Package 'bravo'

May 27, 2026

Type Package

Title Bayesian Screening and Variable Selection

Version 4.0.0

Author Dongjin Li [aut, cre],
Debarshi Chakraborty [aut],
Somak Dutta [aut],
Vivekananda Roy [ctb]

Maintainer Dongjin Li <liyangxiaobei@gmail.com>

Description Performs Bayesian variable screening and selection for ultra-high dimensional linear regression models.Also contains an user friendly web application to perform multi trait GWAS.

License GPL-3

Encoding UTF-8

Depends R (>= 3.0.2), methods

Imports Rcpp (>= 1.0.2), Matrix (>= 1.2-17),dplyr,parallel,doParallel,foreach,shiny,bslib,memuse,shinyjs

LinkingTo Rcpp

NeedsCompilation yes

RoxygenNote 7.3.3

Repository CRAN

Date/Publication 2026-05-27 14:10:08 UTC

Contents

`basic.sven.model` *Run SVEN with Optimal Parameters*

Description

Fits SVEN on (x, y) using the supplied tuned parameters.

Usage

```
basic.sven.model(x, y, params, seed = 2441139)
```

Arguments

`x`	Cleaned SNP matrix.
`y`	Phenotype vector.
`params`	Named numeric vector with lambda and w.
`seed`	Random seed (default 2441139).

bits *Bayesian Iterated Screening (ultra-high, high or low dimensional).*

Description

Perform Bayesian iterated screening in Gaussian regression models

Usage

```
bits(X, y, lam = 1, w = 0.5, pp = FALSE, max.var = nrow(X), verbose = TRUE)
```

Arguments

X	An $n \times p$ matrix. Sparse matrices are supported and every care is taken not to make copies of this (typically) giant matrix. No need to center or scale.
y	The response vector of length n.
lam	The slab precision parameter. Default: 1.
w	The prior inclusion probability of each variable. Default: 1/2.
pp	Boolean: If FALSE (default) the algorithm stops after including max.var many variables. If true, the posterior probability stopping rule is used.
max.var	The maximum number of variables to be included.
verbose	If TRUE (default) will show the variable index included in each iteration.

Value

A list with components

model.pp	An integer vector of the screened model.
postprobs	The sequence of posterior probabilities until the last included variable.
lam	The value of lam, the slab precision parameter.
w	The value of w, the prior inclusion probability.

References

Wang, R., Dutta, S., Roy, V. (2021) Bayesian iterative screening in ultra-high dimensional settings. https://arxiv.org/abs/2107.10175

Examples

```
n=50; p=100;
TrueBeta <- c(rep(5,3),rep(0,p-3))

rho <- 0.6
x1 <- matrix(rnorm(n*p), n, p)
X <- sqrt(1-rho)*x1 + sqrt(rho)*rnorm(n)
y <- 0.5 + X %*% TrueBeta + rnorm(n)
```

```
res<-bits(X,y, pp=TRUE)
res$model.pp # the vector of screened model
res$postprobs # the log (unnormalized) posterior probabilities corresponding to the model.pp.
```

`bwas` *BWAS: Bayesian GWAS for a Single Trait*

Description

Runs SVEN and UNITE for one trait.

Usage

```
bwas(x, y, params, bigx, ehits = 20)
```

Arguments

`x`	Cleaned SNP matrix.
`y`	Phenotype vector.
`params`	Named numeric vector with lambda and w.
`bigx`	Full SNP matrix.
`ehits`	Expected number of hits (default 20).

`calc.runtime` *Estimate SVEN Runtime*

Description

Times a single SVEN run on the given SNP matrix using a random phenotype.

Usage

```
calc.runtime(X)
```

Arguments

`X`	SNP matrix.

Value

Time taken to run sven() once.

clean *Clean SNP Matrix*

Description

Removes duplicate SNPs and filters low MAF variants.

