

Package ‘FastJM’

November 8, 2025

Type Package

Title Semi-Parametric Joint Modeling of Longitudinal and Survival Data

Version 1.5.3

Date 2025-11-08

Maintainer Shanpeng Li <lishanpeng0913@ucla.edu>

Encoding UTF-8

Description A joint model for large-scale, competing risks time-to-event data with singular or multiple longitudinal biomarkers, implemented with the efficient algorithms developed by Li and colleagues (2022) <doi:10.1155/2022/1362913> and <doi:10.48550/arXiv.2506.12741>.

The time-to-event data is modelled using a (cause-specific) Cox proportional hazards regression model with time-fixed covariates. The longitudinal biomarkers are modelled using a linear mixed effects model. The association between the longitudinal submodel and the survival submodel is captured through shared random effects. It allows researchers to analyze large-scale data to model biomarker trajectories, estimate their effects on event outcomes, and dynamically predict future events from patients' past histories. A function for simulating survival and longitudinal data for multiple biomarkers is also included alongside built-in datasets.

License GPL (>= 3)

NeedsCompilation yes

Imports Rcpp (>= 1.0.7), dplyr, nlme, caret, timeROC, future, future.apply, rlang (>= 0.4.11)

LinkingTo Rcpp, RcppEigen

Depends R (>= 3.5.0), survival, utils, MASS, statmod, magrittr

RoxygenNote 7.3.2

LazyData true

Suggests testthat (>= 3.0.0), spelling

Language en-US

Config/testthat/edition 3

Author Shanpeng Li [aut, cre],
 Ning Li [ctb],
 Emily Ouyang [ctb],
 Hong Wang [ctb],
 Jin Zhou [ctb],
 Hua Zhou [ctb],
 Gang Li [ctb]

Repository CRAN

Date/Publication 2025-11-08 14:00:02 UTC

Contents

anova	3
AUCjmcS	4
cdata	5
fitted	6
fixef	7
jmcS	8
MAEQjmcS	12
mvdata	13
mvjmcS	14
mvdata	17
PEjmcS	17
plot.jmcS	19
print	20
print.survfitjmcS	21
print.survfitmvjmcS	21
ranef	22
residuals	23
simmvJMdata	24
summary	25
summary.AUCjmcS	26
summary.MAEQjmcS	27
summary.PEjmcS	27
survfitjmcS	28
survfitmvjmcS	30
vcov	31
ydata	32

Index	33
--------------	-----------

anova*Anova Method for Fitted Joint Models*

Description

Performs a likelihood ratio test between two nested joint models.

Usage

```
## S3 method for class 'jmcs'
anova(object, object2, digits = 4, ...)
```

Arguments

object	an object inheriting from class <code>jmcs</code> , nested in <code>object2</code> .
object2	an object inheriting from class <code>jmcs</code> .
digits	the number of significant digits to use when printing. Default is 4.
...	further arguments passed to or from other methods.

Value

A table to summarize the likelihood ratio test.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[jmcs](#)

Examples

```
# Fit a joint model
fit <- jmcs(ydata = ydata, cdata = cdata,
  long.formula = response ~ time + x1,
  surv.formula = Surv(surv, failure_type) ~ x1 + x2,
  random = ~ time | ID)

fit2 <- jmcs(ydata = ydata, cdata = cdata,
  long.formula = response ~ time + gender + x1 + race,
  surv.formula = Surv(surv, failure_type) ~ x1 + gender + x2 + race,
  random = ~ time | ID)

anova(fit, fit2)
```

AUCjmcS

*Time-dependent AUC/Cindex for joint models***Description**

Time-dependent AUC/Cindex for joint models

Usage

```

AUCjmcS(
  seed = 100,
  object,
  landmark.time = NULL,
  horizon.time = NULL,
  obs.time = NULL,
  method = c("Laplace", "GH"),
  quadpoint = NULL,
  maxiter = NULL,
  n.cv = 3,
  survinitial = TRUE,
  initial.para = FALSE,
  LOCF = FALSE,
  LOCFcovariate = NULL,
  clongdata = NULL,
  metric = c("AUC", "Cindex"),
  ...
)

```

Arguments

<code>seed</code>	a numeric value of seed to be specified for cross validation.
<code>object</code>	object of class 'jmcS'.
<code>landmark.time</code>	a numeric value of time for which dynamic prediction starts..
<code>horizon.time</code>	a numeric vector of future times for which predicted probabilities are to be computed.
<code>obs.time</code>	a character string of specifying a longitudinal time variable.
<code>method</code>	estimation method for predicted probabilities. If Laplace, then the empirical empirical estimates of random effects is used. If GH, then the pseudo-adaptive Gauss-Hermite quadrature is used.
<code>quadpoint</code>	the number of pseudo-adaptive Gauss-Hermite quadrature points if method = "GH".
<code>maxiter</code>	the maximum number of iterations of the EM algorithm that the function will perform. Default is 10000.
<code>n.cv</code>	number of folds for cross validation. Default is 3.

survinitial	Fit a Cox model to obtain initial values of the parameter estimates. Default is TRUE.
initial.para	Initial guess of parameters for cross validation. Default is FALSE.
LOCF	a logical value to indicate whether the last-observation-carried-forward approach applies to prediction. If TRUE, then LOCFcovariate and clongdata must be specified to indicate which time-dependent survival covariates are included for dynamic prediction. Default is FALSE.
LOCFcovariate	a vector of string with time-dependent survival covariates if LOCF = TRUE. Default is NULL.
clongdata	a long format data frame where time-dependent survival covariates are incorporated. Default is NULL.
metric	a string to indicate which metric is used.
...	Further arguments passed to or from other methods.

