE,
  outfile = NULL,
  what = c("fixed", "random")
)
```

Arguments

x	An object of class STA.
...	Further arguments passed to the report function.
trials	A character vector indicating the trials to report. If trials = NULL, all trials are reported.
traits	A character vector indicating the traits to report. If traits = NULL, all traits are reported.
descending	Should the trait be ordered in descending order? Set to FALSE if low values of the trait indicate better performance.
outfile	A character string, the name and location of the output .pdf and .tex file for the report. If NULL, a report with a default name will be created in the current working directory. Trial, trait and the type of model (genotype fixed or random) will be concatenated to the name of the output file. Both knitr and pdflatex don't work well with spaces in file paths and these are therefore disallowed. Relative paths are possible though.
what	A character vector indicating whether the fitted model with genotype as fixed or genotype as random factor should be reported. By default all fitted models in the STA object are reported.

Details

This function uses pdflatex to create a pdf report. For it to run correctly an installation of LaTeX is required. Checking for this is done with Sys.which("pdflatex").

Value

A pdf report and the .tex file and figures folder that can be used to recreate the report.

See Also

Other functions for STA objects: [STAtoCross\(\)](#), [STAtoTD\(\)](#), [plot.STA\(\)](#), [summary.STA\(\)](#)

Examples

```
## Fit model using lme4.
myModel1 <- fitTD(TD = TDHeat05, design = "ibd", traits = "yield",
                 engine = "lme4")

## Create a pdf report summarizing the results for the model with genotype
## as fixed factor.

report(myModel1, outfile = tempfile(fileext = ".pdf"), what = "fixed")

## Create two pdf report summarizing the results for the model with genotype
## as fixed factor and for the model with genotype as random factor. Order
## the results in ascending order.

report(myModel1, outfile = tempfile(fileext = ".pdf"), descending = FALSE)
```

STAtoCross

Convert STA to Cross

Description

Convert an STA object to a cross object from package qtl. Genotypic information should be available in a .csv file.

The only way to create an object of class cross is by importing both the phenotypic and the genotypic data from external files. Therefore the phenotypic data, either the BLUEs or the BLUPs from the fitted model are first written to a temporary file. The genotypic data has to be available in a .csv file in the correct format as well, see `genoFile` for a description of this format. These phenotypic and genotypic files are then imported into a cross object using the `read.cross` function in the qtl package.

Usage

```
STAtoCross(
  STA,
  trial = NULL,
  traits = NULL,
  what = c("BLUEs", "BLUPs"),
```


STAtoTD

*Convert STA to TD***Description**

Convert an STA object to a TD object.

To be able to use the output of a single trial analysis in Genotype-by-Environment (GxE) analysis the output first needs to be converted back to an TD object. This function does exactly that. It extracts BLUEs, BLUPs and their standard errors from the STA object and creates a new TD object using these. Also a column "wt" (weight) may also be added. Weights are then calculated as $1/(SE\ BLUEs)^2$.

Usage

```
STAtoTD(
  STA,
  what = c("BLUEs", "seBLUEs", "BLUPs", "seBLUPs"),
  traits = NULL,
  keep = NULL,
  addWt = FALSE
)
```

Arguments

STA	An object of class STA.
what	A character string containing the statistics to be included as traits in the TD object. Multiple statistics can be included in which case they will appear as <code>statistic_trait</code> in the output
traits	A character string containing the traits to be included in the TD object. If NULL, all traits are exported.
keep	Columns from the TD object used as input for the STA model to be copied to the output. see extractSTA for possible columns to copy. If it is available in TD, the column <code>trial</code> will always be copied.
addWt	Should a column <code>wt</code> be added to the output? If TRUE weight is calculated as $1/(SE\ BLUEs)^2$. If multiple traits are included in the output, multiple weight columns will be added, 1 for each trait. These will be named <code>wt_trait</code> .

Details

Trial information for the trials in the STA object will be copied from the original TD object on which the modeling was done.

See Also

Other functions for STA objects: [STAtoCross\(\)](#), [plot.STA\(\)](#), [report.STA\(\)](#), [summary.STA\(\)](#)

Examples

