

# Package ‘pssm’

April 7, 2015

**Version** 1.0

**Type** Package

**Date** 2015-03-18

**Title** Piecewise Exponential Model for Time to Progression and Time from Progression to Death

**Author** David A. Schoenfeld [aut, cre]

**Maintainer** David A. Schoenfeld <dschoenfeld@partners.org>

**Description** Estimates parameters of a piecewise exponential model for time to progression and time from progression to death with interval censoring of the time to progression and covariates for each distribution using proportional hazards.

**License** GPL-2

**Imports** MASS, abind, numDeriv, MHadaptive

**Depends** methods

**Collate** 'pssm.generate.data.R' 'llikef.R' 'rprog.R' 'rsurv.R'  
'startv.R' 'generic\_functions.R' 'pssm.R' 'pssm.survivalcurv.R'  
'pssm.simulate.R' 'pssm.power.R'  
'pssm.noninferiority.boundary.R'

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-04-07 00:30:38

## R topics documented:

pssm-package	2
plot-methods	3
print-methods	3
pssm	4
pssm-class	5
pssm.generate.data	7
pssm.noninferiority.boundary	8
pssm.power	9

pssm.simulate . . . . .	11
pssm.summary-class . . . . .	12
pssm.survivalcurv . . . . .	13
summary-methods . . . . .	14

<b>Index</b>	<b>15</b>
--------------	-----------

---

pssm-package	<i>Joint proportional hazards model for survival and progression with piecewise exponential hazard</i>
--------------	--

---

## Description

Fits a joint model for interval censored time to progression and subsequent survival. Progression is assumed to always occur before death. The underlying hazard function is assumed to be piecewise constant.

## Details

Package: pssm  
 Type: Package  
 Version: 1.0  
 Date: 2013-06-27  
 License: Public

The function `pssm` fits the model and outputs a `pssm-object` S4 object, this can input to `pssm.survivalcurv` which outputs a function of time, that can be used to create time to progression and survival curves for the model with specified covariate values.

In addition `pssm.simulate` is given to generate data to test the model. Functions "llikef", "rprog", "sprog" are helper functions, methods "plot", "print" and "summary" are available.

## Author(s)

David A. Schoenfeld

Maintainer: Who to complain to <dschoenfeld@partners.org>

## References

~~ Literature or other references for background information ~~

## See Also

`pssm-class`, `pssm.generate.data`, `pssm.object` `pssm`, `pssm.simulate`, `pssm.survivalcurv`  
`plot-methods` `pssm.power`

**Examples**

```
showClass("pssm")
```

---

plot-methods

*Plot method for plot in Package pssm*

---

**Description**

Does several different plots for a pssm object.

**Methods**

`signature(x = "pssm")` With only one argument a solid line is plotted for the survival curve for time to progression and then `m` dashed lines are plotted, each being the survival curve for survival after progression if progression occurs at 0, and at each of the times that the hazard rate changes.

`signature(x = "pssm", type="progression", cov1=, cov2=)` The time for progression curve is plotted note that `cov1` and `cov2` must be matrices with first dimension the number of curves and second dimension the number of covariates.

`signature(x = "pssm", type="progression", cov1=, cov2=)` The time for progression curve is plotted note that `cov1` and `cov2` must be matrices with first dimension the number of curves and second dimension the number of covariates.

**See Also**

[pssm-package](#), [pssm.generate.data](#), [pssm](#), [pssm.simulate](#), [pssm.survivalcurv](#) [pssm-class](#)  
[pssm.power](#)

**Examples**

```
showClass("pssm")
```

---

print-methods

*Print method for print in Package pssm*

---

**Description**

Prints a summary of a pssm object

**Methods**

`signature(x = "pssm")` creates a `pssm.summary` object and prints it. The `pssm.summary` object has the estimate of the covariate coefficients and their standard errors.

`signature(x = "pssm.summary")` Prints the summary object

**See Also**

[pssm-package](#), [pssm.generate.data](#), [pssm.object pssm](#), [pssm.simulate](#), [pssm.survivalcurv](#)  
[pssm-class pssm.power](#)

pssm

*Fits a joint piecewise exponential model for progression and survival***Description**

'pssm' fits a proportional hazards model where the underlying hazard of time to progression and subsequent survival are both piecewise constant. The hazard from progression to death is different for each interval for which the hazard of progression is constant. The covariates for progression and death after progression may be different. Time to progression is interval censored while time to death is right censored. Patients who die must have a progression interval with a right endpoint that is at or before the death time.