Value

a list of matrices with conditional probabilities for subjects.

Author(s)

Shanpeng Li <lshanpeng0913@ucla.edu>

See Also

[jmcs](#), [survfitjmcs](#)

cdata

Simulated competing risks data correlated with ydata

Description

The cdata data frame has 1000 rows and 7 columns.

Usage

```
data(cdata)
```

Format

This data frame contains the following columns:

ID patient identifier.

surv event time.

failure_type event indicator. 0 denotes censoring, 1 risk 1, and 2 risk 2.

x1 continuous variable.

x2 treatment indicator. 0 denotes the placebo group and 1 the treatment group.

gender gender indicator.

race race indicator.

fitted	<i>Fitted values for joint models</i>
--------	---------------------------------------

Description

Extract fitted values for joint models.

Usage

```
## S3 method for class 'jmcs'
fitted(
  object,
  type = c("Marginal", "Subject"),
  process = c("Longitudinal", "Event"),
  ...
)
```

Arguments

object	an object inheriting from class jmcs.
type	for which type of fitted values to calculate.
process	for which sub-model to calculate the fitted values.
...	further arguments passed to or from other methods.

Value

a numeric vector of fitted values.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

Examples

```
fit <- jmcs(ydata = ydata, cdata = cdata,
  long.formula = response ~ time + gender + x1 + race,
  surv.formula = Surv(surv, failure_type) ~ x1 + gender + x2 + race,
  random = ~ time| ID)

# fitted for the longitudinal process
head(cbind(
  "Marg" = fitted(fit, type = "Marginal", process = "Longitudinal"),
  "Subj" = fitted(fit, type = "Subject", process = "Longitudinal")
))
```

```

))
# fitted for the levent process - marginal survival function
head(fitted(fit, type = "Marginal", process = "Event"))

```

fixef

Estimated coefficients estimates for joint models

Description

Extracts the fixed effects for a fitted joint model.

Usage

```
fixef(object, process = c("Longitudinal", "Event"), ...)
```

Arguments

object	an object inheriting from class <code>jmcs</code> or <code>mvjmcs</code> .
process	for which sub-model to extract the estimated coefficients.
...	further arguments passed to or from other methods.

Value

A numeric vector or a list of the estimated parameters for the fitted model.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

Examples

```

# a joint model fit
fit <- jmcs(ydata = ydata, cdata = cdata,
  long.formula = response ~ time + gender + x1 + race,
  surv.formula = Surv(surv, failure_type) ~ x1 + gender + x2 + race,
  random = ~ time | ID)

# fixed effects for the longitudinal process
fixef(fit, process = "Longitudinal")
# fixed effects for the event process
fixef(fit, process = "Event")

```

jmcs

*Joint modeling of longitudinal continuous data and competing risks***Description**

Joint modeling of longitudinal continuous data and competing risks

Usage

```
jmcs(
  ydata,
  cdata,
  long.formula,
  random = NULL,
  surv.formula,
  REML = TRUE,
  quadpoint = NULL,
  maxiter = 10000,
  print.para = FALSE,
  initial.para = NULL,
  survinitial = TRUE,
  tol = 1e-04,
  method = "pseudo-adaptive",
  opt = "nlminb"
)
```

Arguments

ydata	a longitudinal data frame in long format.
cdata	a survival data frame with competing risks or single failure. Each subject has one data entry.
long.formula	a formula object with the response variable and fixed effects covariates to be included in the longitudinal sub-model.
random	a one-sided formula object describing the random effects part of the longitudinal sub-model. For example, fitting a random intercept model takes the form $\sim 1 ID$. Alternatively. Fitting a random intercept and slope model takes the form $\sim x1 + \dots + xn ID$.
surv.formula	a formula object with the survival time, event indicator, and the covariates to be included in the survival sub-model.
REML	a logic object that indicates the use of REML estimator. Default is TRUE.
quadpoint	the number of pseudo-adaptive Gauss-Hermite quadrature points. to be chosen for numerical integration. Default is 6 which produces stable estimates in most dataframes.
maxiter	the maximum number of iterations of the EM algorithm that the function will perform. Default is 10000.

<code>print.para</code>	Print detailed information of each iteration. Default is FALSE, i.e., not to print the iteration details.
<code>initial.para</code>	a list of initialized parameters for EM iteration. Default is NULL.
<code>survinitial</code>	Fit a Cox model to obtain initial values of the parameter estimates. Default is TRUE.
<code>tol</code>	Tolerance parameter. Default is 0.0001.
<code>method</code>	Method for proceeding numerical integration in the E-step. Default is pseudo-adaptive.
<code>opt</code>	Optimization method to fit a linear mixed effects model, either <code>nlminb</code> (default) or <code>optim</code> .

