

Package ‘Umacs’

April 17, 2009

Title Universal Markov chain sampler

Version 0.924

Date 2007/10/10

Author Jouni Kerman <jouni@kerman.com>

Maintainer Jouni Kerman <jouni@kerman.com>

Description Universal Markov Chain Sampler

Namespace Umacs

Suggests rv, R2WinBUGS

Depends R(>= 2.5.1), methods, utils

License GPL-2

Repository CRAN

Date/Publication 2007-10-10 07:51:07

R topics documented:

Umacs-package	2
as.bugs.mcts	3
as.rv.mcts	4
codetemplate-class	5
codetemplatelist-class	6
Data	7
Data-class	8
DFunction	9
DFunction-class	10
Gibbs	11
Gibbs-class	12
LocalFunction	14
LocalFunction-class	15

LocalObject	16
LocalObject-class	17
mcts	18
mcts-class	19
Metropolis	20
Metropolis-class	21
Parameter-class	23
ParameterList-class	24
PMetropolis	25
PMetropolis-class	26
PSMetropolis	27
PSMetropolis-class	29
RawCode	30
RawCode-class	31
Sampler	32
Sampler-class	34
SamplerCheck-class	35
SamplingScheme-class	36
show-methods	37
SMetropolis	37
SMetropolis-class	39
snippet-class	40
snippetlist-class	41
snippets-methods	41
templates-methods	42
Trace	42
Trace-class	43
Index	45

Umacs-package *Creating Iterative Samplers*

Description

Please refer to the vignette: `vignette("Umacs")`.

Details

Package:	Umacs
Version:	0.911
Date:	2007/01/14
Namespace:	Umacs
Enhances:	R2WinBUGS
Depends:	R(>= 1.9.0), methods, utils
License:	GPL version 2 or newer

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

as.bugs.mcts *Coerce an 'mcts' object to a 'bugs' object*

Description

Coerce an mcts object (esp. those made with Umacs samplers) to an rv object (those compatible with the package 'rv')

Usage

```
## S3 method for class 'mcts':  
as.bugs(x)  
as.bugs.mcts(x)
```

Arguments

x An mcts object to coerce

Details

Class 'bugs' is a simulation-based random variable class. The simulations stored inside `obj` will be copied into the components of an 'rv' object.

Value

An object of class `rv`.

Note

Requires package 'rv'.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

Package R2WinBUGS.

as.rv.mcts

Coerce an 'mcts' object to an 'rv' object

Description

Coerce an mcts object (esp. those made with Umacs samplers) to an rv object (those compatible with the package 'rv')

Usage

```
as.rv.mcts(obj)
```

Arguments

`obj` An mcts object to coerce

Details

Class 'rv' is a simulation-based random variable class. The simulations stored inside `obj` will be copied into the components of an 'rv' object.

Value

An object of class `rv`.

Note

Requires package 'rv'.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

codetemplate-class *Class "codetemplate"*

Description

A 'codetemplate' contains a `snippetlist`, which can be merged with 'templates'.

Used internally by [Umacs](#)

Details

More documentation follows (possibly) in later versions.

Slots

snippets: Object of class "snippetlist" ~~

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

```
codetemplatelist-class  
      Class "codetemplatelist"
```

Description

A ‘codetemplatelist’ contains a list of `codetemplates`, containing snippets that can be merged with ‘templates’ at once.

Used internally by [Umacs](#)

Details

More documentation follows (possibly) in later versions.

Slots

slist: Object of class "list" ~~

Methods

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package ‘rv’.

Data

Specify Umacs Parameter As Data

Description

Generates an object of class `Data` that embeds the argument value as an object *locally* in the sampler function generated by `Sampler`. The name of the object is taken to be the associated parameter name.

Usually a call to this function is not necessary, since `Data` is implicitly called when the value of a named parameter in the argument list of `Sampler` is an R object but not a function, numeric object or an instance of the Umacs class `Parameter`.

Usage

```
Data(x)
```

Arguments

`x` a numeric data object to be associated to the model parameter

Details

The parameter is supposed to be ‘unmodeled data,’ specified by the likelihood.

Value

An object of class `Data`.

Note

The `Sampler` function recognizes numeric, named parameters and wraps them by default in `Data(...)`, so usually this function is never explicitly called by the user.

To embed an R function locally into the sampler function, use `LocalFunction` instead.

To embed just any non-function object locally into the sampler function, use `LocalObject` instead.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.
[Sampler](#) for how to create a sampler function; see the vignette for details.
Package 'rv'.

Data-class

Class definition of "Data" (Umacs package)

Description

This class is used only internally, in Umacs.

Details

Users need only use the function [Data](#) inside a [Sampler](#) function call, which outputs a Data object and is directly processed by [Sampler](#).

Slots

data: Object of class "list" ~~
initfun: Object of class "logical" ~~
id: Object of class "numeric" ~~
name: Object of class "character" ~~
stemname: Object of class "character" ~~
basename: Object of class "character" ~~
savename: Object of class "character" ~~
impute: Object of class "logical" ~~
length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package ‘rv’.