**Usage**

```
pssm(progr, survv, dat, intervals = 5, start = NULL, rescale = 1)
```

**Arguments**

progr	a formula object of the form <code>Surv(time0,time1)~covariates</code> for the time to progression, if <code>NULL</code> it fits a piecewise exponential model for survival time only. The progression time is assumed to be interval censored between <code>time0</code> and <code>time1</code> . If progression does not occur then <code>time1=NA</code> .
survv	a formula object of the form <code>Surv(time,cns)~covariates</code> for time from progression to death, if <code>NULL</code> a model for time to progression is fit. The code for "cns" is 1=dead, 0=alive.
dat	A data frame with the data used in the estimation
intervals	An integer indicating the number of intervals in the time to progression or survival model. If both <i>progr</i> and <i>survv</i> are present than <i>intervals</i> is the number of intervals for the progression model. The survival model will have $intervals*(intervals+1)/2$ hazard parameters. If the model doesn't converge the number of intervals will be reduced by the program until convergence occurs.
start	starting values for the parameter vector, suppose $intervals=m$ and there are $n_1$ progression covariates and $n_2$ survival covariates then the parameter vector will be of length, $m + m(m + 1)/2 + n_1 + n_2$ the survival covariates are ordered by progression interval i.e the first $m$ , are survival intervals for patients who progressed between time 0 and 1, the next $m - 1$ or for patients who progressed between time 1 and 2, ... The first $m * (m + 1)/2$ elements are the log-survival hazard, the next $m$ the log-progression hazard followed by the progression covariate vector followed by the survival covariate vector. If <code>NULL</code> the starting values are estimated.
rescale	A number that will multiply the survival and progression times before processing. This will be changed by the program if the maximum survival or progression time is greater than $mr$ .

**Value**

An S4 [pssm.object](#) containing the estimated parameters, their standard errors, their variance covariance matrix, and details of the model that was fit. The generic functions `plot`, `summary` and `print` are defined for the object.

**Author(s)**

David A. Schoenfeld

**See Also**

[pssm-class](#), [pssm-package](#), [pssm.generate.data](#), [pssm.object](#), [pssm.simulate](#), [pssm.survivalcurv](#), [plot-methods](#), [pssm.power](#) [pssm.noninferiority.boundary](#)

