Package 'fitPoly'

February 13, 2025

Type Package

Title Genotype Calling for Bi-Allelic Marker Assays

Version 4.0.0

Date 2025-02-10

Description Genotyping assays for bi-allelic markers (e.g. SNPs) produce signal intensities for the two alleles. 'fitPoly' assigns genotypes (allele dosages) to a collection of polyploid samples based on these signal intensities. 'fitPoly' replaces the older package 'fitTetra' that was limited (a.o.) to only tetraploid populations whereas 'fitPoly' accepts any ploidy level. Reference: Voorrips RE, Gort G, Vosman B (2011) <doi:10.1186/1471-2105-12-172>.
New functions added on conversion of data from SNP array software formats, drawing of XY-scatterplots with or without genotype colors, checking against expected F1 segregation patterns, comparing results from two different assays (probes) for the same SNP, recovery from a saveMarkerModels() crash.

License GPL-2

Depends R (>= 3.2.0)

Imports foreach, devEMF, doParallel, grDevices

Suggests knitr, rmarkdown

RoxygenNote 7.3.1

VignetteBuilder knitr

NeedsCompilation no

Encoding UTF-8

Author Roeland E. Voorrips [aut],
Gerrit Gort [aut],
Alejandro Therese Navarro [aut],
Giorgio Tumino [aut, cre]

Maintainer Giorgio Tumino <giorgio.tumino@wur.nl>

Repository CRAN

Date/Publication 2025-02-13 12:42:05 UTC

Contents

calcRstats *Calculate statistics of R per marker*

Description

Calculate the min, max, mean and 50 for each marker

Usage

```
calcRstats(datpoly, out)
```

Arguments

datpoly	a data frame in long format (input format for fitPoly) with at least columns MarkerName, SampleName, R
out	name of output file; if NA no file is written. File contains the same data as the return value

Details

The data.frame returned by this function is used as input for function selMarkers_byR

Value

a data frame with columns MarkerName, mean, min, q50, q95, max which are all statistics of the R values per marker

calcSegtypeInfo *Build a list of segregation types*

Description

For each possible segregation type in an F1 progeny with given parental ploidy (and ploidy2, if parent2 has a different ploidy than parent1) information is given on the segregation ratios, parental dosages and whether the segregation is expected under polysomic, disomic and/or mixed inheritance.

Usage

```
calcSegtypeInfo(ploidy, ploidy2=NULL)
```

Arguments

ploidy	The ploidy of parent 1 (must be even, 2 (diploid) or larger).
ploidy2	The ploidy of parent 2. If omitted (default=NULL) it is assumed to be equal to ploidy.

Details

The names of the segregation types consist of a short sequence of digits (and sometimes letters), an underscore and a final number. This is interpreted as follows, for example segtype 121_0: 121 means that there are three consecutive dosages in the F1 population with frequency ratios 1:2:1, and the 0 after the underscore means that the lowest of these dosages is nulliplex. So 121_0 means a segregation of 1 nulliplex : 2 simplex : 1 duplex. A monomorphic F1 (one single dosage) is indicated as e.g. 1_4 (only one dosage, the 4 after the underscore means that this is monomorphic quadruplex). If UPPERCASE letters occur in the first part of the name these are interpreted as additional digits with values of A=10 to Z=35, e.g. 18I81_0 means a segregation of 1:8:18:8:1 (using the I as 18), with the lowest dosage being nulliplex.
With higher ploidy levels higher numbers (above 35) may be required. In that case each unique ratio number above 35 is assigned a lowercase letter. E.g. one segregation type in octaploids is 9bcb9_2: a 9:48:82:48:9 segregation where the lowest dosage is duplex.
Segregation types with more than 5 dosage classes are considered "complex" and get codes like c7e_1 (again in octoploids): this means a complex type (the first c) with 7 dosage classes; the e means that this is the fifth type with 7 classes. Again the _1 means that the lowest dosage is simplex. It is always possible (and for all segtype names with lowercase letters it is necessary) to look up the actual segregation ratios in the intratio item of the segtype. For octoploid segtype c7e_1 this shows 0:1:18:69:104:69:18:1:0 (the two 0's mean that nulli- and octoplexes do not occur).

Value

A list with for each different segregation type (segtype) one item. The names of the items are the names of the segtypes. Each item is itself a list with components:

- freq: a vector of the ploidy+1 fractions of the dosages in the F1
- intratios: an integer vector with the ratios as the simplest integers
- expgeno: a vector with the dosages present in this segtype
- allfrq: the allele frequency of the dosage allele in the F1
- polysomic: boolean: does this segtype occur with polysomic inheritance?
- disomic: boolean: does this segtype occur with disomic inheritance?
- mixed: boolean: does this segtype occur with mixed inheritance (i.e. with polysomic inheritance in one parent and disomic inheritance in the other)?
- pardosage: integer matrix with 2 columns and as many rows as there are parental dosage combinations for this segtype; each row has one possible combination of dosages for parent 1 (1st column) and parent 2 (2nd column)
- parmode: logical matrix with 3 columns and the same number of rows as pardosage. The 3 columns are named polysomic, disomic and mixed and tell if this parental dosage combination will generate this segtype under polysomic, disomic and mixed inheritance

Examples

```
si4 <- calcSegtypeInfo(ploidy=4) # two 4x parents: a 4x F1 progeny
print(si4[["11_0"]])

si3 <- calcSegtypeInfo(ploidy=4, ploidy2=2) # a 4x and a diplo parent: a 3x progeny
print(si3[["11_0"]])
```

checkF1 *Identify the best-fitting F1 segregation types*

Description

For a given set of F1 and parental samples, this function finds the best-fitting segregation type. It can perform a dosage shift prior to selecting the segregation type.

Usage

```
checkF1(scores, parent1, parent2, F1, ancestors=character(0),
polysomic, disomic, mixed, ploidy, ploidy2, outfile,
critweight=c(1.0, 0.4, 0.4),
scores_long=TRUE, Pvalue_threshold=0.0001, fracInvalid_threshold=0.05,
fracNA_threshold=0.25, shiftmarkers, parentsScoredWithF1,
shiftParents=parentsScoredWithF1, showAll=FALSE, append_shf=FALSE)
```

Arguments

scores	A data frame as read from the scores file produced by function fitMarkers of package fitPoly (or a subset with at least columns MarkerName, SampleName and geno), or a data frame as returned by function scores2wide. In the first case (default) parameter scores_long must be TRUE, in the second case it must be FALSE.
parent1	character vector with the sample names of parent 1
parent2	character vector with the sample names of parent 2
F1	character vector with the sample names of the F1 individuals
ancestors	character vector with the sample names of any other ancestors or other samples of interest. The dosages of these samples will be shown in the output (shifted if shiftParents TRUE) but they are not used in the selection of the segregation type.
polysomic	if TRUE at least all polysomic segtypes are considered; if FALSE these are not specifically selected (but if e.g. disomic is TRUE, any polysomic segtypes that are also disomic will still be considered)
disomic	if TRUE at least all disomic segtypes are considered (see param polysomic)
mixed	if TRUE at least all mixed segtypes are considered (see param polysomic). A mixed segtype occurs when inheritance in one parent is polysomic (random chromosome pairing) and in the other parent disomic (fully preferential chromosome pairing)
ploidy	The ploidy of parent 1 (must be even, 2 (diploid) or larger).
ploidy2	The ploidy of parent 2. If omitted it is assumed to be equal to ploidy.
outfile	the tab-separated text file to write the output to; if NA a temporary file checkF1.tmp is created in the current working directory and deleted at end

`critweight` NA or a numeric vector containing the weights of three quality criteria; do not need to sum to 1. If NA, the output will not contain a column qall_weights. Else the weights specify how qall_weights will be calculated from quality parameters q1, q2 and q3.

`scores_long` TRUE if scores is in "long format", FALSE if it is in "wide format" (see parameter scores)

`Pvalue_threshold`
a minimum threshold value for the Pvalue of the bestParentfit segtype (with a smaller Pvalue the q1 quality parameter will be set to 0)

`fracInvalid_threshold`
a maximum threshold for the fracInvalid of the bestParentfit segtype (with a larger fraction of invalid dosages in the F1 the q1 quality parameter will be set to 0)

`fracNA_threshold`
a maximum threshold for the fraction of unscored F1 samples (with a larger fraction of unscored samples in the F1 the q3 quality parameter will be set to 0)

`shiftmarkers` if specified, shiftmarkers must be a data frame with columns MarkerName and shift; for the markernames that match exactly (upper/lowercase etc) those in scores, the dosages are increased by the amount specified in column shift, e.g. if shift is -1, dosages 2..ploidy are converted to 1..(ploidy-1) and dosage 0 is a combination of old dosages 0 and 1, for all samples. The segregation check is then performed with the shifted dosages. A shift=NA is allowed, these markers will not be shifted. The sets of markers in scores and shiftmarkers may be different, but markers may occur only once in shiftmarkers. A column shift is added at the end of the returned data frame.
If parameter shiftParents is TRUE, the parental and ancestor scores are shifted as the F1 scores, if FALSE they are not shifted.

`parentsScoredWithF1`
TRUE means parents are scored in the same experiment and the same fitPoly run as the F1, else FALSE. If TRUE, the fraction missing scores and conflicts in the parents tell something about the quality of the scoring. If FALSE (e.g. when the F1 is triploid and the parents are diploid and tetraploid) the quality of the F1 scores can be independent of that of the parents.
If not specified, TRUE is assumed if ploidy2 == ploidy and FALSE if ploidy2 != ploidy

`shiftParents` only used if parameter shiftmarkers is specified. If TRUE, apply the shifts also to the parental and ancestor scores. By default TRUE if parentsScoredWithF1 is TRUE, else FALSE

`showAll` (default FALSE) if TRUE, for each segtype 3 columns are added to the returned data frame with the frqInvalid, Pvalue and matchParents values for these segtype (see the description of the return value)

`append_shf` if TRUE and parameter shiftmarkers is specified, _shf is appended to all marker names where shift is not 0. This is not required for any of the functions in this package but may prevent duplicated marker names when using other software.

Details

For each marker is tested how well the different segregation types fit with the observed parental and F1 dosages. The results are summarized by columns bestParentfit (which is the best fitting segregation type, taking into account the F1 and parental dosages) and columns qall_mult and/or qall_weights (how good is the fit of the bestParentfit segtype: 0=bad, 1=good).
Column bestfit in the results gives the segtype best fitting the F1 segregation without taking account of the parents. This bestfit segtype is used by function correctDosages, which tests for possible "shifts" in the marker models. Both bestfit and bestParentfit are restricted by the parameters polysomic, disomic and mixed. Further they are restricted to segtypes that can only occur when the parental dosages are equal, if parent1 and parent2 list the same samples (but not if both are empty). In case the parents are not scored together with the F1 (e.g. if the F1 is triploid and the parents are diploid and tetraploid) the scores data frame should be edited to contain the parental as well as the F1 scores. In case the diploid and tetraploid parent are scored in the same run of function fitMarkers (package fitPoly) the diploid is initially scored as nulliplex-duplex-quadruplex (dosage 0, 2 or 4); that must be converted to the true diploid dosage scores (0, 1 or 2). Similar corrections are needed with other combinations, such as a diploid parent scored together with a hexaploid population etc.