Value

Object of class `jmcs` with elements

<code>beta</code>	the vector of fixed effects for the linear mixed effects model.
<code>gamma1</code>	the vector of fixed effects for type 1 failure for the survival model.
<code>gamma2</code>	the vector of fixed effects for type 2 failure for the survival model. Valid only if <code>CompetingRisk = TRUE</code> .
<code>nu1</code>	the vector of association parameter(s) for type 1 failure.
<code>nu2</code>	the vector of association parameter(s) for type 2 failure. Valid only if <code>CompetingRisk = TRUE</code> .
<code>H01</code>	the matrix that collects baseline hazards evaluated at each uncensored event time for type 1 failure. The first column denotes uncensored event times, the second column the number of events, and the third columns the hazards obtained by Breslow estimator.
<code>H02</code>	the matrix that collects baseline hazards evaluated at each uncensored event time for type 2 failure. The data structure is the same as <code>H01</code> . Valid only if <code>CompetingRisk = TRUE</code> .
<code>Sig</code>	the variance-covariance matrix of the random effects.
<code>sigma</code>	the variance of the measurement error for the linear mixed effects model.
<code>iter</code>	the total number of iterations until convergence.
<code>convergence</code>	convergence identifier: 1 corresponds to successful convergence, whereas 0 to a problem (i.e., when 0, usually more iterations are required).
<code>vcov</code>	the variance-covariance matrix of all the fixed effects for both models.
<code>sebeta</code>	the standard error of <code>beta</code> .
<code>segamma1</code>	the standard error of <code>gamma1</code> .
<code>segamma2</code>	the standard error of <code>gamma2</code> . Valid only if <code>CompetingRisk = TRUE</code> .
<code>senu1</code>	the standard error of <code>nu1</code> .
<code>senu2</code>	the standard error of <code>nu2</code> . Valid only if <code>CompetingRisk = TRUE</code> .
<code>seSig</code>	the vector of standard errors of covariance of random effects.

<code>seigma</code>	the standard error of variance of measurement error for the linear mixed effects model.
<code>loglike</code>	the log-likelihood value.
<code>fitted</code>	a list with the fitted values: resid the vector of estimated residuals for the linear mixed effects model. fitted the vector of fitted values for the linear mixed effects model. fittedmar the vector of marginal fitted values for the linear mixed effects model. residmar the vector of estimated marginal residuals for the linear mixed effects model.
<code>fittedSurv</code>	the estimated survival rate evaluated at each uncensored event time.
<code>FUNB</code>	the estimated random effects for each subject.
<code>CompetingRisk</code>	logical value; TRUE if a competing event are accounted for.
<code>quadpoint</code>	the number of Gauss Hermite quadrature points used for numerical integration.
<code>ydata</code>	the input longitudinal dataset for fitting a joint model. It has been re-ordered in accordance with descending observation times in <code>cdata</code> .
<code>cdata</code>	the input survival dataset for fitting a joint model. It has been re-ordered in accordance with descending observation times.
<code>PropEventType</code>	a frequency table of number of events.
<code>LongitudinalSubmodel</code>	the component of the <code>long.</code> formula.
<code>SurvivalSubmodel</code>	the component of the <code>surv.</code> formula.
<code>random</code>	the component of the <code>random.</code>
<code>call</code>	the matched call.
<code>Quad.method</code>	the quadrature rule used for integration. If pseudo-adaptive quadrature rule is used, then return pseudo-adaptive. Otherwise return standard.
<code>id</code>	the grouping vector for the longitudinal outcome.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[ranef](#), [fixef](#), [fitted.jmcs](#), [residuals.jmcs](#), [survfitjmcs](#), [plot.jmcs](#), [vcov.jmcs](#)

Examples

```
require(FastJM)
require(survival)
# Load a simulated longitudinal dataset
data(ydata)
# Load a simulated survival dataset with two competing events
data(cdata)
```

```

# Fit a joint model
fit <- jmcs(ydata = ydata, cdata = cdata,
           long.formula = response ~ time + gender + x1 + race,
           surv.formula = Surv(surv, failure_type) ~ x1 + gender + x2 + race,
           random = ~ time| ID)

fit
# Extract the parameter estimates of longitudinal sub-model fixed effects
fixef(fit, process = "Longitudinal")
# Extract the parameter estimates of survival sub-model fixed effects
fixef(fit, process = "Event")
# Obtain the random effects estimates for first 6 subjects
head(ranef(fit))
# Obtain the variance-covariance matrix of all parameter estimates
vcov(fit)
# Obtain the result summaries of the joint model fit
summary(fit, process = "Longitudinal")
summary(fit, process = "Event")
# Prediction of cumulative incidence for competing risks data
# Predict the conditional probabilities for two patients who are alive (censored)
ND <- ydata[ydata$ID %in% c(419, 218), ]
ID <- unique(ND$ID)
NDc <- cdata[cdata$ID %in% ID, ]
survfit <- survfitjmcs(fit,
                      ynewdata = ND,
                      cnewdata = NDc,
                      u = seq(3, 4.8, by = 0.2),
                      method = "GH",
                      obs.time = "time")

survfit
PE <- PEjmcs(fit, seed = 100, landmark.time = 3, horizon.time = c(3.6, 4, 4.4),
             obs.time = "time", method = "GH",
             quadpoint = NULL, maxiter = 1000, n.cv = 3,
             survinitial = TRUE)
Brier <- summary(PE, error = "Brier")
Brier

MAEQ <- MAEQjmcs(fit, seed = 100, landmark.time = 3, horizon.time = c(3.6, 4, 4.4),
                 obs.time = "time", method = "GH",
                 quadpoint = NULL, maxiter = 1000, n.cv = 3,
                 survinitial = TRUE)
APE <- summary(MAEQ, digits = 3)
APE

## evaluate prediction accuracy of fitted joint model using cross-validated mean AUC
AUC <- AUCjmcs(fit, seed = 100, landmark.time = 3, horizon.time = c(3.6, 4, 4.4),
               obs.time = "time", method = "GH",
               quadpoint = NULL, maxiter = 1000, n.cv = 3, metric = "AUC")
summary(AUC, digits = 3)

## or using cross-validated mean C-index
Cindex <- AUCjmcs(fit, seed = 100, landmark.time = 3, horizon.time = c(3.6, 4, 4.4),
                  obs.time = "time", method = "GH",
                  quadpoint = NULL, maxiter = 1000, n.cv = 3, metric = "Cindex")