Usage

```
clean(SNPmat, MAF_threshold = 0.05)
```

Arguments

SNPmat	A sparse SNP matrix (dgCMatrix).
MAF_threshold	Minor Allele Frequency cutoff (default 0.05).

create_param_mat *Create Parameter Grid*

Description

Builds a grid of lambda and w tuning parameters for SVEN.

Usage

```
create_param_mat(x)
```

Arguments

x	Cleaned SNP matrix.

dense2sparse *Convert numeric matrix to sparse matrix*

Description

Reads a numeric genotype file and converts it to a sparse matrix format.

Usage

```
dense2sparse(file.name, num.genotypes, separator, progress = TRUE)
```

Arguments

file.name	Path to the numeric genotype file. Could be (and should be) gzipped.
num.genotypes	Maximum number of genotypes to read. An upper bound is OK.
separator	"\t" or "," etc that separates the entries in a line.
progress	Whether to show a progress bar (default TRUE).

Value

A sparse matrix of class dgCMatix.

---

dense_to_sparse_converter *Launch the Sparse Converter Shiny App*

Description

Opens the Sparse Matrix Converter GUI in your browser.

Usage

```
dense_to_sparse_converter()
```

FDR_corrected *FDR using correlation threshold*

Description

Computes False Discovery Rate using SNP correlation as proximity measure.

Usage

```
FDR_corrected(model, truth, x, threshold = 0.9)
```

Arguments

model	Integer vector of selected SNP indices.
truth	Integer vector of true causal SNP indices.
x	SNP matrix.
threshold	Correlation threshold (default 0.9).

FDR_WS *FDR using window size*

Description

Computes False Discovery Rate using base-pair window as proximity measure.

Usage

```
FDR_WS(model, truth, mapmat, winsize = 1000)
```

Arguments

model	Integer vector of selected SNP indices.
truth	Integer vector of true causal SNP indices.
mapmat	Map matrix with chromosome and position columns.
winsize	Window size in base pairs (default 1000).

FPR_corrected *FPR using correlation threshold*

Description

Computes False Positive Rate using SNP correlation as proximity measure.

Usage

```
FPR_corrected(model, truth, x, threshold = 0.9)
```

Arguments

model	Integer vector of selected SNP indices.
truth	Integer vector of true causal SNP indices.
x	SNP matrix.
threshold	Correlation threshold (default 0.9).

FPR_WS *FPR using window size*

Description

Computes False Positive Rate using base-pair window as proximity measure.

Usage

```
FPR_WS(model, truth, mapmat, winsize = 1000)
```

Arguments

model	Integer vector of selected SNP indices.
truth	Integer vector of true causal SNP indices.
mapmat	Map matrix with chromosome and position columns.
winsize	Window size in base pairs (default 1000).

jcidx *Jaccard Index*

Description

Computes Jaccard index between selected and true causal SNPs.

Usage

```
jcidx(vars, truth)
```

Arguments

vars	Integer vector of selected SNP indices.
truth	Integer vector of true causal SNP indices.

`mip.sven` *Compute marginal inclusion probabilities from a fitted "sven" object.*

Description

This function computes the marginal inclusion probabilities of all variables from a fitted "sven" object.

Usage

```
mip.sven(object, threshold = 0)
```

Arguments

`object`	A fitted "sven" object
`threshold`	marginal inclusion probabilities above this threshold are stored. Default 0.

Value

The object returned is a data frame if the `sven` was run with a single matrix, or a list of two data frames if `sven` was run with a list of two matrices. The first column are the variable names (or numbers if column names of were absent). Only the nonzero marginal inclusion probabilities are stored.

Author(s)

Somak Dutta
Maintainer: Somak Dutta <somakd@iastate.edu>

Examples

```
n <- 50; p <- 100; nonzero <- 3
trueidx <- 1:3
truebeta <- c(4,5,6)
X <- matrix(rnorm(n*p), n, p) # n x p covariate matrix
y <- 0.5 + X[,trueidx] %*% truebeta + rnorm(n)
res <- sven(X=X, y=y)
res$model.map # the MAP model
mip.sven(res)

Z <- matrix(rnorm(n*p), n, p) # another covariate matrix
y2 = 0.5 + X[,trueidx] %*% truebeta  + Z[,1:2] %*% c(-2,-2) + rnorm(n)
res2 <- sven(X=list(X,Z), y=y2)
mip.sven(res2) # two data frames, one for X and another for Z
```

parameter_selection *Select Optimal Tuning Parameters*

Description

Full training step: cleans SNP matrix and tunes SVEN hyperparameters.