```
## Fit model using SpATS.
myModel <- fitTD(TD = TDHeat05, design = "res.rowcol", traits = "yield",
                what = "fixed")

## Create TD object from the fitted model with BLUEs and standard errors.
myTD <- STAtoTD(myModel, what = c("BLUEs", "seBLUEs"))

## Add a weight column in the output.
myTDWt <- STAtoTD(myModel, what = c("BLUEs", "seBLUEs"), addWt = TRUE)
```

summary.STA

*Summarizing objects of class STA***Description**

summary method for class STA.

Usage

```
## S3 method for class 'STA'
summary(
  object,
  trials = NULL,
  trait = NULL,
  nBest = 20,
  sortBy = NULL,
  naLast = TRUE,
  decreasing = TRUE,
  ...
)
```

Arguments

object	An object of class STA.
trials	A character vector indicating the trial to summarize. If <code>trials = NULL</code> a summary is made of all trials in the STA object. If a single trial is selected a full summary for this trial is created. For multiple trials a summary table with the most important statistics is returned.
trait	A character string indicating the trait to summarize. If <code>trait = NULL</code> and only one trait is modeled, this trait is summarized.
nBest	An integer indicating the number of the best genotypes (sorted by either BLUEs or BLUPs) to print. If NA, all genotypes will be printed.
sortBy	A character string specifying how the genotypes will be sorted. Either "BLUEs", "BLUPs" or NA (i.e. no sorting).

naLast	Should missing values in the data be put last when sorting?
decreasing	Should the sort order be decreasing?
...	Further arguments - not used.

See Also

Other functions for STA objects: [STAtoCross\(\)](#), [STAtoTD\(\)](#), [plot.STA\(\)](#), [report.STA\(\)](#)

Examples

```
## Run a single trait analysis using SpATS.
myModel <- fitTD(TD = TDHeat05, design = "res.rowcol", traits = "yield")

## Print a summary of the fitted model.
summary(myModel)
```

summary.TD

*Summarizing objects of class TD***Description**

summary method for class TD.

Usage

```
## S3 method for class 'TD'
summary(
  object,
  ...,
  trial = names(object),
  traits,
  groupBy = NULL,
  what = if (!is.null(groupBy)) { c("nObs", "mean", "sd") } else { c("nObs",
    "nMiss", "mean", "median", "min", "max", "firstQ", "thirdQ", "var") }
)
```

Arguments

object	An object of class TD.
...	Further arguments - currently not used.
trial	A character string specifying the trial to be summarized.
traits	A character vector specifying the traits to be summarized.
groupBy	A character string specifying a column in TD by which the summary should be grouped. If NULL, no grouping is done.

what A character vector indicating which summary statistics should be computed. If `what = "all"`, all available statistics are computed. Possible options are:

- nVals** The number of values, i.e. non-missing + missing values.
- nObs** The number of non-missing observations.
- nMiss** The number of missing values.
- mean** The mean.
- median** The median.
- min** The minimum.
- max** The maximum.
- range** The range (maximum - minimum).
- firstQ** The first (25pct) quantile.
- thirdQ** The third (75pct) quantile.
- sd** The standard deviation.
- seMean** The standard error of mean.
- var** The variance.
- seVar** The standard error of variance.
- CV** The coefficient of variation.
- sum** The sum.
- sumSq** The sum of squares.
- uncorSumSq** The uncorrected sum of squares.
- skew** The skewness.
- seSkew** The standard error of the skewness.
- kurt** The kurtosis.
- seKurt** The standard error of the kurtosis.
- all** All summary statistics.

Value

A table containing the selected summary statistics.

See Also

Other functions for TD objects: [TD](#), [getMeta\(\)](#), [plot.TD\(\)](#)

Examples

```
## Summarize TDHeat05.
summary(TDHeat05, traits = "yield")

