Package 'mstate'

December 17, 2020

Version 0.3.1

Date 2020-12-17

Title Data Preparation, Estimation and Prediction in Multi-State Models

Depends survival (>= 3.1)

Imports rlang, data.table, magrittr, lattice, RColorBrewer, viridis

Suggests cmprsk, ggplot2, knitr, rmarkdown

Description
Contains functions for data preparation, descriptives, hazard estimation and prediction with Aalen-Johansen or simulation in competing risks and multi-state models, see Putter, Fiocco, Geskus (2007) <doi:10.1002/sim.2712>.

License GPL (>= 2)

Encoding UTF-8

URL https://www.lumc.nl/org/bds/research/medische-statistiek/survival-analysis/

NeedsCompilation yes

Repository CRAN

RoxygenNote 7.1.1

BugReports https://github.com/hputter/mstate/issues

VignetteBuilder knitr

Author Hein Putter [aut, cre],
Liesbeth C. de Wreede [aut],
Marta Fiocco [aut],
Ronald B. Geskus [ctb],
Edouard F. Bonneville [ctb],
Damjan Manevski [ctb]

Maintainer Hein Putter <H.Putter@lumc.nl>

Date/Publication 2020-12-17 17:00:03 UTC

R topics documented:

mstate-package *Data preparation, estimation and prediction in multi-state models*

Description

Functions for data preparation, descriptives, (hazard) estimation and prediction (Aalen-Johansen) in competing risks and multi-state models.

Details

Package:	mstate
Type:	Package
Version:	0.2.10
Date:	2016-12-03
License:	GPL 2.0

Author(s)

Liesbeth de Wreede, Marta Fiocco, Hein Putter. Maintainer: Hein Putter <H.Putter@lumc.nl>

References

Putter H, Fiocco M, Geskus RB (2007). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* **26**, 2389–2430.

de Wreede LC, Fiocco M, and Putter H (2010). The mstate package for estimation and prediction in non- and semi-parametric multi-state and competing risks models. *Computer Methods and Programs in Biomedicine* **99**, 261–274.

de Wreede LC, Fiocco M, and Putter H (2011). mstate: An R Package for the Analysis of Competing Risks and Multi-State Models. *Journal of Statistical Software*, Volume 38, Issue 7.

aidssi *Data from the Amsterdam Cohort Studies on HIV infection and AIDS*

Description

These data sets give the times (in years) from HIV infection to AIDS, SI switch and death in 329 men who have sex with men (MSM). Data are from the period until combination anti-retroviral therapy became available (1996). For more background information on the cohort, ccr5 and SI, see Geskus *et al.* (2000, 2003)

Format

aidssi

patnr:	Patient identification number
time:	Time from HIV infection to first of SI appearance and AIDS, or last follow-up
status:	Event indicator; 0 = censored, 1 = AIDS, 2 = SI appearance
cause:	Failure cause; factor with levels "event-free", "AIDS", "SI"
ccr5:	CCR5 genotype; factor with levels "WW" (wild type allele on both chromosomes), "WM" (mutant allele on one chromosome)

aidssi2

patnr:	Patient identification number
entry.time:	Time from HIV infection to cohort entry. Value is zero if HIV infection occurred while in follow-up.
aids.time:	Time from HIV infection to AIDS, or last follow-up if AIDS was not observed
aids.stat:	Event indicator with respect to AIDS, with values 0 (censored) and 1 (AIDS)
si.time:	Time from HIV infection to SI switch, or last follow-up if SI switch was not observed
si.stat:	Event indicator with respect to SI switch, with values 0 (no switch) and 1 (switch)
death.time:	Time from HIV infection to death, or last follow-up if death was not observed
death.stat:	Event indicator with respect to death; 0 = alive, 1 = dead
age.inf:	Age at HIV infection
ccr5:	CCR5 genotype; factor with levels "WW" (wild type allele on both chromosomes), "WM" (mutant allele on one chromosome)

Details

`aidssi` contains follow-up data until the first of AIDS and SI switch. It was used as example for the competing risks analyses in Putter, Fiocco, Geskus (2007) and in Geskus (2016).

`aidssi2` extends the `aidssi` data set in three ways. First, it considers events after the initial one. Second, it includes the entry times of the individuals that entered the study after HIV infection. Third, age at HIV infection has been added as extra covariable. Numbers differ slightly from the `aidssi` data set. Some individuals were diagnosed with AIDS only when they died and others had their last follow-up at AIDS diagnosis. In order to prevent two transitions to happen at the same time, their time to AIDS was shortened by 0.25 years. This data set was used as example for the multi-state analyses in Geskus (2016).

Source

Geskus RB (2000). On the inclusion of prevalent cases in HIV/AIDS natural history studies through a marker-based estimate of time since seroconversion. *Statistics in Medicine* **19**, 1753–1769.

Geskus RB, Miedema FA, Goudsmit J, Reiss P, Schuitemaker H, Coutinho RA (2003). Prediction of residual time to AIDS and death based on markers and cofactors. *Journal of AIDS* **32**, 514–521.

References

Geskus, Ronald B. (2016). *Data Analysis with Competing Risks and Intermediate States.* CRC Press, Boca Raton.

Putter H, Fiocco M, Geskus RB (2007). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* **26**, 2389–2430.

bmt *BMT data from Klein and Moeschberger*

Description

A data frame of 137 rows (patients) and 22 columns. The included variables are

group Disease group; 1 = ALL, 2 = AML Low Risk, 3 = AML High Risk

t1 Time in days to death or last follow-up

t2 Disease-free survival time in days (time to relapse, death or last follow-up)

d1 Death indicator; 1 = dead, 0 = alive

d2 Relapse indicator; 1 = relapsed, 0 = disease-free

d3 Disease-free survival indicator; 1 = dead or relapsed, 0 = alive and disease-free)

ta Time in days to Acute Graft-versus-Host Disease (AGVHD)

da Acute GVHD indicator; 1 = Acute GVHD, 0 = No Acute GVHD

tc Time (days) to Chronic Graft-vrsus-Host Disease (CGVHD)

dc Chronic GVHD indicator; 1 = Chronic GVHD, 0 = No Chronic GVHD

tp Time (days) to platelet recovery

dp Platelet recovery indicator; 1 = platelets returned to normal, 0 = platelets never returned to normal

z1 Patient age in years

z2 Donor age in years

z3 Patient sex; 1 = male, 0 = female

z4 Donor sex; 1 = male, 0 = female

z5 Patient CMV status; 1 = CMV positive, 0 = CMV negative

z6 Donor CMV status; 1 = CMV positive, 0 = CMV negative

z7 Waiting time to transplant in days

z8 FAB; 1 = FAB grade 4 or 5 and AML, 0 = Otherwise

z9 Hospital; 1 = The Ohio State University, 2 = Alferd , 3 = St. Vincent, 4 = Hahnemann

z10 MTX used as a Graft-versus-Host prophylactic; 1 = yes, 0 = no

Format

A data frame, see `data.frame`.

References

Klein and Moeschberger (1997). *Survival Analysis Techniques for Censored and Truncated Data*, Springer, New York.

`crprep.default` *Function to create weighted data set for competing risks analyses*

Description

This function converts a dataset that is in short format (one subject per line) into a counting process format with time-varying weights that correct for right censored and left truncated data. With this data set, analyses based on the subdistribution hazard can be performed.

Usage

```
## Default S3 method:
crprep(
  Tstop,
  status,
  data,
  trans = 1,
  cens = 0,
  Tstart = 0,
  id,
  strata,
  keep,
  shorten = TRUE,
  rm.na = TRUE,
  origin = 0,
  prec.factor = 1000,
  ...
)
```

Arguments

`Tstop`	Either 1) a vector containing the time at which the follow-up is ended, or 2) a character string indicating the column name in `data` that contains the end times (see Details).
`status`	Either 1) a vector describing status at end of follow-up, having the same length as `Tstop`, or 2) a character string indicating the column name that contains this information.
`data`	Data frame in which to interpret `Tstart`, `status`, `Tstart`, `id`, `strata` and `keep`, if given as character value (specification 2, "by name").
`trans`	Values of `status` for which weights are to be calculated.
`cens`	Value that denotes censoring in `status` column.
`Tstart`	Either 1) a vector containing the time at which the follow-up is started, having the same length as `Tstop`, or 2) a character string indicating the column name that contains the entry times, or 3) one numeric value in case it is the same for every subject. Default is 0.

id	Either 1) a vector, having the same length as Tstop, containing the subject identifiers, or 2) a character string indicating the column name containing these subject identifiers. If not provided, a column id is created with subjects having values 1,...,n.
strata	Either 1) a vector of the same length as Tstop, or 2) a character string indicating the column name that contains this information. Weights are calculated for per value in this vector.
keep	Either 1) a data frame or matrix or a numeric or factor vector containing covariate(s) that need to be retained in the output dataset. Number of rows/length should correspond with Tstop, or 2) a character vector containing the column names of these covariates in data.
shorten	Logical. If true, number of rows in output is reduced by collapsing rows within a subject in which weights do not change.
rm.na	Logical. If true, rows for which status is missing are deleted.
origin	Substract origin time units from all Tstop and Tstart times.
prec.factor	Factor by which to multiply the machine's precision. Censoring and truncation times are shifted by prec.factor*precision if event times and censoring/truncation times are equal.
...	Further arguments to be passed to or from other methods. They are ignored in this function.

Details

For each event type as specified via trans, individuals with a competing event remain in the risk set with weights that are determined by the product-limit forms of the time-to-censoring and time-to-entry estimates. Typically, their weights change over follow-up, and therefore such individuals are split into several rows. Censoring weights are always computed. Truncation weights are computed only if Tstart is specified.

If several event types are specified at once, regression analyses using the stacked format data set can be performed (see Putter et al. 2007 and Chapter 4 in Geskus 2016). The data set can also be used for a regression on the cause-specific hazard by restricting to the subset subset=count==0.

Missing values are allowed in Tstop, status, Tstart, strata and keep. Rows for which Tstart or Tstart is missing are deleted.

There are two ways to supply the data. If given "by value" (option 1), the actual data vectors are used. If given "by name" (option 2), the column names are specified, which are read from the data set in data. In general, the second option is preferred.

If data are given by value, the following holds for the naming of the columns in the output data set. If keep, strata or id is a vector from a (sub)-list, e.g. obj$name2$name1, then the column name is based on the most inner part (i.e.\ "name1"). If it is a vector of the form obj[,"name1"], then the column is named "name1". For all other vector specifications, the name is copied as is. If keep is a data.frame or a named matrix, the same names are used for the covariate columns in the output data set. If keep is a matrix without names, then the covariate columns are given the names "V1" until "Vk".

The current function does not allow to create a weighted data set in which the censoring and/or truncation mechanisms depend on covariates via a regression model.

Value

A data frame in long (counting process) format containing the covariates (replicated per subject). The following column names are used:

`Tstart`	start dates of dataset
`Tstop`	stop dates of dataset
`status`	status of the subject at the end of that row
`weight.cens`	weights due to censoring mechanism
`weight.trunc`	weights due to truncation mechanism (if present)
`count`	row number within subject and event type under consideration
`failcode`	event type under consideration

The first column is the subject identifier. If the argument "id" is missing, it has values 1:n and is named "id". Otherwise the information is taken from the `id` argument.

Variables as specified in `strata` and/or `keep` are included as well (see Details).

Author(s)

Ronald Geskus

References

Geskus RB (2011). Cause-Specific Cumulative Incidence Estimation and the Fine and Gray Model Under Both Left Truncation and Right Censoring. *Biometrics* **67**, 39–49.

Geskus, Ronald B. (2016). *Data Analysis with Competing Risks and Intermediate States.* CRC Press, Boca Raton.

Putter H, Fiocco M, Geskus RB (2007). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* **26**, 2389–2430.

Examples

```
data(aidssi)
aidssi.w <- crprep("time", "cause", data=aidssi, trans=c("AIDS","SI"),
                   cens="event-free", id="patnr", keep="ccr5")

# calculate cause-specific cumulative incidence, no truncation,
# compare with Cuminc (also from mstate)
ci <- Cuminc(aidssi$time, aidssi$status)
sf <- survfit(Surv(Tstart,Tstop,status=="AIDS")~1, data=aidssi.w,
              weight=weight.cens, subset=failcode=="AIDS")
plot(sf, fun="event", mark.time=FALSE)
lines(CI.1~time,data=ci,type="s",col="red")
sf <- survfit(Surv(Tstart,Tstop,status=="SI")~1, data=aidssi.w,
              weight=weight.cens, subset=failcode=="SI")
plot(sf, fun="event", mark.time=FALSE)
lines(CI.2~time,data=ci,type="s",col="red")
```

```
# Fine and Gray regression for cause 1
cw <- coxph(Surv(Tstart,Tstop,status=="AIDS")~ccr5, data=aidssi.w,
      weight=weight.cens, subset=failcode=="AIDS")
cw
# This can be checked with the results of crr (cmprsk)
# crr(ftime=aidssi$time, fstatus=aidssi$status, cov1=as.numeric(aidssi$ccr5))

# Gray's log-rank test
aidssi.wCCR <- crprep("time", "cause", data=aidssi, trans=c("AIDS","SI"),
                      cens="event-free", id="patnr", strata="ccr5")
test.AIDS <- coxph(Surv(Tstart,Tstop,status=="AIDS")~ccr5, data=aidssi.wCCR,
                   weights=weight.cens, subset=failcode=="AIDS")
test.SI <- coxph(Surv(Tstart,Tstop,status=="SI")~ccr5, data=aidssi.wCCR,
                 weights=weight.cens, subset=failcode=="SI")
## score test statistic and p-value
c(test.AIDS$score, 1-pchisq(test.AIDS$score,1)) # AIDS
c(test.SI$score, 1-pchisq(test.SI$score,1))     # SI
# This can be compared with the results of cuminc (cmprsk)
# with(aidssi, cuminc(time, status, group=ccr5)$Tests)
# Note: results are not exactly the same
```

`Cuminc` *Calculate nonparametric cumulative incidence functions and associated standard errors*

Description

This function computes nonparametric cumulative incidence functions and associated standard errors for each value of a group variable.

