

Package ‘INLAjoint’

November 10, 2025

Type Package

Title Multivariate Joint Modeling for Longitudinal and Time-to-Event Outcomes with 'INLA'

Version 25.11.10

Description Estimation of joint models for multivariate longitudinal markers (with various distributions available) and survival outcomes (possibly accounting for competing risks) with Integrated Nested Laplace Approximations (INLA). The flexible and user friendly function `joint()` facilitates the use of the fast and reliable inference technique implemented in the 'INLA' package for joint modeling. More details are given in the help page of the `joint()` function (accessible via `?joint` in the R console) and the vignette associated to the `joint()` function (accessible via `vignette(`INLAjoint`)` in the R console).

License GPL-3

URL <https://github.com/DenisRustand/INLAjoint>

BugReports <https://github.com/DenisRustand/INLAjoint/issues>

Encoding UTF-8

LazyData true

Depends R (>= 3.6), utils, stats

Imports numDeriv, lme4, ggplot2, methods, Matrix, nlme, data.table

Additional_repositories <https://inla.r-inla-download.org/R/testing>

RoxygenNote 7.3.3

Suggests INLA, knitr, rmarkdown, splines, graphics, grDevices, R.rsp, survival, JMBayes2, rstanarm, frailtypack, smcure, fmesher

BuildVignettes true

VignetteBuilder R.rsp

Maintainer Denis Rustand <INLAjoint@gmail.com>

NeedsCompilation no

Author Denis Rustand [cre, aut] (ORCID:
<<https://orcid.org/0000-0001-9708-5220>>),
Elias Teixeira Krainski [aut] (ORCID:

<<https://orcid.org/0000-0002-7063-2615>>),
 Haavard Rue [aut] (ORCID: <<https://orcid.org/0000-0002-0222-1881>>),
 Janet van Niekerk [aut] (ORCID:
 <<https://orcid.org/0000-0002-4334-2057>>)

Repository CRAN

Date/Publication 2025-11-10 22:10:02 UTC

Contents

coef.INLAjoint	3
family.INLAjoint	3
fitted.INLAjoint	4
fixef.INLAjoint	4
formula.INLAjoint	5
INLAjoint	5
INLAjoint.ginv	6
INLAjoint.object	6
INLAjoint.rw	9
INLAjoint.rw2	9
INLAjoint.scopy.define	10
joint	10
joint.rerun	15
joint.run	16
logLik.INLAjoint	17
LongMS	17
Longsim	18
nobs.INLAjoint	18
plot.INLAjoint	19
predict.INLAjoint	19
print.plot.INLAjoint	22
ranef.INLAjoint	23
setup_FE_model	23
setup_RE_model	24
setup_S_model	24
setup_Y_model	25
SurvMS	26
Survsim	27

Index	28
--------------	-----------

coef.INLAjoint	<i>Extracts model coefficients from a given model fitted with INLAjoint</i>
----------------	---

Description

This function extracts model coefficients from INLAjoint objects.

Usage

```
## S3 method for class 'INLAjoint'  
coef(object, ...)
```

Arguments

object	an object that contains a model fitted with INLAjoint.
...	Extra arguments.

family.INLAjoint	<i>Extracts family from a given model fitted with INLAjoint</i>
------------------	---

Description

This function extracts family from INLAjoint objects.

Usage

```
## S3 method for class 'INLAjoint'  
family(object, ...)
```

Arguments

object	an object that contains a model fitted with INLAjoint.
...	Extra arguments.

fitted.INLAjoint	<i>Extracts fitted values from a given model fitted with INLAjoint</i>
------------------	--

Description

This function extracts fitted values from INLAjoint objects. Values are associated to a name to keep track of the outcome related to each fitted value.

Usage

```
## S3 method for class 'INLAjoint'  
fitted(object, ...)
```

Arguments

object	an object that contains a model fitted with INLAjoint.
...	Extra arguments.

fixef.INLAjoint	<i>Extracts fixed effects values from a given model fitted with INLAjoint</i>
-----------------	---

Description

This function extracts fixed effects values from INLAjoint objects.

Usage

```
## S3 method for class 'INLAjoint'  
fixef(object, ...)
```

Arguments

object	an object that contains a model fitted with INLAjoint.
...	Extra arguments.

formula.INLAjoint	<i>Extracts formula from a given model fitted with INLAjoint</i>
-------------------	--

Description

This function extracts formula from INLAjoint objects.

Usage

```
## S3 method for class 'INLAjoint'
formula(x, ...)
```

Arguments

x	an object that contains a model fitted with INLAjoint.
...	Extra arguments.

INLAjoint	<i>INLAjoint</i>
-----------	------------------

Description

INLAjoint is a package that fits joint models for multivariate longitudinal markers (with various distributions available) and survival outcomes (possibly accounting for competing risks) with Integrated Nested Laplace Approximations (INLA). The flexible and user friendly function joint() facilitates the use of the fast and reliable inference technique implemented in INLA package for joint modeling. More details are given in the help page of the joint function (accessible via ?joint in the R console), the vignette associated to the joint() function (accessible via vignette("INLAjoint") in the R console).