DFunction

Generate a DFunction object for Umacs Sampler function

Description

Creates a `DFunction` object that is used to embed an R function that performs a Gibbs updating step for a parameter that is a *deterministic function* within the sampler function generated by [Sampler](#).

The main difference between `DFunction` and [Gibbs](#) is that for `DFunction`, there is no initializing function. Consequently it is supposed to be updated only *after* all parameter that it directly depends on are available.

The sampler is built using the Umacs function [Sampler](#); and the argument function (`update`) is embedded in the sampler function.

Usage

```
DFunction(update)
```

Arguments

<code>update</code>	An R function (with no arguments) that samples directly from a (fully) conditional distribution. May refer directly to parameters specified in the <code>codelinkSampler</code> function call. Must return the updated value.
---------------------	---

Details

`DFunction` is to be used only within the `Sampler` function call.

Alternatively one can just supply the function (`update`) in the `Sampler` function call; `Sampler` will call `DFunction` for you.

The arguments of the update functions are ignored. The body of the function can contain references to any parameters that are defined in the `Sampler` function call, or to any variables in an enclosing environment (such as the Global Environment).

The name of the parameter that is being updated *not* specified here, but only in the parameter list in the `Sampler` function call.

Value

An object of class `DFunction-class` to be further processed by the Umacs function `Sampler`.

Note

`DFunction` *must* be associated with a parameter name (in the `Sampler` function call). If you specify a plain function without a name, it will be interpreted as an argument to a `RawCode` function call and *not* as an argument to `DFunction`.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

`Umacs-package` for an overview of the Umacs package.

`Sampler` for how to create a sampler function; see the vignette for details.

Package ‘rv’.

`DFunction-class` *Class definition of "DFunction" (Umacs package)*

Description

This class is used only internally, in Umacs.

Details

Users need only use the function `DFunction` inside a `Sampler` function call, which outputs a Gibbs object and is directly processed by `Sampler`. (Alternatively, a call to `DFunction` can be omitted and only a function specified, as `Sampler` is smart enough to wrap

Slots

id: Object of class "numeric" ~~

name: Object of class "character" ~~

stemname: Object of class "character" ~~

basename: Object of class "character" ~~

savename: Object of class "character" ~~

impute: Object of class "logical" ~~

length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

Methods

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

Gibbs

Generate a Gibbs object for Umacs Sampler function

Description

Creates a `Gibbs` object that is used to initialize an R function that performs a Gibbs updating step within the sampler function generated by [Sampler](#).

The sampler is built using the Umacs function [Sampler](#); and the arguments (`update` and `init`) are embedded in the sampler function.

Usage

`Gibbs(update, init)`

Arguments

<code>update</code>	An R function, with no arguments, that samples directly from a (fully) conditional distribution. May refer directly to variables that are specified in the <code>codelinkSampler</code> function call. Must return the updated value.
<code>init</code>	An R function that returns a (random) starting point for a Markov chain for the parameter. To be executed before the iteration loop (before each chain).

Details

`Gibbs` is to be used only within the `Sampler` function call. The arguments of the `update` and `init` functions are ignored. The body of the function can contain references to any parameters that are defined in the `Sampler` function call, or to any variables in an enclosing environment (such as the Global Environment).

The name of the parameter that is being updated is *not* specified in the `Gibbs` object (and not in any of the argument functions either); a call `Gibbs` *must* be associated with a parameter name within a `Sampler` function call.

Value

An object of class `Gibbs`, to be further processed by the `Umacs` function `Sampler`.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. `Umacs`: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs](#) and [Sampler](#) for examples on how to create sampler functions.

`Gibbs-class`

Class definition of "Gibbs" (Umacs package)

Description

This class is used only internally, in `Umacs`.

Details

Users need only use the function `Gibbs` inside a `Sampler` function call, which outputs a `Gibbs` object and is directly processed by `Sampler`.

Slots

id: Object of class "numeric" ~~
name: Object of class "character" ~~
stemname: Object of class "character" ~~
basename: Object of class "character" ~~
savename: Object of class "character" ~~
impute: Object of class "logical" ~~
length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

`LocalFunction`*Generate a LocalFunction object for Umacs Sampler function*

Description

Generates an object that is used to define a *local* R function in a sampler function generated by [Sampler](#).

Usually a call to this function is not necessary, since `LocalFunction` is implicitly done when the value of a *named* parameter in the argument list of [Sampler](#) is an R function.

Usage

```
LocalFunction(x)
```

Arguments

`x` An R function

Details

`LocalFunction` is to be used only within the `Sampler` function call; the value of the argument (`x`) is made available (locally) in a sampling function that is built using the Umacs function `Sampler`.

The name of the parameter that will have the value (`x`) is *not* specified as an argument, but only within the [Sampler](#) function call.

Value

An object of class `LocalFunction`, to be further processed by the Umacs function [Sampler](#).

Note

A call to `LocalFunction` must be associated with a parameter name.