Value

A data frame with one row per markers, with the following columns:

- m: the sequential number of the marker (as assigned by fitPoly)
- MarkerName: the name of the marker, with _shf appended if the marker is shifted and append_shf is TRUE
- parent1: consensus dosage score of the samples of parent 1
- parent2: consensus dosage score of the samples of parent 2
- F1_0 ... F1_<ploidy>: the number of F1 samples with dosage scores 0 ... <ploidy>
- F1_NA: the number of F1 samples with a missing dosage score
- sample names of parents and ancestors: the dosage scores for those samples
- bestfit: the best fitting segtype, considering only the F1 samples
- frqInvalid_bestfit: for the bestfit segtype, the frequency of F1 samples with a dosage score that is invalid (that should not occur). The frequency is calculated as the number of invalid samples divided by the number of non-NA samples
- Pvalue_bestfit: the chisquare test P-value for the observed distribution of dosage scores vs the expected fractions. For segtypes where only one dosage is expected (1_0, 1_1 etc) the binomial probability of the number of invalid scores is given, assuming an error rate of seg_invalidrate (hard-coded as 0.03)
- matchParent_bestfit: indication how the bestfit segtype matches the consensus dosages of parent 1 and 2: "Unknown"=both parental dosages unknown; "No"=one or both parental dosages known and conflicting with the segtype; "OneOK"= only one parental dosage known, not conflicting with the segtype; "Yes"=both parental dosages known and combination matching with the segtype. This score is initially assigned based on only high-confidence parental consensus scores; if low-confidence dosages are confirmed by the F1, the matchParent for (only) the selected segtype is updated, as are the parental consensus scores.
- bestParentfit: the best fitting segtype that does not conflict with the parental consensus scores

- frqInvalid_bestParentfit, Pvalue_bestParentfit, matchParent_bestParentfit: same as the corresponding columns for bestfit. Note that matchParent_bestParentfit cannot be "No".
- q1_segtypefit: a value from 0 (bad) to 1 (good), a measure of the fit of the bestParentfit segtype based on Pvalue, invalidP and whether bestfit is equal to bestParentfit
- q2_parents: a value from 0 (bad) to 1 (good), based either on the quality of the parental scores (the number of missing scores and of conflicting scores, if parentsScoredWithF1 is TRUE) or on matchParents (No=0, Unknown=0.65, OneOK=0.9, Yes=1, if parentsScoredWithF1 is FALSE)
- q3_fracscored: a value from 0 (bad) to 1 (good), based on the fraction of F1 samples that have a non-missing dosage score
- qall_mult: a value from 0 (bad) to 1 (good), a summary quality score equal to the product q1*q2*q3. Equal to 0 if any of these is 0, hence sensitive to thresholds; a natural selection criterion would be to accept all markers with qall_mult > 0
- qall_weights: a value from 0 (bad) to 1 (good), a weighted average of q1, q2 and q3, with weights as specified in parameter critweight. This column is present only if critweight is specified. In this case there is no "natural" threshold; a threshold for selection of markers must be obtained by inspecting XY-plots of markers over a range of qall_weights values
- shift: if shiftmarkers is specified a column shift is added with for all markers the applied shift (for the unshifted markers the shift value is 0)

qall_mult and/or qall_weights can be used to compare the quality of the SNPs within one analysis and one F1 population but not between analyses or between different F1 populations.
If parameter showAll is TRUE there are 3 additional columns for each segtype with names frqInvalid_<segtype>, Pvalue_<segtype> and matchParent_<segtype>; see the corresponding columns for bestfit for an explanation. These extra columns are inserted directly before the bestfit column.

`checkFilename` *Check if a file can be created*

Description

Checks if a file with that name can be created. If successful, any pre-existing file does not exist any more.

Usage

```
checkFilename(filename, overwrite=TRUE)
```

Arguments

`filename`	a file name with or without path. If the name contains a path the entire path must already exist, else the file cannot be created and the result is FALSE.
`overwrite`	if TRUE an attempt is made to delete any pre-existing file of that name; the function returns FALSE if the file is locked or the user has no rights to delete the file. If FALSE the function returns FALSE if any file of that name already exists, and no attempt is made to remove it

Value

TRUE if a new file can be created, else FALSE. If TRUE, no file of that name exists (any more)

`CodomMarker` *Function to fit a multiple mixture model to a vector of signal ratios of a single bi-allelic marker*

Description

This function fits a specified mixture model to a vector of signal ratios of multiple samples for a single bi-allelic marker. Returns a list with results from the fitted mixture model.

Usage

```
CodomMarker(y, ng, pop.parents=matrix(c(NA,NA), nrow=1),
pop=rep(1, length(y)), mutype=0, sdtype="sd.const", ptype=NA,
clus=TRUE, mu.start=NA, sd=rep(0.075, ng), p=NA,
maxiter=500, maxn.bin=200, nbin=200, plothist=TRUE, nbreaks=40,
maintitle=NULL, closeScreen=TRUE, fPinfo=NA)
```

Arguments

`y`	the vector of signal ratios (each value is from one sample, vector y contains the values for one marker). All values must be between 0 and 1 (inclusive), NAs are not allowed. The minimum length of y is 10*ng.
`ng`	the number of possible genotypes (mixture components) to be fitted: one more than the ploidy of the samples.
`pop.parents`	a matrix with 2 columns and 1 row per population; the cells contain the row numbers of the parental populations in case of an F1 and NA otherwise. The rows must be sorted such that all F1s occur above their parental populations. By default 1 row with elements NA, i.e. all samples belong to a single non-F1 population. If parameter pop is a factor or character vector, its levels or elements must correspond to the rownames of pop.parents.
`pop`	an integer vector specifying the population to which each sample in y belongs. All values must index rows of pop.parents. By default a vector of 1's, i.e. all samples belong to a single non-F1 population. Alternatively pop can be a factor or character vector of which the levels or elements match the rownames of pop.parents
`mutype`	an integer in 0:6; default 0. Describes how to fit the means of the components of the mixture model: with mutype=0 the means are not constrained, requiring ng degrees of freedom. With mutype in 1:6 the means are constrained based on the ng possible allele ratios according to one of 6 models; see Details.

`sdtype` one of "sd.const", "sd.free", "sd.fixed"; default "sd.const". Describes how to fit the standard deviations of the components of the mixture model: with "sd.const" all standard deviations (on the transformed scale) are equal (requiring 1 degree of freedom); with "sd.free" all standard deviations are fitted separately (ng d.f.); with "sd.fixed" all sd's ON THE TRANSFORMED SCALE are equal to parameter sd (0 d.f.).

`ptype` a character vector of length nrow(pop.parents) containing for each population one of "p.free", "p.fixed", "p.HW" or "p.F1". The default NA is interpreted as "p.F1" for F1 populations and "p.free" for all other populations; this is not necessarily the best choice for GWAS panels where "p.HW" may be more appropriate. Describes per population how to fit the mixing proportions of the components of the mixture model: with "p.free", the proportions are not constrained (and require ng-1 degrees of freedom per population); with "p.fixed" the proportions given in parameter p are fixed; with "p.HW" the proportions are calculated per population from an estimated allele frequency, requiring only 1 degree of freedom per population; with "p.F1" polysomic (auto-polyploid) F1 segregation ratios are calculated based on the fitted dosages of the F1 parents and require no extra d.f.

`clus` boolean. If TRUE, the initial means and standard deviations are based on a kmeans clustering of all samples into ng or fewer groups. If FALSE, the initial means are equally spaced on the transformed scale between the values corresponding to 0.02 and 0.98 on the original scale and the initial standard deviations are 0.075 on the transformed scale.

`mu.start` vector of ng values. If present, gives the start values of mu (the means of the mixture components) on the original (untransformed) scale. Must be strictly ascending (mu[i] > mu[i-1]) between 0 and 1 (inclusive). Overrides the start values determined by clus TRUE or FALSE.

`sd` vector of ng values. If present, gives the initial (or fixed, if sd.fixed is TRUE) values of sd (the standard deviations of the mixture components) ON THE TRANSFORMED SCALE. Overrides the start values determined by clus TRUE or FALSE.

`p` a matrix of nrow(pop.parents) rows and ng columns, each row summing to 1. If present, specifies the initial (or fixed, for populations where ptype is "p.fixed") mixing proportions of the mixture model components.

`maxiter` a single integer: the maximum number of times the nls function is called (0 = no limit, default=500).

`maxn.bin` a single integer, default=200: if the length of y is larger than max.nbin the values of y (after arcsine square root transformation) are binned (i.e. the range of y (0 to pi/2) is divided into nbin bins of equal width and the number of y values in each bin is used as the weight of the midpoints of each bin). This results in significant speed improvement with large numbers of samples without noticeable effects on model fitting.

`nbin` a single integer, default=200: the number of bins (see maxn.bin).

`plothist` if TRUE (default) a histogram of y is plotted with the fitted distributions superimposed

nbreaks	number of breaks (default 40) for plotting the histogram; does not have an effect on fitting the mixture model.
maintitle	string, used as title in the plotted histogram.
closeScreen	logical, only has an effect if plothist is TRUE. closeScreen should be TRUE (default) unless CodomMarker will plot on a device that is managed outside CodomMarker.
fPinfo	NA (default), for internal use only. Prevents unneeded checking and recalculation of input parameters when called from fitOneMarker.

Details

This function takes as input a vector of ratios of the signals of two alleles (a and b) at one genetic marker locus (ratios as b/(a+b)), one for each sample, and fits a mixture model with ng components (for a tetraploid species: ng=5 components representing the nulliplex, simplex, duplex, triplex and quadruplex genotypes). Ideally these signal ratios should reflect the possible allele ratios (for a tetraploid: 0, 0.25, 0.5, 0.75, 1) but in real life they show a continuous distribution with a number of more or less clearly defined peaks. The samples can represent multiple populations, each with their own segregation type (polysomic F1 ratios, Hardy-Weinberg ratios or free ratios). Multiple arguments specify what model to fit and with what values the iterative fitting process should start.
Parameter mutype determines how the means of the mixture model components are constrained based on the possible allele ratios, as follows

0 all means are fitted without restrictions (ng parameters)

1 a basic model assuming that both allele signals have a linear response to the allele dosage; one parameter for the ratio of the slopes of the two signal responses, and two parameters for the background levels (intercepts) of both signals (total 3 parameters)

2 as 1, but with the same background level for both signals (2 parameters)

3 as 1, with two parameters for a quadratic effect in the signal responses (5 parameters)

4 as 3, but with the same background level for both signals (4 parameters)

5 as 3, but with the same quadratic parameter for both signal responses (4 parameters)

6 as 5, but with the same background level for both signals (3 parameters)

Value

A list; if an error occurs the only list component is

message the error message

If no error occurs the list has the following components:

loglik the optimized log-likelihood

npar the number of fitted parameters

AIC Akaike's Information Criterion

BIC Bayesian Information Criterion

psi a list with components mu, sigma and p: mu and sigma each a vector of length ng with the means and standard deviations of the components of the fitted mixture model ON THE TRANSFORMED SCALE. p a matrix with one row per population and ng columns: the mixing proportions of the mixture components for each population

post a matrix of ng columns and length(y) rows; each row r gives the ng probabilities that y[r] belongs to the ng components

nobs the number of observations in y (excluding NA's)

iter the number of iterations

message an error message, "" if no error

back a list with components mu.back and sigma.back: each a vector of length ng with the means and standard deviations of the mixture model components back-transformed to the original scale

Examples

```
data(fitPoly_data)
mrkdat <- fitPoly_data$ploidy6$dat6x[fitPoly_data$ploidy6$dat6x$MarkerName == "mrk001",]

# hexaploid, without specified populations
cdm <- CodomMarker(mrkdat$ratio, ng=7)
names(cdm)

# hexaploid, with specified populations (4 F1 populations and a cultivar panel)
# first set the ptype for each population: p.F1 for F1 populations,
# p.HW for the panel, p.free for the F1 parents
ptype <- rep("p.HW", nrow(fitPoly_data$ploidy6$pop.parents))
ptype[!is.na(fitPoly_data$ploidy6$pop.parents[,1])] <- "p.F1"
ptype[unique(fitPoly_data$ploidy6$pop.parents)] <- "p.free" #all F1 parents
cdm <- CodomMarker(y=mrkdat$ratio, ng=7,
                   pop=fitPoly_data$ploidy6$pop,
                   pop.parents=fitPoly_data$ploidy6$pop.parents,
                   mutype=5, ptype=ptype)
```

`combineFiles` *Combine the X and Y intensity scores and the assigned dosage in one file*

Description

This function combines the X and Y intensity values from the fitPoly input with the geno (assigned dosage) from the fitPoly scores output. Useful for producing XY scatterplots with samples colored according to the assigned dosage.

Usage

```
combineFiles(XYdata, scores, controls=character(0))
```

Arguments

XYdata	data.frame with (at least) columns MarkerName, SampleName, X, Y (if present, R and ratio are also copied)
scores	data.frame with scores produced by the fitMarkers function of package fitPoly. It has columns MarkerName and SampleName that are subsets of MarkerName and SampleName in the XYdata, and at least a column geno
controls	a character vector of sample names. The geno (dosage) value of these samples is set to NA

Value

a dataframe with columns MarkerName, SampleName, X, Y (and R and ratio if these columns were present in XYdata), and geno, with all markers and samples from XYdata. The value of geno is set to NA for all MarkerName / sample combinations not present in scores, and for all samples in controls.

---

compareProbes *Compare and combine results from two probes for the same SNP*

Description

On Affymetrix Axiom arrays it is possible to have two probes interrogating the same SNP position. This function compares the dosage scores and checkF1 results of the two probes; if they are sufficiently similar a new marker is generated combining the results of the two probes. A dosage file with the data for the separate probes as well as the combined markers is written with the same format as writeDosagefile, and also a file summarizing the comparison results.