Contact: <INLAjoint@gmail.com>

Usage

```
INLAjoint()
```

INLAjoint.ginv	<i>Setup ginv</i>
----------------	-------------------

Description

Setup random walk input:

Usage

```
INLAjoint.ginv(x, tol = sqrt(.Machine$double.eps), rankdef = NULL)
```

Arguments

x	input
tol	tolerance
rankdef	rank output:

Value

ginv

INLAjoint.object	<i>Fitted joint object</i>
------------------	----------------------------

Description

An object of class INLAjoint returned by the joint function that fits a joint model to multivariate longitudinal and time-to-event data. The following functions can apply to objects of this class: plot, print, summary and priors.used.

Usage

```
INLAjoint.object
```

Format

An object of class NULL of length 0.

Value

A list with the following components:

`names.fixed` a vector with the name of the fixed effects of the model. The corresponding submodel is indicated by the suffix including a letter and a number ("L" for longitudinal and "S" for survival).

`summary.fixed` summary statistics for the fixed effects of the model. The summary statistics sorted by longitudinal and survival components are available by applying the summary function to the INLAjoint object.

`summary.fixed` marginals for the fixed effects of the model.

`mlik` log marginal-likelihood.

`cpo` Conditional Predictive Ordinate.

`gcpo` Group-Conditional Predictive Ordinate.

`po` Predictive ordinate.

`waic` Widely applicable Bayesian information criterion

`model.random` a vector with the name of the random parameters of the model, possibly including the following components:

`RW1 model` and `RW2 model` Random walk of order 1 or 2 corresponding to Bayesian smoothing splines for the baseline hazard risk

`IID model` Univariate random effect.

`IIDKD model` Multivariate random effects.

`Copy` association parameter.

`summary.random` summary statistics for the random parameters of the model.

`marginals.random` marginals for the random parameters of the model.

`size.random` size of the random parameters of the model.

`summary.linear.predictor` summary statistics of the linear predictors.

`marginals.linear.predictor` marginals for the linear predictors.

`summary.fitted.values` summary statistics of the fitted values.

`marginals.fitted.values` marginals for the fitted values.

`size.linear.predictor` size of the linear predictors.

`summary.hyperpar` summary statistics for the hyperparameters of the model. The summary statistics sorted by longitudinal and survival components are available by applying the summary function to the INLAjoint object. Particularly, this is the raw output of INLA and therefore the precision of the residual errors and baseline hazard functions hyperparameters are provided. Similarly, the Cholesky matrix is given for the random-effects. The summary function can easily return either variance and covariance or standard deviations and correlations for all these hyperparameters.

`marginals.hyperpar` marginals for the hyperparameters of the model.

`internal.summary.hyperpar` summary of the internal hyperparameters, this is similar to the summary of the hyperparameters but here they are provided as used for the computations (logarithm scale for residual error and baseline risk hyperparameters).

`internal.marginals.hyperpar` marginals for the internal hyperparameters of the model.
`misc` miscellaneous (as provided in the INLA output).
`dic` Deviance Information Criterion.
`mode` .
`joint.hyper` .
`nhyper` .
`version` Version of INLA.
`cpu.used` Computation time of INLA.
`all.hyper` .
`.args` .
`call` INLA call.
`selection` information about parameters for sampling with `inla.rj` marginal.
`cureVar` informations about cure fraction submodel for mixture cure survival models.
`variant` information about variant for Weibull baseline hazards.
`SurvInfo` some information about survival submodels (names of event indicator and event time variables as well as baseline hazard).
`famLongi` list of distributions for the longitudinal markers.
`corLong` boolean indicating if random effects are correlated accross markers.
`control.link` informations about link function (1=default).
`longOutcome` name of longitudinal outcomes.
`survOutcome` name of survival outcomes.
`assoc` vector with names of all association parameters (longi-surv).
`id` name of the id variable.
`timeVar` name of time variable.
`range` information about range of X-axis values for non-linear associations.
`REstruc` names of the grouped random effects for the longitudinal markers.
`mat_k` contains the list of random effects covariance matrices when they are fixed as they are not part of the estimated parameters (used for displaying them in summary).
`fixRE` list of the size of number of groups of random effects, each element is a boolean indicating if the random effects of the group is fixed (TRUE) or estimated (FALSE).
`lonFacChar` list of factors and character covariates included in the longitudinal submodels to keep track of modalities (used internally when doing predictions to reconstruct categorical covariates).
`survFacChar` same as `lonFacChar` but for survival submodels.
`corRE` list indicating if groups of random effects are correlated within longitudinal submodels.
`basRisk` list of the baseline risk used for each survival submodel.
`priors_used` informations about priors used in the model, internally used to display priors in plots (with argument `priors=TRUE` in the call of the plot function). Note that priors can also be displayed with the function `priors.used()` applied to an INLAjoint object.
`dataLong` name of the longitudinal dataset.
`dataSurv` name of the survival dataset.

