USAGE

```
Bayes.pen (y, x, joint = TRUE, prior = NULL, nIter = 8000, burnIn = 500, thin = 1,
                                    update=200, max.steps=NULL)
```

ARGUMENTS

y	Response vector of length n
x	Matrix of predictors with p columns and n rows. A column of ones should not be included in this matrix, as an intercept will automatically be added.
joint	If TRUE, both the joint and marginal results will be reported. If FALSE, only the marginal results will be reported. It is recommended to keep joint=TRUE unless the dimension is extremely large. The MCMC sampling will remain the same for both methods, but the joint approach requires a LARS algorithm following the MCMC, which can be computationally demanding.
prior	*prior$varE, prior$varBR* List containing the priors for the residual variance and the variance in the Gaussian prior, each providing degree of freedom (*$df*) and scale (*$S*). These are the parameters of the scaled inverse-χ^2 distributions assigned to variance components. The prior expectation of variance parameters is *S/(df-2).* See the help file for package BLR for more details.
nIter, burnIn, thin	Number of MCMC iterations, burn-in, and thinning.
update	How often should the MCMC print to the screen?
max.steps	The maximum number of steps used in the LARS algorithm to compute the final solution path. Default is *8 * min (n - 1, p)* as in LARS.

VALUE

order.joint	The order of entry of the variables for the joint credible sets. As in other penalized regression solution paths, a variable may exit after entering the model, particularly under correlation. If so, this is denoted via a negative index, as in the LARS package. This will be NULL If joint=FALSE.

order.marg	The order of entry of the variables for the marginal credible sets.
df.joint	The degrees of freedom for each solution in the joint credible set sequence. This will be NULL If joint=FALSE.
df.marg	The degrees of freedom for each solution in the marginal credible set sequence.
coef.joint	Estimated coefficients for the sequence of models from the joint credible sets. These are computed via least squares for each model in the sequence. Each row represents a model. The first column is the estimated intercept, with the remaining columns being the slopes. Only models of size up to the rank of the design matrix are given. This will be NULL if joint=FALSE. These estimates are given, but the user may choose to use other estimates for each selected model.
coef.marg	Estimated coefficients for the sequence of models from the marginal credible sets. These are computed via least squares for each model in the sequence. Each row represents a model. The first column is the estimated intercept, with the remaining columns being the slopes. Only models of size up to the rank of the design matrix are given. These estimates are given, but the user may choose to use other estimates for each selected model.
SSE.joint	Residual sum of squares for the sequence of models from the joint credible sets when estimated via least squares. This will be NULL if joint=FALSE.
SSE.marg	Residual sum of squares for the sequence of models from the marginal credible sets when estimated via least squares.

NOTES

The MCMC sampling is based on the R package BLR.

REFERENCE

Bondell, H. D. and Reich, B. J. (2012). Consistent high-dimensional Bayesian variable selection via penalized credible regions. *Journal of the American Statistical Association* – IN PRESS.
http://dx.doi.org/10.1080/01621459.2012.716344

EXAMPLES

```
# n=60, p=50 AR (1) with rho of 0.9

require(mvtnorm)
rho = 0.9
sigma = 1
n = 60
p = 50
times = 1:p
H = abs(outer(times, times, "-"))
V = sigma * rho^H

set.seed(77)
beta = rep(0,p)
beta[11:15] = runif(5)
beta[36:40] = runif(5)

x = rmvnorm(n,rep(0,p),V)
y = x%*%beta + rnorm(n)

# Fit the model
prior = list(varE=list(df=3,S=1),varBR=list(df=3,S=1))
example_fit = Bayes.pen (y, x, prior=prior)
example_fit$order.joint
 >  [1] 14 37 13 39 15 48 38 12 40 23 45 26 22 28  1  4 33 24 36 47 10 50 44 43 29
 >  [26] 41 35  3  7  9 16 19 11 46 27  5 20 25 31 32  6 34 49 42  8 18 17  2 21 30

example_fit$order.marg
 >  [1] 14 37 13 39 48 15 40 38 12 45 28 24 23 22  1 10  4 29 47 26 41 50  7 44 46
 >  [26] 33 36 43  9  3 35 27 11  5 16 20 31 19 34 32 25  6 49 42  8 21  2 30 17 18

#   Note that the true signals are random uniforms on (0, 1) and are on coefficients 11-15
#   and 36-40. Both solutions paths pick out the majority of true signals early in the path.



# Try it with n=60, p=1000

require(mvtnorm)
rho = 0.9
sigma = 1
n = 60
p = 1000
times = 1:p
H = abs(outer(times, times, "-"))
```

```
V = sigma * rho^H

set.seed(77)
beta = rep(0,p)
beta[11:15] = runif(5)
beta[36:40] = runif(5)

x = rmvnorm(n,rep(0,p),V)
y = x%*%beta + rnorm(n)

# Fit the model
prior = list(varE=list(df=3,S=1),varBR=list(df=3,S=1))
example_fit = Bayes.pen (y, x, prior=prior)

# Results are similar as above.
```