```

```
summary(Cindex, digits = 3)
```

MAEQjmcs	<i>A metric of prediction accuracy of joint model by comparing the predicted risk with the empirical risks stratified on different predicted risk group.</i>
----------	--

Description

A metric of prediction accuracy of joint model by comparing the predicted risk with the empirical risks stratified on different predicted risk group.

Usage

```
MAEQjmcs(
  seed = 100,
  object,
  landmark.time = NULL,
  horizon.time = NULL,
  obs.time = NULL,
  method = c("Laplace", "GH"),
  quadpoint = NULL,
  maxiter = 1000,
  n.cv = 3,
  survinitial = TRUE,
  quantile.width = 0.25,
  initial.para = FALSE,
  LOCF = FALSE,
  LOCFcovariate = NULL,
  clongdata = NULL,
  ...
)
```

Arguments

<code>seed</code>	a numeric value of seed to be specified for cross validation.
<code>object</code>	object of class 'jmcs'.
<code>landmark.time</code>	a numeric value of time for which dynamic prediction starts..
<code>horizon.time</code>	a numeric vector of future times for which predicted probabilities are to be computed.
<code>obs.time</code>	a character string of specifying a longitudinal time variable.
<code>method</code>	estimation method for predicted probabilities. If Laplace, then the empirical empirical estimates of random effects is used. If GH, then the pseudo-adaptive Gauss-Hermite quadrature is used.

quadpoint	the number of pseudo-adaptive Gauss-Hermite quadrature points if method = "GH".
maxiter	the maximum number of iterations of the EM algorithm that the function will perform. Default is 10000.
n.cv	number of folds for cross validation. Default is 3.
survinitial	Fit a Cox model to obtain initial values of the parameter estimates. Default is TRUE.
quantile.width	a numeric value of width of quantile to be specified. Default is 0.25.
initial.para	Initial guess of parameters for cross validation. Default is FALSE.
LOCF	a logical value to indicate whether the last-observation-carried-forward approach applies to prediction. If TRUE, then LOCFcovariate and clongdata must be specified to indicate which time-dependent survival covariates are included for dynamic prediction. Default is FALSE.
LOCFcovariate	a vector of string with time-dependent survival covariates if LOCF = TRUE. Default is NULL.
clongdata	a long format data frame where time-dependent survival covariates are incorporated. Default is NULL.
...	Further arguments passed to or from other methods.

Value

a list of matrices with conditional probabilities for subjects.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[jmcs](#), [survfitjmcs](#)

mvcddata

Simulated competing risks data correlated with mvydata

Description

The mvcddata data frame has 500 rows and 5 columns.

Usage

```
data(mvcddata)
```

Format

This data frame contains the following columns:

ID patient identifier.

survtime event time.

cmprsk event indicator. 0 denotes censoring, 1 risk 1, and 2 risk 2.

X21 X21.

X22 X22.

mvjmcs

Joint modeling of multivariate longitudinal and competing risks data

Description

Joint modeling of multivariate longitudinal continuous data and competing risks

Usage

```
mvjmcs(
  ydata,
  cdata,
  long.formula,
  random = NULL,
  surv.formula,
  maxiter = 10000,
  opt = "nlminb",
  tol = 0.005,
  print.para = TRUE,
  initial.para = NULL,
  cpu.cores = NULL
)
```

Arguments

ydata	A longitudinal data frame in long format.
cdata	A survival data frame with competing risks or single failure. Each subject has one data entry.
long.formula	A list of formula objects specifying fixed effects for each longitudinal outcome.
random	A formula or list of formulas describing random effects structures (e.g., $\sim 1 ID$).
surv.formula	A formula for the survival sub-model, including survival time and event indicator.
maxiter	Maximum number of EM iterations. Default is 10000.
opt	Optimization method for mixed model. Default is "nlminb".

tol	Convergence tolerance for EM algorithm. Default is 0.0001.
print.para	Logical; if TRUE, prints parameter values at each iteration.
initial.para	Optional list of initialized parameters. Default is NULL.
cpu.cores	Number of CPU cores for parallel computation. Default is 1.

Details

Function fits a joint model for multiple longitudinal outcomes and competing risks using a fast EM algorithm.

