# Package 'fitPoly'

March 16, 2018

**Type** Package

**Title** Genotype Calling for Bi-Allelic Marker Assays

**Version** 3.0.0

**Date** 2018-03-14

**Author** Roeland E. Voorrips and Gerrit Gort

**Maintainer** Roeland E. Voorrips <roeland.voorrips@wur.nl>

**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>.

**License** GPL-2

**Depends** R (>= 3.2.0)

**Imports** foreach

**Suggests** devEMF, doParallel, grDevices

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-03-16 19:01:05 UTC

## R topics documented:

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)
```

---

# 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 <old-ploidy>.

## 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
```

---

# 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=0.99, 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 saveMarkerModels. |

## 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.

`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.

`saveMarkerModels` *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

```
saveMarkerModels(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=0.99, 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.<br>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 0.99. 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 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 saveMarkerModels. 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

saveMarkerModels 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 saveMarkerModels 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 saveMarkerModels 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
saveMarkerModels(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
saveMarkerModels(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
saveMarkerModels(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
saveMarkerModels(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")
```

# Index