Usage

```
Cuminc(
  time,
  status,
  data,
  group,
  failcodes,
  na.status = c("remove", "extra"),
  variance = TRUE
)
```

Arguments

`time`	Either 1) a numeric vector containing the failure times or 2) a string containing the column name indicating these failure times
`status`	Either 1) a numeric, factor or character vector containing the failure codes or 2) a string containing the column name indicating these failure codes

data	When appropriate, a data frame containing time, status and/or group variables
group	Optionally, name of column in data indicating a grouping variable; cumulative incidence functions are calculated for each value or level of group. If missing no groups are considered
failcodes	A vector indicating which values of status are considered as different causes of failure; other values of status are considered as censorings. If missing and status is numeric, it is assumed that 0 is censoring and all other values indicate failcodes; if missing and status is character or factor, then it is assumed that each of the levels/values of status is a cause of failure
na.status	One of "remove" (default) or "extra", indicating whether subjects with missing cause of failure should be removed or whether missing cause of failure should be treated as a separate cause of failure
variance	Logical value, indicating whether the standard errors of the cumulative incidences should be output (TRUE, the default) or not

Details

The estimated cumulative incidences are as described in Putter, Fiocco & Geskus (2007); the standard errors are the square roots of the Greenwood variance estimators, see eg. Andersen, Borgan, Gill & Keiding (1993), de Wreede, Fiocco & Putter (2009), and they correspond to the variances in eg. Marubini & Valsecchi (1995). In case of no censoring, the estimated cumulative incidences and variances reduce to simple binomial frequencies and their variances.

Value

An object of class "Cuminc", which is a data frame containing the estimated failure-free probabilities and cumulative incidences and their standard errors. The names of the dataframe are time, Surv, seSurv, and cuminc and secuminc followed by the values or levels of the failcodes. If group was specified, a group variable is included with the same name and values/levels as the original grouping variable, and with estimated cumulative incidences (SE) for each value/level of group.

Cuminc is now simply a wrapper around survfit of the survival package with type="mstate", only maintained for backward compatibility. The survfit object is kept as attribute (attr("survfit")), and the print, plot and summary functions are simply print, plot and summary applied to the survfit object. Subsetting the "Cuminc" object results in subsetting the data frame, not in subsetting the survfit object.

Author(s)

Hein Putter <H.Putter@lumc.nl>

References

Andersen PK, Borgan O, Gill RD, Keiding N (1993). *Statistical Models Based on Counting Processes*. Springer, New York.

Marubini E, Valsecchi MG (1995). *Analysing Survival Data from Clinical Trials and Observational Studies*. Wiley, New York.

Putter H, Fiocco M, Geskus RB (2007). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* **26**, 2389–2430.

de Wreede L, Fiocco M, Putter H (2009). The mstate package for estimation and prediction in non- and semi-parametric multi-state models. Submitted. http://www.msbi.nl/multistate.

Examples

```
### These data were used in Putter, Fiocco & Geskus (2007)
data(aidssi)
ci <- Cuminc(time=aidssi$time, status=aidssi$status)
head(ci); tail(ci)
ci <- Cuminc(time="time", status="status", data=aidssi, group="ccr5")
head(ci); tail(ci)

### Some fake data
fake <- data.frame(surv=c(seq(2,10,by=2),seq(1,13,by=3),seq(1,9,by=2),seq(1,13,by=3)),
                   stat=rep(0:3,5),Tstage=c(1:4,rep(1:4,rep(4,4))))
fake$stat[fake$stat==0 & fake$Tstage==2] <- 3
fake$stat[fake$stat==3 & fake$Tstage==1] <- 2
fake
Cuminc(time="surv", status="stat", data=fake)
# If we remove all entries with status=0,
# we should get binomial sample probabilities and corresponding SEs
fake0 <- fake[fake$stat!=0,]
Cuminc(time="surv", status="stat", data=fake0)
```

---

cutLMms *Cut a multi-state data set at a landmark time point*

Description

Given a dataset in long format, for instance generated by msprep, this function cuts a multi-state data frame (object of type "msdata") at a landmark time point LM. Administrative censoring can be applied at time cens, equal for all individuals.

Usage

```
cutLMms(msdata, LM, cens)
```

Arguments

msdata	An object of class "msdata", such as output by msprep
LM	The landmark time point at which the cut is to be made
cens	The time point at which administrative censoring is to be applied; if missing, no administrative censoring will be applied

Details

The function has a similar purpose as the `cutLM` function in the `dynpred` package. Only follow-up after a landmark time point LM is considered, so all subjects who are no longer at risk are removed. Column `time` is updated based on the new Tstart and Tstop.

Value

An object of class `"msdata"` again, containing only follow-up data after LM. The data frame contains an extra column `Tentry` with the time of entry into the present state.

Author(s)

Hein Putter <H.Putter@lumc.nl>

References

L. C. de Wreede, M. Fiocco, and H. Putter (2011). mstate: An R Package for the Analysis of Competing Risks and Multi-State Models. Journal of Statistical Software 38: 7.

Examples

```
tmat <- trans.illdeath(names=c("Tx","PR","RelDeath"))
data(ebmt3) # data from Section 4 of Putter, Fiocco & Geskus (2007)
msebmt <- msprep(time=c(NA,"prtime","rfstime"),status=c(NA,"prstat","rfsstat"),
data=ebmt3,trans=tmat)
# Cut at 5 years
cutLMms(msebmt, LM=1826)
events(cutLMms(msebmt, LM=1826))
```

EBMT cause of death data

Data from the European Society for Blood and Marrow Transplantation (EBMT)

Description

A data frame of 8966 patients transplanted at the EBMT. The included variables are

id Patient identification number

time Time in months from transplantation to death or last follow-up

status Survival status; 0 = censored; 1,...,6 = death due to the following causes: Relapse (1), GvHD (2), Bacterial infections (3), Viral infections (4), Fungal infections (5), Other causes (6)

cod Cause of death as factor with levels "Alive", "Relapse", "GvHD", "Bacterial", "Viral", "Fungal", "Other"

dissub Disease subclassification; factor with levels "AML", "ALL", "CML"

match Donor-recipient gender match; factor with levels "No gender mismatch", "Gender mismatch"

tcd T-cell depletion; factor with levels "No TCD", "TCD", "Unknown"

year Year of transplantation; factor with levels "1985-1989", "1990-1994", "1995-1998"

age Patient age at transplant; factor with levels "<=20", "20-40", ">40"

Format

A data frame, see `data.frame`.

Source

We acknowledge the European Society for Blood and Marrow Transplantation (EBMT) for making available these data. Disclaimer: these data were simplified for the purpose of illustration of the analysis of competing risks and multi-state models and do not reflect any real life situation. No clinical conclusions should be drawn from these data.

References

Fiocco M, Putter H, van Houwelingen JC (2005). Reduced rank proportional hazards model for competing risks. *Biostatistics* **6**, 465–478.

`EBMT data` *Data from the European Society for Blood and Marrow Transplantation (EBMT)*

Description

A data frame of 2279 patients transplanted at the EBMT between 1985 and 1998. These data were used in Fiocco, Putter & van Houwelingen (2008), van Houwelingen & Putter (2008, 2012) and de Wreede, Fiocco & Putter (2011). The included variables are

id Patient identification number

rec Time in days from transplantation to recovery or last follow-up

rec.s Recovery status; 1 = recovery, 0 = censored

ae Time in days from transplantation to adverse event (AE) or last follow-up

ae.s Adverse event status; 1 = adverse event, 0 = censored

recae Time in days from transplantation to both recovery and AE or last follow-up

recae.s Recovery and AE status; 1 = both recovery and AE, 0 = no recovery or no AE or censored

rel Time in days from transplantation to relapse or last follow-up

rel.s Relapse status; 1 = relapse, 0 = censored

srv Time in days from transplantation to death or last follow-up

srv.s Relapse status; 1 = dead, 0 = censored

year Year of transplantation; factor with levels "1985-1989", "1990-1994", "1995-1998"

agecl Patient age at transplant; factor with levels "<=20", "20-40", ">40"

proph Prophylaxis; factor with levels "no", "yes"

match Donor-recipient gender match; factor with levels "no gender mismatch", "gender mismatch"

Format

A data frame, see `data.frame`.

Source

We acknowledge the European Society for Blood and Marrow Transplantation (EBMT) for making available these data. Disclaimer: these data were simplified for the purpose of illustration of the analysis of competing risks and multi-state models and do not reflect any real life situation. No clinical conclusions should be drawn from these data.

References

Fiocco M, Putter H, van Houwelingen HC (2008). Reduced-rank proportional hazards regression and simulation-based prediction for multi-state models. *Statistics in Medicine* **27**, 4340–4358.

van Houwelingen HC, Putter H (2008). Dynamic predicting by landmarking as an alternative for multi-state modeling: an application to acute lymphoid leukemia data. *Lifetime Data Anal* **14**, 447–463.

van Houwelingen HC, Putter H (2012). Dynamic Prediction in Clinical Survival Analaysis. Chapman & Hall/CRC Press, Boca Raton.

de Wreede LC, Fiocco M, and Putter H (2011). mstate: An R Package for the Analysis of Competing Risks and Multi-State Models. *Journal of Statistical Software*, Volume 38, Issue 7.

`EBMT platelet recovery data`

Data from the European Society for Blood and Marrow Transplantation (EBMT)

Description

A data frame of 2204 patients transplanted at the EBMT between 1995 and 1998. These data were used in Section 4 of the tutorial on competing risks and multi-state models (Putter, Fiocco & Geskus, 2007). The included variables are

id Patient identification number

prtime Time in days from transplantation to platelet recovery or last follow-up

prstat Platelet recovery status; 1 = platelet recovery, 0 = censored

rfstime Time in days from transplantation to relapse or death or last follow-up (relapse-free survival time)

rfsstat Relapse-free survival status; 1 = relapsed or dead, 0 = censored

dissub Disease subclassification; factor with levels "AML", "ALL", "CML"

age Patient age at transplant; factor with levels "<=20", "20-40", ">40"

drmatch Donor-recipient gender match; factor with levels "No gender mismatch", "Gender mismatch"

tcd T-cell depletion; factor with levels "No TCD", "TCD"

Format

A data frame, see `data.frame`.

Source

We acknowledge the European Society for Blood and Marrow Transplantation (EBMT) for making available these data. Disclaimer: these data were simplified for the purpose of illustration of the analysis of competing risks and multi-state models and do not reflect any real life situation. No clinical conclusions should be drawn from these data.

References

Putter H, Fiocco M, Geskus RB (2007). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* **26**, 2389–2430.

`EBMT year of relapse data`

Data from the European Society for Blood and Marrow Transplantation (EBMT)

Description

A data frame of 1977 patients transplanted for CML. The included variables are

patid Patient identification number

srv Time in days from transplantation to death or last follow-up

srvstat Survival status; 1 = death; 0 = censored

rel Time in days from transplantation to relapse or last follow-up

relstat Relapse status; 1 = relapsed; 0 = censored

yrel Calendar year of relapse; factor with levels "1993-1996"," 1997-1999", "2000-"

age Patient age at transplant (years)

score Gratwohl score; factor with levels "Low risk", "Medium risk", "High risk"

Format

A data frame, see `data.frame`.

Source

We acknowledge the European Society for Blood and Marrow Transplantation (EBMT) for making available these data. Disclaimer: these data were simplified for the purpose of illustration of the analysis of competing risks and multi-state models and do not reflect any real life situation. No clinical conclusions should be drawn from these data.

ELOS *Expected length of stay*

Description

Given a "probtrans" object, ELOS calculates the (restricted) expected length of stay in each of the states of the multi-state model.

