Package: groupedSurv (via r-universe)

October 25, 2026

Type Package

Title Efficient Estimation of Grouped Survival Models Using the Exact Likelihood Function

Version 1.0.5.1

Date 2023-09-28

Author Jiaxing Lin [aut], Alexander Sibley [aut], Tracy Truong [aut], Kouros Owzar [aut], Zhiguo Li [aut], Layne Rogers [ctb], Yu Jiang [ctb], Janice McCarthy [ctb], Andrew Allen [ctb]

Maintainer Alexander Sibley <dcibioinformatics@duke.edu>

Description These 'Rcpp'-based functions compute the efficient score statistics for grouped time-to-event data (Prentice and Gloeckler, 1978), with the optional inclusion of baseline covariates. Functions for estimating the parameter of interest and nuisance parameters, including baseline hazards, using maximum likelihood are also provided. A parallel set of functions allow for the incorporation of family structure of related individuals (e.g., trios). Note that the current implementation of the frailty model (Ripatti and Palmgren, 2000) is sensitive to departures from model assumptions, and should be considered experimental. For these data, the exact proportional-hazards-model-based likelihood is computed by evaluating multiple variable integration. The integration is accomplished using the 'Cuba' library (Hahn, 2005), and the source files are included in this package. The maximization process is carried out using Brent's algorithm, with the C++ code file from John Burkardt and John Denker (Brent, 2002).

License GPL (>= 2)

Imports Rcpp (>= 0.12.4), doParallel, parallel, foreach, qvalue

LinkingTo Rcpp, RcppEigen, BH

Suggests knitr, snplist, BEDMatrix

VignetteBuilder knitr

BuildVignettes yes

NeedsCompilation yes

RoxygenNote 6.0.1

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

Date/Publication 2023-09-28 22:30:39 UTC

RemoteUrl https://github.com/cran/groupedSurv

RemoteRef HEAD

RemoteSha dea1f25933e6250160c0943bc66422848c13db5b

Contents

groupedSurv-package 2
alphaEstFam 3
betaEst 4
betaEstFam 5
geneStat 6
groupedSurv 9
groupedSurvFam 11
PvalueFam 13
thetaEst 14
varEstFam 15
varLLFam 16

Index **18**

`groupedSurv-package` *Efficient Estimation of Grouped Survival Models Using the Exact Likelihood Function*

Description

These `Rcpp`-based functions compute the efficient score statistics for grouped time-to-event data (Prentice and Gloeckler, 1978), with the optional inclusion of baseline covariates. Functions for estimating the parameter of interest and nuisance parameters, including baseline hazards, using maximum likelihood are also provided. A parallel set of functions allow for the incorporation of family structure of related individuals (e.g., trios). Note that the current implementation of the frailty model (Ripatti and Palmgren, 2000) is sensitive to departures from model assumptions, and should be considered experimental. For these data, the exact proportional-hazards-model-based likelihood is computed by evaluating multiple variable integration. The integration is accomplished using the `Cuhre` algorithm from the `Cuba` library (Hahn, 2005), and the source files of the `Cuhre` function are included in this package. The maximization process is carried out using Brent's algorithm, with the `C++` code file from John Burkardt and John Denker (Brent, 2002).

License: GPL (>= 2)

Details

Package:	groupedSurv
Type:	Package
Version:	1.0.5.1
Date:	2023-09-28
License:	GPL-3

Please refer to the individual function documentation or the included vignette for more information. The package vignette serves as a tutorial for using this package.

Author(s)

Jiaxing Lin <jiaxing.lin@duke.edu>, Alexander Sibley, Tracy Truong, Kouros Owzar, Zhiguo Li; Contributors: Yu Jiang, Janice McCarthy, Andrew Allen

References

Prentice, R.L. and Gloeckler, L.A. (1978). Regression analysis of grouped survival data with application to breast cancer data. Biometrics, 34:1, 57-67.
Ripatti, S. and Palmgren, J. (2000). Estimation of Multivariate Frailty Models Using Penalized Partial Likelihood. Biometrics, 56, 1016-1022.
Hahn, T. (2005). Cuba-a library for multidimensional numerical integration, Computer Physics Communications, 168, 78-95.
Brent, R. (2002). Algorithms for Minimization without Derivatives. Dover, ISBN 0-486-41998-3

`alphaEstFam` *Estimate the Interval Baseline Survival Rates for the Grouped Failure Time Model*

Description

A method to estimate the baseline survival rate for each time interval for a grouped failure time model. The estimation is conducted under the null hypothesis, i.e., that there is no effect from the variable of interest, and is naive to family structure.