Usage

```
parameter_selection(
  X,
  R2 = 0.5,
  betamax = 1,
  n.cores = max(1, parallel::detectCores() - 1),
  hitsize = "all",
  MAF_threshold = 0.05
)
```

Arguments

X	Raw SNP matrix (dgCMatrix).
R2	Heritability (default 0.5).
betamax	Maximum effect size magnitude (default 1).
n.cores	Number of cores (default: detectCores() - 1).
hitsize	One of "all", "small", "medium", "large" (default "all").
MAF_threshold	MAF cutoff (default 0.05).

Value

A list containing the original SNP matrix, cleaned SNP matrix, optimal parameters, MAF threshold, expected hit size, and number of cores used. The list is assigned class "svenetics_trained" for use in downstream functions.

pipeline_single_trait *Run GWAS for a Single Trait*

Description

Runs the full GWAS pipeline for the i-th trait column.

Usage

```
pipeline_single_trait(svenetics_trained_object, i, hitsize = NULL)
```

Arguments

svenetics_trained_object	A trained svenetics object from parameter_selection().
i	Trait index.
hitsize	One of "small", "medium", "large", or NULL.

`predict.sven` *Make predictions from a fitted "sven" object.*

Description

This function makes point predictions and computes prediction intervals from a fitted "sven" object.

Usage

```
## S3 method for class 'sven'
predict(
  object,
  newdata,
  model = c("WAM", "MAP"),
  interval = c("none", "MC", "Z"),
  return.draws = FALSE,
  Nsim = 10000,
  level = 0.95,
  alpha = 1 - level,
  ...
)
```

Arguments

object	A fitted "sven" object
newdata	Matrix of new values for X at which predictions are to be made. Must be a matrix; can be sparse as in Matrix package.
model	The model to be used to make predictions. Model "MAP" gives the predictions calculated using the MAP model; model "WAM" gives the predictions calculated using the WAM. Default: "WAM".
interval	Type of interval calculation. If interval = "none", only point predictions are returned; if interval = "MC", Monte Carlo prediction intervals are returned; if interval = "Z", Z prediction intervals are returned.
return.draws	only required if interval = "MC". if TRUE, the Monte Carlo samples are returned. Default: FALSE.
Nsim	only required if interval = "MC". The Monte Carlo sample size. Default: 10000.
level	Confidence level of the interval. Default: 0.95.
alpha	Type one error rate. Default: 1-level.
...	Further arguments passed to or from other methods.

Value

The object returned depends on "interval" argument. If `interval = "none"`, the object is an `ncol(newdata)` $\times 1$ vector of the point predictions; otherwise, the object is an `ncol(newdata)` $\times 3$ matrix with the point predictions in the first column and the lower and upper bounds of prediction intervals in the second and third columns, respectively.

if return.draws is `TRUE`, a list with the following components is returned:

`prediction`	vector or matrix as above
`mc.draws`	an `ncol(newdata)` $\times$ `Nsim` matrix of the Monte Carlo samples

Author(s)

Dongjin Li and Somak Dutta
Maintainer: Dongjin Li <dongjl@iastate.edu>

References

Li, D., Dutta, S., Roy, V.(2020) Model Based Screening Embedded Bayesian Variable Selection for Ultra-high Dimensional Settings http://arxiv.org/abs/2006.07561