Usage

```
compareProbes(chk, scores,
probe.suffix=c("P","Q","R"), fracdiff.threshold=0.04,
parent1, parent2, F1, ancestors=character(0), other=character(0),
polysomic=TRUE, disomic=FALSE, mixed=FALSE,
ploidy, ploidy2, qall_flavor="qall_mult", shiftParents,
compfile, combscorefile)
```

Arguments

chk	data frame as returned by checkF1, or a subset with at least columns markername, parent1, parent2 (the consensus parental genotypes), the columns for the samples specified by parameters parent1, parent2 and ancestors, and bestParentfit, and containing only rows with selected markers. If a column with a name as specified by qall_flavor (see below) is present this will be written to file compfile, but it is not used: any selection of marker based on qall (or other) must have been made beforehand, and the rows for the unwanted markers must have been deleted from the chk data frame.

For each marker*probe combination there may be an unshifted version (shift==0), a shifted one (shift!=0), both, or neither.
If a column shift is present it will be used to shift the dosages (and their P-values with them). If some markernames end in "_shf" this part will be ignored, but the P and Q suffixes (or alternatives as specified by probe.suffix) are required to distinguish the two probes.

`scores` data frame as read from the scores file produced by function fitMarkers of package fitPoly, with at least columns MarkerName, SampleName, P0 .. P<ploidyF1> and geno (where <ploidyF1> is the ploidy of the F1, i.e. the average of parental ploidy and ploidy2).
If the F1 parents are scored separately, their rows should be added to the scores data.frame for the F1 samples. If their ploidy is different from the F1, the number of their P columns must be adjusted. The P data of the parents are not used, they may all be set to NA.

`probe.suffix` a 3-item character vector specifying the suffixes of the marker names that distinguish the two probes. The first two items identify the two probes; the third item is used to indicate a new marker combining the data from both probes. The three items must be different and have the same number of characters default is c("P","Q","R")

`fracdiff.threshold` if more than this fraction of F1 scores differs between probes, don't combine

`parent1` character vector with the sample names of parent 1

`parent2` character vector with the sample names of parent 2

`F1` character vector with the sample names of the F1 individuals

`ancestors` character vector with the sample names of any other ancestors

`other` other samples that should be treated like the F1

`polysomic` TRUE or FALSE; should be the same as used by checkF1 to calculate the chk data frame

`disomic` TRUE or FALSE; should be the same as used by checkF1 to calculate the chk data frame

`mixed` TRUE or FALSE; should be the same as used by checkF1 to calculate the chk data frame

`ploidy` the ploidy of parent 1 (must be even, 2 (diploid) or larger), and the same as used by checkF1 to calculate the chk data frame

`ploidy2` the ploidy of parent 2. If omitted it is assumed to be equal to ploidy. Should be the same as used by checkF1 to calculate the chk data frame

`qall_flavor` which quality parameter column must be shown in compfile, default "qall_mult". If no quality data are wanted, specify "".

`shiftParents` if there is a column shift in chk the F1 dosages will be shifted. If shiftParents is TRUE the parents and ancestors will be shifted together with the F1, if FALSE only the F1 will be shifted in that case.
If shiftParents is missing or NA it will be set to TRUE except if ploidy2 != ploidy: in that case this will result in an error (because it may be that the parents are not genotyped or scored together with the F1, the user should specify explicitly what to do)

compfile	filename for tab-separated text file summarizing the comparison results; if NA no file is written. For details of the contents see the return value, component compstat
combscorefile	filename for tab-separated text file with the dosages; if NA no file is written. For details of the contents see the return value, component combscores

Details

A combined marker is made in each case that a version of each of the two probe markers is present and they are sufficiently similar. This means that they have been assigned the same bestParentfit segregation type by checkF1, and that the frequency of conflicting scores over all samples is not more than fracdiff.threshold. The combined marker will have NA scores for individuals where both probe markers are missing, the one available score if it is scored for only one of the two probe markers or both scores are equal, and the score with the highest P-value if the scores for both probe markers are unequal.
Any single-probe markers in chk that do not have a bestParentfit segregation type are ignored and will not affect or appear in the output.

Value

A list with two components, compstat and combscores.
compstat is a data frame with columns:

- MarkerName: name of the SNP marker. If a column shift is present in data.frame chk, unshifted and shifted markers will get a "n" or "s" suffixed to the MarkerName
- segtypeP and segtypeQ: the segtype assigned by checkF1 to the first and second probe
- qallP and qallQ: the quality scores specified by parameter qall_flavor, assigned by checkF1 to the two probes
- countP and countQ: the number of versions of each of the probes (0, 1, or 2, depending on whether a shifted, unshifted or both versions were present)
- countR: the number of combinations made of versions of the two probe markers (one for each combination of a version of each of the two probe markers, if they match well enough - see details)

If the chk data frame contains a column shift, there are separate columns for the non-shifted and shifted P and Q probe markers (suffix Pn, Ps, Qn, Qs), and four columns for the R markers (suffix Rnn, Rns, Rsn, Rss where the first n/s indicates if the P was non-shifted or shifted and the second n/s for the Q probe. combscores is a data frame with columns:

- MarkerName: the name of the marker. If the chk data frame contains a column shift, the P and Q marker names are suffixed with n or s, and the R marker names with nn, ns, sn, ss as described above
- segtype: the segregation type
- parental and ancestor samples: the dosages of those samples
- parent1: the consensus dosage for parent1 as determined by checkF1
- parent2: the consensus dosage for parent2 as determined by checkF1
- F1 samples: the dosages for those samples
- other samples: the dosages for those samples

concatbatch *Construct the log, models and scores files from a set of batch files*

Description

Get the data saved by fitMarkers in batch files before it crashed and construct the log, models and score files just as fitMarkers would have done

Usage

```
concatbatch(batchfiles)
```

Arguments

batchfiles	a vector of batch file names as returned by getBatchFiles

Details

all batch files are assumed to contain a list with the same number of element, each either a character vector or a data.frame. This function concatenates the elements across the batch files.
This may be useful if fitMarkers has already been running a long time and then crashed. The partial (or perhaps complete) data can then be recovered from the saved batch files, so that only the remaining markers (if any) need to be processed afterwards.

Value

a list with the concatenated elements: character vectors and/or data.frames

convertStartmeans *A function to convert a set of mixture means from one ploidy to another*

Description

convertStartmeans takes a set of means at one ploidy level (e.g. the fitted means for a tetraploid data set) and uses them to generate a set of means for another ploidy level (e.g. as startmeans for fitting triploid data for the same markers).

Usage

```
convertStartmeans(ploidy, origmeans)
```

Arguments

ploidy	The ploidy to which the means must be converted.
origmeans	A data.frame with a first column MarkerName, followed by <oldploidy+1> columns (names are ignored) that contain the ratio means for dosages 0 to <oldploidy>. Column MarkerName may not contain missing values. On each row the other columns must either all contain NA, or only non-NA values between 0 and 1 in strictly ascending order.

Details

The new means are calculated by linear interpolation between the old means on the asin(sqrt(x)) transformed scale and back-transformed to the original scale; the new means for dosage 0 are equal to the old, and the new means for dosage <ploidy> are equal to the old means for dosage <oldploidy>.

Value

A data.frame like origmeans with the same column MarkerName, now followed by <ploidy+1> columns with the new means.

Examples

```
# means from tetraploid data set:
tetrameans <- data.frame(MarkerName=c("mrk1", "mrk2"), mu0=c(0.02, 0.0),
mu1=c(0.2, 0.25), mu2=c(0.3, 0.5), mu3=c(0.4, 0.75), mu4=c(0.6, 1.0))
# convert to means for triploid data set:
trimeans <- convertStartmeans(ploidy=3, origmeans=tetrameans)
tetrameans
trimeans
```

---

correctDosages *Check if dosage scores may have to be shifted*

Description

fitPoly sometimes uses a "shifted" model to assign dosage scores (e.g. all samples are assigned a dosage one higher than the true dosage). This happens mostly when there are only few dosages present among the samples. This function checks if a shift of +/-1 is possible.

Usage

```
correctDosages(chk, scores, parent1, parent2, ploidy, ploidy2,
polysomic=TRUE, disomic=FALSE, mixed=FALSE, parentsScoredWithF1,
absent.threshold=0.04, outfile=NA)
```

Arguments

chk	data frame returned by function checkF1 when called without shiftmarkers
scores	data.frame with scores as produced by the fitMarkers function of package fitPoly; at least columns MarkerName, SampleName and geno must be present, any other columns are ignored
parent1	character vector with names of the samples of parent 1
parent2	character vector with names of the samples of parent 2
ploidy	The ploidy of parent 1 (must be even, 2 (diploid) or larger).
ploidy2	The ploidy of parent 2. If omitted it is assumed to be equal to ploidy.
polysomic	if TRUE at least all polysomic segtypes are considered; if FALSE these are not specifically selected (but if e.g. disomic is TRUE, any polysomic segtypes that are also disomic will still be considered); same as used in the call to checkF1 that generated data.frame chk
disomic	if TRUE at least all disomic segtypes are considered (see param polysomic); same as used in the call to checkF1 that generated data.frame chk
mixed	if TRUE at least all mixed segtypes are considered (see param polysomic). A mixed segtype occurs when inheritance in one parent is polysomic (random chromosome pairing) and in the other parent disomic (fully preferential chromosome pairing); same as used in the call to checkF1 that generated data.frame chk
parentsScoredWithF1	single logical. TRUE means that parents are scored in the same experiment and the same fitPoly run as the F1, else FALSE. If missing and ploidy2==ploidy, TRUE is assumed. If FALSE, parental scores will not be shifted along with the F1 scores.
absent.threshold	the threshold for the fraction of ALL samples that has the dosage that is assumed to be absent due to mis-fitting of fitPoly; should be at least the assumed error rate of the fitPoly scoring assuming the fitted model is correct
outfile	file name to which the result is written. If NA no file is written.

Details

A shift of -1 (or +1) is proposed when (1) the fraction of all samples with dosage 0 (or ploidy) is below absent.threshold, (2) the bestfit (not bestParentfit!) segtype in chk has one empty dosage on the low (or high) side and more than one empty dosage at the high (or low) side, and (3) the shifted consensus parental dosages do not conflict with the shifted segregation type.
The returned data.frame (or a subset, e.g. based on the values in the fracNotOk and parNA columns) can serve as parameter shiftmarkers in a new call to checkF1.
Based on the quality scores assigned by checkF1 to the original and shifted versions of each marker the user can decide if either or both should be kept. A data.frame combining selected rows of the original and shifted versions of the checkF1 output (which may contain both a shifted and an unshifted version of some markers) can then be used as input to compareProbes or writeDosagefile.

Value

a data frame with columns

- markername
- segtype: the bestfit (not bestParentfit!) segtype from chk
- parent1, parent2: the consensus parental dosages; possibly low-confidence, so may be different from those reported in chk
- shift: -1, 0 or 1: the amount by which this marker should be shifted

The next fields are only calculated if shift is not 0:

- fracNotOk: the fraction of ALL samples that are in the dosage (0 or ploidy) that should be empty if the marker is indeed shifted.
- parNA: the number of parental dosages that is missing (0, 1 or 2)

`drawXYplots` *Draws a series of pages, each with 6 XY-plots showing allele signals and assigned dosages*

Description

Draws 6 XY-plots per page for a series of markers; each XY-plot is drawn by function XY_plot

Usage

```
drawXYplots(dat, markers=NA, out, genocol="grey", pch=1, cex=1,
sel.samples=as.character(unique(XYdat$SampleName)), omit.pch=".",
omit.col=c(rep("black", length(genocol)-1), "grey"), sample.groups=list(),
groups.col="black", groups.pch=1, groups.cex=1, groups.rnd=FALSE,
R.col="black", R.lty=1, drawRthresholds=FALSE,
Rthreshold.param=c(0.95, 0.5, 0), ploidy)
```