Usage

```
alphaEstFam(gtime, delta)
```

Arguments

`gtime`	Vector of observed survival times for each sample.
`delta`	Vector of event indicators for each sample: 1 indicates observed event, 0 indicates censored.

Value

Vector of estimates of the baseline survival rates for each time interval.

Examples

```
# Generate dummy data
gtime <- c(1, 3, 3, 2, 1, 1, 2, 3, 1)
delta <- c(1, 0, 1, 1, 1, 0, 1, 0, 1)

#res   <- alphaEstFam(gtime, delta)
#res
```

---

betaEst *Estimate the Parameter of Interest for the Grouped Survival Model*

Description

A method to estimate the parameter of interest given the baseline survival rates for each time interval and optional covariate nuisance parameters for a grouped survival model. The function supports zero, single, or multiple covariates as dictated by the input data.

Usage

```
betaEst(x, Z=NULL, alpha, theta=NULL, gtime, delta)
```

Arguments

x	Vector of numeric variable of interest for each sample.
Z	Optional `data.frame` or `matrix` of numeric covariate values for each sample.
alpha	Vector of baseline survival rates for each time interval.
theta	Optional vector of estimated nuisance parameters for the covariates.
gtime	Vector of observed survival times for each sample.
delta	Vector of event indicators for each sample: 1 indicates observed event, 0 indicates censored.

Value

Scalar estimate of the parameter of interest.

Examples

```
# Generate dummy data
x     <- c(1, 0, 1, 1, 1, 0, 1, 0, 1)
cov1  <- c(1, 2, 2, 2, 1, 1, 0, 1, 1)
cov2  <- c(2, 2, 1, 0, 1, 0, 1, 1, 0.5)
Z     <- cbind(cov1, cov2)
gtime <- c(1, 3, 3, 2, 1, 1, 2, 3, 1)
delta <- c(1, 0, 1, 1, 1, 0, 1, 0, 1)

#theta <- thetaEst(Z, gtime, delta)

#res   <- betaEst(x, Z, theta$alpha, theta$theta, gtime, delta)
#res
```

---

betaEstFam *Estimate the Fixed Effect Parameter for the Frailty Model*

Description

A method to estimate the fixed effect parameter for a frailty model accounting for family structure of related individuals (e.g., trios). The input data is assumed to be organized such that records for each family occur consecutively, and that records for offspring precede those for parents. The variance matrix for the random effects is assumed to be of the form `var*K`, where `K` is a matrix of kinship coefficients between family members. The following family structures are permitted: (Individual), (Offspring, Offspring), (Offspring, Parent), (Offspring, Parent, Parent), and (Offspring, Offspring, Parent, Parent). Other family structures have not been implemented.

Usage

```
betaEstFam(x, fam_group, fam_role, alpha, var, gtime, delta, lower, upper)
```

Arguments

`x`	Vector of numeric variables of interest for each sample.
`fam_group`	Vector of family IDs for each sample.
`fam_role`	Vector of indicators for the role within a family of each sample, i.e., {"Offspring","Mother","Father"}, or {"o","m","f"}.
`alpha`	Vector of baseline survival rates for each time interval.
`var`	Scalar for frailty variance.
`gtime`	Vector of observed survival times for each sample.
`delta`	Vector of event indicators for each sample: 1 indicates observed event, 0 indicates censored.
`lower`	Scalar for the lower bound of the fixed effect parameter estimation search region.
`upper`	Scalar for the upper bound of the fixed effect parameter estimation search region.

Value

Scalar estimate of the fixed effect parameter.