Examples

```
n = 80; p = 100; nonzero = 5
trueidx <- 1:5
nonzero.value <- c(0.50, 0.75, 1.00, 1.25, 1.50)
TrueBeta = numeric(p)
TrueBeta[trueidx] <- nonzero.value

X <- matrix(rnorm(n*p), n, p)
y <- 0.5 + X %*% TrueBeta + rnorm(n)
res <- sven(X=X, y=y)
newx <- matrix(rnorm(20*p), 20, p)
# predicted values at a new data matrix using MAP model
yhat <- predict(object = res, newdata = newx, model = "MAP", interval = "none")
# 95% Monte Carlo prediction interval using WAM
MC.interval <- predict(object = res, model = "WAM", newdata = newx, interval = "MC", level=0.95)
# 95% Z-prediction interval using MAP model
Z.interval <- predict(object = res, model = "MAP", newdata = newx, interval = "Z", level = 0.95)
```

`run_all_params` *Run SVEN Across Parameter Grid*

Description

Simulates a phenotype and evaluates all parameter combinations via Jaccard index.

Usage

```
run_all_params(k, x, R2 = 0.5, nspike = 20, betamax)
```

Arguments

k	Simulation replicate index.
x	Cleaned SNP matrix.
R2	Heritability (default 0.5).
nspike	Number of causal SNPs to simulate (default 20).
betamax	Maximum effect size magnitude.

---

sven *Selection of variables with embedded screening using Bayesian methods (SVEN) in Gaussian linear models (ultra-high, high or low dimensional).*

Description

SVEN is an approach to selecting variables with embedded screening using a Bayesian hierarchical model. It is also a variable selection method in the spirit of the stochastic shotgun search algorithm. However, by embedding a unique model based screening and using fast Cholesky updates, SVEN produces a highly scalable algorithm to explore gigantic model spaces and rapidly identify the regions of high posterior probabilities. It outputs the log (unnormalized) posterior probability of a set of best (highest probability) models. For more details, see Li et al. (2023, https://doi.org/10.1080/10618600.2022.2074428)

Usage

```
sven(
  X,
  y,
  w = NULL,
  lam = NULL,
  Ntemp = 10,
  Tmax = NULL,
  Miter = 50,
  wam.threshold = 0.5,
  log.eps = -16,
  L = 20,
  verbose = FALSE
)
```

Arguments

X	The $n \times p$ covariate matrix or list of two matrices without intercept. The following classes are supported: `matrix` and `dgCMatrix`. Every care is taken not to make copies of these (typically) giant matrices. No need to center or scale these matrices manually. Scaling is performed implicitly and regression coefficient are returned on the original scale. Typically, in a combined GWAS-TWAS type analysis, `X[[1]]` should be a sparse matrix and `X[[2]]` should be a dense matrix.

y	The response vector of length n. No need to center or scale.
w	The prior inclusion probability of each variable. Default: NULL, whence it is set as $\sqrt{n}/p$ if X is a matrix. Or $(\sqrt{n}/p_1, \sqrt{n}/p_2)$ if X is a list of two matrices with p_1 and p_2 columns.
lam	The slab precision parameter. Default: NULL, whence it is set as n/p^2 for as suggested by the theory of Li et al. (2023). Similarly, it's a vector of length two with values $\sqrt{n}/P_1^2$ and $\sqrt{n}/p_2^2$ when X is a list.
Ntemp	The number of temperatures. Default: 10.
Tmax	The maximum temperature. Default: $\log\log p + \log p$.
Miter	The number of iterations per temperature. Default: 50.
wam.threshold	The threshold probability to select the covariates for WAM. A covariate will be included in WAM if its corresponding marginal inclusion probability is greater than the threshold. Default: 0.5.
log.eps	The tolerance to choose the number of top models. See detail. Default: -16.
L	The minimum number of neighboring models screened. Default: 20.
verbose	If FALSE, the function prints the current temperature SVEN is at; the default is TRUE.