## Summarize TDHeat05 grouping by repId.
summary(TDHeat05, traits = "yield", groupBy = "repId")
```

TD

S3 class TD

Description

`createTD`

Function for creating objects of S3 class TD (Trial Data). The function converts a data.frame to an object of class TD in the following steps:

- Check input data
- Rename columns to default column names - default column names: genotype, trial, loc, year, repId, subBlock, rowCoord, colCoord, rowId, colId, checkId
- Convert column types to default column types - rowCoord and colCoord are converted to numeric columns, all other renamed columns to factor columns. Columns other than the default columns, e.g. traits or other covariates will be included in the output unchanged.
- Split input data by trial - each trial in the input data will become a list item in the output.
- Add meta data - the trial meta data are added as attributes to the different output items. The function parameters starting with "tr" provide the meta data. Their values will be recycled if needed, so by setting a single "trDesign", all trials will get the same design. For trLat, trLong, trDesign and trDate a column in data that contains the information can be specified as well. The meta data can be changed later on using `getMeta` and `setMeta`

`addTD`

Function for adding extra trial data to an existing object of class TD. The data for the new trials will be added after the data for existing trials. It is possible to add data for an already existing trial, but this will cause multiple items in the output with identical names, which might cause problems later on in the analysis. Therefore a warning will be issued in this case.

`dropTD`

Function for removing data for selected trials from an existing object of class TD.

`summary.TD` and `plot.TD` methods are available.

Usage

```
createTD(  
  data,  
  genotype = NULL,  
  trial = NULL,  
  loc = NULL,  
  year = NULL,  
  repId = NULL,  
  subBlock = NULL,  
  plotId = NULL,  
  rowCoord = NULL,  
  colCoord = NULL,  
)
```

```

    rowId = rowCoord,
    colId = colCoord,
    checkId = NULL,
    trLocation = NULL,
    trDate = NULL,
    trDesign = NULL,
    trLat = NULL,
    trLong = NULL,
    trPlWidth = NULL,
    trPlLength = NULL
  )

```

```

addTD(
  TD,
  data,
  genotype = NULL,
  trial = NULL,
  loc = NULL,
  year = NULL,
  repId = NULL,
  subBlock = NULL,
  plotId = NULL,
  rowCoord = NULL,
  colCoord = NULL,
  rowId = rowCoord,
  colId = colCoord,
  checkId = NULL,
  trLocation = NULL,
  trDate = NULL,
  trDesign = NULL,
  trLat = NULL,
  trLong = NULL,
  trPlWidth = NULL,
  trPlLength = NULL
)

```

```
dropTD(TD, rmTrials)
```

Arguments

data	A data.frame containing trial data with at least a column for genotype. The data.frame should be in a wide format, i.e. all available phenotypic data should be in a separate column within the data.frame.
genotype	An optional character string indicating the column in data that contains genotypes.
trial	An optional character string indicating the column in data that contains trials.
loc	An optional character string indicating the column in data that contains trial locations.

year	An optional character string indicating the column in data that contains years.
repId	An optional character string indicating the column in data that contains replicates.
subBlock	An optional character string indicating the column in data that contains sub blocks.
plotId	An optional character string indicating the column in data that contains plots. This column will be combined with trial to a single output factor.
rowCoord	An optional character string indicating the column in data that contains the row coordinates.
colCoord	An optional character string indicating the column in data that contains the column coordinates.
rowId	An optional character string indicating the column in data that contains field rows. If not supplied, this is assumed to be the same as rowCoord.
colId	An optional character string indicating the column in data that contains field columns. If not supplied, this is assumed to be the same as colCoord.
checkId	An optional character string indicating the column in data that contains the check IDs.
trLocation	An optional character vector indicating the locations of the trials. This will be used as default names when creating plots and summaries. If no locations are provided, first the column loc is considered. If this contains one unique value for a trial this is used as trLocation. Otherwise the trial name is used.
trDate	An optional character string indicating the column in data that contains the date of the trial or a date vector indicating the dates of the trials.
trDesign	An optional character string indicating the column in data that contains the design of the trial or a character vector indicating the designs of the trials. Either "none" (no (known) design), " ibd" (incomplete-block design), "res.ibd" (resolvable incomplete-block design), "rcbd" (randomized complete block design), "rowcol" (row-column design) or "res.rowcol" (resolvable row-column design).
trLat	An optional character string indicating the column in data that contains the latitude of the trial or a numerical vector indicating the latitudes of the trials on a scale of -90 to 90.
trLong	An optional character string indicating the column in data that contains the latitude of the trial or a numerical vector indicating the longitudes of the trials on a scale of -180 to 180.
trPlWidth	An optional positive numerical vector indicating the widths of the plots.
trPlLength	An optional positive numerical vector indicating the lengths of the plots.
TD	An object of class TD which should be modified.
rmTrials	A character vector of trials that should be removed.

Value

An object of class TD, a list of data.frames with renamed columns and an attribute renamedCols containing an overview of renamed columns. For each unique value of trial, the output has a data.frame in the list with the same name as the trial. These data.frames have attributes containing the metadata for the corresponding trial. If there is no column for trial, the list will contain one item named after the input data.

See Also

Other functions for TD objects: [getMeta\(\)](#), [plot.TD\(\)](#), [summary.TD\(\)](#)

Examples

