Package ‘spatialAtomizeR’

December 10, 2025

Type Package

Title Spatial Analysis with Misaligned Data Using Atom-Based Regression Models

Version 0.2.3

Date 2025-11-28

Description Implements atom-based regression models (ABRM) for analyzing spatially misaligned data. Provides functions for simulating misaligned spatial data, preparing NIMBLE model inputs, running MCMC diagnostics, and comparing different spatial analysis methods including dasymetric mapping. All main functions return S3 objects with print(), summary(), and plot() methods for intuitive result exploration. Methods are described in Nethery et al. (2023) <doi:10.1101/2023.01.10.23284410>. Further methodological details and software implementation are described in Qian et al. (in review).

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.0.0)

Imports sp, sf, spdep, MASS, raster, dplyr, tidyr, ggplot2, reshape2, nimble, coda, BiasedUrn, stats, utils, grDevices, methods

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

URL `https://github.com/bellayqian/spatialAtomizeR`

BugReports `https://github.com/bellayqian/spatialAtomizeR/issues`

RoxygenNote 7.3.3

NeedsCompilation no

Author Yunzhe Qian [aut, cre],
Rachel Nethery [aut],
Nancy Krieger [ctb] (Contributed to the project conceptualization and manuscript),
Nykesha Johnson [ctb] (Contributed to the project conceptualization and manuscript)

Maintainer Yunzhe Qian <`qyzanemos@gmail.com`>

Repository CRAN

Date/Publication 2025-12-10 21:10:07 UTC

Contents

`biasedUrn_rmfnc` *R Wrapper Function for BiasedUrn Sampling*

Description

Wraps the BiasedUrn::rMFNCHypergeo function for use in NIMBLE models

Usage

```
biasedUrn_rmfnc(total, odds, ni)
```

Arguments

`total`	Integer, total number of items to sample
`odds`	Numeric vector of odds for each category
`ni`	Integer vector of population sizes

Value

Numeric vector of sampled counts

`check_mcmc_diagnostics` *Check MCMC Diagnostics*

Description

Calculates convergence diagnostics including Gelman-Rubin statistics and effective sample sizes for MCMC output

Usage

```
check_mcmc_diagnostics(mcmc_output, sim_metadata = NULL)
```

Arguments

`mcmc_output`	Output from NIMBLE MCMC containing samples
`sim_metadata`	Optional list with simulation metadata

Value

List with diagnostic results including convergence statistics and plots

---

`create_comparison_plots`

Create Comparison Plots

Description

Create Comparison Plots

Usage

```
create_comparison_plots(comparison_data, output_dir, true_params = NULL)
```

Arguments

`comparison_data`	Data frame with comparison results
`output_dir`	Output directory for plots
`true_params`	True parameter values (optional)

Value

No return value, called for side effects. Creates and displays/saves comparison plots of coefficient estimates, bias, and coverage rates across different methods. If output_dir is provided, plots are saved to a PDF file.

`create_diagnostic_plots`

Create Diagnostic Plots

Description

Create Diagnostic Plots

Usage

```
create_diagnostic_plots(chains_list, sim_metadata)
```

Arguments

`chains_list`	List of MCMC chains
`sim_metadata`	Simulation metadata

Value

List with trace and density plots

`create_sensitivity_summary_plots` *Create Sensitivity Summary Plots*

Description

Create Sensitivity Summary Plots

Usage

```
create_sensitivity_summary_plots(combined_results, output_dir)
```

Arguments

`combined_results`	Combined results data frame
`output_dir`	Output directory

Value

No return value, called for side effects. Creates and displays/saves summary plots for sensitivity analysis including correlation effects, bias patterns, and coverage rates. If output_dir is provided, plots are saved to a PDF file.

`create_summary_statistics` *Create Summary Statistics*

Description

Create Summary Statistics

Usage

```
create_summary_statistics(all_results, output_dir)
```

Arguments

`all_results`	Combined results data frame
`output_dir`	Output directory

Value

A list with two components: `method_summary` (data frame containing overall summary statistics for each method, including mean bias, relative bias, and coverage rates) and `param_summary` (data frame containing parameter-specific comparisons across methods). The function also saves these summaries to CSV files in the output directory.

