Package: spFFBS (via r-universe)

May 22, 2026

Title Spatiotemporal Propagation for Multivariate Bayesian Dynamic Learning

Version 0.0-2

Description Implementation of the Forward Filtering Backward Sampling (FFBS) algorithm with Dynamic Bayesian Predictive Stacking (DYNBPS) integration for multivariate spatiotemporal models, as introduced in ``Adaptive Markovian Spatiotemporal Transfer Learning in Multivariate Bayesian Modeling" (Presicce and Banerjee, 2026+) <doi:10.48550/arXiv.2602.08544>. This methodology enables efficient Bayesian multivariate spatiotemporal modeling, utilizing dynamic predictive stacking to improve inference across multivariate time series of spatial datasets. The core functions leverage 'C++' for high-performance computation, making the framework well-suited for large-scale spatiotemporal data analysis in parallel computing environments.

LinkingTo Rcpp, RcppArmadillo

Imports spBPS, Rcpp (>= 1.1.1), foreach, tictoc, abind

Suggests doParallel, mniw, MBA, ggplot2, patchwork, reshape2, knitr, rmarkdown

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

VignetteBuilder knitr

URL https://lucapresicce.github.io/spFFBS/

NeedsCompilation yes

Author Luca Presicce [aut, cre] (ORCID: <https://orcid.org/0009-0005-7062-3523>)

Maintainer Luca Presicce <l.presicce@campus.unimib.it>

Repository https://cran.r-universe.dev

Date/Publication 2026-04-22 15:46:13 UTC

RemoteUrl https://github.com/cran/spFFBS

RemoteRef HEAD

RemoteSha 329b67504fb504854fa9dde2682586ae3bca7916

Contents

`sample_index` *Function to sample integers (index)*

Description

Function to sample integers (index)

Usage

```
sample_index(size, length, p)
```

Arguments

`size`	integer dimension of the set to sample
`length`	integer number of elements to sample
`p`	vector sampling probabilities

Value

vector sample of integers

`spFFBS` *spFFBS: Spatiotemporal Bayesian Pipeline (friendly interface)*

Description

A user-friendly, modular wrapper for running a Bayesian spatiotemporal filtering + weighting pipeline, with optional backward sampling, forecasting, and spatial interpolation.

Usage

```
spFFBS(
  Y,
  G,
  P,
  D,
  grid = list(tau = NULL, phi = NULL),
  prior,
  do_BS = FALSE,
  do_forecast = FALSE,
  do_spatial = FALSE,
  L = 200,
  tnew = NULL,
  spatial = NULL,
  num_threads = 1,
  verbose = TRUE
)
```

Arguments

Y	Response data (3D array or cube).
G	System matrix (cube).
P	Observation matrix (cube).
D	Spatial distance matrix.
grid	List with elements: • tau: numeric vector • phi: numeric vector
prior	Prior list for forward filter (m, C, nu, Psi)
do_BS	Logical: run backward sampling? (default: FALSE)
do_forecast	Logical: run temporal forecasts? (default: FALSE)
do_spatial	Logical: run spatial interpolation? (default: FALSE)
L	Number of posterior samples (default 200)
tnew	Forecast horizon (default 5)
spatial	Optional list for spatial: list(crd = , crdtilde = , Xtilde =)
num_threads	Number of cores for parallel computing (default: 1)
verbose	Logical; print progress messages to the console? (default: TRUE)

Value

A list with the components executed according to the flags.

Examples

```
n <- 50
t <- 5
p <- 2
q <- 2

Y <- array(rnorm(n*q*t), dim = c(n, q, t))
P <- array(rnorm(n*(p+n)*t), dim = c(n, (p+n), t))
G <- array(rnorm((p+n)*(p+n)*t), dim = c((p+n), (p+n), t))
coords <- matrix(runif(n*2), ncol = 2)
D <- as.matrix(dist(coords))

priors <- list("m" = matrix(0, n+p, q),
               "C" = diag(p+n),
               "nu" = 3,
               "Psi" = diag(q))

hyperpar <- list(tau = 0.5, phi = 1)

res <- spFFBS(Y = Y, G = G, P = P, D = D, grid = hyperpar, prior = priors)
```

Index