Examples

```
# Generate dummy data
x     <- c(0, 1, 1, 1, 2, 2, 0, 0, 0)
fam_group <- c('1', '1', '1', '2', '2', '2', '3', '3', '3')
fam_role  <- c("o","f","m","o","f","m","o","f","m")
alpha <- c(0.7500000, 0.6666667, 0.5000000, 0.0000000)
var   <- 0.2
gtime <- c(1, 3, 3, 2, 1, 1, 2, 3, 1)
delta <- c(1, 0, 1, 1, 1, 0, 1, 0, 1)
lower <- 0
upper <- 2

#res   <- betaEstFam(x, fam_group, fam_role, alpha, var, gtime, delta, lower, upper)
#res
```

---

geneStat — *Compute Gene-Level Statistics Based on Efficient Score Statistics for the Grouped Survival Model*

Description

A method to compute gene-level statistics using the contributions of individual samples to the efficient score statistics of SNPs under a grouped survival model. The function supports zero, single, or multiple covariates as dictated by the input data. The function can take either a `gwaa.data` object or a standard `data.frame` or `matrix` as input. The contribution of each sample to the efficient score will be computed separately for each SNP being tested, and then gene-level statistics will be computed using either the default Sequence Kernel Association Test (SKAT) or a user-provided function.

Usage

```
geneStat(x, Z=NULL, GenABEL.data=NULL, alpha, theta=NULL, gtime, delta,
          beta=0, nCores=1,
          FUN=function(Uij,weight){sum((colSums(Uij)*weight)^2)}, geneSet)
```

Arguments

`x`	Vector, `data.frame`, or `matrix` of numeric variables of interest for each sample. Alternatively, if a `gwaa.data` object is given for `GenABEL.data`, this argument may be a vector of strings corresponding to column names in `GenABEL.data@gtdata` to use as variables of interest.
`Z`	Optional `data.frame` or `matrix` of numeric covariate values for each sample. Alternatively, if a `gwaa.data` object is given for `GenABEL.data`, this argument may be a vector of strings corresponding to column names in `GenABEL.data@phdata` to use as covariates.

GenABEL.data	Optional gwaa.data object.
alpha	Vector of baseline survival rates for each time interval.
theta	Optional vector of estimated nuisance parameters for the covariates.
gtime	Vector of observed survival times for each sample.
delta	Vector of event indicators for each sample: 1 indicates observed event, 0 indicates censored.
beta	Scalar for the parameter of interest. Default is 0.
nCores	Integer representing the number of cores to be used for multi-threaded computation. Default is 1.
FUN	A function to compute the gene-level statistics base on the SNP-level efficient scores. The function arguments should include a matrix of each samples' contribution to the efficient scores, Uij, and a vector of weights, weight. Default is the SKAT statistic.
geneSet	A list of lists, where each sub-list contains two vectors: a vector of column names of the variables of interest to be included in that set, and a vector of numerical weights corresponding to each specified variable of interest.

Value

list with two elements: stat, a list of return objects from the default or user-defined FUN, and U, a list of matrices of the contribution of each subject to the efficient score statistics for each SNP in each gene set. Rows correspond to samples, and columns correspond to the variables of interest.

References

Wu, M.C., Lee, S., Cai, T., Li, Y., Boehnke, M. and Lin, X. (2011). Rare-variant association testing for sequencing data with the sequence kernel association test. American Journal of Human Genetics, 89:1, 82-93.
Ionita-Laza, I., Lee, S., Makarov, V., Buxbaum, J.D., and Lin, X. (2013). Sequence kernel association tests for the combined effect of rare and common variants. American Journal of Human Genetics, 92:6, 841-53.