Components of vector parameters specified to be `LocalFunctions` cannot be updated, but those specified to be `Data` can be updated.

To insert 'raw code' in the iteration loop, use the function [RawCode](#).

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.
[Sampler](#) for how to create a sampler function; see the vignette for details.
Package 'rv'.

`LocalFunction-class`

Class definition of "LocalFunction" (Umacs package)

Description

This class is used only internally, in Umacs.

Details

Users need only use the function `LocalFunction` inside a `Sampler` function call, which outputs a `LocalFunction` object and is directly processed by `Sampler`.

Slots

func: Object of class "list" ~~
id: Object of class "numeric" ~~
name: Object of class "character" ~~
stemname: Object of class "character" ~~
basename: Object of class "character" ~~
savename: Object of class "character" ~~
impute: Object of class "logical" ~~
length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package ‘rv’.

LocalObject

Generate a LocalObject object for Umacs Sampler function

Description

Generates an object that is used to initialize an R object locally in a sampler function generated by `Sampler`. Usually a call to this function is not necessary, since it is implicitly done when the value of a named parameter in the argument list of `Sampler` is an R object but not a function, numeric object or an instance of the Umacs class `Parameter`.

Usage

```
LocalObject(x)
```

Arguments

`x` An R object (not a function)

Details

`LocalObject` is to be used only within the `Sampler` function call; the value of the argument (`x`) is made available (locally) in a sampling function that is built using the Umacs function `Sampler`.

The name of the parameter that will have the value (`x`) is *not* specified here, but only within the `Sampler` function call.

The components of a parameter vector initialized as `Data` parameters cannot be individually updated using Gibbs and Metropolis updating schemes.

For local *functions*, use the function `LocalFunction` instead.

Value

An object of class `LocalObject`, to be further processed by the Umacs function `Sampler`.

Note

`LocalObject` must be associated with a parameter name.

Components of vector parameters specified to be `LocalObjects` cannot be updated, but those specified to be `Data` can be updated.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

LocalObject-class *Class definition of "LocalObject" (Umacs package)*

Description

This class is used only internally, in Umacs.

Details

Users need only use the function `LocalObject` inside a `Sampler` function call, which outputs a `LocalObject` object and is directly processed by `Sampler`.

Slots

vble: Object of class "list" ~~
id: Object of class "numeric" ~~
name: Object of class "character" ~~
stemname: Object of class "character" ~~
basename: Object of class "character" ~~
savename: Object of class "character" ~~
impute: Object of class "logical" ~~
length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

mcts

mcts Objects

Description

A wrapper for `new('mcts', ...)`

Usage

```
mcts(chains, n.burnin = 0, n.chains = numeric(0), n.iter = numeric(0), n.sims = n.i
## S3 method for class 'mcts':
print(x, ...)
## S3 method for class 'mcts':
summary(object, ...)
## S3 method for class 'mcts':
as.matrix(x, ...)
```

Arguments

```
chains
n.burnin
n.chains
n.iter
n.sims
col.names
save.all
x           mcts object
object     mcts object
...        Other arguments
```

Value

An object of the class `mcts` ([mcts-class](#)).

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

`mcts-class`*Class 'mcts' ("Markov Chain Time Series")*

Description

`mcts = "Markov Chain Time Series"`; a wrapper for holding matrix of simulation of MCMC chains along with relevant information. Umacs returns simulations in the form of `mcts` objects.

Objects from the Class

Objects can be created using `mcts(...)`

Slots

chains: ("list")
n.burnin: ("numeric")
n.iter: ("numeric")
n.sims: ("numeric")
save.all: ("logical")
col.names: ("character")
summary: Object of class "matrix" ~~

Methods

Note

`mcts` objects can be coerced into random variable (`rv`) objects using `as.rv`. See package ‘`rv`’ for details.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package ‘`rv`’.

Metropolis

Generate a Metropolis object for the Umacs Sampler function

Description

Generates an object including the information necessary to generate a Metropolis updating step for a parameter, within the main loop of an iterative sampler function.

The sampler is built using the Umacs function [Sampler](#); and the arguments (`update` and `init`) are embedded in the sampler function.

Usage

```
Metropolis(logpost = NULL, init, adapt.scale.function = "scaleAdapt.Metropolis", scale.jump, kernel.jump, class)
```

Arguments

<code>logpost</code>	An R function calculating the value (a scalar) of the unnormalized log-posterior function given all other arguments
<code>init</code>	An R function that returns a (random) starting point (scalar, vector, or array) for a Markov chain for the parameter
<code>adapt.scale.function</code>	name of the function used to adapt the proposal kernel (usually, just ignore and use the default)
<code>scale.jump</code>	The scale of the proposal kernel matrix, if adaptation is not to be used
<code>kernel.jump</code>	Proposal-generating density class. Now, the only possible value is ‘Gaussian’.
<code>class.</code>	(For internal use only)

Details

`Metropolis` is to be used only within the `Sampler` function call.