`dasymetric_mapping` *Dasymetric Mapping*

Description

Maps X-grid covariates to Y-grid using centroid-based spatial join

Usage

```
dasymetric_mapping(misaligned_data)
```

Arguments

`misaligned_data`	List with gridx and gridy from simulate_misaligned_data

Value

sf object with Y grid containing mapped X covariates

`fit_dasymetric_model` *Fit Dasymetric Model*

Description

Fits regression model to dasymetrically mapped data

Usage

```
fit_dasymetric_model(mapped_data, outcome_type)
```

Arguments

`mapped_data`	Output from dasymetric_mapping
`outcome_type`	Distribution type: 'normal', 'poisson', or 'binomial'

Value

Data frame with parameter estimates and confidence intervals

`gen_correlated_spat` *Generate Correlated Spatial Effects*

Description

Generate Correlated Spatial Effects

Usage

```
gen_correlated_spat(
  W,
  n_vars,
  rho = 0.6,
  var_spat = 1,
  correlation = 0.5,
  verify = FALSE
)
```

Arguments

`W`	Spatial adjacency matrix
`n_vars`	Number of variables
`rho`	Spatial correlation parameter (default = 0.6)
`var_spat`	Spatial variance (default = 1)
`correlation`	Correlation between variables (default = 0.5)
`verify`	Logical for verification (default = FALSE)

Value

Matrix of spatial effects

`get_abrm_model` *Get ABRM Model Code for NIMBLE*

Description

Returns the NIMBLE code for the Atom-Based Regression Model with mixed-type variables. Automatically registers custom distributions if not already registered.

Usage

```
get_abrm_model()
```

Value

A nimbleCode object containing the model specification

`plot.abrm` *Plot method for abrm objects*

Description

Plot method for abrm objects

Usage

```
## S3 method for class 'abrm'
plot(x, ...)
```

Arguments

`x`	An object of class "abrm"
`...`	Additional arguments (ignored)

Value

Invisibly returns the input object `x`. The function is called for its side effect of displaying MCMC diagnostic plots (trace plots and density plots) if they are available in the abrm object.

`plot.abrm_comparison` *Plot method for abrm_comparison objects*

Description

Plot method for abrm_comparison objects

Usage

```
## S3 method for class 'abrm_comparison'
plot(x, ...)
```

Arguments

`x`	An object of class "abrm_comparison"
`...`	Additional arguments (ignored)

Value

Invisibly returns the input object `x`. The function is called for its side effect of displaying comparison plots between methods if true parameters are available in the comparison object.

`prepare_adjacency_matrices` *Prepare Adjacency Matrices*

Description

Prepare Adjacency Matrices

Usage

```
prepare_adjacency_matrices(gridy_yorder, gridx_xorder)
```

Arguments

`gridy_yorder`	Reordered Y grid
`gridx_xorder`	Reordered X grid

Value

List containing W_x and W_y adjacency matrices

`prepare_nimble_inputs` *Prepare NIMBLE Model Inputs*

Description

Prepare NIMBLE Model Inputs

Usage

```
prepare_nimble_inputs(
  bookkeeping,
  adjacency,
  data,
  norm_idx_x = NULL,
  pois_idx_x = NULL,
  binom_idx_x = NULL,
  norm_idx_y = NULL,
  pois_idx_y = NULL,
  binom_idx_y = NULL,
  dist_y = 2
)
```

Arguments

`bookkeeping`	Output from prepare_spatial_bookkeeping
`adjacency`	Output from prepare_adjacency_matrices
`data`	Original simulated data
`norm_idx_x`	Indices of normal-distributed X covariates
`pois_idx_x`	Indices of Poisson-distributed X covariates
`binom_idx_x`	Indices of binomial-distributed X covariates
`norm_idx_y`	Indices of normal-distributed Y covariates
`pois_idx_y`	Indices of Poisson-distributed Y covariates
`binom_idx_y`	Indices of binomial-distributed Y covariates
`dist_y`	Distribution type for outcome (1=normal, 2=poisson, 3=binomial)

Value

List containing constants, data, and inits for NIMBLE

`prepare_spatial_bookkeeping`

Prepare Spatial Bookkeeping

Description

Prepare Spatial Bookkeeping

Usage

```
prepare_spatial_bookkeeping(data)
```

Arguments

`data`	List containing gridy, gridx, and atoms from simulate_misaligned_data

Value

List of bookkeeping objects for NIMBLE model

`print.abrm` *Print method for abrm objects*

Description

Print method for abrm objects

Usage

```
## S3 method for class 'abrm'
print(x, ...)
```

Arguments

`x`	An abrm object
`...`	Additional arguments (unused)

Value

Invisibly returns the input object `x`. The function is called for its side effect of printing a summary of the ABRM model results including convergence status, number of parameters estimated, and key fit statistics.