See Also

[joint.](#)

INLAjoint.rw	<i>Setup rw</i>
--------------	-----------------

Description

Setup random walk input:

Usage

```
INLAjoint.rw(n, order = 1L, sparse = TRUE, scale.model = FALSE, cyclic = FALSE)
```

Arguments

n	number of knots
order	order 1
sparse	boolean, sparsity
scale.model	boolean, scale
cyclic	boolean, cyclic output:

Value

random walk 1

INLAjoint.rw2	<i>Setup rw2</i>
---------------	------------------

Description

Setup random walk of order 2 input:

Usage

```
INLAjoint.rw2(n, order = 2L, ...)
```

Arguments

n	number of knots
order	order 2
...	additional arguments passed to INLAjoint.rw output:

Value

random walk 2

```
INLAjoint.scopy.define
```

Setup scopy

Description

Setup weights for non-linear effects input:

Usage

```
INLAjoint.scopy.define(n = 5L)
```

Arguments

n number of knots output:

Value

Matrix with weights for scopy

```
joint
```

Fit a multivariate joint model for longitudinal and/or survival data

Description

This function fits a multivariate joint model for longitudinal and/or survival data. The longitudinal outcomes are modeled with mixed effects models and can handle various distributions. The survival outcomes (i.e., terminal event with possibly competing risks) are modeled with Cox proportional hazards regression models. Various association structures can be specified between the longitudinal and survival outcomes. The inference is based on Integrated Nested Laplace Approximations (Rue et al., 2009).

Usage

```
joint(
  formSurv = NULL,
  formLong = NULL,
  dataSurv = NULL,
  dataLong = NULL,
  id = NULL,
  timeVar = NULL,
  family = "gaussian",
  link = "default",
  basRisk = "rw1",
  NbasRisk = 15,
```

```

cutpoints = NULL,
assoc = NULL,
assocSurv = NULL,
corLong = FALSE,
corRE = TRUE,
dataOnly = FALSE,
longOnly = FALSE,
silentMode = FALSE,
reorder = TRUE,
run = TRUE,
control = list()
)

```

Arguments

formSurv	the formula for the time-to-event outcome, with the response given as an <code>inla.surv()</code> object. Keep it as <code>NULL</code> if no survival part is needed and give a list of formulas for competing risks.
formLong	the formula for the longitudinal outcome, structured as with the <code>lme4</code> package (Mächler et al., 2015) for linear mixed-effects models (i.e., including random effects within parenthesis). Keep it as <code>NULL</code> if no longitudinal part is needed and give a list of formulas for multivariate longitudinal.
dataSurv	the dataset for the survival part. Keep it as <code>NULL</code> if no survival part is needed or if the survival data is in the longitudinal dataset (it will extract the last line for each individual as the survival dataset).
dataLong	the dataset for the longitudinal part. Keep it as <code>NULL</code> if no longitudinal part is needed. For multivariate longitudinal models, either give one dataset with all outcomes and covariates or a list of datasets for each longitudinal marker.
id	the name of the variable to identify individuals or grouped repeated measurements. Keep is as <code>NULL</code> if no longitudinal part is needed.
timeVar	a character string (or a vector) giving the name of the time-varying variable(s). Functions of time can be included in formulas, they first need to be set up as a univariate function with name <code>fX</code> , where <code>X</code> is a number between 1 and 20. Then the function can be used directly in the formula (see example below).
family	a character string (or a vector) giving the name of families for the longitudinal outcomes. The list of the available families is given by <code>names(inla.models())\$likelihood</code> .
link	a character string (or a vector) giving the link function associated to the families for the longitudinal outcomes. The links available for a family is given in the associated doc: <code>inla.doc("familyName")</code> . The link should be a vector of the same size as the family parameter and should be set to "default" for default (i.e., identity for gaussian, log for poisson, logit for binomial,...).
basRisk	the baseline risk of event (should be a vector in case of competing risks). It can be defined as parametric with either "exponentialsurv" for exponential baseline or "weibullsurv" for Weibull baseline (note that there are two formulations of the Weibull distribution, see <code>'inla.doc("weibull")'</code> for more details, default is variant = 0). Alternatively, there are two options to avoid parametric assumptions on

	the shape of the baseline risk: "rw1" for random walks of order one prior that corresponds to a smooth spline function based on first order differences. The second option "rw2" assigns a random walk order two prior that corresponds to a smooth spline function based on second order differences. This second option provides a smoother spline compared to order one since the smoothing is then done on the second order.
NbasRisk	the number of intervals for the baseline risk function, only one value should be provided and the same number of intervals is used for each risk submodel in case of competing risks.
cutpoints	a vector with baseline hazard cutpoints if not using equidistant (if not NULL, this replaces the NbasRisk parameter).
assoc	a character string that specifies the association between the longitudinal and survival components. The available options are "CV" for sharing the current value of the linear predictor, "CS" for the current slope, "CV_CS" for the current value and the current slope, "SRE" for shared random effects (i.e., sharing the individual deviation from the mean at time t as defined by the random effects), "SRE_ind" for shared random effect independent (each random effect's individual deviation is associated to an association parameter in the survival sub-model) and "" (empty string) for no association. When there are either multiple longitudinal markers or multiple competing events, this should be a vector. In case of both multiple markers and events, it should be a list with one element per longitudinal marker and each element is a vector of association for each competing event. Keep it as NULL to have no association between longitudinal and survival components or if there is no survival component.
assocSurv	a boolean that indicates if a frailty term (i.e., random effect) from a survival model should be shared into another survival model. The order is important, the first model in the list of survival formulas ('formSurv') should include a random effect and it can be shared in the next formulas. Multiple survival models with random effects can be accommodated and a random effect can be shared in multiple survival models, following the same structure as 'assoc' (i.e., vector of booleans if one random effect is shared in multiple survival and list of vectors if multiple survival models with random effects share their random effects in multiple survival models).
corLong	a boolean that only applies when multiple longitudinal markers are fitted: should the random effects accross markers be correlated (TRUE) or independent (FALSE)? Default is FALSE.
corRE	list of the size of number of groups of random effects (i.e., equal to 1 if there is only one longitudinal marker or if corLong is TRUE and equal to the number of markers otherwise), each element is a boolean indicating if the random effects of the group must be correlated or independent (i.e., diagonal variance-covariance). Default is FALSE.
dataOnly	a boolean to only prepare the data with the correct format without running the model.
longOnly	a boolean to only prepare the data for the longitudinal part of a longitudinal-survival joint model with the correct format without running the model.
silentMode	a boolean that will stop printing messages during computations if turned to TRUE.