The arguments of the `update` and `init` functions are ignored. The body of the function can contain references to any parameters that are defined in the `Sampler` function call, or to any variables in an enclosing environment (such as the Global Environment).

The name of the parameter that is being updated is *not* specified here, but only within the `Sampler` function call.

Value

An object of class `Metropolis`, to be further processed by the `Umacs` function `Sampler`.

Note

If the associated parameter is *scalar-valued*, use `SMetropolis` instead.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. `Umacs`: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

`Umacs-package` for an overview of the `Umacs` package.

`Sampler` for how to create a sampler function; see the vignette for details.

Package ‘rv’.

`Metropolis-class` *Class definition of "Metropolis" (Umacs package)*

Description

This class is used only internally, in `Umacs`.

Details

Users need only use the function `Metropolis` inside a `Sampler` function call, which outputs a `Metropolis` object and is directly processed by `Sampler`.

Slots

id: Object of class "numeric" ~~
name: Object of class "character" ~~
stemname: Object of class "character" ~~
basename: Object of class "character" ~~
savename: Object of class "character" ~~
impute: Object of class "logical" ~~
length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

 Parameter-class *Class definition of "Parameter" (Umacs package)*

Description

This class is used only internally, in Umacs.

Details

This is the base class for all model parameters that need to be updated within the sampler loop. `Parameter` is the parent class for “sampling schemes” (`SamplingScheme-class`, which contain such as `Gibbs` and `Metropolis`), and “raw code” (`RawCode-class`).

Slots

id: ("numeric") an unique id

name: ("character") full specification of the parameter name, with brackets and but without suffixes.

stemname: ("character") ‘stem’ of the name, not including brackets or suffixes, (e.g. “theta”).

basename: ("character") the name of the parameter, including brackets, but not including suffixes such as “.mis” (e.g. "theta[1]")

savename: ("character") the parameter specification as the user specified it, (e.g. "theta[1].mis")

impute: ("logical") (not used)

length: ("numeric") length of the actual ‘missing’ values: `codelength(theta.mis)`

total.length: ("numeric") length of the ‘stemname’ vector : `codelength(theta)`

dimension: ("numeric")

check.ok: ("logical") codeTRUE if parameter check went ok

vector.names: ("character") (not used)

value: ("numeric")

missing: ("numeric")

snippets: Object of class "snippetlist" ~~

Methods

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

ParameterList-class

Class definition of "ParameterList" (Umacs package)

Description

This class is used only internally, in Umacs.

Details

Wrapper for a list of "Parameter" objects.

Slots

params: Object of class "list" ~~

check.ok: Object of class "logical" ~~

slist: Object of class "list" ~~

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

 PMetropolis

 Generate an PMetropolis object for the Umacs Sampler function

Description

Generates an object including the information necessary to generate a “Parallel Metropolis” updating step for a parameter, within the main loop of an iterative sampler function.

The sampler is built using the Umacs function `Sampler`; and the arguments (`update` and `init`) are embedded in the sampler function.

Usage

```
PMetropolis(logpost = NULL, init, proposal = NULL, adapt.scale.function = "scaleAda
```

Arguments

<code>logpost</code>	An R function calculating the value (a vector!) of the unnormalized log-posterior function given all other arguments
<code>init</code>	An R function that returns a (random) starting point (a vector) for a Markov chain for the parameter
<code>proposal</code>	(not used now)
<code>adapt.scale.function</code>	name of the function used to adapt the proposal kernel (usually, just ignore and use the default)
<code>scale.jump</code>	The scale of the proposal kernel matrix, if adaptation is not to be used
<code>class.</code>	(for internal use only; ignore)
<code>byCol</code>	codeTRUE if the parameter is a matrix consisting of columns of independent vectors; the default is codeFALSE: the parameter is a matrix consisting of rows of independent vectors

Details

The ‘PMetropolis’ is short for ‘Parallel Metropolis’: the parameter to be updated is a *matrix* consisting of *rows of independently updateable vectors*. That is, each row of the matrix can be updated independently (“in parallel”) using a (vector) Metropolis updating step. (Alternatively the independent vectors can be in the columns of the matrix, if the argument `byCol` is set to TRUE.)

In other words, you would use `PMetropolis` if conditional on other parameters in the model, you have independent (exchangeable) vector-valued parameters that you wish to update using the Metropolis algorithm.

If instead you have conditionally independent *scalar-valued* parameters, use `PSMetropolis` instead.

An alternative to using `PMetropolis` is to define each row as a separate parameter, but this would be cumbersome and require lots of repetitive coding, and the code would not be optimized.

`PMetropolis` is to be used only within the `Sampler` function call.

The arguments of the update and init functions are ignored. The body of the function can contain references to any parameters that are defined in the `Sampler` function call, or to any variables in an enclosing environment (such as the Global Environment).

The name of the parameter that is being updated is *not* specified here, but only within the `Sampler` function call.

Value

An object of class `PMetropolis`, to be further processed by the Umacs function `Sampler`.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

`Umacs-package` for an overview of the Umacs package.