`print.abrm_comparison` *Print method for abrm_comparison objects*

Description

Print method for abrm_comparison objects

Usage

```
## S3 method for class 'abrm_comparison'
print(x, ...)
```

Arguments

`x`	A abrm_comparison object
`...`	Additional arguments (unused)

Value

Invisibly returns the input object `x`. The function is called for its side effect of printing a summary of method comparison results including the number of simulations and methods compared.

`print.misaligned_data` *Print method for misaligned_data objects*

Description

Print method for misaligned_data objects

Usage

```
## S3 method for class 'misaligned_data'
print(x, ...)
```

Arguments

`x`	A misaligned_data object
`...`	Additional arguments (unused)

Value

Invisibly returns the input object `x`. The function is called for its side effect of printing a summary of the simulated misaligned spatial data including grid dimensions and number of atoms.

`print.sensitivity_analysis` *Print method for sensitivity_analysis objects*

Description

Print method for sensitivity_analysis objects

Usage

```
## S3 method for class 'sensitivity_analysis'
print(x, ...)
```

Arguments

`x`	A sensitivity_analysis object
`...`	Additional arguments (unused)

Value

Invisibly returns the input object `x`. The function is called for its side effect of printing a summary of the sensitivity analysis results including the number of simulations and scenarios tested.

`print_convergence_summary`

Print Convergence Summary

Description

Print Convergence Summary

Usage

```
print_convergence_summary(convergence_results)
```

Arguments

`convergence_results`	Results from check_mcmc_diagnostics

Value

No return value, called for side effects. Prints convergence diagnostics including overall convergence status, minimum effective sample size, median ESS, maximum Rhat statistic, and identifies parameters with high relative variance.

`register_nimble_distributions`

Register Custom NIMBLE Distributions

Description

Registers the custom multivariate non-central hypergeometric distribution for use in NIMBLE models. This function is called automatically when needed.

Usage

```
register_nimble_distributions()
```

Value

Invisible TRUE if successful

Note

The «- operator is used intentionally to create package-level nimbleFunctions accessible across the package environment.

`run_abrm` *Run ABRM Analysis*

Description

Runs the Atom-Based Regression Model on simulated data

Usage

```
run_abrm(
  gridx,
  gridy,
  atoms,
  model_code,
  true_params = NULL,
  norm_idx_x = NULL,
  pois_idx_x = NULL,
  binom_idx_x = NULL,
  norm_idx_y = NULL,
  pois_idx_y = NULL,
  binom_idx_y = NULL,
  dist_y = 2,
  niter = 50000,
  nburnin = 30000,
  nchains = 2,
  thin = 10,
  sim_metadata = NULL,
  save_plots = TRUE,
  output_dir = NULL
)
```

Arguments

`gridx`	The X-grid sf dataframe, containing a numeric area ID variable named 'ID' and covariates named 'covariate_x_1','covariate_x_2',...
`gridy`	The Y-grid sf dataframe, containing a numeric area ID variable named 'ID', covariates named 'covariate_y_1','covariate_y_2',...., and an outcome named 'y'.
`atoms`	The atom sf dataframe, which should contain numeric variables named 'ID_x' and 'ID_y' holding the X-grid and Y-grid cell IDs for each atom, as well as an atom-level population count named 'population'.
`model_code`	NIMBLE model code from get_abrm_model()
`true_params`	The true outcome model regression coefficient parameters, if known (e.g., from simulate_misaligned_data())
`norm_idx_x`	Vector of numeric indices of X-grid covariates (ordered as 'covariate_x_1','covariate_x_2',...) that should be treated as normally-distributed

`pois_idx_x`	Vector of numeric indices of X-grid covariates (ordered as 'covariate_x_1','covariate_x_2',...) that should be treated as Poisson-distributed
`binom_idx_x`	Vector of numeric indices of X-grid covariates (ordered as 'covariate_x_1','covariate_x_2',...) that should be treated as binomial-distributed
`norm_idx_y`	Vector of numeric indices of Y-grid covariates (ordered as 'covariate_y_1','covariate_y_2',...) that should be treated as normally-distributed
`pois_idx_y`	Vector of numeric indices of Y-grid covariates (ordered as 'covariate_y_1','covariate_y_2',...) that should be treated as Poisson-distributed
`binom_idx_y`	Vector of numeric indices of Y-grid covariates (ordered as 'covariate_y_1','covariate_y_2',...) that should be treated as binomial-distributed
`dist_y`	Distribution type for outcome (1=normal, 2=poisson, 3=binomial)
`niter`	Number of MCMC iterations (default: 50000)
`nburnin`	Number of burn-in iterations (default: 30000)
`nchains`	Number of MCMC chains (default: 2)
`thin`	Thinning interval (default: 10)
`sim_metadata`	Optional simulation metadata list
`save_plots`	Logical, whether to save diagnostic plots (default: TRUE)
`output_dir`	Directory for saving outputs (default: NULL)