reorder	a boolean used to prevent reordering of the data according to id.
run	a boolean used to setup a model without running INLA (allows to make modifications prior to run).
control	<p>a list of control values that can be set with <code>control=list()</code>, with components:</p> <p><code>priorFixed</code> list with mean and standard deviations for the Gaussian prior distribution for the fixed effects. Default is <code>list(mean=0, prec=0.01, mean.intercept=0, prec.intercept=0.01)</code>, where <code>mean</code> and <code>prec</code> are the mean and precision (i.e., inverse of the variance) of the fixed effects, respectively and <code>mean.intercept</code> and <code>prec.intercept</code> are the corresponding parameters for the fixed intercept.</p> <p><code>priorAssoc</code> list with mean and standard deviations for the Gaussian prior distribution for the association parameters (does not apply to "SRE_ind" association and shared random effect from survival models (frailty), see next item for those two). Default is <code>list(mean=0, prec=0.01)</code></p> <p><code>priorSRE_ind</code> list with mean and standard deviations for the Gaussian prior distribution for the association of independent random effects ("SRE_ind" and survival frailty random effects shared). Default is <code>list(mean=0, prec=1)</code></p> <p><code>priorRandom</code> list with prior distribution for the multivariate random effects (Inverse Wishart). Default is <code>list(r=10, R=1)</code>. The default behavior automatically adapts the <code>r</code> to be equal to <code>order + 1</code> if not manually specified, making the prior scaled to the dimension of the random effects., see "inla.doc("iidkd")" for more details.</p> <p><code>priorFrailty</code> list with prior distribution for the frailty iid random effects (Inverse Gamma). Default is <code>list(alpha=4, beta=1)</code>, see "inla.doc("iid123")" for more details.</p> <p><code>priorRW</code> Vector for the Penalised Complexity prior of the log precision of random walk baseline hazards. Default is a vector <code>c(0.5, 0.01)</code>. See "inla.doc("RW1")" and "inla.doc("RW2")" for more details.</p> <p><code>initVC</code> list of the size of number of groups of random effects giving initial values for variance-covariance of random effects, first values are variance and then covariances (as displayed in summary). All the elements of the covariance matrix must be fixed but in case of multiple groups of random effects, it is possible to fix initial values for only some groups, then elements in the list that are not initialized must be an empty string.</p> <p><code>initSD</code> same as <code>initVC</code> but to fix standard deviations and correlations instead (only one of these two arguments can be used).</p> <p><code>strata</code> list of the same size as the number of survival submodels, giving the name of covariates for stratified proportional hazards model (default is <code>NULL</code>).</p> <p><code>Kinship</code> matrix for kinship frailty structure in survival models, this must be the inverse of <code>2*kinship</code> and only works when a frailty term is included in the model (default is <code>NULL</code>).</p> <p><code>fixRE</code> list of the size of number of groups of random effects, each element is a boolean indicating if the random effects of the group must be fixed or estimated.</p> <p><code>assocInit</code> Initial value for all the association parameters (default is 0.1).</p>

`int.strategy` a character string giving the strategy for the numerical integration over the hyperparameters used to approximate the marginal posterior distributions of the latent field. Available options are "auto" (default), "ccd", "grid" or "eb" (empirical Bayes). The default strategy uses "grid" for $\text{dim} \leq 2$ and "ccd" otherwise. The empirical Bayes uses only the mode of the approximations for the integration, which speed up and simplifies computations. It can be pictured as a tradeoff between Bayesian and frequentist estimation strategies while the default full Bayesian accounts for uncertainty by using the mode and the curvature at the mode. See `?control.inla` for additional details.

`Ntrials` Number of trials for binomial and Betabinomial distributions, default is NULL.

`cpo` TRUE/FALSE: Default is FALSE, set to TRUE to compute the Conditional Predictive Ordinate.

`cfg` TRUE/FALSE: Default is FALSE, set to TRUE to be able to sample from the posterior distribution.