`Sampler` for how to create a sampler function; see the vignette for details.

Package 'rv'.

`PMetropolis-class` *Class definition of "PMetropolis" (Umacs package)*

Description

This class is used only internally, in Umacs.

Details

Users need only use the function `PMetropolis` inside a `Sampler` function call, which outputs a `PMetropolis` object and is directly processed by `Sampler`.

Slots

byCol: Object of class "logical" ~~

id: Object of class "numeric" ~~

name: Object of class "character" ~~

stemname: Object of class "character" ~~

basename: Object of class "character" ~~

savename: Object of class "character" ~~

```

impute: Object of class "logical" ~~
length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

```

Methods

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package ‘rv’.

PSMetropolis

Generate an PSMetropolis object for the Umacs Sampler function

Description

Generates an object including the information necessary to generate a Metropolis updating step for a parameter, within the main loop of an iterative sampler function.

The sampler is built using the Umacs function `Sampler`; and the arguments (`update` and `init`) are embedded in the sampler function.

See `Sampler` for further details.

Usage

```
PSMetropolis(logpost = NULL, init, proposal = NULL, adapt.scale.function = "scaleAc
```

Arguments

<code>logpost</code>	An R function calculating the value (a vector!) of the unnormalized log-posterior function given all other arguments
<code>init</code>	An R function that returns a (random) starting point (a vector) for a Markov chain for the parameter
<code>proposal</code>	(not used now)
<code>adapt.scale.function</code>	name of the function used to adapt the proposal kernel (usually, just ignore and use the default)
<code>scale.jump</code>	The scale of the proposal kernel matrix, if adaptation is not to be used
<code>class.</code>	(for internal use only; ignore)

Details

PSMetropolis is a special case of the PMetropolis sampling scheme. The parameter to be updated is a *vector* consisting of *independently updateable scalars*. That is, each component of the vector can be updated independently (“in parallel”) using a (scalar) Metropolis updating step.

In other words, you would use PSMetropolis if conditional on other parameters in the model, you have independent (exchangeable) vector-valued parameters that you wish to update using the Metropolis algorithm.

If instead you have conditionally independent *vector-valued* parameters, that is, a *matrix* of independently updateable rows or columns, use PMetropolis.

An alternative to using PSMetropolis is to define each component as a separate parameter, but this would be very cumbersome and require lots of repetitive coding, and the code would not be optimized.

PSMetropolis is to be used only within the `Sampler` function call.

The arguments of the `update` and `init` functions are ignored. The body of the function can contain references to any parameters that are defined in the `Sampler` function call, or to any variables in an enclosing environment (such as the Global Environment).

The name of the parameter that is being updated is *not* specified here, but only within the `Sampler` function call.

See the documentation for examples.

Value

An object of class PSMetropolis to be further processed by the Umacs function `Sampler`.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

PSMetropolis-class *Class definition of "PSMetropolis" (Umacs package)*

Description

This class is used only internally, in Umacs.

Details

Users need only use the function `PSMetropolis` inside a `Sampler` function call, which outputs a `PSMetropolis` object and is directly processed by `Sampler`.

Slots

id: Object of class "numeric" ~~
name: Object of class "character" ~~
stemname: Object of class "character" ~~
basename: Object of class "character" ~~
savename: Object of class "character" ~~
impute: Object of class "logical" ~~
length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

 RawCode

Generate a RawCode object for Umacs Sampler function

Description

Generates a RawCode object that is used to initialize an R function that performs a Gibbs updating set within the sampler function.

Usage

```
RawCode(update, init = NULL)
```

Arguments

<code>update</code>	An R function to be executed in the iteration loop
<code>init</code>	An R function to be executed before the iteration loop (chain); by default no code (codeNULL)

Details

RawCode is to be used only within the Sampler function call; the value of the argument (`x`) is made available (locally) in a sampling function that is built using the Umacs function Sampler

The name of the parameter that will have the value (`x`) is *not* specified here, but only within the Sampler function call.

Value

An object of class RawCode to be further processed by the Umacs function Sampler

Note

Usually a call to this function is not necessary, since it is implicitly done when an R function is given in the argument list of the Sampler function call without a name.

RawCode *must* be associated with a parameter name. Components of vector parameters specified to be RawCodes cannot be updated, but those specified to be Data can be updated.

To specify a local function that is supposed to be available, use the function LocalFunction.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

RawCode-class

Class definition of "RawCode" (Umacs package)

Description

This class is used only internally, in Umacs.

Details

Users need only use the function [RawCode](#) inside a [Sampler](#) function call, which outputs a RawCode object and is directly processed by [Sampler](#).

Slots

id: Object of class "numeric" ~~
name: Object of class "character" ~~
stemname: Object of class "character" ~~
basename: Object of class "character" ~~
savename: Object of class "character" ~~
impute: Object of class "logical" ~~
length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

Sampler

Generate a Markov chain sampler

Description

Generates a Markov chain sampling function given a list of parameters

Usage

```
Sampler(...)
```

Arguments

... A list of parameters including Gibbs and/or Metropolis updating routines, Raw functions and other objects

Details

`Sampler` generates an R function to produce simulations wrapped in an `mcts` object that can be coerced into random variable objects, for example.