Value

Object of class `mvjmcs` with elements

beta	the vector of all biomarker-specific fixed effects for the linear mixed effects sub-models.
betaList	the list of biomarker-specific fixed effects for the linear mixed effects sub-model.
gamma1	the vector of fixed effects for type 1 failure for the survival model.
gamma2	the vector of fixed effects for type 2 failure for the survival model. Valid only if <code>CompetingRisk = TRUE</code> .
alpha1	the vector of association parameter(s) for type 1 failure.
alpha2	the vector of association parameter(s) for type 2 failure. Valid only if <code>CompetingRisk = TRUE</code> .
H01	the matrix that collects baseline hazards evaluated at each uncensored event time for type 1 failure. The first column denotes uncensored event times, the second column the number of events, and the third columns the hazards obtained by Breslow estimator.
H02	the matrix that collects baseline hazards evaluated at each uncensored event time for type 2 failure. The data structure is the same as H01. Valid only if <code>CompetingRisk = TRUE</code> .
Sig	the variance-covariance matrix of the random effects.
sigma	the vector of the variance of the biomarker-specific measurement error for the linear mixed effects sub-models.
iter	the total number of iterations until convergence.
convergence	convergence identifier: 1 corresponds to successful convergence, whereas 0 to a problem (i.e., when 0, usually more iterations are required).
vcov	the variance-covariance matrix of all the fixed effects for both models.
FisherInfo	the Empirical Fisher information matrix.
Score	a matrix of the score function for all subjects.
sebeta	the standard error of beta.
segamma1	the standard error of gamma1.
segamma2	the standard error of gamma2. Valid only if <code>CompetingRisk = TRUE</code> .
sealpha1	the standard error of nu1.

sealpha2	the standard error of nu2. Valid only if CompetingRisk = TRUE.
seSig	the vector of standard errors of covariance of random effects.
sesigma	the standard error of variance of biomarker-specific measurement error for the linear mixed effects sub-models.
pos.mode	the posterior mode of the conditional distribution of random effects.
pos.cov	the posterior covariance of the conditional distribution of random effects.
CompetingRisk	logical value; TRUE if a competing event are accounted for.
ydata	the input longitudinal dataset for fitting a joint model. It has been re-ordered in accordance with descending observation times in cdata.
cdata	the input survival dataset for fitting a joint model. It has been re-ordered in accordance with descending observation times.
PropEventType	a frequency table of number of events.
LongitudinalSubmodel	the component of the long.formula.
SurvivalSubmodel	the component of the surv.formula.
random	the component of the random.
call	the matched call.
id	the grouping vector for the longitudinal outcome.
opt	the numerical optimizer for obtaining the initial guess of the parameters in the linear mixed effects sub-models.
runtime	the total computation time.

Examples

```
require(FastJM)
require(survival)
require(future)
require(future.apply)

data(mvcdata)
data(mvydata)

# Fit joint model with two biomarkers
fit <- mvjmcs(ydata = mvydata, cdata = mvcdata,
             long.formula = list(Y1 ~ X11 + X12 + time,
                                Y2 ~ X11 + X12 + time),
             random = list(~ time | ID,
                           ~ 1 | ID),
             surv.formula = Surv(survtime, cmprsk) ~ X21 + X22, maxiter = 1000, opt = "optim",
             tol = 1e-3, print.para = FALSE)
fit

# Extract the parameter estimates of longitudinal sub-model fixed effects
```

```

fixef(fit, process = "Longitudinal")

# Extract the parameter estimates of survival sub-model fixed effects
fixef(fit, process = "Event")

# Obtain the random effects estimates for first 6 subjects
head(ranef(fit))

```

mvydata

Simulated bivariate longitudinal data

Description

The mvydata data frame has 4060 rows and 6 columns.

Usage

```
data(mvydata)
```

Format

This data frame contains the following columns:

ID patient identifier.

time visit time.

Y1 response variable of biomarker 1.

Y2 response variable of biomarker 2.

X11 X11.

X12 X12.

PEjmcs

A metric of prediction accuracy of joint model by comparing the predicted risk with the counting process.

Description

A metric of prediction accuracy of joint model by comparing the predicted risk with the counting process.

Usage

```
PEjmcs(
  seed = 100,
  object,
  landmark.time = NULL,
  horizon.time = NULL,
  obs.time = NULL,
  method = c("Laplace", "GH"),
  quadpoint = NULL,
  maxiter = NULL,
  n.cv = 3,
  survinitial = TRUE,
  initial.para = FALSE,
  LOCF = FALSE,
  LOCFcovariate = NULL,
  clongdata = NULL,
  ...
)
```

Arguments

<code>seed</code>	a numeric value of seed to be specified for cross validation.
<code>object</code>	object of class 'jmcs'.
<code>landmark.time</code>	a numeric value of time for which dynamic prediction starts..
<code>horizon.time</code>	a numeric vector of future times for which predicted probabilities are to be computed.
<code>obs.time</code>	a character string of specifying a longitudinal time variable.
<code>method</code>	estimation method for predicted probabilities. If Laplace, then the empirical empirical estimates of random effects is used. If GH, then the pseudo-adaptive Gauss-Hermite quadrature is used.
<code>quadpoint</code>	the number of pseudo-adaptive Gauss-Hermite quadrature points if method = "GH".
<code>maxiter</code>	the maximum number of iterations of the EM algorithm that the function will perform. Default is 10000.
<code>n.cv</code>	number of folds for cross validation. Default is 3.
<code>survinitial</code>	Fit a Cox model to obtain initial values of the parameter estimates. Default is TRUE.
<code>initial.para</code>	Initial guess of parameters for cross validation. Default is FALSE.
<code>LOCF</code>	a logical value to indicate whether the last-observation-carried-forward approach applies to prediction. If TRUE, then LOCFcovariate and clongdata must be specified to indicate which time-dependent survival covariates are included for dynamic prediction. Default is FALSE.
<code>LOCFcovariate</code>	a vector of string with time-dependent survival covariates if LOCF = TRUE. Default is NULL.