Usage

```
ELOS(pt, tau)
```

Arguments

pt	An object of class "probtrans"
tau	The horizon until which ELOS is calculated; if missing, the maximum of the observed transition times is taken

Details

The object pt needs to be a "probtrans" object, obtained with forward prediction (the default, direction="forward", in the call to probtrans). The restriction to tau is there because, as in ordinary survival analysis, the probability of being in a state can be positive until infinity, resulting in infinite values. The (restricted, until tau) expected length of stay in state h, given in state g at time s, is given by the integral from s to tau of P_gh(s,t), see for instance Beyersmann and Putter (2014).

Value

A K x K matrix (with K number of states), with the (g,h)'th element containing E_gh(s,tau). The starting time point s is inferred from pt (the smallest time point, should be equal to the predt value in the call to probtrans. The row- and column names of the matrix have been named "from1" until "fromK" and "in1" until "inK", respectively.

Author(s)

Hein Putter <H.Putter@lumc.nl>

Examples

```
# transition matrix for illness-death model
tmat <- trans.illdeath()
# data in wide format, for transition 1 this is dataset E1 of
# Therneau & Grambsch (2000)
tg <- data.frame(illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
        dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
        x1=c(1,1,1,0,0,0),x2=c(6:1))
# data in long format using msprep
tglong <- msprep(time=c(NA,"illt","dt"),status=c(NA,"ills","ds"),
data=tg,keep=c("x1","x2"),trans=tmat)
# events
events(tglong)
table(tglong$status,tglong$to,tglong$from)
# expanded covariates
tglong <- expand.covs(tglong,c("x1","x2"))
# Cox model with different covariate
cx <- coxph(Surv(Tstart,Tstop,status)~x1.1+x2.2+strata(trans),
        data=tglong,method="breslow")
summary(cx)
# new data, to check whether results are the same for transition 1 as
# those in appendix E.1 of Therneau & Grambsch (2000)
newdata <- data.frame(trans=1:3,x1.1=c(0,0,0),x2.2=c(0,1,0),strata=1:3)
HvH <- msfit(cx,newdata,trans=tmat)
# probtrans
pt <- probtrans(HvH,predt=0)
# ELOS until last observed time point
ELOS(pt)
# Restricted ELOS until tau=10
ELOS(pt, tau=10)
```

`etm2msdata` *Converts between etm and msdata format*

Description

Converts multi-state data back and forth between etm and msdata formats. Covariates have to be dealt with separately.

Usage

```
etm2msdata(etmdata, id, tra, covs)
```

Arguments

`etmdata`	Multi-state data in `etm` format
`id`	Column name identifying the subject id

tra	Transition matrix in etm format
covs	Vector of column names containing covariates to be included

Details

msdata2etm will convert from msdata format to etm format; etm2msdata will convert from etm format to msdata format. Both msdata2etm and etm2msdata work with basic time-fixed covariates. Time-dependent covariates are not supported. The function msdata2etm will work for transition-specific covariates, but the result does not really make much sense when used in etm.

Author(s)

Hein Putter <H.Putter@lumc.nl>

Examples

```
# Transition matrix for illness-death model
tmat <- trans.illdeath()
# Data in wide format, for transition 1 this is dataset E1 of
# Therneau & Grambsch (T&G)
tg <- data.frame(id=1:6,illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
                 dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
                 x1=c(1,1,1,0,0,0),x2=c(6:1))
# Data in long format using msprep
tglong <- msprep(time=c(NA,"illt","dt"),status=c(NA,"ills","ds"),
                 data=tg,keep=c("x1","x2"),trans=tmat, id="id")
# Same thing in etm format
tra <- trans2tra(tmat)
tgetm <- msdata2etm(tglong, id="id")
tgetm <- msdata2etm(tglong, id="id", covs=c("x1", "x2")) # with covariates
# And back
etm2msdata(tgetm, id="id", tra=tra)
etm2msdata(tgetm, id="id", tra=tra, covs=c("x1", "x2")) # with covariates
```

---

events *Count number of observed transitions*

Description

Given a dataset in long format, for instance generated by msprep, and a transition matrix for the multi-state model, this function counts the number of observed transitions in the multi-state model and gives their percentages.

Usage

```
events(msdata)
```

Arguments

msdata	An object of class "msdata", such as output by msprep

Details

Although msdata does not need to be the result of a call to msprep, it does need to be an object of class "msdata", which is essentially a data frame in long format, with one row for each transition for which the subject is at risk. The columns from, to, and status need to be present, with appropriate meaning, see msprep. The msdata argument needs to have a "trans" attributes, which holds the transition matrix of the multi-state model.

Value

A list containing two tables, the first, called Frequencies, with the number of observed transitions in the multi-state model occurring in msdata, the second, called Proportions, with the corresponding proportions.

Author(s)

Hein Putter <H.Putter@lumc.nl>

Examples

```
tmat <- trans.illdeath(names=c("Tx","PR","RelDeath"))
data(ebmt3) # data from Section 4 of Putter, Fiocco & Geskus (2007)
msebmt <- msprep(time=c(NA,"prtime","rfstime"),status=c(NA,"prstat","rfsstat"),
data=ebmt3,trans=tmat)
events(msebmt) # see Fig 13 of Putter, Fiocco & Geskus (2007)
```

---

expand.covs *Expand covariates in competing risks dataset in stacked format*

Description

Given a competing risks dataset in stacked format, and one or more covariates, this function adds type-specific covariates to the dataset. The original dataset with the type-specific covariates appended is returned.

Usage

```
expand.covs(data, ...)
```

Arguments

data	An object of class "msdata", such as output by msprep
...	Further arguments to be passed to or from other methods. They are ignored in this function.

Details

Type-specific covariates can be used to analyse separate effects on all event types in a single analysis based on a stacked data set (Putter, Fiocco & Geskus (2007) and Geskus (2016)). It is only unambiguously defined for numeric covariates or for explicit codings. Rows that contain the data for that specific event type have the value copied from the original covariate in case it is numeric. In all other rows it has the value zero. If the covariate is a factor, it will be expanded on the design matrix given by `model.matrix`. For standard "treatment contrasts" this means that dummy variables are created. If the covariate is a factor, the column name combines the name of the covariate with the specific event type. If `longnames=TRUE`, both parts are intersected by the specific labels in the coding. Missing values in the basic covariates are allowed and result in missing values in the expanded covariates.

Value

An data frame object of the same class as the data argument, containing the design matrix for the type-specific covariates, either on its own (`append=FALSE`) or appended to the data (`append=TRUE`).

Author(s)

Ronald Geskus and Hein Putter <H.Putter@lumc.nl>

References

Putter H, Fiocco M, Geskus RB (2007). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* **26**, 2389–2430.

Geskus, Ronald B. (2016). *Data Analysis with Competing Risks and Intermediate States.* CRC Press, Boca Raton.

See Also

`expand.covs.msdata`.

Examples

```
# small data set in stacked format
tg <- data.frame(time=c(5,5,1,1,9,9),status=c(1,0,2,2,0,1),failcode=rep(c("I","II"),3),
        x1=c(1,1,2,2,2,2),x2=c(3,3,2,2,1,1))
tg$x1 <- factor(tg$x1,labels=c("male","female"))
# expanded covariates
expand.covs(tg,covs=c("x1","x2"))
expand.covs(tg,covs=c("x1","x2"),longnames=TRUE)
expand.covs(tg,covs=c("x1","x2"),append=FALSE)
```

`expand.covs.msdata` *Expand covariates in multi-state dataset in long format*

Description

Given a multi-state dataset in long format, and one or more covariates, this function adds transition-specific covariates, expanding the original covariate(s), to the dataset. The original dataset with the transition-specific covariates appended is returned.

Usage

```
## S3 method for class 'msdata'
expand.covs(data, covs, append = TRUE, longnames = TRUE, ...)
```

Arguments

`data`	An object of class `"msdata"`, such as output by `msprep`
`covs`	A character vector containing the names of the covariates in `data` to be expanded
`append`	Logical value indicating whether or not the design matrix for the expanded covariates should be appended to the data (default=`TRUE`)
`longnames`	Logical value indicating whether or not the labels are to be used for the names of the expanded covariates that are categorical (default=`TRUE`); in case of `FALSE` numbers from 1 up to the number of contrasts are used
`...`	Further arguments to be passed to or from other methods. They are ignored in this function.

Details

For a given basic covariate `Z`, the transition-specific covariate for transition `s` is called `Z.s`. The concept of transition-specific covariates in the context of multi-state models was introduced by Andersen, Hansen & Keiding (1991), see also Putter, Fiocco & Geskus (2007). It is only unambiguously defined for numeric covariates or for explicit codings. Then it will take the value 0 for all rows in the long format dataframe for which `trans` does not equal `s`. For the rows for which `trans` equals `s`, the original value of `Z` is copied. In `expand.covs`, when a given covariate is a factor, it will be expanded on the design matrix given by `model.matrix`. Missing values in the basic covariates are allowed and result in missing values in the expanded covariates.

Value

An object of class 'msdata', containing the design matrix for the transition- specific covariates, either on its own (`append=FALSE`) or appended to the data (`append=TRUE`).

Author(s)

Hein Putter <H.Putter@lumc.nl>

References

Andersen PK, Hansen LS, Keiding N (1991). Non- and semi-parametric estimation of transition probabilities from censored observation of a non-homogeneous Markov process. *Scandinavian Journal of Statistics* **18**, 153–167.

Putter H, Fiocco M, Geskus RB (2007). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* **26**, 2389–2430.

Examples

```
# transition matrix for illness-death model
tmat <- trans.illdeath()
# small data set in wide format
tg <- data.frame(illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
        dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
        x1=c(1,1,1,2,2,2),x2=c(6:1))
tg$x1 <- factor(tg$x1,labels=c("male","female"))
# data in long format using msprep
tglong <- msprep(time=c(NA,"illt","dt"),
        status=c(NA,"ills","ds"),data=tg,
        keep=c("x1","x2"),trans=tmat)
# expanded covariates
expand.covs(tglong,c("x1","x2"),append=FALSE)
expand.covs(tglong,"x1")
expand.covs(tglong,"x1",longnames=FALSE)
```

Liver cirrhosis data *Abnormal prothrombin levels in liver cirrhosis*

Description

A data frame of 488 liver cirrhosis patients from a randomized clinical trial concerning prednisone treatment in these patients. The dataset is in long format. The included variables are

id Patient identification number

from Starting state

to Receiving state

trans Transition number

Tstart Starting time

Tstop Transition time

status Status variable; 1=transition, 0=censored

treat Treatment; factor with levels "Placebo", "Prednisone"

Format

A data frame, see `data.frame`.

Details

This data was kindly provided by Per Kragh Andersen. It was introduced in Andersen, Borgan, Gill & Keiding (1993), Example 1.3.12, and used as illustration for computation of transition probabilities in multi-state models, see Sections IV.4 (Example IV.4.4) and VII.2 (Example VII.2.10).

References

Andersen PK, Borgan O, Gill RD, Keiding N (1993). *Statistical Models Based on Counting Processes*. Springer, New York.

`LMAJ` *Landmark Aalen-Johansen method*

Description

This function implements the landmark Aalen-Johansen method of Putter & Spitoni (2016) for non-parametric estimation of transition probabilities in non-Markov models.

Usage

```
LMAJ(msdata, s, from, method = c("aalen", "greenwood"))
```

Arguments

`msdata`	An "`msdata`" object, as for instance prepared by `link{msprep}`
`s`	The prediction time point s from which transition probabilities are to be obtained
`from`	Either a single state or a set of states in the state space 1,...,S
`method`	The method for calculating variances, as in `probtrans`

Value

A data frame containing estimates and associated standard errors of the transition probabilities P(X(t)=k | X(s) in `from`) with `s` and `from` the arguments of the function.