`safemode` TRUE/FALSE: use the INLA safe mode (automatically reruns in case of negative eigenvalue(s) in the Hessian, reruns with adjusted starting values in case of crash). Default is TRUE (activated). The message "****inla.core.safe****" appears when the safe mode is running, it improves the inference of the hyperparameters and can be ignored. To remove this safe mode, switch the boolean to FALSE (it can save some computation time but may return slightly less precise estimates for some hyperparameters).

`n_NL` Number of knots for non-linear associations (random walk of order 2).

`NLpriorAssoc` Priors for non-linear effects, it is a list of 3 elements "mean", "slope" and "spline", each element including 3 options: "mean", "prec" and "initial" for the mean, precision of the prior and initial values.

`rerun` TRUE/FALSE: the model reruns to improve numerical stability (default is FALSE).

`tolerance` accuracy in the inner optimization (default is 0.005).

`h` step-size for the hyperparameters (default is 0.005).

`verbose` TRUE/FALSE: prints details of the INLA algorithm. Default is FALSE.

`keep` TRUE/FALSE: keep internal files. Default is FALSE. (expert option)

`lightmode` Controls the amount of memory in the saved object. Default is 0 (save all), when set to 1 a lighter version is saved but keeps required information for `plot()` and `predict()` while when set to 2 the light version also removes `.args` and `misc` (for an even lighter object).

Value

An object of class `INLAjoint`. See [INLAjoint.object](#) for details.

References

Rustand, D., van Niekerk, J., Teixeira Krainski, E., Rue, H. and Proust-Lima, C. (2023). Fast and flexible inference for joint models of multivariate longitudinal and survival data using integrated

nested Laplace approximations. *Biostatistics*, 2023, kxad019. <https://doi.org/10.1093/biostatistics/kxad019>
<https://arxiv.org/abs/2203.06256>

Rustand, D., van Niekerk, J., Rue, H., Tournigand, C., Rondeau, V. and Briollais, L. (2023). Bayesian Estimation of Two-Part Joint Models for a Longitudinal Semicontinuous Biomarker and a Terminal Event with R-INLA: Interests for Cancer Clinical Trial Evaluation. *Biometrical Journal*, 65, 2100322. <https://doi.org/10.1002/bimj.202100322> <https://arxiv.org/abs/2010.13704>

Rue, H., Martino, S. and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71: 319-392. <https://doi.org/10.1111/j.1467-9868.2008.00700.x>

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1), 1–48. <https://doi.org/10.18637/jss.v067.i01>

Contact: <INLAjoint@gmail.com>

Examples

```
# joint model with 3 longitudinal / 3 competing risks of event
data(Longsim)
data(Survsim)
f1 <- function(x) x^2 # quadratic function of time for first marker
Nsplines <- splines::ns(Longsim$time, knots=2) # 2 ns splines for second marker
f2 <- function(x) predict(Nsplines, x)[,1]
f3 <- function(x) predict(Nsplines, x)[,2]

if(requireNamespace("INLA")){
  JMINLA <- joint(
    formLong = list(Y1 ~ time + f1(time) + ctsX + binX + (1 + time + f1(time) | Id),
                    Y2 ~ time + f2(time) + f3(time) + ctsX + binX + (1 | Id),
                    Y3 ~ time + ctsX + binX + (1 | Id)),
    formSurv = list(INLA::inla.surv(deathTimes, Event1) ~ binX + ctsX,
                    INLA::inla.surv(deathTimes, Event2) ~ binX,
                    INLA::inla.surv(deathTimes, Event3) ~ ctsX),
    dataLong = Longsim, dataSurv=Survsim, id = "Id", timeVar = "time", corLong=TRUE,
    family = c("gaussian", "poisson", "binomial"), basRisk = c("rw1", "rw1", "rw1"),
    assoc = list(c("CV", "CS", ""), c("CV", "", "SRE"), c("", "CV", "")),
    control=list(int.strategy="eb"))

  summary(JMINLA)
  # 'sdcor' to switch from variance-covariance to standard
  # deviation-correlation and 'hazr' to switch parameters
  # in survival submodels from mean to hazard ratios (exp(mean)).
  summary(JMINLA, sdcor=TRUE, hazr=TRUE)
}
```

Description

Reruns `inla()` for an object of class `INLAjoint` returned by the `joint` function. The rerun starts with posterior values from previous run and can sometimes improve the model fit (for very complex models or unstable parameter estimates due to low information in the data)

Usage

```
joint.rerun(model, ...)
```

Arguments

<code>model</code>	an object containing a model fitted with the <code>joint()</code> function.
<code>...</code>	Extra arguments.

Value

An object of class `INLAjoint` containing a model fitted with the `joint()` function.

See Also

[joint.](#)

`joint.run`

Run a model fitted with `INLAjoint`

Description

Runs `inla()` for an object of class `INLAjoint` returned by the `joint` function with argument `'run'` set to `FALSE`. The rerun starts with posterior values from previous run and can sometimes improve the model fit (for very complex models or unstable parameter estimates due to low information in the data)

Usage

```
joint.run(model, silentMode = FALSE, class = "INLAjoint", ...)
```

Arguments

<code>model</code>	an object containing a model fitted with the <code>joint()</code> function.
<code>silentMode</code>	boolean to display messages about the fit procedure. Default is <code>FALSE</code> .
<code>class</code>	defines the class of the object created. Default is <code>"INLAjoint"</code> but can be switched to <code>"inla"</code> .
<code>...</code>	Extra arguments.