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
#generate data
set.seed(359)
u<-pssm.generate.data(theta1=.5,theta2=0,phaz.progression=rep(log(-log(.3)/4),5),
  phaz.survival=rep(log(-log(.2)/4),15),accrual=2, followup=2.9,m=5,n=100,times=c(.8,2.1,3.4))
#estimate based on joint model
ps<-pssm(surv(tprog0,tprog1)~rx,surv(tdeath,cdeath)~rx,u,intervals=3,rescale=1)
summary(ps)
plot(ps,type='survival',cov1=matrix(c(1,0),2,1),cov2=matrix(c(1,0),2,1))
plot(ps)
plot(ps,type='progression',cov1=matrix(c(1,0),2,1),cov2=matrix(c(1,0),2,1))
#estimate time to progression only (tumor free survival)
ps1=pssm(surv(tprog0,tprog1)~rx,NULL,u,intervals=3,rescale=1)
summary(ps1)
plot(ps1)
plot(ps1,type='progression',cov1=matrix(c(1,0),2,1),cov2=matrix(c(1,0),2,1))
#estimate survival only
ps2=pssm(NULL,surv(tdeath,cdeath)~rx,u,intervals=3,rescale=1)
summary(ps2)
plot(ps2)
plot(ps2,type='survival',cov1=matrix(c(1,0),2,1),cov2=matrix(c(1,0),2,1))
```

---

pssm-class

*Class "pssm"*

---

**Description**

This class fits a joint piecewise exponential model to interval censored progression data followed by death as well as a piecewise exponential model to interval censored progression data without subsequent death data or to right censored survival data. It produces a `pssm` object.

## Objects from the Class

Objects can be created by calls of the form `ps=pssm(...)` See:[pssm.object](#)

## Slots

`call`: Object of class "call"  
`convergence`: Object of class "numeric" Integer which is zero if algorithm converged  
`loglike`: object of class "function" log-likelihood function  
`estimates`: Object of class "numeric" Numeric vector of estimated parameters (see [pssm.object](#)), note hazards are log transformed  
`se.estimates`: Object of class "numeric" Numeric vector of standard errors of the estimated parameters  
`covariance.estimates`: Object of class "matrix" Variance covariance matrix of estimates  
`estimates.progression`: Object of class "numeric" Covariate coefficients for the time to progression covariates  
`se.estimates.progression`: Object of class "numeric" Standard Errors of the above estimates  
`estimates.survival`: Object of class "numeric" Covariate coefficients for the time from progression to death covariates  
`se.estimates.survival`: Object of class "numeric" Standard Errors of the above estimates  
`hazard.progression`: Object of class "numeric" Log-hazard of progression  
`hazard.survival`: Object of class "numeric" Log-hazard of survival after progression  
`intervals`: Object of class "integer" Number of intervals used in estimation  
`rescale`: Object of class "numeric" Constant that was used to rescale data before estimation  
`formula.progression`: Object of class "formula"  
`formula.survival`: Object of class "formula"  
`progression.covariate.list`: Object of class "character"  
`survival.covariate.list`: Object of class "character"  
`message`: Object of class "character" Message from program

## Methods

**plot** signature(x = "pssm"): ... See [plot-methods](#)  
**print** signature(x = "pssm"):  
**summary** signature(object = "pssm"): Creates summary object with covariate effects

## Author(s)

David A. Schoenfeld

## See Also

[pssm-package](#), [pssm.generate.data](#), [pssm](#), [pssm.simulate](#), [pssm.survivalcurv](#), [plot-methods](#), [pssm.power](#)

**Examples**

```
showClass("pssm")
```

---

pssm.generate.data	<i>Generates data with a piecewise exponential proportional hazards model for progression and survival</i>
--------------------	--

---

**Description**

Generates data for the pssm model

**Usage**

```
pssm.generate.data(theta1 = 0.2, theta2 = 0.2,
  phaz.progression = log(-log(0.3)/4) * rep(1, 5),
  phaz.survival = log(-log(0.15)/4) * rep(1, 15), accrual = 3,
  followup = 2, m = 5, n = 400, times = NULL, delta = 0.15, alloc=c(1,1), seed=NULL)
```

**Arguments**

theta1	Coefficient for treatment covariate for time to progression
theta2	Coefficient for treatment covariate for survival after progression
phaz.progression	log-hazard vector for progression
phaz.survival	log-hazard vector for survival
accrual	accrual time
followup	follow up time
m	number of intervals, maximum of times
n	number of samples
times	vector of planned times that progression is assessed, if NULL delta isn't used and times are between $(2*i-1)*m/8$ , $(2*(i+1)-1)*m/8$ for $i=1, \dots, m-2$
delta	variation around the assessment times
alloc	Allocation between control and treatment group $c(1, 1)$ is one to one allocation.
seed	Seed for the random number generator if you don't want the data that is analyzed to change.

**Value**

Data frame

```
tprog0,tprog1,cdeath,tdeath,rx=c(rep(0,n/2),rep(1,n/2))
```

tprog0	Last time the patient was free of progressive disease
tprog1	First time progressive disease was noted, NA if no progression
cdeath	1 if the patient died, 0 otherwise
tdeath	Time of death or last follow up
rx	Treatment indicator a covariate which is 0 or 1

**Author(s)**

David Schoenfeld

**See Also**

[pssm-class](#), [pssm-package](#), [pssm.object](#) [pssm](#), [pssm.simulate](#), [pssm.survivalcurv](#) [plot-methods](#)  
[pssm.power](#)

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
u=pssm.generate.data(theta1=.5,theta2=0,phaz.progression=rep(log(-log(.3)/4),5),
phaz.survival=rep(log(-log(.2)/4),15),accrual=2, followup=2.9,m=5,n=300,times=c(.8,2.1,3.4))
```