Author(s)

Hein Putter <H.Putter@lumc.nl>

References

H. Putter and C. Spitoni (2016). Estimators of transition probabilities in non-Markov multi-state models. Submitted.

Examples

```
data(prothr)
tmat <- attr(prothr, "trans")
pr0 <- subset(prothr, treat=="Placebo")
attr(pr0, "trans") <- tmat
pr1 <- subset(prothr, treat=="Prednisone")
attr(pr1, "trans") <- tmat
c0 <- coxph(Surv(Tstart, Tstop, status) ~ strata(trans), data=pr0)
c1 <- coxph(Surv(Tstart, Tstop, status) ~ strata(trans), data=pr1)
msf0 <- msfit(c0, trans=tmat)
msf1 <- msfit(c1, trans=tmat)
# Comparison as in Figure 2 of Titman (2015)
# Aalen-Johansen
pt0 <- probtrans(msf0, predt=1000)[[2]]
pt1 <- probtrans(msf1, predt=1000)[[2]]
par(mfrow=c(1,2))
plot(pt0$time, pt0$pstate1, type="s", lwd=2, xlim=c(1000,4000), ylim=c(0,0.61),
     xlab="Time since randomisation (days)", ylab="Probability")
lines(pt1$time, pt1$pstate1, type="s", lwd=2, lty=3)
legend("topright", c("Placebo", "Prednisone"), lwd=2, lty=1:2, bty="n")
title(main="Aalen-Johansen")
# Landmark Aalen-Johansen
LMpt0 <- LMAJ(msdata=pr0, s=1000, from=2)
LMpt1 <- LMAJ(msdata=pr1, s=1000, from=2)
plot(LMpt0$time, LMpt0$pstate1, type="s", lwd=2, xlim=c(1000,4000), ylim=c(0,0.61),
     xlab="Time since randomisation (days)", ylab="Probability")
lines(LMpt1$time, LMpt1$pstate1, type="s", lwd=2, lty=3)
legend("topright", c("Placebo", "Prednisone"), lwd=2, lty=1:2, bty="n")
title(main="Landmark Aalen-Johansen")
```

`MarkovTest` *Log-rank based test for the validity of the Markov assumption*

Description

Log-rank based test for the validity of the Markov assumption

Usage

```
MarkovTest(
  data,
  id,
  formula = NULL,
  transition,
  grid,
  B = 1000,
  fn = list(function(x) mean(abs(x), na.rm = TRUE)),
```

```
  fn2 = list(function(x) mean(x, na.rm = TRUE)),
  min_time = 0,
  other_weights = NULL,
  dist = c("poisson", "normal")
)
```

Arguments

data	Multi-state data in msdata format. Should also contain (dummy codings of) the relevant covariates; no factors allowed
id	Column name in data containing subject id
formula	Right-hand side of the formula. If NULL will fit with no covariates (formula="1" will also work), offset terms can also be specified.
transition	Transition number of the transition to be tested (in the transition matrix as attribute to data)
grid	Grid of time points at which to compute the statistic
B	Number of wild bootstrap replications to perform
fn	A list of summary functions to be applied to the individual zbar traces (or a list of lists)
fn2	A list of summary functions to be applied to the overall chi-squared trace
min_time	The minimum time for calculating optimal weights
other_weights	Other (than optimal) weights can be specified here
dist	Distribution of wild bootstrap random weights, either "poisson" for centred Poisson (default), or "normal" for standard normal

Details

Function MarkovTest performs the log-rank test described in Titman & Putter (2020). Function optimal_weights_matrix implements the optimal weighting for the state-specific trace. Function optimal_weights_multiple implements the optimal weighting for the chi-squared trace.

Value

MarkovTest returns an object of class "MarkovTest", which is a list with the following items:

orig_stat	Summary statistic for each of the starting states
orig_ch_stat	Overall chi-squared summary statistic
p_stat_wb	P-values corresponding to each of the summary statistics for each starting state
p_ch_stat_wb	P-values for overall chi-squared summary statistic
b_stat_wb	Bootstrap summary statistics for each of the starting states
zbar	Individual traces for each of the starting states
nobs_grid	The number of events after time s for each s in the grid
Nsub	Number of patients who are ever at risk of the transition of interest
est_quant	Pointwise 2.5 and 97.5 quantile limits for each of the traces

obs_chisq_trace	Trace of the chi-squared statistic
nch_wb_trace	Individual values of the chi-squared statistic trace for the wild bootstrap samples
n_wb_trace	Individual values of the log-rank z statistic traces for the wild bootstrap samples
est_cov	Estimated covariance matrix between the log-rank statistics at each grid point
transition	The transition number tested
from	The from state of the transition tested
to	The to state of the transition tested
B	The number of wild bootstrap replications
dist	The distribution used in the wild bootstrap
qualset	Set of qualifying states corresponding to the components of the above traces
coxfit	Fitted coxph object
fn	List of functions applied to state-specific trace
fn2	List of functions applied to overall trace

Author(s)

Andrew Titman <a.titman@lancaster.ac.uk>, transported to mstate by Hein Putter <H.Putter@lumc.nl>

References

Titman AC, Putter H (2020). General tests of the Markov property in multi-state models. *Biostatistics* To appear.

Examples

```
## Not run:
# Example provided by the prothrombin data
data("prothr")
# Apply Markov test to grid of monthly time points over the first 7.5 years
year <- 365.25
month <- year / 12
grid <- month * (1 : 90)
# Markov test for transition 1 (wild bootstrap based on 25 replications, 1000 recommended)
MT <- MarkovTest(prothr, id = "id", transition = 1,
                 grid = grid, B = 25)

# Plot traces
plot(MT, grid, what="states", idx=1:10, states=rownames(attr(prothr, "trans")),
     xlab="Days since randomisation", ylab="Log-rank test statistic",
     main="Transition Normal -> Low")
plot(MT, grid,what="overall", idx=1:10,
     xlab="Days since randomisation", ylab="Chi-square test statistic",
     main="Transition Normal -> Low")

# Example using optimal weights and adjustment for covariates
```

```
oweights_fun <-
  optimal_weights_matrix(prothr, id = "id", grid=grid, transition = 1,
                         other_weights=list(
                           function(x) mean(abs(x),na.rm=TRUE),
                           function(x) max(abs(x),na.rm=TRUE)))

oweights_chi <- optimal_weights_multiple(prothr, id = "id", grid=grid, transition = 1)

# Formula in MarkovTest only works for continuous covariates and dummy coded variables
# No factors allowed
prothr$prednisone <- as.numeric(prothr$treat == "Prednisone")
MT <- MarkovTest(prothr, id = "id",
                 formula = "prednisone",
                 transition = 1,
                 grid = grid, B = 25,
                 fn = oweights_fun,
                 fn2 = list(
                   function(x) weighted.mean(x, w=oweights_chi, na.rm=TRUE),
                   function(x) mean(x, na.rm=TRUE),
                   function(x) max(x, na.rm=TRUE)))

## End(Not run)
```

`msboot` *Bootstrap function in multi-state models*

Description

A generic nonparametric bootstrapping function for multi-state models.

Usage

```
msboot(theta, data, B = 5, id = "id", verbose = 0, ...)
```

Arguments

`theta`	A function of `data` and perhaps other arguments, returning the value of the statistic to be bootstrapped; the output of theta should be a scalar or numeric vector
`data`	An object of class 'msdata', such as output from `msprep`
`B`	The number of bootstrap replications; the default is taken to be quite small (5) since bootstrapping can be time-consuming
`id`	Character string indicating which column identifies the subjects to be resampled
`verbose`	The level of output; default 0 = no output, 1 = print the replication
`...`	Any further arguments to the function `theta`

Details

The function msboot samples randomly with replacement subjects from the original dataset data. The individuals are identified with id, and bootstrap datasets are produced by concatenating all selected rows.

Value

Matrix of dimension (length of output of theta) x B, with b'th column being the value of theta for the b'th bootstrap dataset

Author(s)

Marta Fiocco, Hein Putter <H.Putter@lumc.nl>

References

Fiocco M, Putter H, van Houwelingen HC (2008). Reduced-rank proportional hazards regression and simulation-based prediction for multi-state models. *Statistics in Medicine* **27**, 4340–4358.

Examples

```
tmat <- trans.illdeath()
data(ebmt1)
covs <- c("score","yrel")
msebmt <- msprep(time=c(NA,"rel","srv"),status=c(NA,"relstat","srvstat"),
data=ebmt1,id="patid",keep=covs,trans=tmat)
# define a function (this one returns vector of regression coef's)
regcoefvec <- function(data) {
  cx <- coxph(Surv(Tstart,Tstop,status)~score+strata(trans),
          data=data,method="breslow")
  return(coef(cx))
}
regcoefvec(msebmt)
set.seed(1234)
msboot(theta=regcoefvec,data=msebmt,id="patid")
```

---

msdata2etm *msdata to etm format*

Description

msdata to etm format

Usage

```
msdata2etm(msdata, id, covs)
```

Arguments

msdata	Multi-state data in msdata format, as used in mstate
id	Column name identifying the subject id
covs	Vector of column names containing covariates to be included

---

msfit *Compute subject-specific transition hazards with (co-)variances*

Description

This function computes subject-specific or overall cumulative transition hazards for each of the possible transitions in the multi-state model. If requested, also the variances and covariances of the estimated cumulative transition hazards are calculated.

Usage

```
msfit(
  object,
  newdata,
  variance = TRUE,
  vartype = c("aalen", "greenwood"),
  trans
)
```

Arguments

object	A coxph object describing the fit of the multi-state model
newdata	A data frame with the same variable names as those that appear in the coxph formula. Its use is somewhat different from survfit. See Details. The argument newdata may be omitted only if the right hand side of the formula in the coxph object is ~strata(trans)
variance	A logical value indicating whether the (co-)variances of the subject-specific transition hazards should be computed. Default is TRUE
vartype	A character string specifying the type of variances to be computed (so only needed if variance=TRUE). Possible values are "aalen" or "greenwood"
trans	Transition matrix describing the states and transitions in the multi-state model. See trans in msprep for more detailed information

Details

The data frame needs to have one row for each transition in the multi-state model. An additional column strata (numeric) is needed to describe for each transition to which stratum it belongs. The name has to be strata, even if in the original coxph call another variable was used. For details refer to de Wreede, Fiocco & Putter (2010). So far, the results have been checked only for the "breslow" method of dealing with ties in coxph, so this is recommended.

Value

An object of class "msfit", which is a list containing

Haz	A data frame with time, Haz, trans, containing the estimated subject-specific hazards for each of the transitions in the multi-state model
varHaz	A data frame with time, Haz, trans1, trans2 containing the variances (trans1=trans2) and covariances (trans1<trans2) of the estimated hazards. This element is only returned when variance=TRUE
trans	The transition matrix used

Author(s)

Hein Putter <H.Putter@lumc.nl>

References

Putter H, Fiocco M, Geskus RB (2007). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* **26**, 2389–2430.

Therneau TM, Grambsch PM (2000). *Modeling Survival Data: Extending the Cox Model*. Springer, New York.

de Wreede LC, Fiocco M, and Putter H (2010). The mstate package for estimation and prediction in non- and semi-parametric multi-state and competing risks models. *Computer Methods and Programs in Biomedicine* **99**, 261–274.

de Wreede LC, Fiocco M, and Putter H (2011). mstate: An R Package for the Analysis of Competing Risks and Multi-State Models. *Journal of Statistical Software*, Volume 38, Issue 7.

See Also

plot.msfit

Examples

```
# transition matrix for illness-death model
tmat <- trans.illdeath()
# data in wide format, for transition 1 this is dataset E1 of
# Therneau & Grambsch (2000)
tg <- data.frame(illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
        dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
        x1=c(1,1,1,0,0,0),x2=c(6:1))
# data in long format using msprep
tglong <- msprep(time=c(NA,"illt","dt"),status=c(NA,"ills","ds"),
data=tg,keep=c("x1","x2"),trans=tmat)
# events
events(tglong)
table(tglong$status,tglong$to,tglong$from)
# expanded covariates
tglong <- expand.covs(tglong,c("x1","x2"))
# Cox model with different covariate
```

```
cx <- coxph(Surv(Tstart,Tstop,status)~x1.1+x2.2+strata(trans),
data=tglong,method="breslow")
summary(cx)
# new data, to check whether results are the same for transition 1 as
# those in appendix E.1 of Therneau & Grambsch (2000)
newdata <- data.frame(trans=1:3,x1.1=c(0,0,0),x2.2=c(0,1,0),strata=1:3)
msfit(cx,newdata,trans=tmat)
```

`msprep` *Function to prepare dataset for multi-state modeling in long format from dataset in wide format*

Description

This function converts a dataset which is in wide format (one subject per line, multiple columns indicating time and status for different states) into a dataset in long format (one line for each transition for which a subject is at risk). Selected covariates are replicated per subjects.

Usage

```
msprep(time, status, data, trans, start, id, keep)
```

Arguments

`time`	Either 1) a matrix or data frame of dimension n x S (n being the number of individuals and S the number of states in the multi-state model), containing the times at which the states are visited or last follow-up time, or 2) a character vector of length S containing the column names indicating these times. In the latter cases, some elements of `time` may be NA, see Details
`status`	Either 1) a matrix or data frame of dimension n x S, containing, for each of the states, event indicators taking the value 1 if the state is visited or 0 if it is not (censored), or 2) a character vector of length S containing the column names indicating these status variables. In the latter cases, some elements of `status` may be NA, see Details
`data`	Data frame (not a tibble) in wide format in which to interpret `time`, `status`, `id` or `keep`, if appropriate
`trans`	Transition matrix describing the states and transitions in the multi-state model. If S is the number of states in the multi-state model, `trans` should be an S x S matrix, with (i,j)-element a positive integer if a transition from i to j is possible in the multi-state model, `NA` otherwise. In particular, all diagonal elements should be `NA`. The integers indicating the possible transitions in the multi-state model should be sequentially numbered, 1,...,K, with K the number of transitions
`start`	List with elements `state` and `time`, containing starting states and times of the subjects in the data. Default is `NULL`, in which case all subjects start in state 1 at time 0. If a single state and time are given this state and time is used for all subjects, otherwise the length of `state` and `time` should equal the number of subjects in `data`

id	Either 1) a vector of length n containing the subject identifications, or 2) a character string indicating the column name containing these subject ids. If not provided, "id" will be assigned with values 1,...,n
keep	Either 1) a data frame or matrix with n rows or a numeric or factor vector of length n containing covariate(s) that need to be retained in the output dataset, or 2) a character vector containing the column names of these covariates in data

Details

For msprep, the transition matrix should correspond to an irreversible acyclic Markov chain. In particular, on the diagonals only NAs are allowed.