Examples

```
# Generate dummy data
rm(list=ls())
set.seed(111)
n <- 1000

# covariate parameters
theta <- c(0.2, 0.2)

# covariate data (centered at 0)
z1 <- rnorm(n)
z2 <- rbinom(n, 1, 0.5) - 0.5
Z <- matrix(cbind(z1, z2), ncol = 2)
```

```
# continuous survival time
lam0 <- 1
cmax <- 3
lami <- lam0 * exp(Z[,1]*theta[1]+Z[,2]*theta[2])
stime <- rexp(n, lami)
ctime <- runif(n, 0, cmax)
delta <- stime < ctime
otime <- pmin(stime, ctime)

# number of observation times
ntps <- 5

# number of intervals
r <- ntps + 1

# last observation time
maxbreakq <- 0.85
maxbreak <- qexp(maxbreakq, lam0)

# grouped survival times
breaks <- (1:ntps) * (maxbreak/ntps)
gtime <- findInterval(otime, breaks) + 1
delta[gtime == r] <- FALSE
dctime <- findInterval(ctime, breaks) + 1
delta[gtime == dctime] <- FALSE
delta <- as.numeric(delta)
gtime[which(gtime == r)] <- Inf

# estimate nuisance parameters
#thetaest <- thetaEst(Z, gtime, delta)

# SNP and gene information
#geneInfo <- data.frame(gene=c("BRCA1","BRCA2"), chr=c(17,13),
#                       start=c(41196312, 32889611), end=c(41277500, 32973805),
#                       stringsAsFactors=FALSE)
#snpInfo <- data.frame(chr=c(17,17,13,13),
#                      pos=c(41211653,41213996,32890026,32890572),
#                      rsid=c("rs8176273","rs8176265","rs9562605","rs1799943"),
#                      stringsAsFactors=FALSE)

# use snplist package to create gene sets
#library(snplist)
#setGeneTable(geneInfo)
#setSNPTable(snpInfo)
#geneset <- makeGeneSet()

# simulate sample values for SNPs
#G <- matrix(rbinom(n*nrow(snpInfo), 2, 0.5), ncol=nrow(snpInfo))
#colnames(G) <- snpInfo$rsid

# dummy weights for the SNPs in the gene sets
#for(i in seq_len(length(geneset))){
#    weight <- rep(1, length(geneset[[i]]))
```

```
#    geneset[[i]] <- list(geneset[[i]], weight)

#}

# compute gene-level statistics based on the default SKAT function
#res <- geneStat(x=G, Z=Z, alpha=thetaest$alpha, theta=thetaest$theta,
#                gtime=gtime, delta=delta, geneSet=geneset, beta=0, nCores=1)
#res$stat
```

---

groupedSurv *Compute the Efficient Score Statistic for the Grouped Survival Model*

Description

A method to compute the efficient score statistic for a grouped survival model. The function supports zero, single, or multiple covariates as dictated by the input data. The function can take either a gwaa.data object or a standard data.frame or matrix as input. If more than one variable of interest is provided, the efficient score will be computed for each one separately. For each variable tested, the function returns the unadjusted marginal asymptotic p-value, as well as the family-wise error rate (FWER, Bonferroni correction) adjusted p-value and local false-discovery rates (FDR, Storey's Q-values).

Usage

```
groupedSurv(x, Z=NULL, GenABEL.data=NULL, alpha, theta=NULL,
  gtime, delta, beta=0, nCores=1, reScore=FALSE)
```

Arguments

x	Vector, data.frame, or matrix of numeric variables of interest for each sample. Alternatively, if a gwaa.data object is given for GenABEL.data, this argument may be a vector of strings corresponding to column names in GenABEL.data@gtdata to use as variables of interest, set NULL will use all the variables in GenABEL.data@gtdata as variables of interest.
Z	Optional data.frame or matrix of numeric covariate values for each sample. Alternatively, if a gwaa.data object is given for GenABEL.data, this argument may be a vector of strings corresponding to column names in GenABEL.data@phdata to use as covariates.
GenABEL.data	Optional gwaa.data object.
alpha	Vector of baseline survival rates for each time interval.
theta	Optional vector of estimated nuisance parameters for the covariates.
gtime	Vector of observed survival times for each sample.
delta	Vector of event indicators for each sample: 1 indicates observed event, 0 indicates censored.
beta	Scalar for the parameter of interest. Default is 0.

nCores	Integer representing the number of cores to be used for multi-threaded computation. Default is 1.
reScore	Boolean indicating whether to return the full efficient scores matrix. Default is FALSE

Value

If reScore=FALSE, the function returns a data.frame with one row for each variables of interest and four columns: stat, the efficient score statistic, pvalue, unadjusted p-value, FWER, the family-wise error rate adjusted p-value (Bonferroni correction), and FDR, the local false-discovery rate (Storey's Q-value). If reScore=TRUE, a list is returned. The first element is the data.frame of efficient score statistics and p-values, and the second is a matrix of contributions of each sample to the efficient score statistics of each variable of interest. Rows correspond to samples, and columns correspond to the variables of interest.