Arguments

`dat`	a data.frame with at least columns MarkerName, SampleName, X and Y; column geno (if present) is also used
`markers`	either a vector with names of markers to plot or a data.frame with at least column MarkerName containing the names of the markers to plot; default NA means all markers in dat. If markers is a data.frame that also has a column shift, the geno values in dat will be shifted accordingly. This allows to use (a selection from) the output of checkF1 (with or without parameter shiftmarkers) as input for drawXYplots.
`out`	base for filenames of output, will be extended with _pagenumber.png; may include a path, but the directory where the plot files are to be saved must already exist

genocol	a vector of color values to be used for plotting the sel.samples according on their geno (dosage) value; if only one value is given (default) all samples are plotted in that color
pch	the plot character to plot the sel.samples; default 1 is an open circle
cex	the relative size of the sample symbols
sel.samples	character vector (not a factor) with the names of the important samples: those that must be plotted in colors genocol and symbol pch
omit.pch	the plot character to use for the other samples, default a dot
omit.col	vector of two colors to use for the other samples; the first color is used for sample with a geno (dosage) value, the second color for unscored samples
sample.groups	a list specifying samples to be highlighted in a different color and/or symbol and/or size. For each group of sample the list has one vector of sample names; the list may also be empty
groups.col	a vector or color values, one for each item (vector of sample names) in sample.groups; recycled if shorter than sample.groups
groups.pch	a vector of plot symbols, one for each item in sample.groups; recycled if shorter than sample.groups
groups.cex	a vector of relative symbol sizes, one for each item in sample.groups; recycled if shorter than sample.groups
groups.rnd	FALSE (default) or TRUE. If FALSE, all samples in sample.groups are drawn in the order in which they appear in XYdat; if TRUE they are drawn in a random order. Note that the samples are never drawns in group order (except if the samples are already in group order in XYdat and groups.rnd=FALSE).
R.col	a vector of color values for drawing R thresholds, one for each value of Rthreshold.param; recycled if needed
R.lty	a vector of line types for drawing the R thresholds, one for each value of Rthreshold.param; recycled if needed
drawRthresholds	FALSE (default) or TRUE: whether R thresholds should be drawn
Rthreshold.param	either a list of vectors each of length 3 (for multiple R thresholds) or one vector of length 3 (for one R threshold). Each vector defines one R threshold and is based on a specified quantile of the distribution of R values for the current marker. The first number in each vector is the R quantile, the second is a number to multiply that R quantile with, and the third is the minimum value of the result. The default of c(0.95, 0.5, 0) means that 0.5 * the 95 always higher than 0, the minimum result). This is often a good cut-off value to discard samples, or to signal markers with many samples below that value.
ploidy	a single integer specifying the ploidy, only needed if dat contains a column geno

Value

The function produces a series of pages with plots and returns NULL

`expandUnknownParents` *Generate markers with all combinations of parental scores*

Description

For markers where the segregation type in the F1 is known but the parental consensus scores are missing, this function generates multiple versions of the marker, each with a different combination of parental scores matching the segregation and a different suffix to the marker name.

Usage

```
expandUnknownParents(scores, sep="@",
polysomic=TRUE, disomic=FALSE, mixed=FALSE,
ploidy, ploidy2, scorefile)
```

Arguments

`scores`	a data frame as returned by writeDosagefile, or the combscores item in the return value of compareProbes or removeRedundant
`sep`	a short string (one or more characters, default "@") to separate the original marker name from the consecutive letters (a, b, c etc) that identify the different versions of the marker. Markers that already have parental scores are not modified
`polysomic`	TRUE or FALSE; should be the same as used by checkF1
`disomic`	TRUE or FALSE; should be the same as used by checkF1
`mixed`	TRUE or FALSE; should be the same as used by checkF1
`ploidy`	the ploidy of parent 1 (must be even, 2 (diploid) or larger), and the same as used by checkF1
`ploidy2`	the ploidy of parent 2. If omitted it is assumed to be equal to ploidy. Should be the same as used by checkF1
`scorefile`	a filename to which the result is written; if NA no file is written

Value

A data frame with the same format and contents as parameter scores, with each marker where parental scores were missing expanded to multiple rows, one per parental dosage combination

`F1Dosages2Matrix` *Convert the F1 dosage scores data.frame or file to polymapR input matrix*

Description

Functions writeDosagefile, compareProbes and removeRedundant produce score files. This function reads such a file and converts the result to a data.frame as needed by the polymapR package.

Usage

```
F1Dosages2Matrix(dosages, outfile=NA, dec=".", sep=",")
```

Arguments

`dosages`	name of an F1 dosage scores file as produced by functions writeDosagefile, compareProbes or removeRedundant, or a data frame read from such a file
`outfile`	(path and) name of an output file, default NA means that no file is written. The file is written by function writeDatfile; normally using the default parameters but if file extension is 'csv', quote is set to TRUE and the specified dec and sep parameters are used
`dec`	character to use as decimal separator in the output file, default "."; only used if extension of outfile is 'csv' (otherwise ".")
`sep`	character to use as field separator in the output file, default ","; only used if extension of outfile is 'csv' (otherwise tab character)

Value

A matrix as needed by the polymapR package: one row per marker, one column per sample, containing integer scores (0 .. ploidy) or NAs. Row names are marker names, column names are sample names; the first two columns are the (consensus) scores of Parent1 and Parent2.
If outfile is not NA also an output file is written: if the extension is 'csv' a csv file is written with the specified decimal and field separators (default '.' and ',') and with row and column names quoted, else a tab-separated file.

`fitMarkers` *Function to fit mixture models for series of markers and save the results to files*

Description

This is the main function that calls fitOneMarker for a series of markers and saves the tabular, graphical and log output to files. Most of the arguments are identical to those of fitOneMarker and are directly passed through.

Usage

```
fitMarkers(ploidy, markers=NA, data, diplo=NULL, select=TRUE,
diploselect=TRUE, pop.parents=NULL, population=NULL, parentalPriors=NULL,
samplePriors=NULL, startmeans=NULL, maxiter=40, maxn.bin=200, nbin=200,
sd.threshold=0.1, p.threshold=0.9, call.threshold=0.6, peak.threshold=0.85,
try.HW=TRUE, dip.filter=1, sd.target=NA,
filePrefix, rdaFiles=FALSE, allModelsFile=FALSE,
plot="none", plot.type="png", ncores=1)
```

Arguments

ploidy	The ploidy level, 2 or higher: 2 for diploids, 3 for triploids etc.
markers	NA or a character or numeric vector specifying the markers to be fitted. If a character vector, names should match the MarkerName column of data; if numeric, the numbers index the markers based on the alphabetic order of the MarkerNames in data.
data	A data frame with the polyploid samples, with (at least) columns MarkerName, SampleName and ratio, where ratio is the Y-allele signal divided by the sum of the X- and Y-allele signals: ratio == Y/(X+Y)
diplo	NULL or a data frame like data, with the diploid samples and (a subset of) the same markers as in data. Genotypic scores for diploid samples are calculated according to the best-fitting model calculated for the polyploid samples and therefore may range from 0 (nulliplex) to <ploidy>, with the expected dosages 0 and <ploidy> for the homozygotes and <ploidy/2> for the heterozygotes. diplo can also be used for any other samples that need to be scored, but that should not affect the fitted models.
select	A logical vector, recycled if shorter than nrow(data): indicates which rows of data are to be used (default TRUE, i.e. keep all rows)
diploselect	A logical vector like select, matching diplo instead of data
pop.parents	NULL or a data.frame specifying the population structure. The data frame has 3 columns: the first containing population ID's, the 2nd and 3rd with the population ID's of the parents of these populations (if F1's) or NA (if not). The population ID's should match those in parameter population. If pop.parents is NULL all samples are considered to be in one population, and parameter population should be NULL (default).
population	NULL or a data.frame specifying to which population each sample belongs. The data frame has two columns, the first containing the SampleName (containing all SampleNames occurring in data), the second column containing population ID's that match pop.parents. In both columns NA values are not allowed. Parameters pop.parents and population should both be NULL (default) or both be specified.
parentalPriors	NULL or a data frame specifying the prior dosages for the parental populations. The data frame has one column MarkerName followed by one column for each F1 parental population. Column names (except first) are population ID's matching the parental populations in pop.parents. In case there is just one F1 population in pop.parents, it is possible to have two columns for both parental populations instead of one (allowing two specify two different prior dosages); in

that case both columns for each parent have the same caption. Each row specifies the priors for one marker. The contents of the data frame are dosages, as integers from 0 to <ploidy>; NA values are allowed.
Note: when reading the data frame with read.table or read.csv, set check.names=FALSE so column names (population ID's) are not changed.

`samplePriors` NULL or a data.frame specifying prior dosages for individual samples. The first column called MarkerName is followed by one column per sample; not all samples in data need to have a column here, only those samples for which prior dosages for one or more markers are available. Each row specifies the priors for one marker. The contents of the data frame are dosages, as integers from 0 to <ploidy>; NA values are allowed.
Note: when reading the data frame with read.table or read.csv, set check.names=FALSE so column names (population ID's) are not changed.

`startmeans` NULL or a data.frame specifying the prior means of the mixture distributions. The data frame has one column MarkerName, followed by <ploidy+1> columns with the prior ratio means on the original (untransformed) scale. Each row specifies the means for one marker in strictly ascending order (all means NA is allowed, but markers without start means can also be omitted).

`maxiter` A single integer, passed to CodomMarker, see there for explanation

`maxn.bin` A single integer, passed to CodomMarker, see there for explanation

`nbin` A single integer, passed to CodomMarker, see there for explanation

`sd.threshold` The maximum value allowed for the (constant) standard deviation of each peak on the arcsine - square root transformed scale, default 0.1. If the optimal model has a larger standard deviation the marker is rejected. Set to a large value (e.g. 1) to disable this filter.

`p.threshold` The minimum P-value required to assign a genotype (dosage) to a sample; default 0.9. If the P-value for all possible genotypes is less than p.threshold the sample is assigned genotype NA. Set to 1 to disable this filter.

`call.threshold` The minimum fraction of samples to have genotypes assigned ("called"); default 0.6. If under the optimal model the fraction of "called" samples is less than call.threshold the marker is rejected. Set to 0 to disable this filter.

`peak.threshold` The maximum allowed fraction of the scored samples that are in one peak; default 0.85. If any of the possible genotypes (peaks in the ratio histogram) contains more than peak.threshold of the samples the marker is rejected (because the remaining samples offers too little information for reliable model fitting).

`try.HW` Logical: if TRUE (default), try models with and without a constraint on the mixing proportions according to Hardy-Weinberg equilibrium ratios. If FALSE, only try models without this constraint. Even when the HW assumption is not applicable, setting try.HW to TRUE often still leads to a better model. For more details on how try.HW is used see the Details section of function fitOneMarker.

`dip.filter` if 1 (default), select best model only from models that do not have a dip (a lower peak surrounded by higher peaks: these are not expected under Hardy-Weinberg equilibrium or in cross progenies). If all fitted models have a dip still the best of these is selected. If 2, similar, but if all fitted models have a dip the marker is rejected. If 0, select best model among all fitted models, including those with a dip.

`sd.target` If the fitted standard deviation of the peaks on the transformed scale is larger than sd.target a penalty is given (see Details section of function fitOneMarker); default NA i.e. no penalty is given.

`filePrefix` partial file name, possibly including an absolute or relative file path. filePrefix must always be specified. All output files will have filePrefix prefixed to their name so it is clear they are all derived from the same call to fitMarkers. If filePrefix includes a file path all output files will be saved there; if a filePrefix is specified that does not include a a path the output will be saved in the working directory.

`rdaFiles` logical, default FALSE. The tabular output (scorefile, diploscorefile, modelfile, allmodelsfile) is saved as tab-separated text files with extension .dat or as an .RData file if this parameter is FALSE or TRUE respectively.

`allModelsFile` logical, default FALSE. If TRUE an allmodelsfile is saved with all models that have been tried for each marker; also the log file will contain a few lines for each marker. This information is mostly useful for debugging and locating problems.

`plot` String, "none" (default), "fitted" or "all". If "fitted" a plot of the best fitting model and the assigned genotypes is saved with filename <marker number><marker name>.<plot.type>, preceded by "rejected_" if the marker was rejected. If "all", small plots of all models are saved to files (8 per file) with filename <"plots"><marker number><marker name><pagenr>.<plot.type> in addition to the plot of the best fitting model.

`plot.type` String, "png" (default), "emf", "svg" or "pdf". Indicates format for saving the plots.

`ncores` The number of processor cores to use for parallel processing, default 1. Specifying more cores than available may cause problems. Note that the implementation under Windows involves duplicating the input data (under Linux that does not happen, nor under Windows if ncores=1), so if under Windows memory size is a problem it would be better to run several R instances simultaneously, each with ncores=1, each processing part of the data.

