

Package ‘SIMICO’

May 7, 2026

Type Package

Title Set-Based Inference for Multiple Interval-Censored Outcomes

Version 0.2.0

Description

Contains tests for association between a set of genetic variants and multiple correlated outcomes that are interval censored. Interval-censored data arises when the exact time of the onset of an outcome of interest is unknown but known to fall between two time points.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.1

Imports bindata, fastGHQuad, CompQuadForm, stats, ICSKAT

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Jaihee Choi [aut, cre],
Ryan Sun [aut]

Maintainer Jaihee Choi <jaiheechoi01@gmail.com>

Repository CRAN

Date/Publication 2022-11-25 23:10:02 UTC

Contents

fd_term	2
gammasigma	3
gammatheta	4
gammatheta_off	5
gamma_fd	6
gamma_off	7
gamma_on	8
get_A	9
Get_CausalSNPs_bynum	9

haz_left	10
haz_right	10
sd_off	11
sd_on	12
simico_fit_null	12
simico_gen_dat	14
simico_out	16
sim_gmat	18
ss_fd	19
ss_sd	20
st_off	21
surv_diff	22
surv_left	22
surv_right	23
without_one_phen	23
without_two_phen	24

Index **25**

fd_term	<i>d/d_theta_l</i>
---------	--------------------

Description

Calculate the first derivative of the theta terms for outcome l.

Usage

```
fd_term(l, temp_beta, phen,d, apply_diffs,
        A_i, no_l_all,HL_array, HR_array)
```

Arguments

l	Outcome of interest.
temp_beta	Vector of fitted coefficients.
phen	list containing the covariate design matrices.
d	Number of quadrature nodes.
apply_diffs	Matrix containing the differences between survival functions of the left and right time intervals.
A_i	Product of apply_diffs across all outcomes k summed over all quadrature nodes d.
no_l_all	n x (k - 1) matrix containing the product of apply_diffs across all outcomes k excluding the current outcome l.
HL_array	n x k matrix containing all the hazard values for the left times.
HR_array	n x k matrix containing all the hazard values for the right times.

Value

The output is a $1 \times (p + 2)$ vector of the first derivative terms for outcome l .

<code>gammasigma</code>	$d^2/d_{\text{gamma}} d_{\text{sigma}}^2$
-------------------------	---

Description

Calculates the Information matrix term of $I_{\text{eta gamma}}$ for one outcome of interest l .

Usage

```
gammasigma(
  l, HL_array, HR_array, tpos_all, obs_all,
  apply_diffs, temp_beta, A_i, xDats, no_l_all,
  no_two_all, gMat, a1, a2, k, d)
```

Arguments

<code>l</code>	Index of first outcome of interest.
<code>HL_array</code>	$n \times K$ matrix containing all the hazard values for the left times.
<code>HR_array</code>	$n \times K$ matrix containing all the hazard values for the right times.
<code>tpos_all</code>	$n \times k$ matrix containing a indicator for whether that time is left-censored or not.
<code>obs_all</code>	$n \times k$ matrix containing a indicator for whether that time is right-censored or not.
<code>apply_diffs</code>	Matrix containing the differences between survival functions of the left and right time intervals.
<code>temp_beta</code>	vector of fitted coefficients.
<code>A_i</code>	Product of <code>apply_diffs</code> across all outcomes K summed over all quadrature nodes D .
<code>xDats</code>	List of design matrices.
<code>no_l_all</code>	$n \times (k - 1)$ matrix containing the product of <code>apply_diffs</code> across all outcomes K excluding the current outcome l .
<code>no_two_all</code>	$n \times (k - 2)$ matrix containing the product of <code>apply_diffs</code> across all outcomes k excluding the outcomes l and m .
<code>gMat</code>	$n \times q$ matrix of genetic information.
<code>a1</code>	First shape parameter of beta parameter.
<code>a2</code>	Second shape parameter of beta parameter.
<code>k</code>	Total number of outcomes.
<code>d</code>	Number of quadrature nodes.