The transition matrix, if irreversible and acyclic, will have starting states, i.e. states into which no transitions are possible. For these starting states NAs are allowed in the time and status arguments, either as columns, when specified as matrix or data frame, or as elements of the character vector when specified as character vector.

The function msprep uses a recursive algorithm through calls to the recursive function msprepEngine. First, with the current transition matrix, all starting states are detected (defined as states into which there are no transitions). For each of these starting states, all subjects starting from that state are selected and for each subject the next visited state is detected by looking at all transitions from that starting state and determining the smallest transition time with status=1. The recursive msprepEngine is called again with the starting states deleted from the transition matrix and with subjects deleted that either reached an absorbing state or that were censored. For the remaining subjects the starting states and times are updated in the next call. Datasets returned from the msprepEngine calls are appended to the current dataset in long format and finally sorted.

A warning is issued for a subject, if multiple transitions exist with the same smallest transition time (and status=0). In such cases the next transition cannot be determined unambiguously, and the state with the smallest number is chosen. In our experience, occasionally the shortest transition time has status=0, while a higher transition time has status=1. Then this larger transition time and the corresponding transition is selected. No warning is issued for these data inconsistencies.

Value

An object of class "msdata", which is a data frame in long (counting process) format containing the subject id, the covariates (replicated per subject), and

from	the starting state
to	the receiving state
trans	the transition number
Tstart	the starting time of the transition
Tstop	the stopping time of the transition
status	status variable, with 1 indicating an event (transition), 0 a censoring

The "msdata" object has the transition matrix as "trans" attribute.

Author(s)

Hein Putter <H.Putter@lumc.nl> and Marta Fiocco

References

Putter H, Fiocco M, Geskus RB (2007). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* **26**, 2389–2430.

Examples

```
# transition matrix for illness-death model
tmat <- trans.illdeath()
# some data in wide format
tg <- data.frame(stt=rep(0,6),sts=rep(0,6),
        illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
        dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
        x1=c(1,1,1,2,2,2),x2=c(6:1))
tg$x1 <- factor(tg$x1,labels=c("male","female"))
tg$patid <- factor(2:7,levels=1:8,labels=as.character(1:8))
# define time, status and covariates also as matrices
tt <- matrix(c(rep(NA,6),tg$illt,tg$dt),6,3)
st <- matrix(c(rep(NA,6),tg$ills,tg$ds),6,3)
keepmat <- data.frame(gender=tg$x1,age=tg$x2)
# data in long format using msprep
msprep(time=tt,status=st,trans=tmat,keep=as.matrix(keepmat))
msprep(time=c(NA,"illt","dt"),status=c(NA,"ills","ds"),data=tg,
id="patid",keep=c("x1","x2"),trans=tmat)
# Patient no 5, 6 now start in state 2 at time t=4 and t=10
msprep(time=tt,status=st,trans=tmat,keep=keepmat,
        start=list(state=c(1,1,1,1,2,2),time=c(0,0,0,0,4,10)))
```

`mssample` *Sample paths through a multi-state model*

Description

Given cumulative transition hazards sample paths through the multi-state model.

Usage

```
mssample(
  Haz,
  trans,
  history = list(state = 1, time = 0, tstate = NULL),
  beta.state = NULL,
  clock = c("forward", "reset"),
  output = c("state", "path", "data"),
  tvec,
  cens = NULL,
  M = 10,
  do.trace = NULL
)
```

Arguments

Haz	Cumulative hazards to be sampled from. These should be given as a data frame with columns time, Haz, trans, for instance as the Haz list element given by msfit.
trans	Transition matrix describing the multi-state model. See trans in msprep for more detailed information
history	A list with elements state, specifying the starting state(s), time, the starting time(s), and tstate, a numeric vector of length the number of states, specifying at what times states have been visited, if appropriate. The default of tstate is NULL; more information can be found under Details. The elements state and time may either be scalars or vectors, in which case different sampled paths may start from different states or at different times. By default, all sampled paths start from state 1 at time 0.
beta.state	A matrix of dimension (no states) x (no transitions) specifying estimated effects of times at which earlier states were reached on subsequent transitions. If these are not in the model, the value NULL (default) suffices; more information can be found under Details
clock	Character argument, either "forward" (default) or "reset", specifying whether the time-scale of the cumulative hazards is in forward time ("forward") or duration in the present state ("reset")
output	One of "state", "path", or "data", specifying whether states, paths, or data should be output.
tvec	A numeric vector of time points at which the states or paths should be evaluated. Ignored if output="data"
cens	An independent censoring distribution, given as a data frame with time and Haz
M	The number of sampled trajectories through the multi-state model. The default is 10, since the procedure can become quite time-consuming
do.trace	An integer, specifying that the replication number should be written to the console every do.trace replications. Default is NULL in which case no output is written to the console during the simulation

Details

The procedure is described in detail in Fiocco, Putter & van Houwelingen (2008). The argument beta.state and the element tstate from the argument history are meant to incorporate situations where the time at which some previous states were visited may affect future transition rates. The relation between time of visit of state s and transition k is assumed to be linear on the log-hazards; the corresponding regression coefficient is to be supplied as the (s,k)-element of beta.state, which is 0 if no such effect has been included in the model. If no such effects are present, then beta.state=NULL (default) suffices. In the tstate element of history, the s-th element is the time at which state s was visited. This is only relevant for states which have been visited prior to the beginning of sampling, i.e. before the time element of history; the elements of tstate are internally updated when in the sampling process new states are visited (only if beta.state is not NULL to avoid unnecessary computations).

Value

M simulated paths through the multi-state model given by `trans` and `Haz`. It is either a data frame with columns `time`, `pstate1`, ..., `pstateS` for S states when `output="state"`, or with columns `time`, `ppath1`,..., `ppathP` for the P paths specified in `paths`(trans) when `output="path"`. When `output="data"`, the sampled paths are stored in an `"msdata"` object, a data frame in long format such as that obtained by `msprep`. This may be useful for (semi-)parametric bootstrap procedures, in which case `cens` may be used as censoring distribution (assumed to be independent of all transition times and independent of any covariates).

Author(s)

Marta Fiocco, Hein Putter <H.Putter@lumc.nl>

References

Fiocco M, Putter H, van Houwelingen HC (2008). Reduced-rank proportional hazards regression and simulation-based prediction for multi-state models. *Statistics in Medicine* **27**, 4340–4358.

Examples

```
# transition matrix for illness-death model
tmat <- trans.illdeath()
# data in wide format, for transition 1 this is dataset E1 of
# Therneau & Grambsch (T&G)
tg <- data.frame(illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
        dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
        x1=c(1,1,1,0,0,0),x2=c(6:1))
# data in long format using msprep
tglong <- msprep(time=c(NA,"illt","dt"),status=c(NA,"ills","ds"),
        data=tg,keep=c("x1","x2"),trans=tmat)
# expanded covariates
tglong <- expand.covs(tglong,c("x1","x2"))
# Cox model with different covariate
cx <- coxph(Surv(Tstart,Tstop,status)~x1.1+x2.2+strata(trans),
        data=tglong,method="breslow")
# new data, to check whether results are the same for transition 1 as T&G
newdata <- data.frame(trans=1:3,x1.1=c(0,0,0),x2.2=c(0,1,0),strata=1:3)
fit <- msfit(cx,newdata,trans=tmat)
tv <- unique(fit$Haz$time)
# mssample
set.seed(1234)
mssample(Haz=fit$Haz,trans=tmat,tvec=tv,M=100)
set.seed(1234)
paths(tmat)
mssample(Haz=fit$Haz,trans=tmat,tvec=tv,M=100,output="path")
set.seed(1234)
mssample(Haz=fit$Haz,trans=tmat,tvec=tv,M=100,output="data",do.trace=25)
```

paths *Find all possible trajectories through a given multi-state model*

Description

For a given multi-state model, specified through a transition matrix, paths recursively finds all the possible trajectories or paths through that multi-state starting from a specified state. DO NOT USE for reversible or cyclic multi-state models.

Usage

```
paths(trans, start = 1)
```

Arguments

trans	The transition matrix describing the multi-state model, see msprep
start	The starting state for the trajectories

Details

The function is recursive. It starts in start, looks at what states can be visited from start, and appends the results of the next call to the current value (matrix). If the transition matrix contains loops, the function will find infinitely many paths, so do not use paths for reversible or cyclic multi-state models. A warning is not yet incorporated!

Value

A matrix, each row of which specifies a possible path through the multi-state model.

Author(s)

Hein Putter <H.Putter@lumc.nl>

Examples

```
tmat <- matrix(NA,5,5)
tmat[1,2:3] <- 1:2
tmat[1,5] <- 3
tmat[2,4:5] <- 4:5
tmat[3,4:5] <- 6:7
tmat[4,5] <- 8
paths(tmat)
paths(tmat, start=3)
```

`plot.Cuminc` *Plot method for Cuminc objects*

Description

Plot the estimates of the non-parametric Aalen-Johansen estimate of the cumulative incidence functions (competing risks data). Note this is a method for `mstate::Cuminc` and not `cmprsk::cuminc`. Both return the same estimates, though the former does so in a dataframe, and the latter in the list.

Usage

```
## S3 method for class 'Cuminc'
plot(
  x,
  use.ggplot = FALSE,
  xlab = "Time",
  ylab = "Probability",
  xlim,
  ylim,
  lty,
  legend,
  cols,
  conf.type = c("log", "plain", "none"),
  conf.int = 0.95,
  legend.pos = "right",
  facet = FALSE,
  ...
)
```

Arguments

`x`	Object of class `"Cuminc"` to be printed or plotted
`use.ggplot`	Default FALSE, set TRUE for ggplot version of plot
`xlab`	A title for the x-axis; default is `"Time"`
`ylab`	A title for the y-axis; default is `"Probability"`
`xlim`	The x limits of the plot(s), default is range of time
`ylim`	The y limits of the plot(s); if ylim is specified for type="separate", then all plots use the same ylim for y limits
`lty`	The line type, see `par`; default is 1
`legend`	Character vector corresponding to number of absorbing states. In case of a grouped `"Cuminc"` object, with facet = FALSE the length of the vector is number absorbing states * group levels. Only relevant when use.ggplot = TRUE
`cols`	Vector (numeric or character) specifying colours of the lines
`conf.type`	Type of confidence interval - either "log" or "plain" . See function details for details.

conf.int	Confidence level (%) from 0-1 for probabilities, default is 0.95 (95% CI). Setting to 0 removes the CIs.
legend.pos	The position of the legend, see legend; default is "topleft"
facet	Logical, in case of group used for "Cuminc", facet by it - only relevant when use.ggplot = TRUE
...	Further arguments to plot or print method

Details

Grouped cumulative incidences can be plotted either in the same plot or in facets, see the facet argument.

Value

A ggplot object if use.ggplot = T used, otherwise NULL.

Author(s)

Edouard F. Bonneville <e.f.bonneville@lumc.nl>

Examples

```
library(ggplot2)

data("aidssi")
head(aidssi)
si <- aidssi

# No grouping
cum_incid <- Cuminc(
time = "time",
status = "status",
data = si
)

plot(
x = cum_incid,
use.ggplot = TRUE,
conf.type = "none",
lty = 1:2,
conf.int = 0.95
)

# With grouping
cum_incid_grp <- Cuminc(
time = "time",
status = "status",
group = "ccr5",
data = si
)
```

```
plot(
 x = cum_incid_grp,
 use.ggplot = TRUE,
 conf.type = "none",
 lty = 1:4,
 facet = TRUE
)
```

`plot.MarkovTest` *Plot method for a MarkovTest object*

Description

Plot method for an object of class 'MarkovTest'. It plots the trace of the log-rank statistics provided by `MarkovTest`.