Details

fitMarkers calls fitOneMarker for all markers specified by parameter markers. The markers are processed in batches; the number of markers per batch is printed to the console when fitMarkers is started. If multiple cores are used the batches are processed in parallel.
During the processing a series of RData files (2 for each batch) is saved in the directory specified in filePrefix. At the end these are combined into the required output files and then deleted. If something goes wrong at any stage, the files for the completed batches are still available and can be combined manually, avoiding the need to re-run the process for the completed batches. The output files consist of:

- <filePrefix>.log: a logfile containing several lines listing the input parameters. If parameter allModelsFile is TRUE the logfile also contains several text lines per marker, corresponding to component "log" in the result of fitOneMarker
- <filePrefix>_scores.dat (or .RData) a file containing one line per polyploid sample for every marker that could be fitted, corresponding to component "scores" in the result of fitOneMarker

- <filePrefix>_diploscores.dat (or .RData) a file containing one line per diploid sample for every marker that could be fitted, corresponding to component "diploscores" in the result of fitOneMarker. This file is only produced if parameter diplo is not missing
- <filePrefix>_models.dat (or .RData) a file containing one line per marker, corresponding to component "modeldata" in the result of fitOneMarker: the selected model for each marker, with several statistics
- <filePrefix>_allmodels.dat (or .RData) as the models file, but containing all models fitted for each marker, not only the selected model, marker, corresponding to component "allmodeldata" in the result of fitOneMarker. This file is only produced if parameter allModelsFile is TRUE

Additionally, if plot != "none", plot files are generated in directory <filePrefix>_plots

Value

NULL. The result of fitMarkers is a set of output files.

Examples

```
# These examples run for a total of about 55 sec.
# All output files are saved in tempdir() and subdirectories of it.

data(fitPoly_data)

# tetraploid, with no populations and with sample prior dosages
fitMarkers(ploidy=4, data=fitPoly_data$ploidy4$dat4x,
                 samplePriors=fitPoly_data$ploidy4$sampPriors4x,
                 filePrefix=paste0(tempdir(),"/4xA"),
                 allModelsFile=TRUE,
                 plot="fitted")

# tetraploid, with specified populations and parental and sample prior dosages
fitMarkers(ploidy=4, data=fitPoly_data$ploidy4$dat4x,
                 population=fitPoly_data$ploidy4$pop4x,
                 pop.parents=fitPoly_data$ploidy4$pop.par4x,
                 parentalPriors=fitPoly_data$ploidy4$parPriors4x,
                 samplePriors=fitPoly_data$ploidy4$sampPriors4x,
                 filePrefix=paste0(tempdir(),"/4xB"),
                 allModelsFile=TRUE,
                 plot="fitted")

# hexaploid, no populations or prior information
fitMarkers(ploidy=6, data=fitPoly_data$ploidy6$dat6x,
                 filePrefix=paste0(tempdir(),"/6xA"),
                 allModelsFile=TRUE,
                 plot="fitted")

# hexaploid, with specified populations, prior dosages of parents and other samples
# and prior means of the mixture components
fitMarkers(ploidy=6, data=fitPoly_data$ploidy6$dat6x,
                 population=fitPoly_data$ploidy6$pop6x,
                 pop.parents=fitPoly_data$ploidy6$pop.par6x,
                 startmeans=fitPoly_data$ploidy6$startmeans6x,
```

```
parentalPriors=fitPoly_data$ploidy6$parPriors6x,
samplePriors=fitPoly_data$ploidy6$sampPriors6x,
filePrefix=paste0(tempdir(),"/6xB"),
plot="fitted")
```

`fitOneMarker` *Function to fit multiple mixture models to signal ratios of a single bi-allelic marker*

Description

This function takes a data frame with allele signal ratios for multiple bi-allelic markers and samples, and fits multiple mixture models to a selected marker. It returns a list, reporting on the performance of these models, selecting the best one based on the BIC criterion, optionally plotting results.

Usage

```
fitOneMarker(ploidy, marker, data, diplo=NULL, select=TRUE,
diploselect=TRUE, pop.parents=NULL, population=NULL, parentalPriors=NULL,
samplePriors=NULL, startmeans=NULL, maxiter=40, maxn.bin=200, nbin=200,
sd.threshold=0.1, p.threshold=0.9, call.threshold=0.6, peak.threshold=0.85,
try.HW=TRUE, dip.filter=1, sd.target=NA,
plot="none", plot.type="png", plot.dir, sMMinfo=NULL)
```

Arguments

`ploidy`	The ploidy level, 2 or higher: 2 for diploids, 3 for triploids etc.
`marker`	A marker name of number. Used to select the data for one marker, referring to the MarkerName column of parameter data. If a number, the number of the marker based on alphabetic order of the MarkerNames in data.
`data`	A data frame with the polyploid samples, with (at least) columns MarkerName, SampleName and ratio, where ratio is the Y-allele signal divided by the sum of the X- and Y-allele signals: ratio == Y/(X+Y)
`diplo`	NULL or a data frame like data, with the diploid samples and (a subset of) the same markers as in data. Genotypic scores for diploid samples are calculated according to the best-fitting model calculated for the polyploid samples and therefore may range from 0 (nulliplex) to <ploidy>, with the expected dosages 0 and <ploidy> for the homozygotes and <ploidy/2> for the heterozygotes. Note that diplo can also be used for any other samples that need to be scored, but that should not affect the fitted models.
`select`	A logical vector, recycled if shorter than nrow(data): indicates which rows of data are to be used (default TRUE, i.e. keep all rows)
`diploselect`	A logical vector like select, matching diplo instead of data

`pop.parents` NULL or a data.frame specifying the population structure. The data frame has 3 columns: the first containing population IDs, the 2nd and 3rd with the population IDs of the parents of these populations (if F1's) or NA (if not). The poopulation IDs should match those in parameter population. If pop.parents is NULL all samples are considered to be in one population, and parameter population should also be NULL (default).

`population` NULL or a data.frame specifying to which population each sample belongs. The data frame has two columns, the first containing the SampleName (containing all SampleNames occurring in data), the second column containing population IDs that match pop.parents. In both columns NA values are not allowed. Parameters pop.parents and population should both be NULL (default) or both be specified.

`parentalPriors` NULL or a data frame specifying the prior dosages for the parental populations. The data frame has one column MarkerName followed by one column for each F1 parental population. Column names (except first) are population IDs matching the parental populations in pop.parents. In case there is just one F1 population in pop.parents, it is possible to have two columns for both parental populations instead of one (allowing two specify two different prior dosages); in that case both columns for each parent have the same caption. Each row specifies the priors for one marker. The contents of the data frame are dosages, as integers from 0 to <ploidy>; NA values are allowed.
Note: when reading the data frame with read.table or read.csv, set check.names=FALSE so column names (population IDs) are not changed.

`samplePriors` NULL or a data.frame specifying prior dosages for individual samples. The first column called MarkerName is followed by one column per sample; not all samples in data need to have a column here, only those samples for which prior dosages for one or more markers are available. Each row specifies the priors for one marker. The contents of the data frame are dosages, as integers from 0 to <ploidy>; NA values are allowed.
Note: when reading the data frame with read.table or read.csv, set check.names=FALSE so column names (population IDs) are not changed.

`startmeans` NULL or a data.frame specifying the prior means of the mixture distributions. The data frame has one column MarkerName, followed by <ploidy+1> columns with the prior means on the original (untransformed) scale. Each row specifies the means for one marker in strictly ascending order (all means NA is allowed, but markers without start means can also be omitted).

`maxiter` A single integer, passed to CodomMarker, see there for explanation

`maxn.bin` A single integer, passed to CodomMarker, see there for explanation

`nbin` A single integer, passed to CodomMarker, see there for explanation

`sd.threshold` The maximum value allowed for the (constant) standard deviation of each peak on the arcsine - square root transformed scale, default 0.1. If the optimal model has a larger standard deviation the marker is rejected. Set to a large value (e.g. 1) to disable this filter.

`p.threshold` The minimum P-value required to assign a genotype (dosage) to a sample; default 0.99. If the P-value for all possible genotypes is less than p.threshold the sample is assigned genotype NA. Set to 1 to disable this filter.

`call.threshold` The minimum fraction of samples to have genotypes assigned ("called"); default 0.6. If under the optimal model the fraction of "called" samples is less than call.threshold the marker is rejected. Set to 0 to disable this filter.

`peak.threshold` The maximum allowed fraction of the scored samples that are in one peak; default 0.85. If any of the possible genotypes (peaks in the ratio histogram) contains more than peak.threshold of the samples the marker is rejected (because the remaining samples offers too little information for reliable model fitting).

`try.HW` Logical: if TRUE (default), try models with and without a constraint on the mixing proportions according to Hardy-Weinberg equilibrium ratios. If FALSE, only try models without this constraint. Even when the HW assumption is not applicable, setting try.HW to TRUE often still leads to a better model. For more details on how try.HW is used see the Details section.

`dip.filter` if 1 (default), select best model only from models that do not have a dip (a lower peak surrounded by higher peaks: these are not expected under Hardy-Weinberg equilibrium or in cross progenies). If all fitted models have a dip still the best of these is selected. If 2, similar, but if all fitted models have a dip the marker is rejected. If 0, select best model among all fitted models, including those with a dip.

`sd.target` If the fitted standard deviation of the peaks on the transformed scale is larger than sd.target a penalty is given (see Details); default NA i.e. no penalty is given.

`plot` String, "none" (default), "fitted" or "all". If "fitted" a plot of the best fitting model and the assigned genotypes is saved with filename <marker number><marker name>.<plot.type>, preceded by "rejected_" if the marker was rejected. If "all", small plots of all models are saved to files (8 per file) with filename <"plots"><marker number><A..F><marker name>.<plot.type> in addition to the plot of the best fitting model.

`plot.type` String, "png" (default), "emf", "svg" or "pdf". Indicates format for saving the plots.

`plot.dir` String, the directory where to save the plot files. Must be specified if plot is not "none". Set this to "" to save plot files in the current working directory.

`sMMinfo` NULL (default), for internal use only. Prevents unneeded checking and recalculation of input parameters when called from fitMarkers.

Details

fitOneMarker fits a series of mixture models for the given marker by repeatedly calling CodomMarker and selects the optimal one. The initial models vary according to the values of try.HW, pop.parents, parentalPriors, samplePriors and startmeans:

- no pop.parents, try.HW FALSE: 4 models with different constraints on the means (different or equal X and Y background signal, ratio a linear or quadratic function of dosage), no restrictions on the mixing proportions (the fractions of samples in each dosage peak)
- no pop.parents, try.HW TRUE: The previous 4 models are fitted and also 4 models with the same restrictions on the means and the mixing proportions restricted to Hardy-Weinberg ratios (assuming polysomic inheritance)

- pop.parents specified, no parentalPriors / samplePriors / startmeans, try.HW FALSE: 4 models are fitted with the same restrictions on the means as above, but with different restrictions on the mixing proportions for each population: no restriction on parental populations, none on accession panels, polysomic F1 segregation ratios on F1 populations. Additionally 4 models are fitted with all samples considered as one population, with the same 4 models for the means and no restrictions on mixing proportions.
- pop.parents specified, no parentalPriors / samplePriors / startmeans, try.HW TRUE: 4 models are fitted with the same restrictions on the means as above, but with different restrictions on the mixing proportions for each population: no restriction on parental populations, HW-ratios for accession panels, polysomic F1 segregation ratios on F1 populations. Additionally 4 models are fitted with all samples considered as one population, with the same 4 models for the means and mixing proportions according to HW ratios.
- pop.parents and parentalPriors specified, try.HW FALSE: 4 models are fitted with the same restrictions on the means as above, but with different restrictions on the mixing proportions for each population: no restriction on parental populations and the accession panels, polysomic F1 segregation ratios on F1 populations ignoring the parental priors. Additionally 4 models are fitted with the same restrictions on the means and mixing proportions of the accession panels, but where the mixing proportions of the parental populations are set to (almost) 1 for the prior dosage and (almost) 0 for all other dosages, and those for the F1 populations to the polysomic segregation ratios expected for the parental priors.
- pop.parents and parentalPriors specified, try.HW TRUE: same as with try.HW FALSE, except that the mixing proportions of accession panels are now restricted to HW ratios.
- if parentalPriors and/or samplePriors are specified, these and the signal ratios of the corresponding samples are (also) used to estimate starting values of the mixture component means in the EM algorithm. Alternatively startmeans can be specified directly.

Because convergence to the optimal solution often fails, the models are fitted with several start values for the <ploidy+1> means of the mixture distributions: (1) based on initial clustering of the ratios, (2) based on a uniform distribition from 0.02 to pi/2-0.02 on the asin(sqrt(x)) scale, and (3) if startmeans are specified or can be calculated from samplePriors and/or parentalPriors these are used for a third set of model fits.
The main difference between parentalPriors and samplePriors is that parentalPriors are treated as fixed (and if both parents of an F1 population have priors, the F1 segregation is also fixed) while samplePriors are only used to calculate starting ratio means for each dosage. Depending on the confidence the user has in the prior dosages of the parents they can be supplied as parentalPriors or samplePriors. In some cases an additional fit is performed with a modified set of initial means.
An optimal model is selected based on the Bayesian Information Criterium (BIC), which takes into account the Log-Likelihood and the number of fitted parameters of the models. If sd.target is specified and the standard deviation of the mixture model components is larger than this target a penalty is applied, making is less likely that that model is selected.
The plots consist of one histogram per (non-parent) population showing the frequency distribution of the signal ratios of the samples in that population. The fitted model is shown in green (density and means), and for F1 populations the samples of parent 1 and 2 are shown as red and blue triangles.
If diplodata are present, a histogram for the diploid samples is plotted in the top histogram (diploid bars are narrower and gray). The diploid bars are scaled so the maximum bar is half the maximum polyploid bar. At the bottom of the plot for the fitted model a rug plot shows the scores of each sample, while the bottom (red) samples are unscored.