Value

The output is a matrix containing the component of the information matrix of the gamma and sigma^2 parameters for outcome l .

gammatheta	$d^2/d_{\gamma}d_{\theta_k}$
------------	------------------------------

Description

Calculates the Information matrix term of $I_{\eta \gamma}$ for outcome k .

Usage

```
gammatheta(l, HL_array, HR_array, tpos_all, obs_all, apply_diffs,
           temp_beta, A_i, xDats, no_l_all, gMat, a1, a2, d)
```

Arguments

<code>l</code>	Index of first outcome of interest.
<code>HL_array</code>	$n \times k$ matrix containing all the hazard values for the left times.
<code>HR_array</code>	$n \times k$ matrix containing all the hazard values for the right times.
<code>tpos_all</code>	$n \times k$ matrix containing a indicator for whether that time is left-censored or not.
<code>obs_all</code>	$n \times k$ matrix containing a indicator for whether that time is right-censored or not.
<code>apply_diffs</code>	Matrix containing the differences between survival functions of the left and right time intervals.
<code>temp_beta</code>	vector of fitted coefficients.
<code>A_i</code>	Product of <code>apply_diffs</code> across all outcomes k summed over all quadrature nodes d .
<code>xDats</code>	List of design matrices for all outcomes.
<code>no_l_all</code>	$n \times (k - 1)$ matrix containing the product of <code>apply_diffs</code> across all outcomes k excluding the current outcome l .
<code>gMat</code>	$n \times q$ matrix of genetic information.
<code>a1</code>	First shape parameter of beta parameter.
<code>a2</code>	Second shape parameter of beta parameter.
<code>d</code>	Number of quadrature nodes.

Value

The output is a matrix containing the component of the information matrix of the gamma and theta parameters for outcome l .

gammatheta_off $d^2/d_gamma_l dtheta_m$

Description

Calculates the Information matrix term of I_{η} gamma for outcomes l and m

Usage

```
gammatheta_off(l,m, HL_array, HR_array, xAll, apply_diffs, temp_beta,
  A_i, no_l_all, no_two_all, gMat, a1, a2, k, d)
```

Arguments

l	Index of first outcome of interest.
m	Index of second outcome of interest.
HL_array	n x k matrix containing all the hazard values for the left times.
HR_array	n x k matrix containing all the hazard values for the right times.
xAll	List of design matrices and censoring terms.
apply_diffs	Matrix containing the differences between survival functions of the left and right time intervals.
temp_beta	vector of fitted coefficients.
A_i	Product of apply_diffs across all outcomes K summed over all quadrature nodes d.
no_l_all	n x (k - 1) matrix containing the product of apply_diffs across all outcomes k excluding the current outcome l.
no_two_all	n x (k - 2) matrix containing the product of apply_diffs across all outcomes k excluding the outcomes l and m.
gMat	n x q matrix of genetic information.
a1	First shape parameter of beta parameter.
a2	Second shape parameter of beta parameter.
k	Total number of outcomes.
d	Number of quadrature nodes.

Value

The output is a matrix containing the component of the information matrix of the gamma and theta parameters for outcomes l and m.

gamma_fd	d/d_gamma_l
----------	-----------------

Description

Calculates the gradient term for U_g for the score statistic.

Usage

```
gamma_fd(l, HL_array, HR_array, tpos_all, obs_all,
         temp_beta, A_i, no_l_all, gMat, a1, a2, d)
```

Arguments

l	Index of first outcome of interest.
HL_array	n x k matrix containing all the hazard values for the left times.
HR_array	n x k matrix containing all the hazard values for the right times.
tpos_all	n x k matrix containing a indicator for whether that time is left-censored or not.
obs_all	n x k matrix containing a indicator for whether that time is right-censored or not.
temp_beta	Vector of fitted coefficients.
A_i	Product of apply_diffs across all outcomes k summed over all quadrature nodes d.
no_l_all	n x (K - 1) matrix containing the product of apply_diffs across all outcomes K excluding the current outcome l.
gMat	n x q matrix of genetic information.
a1	First shape parameter of beta parameter.
a2	Second shape parameter of beta parameter.
d	Number of quadrature nodes.