A `Sampler` function call features a named argument list that represents the named parameters in a Gibbs or Metropolis sampler. The names are specified by the user (e.g. "theta", "alpha") and the values are either constants (vectors of data frames), special Umacs objects implementing a sampling scheme (`Gibbs`, `Metropolis`, etc.). Additionally, the argument list can contain assignments to special internal variables, or unnamed arguments such as raw code to be embedded in the sampling loop.

Special Umacs variables that are identified by the prepending dot (e.g. `.logpost`) and are not thus actual parameters in the model.

No user-defined parameter can start with a dot.

Unnamed arguments are either ‘raw’ functions that are embedded ‘as is’ inside the sampling loop, or special Umacs objects such as [Trace](#) objects, which provide a trace plots of a specified variable. The code that the user provides in the form of direct sampling functions (via [Gibbs](#)) or log-posterior functions (via the various Metropolis updating schemes) can have references to ‘global’ variables that are available in the workspace, but also to ‘local’ variables that are specified within the `Sampler` function call.

Ideally, the function should *only* have references to local variables so that the resulting sampler function is a stand-alone function that encapsulates a complete sampling environment independent of the working environment.

‘Raw code’ that is specified as an unnamed function in the argument list is embedded as a function, so any assignments done using the `<-` operator are local. To change directly the value of a *parameter*, use the `<<-` operator.

The allowed values for the *parameters* that (are named by the user and) can be passed as arguments are:

A constant numeric value (vector or array). The variable is made available to the functions embedded in the sampler (but not in Global Environment!), or, a `SamplingScheme` object that is built using one of the following possible functions: [Gibbs](#), [Metropolis](#), [SMetropolis](#), [PMetropolis](#), [PSMetropolis](#).

These objects provide the updating and initialization functions for a Gibbs updating step, or the log-posterior function and an initialization function for a Metropolis updating step.

(Technically speaking, a ‘Gibbs’ updating step is just a special case of a ‘Metropolis’ updating step in the sense that all samples generated using the function of a ‘Gibbs’ updating step are accepted unconditionally.)

The allowed values for the *unnamed* objects that can be passed as arguments are a [Trace](#) object, or a plain function (‘raw code’). The plain function is converted internally first to a [RawCode](#) object, so giving a plain function is made possible just for convenience. See [RawCode](#) for details.

Value

An R function that can be run to produce the simulations.

See the help page for [Umacs](#) to learn about the usage of the sampling function.

The function generated by `Sampler` returns an `mcts` object that can be converted into random variable objects.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

Sampler-class

Class definition of "Sampler" (Umacs package)

Description

This class is used only internally, in Umacs.

Details

Users need only use the function [Sampler](#).

Slots

params: ("ParameterList") A single object containing the parameters of the sampler.

all.param.names: ("character") names of the parameters that are 'random'

check.only: ("logical") flag to

snippets: ("snippetlist") Snippets of which sampler is built from

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

`SamplerCheck-class` *Class definition of "SamplerCheck" (Umacs package)*

Description

This class is used only internally, in Umacs.

Details

This class is a derivative of the [Sampler-class](#), used for doing a test run on the sampler before passing the sampling function to the user.

Slots

params: Object of class "ParameterList" ~~
all.param.names: Object of class "character" ~~
check.only: Object of class "logical" ~~
snippets: Object of class "snippetlist" ~~

Methods

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

SamplingScheme-class

Class definition of "SamplingScheme" (Umacs package)

Description

This class is used only internally, in Umacs.

Details

This is the base class for all model parameters that need a Metropolis- style updating routine. SamplingScheme is the parent class for such classes as [Gibbs](#) and [Metropolis](#).

Slots

id: Object of class "numeric" ~~
name: Object of class "character" ~~
stemname: Object of class "character" ~~
basename: Object of class "character" ~~
savename: Object of class "character" ~~
impute: Object of class "logical" ~~
length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

Methods

Note

RawCode ([RawCode-class](#)) inherits directly from [Parameter-class](#).

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

 show-methods

Show a summary of an mcts object

Description

Displays the summary of the simulations stored inside an mcts object.

Methods

obj = "mcts" Displays the summary of the simulations stored inside an mcts object.

 SMetropolis

Generate an SMetropolis object for the Umacs Sampler function

Description

Generates an object including the information necessary to generate an SMetropolis updating step for a parameter, within the main loop of an iterative sampler function.

The sampler is built using the Umacs function [Sampler](#); and the arguments (update and init) are embedded in the sampler function.

See [Sampler](#) for further details.