Usage

```
## S3 method for class 'MarkovTest'
plot(
  x,
  y,
  what = c("states", "overall"),
  idx = NULL,
  quantiles = TRUE,
  qsup,
  states,
  xlab,
  ylab,
  main,
  ...
)
```

Arguments

`x`	Object of class 'MarkovTest'
`y`	The grid at which `MarkovTest` was calculated
`what`	Choose "states" for plotting state-specific traces, and "overall" for the overall chi-squared trace
`idx`	Vector of indices of wild bootstrap traces to plot
`quantiles`	Boolean whether or not to plot the 2.5 and 97.5 percent quantiles, default is `TRUE`
`qsup`	The index of the function in either `fn` (when plotting state-specific) or `fn2` (when plotting overall) to plot along with the traces; when missing this line is not included
`states`	Number of the qualifying state(s) to plot trace for

xlab	Text for x-axis label
ylab	Text for y-axis label
main	Text for title (main)
...	Further arguments to plot

Value

No return value

Author(s)

Hein Putter <H.Putter@lumc.nl>

See Also

MarkovTest

Examples

```
## Not run:
# Example provided by the prothrombin data
data("prothr")
# Apply Markov test to grid of monthly time points over the first 7.5 years
year <- 365.25
month <- year / 12
grid <- month * (1:90)
# Markov test for transition 1 (wild bootstrap based on 100 replications)
MT <- MarkovTest(prothr, id = "id", transition = 1,
                 grid = grid, B = 100)

plot(MT, grid, what="states", idx=1:50, states=rownames(attr(prothr, "trans")),
     xlab="Days since randomisation", ylab="Log-rank test statistic",
     main="Transition Normal -> Low")

plot(MT, grid,what="overall", idx=1:50,
     xlab="Days since randomisation", ylab="Chi-square test statistic",
     main="Transition Normal -> Low")

plot(MT, grid, what="states", quantiles=FALSE) # only trace
plot(MT, grid, what="states") # trace plus quantiles (default)
plot(MT, grid, what="states", idx=1:10) # trace plus quantiles, plus first 10 bootstrap traces

plot(MT, grid, what="overall", quantiles=FALSE) # only trace
plot(MT, grid, what="overall") # trace plus quantiles (default)
plot(MT, grid, what="overall", idx=1:10) # trace plus quantiles, plus first 10 bootstrap traces


## End(Not run)
```

plot.msfit *Plot method for an msfit object*

Description

Plot method for an object of class "msfit". It plots the estimated cumulative transition intensities in the multi-state model.

Usage

```
## S3 method for class 'msfit'
plot(
  x,
  type = c("single", "separate"),
  cols,
  xlab = "Time",
  ylab = "Cumulative hazard",
  ylim,
  lwd,
  lty,
  legend,
  legend.pos = "right",
  bty = "n",
  use.ggplot = FALSE,
  xlim,
  scale_type = "fixed",
  ...
)
```

Arguments

x	Object of class "msfit", containing estimated cumulative transition intensities for all transitions in a multi-state model
type	One of "single" (default) or "separate"; in case of "single", all estimated cumulative hazards are drawn in a single plot, in case of "separate", separate plots are shown for the estimated transition intensities
cols	A vector specifying colors for the different transitions; default is 1:K (K no of transitions), when type="single", and 1 (black), when type="separate"
xlab	A title for the x-axis; default is "Time"
ylab	A title for the y-axis; default is "Cumulative hazard"
ylim	The y limits of the plot(s); if ylim is specified for type="separate", then all plots use the same ylim for y limits
lwd	The line width, see par; default is 1
lty	The line type, see par; default is 1

legend	Character vector of length equal to the number of transitions, to be used in a legend; if missing, these will be taken from the row- and column-names of the transition matrix contained in x$trans. Also used as titles of plots for type="separate"
legend.pos	The position of the legend, see legend; default is "topleft"
bty	The box type of the legend, see legend
use.ggplot	Default FALSE, set TRUE for ggplot version of plot
xlim	Limits of x axis, relevant if use_ggplot = T
scale_type	"fixed", "free", "free_x" or "free_y", see scales argument of facet_wrap(). Only relevant for use_ggplot = T.
...	Further arguments to plot

Value

No return value

Author(s)

Hein Putter <H.Putter@lumc.nl>

Edouard F. Bonneville <e.f.bonneville@lumc.nl>

See Also

msfit

Examples

```
# transition matrix for illness-death model
tmat <- trans.illdeath()
# data in wide format, for transition 1 this is dataset E1 of
# Therneau & Grambsch (2000)
tg <- data.frame(illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
        dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
        x1=c(1,1,1,0,0,0),x2=c(6:1))
# data in long format using msprep
tglong <- msprep(time=c(NA,"illt","dt"),status=c(NA,"ills","ds"),
data=tg,keep=c("x1","x2"),trans=tmat)
# events
events(tglong)
table(tglong$status,tglong$to,tglong$from)
# expanded covariates
tglong <- expand.covs(tglong,c("x1","x2"))
# Cox model with different covariate
cx <- coxph(Surv(Tstart,Tstop,status)~x1.1+x2.2+strata(trans),
data=tglong,method="breslow")
summary(cx)
# new data, to check whether results are the same for transition 1 as
# those in appendix E.1 of Therneau & Grambsch (2000)
```

```
newdata <- data.frame(trans=1:3,x1.1=c(0,0,0),x2.2=c(0,1,0),strata=1:3)
msf <- msfit(cx,newdata,trans=tmat)
# standard plot
plot(msf)
# specifying line width, color, and legend
plot(msf,lwd=2,col=c("darkgreen","darkblue","darkred"),legend=c("1->2","1->3","2->3"))
# separate plots
par(mfrow=c(2,2))
plot(msf,type="separate",lwd=2)
par(mfrow=c(1,1))

# ggplot version - see vignette for details
library(ggplot2)
plot(msf, use.ggplot = TRUE)
```

`plot.probtrans` *Plot method for a probtrans object*

Description

Plot method for an object of class 'probtrans'. It plots the transition probabilities as estimated by probtrans.

Usage

```
## S3 method for class 'probtrans'
plot(
  x,
  from = 1,
  type = c("filled", "single", "separate", "stacked"),
  ord,
  cols,
  xlab = "Time",
  ylab = "Probability",
  xlim,
  ylim,
  lwd,
  lty,
  cex,
  legend,
  legend.pos = "right",
  bty = "n",
  xaxs = "i",
  yaxs = "i",
  use.ggplot = FALSE,
  conf.int = 0.95,
  conf.type = c("log", "plain", "none"),
```

```
  label,
  ...
)
```

Arguments

x	Object of class 'probtrans', containing estimated transition probabilities
from	The starting state from which the probabilities are used to plot
type	One of "stacked" (default), "filled", "single" or "separate"; in case of "stacked", the transition probabilities are stacked and the distance between two adjacent curves indicates the probability, this is also true for "filled", but the space between adjacent curves are filled, in case of "single", the probabilities are shown as different curves in a single plot, in case of "separate", separate plots are shown for the estimated transition probabilities
ord	A vector of length equal to the number of states, specifying the order of plotting in case type="stacked" or "filled"
cols	A vector specifying colors for the different transitions; default is a palette from green to red, when type="filled" (reordered according to ord, and 1 (black), otherwise
xlab	A title for the x-axis; default is "Time"
ylab	A title for the y-axis; default is "Probability"
xlim	The x limits of the plot(s), default is range of time
ylim	The y limits of the plot(s); if ylim is specified for type="separate", then all plots use the same ylim for y limits
lwd	The line width, see par; default is 1
lty	The line type, see par; default is 1
cex	Character size, used in text; only used when type="stacked" or "filled"
legend	Character vector of length equal to the number of transitions, to be used in a legend; if missing, numbers will be used; this and the legend arguments following are ignored when type="separate"
legend.pos	The position of the legend, see legend; default is "topleft"
bty	The box type of the legend, see legend
xaxs	See par, default is "i", for type="stacked"
yaxs	See par, default is "i", for type="stacked"
use.ggplot	Default FALSE, set TRUE for ggplot version of plot
conf.int	Confidence level (%) from 0-1 for probabilities, default is 0.95 (95% CI). Setting to 0 removes the CIs.
conf.type	Type of confidence interval - either "log" or "plain" . See function details for details.
label	Only relevant for type = "filled" or "stacked", set to "annotate" to have state labels on plot, or leave unspecified.
...	Further arguments to plot

Details

Regarding confidence intervals: let p denote a predicted probability, σ its estimated standard error, and $z_{\alpha/2}$ denote the critical value of the standard normal distribution at confidence level $1-\alpha$.

The confidence interval of type "plain" is then

$$p \pm z_{\alpha/2} * \sigma$$

The confidence interval of type "log", based on the Delta method, is then

$$\exp(\log(p) \pm z_{\alpha/2} * \sigma/p)$$

Value

No return value

Author(s)

Hein Putter <H.Putter@lumc.nl>

Edouard F. Bonneville <e.f.bonneville@lumc.nl>

See Also

probtrans

Examples

```
# transition matrix for illness-death model
tmat <- trans.illdeath()
# data in wide format, for transition 1 this is dataset E1 of
# Therneau and Grambsch (2000)
tg <- data.frame(illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
        dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
        x1=c(1,1,1,0,0,0),x2=c(6:1))
# data in long format using msprep
tglong <- msprep(time=c(NA,"illt","dt"),status=c(NA,"ills","ds"),
data=tg,keep=c("x1","x2"),trans=tmat)
# events
events(tglong)
table(tglong$status,tglong$to,tglong$from)
# expanded covariates
tglong <- expand.covs(tglong,c("x1","x2"))
# Cox model with different covariate
cx <- coxph(Surv(Tstart,Tstop,status)~x1.1+x2.2+strata(trans),
data=tglong,method="breslow")
summary(cx)
# new data, to check whether results are the same for transition 1 as
# those in appendix E.1 of Therneau and Grambsch (2000)
newdata <- data.frame(trans=1:3,x1.1=c(0,0,0),x2.2=c(0,1,0),strata=1:3)
msf <- msfit(cx,newdata,trans=tmat)
# probtrans
```

```
pt <- probtrans(msf,predt=0)
# default plot
plot(pt,ord=c(2,3,1),lwd=2,cex=0.75)
# filled plot
plot(pt,type="filled",ord=c(2,3,1),lwd=2,cex=0.75)
# single plot
plot(pt,type="single",lwd=2,col=rep(1,3),lty=1:3,legend.pos=c(8,1))
# separate plots
par(mfrow=c(2,2))
plot(pt,type="sep",lwd=2)
par(mfrow=c(1,1))

# ggplot version - see vignette for details
library(ggplot2)
plot(pt, ord=c(2,3,1), use.ggplot = TRUE)
```

`print.MarkovTest` *Print method for a MarkovTest object*

Description

Print method for an object of class 'MarkovTest'

Usage

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

Arguments

`x`	Object of class 'markovTest', as obtained by call to `MarkovTest`
`...`	Further arguments to print

Value

No return value

Author(s)

Hein Putter <H.Putter@lumc.nl>

See Also

`MarkovTest`

Examples

```
## Not run:
# Example provided by the prothrombin data
data("prothr")
# Apply Markov test to grid of monthly time points over the first 7.5 years
year <- 365.25
month <- year / 12
grid <- month * (1:90)
# Markov test for transition 1 (wild bootstrap based on 25 replications for brevity)
MT <- MarkovTest(prothr, id = "id", transition = 1,
                 grid = grid, B = 25)
MT

## End(Not run)
```

`print.msdata` *Print method for a msdata object*

Description

Print method for an object of class 'msdata'

Usage

```
## S3 method for class 'msdata'
print(x, trans = FALSE, ...)
```

Arguments

`x`	Object of class 'msdata', as prepared for instance by `msprep`
`trans`	Boolean specifying whether or not the transition matrix should be printed as well; default is `FALSE`
`...`	Further arguments to print

Value

No return value

Author(s)

Hein Putter <H.Putter@lumc.nl>

See Also

`probtrans`

Examples

```
# transition matrix for illness-death model
tmat <- trans.illdeath()
# some data in wide format
tg <- data.frame(stt=rep(0,6),sts=rep(0,6),
        illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
        dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
        x1=c(1,1,1,2,2,2),x2=c(6:1))
tg$x1 <- factor(tg$x1,labels=c("male","female"))
tg$patid <- factor(2:7,levels=1:8,labels=as.character(1:8))
# define time, status and covariates also as matrices
tt <- matrix(c(rep(NA,6),tg$illt,tg$dt),6,3)
st <- matrix(c(rep(NA,6),tg$ills,tg$ds),6,3)
keepmat <- data.frame(gender=tg$x1,age=tg$x2)
# data in long format using msprep
msp <- msprep(time=tt,status=st,trans=tmat,keep=as.matrix(keepmat))
print(msp)
print(msp, trans=TRUE)
```

---

print.summary.msfit *Print method for summary.msfit object*

Description

Print method for summary.msfit object

Usage

```
## S3 method for class 'summary.msfit'
print(x, complete = FALSE, ...)
```

Arguments

x	Object of class 'summary.msfit', to be printed
complete	Whether or not the complete estimated cumulative transition intensities should be printed (TRUE) or not (FALSE); default is FALSE, in which case the estimated cumulative transition hazards will be printed for the first and last 6 time points of each transition or of the selected times (or all when there are at most 12 of these time points
...	Further arguments to print

Examples

```
## Not run:
# If all time points should be printed, specify complete=TRUE in the print statement
print(x, complete=TRUE)

## End(Not run)
```

`print.summary.probtrans` *Print method for a summary.probtrans object*

Description

Print method for a summary.probtrans object

Usage

```
## S3 method for class 'summary.probtrans'
print(x, complete = FALSE, ...)
```

Arguments

`x`	Object of class 'summary.probtrans', to be printed
`complete`	Whether or not the complete estimated transition probabilities should be printed (`TRUE`) or not (`FALSE`); default is `FALSE`, in which case the estimated transition probilities will be printed for the first and last 6 time points of each starting state or of the selected times (or all when there are at most 12 of these time points
`...`	Further arguments to print

Examples

```
## Not run:
# If all time points should be printed, specify complete=TRUE in the print statement
print(x, complete=TRUE)

## End(Not run)
```

probtrans *Compute subject-specific or overall transition probabilities with standard errors*

Description

This function computes subject-specific or overall transition probabilities in multi-state models. If requested, also standard errors are calculated.