Value

An object of class `INLAjoint` containing a model fitted with the `joint()` function.

See Also[joint.](#)

logLik.INLAjoint	<i>Extracts log-likelihood value from a given model fitted with INLAjoint</i>
------------------	---

Description

This function extracts log-likelihood value from INLAjoint objects.

Usage

```
## S3 method for class 'INLAjoint'
logLik(object, ...)
```

Arguments

object	an object that contains a model fitted with INLAjoint.
...	Extra arguments.

LongMS	<i>Simulated univariate longitudinal dataset</i>
--------	--

Description

A simulated dataset used to illustrate joint models with univariate longitudinal and multi-state survival component (see joint longitudinal and multi-state model in INLAjoint vignette). The corresponding survival dataset is named 'SurvMS'.

Usage

```
LongMS
```

Format

```
## 'LongMS' A data frame with 96 rows and 7 columns:
```

id Individual id

time Visit times

X Continuous covariate

y Gaussian longitudinal outcome ...

Longsim	<i>Simulated multivariate longitudinal dataset</i>
---------	--

Description

A simulated dataset used to illustrate some models (e.g., example of joint() function in the documentation). The corresponding survival dataset is named 'Survsim'.

Usage

```
Longsim
```

Format

```
## 'Longsim' A data frame with 96 rows and 7 columns:
```

Id Individual id

time Visit times

ctsX Continuous covariate

binX Binary covariate

Y1, Y2, Y3 Gaussian Longitudinal outcomes ...

nobs.INLAjoint	<i>Extracts number of observations of each composant from a given model fitted with INLAjoint</i>
----------------	---

Description

This function extracts number of observations of each composant from INLAjoint objects.

Usage

```
## S3 method for class 'INLAjoint'
nobs(object, ...)
```

Arguments

object	an object that contains a model fitted with INLAjoint.
...	Extra arguments.

plot.INLAjoint	<i>Plot the output from a multivariate joint model for longitudinal and/or survival data</i>
----------------	--

Description

This function provide plots for the output of a multivariate joint model for longitudinal and/or survival data. The output can be stored into an object and manipulated as a list of ggplot outputs, described bellow.

Usage

```
## S3 method for class 'INLAjoint'
plot(x, ...)
```

Arguments

x	an object with the output of the the joint function
...	Extra arguments including: sdcor: logical indicating if the random effects correlation are to be shown. If FALSE the covariance is shown. priors: logical indicating if the priors are added to the posterior marginals plots.

Value

return a named list of ggplot objects containing:

Outcomes as a list of length equal the number of longitudinal outcomes plus the number of survival outcomes, each one including the plot for the posterior marginals of the associated fixed effects and residual or baseline variance (or standard error). Each element contains the plot for the posterior marginal.

Covariances the plots for the posterior marginal distribution of the covariance parameters.

Associations the plots for the posterior marginal distribution of the association parameters.

Random The plot for the fitted baseline risk functions as shown as the posterior mean and credible interval.

predict.INLAjoint	<i>Computes predictions for a given model fitted with INLAjoint</i>
-------------------	---

Description

This function allows to compute predictions for a given model fitted with INLAjoint, the default behavior (without arguments) returns fitted values for each component of the model. It is also possible to supply a dataset for which predictions are required, this dataset must have the same structure as the dataset used for the model fitting (i.e., same columns). The default returned predictions corresponds to the linear predictors for each outcomes.

Usage

```
## S3 method for class 'INLAjoint'
predict(
  object,
  newData = NULL,
  newDataSurv = NULL,
  timePoints = NULL,
  NtimePoints = 50,
  NsampleHY = 20,
  NsampleFE = 20,
  NsampleRE = 50,
  id = NULL,
  Csurv = NULL,
  startTime = NULL,
  horizon = NULL,
  baselineHaz = "interpolation",
  return.samples = FALSE,
  FEonly = FALSE,
  survival = FALSE,
  CIF = FALSE,
  inv.link = FALSE,
  NidLoop = "auto",
  resErrLong = FALSE,
  set.samples = NULL,
  silentMode = FALSE,
  ...
)
```

Arguments

object	an object that contains a model fitted with INLAjoint.
newData	a dataset with the same columns as those used to fit the model. When using a longitudinal marker to predict longitudinal and subsequent survival outcomes, only the longitudinal information (i.e., structure of the longitudinal data) is required. It is also possible to predict the average trajectories conditional on covariates by setting the value of the longitudinal outcomes included in the model to NA.
newDataSurv	a dataset for survival information (only useful when both longitudinal and survival data are provided for the predictions, otherwise using the argument newData is working too).
timePoints	a vector of the time points at which predictions are computed (for both longitudinal and survival outcomes), this also control the precision of the integration for time-dependent shared terms and the computation of cumulative risks (e.g., for survival or CIF curves), thus many time points will increase the accuracy of predictions. Default is NULL as these time points are automatically computed when not defined manually.
NtimePoints	number of time points at which the predictions are computed (for both longitudinal and survival outcomes), these time points are equidistant between time