Value

A list with components:

log A character vector with the lines of the log text.

modeldata A data frame as allmodeldata (see below) with only the one row with data on the selected model.

allmodeldata A data frame with for each tried model one row with the marker number, marker name, number of samples and (if the marker is not rejected) data of the fitted model (see below).

scores A data frame with the name and data for all samples (including NA's for the samples that were not selected, see parameter select), with columns:
marker (the sequential number of the marker (based on alphabetic order of the marker names in data)
MarkerName
SampleName
ratio (the given ratio from parameter data)
P0 .. P<ploidy> (the probabilities that this sample belongs to each of the <ploidy+1> mixture components)
maxgeno (0..ploidy, the genotype = mixture component with the highest P value)
maxP (the P value for this genotype)
geno (the assigned genotype number: same as maxgeno, or NA if maxP < p.threshold).

diploscores A data frame like scores for the samples in the data frame supplied with argument diplo. If diplo is NA also diploscores will be NA.

The modeldata and allmodeldata data frames present data on a fitted model. modeldata presents data on the selected model; allmodeldata lists all attempted models. Both data frames contain the following columns:

marker the sequential number of the marker (based on alphabetic order of the marker names in data)

MarkerName the name of the marker

m the number of the fitted model

model the type of the fitted model. Possible values are "b1", "b2", "b1,q", "b2,q", each by itself or followed by "HW" or "pop". The first 4 refer to the models for the mixture means: b1 and b2 indicate 1 or 2 parameters for signal background, q indicates that a quadratic term in the signal response was fitted as well. HW and pop refer to the restrictions on the mixing proportions: HW indicates that the mixing proportions were constrained according to Hardy-Weinberg equilibrium ratios in case of only one population, pop indicates that multiple populations were fitted (see Details section). For more details see Voorrips et al (2011), doi:10.1186/1471-2105-12-172.

nsamp the number of samples for this marker for which select==TRUE, i.e. the number on which the call rate is based.

nsel the number of these samples that have a non-NA ratio value

npar the number of free parameters fitted

iter the number of iterations to reach convergence

dip whether the model had a dip (a smaller peak surrounded by larger peaks): 0=no, 1=yes

LL the log-likelihood of the model

AIC Akaike's Information Criterion

BIC Bayesian Information Criterion

selcrit the selection criterion; the model with the lowest selcrit is selected. If argument sd.target is NA selcrit is equal to BIC, else selcrit is larger than BIC if the standard deviation of the mixture components is larger than sd.target; see Details for details.

minsepar a measure of the minimum peak separation. Each difference of the means of two successive mixture components is divided by the average of the standard deviations of the two components. The minimum of the values is reported. All calculations are on the arcsine-square root transformed scale.

meanP For each sample the maximum probability of belonging to any mixture component is calculated. The average of these P values is reported in meanP

P80 .. P99 the fraction of samples that have a probability of at least 0.80 .. 0.99 to belong to one of the mixture components (by default a level of 0.99 is required to assign a genotype score to a sample)

muact0 .. the actual means of the samples in each of the mixture components for dosages 0 .. <ploidy> on the transformed scale

sdact0 .. the actual standard deviations of the samples in each of the mixture components for dosages 0 .. <ploidy> on the transformed scale

mutrans0 .. the means of the mixture components for dosages 0 .. <ploidy> on the transformed scale

sdtrans0 .. the standard deviations of the mixture components for dosages 0 .. <ploidy> on the transformed scale

P0 .. the mixing proportions of the mixture components for dosages 0 to <ploidy>. If multiple populations are specified there are two possibilities: (1) the specified population structure is used in the current model; then for each population the mixing proportions are given as <npop> sequences of <ploidy+1> fractions, or (2) the population structure is ignored for the current model, the mixing proportions are given in the first sequence of <ploidy+1> fractions and all following sequences are filled with NA. The the item names are adapted to have the population names between the P and the dosage

mu0 .. the model means of the <ploidy+1> mixture components back-transformed to the original scale

sd0 .. the model standard deviations of the <ploidy+1> mixture components back-transformed to the original scale

message if no model was fitted or the model was rejected, the reason is reported here

Examples

```
# These examples run for a total of about 9 sec.

data(fitPoly_data)

# triploid, no specified populations
fp <- fitOneMarker(ploidy=3, marker="mrk039",
```

```
                    data=fitPoly_data$ploidy3$dat3x)

# tetraploid, specified populations
# plot of the fitted model saved in tempdir()
fp <- fitOneMarker(ploidy=4, marker=2,
                    data=fitPoly_data$ploidy4$dat4x,
                    population=fitPoly_data$ploidy4$pop4x,
                    pop.parents=fitPoly_data$ploidy4$pop.par4x,
                    plot="fitted",
                    plot.dir=paste0(tempdir(),"/fpPlots4x"))

# hexaploid, specified populations, start values for means,
# plot of the fitted model saved in tempdir()
fp <- fitOneMarker(ploidy=6, marker=1,
                    data=fitPoly_data$ploidy6$dat6x,
                    population=fitPoly_data$ploidy6$pop6x,
                    pop.parents=fitPoly_data$ploidy6$pop.par6x,
                    startmeans=fitPoly_data$ploidy6$startmeans6x,
                    plot="fitted", plot.dir=paste0(tempdir(),"/fpPlots6x"))
```

`fitPoly` *fitPoly: a package for assigning dosage scores based on SNP array data*

Description

fitPoly (an evolved version of package fitTetra) fits mixture models to the distribution of intensity ratios Y/(X+Y) (where X and Y are the intensities of the signals produced by the A and B alleles of bi-allelic markers) and uses these to assign genotypes (dosages). The main differences compared with fitTetra are that it can handle any ploidy level, and multiple populations that can be either F1 populations (and their parents) or panels of accessions. There are also improvements in accuracy, speed and the possibility to use prior dosage information.

`fitPolyTools` *fitPolyTools: a package with functions related to package fitPoly*

Description

Package fitPolyTools offers some functions for preparing and selecting input data for package fitPoly, and for inspection and analysis of polyploid dosage calls as produced by fitPoly. These analyses focus on the segregation of dosages in an F1 progeny of two parents. The functions can be grouped as follows:

data preparation functions

Functions readFullDataTable and readAxiomSummary convert files generated by Illumina's GenomeStudio and Affymetrix Power Tools software to the input format for fitPoly::fitPoly and fitPoly::fitMarkers.

data visualization functions

Functions XYplot and drawXYplots draw scatterplots of the samples for one marker at a time, possibly with colors indicating the assigned dosages and/or highlighting specific samples. Function combineFiles combines X and Y columns from one file or data.frame with column geno from another.

segregation type functions

The functions calcSegtypeInfo, selSegtypeInfo and segtypeInfoSummary produce information on the possible F1 segregation types. These functions are primarily developed for use in other functions but are also available for users.

identification of the probable segregation types

Function checkF1 gives for all markers the most likely (best fitting) segregation type, based on the F1 and the parental samples. Function correctDosages checks for possibly "shifted" markers (e.g. all dosages scored one lower or higher than the true dosage). Function checkF1 can apply shifts when specified.

combining data from two probes for the same SNP

On Affymetrix Axiom SNP arrays it is possible to have the same SNP interrogated from both sides by two different probes. Function compareProbes compares the results from both probes and produces a new, merged marker if they are sufficiently similar. It also writes the results to a data file suitable for linkage mapping functions (package polymapR).
Function removeRedundant takes the result of compareProbes; if a merged marker is present it removes the corresponding separate probe markers.

producing a data file for mapping

Function writeDosagefile produces a data file suitable for linkage mapping functions (package polymapR). This data file has the same format as that produced by compareProbes and removeRedundant (see previous section).
Function F1Dosages2Matrix reads such a dosages file and returns a matrix in the format required by the polymapR package.

reformatting of fitPoly scores file

Function scores2wide reformats the scores file produced by the fitMarkers function of the fitPoly package to "wide" format.

recovery functions

Functions getBatchfiles and concatbatch allow to recover results from already processed batches of markers, in case fitMarkers crashes.

general tools

These tools may be useful by themselves and are therefore made available too. They include leftstr and rightstr: two functions that return substrings from the left or right side, and function checkFilename which tests if a file with the name can be created. Functions readDatfile and writeDatfile are wrappers for read.table and write.table with different default options, suitable for the file formats expected or produced by functions in this package oand fitPoly.

`fitPoly_data` *Small fitPoly input datasets for testing and examples*

Description

A list with small datasets of four different ploidy levels for testing and examples

Usage

```
data(fitPoly_data)
```

Details

list fitPoly_data contains the following items:

- ploidy2: a diploid dataset with only the SNP array signal ratios
- ploidy3: a triploid dataset with in addition to the signal ratios two data.frames specifying the population structure
- ploidy4: a tetraploid dataset similar to the triploid dataset and additionally prior dosage information of the F1 population parents and of a few other samples
- ploidy6: a hexaploid dataset similar to the tetraploid dataset and additionally the 7 starting means for some of the markers

Each of the items contains one or more elements, postfixed by 2x, 3x, 4x or 6x depending on the ploidy:

- dat: a data.frame with at least columns MarkerName, SampleName and ratio with the signal ratios to be analyzed
- pop: a data.frame with columns SampleName and Population, specifying the population to which each sample belongs
- pop.par: a matrix specifying what are the parents of each population (if any)
- parPriors: a data.frame specifying prior known allele dosages for the F1 parents
- sampPriors: a data.frame specifying the prior known dosages for other samples
- startmeans: a data.frame with prior known means for the (ploidy+1) mixture model components

In addition the ploidy6 component has elements pop and pop.parents (no suffix) which are equivalent to pop6x and pop.par6x, in the format required by function codomMarker.

get.genocol *Generate a set of genotype (dosage) colors for XY-plots*

Description

For each of the dosages 0 to ploidy a color is generated, and an additional one for samples with no dosage assigned.

Usage

```
get.genocol(ploidy, pastel=TRUE)
```

Arguments

ploidy	any ploidy level >= 2
pastel	if TRUE (default) light (pastel) colors are generated, else more intense colors

Details

the colors range from red (first item, dosage 0) via blue (dosage ploidy/2) to green (dosage ploidy); the color for missing dosage is grey

Value

A vector of <ploidy + 2> color values; items 1 .. ploidy+1 for dosage 0 .. ploidy and item ploidy+2 for missing dosages

getBatchFiles *get the names of all batch files present*

Description

This function gets the names of all batch files already produced by fitMarkers; used to recover data after a crash.

Usage

```
getBatchFiles(filePrefix, set)
```

Arguments

filePrefix	the same filePrefix as used in the fitMarkers call
set	one of "models" or "scores"

Value

a vector of batch file names, either the scores set or the models set

leftstr *Get substrings from the lefthand side*

Description

Get substrings from the lefthand side

Usage

```
leftstr(x, n)
```

Arguments

x	a character vector (or something having an as.character method)
n	a single number: if n>=0, the leftmost n characters of each element of x are selected, if n<0 the (-n) rightmost characters are omitted

Value

character vector with leftside substrings of x

listSegtypes *conversion of segtype code to F1 segregation ratios*

Description

Produce a matrix with the F1 segregation ratios (as integers) for all segregation types for the given ploidy

Usage

```
listSegtypes(ploidy, ploidy2=NULL)
```

Arguments

ploidy	the ploidy of parent 1 of the F1, or of the F1 itself if ploidy2 is NULL
ploidy2	the ploidy of parent 2 of the F1

Details

If ploidy2 is not NULL, ploidy and ploidy2 are the ploidy levels of the two parents, and both must be even. If ploidy2 is NULL, ploidy is the ploidy of the F1; if even, both parents are assumed to have the same ploidy; if odd, parent 1 and parent 2 are asumed to have ploidy-1 and ploidy+1. This function calls calcSegtypeInfo and is a convenience function; it is not used by any other functions. For more information, including parental dosages for each segregation, use calcSegtypeInfo and segtypeInfoSummary.