Value

The output is a vector containing the first derivative with respect to gamma.

gamma_off	$d^2/d_gamma_ldgamma_m$
-----------	----------------------------

Description

Calculates the [off-diagonal] Information matrix term for Γ with respect to outcome l and outcome m .

Usage

```
gamma_off(l, m, HL_array, HR_array,
         tpos_all, obs_all, temp_beta, A_i,
         no_l_all, no_two_all, gMat, a1, a2, k, d)
```

Arguments

<code>l</code>	Index of first outcome of interest.
<code>m</code>	Index of second outcome of interest.
<code>HL_array</code>	$n \times k$ matrix containing all the hazard values for the left times.
<code>HR_array</code>	$n \times k$ matrix containing all the hazard values for the right times.
<code>tpos_all</code>	$n \times k$ matrix containing a indicator for whether that time is left-censored or not.
<code>obs_all</code>	$n \times k$ matrix containing a indicator for whether that time is right-censored or not.
<code>temp_beta</code>	Vector of fitted coefficients.
<code>A_i</code>	Product of <code>apply_diffs</code> across all outcomes k summed over all quadrature nodes d .
<code>no_l_all</code>	$n \times (k - 1)$ matrix containing the product of <code>apply_diffs</code> across all outcomes k excluding the current outcome l .
<code>no_two_all</code>	$n \times (k - 2)$ matrix containing the product of <code>apply_diffs</code> across all outcomes k excluding the outcomes l and m .
<code>gMat</code>	$n \times q$ matrix of genetic information.
<code>a1</code>	First shape parameter of beta parameter.
<code>a2</code>	Second shape parameter of beta parameter.
<code>k</code>	Total number of outcomes.
<code>d</code>	Number of quadrature nodes.

Value

The output is a matrix containing the component of the information matrix of the gamma parameter for outcomes l and m .

gamma_on	$d^2/d_{\text{gamma}} d_{\text{gamma}_l}$
----------	---

Description

Calculates the [on-diagonal] Information matrix term for I_{gamma} with respect to outcome l .

Usage

```
gamma_on(l, HL_array, HR_array, tpos_all, obs_all,
        temp_beta, A_i, no_l_all, gMat, a1, a2, d)
```

Arguments

<code>l</code>	Index of first outcome of interest.
<code>HL_array</code>	$n \times k$ matrix containing all the hazard values for the left times.
<code>HR_array</code>	$n \times k$ matrix containing all the hazard values for the right times.
<code>tpos_all</code>	$n \times k$ matrix containing a indicator for whether that time is left-censored or not.
<code>obs_all</code>	$n \times k$ matrix containing a indicator for whether that time is right-censored or not.
<code>temp_beta</code>	Vector of fitted coefficients.
<code>A_i</code>	Product of <code>apply_diffs</code> across all outcomes k summed over all quadrature nodes d .
<code>no_l_all</code>	$n \times (k - 1)$ matrix containing the product of <code>apply_diffs</code> across all outcomes k excluding the current outcome l .
<code>gMat</code>	$n \times q$ matrix of genetic information.
<code>a1</code>	First shape parameter of beta parameter.
<code>a2</code>	Second shape parameter of beta parameter.
<code>d</code>	Number of quadrature nodes.

Value

The output is a matrix containing the component of the information matrix of the gamma parameter for outcome l .

get_A	<i>Get A vector</i>
-------	---------------------

Description

Product of difference of survival terms of the left and right interval times, across all outcomes k, summed over all quadrature nodes d.

Usage

```
get_A(store, weights, d, n)
```

Arguments

store	Matrix of difference of survival values of the left and right time intervals.
weights	Gaussian quadrature weights.
d	Total number of Gaussian quadrature nodes.
n	Total number of observations.

Value

The output is a vector used to compute the derivative terms.