Usage

```
probtrans(
  object,
  predt,
  direction = c("forward", "fixedhorizon"),
  method = c("aalen", "greenwood"),
  variance = TRUE,
  covariance = FALSE
)
```

Arguments

object	msfit object containing estimated cumulative hazards for each of the transitions in the multi-state model and, if standard errors are requested, (co)variances of these cumulative hazards for each pair of transitions
predt	A positive number indicating the prediction time. This is either the time at which the prediction is made (if direction= "forward") or the time for which the prediction is to be made (if direction="fixedhorizon")
direction	One of "forward" (default) or "fixedhorizon", indicating whether prediction is forward or for a fixed horizon
method	A character string specifying the type of variances to be computed (so only needed if either variance or covariance is TRUE). Possible values are "aalen" or "greenwood"
variance	Logical value indicating whether standard errors are to be calculated (default is TRUE)
covariance	Logical value indicating whether covariances of transition probabilities for different states are to be calculated (default is FALSE)

Details

For details refer to de Wreede, Fiocco & Putter (2010).

Value

An object of class "probtrans", which is a list of which item [[s]] contains a data frame with the estimated transition probabilities (and standard errors if variance=TRUE) from state s. If covariance=TRUE, item varMatrix contains an array of dimension K^2 x K^2 x (nt+1) (with K the number of states and nt the distinct transition time points); the time points correspond to those in the data frames with the estimated transition probabilities. Finally, there are items trans, method, predt, direction, recording the transition matrix, and the method, predt and direction arguments used in the call to probtrans. Plot and summary methods have been defined for "probtrans" objects.

Author(s)

Liesbeth de Wreede and Hein Putter <H.Putter@lumc.nl>

References

Andersen PK, Borgan O, Gill RD, Keiding N (1993). *Statistical Models Based on Counting Processes*. Springer, New York.

Putter H, Fiocco M, Geskus RB (2007). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* **26**, 2389–2430.

Therneau TM, Grambsch PM (2000). *Modeling Survival Data: Extending the Cox Model*. Springer, New York.

de Wreede LC, Fiocco M, and Putter H (2010). The mstate package for estimation and prediction in non- and semi-parametric multi-state and competing risks models. *Computer Methods and Programs in Biomedicine* **99**, 261–274.

de Wreede LC, Fiocco M, and Putter H (2011). mstate: An R Package for the Analysis of Competing Risks and Multi-State Models. *Journal of Statistical Software*, Volume 38, Issue 7.

Examples

```
# transition matrix for illness-death model
tmat <- trans.illdeath()
# data in wide format, for transition 1 this is dataset E1 of
# Therneau & Grambsch (2000)
tg <- data.frame(illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
        dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
        x1=c(1,1,1,0,0,0),x2=c(6:1))
# data in long format using msprep
tglong <- msprep(time=c(NA,"illt","dt"),status=c(NA,"ills","ds"),
data=tg,keep=c("x1","x2"),trans=tmat)
# events
events(tglong)
table(tglong$status,tglong$to,tglong$from)
# expanded covariates
tglong <- expand.covs(tglong,c("x1","x2"))
# Cox model with different covariate
cx <- coxph(Surv(Tstart,Tstop,status)~x1.1+x2.2+strata(trans),
        data=tglong,method="breslow")
```

```
summary(cx)
# new data, to check whether results are the same for transition 1 as
# those in appendix E.1 of Therneau & Grambsch (2000)
newdata <- data.frame(trans=1:3,x1.1=c(0,0,0),x2.2=c(0,1,0),strata=1:3)
HvH <- msfit(cx,newdata,trans=tmat)
# probtrans
pt <- probtrans(HvH,predt=0)
# predictions from state 1
pt[[1]]
```

`redrank` *Reduced rank proportional hazards model for competing risks and multi-state models*

Description

This function estimates regression coefficients in reduced rank proportional hazards models for competing risks and multi-state models.

Usage

```
redrank(
  redrank,
  full = ~1,
  data,
  R,
  strata = NULL,
  Gamma.start,
  method = "breslow",
  eps = 1e-05,
  print.level = 1
)
```

Arguments

`redrank`	Survival formula, starting with either Surv(time,status) ~ or with Surv(Tstart,Tstop,status) ~, followed by a formula containing covariates for which a reduced rank estimate is to be found
`full`	Optional, formula specifying that part which needs to be retained in the model (so not subject to reduced rank)
`data`	Object of class 'msdata', as prepared for instance by `msprep`, in which to interpret the `redrank` and, optionally, the `full` formulas
`R`	Numeric, indicating the rank of the solution
`strata`	Name of covariate to be used inside the `strata` part of `coxph`
`Gamma.start`	A matrix of dimension K x R, with K the number of transitions and R the rank, to be used as starting value

method	The method for handling ties in coxph
eps	Numeric value determining when the iterative algorithm is finished (when for two subsequent iterations the difference in log-likelihood is smaller than eps)
print.level	Determines how much output is written to the screen; 0: no output, 1: iterations, for each iteration solutions of Alpha, Gamma, log-likelihood, 2: also the Cox regression results

Details

For details refer to Fiocco, Putter & van Houwelingen (2005, 2008).

Value

A list with elements

Alpha	the Alpha matrix
Gamma	the Gamma matrix
Beta	the Beta matrix corresponding to covariates
Beta2	the Beta matrix corresponding to fullcovs
cox.itr1	the coxph object resulting from the last call giving Alpha
alphaX	the matrix of prognostic scores given by Alpha, n x R, with n number of subjects
niter	the number of iterations needed to reach convergence
df	the number of regression parameters estimated
loglik	the log-likelihood

Author(s)

Marta Fiocco and Hein Putter <H.Putter@lumc.nl>

References

Fiocco M, Putter H, van Houwelingen JC (2005). Reduced rank proportional hazards model for competing risks. *Biostatistics* **6**, 465–478.

Fiocco M, Putter H, van Houwelingen HC (2008). Reduced-rank proportional hazards regression and simulation-based prediction for multi-state models. *Statistics in Medicine* **27**, 4340–4358.

Putter H, Fiocco M, Geskus RB (2007). Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* **26**, 2389–2430.

Examples

```
## Not run:
    # This reproduces the results in Fiocco, Putter & van Houwelingen (2005)
    # Takes a while to run
    data(ebmt2)
    # transition matrix for competing risks
   tmat <- trans.comprisk(6,names=c("Relapse","GvHD","Bacterial","Viral","Fungal","Other"))
```

```
    # preparing long dataset
    ebmt2$stat1 <- as.numeric(ebmt2$status==1)
    ebmt2$stat2 <- as.numeric(ebmt2$status==2)
    ebmt2$stat3 <- as.numeric(ebmt2$status==3)
    ebmt2$stat4 <- as.numeric(ebmt2$status==4)
    ebmt2$stat5 <- as.numeric(ebmt2$status==5)
    ebmt2$stat6 <- as.numeric(ebmt2$status==6)
    covs <- c("dissub","match","tcd","year","age")
    ebmtlong <- msprep(time=c(NA,rep("time",6)),
                       stat=c(NA,paste("stat",1:6,sep="")),
                       data=ebmt2,keep=covs,trans=tmat)

    # The reduced rank 2 solution
    rr2 <- redrank(Surv(Tstart,Tstop,status) ~ dissub+match+tcd+year+age,
                   data=ebmtlong, R=2)
    rr3$Alpha; rr3$Gamma; rr3$Beta; rr3$loglik
    # The reduced rank 3 solution
    rr3 <- redrank(Surv(Tstart,Tstop,status) ~ dissub+match+tcd+year+age,
                   data=ebmtlong, R=3)
    rr3$Alpha; rr3$Gamma; rr3$Beta; rr3$loglik
    # The reduced rank 3 solution, with no reduction on age
    rr3 <- redrank(Surv(Tstart,Tstop,status) ~ dissub+match+tcd+year, full=~age,
                   data=ebmtlong, R=3)
    rr3$Alpha; rr3$Gamma; rr3$Beta; rr3$loglik
    # The full rank solution
    fullrank <- redrank(Surv(Tstart,Tstop,status) ~ dissub+match+tcd+year+age,
                        data=ebmtlong, R=6)
    fullrank$Beta; fullrank$loglik

## End(Not run)
```

`summary.Cuminc` *Summary method for a summary.Cuminc object*

Description

Summary method for a summary.Cuminc object

Usage

```
## S3 method for class 'Cuminc'
summary(object, ...)
```

Arguments

`object`	Object of class 'Cuminc', to be summarised
`...`	Further arguments to summarise

summary.msfit *Summary method for an msfit object*

Description

Summary method for an object of class 'msfit'. It prints a selection of the estimated cumulative transition intensities, and, if requested, also of the (co)variances.

Usage

```
## S3 method for class 'msfit'
summary(
  object,
  times,
  transitions,
  variance = TRUE,
  conf.int = 0.95,
  conf.type = c("log", "none", "plain"),
  extend = FALSE,
  ...
)
```

Arguments

object	Object of class 'msfit', containing estimated cumulative transition intensities for all transitions in a multi-state model
times	Time points at which to evaluate the cumulative transition hazards
transitions	The transition for which to summarize the cumulative transition hazards
variance	Whether or not the standard errors of the estimated cumulative transition intensities should be printed; default is TRUE
conf.int	The proportion to be covered by the confidence intervals, default is 0.95
conf.type	The type of confidence interval, one of "log", "none", or "plain". Defaults to "log"
extend	logical value: if TRUE, prints information for all specified times, even if there are no subjects left at the end of the specified times. This is only valid if the times argument is present
...	Further arguments to summary

Value

Function summary.msfit returns an object of class "summary.msfit", which is a list (for each from state) of cumulative transition hazaards at the specified (or all) time points. The print method of a summary.probtrans doesn't return a value.

Author(s)

Hein Putter <H.Putter@lumc.nl>

See Also

msfit

Examples

```
# Start with example from msfit
tmat <- trans.illdeath()
tg <- data.frame(illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
                 dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
                 x1=c(1,1,1,0,0,0),x2=c(6:1))
tglong <- msprep(time=c(NA,"illt","dt"),status=c(NA,"ills","ds"),
                 data=tg,keep=c("x1","x2"),trans=tmat)
tglong <- expand.covs(tglong,c("x1","x2"))
cx <- coxph(Surv(Tstart,Tstop,status)~x1.1+x2.2+strata(trans),
            data=tglong,method="breslow")
newdata <- data.frame(trans=1:3,x1.1=c(0,0,0),x2.2=c(0,1,0),strata=1:3)
msf <- msfit(cx,newdata,trans=tmat)

# Default, all transitions, with SE
summary(msf)
summary(msf, conf.type="plain")
# Only transitions 1 and 3
summary(msf, tra=c(1, 3))
# Default is 95% confidence interval, change here to 90%
summary(msf, conf.int=0.90)
# Do not show variances (nor confidence intervals)
summary(msf, variance=FALSE)
# Cumulative hazards only at specified time points
summary(msf, times=seq(0, 15, by=3))
# Last specified time point is larger than last observed, not printed
# Use extend=TRUE as in summary.survfit
summary(msf, times=seq(0, 15, by=3), extend=TRUE)
# Different types of confidence intervals, default is log
summary(msf, times=seq(0, 15, by=3), conf.type="plain")
summary(msf, times=seq(0, 15, by=3), conf.type="no")
# When the number of time points specified is larger than 12, head and tail is shown
x <- summary(msf, times=seq(5, 8, by=0.25))
x
```

---

summary.probtrans *Summary method for a probtrans object*

Description

Summary method for an object of class 'probtrans'. It prints a selection of the estimated transition probabilities, and, if requested, also of the variances.

Usage

```
## S3 method for class 'probtrans'
summary(
  object,
  times,
  from = 1,
  to = 0,
  variance = TRUE,
  conf.int = 0.95,
  conf.type = c("log", "none", "plain"),
  extend = FALSE,
  ...
)
```

Arguments

object	Object of class 'probtrans', containing estimated transition probabilities from and to all states in a multi-state model
times	Time points at which to evaluate the transition probabilites
from	Specifies from which state the transition probabilities are to be printed. Should be subset of 1:S, with S the number of states in the multi-state model. Default is print from state 1 only. User can specify from=0 to print transition probabilities from all states
to	Specifies the transition probabilities to which state are to be printed. User can specify to=0 to print transition probabilities to all states. This is also the default
variance	Whether or not the standard errors of the estimated transition probabilities should be printed; default is TRUE
conf.int	The proportion to be covered by the confidence intervals, default is 0.95
conf.type	The type of confidence interval, one of "log", "none", or "plain". Defaults to "log"
extend	logical value: if TRUE, prints information for all specified times, even if there are no subjects left at the end of the specified times. This is only valid if the times argument is present
...	Further arguments to print

Value

Function summary.probtrans returns an object of class "summary.probtrans", which is a list (for each from state) of transition probabilities at the specified (or all) time points. The print method of a summary.probtrans doesn't return a value.