Value

a matrix with one row for each segregation type and one column for each possible F1 dosage, with the integer ratios of the dosages for each segregation type.

`makeFitPolyFiles` *Make input files for fitPoly containing only selected rows and columns*

Description

Remove entire markers and ratios for samples within markers from fitPoly inputfiles if their R levels are below some thresholds; also keep only the MarkerName, SampleName, X, Y, R and ratio columns

Usage

```
makeFitPolyFiles(datpoly, datdiplo=NA, out,
filetype=c("dat", "RData")[2], Rquantile=0.95, marker.threshold,
Rthreshold.param=c(0.95, 0.5, 0))
```

Arguments

`datpoly`	data.frame as produced by readFullDataTable, readAxiomSummary or splitNrenameSamples, with at least columns MarkerName, SampleName, R and ratio
`datdiplo`	data.frame as produced by splitNrenameSamples, with at least columns MarkerName, SampleName, R and ratio, containing the data for the diploid samples; or NA (default)
`out`	output files will be named out + _poly.RData and _diplo.RData (or .dat if filetype is set to "dat"). If NA no files are written
`filetype`	"RData" (default) or "dat": the format to save the data in
`Rquantile`	a value between 0 and 1: the R (= X + Y) quantile on which selection of markers is based
`marker.threshold`	the minimum value of the Rquantile for a marker to be selected
`Rthreshold.param`	a vector of length 3, defining an R threshold for selecting samples within each marker. The first number is the R quantile, the second is a number to multiply that R quantile with, and the third is the minimum value of the result. The default of c(0.95, 0.5, 0) means that 0.5 * the 95 (which is always higher than 0, the minimum result). This is often a good cut-off value to discard samples within a marker.

Details

All rows for markers not meeting the marker.threshold for the R quantile are removed from the original data.frames. For samples within the remaining markers where the R value is below the marker-specific threshold the ratio is set to NA, the row for that sample is not removed. The same thresholds are applied to datpoly and datdiplo.

Value

a list with two elements: datpoly is a filtered version of parameter datpoly, and if param datdiplo was specified the second element of the returned list is datdiplo: a filtered version of parameter datdiplo. Either or both may have 0 rows left after filtering. If no datdiplo was specified the second element of the list is NA.
If parameter out is specified the element(s) of this list are also saved as files.

`readAxiomSummary` *convert an Affymetrix AxiomCT1.summary file to the import format for fitPoly*

Description

An Affymetrix AxiomCT1.summary.txt file in wide format (samples side-by-side) is converted to a fitPoly input file in long format

Usage

```
readAxiomSummary(AXdata, markergroups=list(),
out, filetype=c("dat", "RData")[2])
```

Arguments

`AXdata`	name of an AxiomCT1.summary.txt tab-separated text file exported from Affymetrix Power Tools (APT); or a data.frame in the same format: the file may start with a number of comment lines (starting with "#") followed by a header line of which the first caption is "probeset_id" and all other captions are sample names, all ending with ".CEL". For each marker there are two lines, with probeset_id ending in "-A" and "-B"; the two lines for each marker are consecutive. In addition there may be any number of control markers, for each of which there is only one line, with probeset_id ending in "-A".
`markergroups`	a list with character vectors of marker names, or integer vectors of marker numbers in file order. If the data set is large, the conversion to long format may exceed memory limits. In these cases the data can be split into marker groups that are converted separately and each saved to a file (or returned as list items, but if that is possible splitting into groups would not be needed)
`out`	the name of an output file (without extension). If a list of markergroups is given, out must be a valid file name (without extension); in that case multiple output files are created with filenames in which the list element numbers are appended to out. If no markergroups are specified out may also be set to "" or NA; in that case no file is created and the converted data are only returned as function result. If out is not "" or NA, then also a file <out>_meanR.dat is saved with for all samples their mean R value and number of missing data (over all markers)
`filetype`	either "dat" or "RData" (default): the former produces tab-separated text files, the latter saves RData files with the converted data in a data frame with name "dat"

Details

The wide-format input is converted to a long-format form with columns MarkerName, SampleName, X, Y, R (= X + Y) and ratio (= Y / R). The X and Y signal intensities are obtained from the <marker>-A and <marker>-B rows in the input data, R and ratio are calculated from these values.

Value

If no markergroups are specified, a data.frame is returned with columns MarkerName, SampleName, X, Y, R (= X + Y), ratio (= Y / (X+Y)).
If a list of markergroups is specified the function result is NULL and the converted data.frames are only saved as files.
If the saved files are RData files, they all contain one data.frame named "dat".

---

readDatfile *User-friendly wrapper for read.table*

Description

A wrapper for read.table that has default parameter values for reading tab-separated files as used in packages fitPoly and fitPolyTools

Usage

```
readDatfile(file, header=TRUE, sep="\t", check.names=FALSE, ...)
```

Arguments

file	the name of the file which the data are to be read from
header	a logical value indicating whether the file contains the names of the data.frame columns as its first line
sep	the field separator character
check.names	logical. If FALSE (default), column names are not checked. This is important if column names are the names of samples, markers etc that may not be syntactically valid variable names. If TRUE then the names of the variables in the data frame are checked to ensure that they are syntactically valid variable names. If necessary they are adjusted (by make.names) so that they are, and also to ensure that there are no duplicates
...	Further arguments to be passed to read.table

Value

A data.frame containing a representation of the data in the file

readFullDataTable *convert a GenomeStudio FullDataTable file to the import format for fitPoly*

Description

A GenomeStudio file in wide format (samples side-by-side) is converted to a fitPoly input file in long format

Usage

```
readFullDataTable(filename, rawXY=FALSE,
markergroups=list(), out, filetype=c("dat","RData")[2])
```

Arguments

filename	name of a FullDataTable tab-separated text file exported from Illumina's GenomeStudio. The file must contain a column "Name" with the marker names, and for each sample a pair of columns "sample.X" and "sample.Y" if rawXY is FALSE, or "sample.X raw" and "sample.Y raw" (note the space) if rawXY is TRUE. Further columns may be present but are not read.
rawXY	if FALSE (default) the normalized .X and .Y columns are read; if TRUE the "raw" columns (.X raw and .Y raw) are read instead.
markergroups	a list with character vectors of marker names, or integer vectors of marker numbers in file order. If the data set is large, the conversion to long format may exceed memory limits. In these cases the data can be split into marker groups that are converted separately and each saved to a separate file. If the list is empty (default) all markers are converted as one block.
out	the name of an output file (without extension). If a list of markergroups is given, out must be a valid file name (without extension); in that case multiple output files are created with filenames in which the list element numbers are appended to out. If no markergroups are specified out may also be set to "" or NA; in that case no file is created and the converted data are only returned as function result. If out is not "" or NA, then also a file <out>_meanR.dat is saved with for all samples their mean R value and number of missing data (over all markers)
filetype	either "dat" or "RData" (default): the former produces tab-separated text files, the latter saves RData files with the converted data in a data frame with name "dat".

Details

The wide-format input is converted to a long-format form with columns MarkerName, SampleName, X, Y, R (= X + Y) and ratio (= Y / R). The X and Y signal intensities are obtained from the <sample>.X and <sample>.Y columns in the input data (or from the <sample>.X raw and <sample>.Y raw columns if rawXY is TRUE). R and ratio are calculated from these values and not read from the input data.

Value

If no markergroups are specified, a data.frame is returned with columns MarkerName, SampleName, X and Y (also if raw data are read, the column names are X and Y), R (= X + Y), ratio (= Y / (X+Y)).
If a list of markergroups is specified the function result is NULL and the converted data.frames are only saved as files.
If the saved files are RData files, they all contain one data.frame named "dat".

`removeRedundant` *Remove redundant single-probe markers*

Description

If for a SNP both probes produced very similar results, function compareProbes generated an extra marker combining the results from both probes. In that case the original single-probe markers are redundant and this function removes those data from the compstat and combscores data frames.

Usage

```
removeRedundant(compstat, combscores, compfile, combscorefile)
```

Arguments

`compstat`	a data.frame as returned by compareProbes in the compstat item of the return value and in the compfile
`combscores`	a data.frame as returned by compareProbes in the
`compfile`	a filename to which the compstat part of the results is written; if NA no file is written
`combscorefile`	a filename to which the compscores part of the results is written; if NA no file is written

Value

A list with two components, compstat and combscores. These are identical to the parameters compstat and combscores, with the redundant single-probe markers removed.For their contents see function compareProbes.

rightstr *Get substrings from the righthand side*

Description

Get substrings from the righthand side

Usage

```
rightstr(x, n)
```

Arguments

x	a character vector (or something having an as.character method)
n	a single number: if n>=0, the rightmost n characters of each element of x are selected, if n<0 the (-n) leftmost characters are omitted

Value

character vector with rightside substrings of x

samplestats *Statistics for each F1 sample over all markers*

Description

Tabulate for each F1 sample in how many markers it has a missing, invalid or valid score.

Usage

```
samplestats(chk, scores, F1, qall_flavor="qall_mult",
qall_threshold=0, ploidy, ploidy2, scores_long=TRUE)
```

Arguments

chk	a data frame as returned by checkF1
scores	A data frame as read from the scores file produced by function fitMarkers of package fitPoly (or a subset with at least columns MarkerName, SampleName and geno), or a data frame as returned by function scores2wide. In the first case (default) parameter scores_long must be TRUE, in the second case it must be FALSE.
F1	character vector with the sample names of the F1 individuals
qall_flavor	which quality parameter column must be shown in compfile, default "qall_mult". If no quality data are wanted, specify "".

qall_threshold	only markers with a qall score > qall.threshold are included in the tabulation
ploidy	the ploidy of parent 1 (must be even, 2 (diploid) or larger), and the same as used by checkF1 to calculate the chk data frame
ploidy2	the ploidy of parent 2. If omitted it is assumed to be equal to ploidy. Should be the same as used by checkF1 to calculate the chk data frame
scores_long	TRUE if scores is in "long format", FALSE if it is in "wide format" (see parameter scores)

Value

A matrix with samples in rows and 3 columns: missing, invalid, valid, giving for each sample the number of markers where it has a missing, invalid or valid dosage score

---

saveMarkerModels *DEPRECATED: Function to fit mixture models for series of markers and save the results to files*

Description

This is the old name of the function ‘fitMarkers()‘, the main function of ‘fitPoly‘. It is kept only for backwards-compatibility. The only difference between the two is the default parameter of the genotyping probability threshold ‘p.threshold‘, it is 0.99 in ‘saveMarkerModels()‘ (as was originally set) and it is 0.9 in the current ‘fitMarkers()‘.

Usage

```
saveMarkerModels(..., p.threshold = 0.99)
```

Arguments

...	All parameters allowed in the function ‘fitMarkers()‘. For a full description see the help of that function.
p.threshold	The minimum P-value required to assign a genotype (dosage) to a sample; default 0.99 (very stringent). If the P-value for all possible genotypes is less than p.threshold the sample is assigned genotype NA. Set to 1 to disable this filter.

Value

See ‘fitMarkers()‘ documentation for a full description.

scores *A data set with dosage scores generated by fitPoly*

Description

A data frame as saved by function fitMarkers in package fitPoly

Usage

```
data(scores)
```

Details

scores contains the following columns:

- marker: numeric, the sequential number of the markers in the fitMarkers input file
- MarkerName: factor, names of the markers (SNPs)
- SampleName: factor names of the individuals / DNA samples
- ratio: numeric, the ratios of the Y signal to the total signal intensity
- P0 ... P4: numeric, the probabilities that the observed signal ratio reflects a true Y allele dosage of 0 ... 4, according to the selected mixture model
- maxgeno: numeric, the dosage with the maximum probability
- maxP: numeric, the maximum probability, i.e. the probability of maxgeno
- geno: numeric, equal to maxgeno if maxP was above the specified threshold (p.threshold=0.99), else NA

scores2wide *Convert a scores file from long to wide format*

Description

The fitMarkers function of package fitPoly returns a scores file in "long" format: one row for each MarkerName / SampleName combination. This function creates a file in "wide" format (samples in columns, markers in rows) containing the data from the geno column of the scores file. Faster, less memory-expensive and easier to use than reshape.

Usage

```
scores2wide(scores, outfile=NA)
```

Arguments

scores	a data frame read from the scorefile output of saveMarkeModels in package fitPoly. The order of the markers within each sample does not have to be the same or v.v., there is no requirement for any ordering of scores. However if scores is not ordered by MarkerName and SampleName and/or not all markers are present for all samples (or v.v.) this results in a slower processing.
outfile	a file name to which the result is written. If NA no file is written.