`clongdata` a long format data frame where time-dependent survival covariates are incorporated. Default is NULL.

`...` Further arguments passed to or from other methods.

Value

a list of matrices with conditional probabilities for subjects.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[jmcs](#), [survfitjmcs](#)

plot.jmcs	<i>Fitted values for joint models</i>
-----------	---------------------------------------

Description

Plot Diagnostics for Joint Models.

Usage

```
## S3 method for class 'jmcs'
plot(x, add.smooth = getOption("add.smooth"), ...)
```

Arguments

`x` x of class 'jmcs'.

`add.smooth` logical; if TRUE a smooth line is superimposed in the "Residuals vs Fitted" plot.

`...` further arguments passed to or from other methods.

Value

The first two plots are longitudinal sub-model diagnostics and the last two are marginal survival function and marginal cumulative hazard.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

Examples

```
fit <- jmcs(ydata = ydata, cdata = cdata,
            long.formula = response ~ time + gender + x1 + race,
            surv.formula = Surv(surv, failure_type) ~ x1 + gender + x2 + race,
            random = ~ time | ID)

oldpar <- par(mfrow = c(2, 2))
plot(fit)
par(oldpar)
```

print	<i>Print jmcs</i>
-------	-------------------

Description

Print jmcs
 Print mvjmcs

Usage

```
## S3 method for class 'jmcs'
print(x, digits = 4, ...)

## S3 method for class 'mvjmcs'
print(x, digits = 4, ...)
```

Arguments

x	Object of class 'mvjmcs'.
digits	the number of significant digits to use when printing.
...	Further arguments passed to or from other methods.

Value

a summary of data, joint model, log likelihood, and parameter estimates.
 a summary of data, joint model, log likelihood, and parameter estimates.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[jmcs](#)
[mvjmcs](#)

print.survfitjmc	<i>Print survfitjmc</i>
------------------	-------------------------

Description

Print survfitjmc

Usage

```
## S3 method for class 'survfitjmc'  
print(x, ...)
```

Arguments

x	x of class 'survfitjmc'.
...	Further arguments passed to or from other methods.

Value

a list of matrices with conditional probabilities for subjects.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[jmc](#), [survfitjmc](#)

print.survfitmvjmc	<i>Print survfitmvjmc</i>
--------------------	---------------------------

Description

Print survfitmvjmc

Usage

```
## S3 method for class 'survfitmvjmc'  
print(x, ...)
```

Arguments

x	x of class 'survfitmvjmc'.
...	Further arguments passed to or from other methods.

Value

a list of matrices with conditional probabilities for subjects.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[mvjmcs](#), [survfitmvjmcs](#)

ranef

Random effects estimates for joint models

Description

Extracts the posterior mean of the random effects for a fitted joint model.

Usage

```
ranef(object, ...)
```

Arguments

object an object inheriting from class `jmcs` or `mvjmcs`.
... further arguments passed to or from other methods.

Value

a matrix of random effects estimates.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[jmcs](#)

Examples

```
# a joint model fit
fit <- jmcs(ydata = ydata, cdata = cdata,
            long.formula = response ~ time + gender + x1 + race,
            surv.formula = Surv(surv, failure_type) ~ x1 + gender + x2 + race,
            random = ~ time | ID)

# extract random effects estimates
head(ranef(fit))
```

residuals*Residuals for joint models*

Description

Extract residuals for joint models.

Usage

```
## S3 method for class 'jmcs'
residuals(object, type = c("Marginal", "Subject"), ...)
```

Arguments

object	an object inheriting from class <code>jmcs</code> .
type	what type of residuals to calculate.
...	further arguments passed to or from other methods.

Value

a vector of residuals of the longitudinal sub-model.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[jmcs](#)

Examples

```
# a joint model fit
fit <- jmcs(ydata = ydata, cdata = cdata,
           long.formula = response ~ time + gender + x1 + race,
           surv.formula = Surv(surv, failure_type) ~ x1 + gender + x2 + race,
           random = ~ time | ID)

# residuals of the longitudinal sub-model
head(cbind(
  "Marg" = residuals(fit, type = "Marginal"),
  "Subj" = residuals(fit, type = "Subject")
))
```

simmvJMdata

*Joint modeling of multivariate longitudinal and competing risks data***Description**

Data simulation from the joint model of multivariate longitudinal biomarkers and time-to-event data

Usage

```
simmvJMdata(
  seed = 100,
  N = 200,
  increment = 0.7,
  beta = list(beta1 = c(5, 1.5, 2, 1), beta2 = c(10, 1, 2, 1)),
  sigma = c(0.5, 0.5),
  gamma1 = c(1, 0.5),
  gamma2 = c(-0.5, 0.5),
  alpha1 = list(alpha11 = c(0.5, 0.7), alpha12 = c(-0.5, 0.5)),
  alpha2 = list(alpha21 = c(0.5, 0.7), alpha22 = c(-0.5, 0.5)),
  lambda1 = 0.05,
  lambda2 = 0.025,
  CL = 5,
  CU = 10,
  covb = diag(rep(1, 4)),
  missprob = 0,
  CR = TRUE
)
```