```
## Create a data.frame to be converted to TD object.
## The data consists of genotype, trial, row and column information and
## two traits, yield and flowering time.
datT1 <- data.frame(geno = paste0("G", 1:10), tr = "T1",
                   row = rep(1:5, each = 2), col = rep(1:2, times = 5),
                   yield = 1:10, flowering = 3:12)

## Convert data.frame to TD object.
TDT1 <- createTD(data = datT1, genotype = "geno", trial = "tr",
                rowCoord = "row", colCoord = "col")

## Create a second data.frame similar to the first with data for a second trial.
datT2 <- data.frame(geno = paste0("G", 1:10), tr = "T2",
                   row = rep(1:2, each = 5), col = rep(1:5, times = 2),
                   yield = 10:1, flowering = 12:3)

## Add this data to the TD object created above.
TDTot <- addTD(TD = TDT1, data = datT2, genotype = "geno", trial = "tr",
              rowCoord = "row", colCoord = "col")

## Drop the data for the first trial from the object.
TDT2 <- dropTD(TD = TDTot, rmTrials = "T1")
```

TDHeat05

Field data for a wheat experiment in Mexico

Description

A dataset converted to a TD object containing raw plot data for one trial from a series of wheat trials conducted in Mexico by CIMMYT. The different trials took place under different regimes of irrigation and temperature, there were 4 trials across two years, labelled as DRIP05, HEAT05, HEAT06 and IRR106. The TD object only contains the data for HEAT05. Within each trial, a set of 167 progeny of a RIL (Recombinant Inbred Line; 8 generations) population were tested alongside the population parents (Seri and Babax). A lattice design with two replicates was used for each trial. In the first replicate the entries were not randomized, as they were considered to be a random selection from a population.

Usage

TDHeat05

Format

A TD object, a list containing a data.frame with the following columns:

trial trial, a combination of watering regime, year and nitrogen treatment

genotype genotype

Plot plot number in the field

repId replicate

subBlock block id

rowId row within the field (as factor)

colId column within the field (as factor)

yield yield in grams per square meter

rowCoord row within the field (as numerical value)

colCoord column within the field (as numerical value)

TDMaize

Field data for a maize experiment in Tlaltizapan, Mexico

Description

A dataset converted into a TD object containing data corresponding to an F2 maize reference population from CIMMYT maize drought breeding program, which was derived from the cross of a drought-tolerant line (P1) with a drought susceptible line (P2) as described in detail by Ribaut et al. (1996, 1997).

DNA from 211 F2 plants was extracted to produce information for 132 co-dominant markers on 10 linkage groups. Phenotypic evaluations were performed on 211 F2:3 families, each one derived from an original F2 plant. The families were evaluated under different water and nitrogen regimes during 1992, 1994 and 1996. In the winter of 1992 three water regimes were imposed on the trials: well watered (WW), intermediate stress (IS) and severe stress (SS). In the winter of 1994, only the IS and SS trials were available. Nitrogen availability varied in the 1996 trials, with two low nitrogen treatments (LN, in winter and summer) and one high-nitrogen treatment (HN in summer). In each of the trials, five traits were evaluated but only grain yield is included in the data.