	0 and horizon time. This also control the precision of the integration for time-dependent shared terms and the computation of cumulative risks (e.g., for survival or CIF curves), thus many time points will increase the accuracy of predictions.
NsampleHY	number of samples for hyperparameters used to assess uncertainty when computing predictions. Default is 20.
NsampleFE	number of samples of fixed effects for each hyperparameters samples used to assess uncertainty when computing predictions. Default is 30 (i.e., 30 x NsampleHY).
NsampleRE	number of random effects realizations for each sample specified in 'NsampleHY' and 'NsampleFE'. Default is 50 (i.e., 50 x NsampleFE x NsampleHY, resulting in 20000 random effects samples per new individual with default values). These random effects realizations are conditional on observed longitudinal outcomes values provided in 'newData' and survival time provided in 'newDataSurv' when a survival model is included. If 'newDataSurv' is NULL, they are conditional on survival up to latest longitudinal recorded measurement. When outcomes are set to NA, the realizations are sampled from the marginal distribution of random effects.
id	name of the individual id variable, default is NULL as it is automatically grabbed from the fitted model but when fitting simple survival models, providing id when fitting the model is not mandatory and thus this can be useful (an explicit message is printed in this specific case).
Csurv	conditional survival, gives the starting value of the at-risk period (i.e., starting value at which risk predictions for survival models are computed). Default is the last longitudinal observation time provided in 'newData' but this is replaced by the value of 'Csurv' when provided.
startTime	define a starting time for predictions.
horizon	horizon of the prediction.
baselineHaz	method used to evaluate the baseline hazard value, default is 'interpolation' which is currently recommended. Experimental alternatives are being developed, including 'splines' for an interpolation with splines but has not been properly validated with simulations yet.
return.samples	boolean, when set to TRUE the samples are returned instead of summary statistics over the samples. Default is FALSE.
FOnly	boolean, when set to TRUE, only fixed effects are involved for predictions computations.
survival	boolean, when set to TRUE the summary statistics over survival functions are computed in addition to the summary statistics over the risk functions.
CIF	boolean, when set to TRUE the summary statistics over cumulative incidence functions are computed in addition to the summary statistics over the risk functions. Only applies to competing risks.
inv.link	boolean, when set to TRUE the summary statistics are computed over the predictions of longitudinal components after applying the inverse link function for each samples in addition to the summary statistics over the linear predictors.

NidLoop	Gives the number of individuals for which we compute predictions at once. For large number of individuals, this will loop over groups of 'NidLoop' individuals and could make predictions computations faster.
resErrLong	boolean, when set to TRUE the residual error for Gaussian or lognormal longitudinal outcomes is added to the uncertainty of predictions (default is FALSE which predicts the true underlying value of the longitudinal marker, i.e., error-free).
set.samples	replace random effects with pre-sampled values.
silentMode	a boolean that will stop printing messages during computations if turned to TRUE.
...	Extra arguments.

print.plot.INLAjoint	<i>Prints plot the output from a multivariate joint model for longitudinal and/or survival data</i>
----------------------	---

Description

The plots of a model are grouped by categories, first are the fixed effects and residual error of longitudinal and survival submodels, referred to as 'Outcomes' or 'O'. Then the variance-covariance of random effects (or standard deviations and correlations when the argument 'sdcor' is set to TRUE in the call of the plot function), referred to as 'Covariances' or 'C'. Association parameters referred to as 'Associations' or 'A' for linear associations and 'NL_Associations' or 'N' for non-linear associations. Baseline hazard curves referred to as 'Baseline' or 'B' and baseline hazard related parameters referred to as 'ParamBaseline' or 'P'. It is possible to select specific plots to print by specifying the names or corresponding letters in the argument 'which'.

Usage

```
## S3 method for class 'plot.INLAjoint'
print(
  x,
  which = c("all", "Outcomes", "Covariances", "Associations", "Baseline"),
  ...
)
```

Arguments

x	an object with the output of the the joint function
which	name of required plots. Default is "all". It can be a character or named list. If it is a list, each element can be either character or numeric to select from the elements to be visualized. Ex.: which = list(Outcomes = "L1") and list(Outcomes = 1) will produce the same output.
...	Extra arguments.

ranef.INLAjoint	<i>Extracts random effects values from a given model fitted with INLA-joint</i>
-----------------	---

Description

This function extracts random effects values from INLAjoint objects.

Usage

```
## S3 method for class 'INLAjoint'
ranef(object, ...)
```

Arguments

object	an object that contains a model fitted with INLAjoint.
...	Extra arguments.

setup_FE_model	<i>Setup fixed effects part for longitudinal marker k</i>
----------------	---

Description

Setup fixed effects part for longitudinal marker k (internal function) input:

Usage

```
setup_FE_model(formula, dataset, timeVar, k, dataOnly)
```

Arguments

formula	with lme4 format (fixed effects and random effects in the same object)
dataset	that contains the outcome
timeVar	name of time variable
k	identifies the longitudinal marker among 1:K markers
dataOnly	boolean for internal use, indicates if only preparing data (i.e., not fitting the model) output:

Value

colnames(FE) names of the fixed effects (interactions are separated by ".X." instead of ":" to facilitate their manipulation)
 FE values of the fixed effects

setup_RE_model	<i>Setup random effects part for longitudinal marker k</i>
----------------	--