Get_CausalSNPs_bynum	<i>Get_CausalSNPs_bynum()</i>
----------------------	-------------------------------

Description

Matrix of subsetted genetic information.

Usage

```
Get_CausalSNPs_bynum(gMat, num, Causal.MAF.Cutoff)
```

Arguments

gMat	Matrix of SNPs.
num	Number of causal variants.
Causal.MAF.Cutoff	Minor allele frequency value cutoff for causal SNPs.

Value

Output is a vector of indices to subset the full genetic matrix.

haz_left	$H_{ik}(L_{ik})$
----------	------------------

Description

Calculates the hazard function of the left time interval for outcome l.

Usage

```
haz_left(l, d, temp_beta, phen, r1, k)
```

Arguments

l	Outcome of interest.
d	Total number of Gaussian quadrature nodes.
temp_beta	vector of fitted coefficients.
phen	list of data matrices containing both left and right information.
r1	Gaussian quadrature nodes.
k	Total number of outcomes.

Value

The output is a vector of the hazard values of the left times.

haz_right	$H_{ik}(R_{ik})$
-----------	------------------

Description

Calculates the hazard function of the right time interval for outcome l.

Usage

```
haz_right(l, d, temp_beta, phen, r1, k)
```

Arguments

l	Outcome of interest.
d	Total number of Gaussian quadrature nodes.
temp_beta	vector of fitted coefficients.
phen	list of data matrices containing both left and right information.
r1	Gaussian quadrature nodes.
k	Total number of outcomes.

Value

The output is a vector of the hazard values of the right times.

sd_off	$d^2/d_{\theta} d_{\sigma}^2$
--------	-------------------------------

Description

Calculates the Information matrix term of $I_{\theta} \sigma^2$ for outcomes l and m .

Usage

```
sd_off(l, m, phen_l, phen_m, temp_beta, d, apply_diffs, A_i,
      HL_array, HR_array, no_l_all, no_two_all, tpos_all, obs_all, k)
```

Arguments

l	Index of first outcome of interest.
m	Index of second outcome of interest.
phen_l	List containing the left and right design matrices and interval times for outcome l .
phen_m	List containing the left and right design matrices and interval times for outcome m .
temp_beta	vector of fitted coefficients.
d	Total number of quadrature nodes.
apply_diffs	Matrix containing the differences between survival functions of the left and right time intervals.
A_i	Product of <code>apply_diffs</code> across all outcomes k summed over all quadrature nodes d .
HL_array	$n \times k$ matrix containing all the hazard values for the left times.
HR_array	$n \times k$ matrix containing all the hazard values for the right times.
no_l_all	$n \times (k - 1)$ matrix containing the product of <code>apply_diffs</code> across all outcomes k excluding the current outcome l .
no_two_all	$n \times (k - 2)$ matrix containing the product of <code>apply_diffs</code> across all outcomes K excluding outcomes l and m .
tpos_all	$n \times k$ matrix containing a indicator for whether that time is left-censored or not.
obs_all	$n \times k$ matrix containing a indicator for whether that time is right-censored or not.
k	Total number of outcomes.

Value

The output is a matrix containing the component of the information matrix of the σ and θ parameters.

sd_on	$d^2/d\sigma^2$
-------	-----------------

Description

Calculates the Information matrix term of I_{σ^2} for outcome l .

Usage

```
sd_on(l, k, temp_beta, phen, d, apply_diffs, A_i,
      no_l_all, HL_array, HR_array)
```

Arguments

l	Index of first outcome of interest.
k	Total number of outcomes.
temp_beta	vector of fitted coefficients.
phen	List containing the left and right design matrices and interval times for outcome l .
d	Total number of quadrature nodes.
apply_diffs	Matrix containing the differences between survival functions of the left and right time intervals.
A_i	Product of apply_diffs across all outcomes K summed over all quadrature nodes D .
no_l_all	$n \times (k - 1)$ matrix containing the product of apply_diffs across all outcomes k excluding the current outcome l .
HL_array	$n \times k$ matrix containing all the hazard values for the left times.
HR_array	$n \times k$ matrix containing all the hazard values for the right times.