Value

A data frame with column names marker, MarkerName and all sample names, and one row per marker. Marker is the sequential number of the marker as reported in the scores data frame.

---

segtypeInfoSummary *Summarize the segtypeInfo list*

Description

From a list of segregation types as produced by calcSegtypeInfo or selSegtypeInfo, produce a data frame that only lists the parental dosage combinations for each segtype and whether these produce the segtype under polysomic, disomic and/or mixed inheritance. Useful to quickly look up which segtypes match a given parental dosage combination.

Usage

```
segtypeInfoSummary(segtypeInfo)
```

Arguments

segtypeInfo	a list as returned by calcSegtypeInfo or selSegtypeInfo

Value

A data frame summarizing the segtypeInfo list, with columns:

- segtype: the name of the segregation type (see details of calcSegtypeInfo)
- segtypenr: the sequential number of the segtype in parameter segtypeInfo
- parent1, parent2: dosages of the two parents
- par.poly, par.di, par.mixed: whether these parental dosages produce this segtype under polysomic, disomic and/or mixed inheritance

selMarkers_byR *Select markers at specified R levels*

Description

Select markers based on their R statistics, for studying the relation between R level and marker quality

Usage

```
selMarkers_byR(Rstats, Rlevels, mrkperlevel=1, stat="q95")
```

Arguments

Rstats	a data frame as returned by calcRstats
Rlevels	a vector of R levels in increasing order (preferably from the minimum to the maximum of stat over all markers in Rstats)
mrkperlevel	number, default 1: the number of markers to select at each of the levels in Rlevels
stat	the name of one of the statistics columns in Rstats

Details

The return value of this function is intended to be used for studying the relation between the marker quality and the value of the chosen R statistic, e.g. by drawing XY-plots of each of the selected markers. By finding a suitable threshold for the R statistic bas markers could be excluded from evaluation by fitPoly, saving time because bad markers take the longest to be scored and are then often rejected anyway.

Value

a selection from Rstats, in ascending order of column stat. The first <mrkperlevel> markers with stat above each Rlevel are returned; duplicated selections are removed (so the number of returned markers may be less than length(Rlevels) * mrkperlevel)

selMarkers_qall *Sample markers at several qall levels*

Description

In order to get an idea of the quality of markers with different levels of qall, this function samples marker numbers at a specified set of qall values.

Usage

```
selMarkers_qall(chk, qall_levels, mrkperlevel=1,
qall_flavor="qall_mult")
```

Arguments

chk	a data frame as returned by checkF1
qall_levels	a numeric vector with the levels of the quality parameter (qall_mult or qall_weights, see parameter qall_flavo) at which markers should be selected
mrkperlevel	the number of markers to select per level of qall
qall_flavor	which qall to use: by default qall_mult, also qall_weights is possible

Value

A subset of data frame chk with selected rows, in ascending order of the qall_flavor column. The first <mrkperlevel> markers with qall >= each qall.level are returned; duplicated selections are removed (so the number of returned markers may be less than length(qall_levels) * mrkperlevel)

---

selSegtypeInfo *Restrict a list of segregation types to specified inheritance modes*

Description

From a list of segregation types as produced by calcSegtypeInfo, this function selects only those segtypes that occur with polysomic, disomic and/or mixed inheritance if the corresponding parameters are set to TRUE, and from those segtypes only the parental dosages with the same restrictions are retained.

Usage

```
selSegtypeInfo(segtypeInfo, polysomic, disomic, mixed,
selfing=FALSE)
```

Arguments

segtypeInfo	a list as returned by calcSegtypeInfo
polysomic	If TRUE all segtypes with poly TRUE are retained, and from those segtypes all parental dosage combinations with parmode[,1] TRUE
disomic	If TRUE all segtypes with di TRUE are retained, and from those segtypes all parental dosage combinations with parmode[,2] TRUE
mixed	If TRUE all segtypes with mixed TRUE are retained, and from those segtypes all parental dosage combinations with parmode[,3] TRUE
selfing	if TRUE only segtypes are retained that can be the result of a selfing, i.e. both parental dosages equal (although it allows disomic segtypes where the parental distribution of alleles over the subgenomes is different, e.g. aa/bb x ab/ab -> 121_1); default FALSE

Value

A list like segtypeInfo, modified as specified by parameters polysomic, disomic and mixed

splitNrenameSamples *Rename samples from array codes to user codes, and split diploid from polyploid samples*

Description

After conversion of array data to fitPoly format, this function replaces the array codes for the samples to user codes, and it splits the data into a part for diploid and one for polyploid samples if both are present.

Usage

```
splitNrenameSamples(dat, sampletable, SampleID, CustomerID,
Ploidy=NULL, out, filetype=c("dat","RData")[2])
```

Arguments

dat	a data frame in "long" format as returned by readFullDataTable or readAxiomSummary
sampletable	a data frame with at least the columns for SampleID, CustomerID and Ploidy (these columns may have any name, specified by the next parameters)
SampleID	the title of the column in sampletable containing the codes for the samples in the array file - there must be a one-to-one relation between the array codes and the user codes!
CustomerID	the title of the column in sampletable containing the user codes for the samples - there must be a one-to-one relation between the array codes and the user codes!
Ploidy	the title of the column in sampletable containing the ploidy of the samples; default NULL means no split based on ploidy must be done. If not NULL, for each different ploidy a separate data.frame will be created and samples with ploidy missing or "" will be omitted from all files.
out	the base filename of the output files; to this will be appended the the ploidy level(s) if Ploidy is not NULL, and an extension .dat or .RData, depending on filetype). Setting out to NA or "" results in no output file(s) being created
filetype	either "dat" or "RData" (default): the former produces tab- separated text files (one for each element of the return value), the latter saves one RData file containing a list named dat, identical to the return value

Value

if Ploidy not specified, the original dat data.frame with substituted SampleNames; if Ploidy specified a list with one element for each different value in the Ploidy column; the names of the elements are then these ploidy values. Each element is a subset of the original dat data.frame containing only the rows with samples of that ploidy, and with substituted SampleNames

`writeDatfile` *User-friendly wrapper for write.table*

Description

A wrapper for write.table that has default parameter values for writing tab-separated files as used in packages fitPoly and fitPolyTools

Usage

```
writeDatfile(x, file, quote=FALSE, sep="\t", na="",
row.names=FALSE, col.names=TRUE, logical_01=FALSE, ...)
```

Arguments

`x`	the object to be written, preferably a matrix or data frame. If not, it is attempted to coerce x to a data frame.
`file`	either a character string naming a file or a connection open for writing. "" indicates output to the console.
`quote`	a logical value (TRUE or FALSE) or a numeric vector. If TRUE, any character or factor columns will be surrounded by double quotes. If a numeric vector, its elements are taken as the indices of columns to quote. In both cases, row and column names are quoted if they are written. If FALSE (default), nothing is quoted.
`sep`	the field separator string. Values within each row of x are separated by this string.
`na`	the string to use for missing values in the data
`row.names`	either a logical value indicating whether the row names of x are to be written along with x, or a character vector of row names to be written.
`col.names`	either a logical value indicating whether the column names of x are to be written along with x, or a character vector of column names to be written.
`logical_01`	FALSE (default) or TRUE; if TRUE, logical columns of x are converted to 0/1/NA numeric values before writing the file
`...`	Further arguments to be passed to write.table

writeDosagefile *Write a file with segregation types and dosage scores*

Description

Write a file with for each marker the segregation type and the dosage scores of the parental and ancestor samples, the parental consensus dosages and the F1 samples.

Usage

```
writeDosagefile(chk, scores, parent1, parent2, F1,
ancestors=character(0), other=character(0),
polysomic=TRUE, disomic=FALSE, mixed=FALSE,
ploidy, ploidy2, shiftParents, scorefile)
```

Arguments

chk	data frame as returned by checkF1
scores	data frame as read from the scores file produced by function fitMarkers of package fitPoly (or a subset with at least columns MarkerName, SampleName and geno)
parent1	character vector with the sample names of parent 1
parent2	character vector with the sample names of parent 2
F1	character vector with the sample names of the F1 individuals
ancestors	character vector with the sample names of any other ancestors
other	other samples that should be treated like the F1
polysomic	TRUE or FALSE; should be the same as used by checkF1 to calculate the chk data frame
disomic	TRUE or FALSE; should be the same as used by checkF1 to calculate the chk data frame
mixed	TRUE or FALSE; should be the same as used by checkF1 to calculate the chk data frame
ploidy	the ploidy of parent 1 (must be even, 2 (diploid) or larger) and the same as used by checkF1 to calculate the chk data frame
ploidy2	the ploidy of parent 2. If omitted it is assumed to be equal to ploidy. Should be the same as used by checkF1 to calculate the chk data frame
shiftParents	if there is a column shift in chk the F1 dosages will be shifted. If shiftParents is TRUE the parents and ancestors will be shifted together with the F1, if FALSE only the F1 will be shifted in that case. If shiftParents is missing or NA it will be set to TRUE except if ploidy2 != ploidy: in that case this will result in an error (because it may be that the parents are not genotyped or scored together with the F1, the user should specify explicitly what to do)
scorefile	filename for tab-separated text file with the dosages,; if NA no file is written. For details of the contents see the return value

Value

A data frame with columns:

- MarkerName: the name of the marker
- segtype: the segregation type
- parental and ancestor samples: the dosages of those samples
- parent1: the consensus dosage for parent1 as determined by checkF1
- parent2: the consensus dosage for parent2 as determined by checkF1
- F1 samples: the dosages for those samples
- other samples: the dosages for those samples

XYdat *A data set containing SNP array data*

Description

A data frame containing SNP array signal intensity data

Usage

```
data(XYdat)
```

Details

XYdat contains the following columns:

- MarkerName: factor, names of the markers (SNPs)
- SampleName: factor names of the individuals / DNA samples
- X: numeric, the signal intensities of the X allele
- Y: numeric, the signal intensities of the Y allele
- R: numeric, the total signal intensities; R = X + Y
- ratio: numeric, the ratios of the Y signal to the total signal intensity; ratio = Y / R

XY_plot *Draws an XY-plot showing allele signals and assigned dosages*

Description

Draws an XY-plot for one markers showing the X and Y signals of each sample and their assigned dosages

Usage

```
XY_plot(title="", XYdat, shift=0, ploidy=NULL,
genocol="grey", pch=1, cex=1,
sel.samples=as.character(unique(XYdat$SampleName)), omit.pch=".",
omit.col=NULL, sample.groups=list(), groups.col="black", groups.pch=1,
groups.cex=1, groups.rnd=FALSE, R.thresholds=NA, R.col="black", R.lty=1)
```

Arguments

title	the main title above the plot
XYdat	a data frame with at least columns SampleName, X and Y; column geno (if present) is also used. Contains data for all samples, one SNP. geno is the dosage (0 .. <ploidy>), with NA or <ploidy+1> for missing dosage info
shift	a single integer, default 0: by how much should the geno be shifted?
ploidy	a single integer specifying the ploidy; default NA, only needed if there is a column geno in XYdat
genocol	a vector of color values to be used for plotting the sel.samples according on their geno (dosage) value; if only one value is given (default) all samples are plotted in that color
pch	the plot character to plot the sel.samples; default 1 is an open circle
cex	the relative size of the sample symbols
sel.samples	character vector (not a factor) with the names of the important samples: those that must be plotted in colors genocol and symbol pch
omit.pch	the plot character to use for the other samples, default a dot
omit.col	vector of two colors to use for the other samples; the first color is used for sample with a geno (dosage) value, the second color for unscored samples; recycled if needed, with NA (default) the non-selected samples are not plotted
sample.groups	a list specifying samples to be highlighted in a different color and/or symbol and/or size. For each group of samples the list has one vector of sample names; the list may also be empty
groups.col	a vector or color values, one for each item (vector of sample names) in sample.groups; recycled if shorter than sample.groups
groups.pch	a vector of plot symbols, one for each item in sample.groups; recycled if shorter than sample.groups

groups.cex	a vector of relative symbol sizes, one for each item in sample.groups; recycled if shorter than sample.groups
groups.rnd	FALSE (default) or TRUE. If FALSE, all samples in sample.groups are drawn in the order in which they appear in XYdat; if TRUE they are drawn in a random order. Note that the samples are never drawns in group order (except is the samples are already in group order in XYdat and groups.rnd=FALSE).
R.thresholds	a vector of thresholds for R to plot; is NA (default) no R thresholds are plotted
R.col	a vector of color values for drawing R thresholds, one for each value of R.thresholds; recycled if needed
R.lty	a vector of line types for drawing the R thresholds, one for each value of R.thresholds; recycled if needed

Value

The function produces an XY-plot and returns NULL

Index