References

Bonferroni, C.E. (1935). Il calcolo delle assicurazioni su gruppi di teste. Studi in Onore del Professore Salvatore Ortu Carbon, 13-60.
Storey, J.D. (2003). The positive false discovery rate: a Bayesian interpretation and the q-value. Annals of Statistics, 31:6, 2013-2035.
Storey, J.D., Taylor, J.E., and Siegmund, D. (2004). Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. Journal of the Royal Statistical Society Series B-Statistical Methodology, 66, 187-205

Examples

```
# Generate dummy data
set.seed(111)
n <- 1000

# effect size
beta <- 0.4

# covariate parameters
theta <- c(0.2, 0.2)

# variable of interest associated with outcome
MAF <- 0.05
x <- matrix(rbinom(n, 2, MAF), ncol = 1)

# covariate data (centered at 0)
z1 <- rnorm(n)
z2 <- rbinom(n, 1, 0.5) - 0.5
Z  <- matrix(cbind(z1, z2), ncol = 2)

# continuous survival time
lam0 <- 1
cmax <- 3
lami <- lam0 * exp(x*beta + Z[,1]*theta[1]+Z[,2]*theta[2])
```

```
stime <- rexp(n, lami)
ctime <- runif(n, 0, cmax)
delta <- stime < ctime
otime <- pmin(stime, ctime)

# additional variables of interest
xMore <- matrix(rbinom(n*100, 2, MAF), ncol = 100)
xMore <- cbind(x, xMore)

# number of observation times
ntps <- 5

# number of intervals
r <- ntps + 1

# last observation time
maxbreakq <- 0.85
maxbreak <- qexp(maxbreakq, lam0)

# grouped failure times
breaks <- (1:ntps) * (maxbreak/ntps)
gtime <- findInterval(otime, breaks) + 1
delta[gtime == r] <- FALSE
dctime <- findInterval(ctime, breaks) + 1
delta[gtime == dctime] <- FALSE
delta <- as.numeric(delta)
gtime[which(gtime == r)] <- Inf

# estimate nuisance parameters
#thetaest <- thetaEst(Z, gtime, delta)

# compute efficient score statistics
# eff <- groupedSurv(x=xMore, Z=Z, alpha=thetaest$alpha, theta=thetaest$theta,
#  gtime=gtime, delta=delta, beta=0, nCores=1)
# head(eff)
```

`groupedSurvFam` *Compute the Efficient Score Statistic for the Frailty Model*

Description

A method to compute the efficient score statistic for a frailty model, accounting for family structure of related individuals (e.g., trios). The input data is assumed to be organized such that records for each family occur consecutively, and that records for offspring precede those for parents. The variance matrix for the random effects is assumed to be of the form `var*K`, where `K` is a matrix of kinship coefficients between family members. The following groupings are permitted: (Individual), (Offspring, Offspring), (Offspring, Parent), (Offspring, Parent, Parent), and (Offspring, Offspring, Parent, Parent). Other family structures have not been implemented.

Usage

```
groupedSurvFam(x, fam_group, fam_role, alpha, var, gtime, delta, beta=0, nCores=1)
```

Arguments

x	Vector or matrix of numeric variables of interest for each sample.
fam_group	Vector of family IDs for each sample.
fam_role	Vector of indicators for the role within a family of each sample, i.e., {"Offspring","Mother","Father"}, or {"o","m","f"}.
alpha	Vector of baseline survival rates for each time interval.
var	Scalar for frailty variance.
gtime	Vector of observed survival times for each sample.
delta	Vector of event indicators for each sample: 1 indicates observed event, 0 indicates censored.
beta	Scalar for the fixed effect size. Default is 0.
nCores	Integer representing the number of cores to be used for multi-threaded computation. Default is 1.

Value

Matrix of contributions of each family to the efficient score statistics of each variable of interest. Rows correspond to families, and columns correspond to the variables of interest.