Arguments

seed	a random seed number specified for simulating a joint model dataset.
N	an integer to specify the sample size.
increment	a scalar to specify the increment of visit time for longitudinal measurements.
beta	a list of true parameters for the linear mixed effects sub-models. Each component of the list correspond to a specific biomarker.
sigma	a vector of true error variance for all biomarkers.
gamma1	a vector of true parameters of survival fixed effects for failure 1.
gamma2	a vector of true parameters of survival fixed effects for failure 2.
alpha1	a list of true parameters for the association parameters for failure 1. Each component of the list correspond to a specific biomarker.
alpha2	a list of true parameters for the association parameters for failure 2. Each component of the list correspond to a specific biomarker.
lambda1	the baseline hazard rate of failure 1. An exponential distribution with a rate parameter of lambda1 is assumed.

lambda2	the baseline hazard rate of failure 2. An exponential distribution with a rate parameter of lambda2 is assumed.
CL	a lower limit of a uniform distribution to be specified for the censoring time.
CU	an upper limit of a uniform distribution to be specified for the censoring time.
covb	a matrix of variance-covariance matrix of random effects.
missprob	a scalar (ranging from 0 to 1) to specify the probability of missing longitudinal observations. Default is 0.
CR	logical; if TRUE, simulate competing risks time-to-event data with 2 failures. Default is TRUE.

Value

a list of datasets for both longitudinal and survival data with the elements

mvydata	a long-format data frame of longitudinal data.
mvcddata	a dataframe of survival data.

summary

*Anova Method for Fitted Joint Models***Description**

Produce result summaries of a joint model fit.

Produce result summaries of a joint model fit.

Usage

```
## S3 method for class 'jmcs'
summary(object, process = c("Longitudinal", "Event"), digits = 4, ...)

## S3 method for class 'mvjmcs'
summary(object, process = c("Longitudinal", "Event"), digits = 4, ...)
```

Arguments

object	an object inheriting from class mvjmcs.
process	for which model (i.e., longitudinal model or survival model) to extract the estimated coefficients.
digits	the number of significant digits to use when printing. Default is 4.
...	further arguments passed to or from other methods.

Value

A table to summarize the model results.

A table to summarize the model results.

See Also[jmc](#)[mvjmc](#)

`summary.AUCjmc`*Print AUCjmc*

Description

Print AUCjmc

Usage

```
## S3 method for class 'AUCjmc'  
summary(object, digits = 4, ...)
```

Arguments

<code>object</code>	object of class 'AUCjmc'.
<code>digits</code>	number of digits of decimal to be printed.
<code>...</code>	Further arguments passed to or from other methods.

Value

a list of matrices with conditional probabilities for subjects.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also[jmc](#), [survfitjmc](#)

summary.MAEQjmc	<i>Print MAEQjmc</i>
-----------------	----------------------

Description

Print MAEQjmc

Usage

```
## S3 method for class 'MAEQjmc'  
summary(object, digits = 3, ...)
```

Arguments

object	object of class 'MAEQjmc'.
digits	number of decimal points to be rounded.
...	Further arguments passed to or from other methods.

Value

a list of matrices with conditional probabilities for subjects.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[jmc](#), [survfitjmc](#)

summary.PEjmc	<i>Print PEjmc</i>
---------------	--------------------

Description

Print PEjmc

Usage

```
## S3 method for class 'PEjmc'  
summary(object, error = c("MAE", "Brier"), ...)
```

Arguments

object	object of class 'PEjmcs'.
error	a character string that specifies the loss function.
...	Further arguments passed to or from other methods.

Value

a list of matrices with conditional probabilities for subjects.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[jmcs](#), [survfitjmcs](#)

survfitjmcs

Prediction in Joint Models

Description

This function computes the conditional probability of surviving later times than the last observed time for which a longitudinal measurement was available.

Usage

```
survfitjmcs(
  object,
  seed = 100,
  ynewdata = NULL,
  cnewdata = NULL,
  u = NULL,
  Last.time = NULL,
  obs.time = NULL,
  LOCF = FALSE,
  LOCFcovariate = NULL,
  clongdata = NULL,
  method = c("Laplace", "GH"),
  quadpoint = NULL,
  ...
)
```

Arguments

<code>object</code>	an object inheriting from class <code>jmcs</code> .
<code>seed</code>	a random seed number to proceed Monte Carlo simulation. Default is 100.
<code>ynewdata</code>	a data frame that contains the longitudinal and covariate information for the subjects for which prediction of survival probabilities is required.
<code>cnewdata</code>	a data frame that contains the survival and covariate information for the subjects for which prediction of survival probabilities is required.
<code>u</code>	a numeric vector of times for which prediction survival probabilities are to be computed.
<code>Last.time</code>	a numeric vector or character string. This specifies the known time at which each of the subjects in <code>cnewdata</code> was known to be alive. If <code>NULL</code> , then this is automatically taken as the survival time of each subject. If a numeric vector, then it is assumed to be greater than or equals to the last available longitudinal time point for each subject. If a character string, then it should be a variable in <code>cnewdata</code> .
<code>obs.time</code>	a character string of specifying a longitudinal time variable in <code>ynewdata</code> .
<code>LOCF</code>	a logical value to indicate whether the last-observation-carried-forward approach applies to prediction. If <code>TRUE</code> , then <code>LOCFcovariate</code> and <code>clongdata</code> must be specified to indicate which time-dependent survival covariates are included for dynamic prediction. Default is <code>FALSE</code> .
<code>LOCFcovariate</code>	a vector of string with time-dependent survival covariates if <code>LOCF = TRUE</code> . Default is <code>NULL</code> .
<code>clongdata</code>	a long format data frame where time-dependent survival covariates are incorporated. Default is <code>NULL</code> .
<code>method</code>	a character string specifying the type of probability approximation; if <code>Laplace</code> , then a first order estimator is computed. If <code>GH</code> , then the standard Gauss-Hermite quadrature is used instead.
<code>quadpoint</code>	number of quadrature points used for estimating conditional probabilities when <code>method = "GH"</code> . Default is <code>NULL</code> . If <code>method = "GH"</code> , then use the same amount of quadrature points obtained from object.
<code>...</code>	further arguments passed to or from other methods.