Value

The output is a single value for the second derivative with respect to σ .

simico_fit_null	<i>simico_fit_null()</i>
-----------------	--------------------------

Description

Fit the null model via newton raphson for multiple outcomes interval-censored skat.

Usage

```
simico_fit_null(init_beta, epsilon, xDats, lt_all, rt_all, k, d)
```

Arguments

init_beta	Starting values for NR.
epsilon	Stopping criterion for NR.
xDats	List of left and right design matrices.
lt_all	n x k matrix of left times.
rt_all	n x k matrix of right times.
k	Total number of outcomes.
d	Total number of quadrature nodes.

Value

beta_fit	Vector of fitted coefficients.
iter	Number of iterations needed for the Newton-Raphson to converge.
diff	Difference between the current values of temp_beta and the previous iteration of temp_beta.
jmat	Information matrix of the theta parameters.
grad	Vector of the first derivaive of the theta parameters.

Examples

```
# Set number of outcomes
k = 2

# Set number of observations
n = 100

# Set number of covariates
p = 2

# Set number of SNPs
q = 50

# Set number of causal SNPs
num = 5

# Set number of quadrature nodes
d = 100

# Variance of subject-specific random effect
tauSq = 1

# Define the effect sizes
effectSizes <- c(.03, .15)

# Set MAF cutoff for causal SNPs
Causal.MAF.Cutoff = 0.1

# the baseline cumulative hazard function
```

```

bhFunInv <- function(x) {x}

set.seed(1)

# Generate covariate matrix
xMat <- cbind(rnorm(n), rbinom(n=n, size=1, prob=0.5))

# Generate genetic matrix
gMat <- matrix(data=rbinom(n=n*q, size=2, prob=0.1), nrow=n)

# Get indices to specific select causal variants
idx <- Get_CausalSNPs_bynum(gMat, num, Causal.MAF.Cutoff)

# Subset the gMat
gMatCausal <- gMat[,idx]

# Generate the multiple outcomes
exampleDat <- simico_gen_dat(bhFunInv = bhFunInv, obsTimes = 1:3,
                           windowHalf = 0.1, n, p, k, tauSq, gMatCausal,
                           xMat, effectSizes)

# Set the initial estimate values
init_beta <-c (rep(c(0, 0, 0, 1, 0), k), 1)

# Run the Newton-Raphson
nullFit <- simico_fit_null(init_beta = init_beta,
                          epsilon = 10^-5, xDats = exampleDat$fullDat$xDats,
                          lt_all = exampleDat$leftTimesMat,
                          rt_all = exampleDat$rightTimesMat,
                          k = k, d = d)

```

simico_gen_dat	simico_gen_dat()
----------------	------------------

Description

Generate multiple interval-censored data under proportional hazards model.

Usage

```
simico_gen_dat(bhFunInv, obsTimes = 1:3, windowHalf = 0.1,
              n, p, k, tauSq, gMatCausal, xMat, effectSizes)
```

Arguments

bhFunInv	The inverse of the baseline hazard function.
obsTimes	Vector of the intended observation times.

windowHalf	The amount of time before or after the intended obsTimes that a visit might take place.
n	Total number of observations.
p	Total number of covariates.
k	Total number of outcomes.
tauSq	Variance of the subject specific random effect.
gMatCausal	Matrix of subsetting genetic information for only a select causal SNPs.
xMat	Matrix of covariates.
effectSizes	Vector of genetic effect sizes. Should be entered as a vector the same length as the number of outcomes.

Value

exactTimesMat	n x k matrix containing the simulated exact times that the event occurred.
leftTimesMat	n x k matrix containing the left time interval that is observed.
rightTimesMat	n x k matrix containing the right time interval that is observed.
obsInd	n x k matrix containing a indicator for whether that time is right-censored or not.
tposInd	n x k matrix containing a indicator for whether that time is left-censored or not.
fullDat	Data in complete form to enter into SIMICO functions.