Description

Setup random effects part for longitudinal marker k (internal function) input:

Usage

```
setup_RE_model(formula, dataset, k)
```

Arguments

- formula with lme4 format (fixed effects and random effects in the same object)
- dataset that contains the outcome
- k identifies the longitudinal marker among 1:K markers output:

Value

- colnames(RE_mat) names of the random effects
- RE_mat values of the random effects

setup_S_model	<i>Setup survival part for outcome m</i>
---------------	--

Description

Setup survival part for outcome m (internal function) input:

Usage

```
setup_S_model(  
  formula,  
  formLong,  
  dataSurv,  
  LSurvdat,  
  timeVar,  
  assoc,  
  id,  
  m,  
  K,  
  M,  
  NFT,  
  corLong,  
  dataOnly,
```



```
SurvInfo,  
strata  
)
```

Arguments

formula	with inla.surv() object as an outcome
formLong	formula from the longitudinal part, if any
dataSurv	dataset(s) for the survival part
LSurvdat	dataset for the longitudinal part converted to survival format (internal, used to get covariates if missing in the survival dataset when sharing linear predictors including covariates from longitudinal into survival)
timeVar	names of the variables that are time-dependent (only linear for now)
assoc	association parameters between K longitudinal outcomes and M survival outcomes (list of K vectors of size M)
id	name of the variable that gives the individual id
m	identifies the outcome among 1:M time-to-event outcomes
K	number of longitudinal outcomes
M	number of survival outcomes
NFT	maximum number of functions of time (fixed value)
corLong	boolean that indicates if random effects across longitudinal markers are correlated, when multiple longitudinal markers are included in the model
dataOnly	boolean for internal use, indicates if only preparing data (i.e., not fitting the model)
SurvInfo	information about survival submodels for internal use
strata	covariate for stratified proportional hazards model output:

Value

YS_data includes the values of the survival outcome and covariates associated to this survival part, with the association parameters but the provided id are temporary and they will be updated after the cox expansion to make them unique and allow for time dependency

YSformF formula for this survival outcome (not including association parameters)

setup_Y_model	<i>Setup outcome for longitudinal marker</i>
---------------	--

Description

Setup outcome for longitudinal marker (internal function) input:

Usage

```
setup_Y_model(formula, dataset, family, k)
```

Arguments

<code>formula</code>	with lme4 format (fixed effects and random effects in the same object)
<code>dataset</code>	that contains the outcome
<code>family</code>	of the outcome (given to check if the distribution matches but the check is not done yet)
<code>k</code>	identifies the longitudinal marker among 1:K markers output:

Value

<code>YL.name</code>	name of the outcome
<code>YL</code>	values of the outcome

SurvMS	<i>Simulated multi-state survival dataset</i>
--------	---

Description

A simulated dataset used to illustrate multi-state models (see multi-state model and joint longitudinal and multi-state model in INLAjoint vignette). The corresponding longitudinal dataset is named 'LongMS'.

Usage

```
SurvMS
```

Format

'SurvMS' A list of 3 data frames with 50 rows and 5 columns, 50 rows and 5 columns, 28 rows and 5 columns for each of the 3 transition intensities, respectively:

id Individual id
Tstart Beginning of the 'at-risk' period
Tstop Event time
status Transition indicator
X Continuous covariate ...

Survsim

Simulated competing risks survival dataset

Description

A simulated dataset used to illustrate some models (e.g., example of `joint()` function in the documentation). The corresponding longitudinal dataset is named 'Longsim'.

Usage

Survsim

Format

'Survsim' A data frame with 15 rows and 5 columns:

Id Individual id

deathTimes Event time

ctsX Continuous covariate

binX Binary covariate

Event1, Event2, Event3 Event indicator for the 3 competing risks (individual is censored when the 3 events are 0) ...

Index

- * **competing**
 - INLAjoint.object, 6
- * **datasets**
 - LongMS, 17
 - Longsim, 18
 - SurvMS, 26
 - Survsim, 27
- * **joint**
 - INLAjoint.object, 6
- * **longitudinal**
 - INLAjoint.object, 6
- * **model**
 - INLAjoint.object, 6
- * **multivariate**
 - INLAjoint.object, 6
- * **risks**
 - INLAjoint.object, 6
- * **survival**
 - INLAjoint.object, 6

coef.INLAjoint, 3

family.INLAjoint, 3

fitted.INLAjoint, 4

fixef.INLAjoint, 4

formula.INLAjoint, 5

INLAjoint, 5

INLAjoint.ginv, 6

INLAjoint.object, 6, 14

INLAjoint.rw, 9

INLAjoint.rw2, 9

INLAjoint.scopy.define, 10

joint, 9, 10, 16, 17, 19, 22

joint.rerun, 15

joint.run, 16

logLik.INLAjoint, 17

LongMS, 17

Longsim, 18

nobs.INLAjoint, 18

plot.INLAjoint, 19

predict.INLAjoint, 19

print.plot.INLAjoint, 22

ranef.INLAjoint, 23

setup_FE_model, 23

setup_RE_model, 24

setup_S_model, 24

setup_Y_model, 25

SurvMS, 26

Survsim, 27