Value

a list of matrices with conditional probabilities for subjects.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[jmcs](#)

survfitmvjmcs

*Prediction in Joint Models***Description**

This function computes the conditional probability of surviving later times than the last observed time for which a longitudinal measurement was available.

Usage

```
survfitmvjmcs(
  object,
  seed = 100,
  ynewdata = NULL,
  cnewdata = NULL,
  u = NULL,
  Last.time = NULL,
  obs.time = NULL,
  LOCF = FALSE,
  LOCFcovariate = NULL,
  clongdata = NULL,
  ...
)
```

Arguments

<code>object</code>	an object inheriting from class <code>mvjmcs</code> .
<code>seed</code>	a random seed number to proceed Monte Carlo simulation. Default is 100.
<code>ynewdata</code>	a data frame that contains the longitudinal and covariate information for the subjects for which prediction of survival probabilities is required.
<code>cnewdata</code>	a data frame that contains the survival and covariate information for the subjects for which prediction of survival probabilities is required.
<code>u</code>	a numeric vector of times for which prediction survival probabilities are to be computed.
<code>Last.time</code>	a numeric vector or character string. This specifies the known time at which each of the subjects in <code>cnewdata</code> was known to be alive. If <code>NULL</code> , then this is automatically taken as the survival time of each subject. If a numeric vector, then it is assumed to be greater than or equals to the last available longitudinal time point for each subject. If a character string, then it should be a variable in <code>cnewdata</code> .
<code>obs.time</code>	a character string of specifying a longitudinal time variable in <code>ynewdata</code> .
<code>LOCF</code>	a logical value to indicate whether the last-observation-carried-forward approach applies to prediction. If <code>TRUE</code> , then <code>LOCFcovariate</code> and <code>clongdata</code> must be specified to indicate which time-dependent survival covariates are included for dynamic prediction. Default is <code>FALSE</code> .

`LOCFcovariate` a vector of string with time-dependent survival covariates if `LOCF = TRUE`. Default is `NULL`.

`clongdata` a long format data frame where time-dependent survival covariates are incorporated. Default is `NULL`.

`...` further arguments passed to or from other methods.

Value

a list of matrices with conditional probabilities for subjects.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[mvjmcs](#)

vcov	<i>Variance-covariance matrix of the estimated parameters for joint models</i>
------	--

Description

Extract variance-covariance matrix for joint models.

Usage

```
## S3 method for class 'jmcs'
vcov(object, ...)
```

Arguments

`object` an object inheriting from class `jmcs`.

`...` further arguments passed to or from other methods.

Value

a matrix of variance covariance of all parameter estimates.

Author(s)

Shanpeng Li <lishanpeng0913@ucla.edu>

See Also

[jmcs](#)

ydata	<i>Simulated longitudinal data</i>
-------	------------------------------------

Description

The ydata data frame has 3067 rows and 6 columns.

Usage

```
data(ydata)
```

Format

This data frame contains the following columns:

ID patient identifier.

response response variable.

time visit time.

x1 treatment indicator. 0 denotes the placebo group and 1 the treatment group.

gender gender indicator.

race race indicator.

Index

* datasets

cdata, [5](#)

mvdata, [13](#)

mvydata, [17](#)

ydata, [32](#)

anova, [3](#)

AUCjmcs, [4](#)

cdata, [5](#)

fitted, [6](#)

fitted.jmcs, [10](#)

fixef, [7](#), [10](#)

jmcs, [3](#), [5](#), [8](#), [13](#), [19–23](#), [26–29](#), [31](#)

MAEQjmcs, [12](#)

mvdata, [13](#)

mvjmcs, [14](#), [20](#), [22](#), [26](#), [31](#)

mvydata, [17](#)

PEjmcs, [17](#)

plot.jmcs, [10](#), [19](#)

print, [20](#)

print.survfitjmcs, [21](#)

print.survfitmvjmcs, [21](#)

ranef, [10](#), [22](#)

residuals, [23](#)

residuals.jmcs, [10](#)

simmvJMdata, [24](#)

summary, [25](#)

summary.AUCjmcs, [26](#)

summary.MAEQjmcs, [27](#)

summary.PEjmcs, [27](#)

survfitjmcs, [5](#), [10](#), [13](#), [19](#), [21](#), [26–28](#), [28](#)

survfitmvjmcs, [22](#), [30](#)

vcov, [31](#)

vcov.jmcs, [10](#)

ydata, [32](#)