Examples

```
# Generate dummy data
set.seed(111)
n <- 24

# effect size
beta <- 0.3

# covariate parameters
theta <- c(0.2, 0.2)

# variable of interest
MAF <- 0.05
x <- matrix(rbinom(n, 2, MAF), ncol = 1)

# covariate data (centered at 0)
z1 <- rnorm(n)
z2 <- rbinom(n, 1, 0.5) - 0.5
Z  <- matrix(cbind(z1, z2), ncol = 2)

# continuous survival time
lam0 <- 1
cmax <- 3
lami <- lam0 * exp(x*beta + Z[,1]*theta[1]+Z[,2]*theta[2])
```

```
stime <- rexp(n, lami)
ctime <- runif(n, 0, cmax)
delta <- stime < ctime
otime <- pmin(stime, ctime)

# number of observation times
ntps <- 5

# number of intervals
r <- ntps + 1

# last observation time
maxbreakq <- 0.85
maxbreak <- qexp(maxbreakq, lam0)

# grouped failure times
breaks <- (1:ntps) * (maxbreak/ntps)
gtime <- findInterval(otime, breaks) + 1
delta[gtime == r] <- FALSE
dctime <- findInterval(ctime, breaks) + 1
delta[gtime == dctime] <- FALSE
delta <- as.numeric(delta)

# family-specific information
m <- n/3
fam_role <- rep(c("o","f","m"),m)
fam_group <- as.character(rep(1:m,rep(3,m)))

# nuisance parameters
#alpha <- thetaEst(Z, gtime,delta)$alpha
#var <- 0.2

# compute efficient score statistics
#res   <- groupedSurvFam(x, fam_group, fam_role, alpha, var, gtime, delta)
#res
```

`PvalueFam` *Estimate P-values of the Efficient Score Statistics for the Frailty Model*

Description

A method to estimate the p-value for the efficient score statistics for the frailty model.

Usage

```
PvalueFam(U)
```

Arguments

`U`	`Matrix` of contributions of each family to the efficient score statistics of each variable of interest, (i.e., the output from `groupedSurvFam`).

Value

A `data.frame` with one row for each variables of interest and four columns: `stat`, the efficient score statistic, `pvalue`, unadjusted p-value, `FWER`, the family-wise error rate adjusted p-value (Bonferroni correction), and `FDR`, the local false-discovery rate (Storey's Q-value).

References

Bonferroni, C.E. (1935). Il calcolo delle assicurazioni su gruppi di teste. Studi in Onore del Professore Salvatore Ortu Carbon, 13-60.
Storey, J.D. (2003). The positive false discovery rate: a Bayesian interpretation and the q-value. Annals of Statistics, 31:6, 2013-2035.
Storey, J.D., Taylor, J.E., and Siegmund, D. (2004). Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. Journal of the Royal Statistical Society Series B-Statistical Methodology, 66, 187-205

Examples

```
# Generate dummy data
U <- matrix(rnorm(10000), ncol=50)

#res   <- PvalueFam(U)
#res
```

`thetaEst` *Estimate the Baseline Survival Rates and Covariate Parameters for the Grouped Survival Model*

Description

A method to estimate the baseline survival rate for each time interval and the covariate nuisance parameters for a grouped survival model. The estimation is conducted under the null hypothesis, i.e., that there is no effect from the variable of interest.

Usage

```
thetaEst(Z=NULL, gtime, delta, method="BFGS")
```

Arguments

`Z`	Optional `data.frame` or `matrix` of numeric covariate values for each sample.
`gtime`	Vector of observed survival times for each sample.
`delta`	Vector of event indicators for each sample: 1 indicates observed event, 0 indicates censored.
`method`	String indicating optimization method, passed to `stats::optim`. Supports `"BFGS"` and `"CG"`. Default is `"BFGS"`.

Value

A list containing two vectors: `alpha`, the baseline survival rate for each time interval, and `theta`, the estimated nuisance parameters of the covariates.

Examples

```
# Generate dummy data
cov1  <- c(1, 2, 2, 2, 1, 1, 0, 1, 1)
cov2  <- c(2, 2, 1, 0, 1, 0, 1, 1, 0.5)
Z     <- cbind(cov1, cov2)
gtime <- c(1, 3, 3, 2, 1, 1, 2, 3, 1)
delta <- c(1, 0, 1, 1, 1, 0, 1, 0, 1)

#res   <- thetaEst(Z, gtime, delta)
#res
```

`varEstFam` *Estimate the Frailty Variance for the Frailty Model*

Description

A method to estimate the frailty variance for a frailty model accounting for family structure of related individuals (e.g., trios). The input data is assumed to be organized such that records for each family occur consecutively, and that records for offspring precede those for parents. The variance matrix for the random effects is assumed to be of the form `var*K`, where `K` is a matrix of kinship coefficients between family members. The following groupings are permitted: (Individual), (Offspring, Offspring), (Offspring, Parent), (Offspring, Parent, Parent), and (Offspring, Offspring, Parent, Parent). Other family structures have not been implemented.