---

pssm.noninferiority.boundary

*Non inferiority boundary as a function of the prior precision*

---

**Description**

Generates a function of time for specified covariate values from a "pssm" object created by pssm that fits a joint proportional hazards and survival model using a piecewise exponential underlying hazard function

**Usage**

```
pssm.noninferiority.boundary(x,time,cov1,cov2,
                             approximate=TRUE,alpha=0.05,iterations=50000)
```

**Arguments**

x	A pssm object created by pssm
time	The time point at which the treatments are to be compared
cov1	a $a * b$ matrix of values of the $b$ covariates affecting the time to progression
cov2	a matrix of values of the covariates affecting survival after progression with the same number of rows as cov1
approximate	If false it uses a metropolis hasting MCMC to calculate the upper boundary otherwise it uses a normal approximation for the posterior
alpha	The one sided alpha level of the bound
iterations	Number of MCMC iterations, only used if approximate=FALSE



**Value**

A function is returned that takes a vector input. For each element the function outputs the lower alpha confidence bound of the survival difference

**Author(s)**

David A. Schoenfeld

**See Also**

[pssm-class](#), [pssm-package](#), [pssm.generate.data](#), [pssm.object](#), [pssm](#), [pssm.simulate](#), [plot-methods](#), [pssm.power](#)

---

pssm.power

*A function to calculate the power for pssm*

---

**Description**

This function calculates the power, the sample size required for a given power, or the value of the parameter(delta) for a given sample size and power for testing for treatment effect on time to progression and for test for non-inferiority in terms of the survival at a fixed time point

**Usage**

```
pssm.power(parameter=c("N", "Delta", "Power", "Power-Simulation"),
            endpoint=c("Progression", "Survival"), progression.delta=0.2, nsamp=100, alpha=0.025,
            pow=0.80, tsurv=3, h0=0, prior.surv=0, alloc=c(1,1), phaz=log(.2), shaz=log(.2), accrual=3,
            followup=2, inc=0.05, delta=0, intervals=1, nsims=500, n=5000, seed=NULL)
```

**Arguments**

parameter	A character vector of parameters to be calculated. For "N", "Delta" and "Power" the calculation proceeds by first calculating the standard error of the parameter measuring the treatment effect in a trial with n (default 3000) patients it then multiplies by the square root of n to get the proper factor to calculate the parameter called for. For "Power-Simulation" it runs a simulation to calculate the power for a specified sample size nsamp and progression.delta. This was included as a check on the asymptotics which is the basis of using one simulation to get a standard error and then scaling up or down to calculate the sample size or detectable difference.
endpoint	A character vector with either "Progression" or "Survival". In the first case it assumes that endpoint for which the sample size is being calculated is the parameter measuring the effect of treatment on the time to progression, which is log of the hazard ratio of placebo to active treatment. In the second it is the difference in the survival curves at the time tsurv

progression.delta	The log of the hazard ratio of placebo to active treatment. This used to calculate the sample size "N" and "Power". When Survival is the endpoint it is used to calculate the survival difference due to the effect of time to progression on overall survival.
nsamp	This is the sample size used to calculate "Power" or "Delta".
alpha	The one sided significance level to be used.
pow	The power required. This is used in the calculation of "N" and "Delta".
tsurv	The time used for the survival difference.
h0	The non-inferiority boundary for the survival difference given as a positive number.
prior.surv	The prior precision of the parameter that measures the effect of treatment on survival after progression. Note that this is $1/\sigma^2$ where $\sigma$ is the prior standard deviation of this parameter.
alloc	Allocation between control and treatment group c(1, 1) is one to one allocation.
phaz	Log hazard of progression in the placebo group
shaz	Log hazard of survival after progression in the placebo and treatment group
accrual	Accrual period
followup	Follow up period.
inc	Time period between visits for accessing progression.
delta	Variation around visit times
intervals	Number of intervals to fit hazards
nsims	Number of simulations
n	Sample size used to estimate the standard Error.
seed	Seed for the random number generator if you don't want the data that is analyzed to change.

**Value**

A data frame is returned with variables, "parameter", "endpoint" and "value".