Examples

```
# Set number of outcomes
k = 2

# Set number of observations
n = 100

# Set number of covariates
p = 2

# Set number of SNPs
q = 50

# Set number of causal SNPs
num = 5

# Set number of quadrature nodes
d = 100

# Variance of subject-specific random effect
tauSq = 1

# Define the effect sizes
effectSizes <- c(.03, .15)

# Set MAF cutoff for causal SNPs
```

```

Causal.MAF.Cutoff = 0.1

# the baseline cumulative hazard function
bhFunInv <- function(x) {x}

set.seed(1)

# Generate covariate matrix
xMat <- cbind(rnorm(n), rbinom(n=n, size=1, prob=0.5))

# Generate genetic matrix
gMat <- matrix(data=rbinom(n=n*q, size=2, prob=0.1), nrow=n)

# Get indices to specific select causal variants
idx <- Get_CausalSNPs_bynum(gMat, num, Causal.MAF.Cutoff)

# Subset the gMat
gMatCausal <- gMat[,idx]

# Generate the multiple outcomes
exampleDat <- simico_gen_dat(bhFunInv = bhFunInv, obsTimes = 1:3,
                             windowHalf = 0.1, n, p, k, tauSq, gMatCausal,
                             xMat, effectSizes)

```

simico_out

Get P-Values

Description

Calculate test statistic and p-values for multiple outcome test and multiple burden test.

Usage

```
simico_out(nullFit, xDats, lt_all, rt_all, Itt, a1, a2, G, k, d)
```

Arguments

nullFit	Results of the Newton-Raphson: estimates of the beta coefficients.
xDats	List of design matrices.
lt_all	Matrix containing the generated left interval times.
rt_all	Matrix containing the generated right interval times.
Itt	I_theta theta - Information matrix of theta.
G	n x q matrix of genetic information.
a1	First shape parameter of beta parameter.
a2	Second shape parameter of beta parameter.
k	Total number of outcomes.
d	Number of quadrature nodes.

Value

multQ	Score statistic for multiple outcome test.
multP	P-value for multiple outcome test.
burdQ	Score statistic for multiple burden test.
burdP	P-value for multiple burden test.

Examples

```
# Set number of outcomes
k = 2

# Set number of observations
n = 100

# Set number of covariates
p = 2

# Set number of SNPs
q = 50

# Set number of causal SNPs
num = 5

# Set number of quadrature nodes
d = 100

# Variance of subject-specific random effect
tauSq = 1

# Define the effect sizes
effectSizes <- c(.03, .15)

# Set MAF cutoff for causal SNPs
Causal.MAF.Cutoff = 0.1

# the baseline cumulative hazard function
bhFunInv <- function(x) {x}

set.seed(1)

# Generate covariate matrix
xMat <- cbind(rnorm(n), rbinom(n=n, size=1, prob=0.5))

# Generate genetic matrix
gMat <- matrix(data=rbinom(n=n*q, size=2, prob=0.1), nrow=n)

# Get indices to specific select causal variants
idx <- Get_CausalSNPs_bynum(gMat, num, Causal.MAF.Cutoff)

# Subset the gMat
gMatCausal <- gMat[,idx]
```

```

# Generate the multiple outcomes
exampleDat <- simico_gen_dat(bhFunInv = bhFunInv, obsTimes = 1:3,
                           windowHalf = 0.1, n, p, k, tauSq, gMatCausal,
                           xMat, effectSizes)

# Set the initial estimate values
init_beta <-c (rep(c(0, 0, 0, 1, 0), k), 1)

# Run the newton-raphson
nullFit <- simico_fit_null(init_beta = init_beta,
                          epsilon = 10^-5, xDats = exampleDat$fullDat$xDats,
                          lt_all = exampleDat$leftTimesMat,
                          rt_all = exampleDat$rightTimesMat,
                          k = k, d = d)

# Get the test statistics p-values
out <- simico_out(nullFit = nullFit$beta_fit,
                 xDats = exampleDat$fullDat$xDats,
                 lt_all = exampleDat$leftTimesMat,
                 rt_all = exampleDat$rightTimesMat,
                 Itt = nullFit$jmat, a1 = 1, a2 = 25,
                 G = gMat, k = k, d = d)

# Print results
# Score statistic
out$multQ

# P-values
out$multP

```

sim_gmat

Simulate genetic matrix.