Usage

```
SMetropolis(logpost = NULL, init, adapt.scale.function = "scaleAdapt.Metropolis", s
```

Arguments

<code>logpost</code>	An R function calculating the value (a scalar) of the unnormalized log-posterior function given all other arguments
<code>init</code>	An R function that returns a (random) starting point (a scalar!) for a Markov chain for the parameter
<code>adapt.scale.function</code>	name of the function used to adapt the proposal kernel (usually, just ignore and use the default)
<code>scale.jump</code>	The scale of the proposal kernel matrix, if adaptation is not to be used
<code>class.</code>	(for internal use only)

Details

SMetropolis is a special case of the Metropolis sampling scheme; it is optimized for scalar parameters.

SMetropolis is to be used only within the Sampler function call.

The arguments of the update and init functions are ignored. The body of the function can contain references to any parameters that are defined in the Sampler function call, or to any variables in an enclosing environment (such as the Global Environment).

The name of the parameter that is being updated is *not* specified here, but only within the Sampler function call.

Value

An object of class SMetropolis, to be further processed by the Umacs function [Sampler](#).

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

SMetropolis-class *Class definition of "SMetropolis" (Umacs package)*

Description

This class is used only internally, in Umacs.

Details

Users need only use the function `SMetropolis` inside a `Sampler` function call, which outputs a `SMetropolis` object and is directly processed by `Sampler`.

Slots

id: Object of class "numeric" ~~
name: Object of class "character" ~~
stemname: Object of class "character" ~~
basename: Object of class "character" ~~
savename: Object of class "character" ~~
impute: Object of class "logical" ~~
length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

Methods

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

snippet-class *Class "snippet"*

Description

A class whose instances accommodate “code snippets” created by `substitute()` or `quote()` etc.

Used internally by [Umacs](#)

Slots

code: Object of class "list" ~~

args: Object of class "list" ~~

name: Object of class "character" ~~

output: Object of class "numeric" ~~

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

snippetlist-class *Class "snippetlist"*

Description

A class whose instance accommodates a list of “code snippets” of class “snippet.”
Used internally by [Umacs](#)

Slots

snippets: Object of class "list" ~~

Methods

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.
[Sampler](#) for how to create a sampler function; see the vignette for details.
Package ‘rv’.

snippets-methods *Get the ‘snippets’ of an object*

Description

Used internally in [Umacs](#).

Methods

obj = "snippetlist" ~~describe this method here
obj = "codetemplate" ~~describe this method here
obj = "codetemplatelist" ~~describe this method here

templates-methods *Get the 'templates' of an object*

Description

Used internally in [Umacs](#).

Methods

obj = "codetemplate" ~~describe this method here
obj = "Parameter" ~~describe this method here
obj = "SamplingScheme" ~~describe this method here
obj = "RawCode" ~~describe this method here
obj = "Gibbs" ~~describe this method here
obj = "Data" ~~describe this method here
obj = "DFunction" ~~describe this method here
obj = "LocalObject" ~~describe this method here
obj = "LocalFunction" ~~describe this method here
obj = "ParameterList" ~~describe this method here
obj = "Sampler" ~~describe this method here
obj = "SamplerCheck" ~~describe this method here
obj = "Trace" ~~describe this method here
obj = "Metropolis" ~~describe this method here
obj = "SMetropolis" ~~describe this method here
obj = "PSMetropolis" ~~describe this method here
obj = "PMetropolis" ~~describe this method here

Trace

Generate a Trace object for Umacs Sampler function

Description

Generates an object including the information necessary to show a trace plot of a scalar parameter in a Umacs sampling function.

The sampler is built using the Umacs function [Sampler](#); this function is only to be used within the [Sampler](#) function call.

Usage

```
Trace(name, label = name, ylim = NULL)
```

Arguments

<code>name</code>	Name of the scalar parameter to be traced, e.g. "codetheta[1]".
<code>label</code>	Label to be shown in the trace plot window. By default the same as codename.
<code>ylim</code>	Range of the vertical axis. By default, the range is extended as necessary to display the highs and the lows of the trace.

Details

If `ylim` is not specified (if it is `NULL`), the range of the vertical axis is adjusted automatically.

`Trace` is to be used only within the `Sampler` function call.

Value

An object of class `Trace`, to be further processed by the `Umacs` function `Sampler`.

Author(s)

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. `Umacs`: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

`Umacs-package` for an overview of the `Umacs` package.

`Sampler` for how to create a sampler function; see the vignette for details.

Package 'rv'.

`Trace-class`*Class definition of "Trace" (Umacs package)*

Description

This class is used only internally, in `Umacs`.

Details

Users need only use the function `Trace` inside a `Sampler` function call, which outputs a `Trace` object and is directly processed by `Sampler`.

Slots

id: Object of class "numeric" ~~
name: Object of class "character" ~~
stemname: Object of class "character" ~~
basename: Object of class "character" ~~
savename: Object of class "character" ~~
impute: Object of class "logical" ~~
length: Object of class "numeric" ~~
total.length: Object of class "numeric" ~~
dimension: Object of class "numeric" ~~
check.ok: Object of class "logical" ~~
vector.names: Object of class "character" ~~
value: Object of class "numeric" ~~
missing: Object of class "numeric" ~~
snippets: Object of class "snippetlist" ~~

Methods**Author(s)**

Jouni Kerman <jouni@kerman.com>

References

Kerman, Jouni. Umacs: A Universal Markov Chain Sampler. Technical report, Columbia University, New York.