Author(s)

Hein Putter <H.Putter@lumc.nl>

See Also

probtrans

Examples

```
# First run the example of probtrans
tmat <- trans.illdeath()
tg <- data.frame(illt=c(1,1,6,6,8,9),ills=c(1,0,1,1,0,1),
                 dt=c(5,1,9,7,8,12),ds=c(1,1,1,1,1,1),
                 x1=c(1,1,1,0,0,0),x2=c(6:1))
tglong <- msprep(time=c(NA,"illt","dt"),status=c(NA,"ills","ds"),
                 data=tg,keep=c("x1","x2"),trans=tmat)
tglong <- expand.covs(tglong,c("x1","x2"))
cx <- coxph(Surv(Tstart,Tstop,status)~x1.1+x2.2+strata(trans),
            data=tglong,method="breslow")
newdata <- data.frame(trans=1:3,x1.1=c(0,0,0),x2.2=c(0,1,0),strata=1:3)
HvH <- msfit(cx,newdata,trans=tmat)
pt <- probtrans(HvH,predt=0)

# Default, prediction from state 1
summary(pt)
# Only from states 1 and 3
summary(pt, from=c(1, 3))
# Use from=0 for prediction from all states
summary(pt, from=0)
# Only to states 1 and 2
summary(pt, to=1:2)
# Default is 95% confidence interval, change here to 90%
summary(pt, to=1:2, conf.int=0.90)
# Do not show variances (nor confidence intervals)
summary(pt, to=1:2, variance=FALSE)
# Transition probabilities only at specified time points
summary(pt, times=seq(0, 15, by=3))
# Last specified time point is larger than last observed, not printed
# Use extend=TRUE as in summary.survfit
summary(pt, times=seq(0, 15, by=3), extend=TRUE)
# Different types of confidence intervals, default is log
summary(pt, times=seq(0, 15, by=3), conf.type="plain")
summary(pt, times=seq(0, 15, by=3), conf.type="no")
# When the number of time points specified is larger than 12, head and tail is shown
x <- summary(pt, times=seq(5, 8, by=0.25))
x
```

trans2tra *Convert transition matrix from mstate to etm format*

Description

Convert transition matrix from mstate to etm format

Usage

```
trans2tra(trans)
```

Arguments

trans	Transition matrix in mstate format

transhelp *Help functions for transition matrix*

Description

Help functions to get insight into the structure of a transition matrix.

Arguments

trans	Transition matrix, for instance produced by transMat), trans.comprisk, or trans.illdeath

Details

Function to.trans2 simply lists the transitions in trans in a data frame; function trans2Q converts trans to a Q matrix, the (j,k)th element of which contains the (shortest) number of transitions needed to travel from the jth to the kth state; function absorbing returns a vector (named if trans contains row or columnc names) with the state numbers that are absorbing; function is.circular returns (a Boolean) whether the transition matrix specified in trans is circular or not.

Value

See details.

Author(s)

Hein Putter <H.Putter@lumc.nl>

Examples

```
# Irreversible illness-death model
tmat <- trans.illdeath(c("Healthy", "Illness", "Death"))
tmat
to.trans2(tmat)
trans2Q(tmat)
absorbing(tmat)
is.circular(tmat)
# Reversible illness-death model
tmat <- transMat(x = list( c(2, 3), c(1, 3), c() ),
                 names = c("Healthy", "Illness", "Death"))
tmat
to.trans2(tmat)
trans2Q(tmat)
absorbing(tmat)
is.circular(tmat)
```

`transMat` *Define transition matrix for multi-state model*

Description

Define transition matrices for multi-state model. Specific functions for defining such transition matrices are pre-defined for common multi-state models like the competing risks model and the illness-death model.

Usage

```
transMat(x, names)
```

Arguments

`x`	List of possible transitions; x[[i]] consists of a vector of state numbers reachable from state i
`names`	A character vector containing the names of either the competing risks or the states in the multi-state model specified by the competing risks or illness-death model. `names` should have the same length as the list `x` (for `transMat`), or either `K` or `K+1` (for `trans.comprisk`), or 3 (for `trans.illdeath`)

Details

If `names` is missing, the names `"eventfree"`, `"cause1"`, etcetera are assigned in `trans.comprisk`, or `"healthy"`, `"illness"`, `"death"` in `trans.illdeath`.

Value

A transition matrix describing the states and transitions in the multi-state model.

Author(s)

Steven McKinney <smckinney@bccrc.ca>; Hein Putter <H.Putter@lumc.nl>

Examples

```
transMat(list(c(2, 3), c(), c(1, 2)),
names = c("Disease-free", "Death", "Relapsed"))
tmat <- transMat(x = list( c(2, 3), c(1, 3), c() ),
                 names = c("Normal", "Low", "Death"))
tmat
transListn <- list("Normal" = c(2, 3), "Low" = c(1, 3), "Death" = c())
transMat(transListn)
trans.comprisk(3)
trans.comprisk(3,c("c1","c2","c3"))
trans.comprisk(3,c("nothing","c1","c2","c3"))
trans.illdeath()
trans.illdeath(c("nothing","ill","death"))
```

---

vis.mirror.pt *Mirror plot comparing two probtrans objects*

Description

A mirror plot for comparing two different "probtrans" objects. Useful for comparing predicted probabilities for different levels of a covariate, or for different subgroups.

Usage

```
vis.mirror.pt(
  x,
  titles,
  size_titles = 5,
  horizon = NULL,
  breaks_x_left,
  breaks_x_right,
  from = 1,
  cols,
  ord,
  xlab = "Time",
  ylab = "Probability",
  legend.pos = "right"
)
```

Arguments

x	A list of two "probtrans" objects. The first element will be on the left of the mirror plot, and the second on the right
titles	A character vector c("Title for left", "Title for right")
size_titles	Numeric, size of the title text
horizon	Numeric, position denoting (in time) where to symmetrically mirror the plots. Default is maximum follow-up of from both plots.
breaks_x_left	Numeric vector specifying axis breaks on the left plot
breaks_x_right	Numeric vector specifying axis breaks on the right plot
from	The starting state from which the probabilities are used to plot
cols	A vector specifying colors for the different transitions; default is a palette from green to red, when type="filled" (reordered according to ord, and 1 (black), otherwise
ord	A vector of length equal to the number of states, specifying the order of plotting in case type="stacked" or "filled"
xlab	A title for the x-axis, default is "Time"
ylab	A title for the y-axis, default is "Probability"
legend.pos	Position of the legend, default is "right"

Value

A ggplot2 object.

Author(s)

Edouard F. Bonneville <e.f.bonneville@lumc.nl>

See Also

plot.probtrans

Examples

```
library(ggplot2)

data("aidssi")
head(aidssi)
si <- aidssi

# Prepare transition matrix
tmat <- trans.comprisk(2, names = c("event-free", "AIDS", "SI"))

# Run msprep
si$stat1 <- as.numeric(si$status == 1)
si$stat2 <- as.numeric(si$status == 2)
```

```
silong <- msprep(
time = c(NA, "time", "time"),
status = c(NA, "stat1", "stat2"),
data = si, keep = "ccr5", trans = tmat
)

# Run cox model
silong <- expand.covs(silong, "ccr5")
c1 <- coxph(Surv(time, status) ~ ccr5WM.1 + ccr5WM.2 + strata(trans),
            data = silong)

# 1. Prepare reference patient data - both CCR5 genotypes
WW <- data.frame(
ccr5WM.1 = c(0, 0),
ccr5WM.2 = c(0, 0),
trans = c(1, 2),
strata = c(1, 2)
)

WM <- data.frame(
ccr5WM.1 = c(1, 0),
ccr5WM.2 = c(0, 1),
trans = c(1, 2),
strata = c(1, 2)
)

# 2. Make msfit objects
msf.WW <- msfit(c1, WW, trans = tmat)
msf.WM <- msfit(c1, WM, trans = tmat)

# 3. Make probtrans objects
pt.WW <- probtrans(msf.WW, predt = 0)
pt.WM <- probtrans(msf.WM, predt = 0)

# Mirror plot split at 10 years - see vignette for more details
vis.mirror.pt(
x = list(pt.WW, pt.WM),
titles = c("WW", "WM"),
horizon = 10
)
```

`vis.multiple.pt` *Visualise multiple probtrans objects*

Description

Helper function allow to visualise state probabilities for different reference patients/covariates. Multiple `"probtrans"` objects are thus needed.

Usage

```
vis.multiple.pt(
  x,
  from = 1,
  to,
  xlab = "Time",
  ylab,
  xlim = NULL,
  ylim = NULL,
  cols,
  lwd,
  labels,
  conf.int = 0.95,
  conf.type = c("log", "plain", "none"),
  legend.title
)
```

Arguments

x	A list of "probtrans" objects
from	The starting state from which the probabilities are used to plot Numeric, as in plot.probtrans
to	(Numeric) destination state
xlab	A title for the x-axis; default is "Time"
ylab	A title for the y-axis; default is "Probability"
xlim	The x limits of the plot(s), default is range of time
ylim	The y limits of the plot(s); if ylim is specified for type="separate", then all plots use the same ylim for y limits
cols	A vector specifying colors for the different transitions; default is a palette from green to red, when type="filled" (reordered according to ord, and 1 (black), otherwise
lwd	The line width, see par; default is 1
labels	Character vector labelling each element of x (e.g. label for a reference patient) - so labels = c("Patient 1", "Patient 2")
conf.int	Confidence level (%) from 0-1 for probabilities, default is 0.95 (95% CI). Setting to 0 removes the CIs.
conf.type	Type of confidence interval - either "log" or "plain" . See function details for details.
legend.title	Character - title of legend

Value

A ggplot object.

Author(s)

Edouard F. Bonneville <e.f.bonneville@lumc.nl>

Examples

```
library(ggplot2)

data("aidssi")
head(aidssi)
si <- aidssi

# Prepare transition matrix
tmat <- trans.comprisk(2, names = c("event-free", "AIDS", "SI"))

# Run msprep
si$stat1 <- as.numeric(si$status == 1)
si$stat2 <- as.numeric(si$status == 2)

silong <- msprep(
time = c(NA, "time", "time"),
status = c(NA, "stat1", "stat2"),
data = si, keep = "ccr5", trans = tmat
)

# Run cox model
silong <- expand.covs(silong, "ccr5")
c1 <- coxph(Surv(time, status) ~ ccr5WM.1 + ccr5WM.2 + strata(trans),
            data = silong)

# 1. Prepare patient data - both CCR5 genotypes
WW <- data.frame(
ccr5WM.1 = c(0, 0),
ccr5WM.2 = c(0, 0),
trans = c(1, 2),
strata = c(1, 2)
)

WM <- data.frame(
ccr5WM.1 = c(1, 0),
ccr5WM.2 = c(0, 1),
trans = c(1, 2),
strata = c(1, 2)
)

# 2. Make msfit objects
msf.WW <- msfit(c1, WW, trans = tmat)
msf.WM <- msfit(c1, WM, trans = tmat)

# 3. Make probtrans objects
pt.WW <- probtrans(msf.WW, predt = 0)
pt.WM <- probtrans(msf.WM, predt = 0)
```

```
# Plot - see vignette for more details
vis.multiple.pt(
x = list(pt.WW, pt.WM),
from = 1,
to = 2,
conf.type = "log",
cols = c(1, 2),
labels = c("Pat WW", "Pat WM"),
legend.title = "Ref patients"
)
```

`xsect` *Make a cross-section of multi-state data at a given time point*

Description

Given a dataset in long format, for instance generated by `msprep`, this function takes a cross-section at a given time point, to list the subjects under observation (at risk) at that time point and the states currently occupied.

Usage

```
xsect(msdata, xtime = 0)
```

Arguments

`msdata`	An object of class `"msdata"`, such as output by `msprep`
`xtime`	The time point at which the intersection is to be made

Details

It is possible that subjects have moved to one of the absorbing states prior to `xtime`; this is NOT taken into account. The function `xsect` only concerns subjects currently (at `time`) at risk.

Value

A list containing `idstate`, a data frame containing `id`'s and `state`, the number of the state currently occupied; `atrisk`, the number at risk, and `prop`, a table counting how many of those at risk occupy which state.

Author(s)

Hein Putter <H.Putter@lumc.nl>

Examples

```
tmat <- trans.illdeath(names=c("Tx","PR","RelDeath"))
data(ebmt3) # data from Section 4 of Putter, Fiocco & Geskus (2007)
msebmt <- msprep(time=c(NA,"prtime","rfstime"),status=c(NA,"prstat","rfsstat"),
data=ebmt3,trans=tmat)
# At the start everyone is in state 1 (default xtime=0 is used)
xsect(msebmt)
# At 5 years
xsect(msebmt, xtime=1826)
```

Index