Usage

TDMaize

Format

A TD object, a list containing 8 data.frames, each with the following columns:

trial trial, a combination of watering regime, year and nitrogen treatment

genotype genotype

yld grain yield in tons

Source

<https://link.springer.com/article/10.1007/BF00221905>

References

Ribaut JM, Hoisington DA, Deutsch JA, Jiang C, Gonzalez de Leon D (1996) Identification of quantitative trait loci under drought conditions in tropical maize.1. Flowering parameters and the anthesis-silking interval. *Theor Appl Genet* 92:905–914

Ribaut JM, Jiang C, Gonzalez de Leon D, Edmeades GO, Hoisington DA (1997) Identification of quantitative trait loci under drought conditions in tropical maize.2. Yield components and marker-assisted selection strategies. *Theor Appl Genet* 94:887–896

wheatChl

Field data for a wheat experiment in Chili

Description

A dataset containing field data from an experiment with wheat in Chili described in detail by Lado (2013). The experiment was performed in 2 locations in Chili with 2 different drought regimes in 2011 and 2012 for the first location and 1 trial in 2012 for the second location. For 384 genotypes 4 traits were measured in 2011 but in 2012 only grain yield was measured.

Usage

wheatChl

Format

A data.frame with 4000 rows and 11 columns:

rep replicate

bl block id

trt genotype

row row within the field

col column within the field

DH Days to Heading, the number of days from sowing till 50\ the spikes emerged

GY Grain Yield, in tons

NKS Number of Kernels per Spike, calculated from 25 randomly selected spikes per plot

TKW Thousand Kernel Weight, in grams, calculated from 25 randomly selected spikes per plot

trial trial, a combination of location and year

year year

Source

<https://www.g3journal.org/content/3/12/2105/>

References

Lado, Bettina, Ivan Matus, Alejandra Rodríguez, Luis Inostroza, Jesse Poland, François Belzile, Alejandro del Pozo, Martín Quincke, Marina Castro, and Jarislav von Zitzewitz. 2013. “Increased Genomic Prediction Accuracy in Wheat Breeding Through Spatial Adjustment of Field Trial Data.” *G3: Genes|Genomes|Genetics* 3 (12): 2105–14. doi:10.1534/g3.113.007807.

Index

* datasets

TDHeat05, [30](#)

TDMaize, [31](#)

wheatCh1, [32](#)

* functions for STA objects

plot.STA, [14](#)

report.STA, [20](#)

STAtoCross, [21](#)

STAtoTD, [23](#)

summary.STA, [24](#)

* functions for TD objects

getMeta, [11](#)

plot.TD, [16](#)

summary.TD, [25](#)

TD, [27](#)

addTD (TD), [27](#)

as.data.frame.extract, [2](#)

createTD (TD), [27](#)

dropTD (TD), [27](#)

extract, [3](#)

extractSTA, [3, 5, 23](#)

fitTD, [4, 6, 7, 9](#)

getMeta, [11, 18, 26, 30](#)

outlierSTA, [12](#)

plot.STA, [14, 21–23, 25](#)

plot.TD, [12, 16, 26, 27, 30](#)

PSANOVA, [8, 9](#)

read.cross, [22](#)

report, [19](#)

report.STA, [15, 19, 20, 22, 23, 25](#)

setMeta, [8](#)

setMeta (getMeta), [11](#)

STAtoCross, [15, 21, 21, 23, 25](#)

STAtoTD, [15, 21, 22, 23, 25](#)

summary.STA, [15, 21–23, 24](#)

summary.TD, [12, 18, 25, 27, 30](#)

TD, [3, 5, 7, 9, 12, 18, 26, 27](#)

TDHeat05, [30](#)

TDMaize, [31](#)

wheatCh1, [32](#)