Description

Simulates a $n \times q$ genetic matrix with the option to specify the common pairwise correlation.

Usage

```
sim_gmat(n, q, rho)
```

Arguments

n	Total number of observations.
q	Total number of SNPs.
rho	Common pairwise correlation parameter.

Value

The result of a $n \times q$ genetic matrix of q SNPs.

Examples

```
# Set sample size
n = 100

# Set number of SNPs
q = 5

# Set common pairwise correlation
rho = 0.1

# Simulate genetic matrix
gMat <- sim_gmat(n, q, rho)
```

ss_fd	<i>d/d_sigma^2</i>
-------	--------------------

Description

Calculates the first derivative term with respect to σ^2 .

Usage

```
ss_fd(l, phen, HL_array, HR_array, tpos_all, obs_all,
      apply_diffs, temp_beta, A_i, no_l_all, k, d)
```

Arguments

l	Index of first outcome of interest.
phen	List containing all the left and right design matrices.
HL_array	$n \times k$ matrix containing all the hazard values for the left times.
HR_array	$n \times k$ matrix containing all the hazard values for the right times.
tpos_all	$n \times k$ matrix containing a indicator for whether that time is left-censored or not.
obs_all	$n \times k$ matrix containing a indicator for whether that time is right-censored or not.
apply_diffs	Matrix containing the differences between survival functions of the left and right time intervals.
temp_beta	vector of fitted coefficients.
A_i	Product of apply_diffs across all outcomes k summed over all quadrature nodes d .
no_l_all	$n \times (k - 1)$ matrix containing the product of apply_diffs across all outcomes k excluding the current outcome l .
k	Total number of outcomes.
d	Number of quadrature nodes.

Value

The output is a single value for the first derivative with respect to sigma.

ss_sd	$d^2/d_sigma^{2^2}$
-------	----------------------

Description

Calculates the second derivative term with respect to σ^2 .

Usage

```
ss_sd(HL_array, HR_array, xAll, apply_diffs, temp_beta,
      A_i, no_l_all, no_two_all, k, d)
```

Arguments

HL_array	n x k matrix containing all the hazard values for the left times.
HR_array	n x k matrix containing all the hazard values for the right times.
xAll	List containing the left and right matrices and event times.
apply_diffs	Matrix containing the differences between survival functions of the left and right time intervals.
temp_beta	vector of fitted coefficients.
A_i	Product of apply_diffs across all outcomes K summed over all quadrature nodes D.
no_l_all	n x (k - 1) matrix containing the product of apply_diffs across all outcomes k excluding the current outcome l.
no_two_all	n x (k - 2) matrix containing the product of apply_diffs across all outcomes k excluding outcomes l and m.
k	Total number of outcomes.
d	Number of quadrature nodes.

Value

The output is a single value for the second derivative with respect to σ^2 .

st_off	$d^2/d_{\theta} d_{\sigma}^2$
--------	-------------------------------

Description

Calculates the Information matrix term of I_{η} theta for one outcome of interest l.

Usage

```
st_off(l, HL_array, HR_array, xAll, apply_diffs,
      temp_beta, A_i, no_l_all, no_two_all, k, d)
```

Arguments

l	Index of first outcome of interest.
HL_array	n x k matrix containing all the hazard values for the left times.
HR_array	n x k matrix containing all the hazard values for the right times.
xAll	List containing the left and right matrices and event times.
apply_diffs	Matrix containing the differences between survival functions of the left and right time intervals.
temp_beta	vector of fitted coefficients.
A_i	Product of apply_diffs across all outcomes K summed over all quadrature nodes D.
no_l_all	n x (k - 1) matrix containing the product of apply_diffs across all outcomes k excluding the current outcome l.
no_two_all	n x (k - 2) matrix containing the product of apply_diffs across all outcomes k excluding outcomes l and m.
k	Total number of outcomes.
d	Number of quadrature nodes.