**Author(s)**

David A. Schoenfeld

**See Also**

[pssm-class](#), [pssm-package](#), [pssm.generate.data](#), [pssm.object](#), [pssm](#), [pssm.simulate](#), [pssm.survivalcurv](#), [plot-methods](#)

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
pssm.power(parameter="Power",progression.delta=.4,nsamp=450,h0=0.04,nsims=100)
```

---

pssm.simulate                      *Simulates pssm analysis*

---

**Description**

Simulates survival and progression model

**Usage**

```
pssm.simulate(nruns, theta1 = 0.2, theta2 = 0.2,
  phaz.progression = log(-log(0.3)/4) * rep(1, 5),
  phaz.survival = log(-log(0.15)/4) * rep(1, 15), accrual = 3, followup = 2,
  m = 5, n = 500, rescale = 1, etime=4.5, seed=NULL)
```

**Arguments**

nruns	number of simulations
theta1	Coefficient for treatment covariate for time to progression
theta2	Coefficient for treatment covariate for survival after progression
phaz.progression	log-hazard vector for progression
phaz.survival	log-hazard vector for survival
accrual	accrual time
followup	follow up time
m	number of intervals, maximum of times
n	number of samples
rescale	Factor to multiply times by
etime	The time that survival is evaluated
seed	Seed for the random number generator if you don't want the data that is analyzed to change.

**Value**

A three entry list

objects	A nruns by 5 list of "pssm" objects which result from five estimates: 1-Both survival and progression, 2-progression only, 3-tumor free progression, 4-survival on treatment 0, 5-survival on treatment 1. Note that the tumor-free progression rates are estimated assuming that progression occurs at the first time it is detected.
ests	estimates for treatment effects followed by death rate and progression rate
se.ests	standard errors for the treatment effects

**Author(s)**

David A. Schoenfeld

**See Also**

[pssm-class](#), [pssm-package](#), [pssm.generate.data](#), [pssm.object](#), [pssm](#), [pssm.survivalcurv](#), [plot-methods](#), [pssm.power](#)

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
#creates 1 simulations with 50 observations
pss=pssm.simulate(nruns=1,theta1=.2,theta2=.2,phaz.progression=log(-log(.3)/4)*rep(1,3),
  phaz.survival=log(-log(.15)/4)*rep(1,6),accrual=1, followup=2, ,m=3,n=50, rescale=1)
#Summary of run 1
summary(pss$objects[[1]][[1]])
```

---

pssm.summary-class      *Class "pssm.summary"*

---

**Description**

The class produced by the method `summary` applied to a `pssm` object

**Slots**

`call`: Character representation of calling function  
`convergence`: Character string indicating convergence  
`coefficients`: The coefficients of the covariates and their standard errors as a data frame  
`confidence.bounds`: Confidence bounds on the coefficients as a data frame

**Author(s)**

David A. Schoenfeld

**See Also**

[summary-methods](#)

---

pssm.survivalcurv      *time to progression and time to death function for a "pssm" object*

---

### Description

Generates a function of time for specified covariate values from a "pssm" object created by pssm that fits a joint proportional hazards and survival model using a piecewise exponential underlying hazard function

### Usage

```
pssm.survivalcurv(x, cov1, cov2, timeToProgression = FALSE, covariance = TRUE)
```

### Arguments

x	A pssm object created by pssm
cov1	a $a * b$ matrix of values of the $b$ covariates affecting the time to progression
cov2	a matrix of values of the covariates affecting survival after progression with the same number of rows as cov1
timeToProgression	if FALSE estimates the survival curve, if TRUE estimates two probabilities, the probability of being disease free before $t$ and the probability of progressing before $t$ but surviving after $t$
covariance	if TRUE the covariance matrix is returned as an attribute of the function value

### Details

pssm.survivalcurv returns a function the argument of which is the vector of times for which survival probabilities are desired.

### Value

A function is returned, the input to the function is a vector of times, and an optional parameter indicating the prior precision on the estimate of the -log hazard ratio of the effect of survival after progression on the last covariate in the survival model (presumed to be treatment) and the output is a data frame with columns described below:

Note that to conduct the bayesian analysis the Covariance needs to be set to T.

rep	indicates what is estimated (see below), values are "s1" or "s2"
time	Time, $t$
covariates	Columns indicating covariates for survival and progression

**estimate** Estimate, If `timeToProgression` is `TRUE` and the estimation was done with both survival and time to progression the "s1" value is the probability that a patient will progress before time  $t$  but survive longer than  $t$ . In that case the value at "s2" is the probability a patient will be disease free before  $t$ . Otherwise `rep` will only equal "s1" and it will be the probability that survival or progression occurs latter than  $t$  as the case may be.