Details

SVEN is developed based on a hierarchical Gaussian linear model with priors placed on the regression coefficients as well as on the model space as follows:

$$y|X, \beta_0, \beta, \gamma, \sigma^2, w, \lambda \sim N(\beta_0 1 + X_\gamma \beta_\gamma, \sigma^2 I_n)$$

$$\beta_i|\beta_0, \gamma, \sigma^2, w, \lambda \overset{indep.}{\sim} N(0, \gamma_i \sigma^2/\lambda),\ i = 1, \ldots, p,$$

$$(\beta_0, \sigma^2)|\gamma, w, p \sim p(\beta_0, \sigma^2) \propto 1/\sigma^2$$

$$\gamma_i|w, \lambda \overset{iid}{\sim} Bernoulli(w)$$

where X_γ is the $n \times |\gamma|$ submatrix of X consisting of those columns of X for which $\gamma_i = 1$ and similarly, β_γ is the $|\gamma|$ subvector of β corresponding to γ. Degenerate spike priors on inactive variables and Gaussian slab priors on active covariates makes the posterior probability (up to a normalizing constant) of a model $P(\gamma|y)$ available in explicit form (Li et al., 2020).

The variable selection starts from an empty model and updates the model according to the posterior probability of its neighboring models for some pre-specified number of iterations. In each iteration, the models with small probabilities are screened out in order to quickly identify the regions of high posterior probabilities. A temperature schedule is used to facilitate exploration of models separated by valleys in the posterior probability function, thus mitigate posterior multimodality associated with variable selection models. The default maximum temperature is guided by the asymptotic posterior model selection consistency results in Li et al. (2020).

SVEN provides the maximum a posteriori (MAP) model as well as the weighted average model (WAM). WAM is obtained in the following way: (1) keep the best (highest probability) K distinct models $\gamma^{(1)}, \ldots, \gamma^{(K)}$ with

$$\log P\left(\gamma^{(1)}|y\right) \geq \cdots \geq \log P\left(\gamma^{(K)}|y\right)$$

where K is chosen so that $\log\left\{P\left(\gamma^{(K)}|y\right)/P\left(\gamma^{(1)}|y\right)\right\} >$ `log.eps`; (2) assign the weights

$$w_i = P(\gamma^{(i)}|y)/\sum_{k=1}^{K} P(\gamma^{(k)}|y)$$

to the model $\gamma^{(i)}$; (3) define the approximate marginal inclusion probabilities for the jth variable as

$$\hat{\pi}_j = \sum_{k=1}^{K} w_k I(\gamma_j^{(k)} = 1).$$

Then, the WAM is defined as the model containing variables j with $\hat{\pi}_j >$ `wam.threshold`. SVEN also provides all the top K models which are stored in an $p \times K$ sparse matrix, along with their corresponding log (unnormalized) posterior probabilities.

When `X` is a list with two matrices, say, `W` and `Z`, the above method is extended to `ncol(W)+ncol(Z)` dimensional regression. However, the hyperparameters `lam` and `w` are chosen separately for the two matrices, the default values being `nrow(W)/ncol(W)^2` and `nrow(Z)/ncol(Z)^2` for `lam` and `sqrt(nrow(W))/ncol(W)` and `sqrt(nrow(Z))/ncol(Z)` for `w`.

The marginal inclusion probabities can be extracted by using the function `mip`.

Value

A list with components

`model.map`	A vector of indices corresponding to the selected variables in the MAP model.
`model.wam`	A vector of indices corresponding to the selected variables in the WAM.
`model.top`	A sparse matrix storing the top models.
`beta.map`	The ridge estimator of regression coefficients in the MAP model.
`beta.wam`	The ridge estimator of regression coefficients in the WAM.
`mip.map`	The marginal inclusion probabilities of the variables in the MAP model.
`mip.wam`	The marginal inclusion probabilities of the variables in the WAM.
`pprob.map`	The log (unnormalized) posterior probability corresponding to the MAP model.
`pprob.top`	A vector of the log (unnormalized) posterior probabilities corresponding to the top models.
`stats`	Additional statistics.

Author(s)

Dongjin Li, Debarshi Chakraborty, and Somak Dutta
Maintainer: Dongjin Li <liyangxiaobei@gmail.com>

References

Li, D., Dutta, S., and Roy, V. (2023). Model based screening embedded Bayesian variable selection for ultra-high dimensional settings. Journal of Computational and Graphical Statistics, 32(1), 61-73.