Value

The output is a matrix containing the component of the information matrix of the theta eta parameters for outcome l.

surv_diff	$S_{ik}(L_{ik}) - S_{ik}(R_{ik})$
-----------	-----------------------------------

Description

Calculates the difference between the survival functions of the left and right time intervals for outcome k for quadrature node d.

Usage

```
surv_diff(l, d, temp_beta, phen, r1, k)
```

Arguments

l	Outcome of interest.
d	Total number of Gaussian quadrature nodes.
temp_beta	Vector of fitted coefficients.
phen	List of data matrices containing both left and right information.
r1	Gaussian quadrature nodes.
k	Total number of outcomes.

Value

The output is a vector of the difference of the survival values of the left times and right times.

surv_left	$S_{ik}(L_{ik})$
-----------	------------------

Description

Calculates the survival function of the left time interval for outcome k for quadrature node d.

Usage

```
surv_left(l, d, temp_beta, phen, r1, k)
```

Arguments

l	Outcome of interest.
d	Total number of Gaussian quadrature nodes.
temp_beta	Vector of fitted coefficients.
phen	List of data matrices containing both left and right information.
r1	Gaussian quadrature nodes.
k	Total number of outcomes.

Value

The output is a vector of the survival values of the left times.

surv_right	$S_{ik}(R_{ik})$
------------	------------------

Description

Calculates the survival function of the right time interval for outcome k for quadrature node d.

Usage

```
surv_right(l, d, temp_beta, phen, r1, k)
```

Arguments

l	Outcome of interest.
d	Total number of Gaussian quadrature nodes.
temp_beta	Vector of fitted coefficients.
phen	List of data matrices containing both left and right information.
r1	Gaussian quadrature nodes.
k	Total number of outcomes.

Value

The output is a vector of the survival values of the left times.

without_one_phen	<i>Survival Difference Product without Outcome l</i>
------------------	--

Description

Calculate the product of the difference between survival terms excluding that of the outcome of interest.

Usage

```
without_one_phen(l, k, store)
```

Arguments

l	Outcome of interest.
k	Total number of outcomes.
store	Array of difference between left and right survival values.

Value

A $n \times (k-1)$ matrix where each column is the product of all the differences of left and right survival values across all outcomes excluding the column index outcome.

without_two_phen	<i>Survival Difference Product without Outcomes l and m</i>
------------------	---

Description

Difference of survival functions multiplied across all outcomes excluding outcomes l and m.

Usage

```
without_two_phen(l, m, k, store, n, d)
```

Arguments

l	The first outcome of interest.
m	The second outcome of interest.
k	Total number of outcomes.
store	Array of difference between left and right survival values.
n	Total number of observation.
d	Total number of quadrature nodes.

Value

A $n \times (k-2)$ matrix containing the product of all the differences of left and right survival values across all outcomes excluding outcomes l and m.

Index

[fd_term](#), 2

[gamma_fd](#), 6

[gamma_off](#), 7

[gamma_on](#), 8

[gammasigma](#), 3

[gammatheta](#), 4

[gammatheta_off](#), 5

[get_A](#), 9

[Get_CausalSNPs_bynum](#), 9

[haz_left](#), 10

[haz_right](#), 10

[sd_off](#), 11

[sd_on](#), 12

[sim_gmat](#), 18

[simico_fit_null](#), 12

[simico_gen_dat](#), 14

[simico_out](#), 16

[ss_fd](#), 19

[ss_sd](#), 20

[st_off](#), 21

[surv_diff](#), 22

[surv_left](#), 22

[surv_right](#), 23

[without_one_phen](#), 23

[without_two_phen](#), 24