### Author(s)

David A. Schoenfeld

### See Also

[pssm-class](#), [pssm-package](#), [pssm.generate.data](#), [pssm.object](#), [pssm](#), [pssm.simulate](#), [plot-methods](#), [pssm.power](#)

### Examples

```
#generate data for plot
u<-pssm.generate.data(theta1=.5,theta2=0,phaz.progression=rep(log(-log(.3)/4),5),
  phaz.survival=rep(log(-log(.15)/4),15),accrual=2, followup=2.9,m=5,
  n=50,times=c(1,2,3),delta=0.5)
#estimate parameters
ps<-pssm(surv(tprog0,tprog1)~rx,surv(tdeath,cdeath)~rx,dat=u,intervals=3)
#calculate survival curves
vs<-pssm.survivalcurv(ps,cov1=matrix(c(0,1),2,1),cov2=matrix(c(0,1),2,1),covariance=TRUE)
t=c(0,2,4,4.99)
curves=vs(t)
#plot survival curves
plot(t,curves$estimate[curves$rx==0],type='l',lty=2,ylim=c(0,1),
  main='Survival Curve',xlab='Time',ylab='Probability of Survival')
points(t,curves$estimate[curves$rx==1],type='l',lty=1,xlim=c(0,5))
```

---

summary-methods

*Methods for Function summary in Package pssm*

---

### Description

Methods for function summary in package **base**

### Methods

`signature(object = "pssm")` Produces a `pssm.summary` object with the coefficients of the covariates, their standard erros and confidence bounds in a data frame.

# Index

- \*Topic **Power**
    - pssm.power, 9
  - \*Topic **Sample Size**
    - pssm.power, 9
  - \*Topic **other possible keyword(s)**
    - plot-methods, 3
  - \*Topic **classes**
    - pssm-class, 5
  - \*Topic **interval censoring**
    - pssm, 4
    - pssm.generate.data, 7
    - pssm.noninferiority.boundary, 8
    - pssm.simulate, 11
    - pssm.survivalcurv, 13
  - \*Topic **methods**
    - plot-methods, 3
    - print-methods, 3
    - summary-methods, 14
  - \*Topic **package**
    - pssm, 4
    - pssm-package, 2
    - pssm.generate.data, 7
    - pssm.simulate, 11
  - \*Topic **progression free survival**
    - pssm, 4
    - pssm.generate.data, 7
    - pssm.noninferiority.boundary, 8
    - pssm.simulate, 11
    - pssm.survivalcurv, 13
  - \*Topic **survival**
    - pssm, 4
    - pssm.generate.data, 7
    - pssm.noninferiority.boundary, 8
    - pssm.simulate, 11
    - pssm.survivalcurv, 13
- plot, ANY-method (plot-methods), 3  
plot, pssm-method (plot-methods), 3
- plot-methods, 3  
print, ANY-method (print-methods), 3  
print, pssm-method (print-methods), 3  
print, pssm.summary-method (print-methods), 3  
print-methods, 3  
pssm, 2-4, 4, 6, 8-10, 12, 14  
pssm-class, 5  
pssm-object (pssm-class), 5  
pssm-package, 2  
pssm.class (pssm-class), 5  
pssm.generate.data, 2-6, 7, 9, 10, 12, 14  
pssm.noninferiority.boundary, 5, 8  
pssm.object, 2, 4-6, 8-10, 12, 14  
pssm.object (pssm-class), 5  
pssm.package (pssm-package), 2  
pssm.power, 2-6, 8, 9, 9, 12, 14  
pssm.simulate, 2-6, 8-10, 11, 14  
pssm.summary-class, 12  
pssm.survivalcurv, 2-6, 8, 10, 12, 13
- summary, ANY-method (summary-methods), 14  
summary, pssm-method (summary-methods), 14  
summary-methods, 14