Usage

```
varEstFam(x, fam_group, fam_role, alpha, gtime, delta, lower, upper, beta=0)
```

Arguments

`x`	Vector of numeric variables of interest for each sample.
`fam_group`	Vector of family IDs for each sample.
`fam_role`	Vector of indicators for the role within a family of each sample, i.e., {"Offspring","Mother","Father"}, or {"o","m","f"}.
`alpha`	Vector of baseline survival rates for each time interval.
`gtime`	Vector of observed survival times for each sample.
`delta`	Vector of event indicators for each sample: 1 indicates observed event, 0 indicates censored.
`lower`	Scalar for the lower bound of the variance estimation search region.
`upper`	Scalar for the upper bound of the variance estimation search region.
`beta`	Scalar for the fixed effect size. Default is 0.

Value

Scalar estimate of the frailty variance.

Examples

```
# Generate dummy data
x     <- c(0, 1, 1, 1, 2, 2, 0, 0, 0)
fam_group <- c('1', '1', '1', '2', '2', '2', '3', '3', '3')
fam_role  <- c("o","f","m","o","f","m","o","f","m")
alpha <- c(0.7500000, 0.6666667, 0.5000000, 0.0000000)
gtime <- c(1, 3, 3, 2, 1, 1, 2, 3, 1)
delta <- c(1, 0, 1, 1, 1, 0, 1, 0, 1)
lower <- 0
upper <- 2
beta  <- 1

#res   <- varEstFam(x, fam_group, fam_role, alpha, gtime, delta, lower, upper, beta)
#res
```

---

varLLFam — *Log-Likelihood of the Frailty Variance for the Discrete-Time Frailty Model*

Description

A method to compute the log-likelihood of the frailty variance for a frailty model accounting for family structure of related individuals (e.g., trios). The input data is assumed to be organized such that records for each family occur consecutively, and that records for offspring precede those for parents. The variance matrix for the random effects is assumed to be of the form `var*K`, where `K` is a matrix of kinship coefficients between family members. The following groupings are permitted: (Individual), (Offspring, Offspring), (Offspring, Parent), (Offspring, Parent, Parent), and (Offspring, Offspring, Parent, Parent). Other family structures have not been implemented.

Usage

```
varLLFam(x, fam_group, fam_role, alpha, var, gtime, delta, beta=0)
```

Arguments

`x`	Vector of numeric variables of interest for each sample.
`fam_group`	Vector of family IDs for each sample.
`fam_role`	Vector of indicators for the role within a family of each sample, i.e., {"Offspring","Mother","Father"}, or {"o","m","f"}.
`alpha`	Vector of baseline survival rates for each time interval.
`var`	Scalar for frailty variance.
`gtime`	Vector of observed survival times for each sample.

delta	Vector of event indicators for each sample: 1 indicates observed event, 0 indicates censored.
beta	Scalar for the fixed effect size. Default is 0.

Value

Scalar for the log-likelihood of the frailty variance.

Examples

```
# Generate dummy data
x     <- c(0, 1, 1, 1, 2, 2, 0, 0, 0)
fam_group <- c('1', '1', '1', '2', '2', '2', '3', '3', '3')
fam_role <- c("o","f","m","o","f","m","o","f","m")
alpha <- c(0.7500000, 0.6666667, 0.5000000, 0.0000000)
var <- 0.1
gtime <- c(1, 3, 3, 2, 1, 1, 2, 3, 1)
delta <- c(1, 0, 1, 1, 1, 0, 1, 0, 1)
beta  <- 1.0

#res   <- varLLFam(x, fam_group, fam_role, alpha, var, gtime, delta, beta)
#res
```

Index

∗ **package**
groupedSurv-package, 2

alphaEstFam, 3

betaEst, 4
betaEstFam, 5

geneStat, 6
groupedSurv, 9
groupedSurv-package, 2
groupedSurvFam, 11

PvalueFam, 13

thetaEst, 14

varEstFam, 15
varLLFam, 16