See Also

[mip.sven()] for marginal inclusion probabilities, [predict.sven()](via [predict()]) for prediction for .

Examples

```
n <- 50; p <- 100; nonzero <- 3
trueidx <- 1:3
truebeta <- c(4,5,6)
X <- matrix(rnorm(n*p), n, p) # n x p covariate matrix
y <- 0.5 + X[,trueidx] %*% truebeta + rnorm(n)
res <- sven(X=X, y=y)
res$model.map # the MAP model


Z <- matrix(rnorm(n*p), n, p) # another covariate matrix
y2 = 0.5 + X[,trueidx] %*% truebeta  + Z[,1:2] %*% c(-2,-2) + rnorm(n)
res2 <- sven(X=list(X,Z), y=y2)
```

`svenetics` *Launch the SVENETICS Shiny App*

Description

Opens the SVENETICS GUI in your browser.

Usage

```
svenetics()
```

`svenetics_pipeline` *Full Multi-Trait GWAS Pipeline*

Description

Runs GWAS across all traits and saves results as CSVs.

Usage

```
svenetics_pipeline(
  svenetics_trained_object,
  traitfile,
  hitsizes = NULL,
  save_dir = "~/SVENETICS_RESULTS"
)
```

Arguments

svenetics_trained_object	A trained svenetics object from parameter_selection().
traitfile	Data frame with sample IDs in column 1 and traits in remaining columns.
hitsizes	Character vector of hit sizes per trait, or NULL for "medium" across all.
save_dir	Directory path where results will be saved (default "~/SVENETICS_RESULTS").

Value

Saves the selected SNPs and their MIPs for each trait as separate CSV files in the specified directory. Also returns a list of data frames with the results for each trait.

TPR_corrected *TPR using correlation threshold*

Description

Computes True Positive Rate using SNP correlation as proximity measure.

Usage

```
TPR_corrected(model, truth, x, threshold = 0.9)
```

Arguments

model	Integer vector of selected SNP indices.
truth	Integer vector of true causal SNP indices.
x	SNP matrix.
threshold	Correlation threshold (default 0.9).

TPR_WS *TPR using window size*

Description

Computes True Positive Rate using base-pair window as proximity measure.

Usage

```
TPR_WS(model, truth, mapmat, winsize = 1000)
```

Arguments

model	Integer vector of selected SNP indices.
truth	Integer vector of true causal SNP indices.
mapmat	Map matrix with chromosome and position columns.
winsize	Window size in base pairs (default 1000).

---

tune.sven *Tune SVEN Parameters*

Description

Runs 100 parallel simulations and returns the optimal (lambda, w) pair.

Usage

```
tune.sven(x, R2 = 0.5, ehits = 20, betamax = 1, n.cores)
```

Arguments

x	Cleaned SNP matrix.
R2	Heritability (default 0.5).
ehits	Expected number of causal SNPs.
betamax	Maximum effect size magnitude (default 1).
n.cores	Number of cores for parallel computation.

---

tune.sven.all *Tune SVEN for All Hit Sizes*

Description

Tunes SVEN parameters for small, medium, and large expected hit sizes.

Usage

```
tune.sven.all(x, R2 = 0.5, betamax = 1, n.cores)
```

Arguments

x	Cleaned SNP matrix.
R2	Heritability (default 0.5).
betamax	Maximum effect size magnitude (default 1).
n.cores	Number of cores for parallel computation.

`unite.sven` *UNITE: Post-process SVEN Model*

Description

Propagates MIPs from SVEN hits to the full SNP matrix via LD.

Usage

```
unite.sven(x, bigx, basic_sven_object, y, ehits, threshold = 0)
```

Arguments

`x`	Cleaned SNP matrix.
`bigx`	Full (uncleaned) SNP matrix.
`basic_sven_object`	Fitted SVEN object.
`y`	Phenotype vector.
`ehits`	Expected number of hits.
`threshold`	MIP threshold (default 0).

Index