See Also

[Umacs-package](#) for an overview of the Umacs package.

[Sampler](#) for how to create a sampler function; see the vignette for details.

Package 'rv'.

Index

*Topic **iteration**

- codetemplate-class, 4
- codetemplatelist-class, 5
- Data, 6
- Data-class, 7
- DFunction, 8
- DFunction-class, 9
- Gibbs, 10
- Gibbs-class, 11
- LocalFunction, 13
- LocalFunction-class, 14
- LocalObject, 15
- LocalObject-class, 16
- mcts, 17
- mcts-class, 18
- Metropolis, 19
- Metropolis-class, 20
- Parameter-class, 22
- ParameterList-class, 23
- PMetropolis, 24
- PMetropolis-class, 25
- PSMetropolis, 26
- PSMetropolis-class, 28
- RawCode, 29
- RawCode-class, 30
- Sampler, 31
- Sampler-class, 33
- SamplerCheck-class, 34
- SamplingScheme-class, 35
- show-methods, 36
- SMetropolis, 36
- SMetropolis-class, 38
- snippet-class, 39
- snippetlist-class, 40
- snippets-methods, 40
- templates-methods, 41
- Trace, 41
- Trace-class, 42
- Umacs-package, 1

*Topic **manip**

- as.bugs.mcts, 2
- as.rv.mcts, 3

- as.bugs (*as.bugs.mcts*), 2
- as.bugs.mcts, 2
- as.matrix.mcts (*mcts*), 17
- as.rv.mcts, 3

- codetemplate-class, 4
- codetemplatelist-class, 5

- Data, 6, 7
- Data-class, 7
- DFunction, 8, 8, 9
- DFunction-class, 9
- DFunction-class, 9

- Gibbs, 8, 10, 11, 22, 32, 35
- Gibbs-class, 11

- LocalFunction, 6, 13, 14, 15
- LocalFunction-class, 14
- LocalObject, 6, 15, 15, 16
- LocalObject-class, 16

- mcts, 17, 32
- mcts-class, 18
- mcts-class, 18
- Metropolis, 19, 20, 22, 32, 35
- Metropolis-class, 20

- Parameter-class, 35
- Parameter-class, 22
- ParameterList-class, 23
- PMetropolis, 24, 25, 32
- PMetropolis-class, 25
- print.mcts (*mcts*), 17
- PSMetropolis, 26, 28, 32
- PSMetropolis-class, 28

- RawCode, 9, 13, 29, 30, 32
- RawCode-class, 22, 35
- RawCode-class, 30

- Sampler, 4–21, 23–30, 31, 31, 33, 34, 36–43
- Sampler-class, 34
- Sampler-class, 33
- SamplerCheck-class, 34
- SamplingScheme-class, 22
- SamplingScheme-class, 35
- show, mcts-method (*show-methods*), 36
- show-methods, 36
- SMetropolis, 20, 32, 36, 38
- SMetropolis-class, 38
- snippet-class, 39
- snippetlist-class, 40
- snippets (*snippets-methods*), 40
- snippets, codetemplate-method (*snippets-methods*), 40
- snippets, codetemplatelist-method (*snippets-methods*), 40
- snippets, snippetlist-method (*snippets-methods*), 40
- snippets-methods, 40
- summary.mcts (*mcts*), 17

- templates (*templates-methods*), 41
- templates, codetemplate-method (*templates-methods*), 41
- templates, Data-method (*templates-methods*), 41
- templates, DFunction-method (*templates-methods*), 41
- templates, Gibbs-method (*templates-methods*), 41
- templates, LocalFunction-method (*templates-methods*), 41
- templates, LocalObject-method (*templates-methods*), 41
- templates, Metropolis-method (*templates-methods*), 41
- templates, Parameter-method (*templates-methods*), 41
- templates, ParameterList-method (*templates-methods*), 41
- templates, PMetropolis-method (*templates-methods*), 41
- templates, PSMetropolis-method (*templates-methods*), 41
- templates, RawCode-method (*templates-methods*), 41
- templates, Sampler-method (*templates-methods*), 41
- templates, SamplerCheck-method (*templates-methods*), 41
- templates, SamplingScheme-method (*templates-methods*), 41
- templates, SMetropolis-method (*templates-methods*), 41
- templates, Trace-method (*templates-methods*), 41
- templates-methods, 41
- Trace, 32, 41, 42
- Trace-class, 42

- Umacs, 4, 5, 11, 32, 39–41
- Umacs (*Umacs-package*), 1
- Umacs-package, 4, 5, 7–10, 12, 14–21, 23, 25, 26, 28–31, 33, 34, 36, 37, 39, 40, 42, 43
- Umacs-package, 1