Value

List containing MCMC results and parameter estimates

`run_both_methods` *Run Both Methods and Compare*

Description

Runs both ABRM and dasymetric mapping methods and compares results

Usage

```
run_both_methods(
  sim_data,
  sim_metadata,
  model_code,
  nimble_params,
  output_dir,
  norm_idx_x,
  pois_idx_x,
  binom_idx_x,
  norm_idx_y,
  pois_idx_y,
```

```
  binom_idx_y,
  dist_y,
  outcome_type
)
```

Arguments

`sim_data`	List of data elements to be used in the ABRM, structured like the output from the simulate_misaligned_data() function. The first element of this list is the Y-grid sf dataframe (named 'gridy'), containing a numeric area ID variable named 'ID_y', covariates named 'covariate_y_1','covariate_y_2',...., and an outcome named 'y'. The second element of this list is the X-grid sf dataframe (named 'gridx'), containing a numeric area ID variable named 'ID_x' and covariates named 'covariate_x_1','covariate_x_2',... The third element of the list is the atom sf dataframe (named 'atoms'), which should contain variables named 'ID_x' and 'ID_y' holding the X-grid and Y-grid cell IDs for each atom, as well as an atom-level population count named 'population'.
`sim_metadata`	Simulation metadata
`model_code`	NIMBLE model code
`nimble_params`	List of NIMBLE parameters (niter, nburnin, thin, nchains)
`output_dir`	Output directory
`norm_idx_x`	Indices of normal X covariates
`pois_idx_x`	Indices of Poisson X covariates
`binom_idx_x`	Indices of binomial X covariates
`norm_idx_y`	Indices of normal Y covariates
`pois_idx_y`	Indices of Poisson Y covariates
`binom_idx_y`	Indices of binomial Y covariates
`dist_y`	Distribution type for outcome (1=normal, 2=poisson, 3=binomial)
`outcome_type`	Outcome distribution name

Value

List with combined comparison, ABRM results, and dasymetric results

---

`run_nimble_model` *Run NIMBLE Model with Diagnostics*

Description

Run NIMBLE Model with Diagnostics

Usage

```
run_nimble_model(
  constants,
  data,
  inits,
  sim_metadata = NULL,
  model_code,
  niter = 50000,
  nburnin = 30000,
  nchains = 2,
  thin = 10,
  save_plots = TRUE,
  output_dir = NULL
)
```

Arguments

`constants`	List of model constants
`data`	List of data
`inits`	List of initial values
`sim_metadata`	List with simulation metadata (optional)
`model_code`	NIMBLE code object
`niter`	Number of MCMC iterations (default: 50000)
`nburnin`	Number of burn-in iterations (default: 30000)
`nchains`	Number of MCMC chains (default: 2)
`thin`	Thinning interval (default: 10)
`save_plots`	Logical, whether to save diagnostic plots (default: TRUE)
`output_dir`	Directory for saving plots (default: NULL)

Value

List containing MCMC samples, summary, and convergence diagnostics

---

`run_sensitivity_analysis` *Run Sensitivity Analysis*

Description

Performs sensitivity analysis across different correlation structures

Usage

```
run_sensitivity_analysis(
  correlation_grid = c(0.2, 0.6),
  n_sims_per_setting = 3,
  base_params = list(dist_covariates_x = c("normal", "poisson", "binomial"),
    dist_covariates_y = c("normal", "poisson", "binomial"), dist_y = "poisson",
   x_intercepts = c(4, -1, -1), y_intercepts = c(4, -1, -1), beta0_y = -1, beta_x =
    c(-0.03, 0.1, -0.2), beta_y = c(0.03, -0.1, 0.2)),
  mcmc_params = list(niter = 50000, nburnin = 30000, thin = 10, nchains = 2),
  model_code,
  base_seed = 123,
  output_dir = NULL
)
```

Arguments

`correlation_grid`	Vector of correlation values to test
`n_sims_per_setting`	Number of simulations per correlation setting
`base_params`	List of base simulation parameters
`mcmc_params`	List of MCMC parameters
`model_code`	NIMBLE model code
`base_seed`	Base random seed
`output_dir`	Output directory for results (default: NULL, uses tempdir())

Value

List with combined results, summary statistics, and output directory

---

`simulate_misaligned_data`

Simulate Misaligned Spatial Data

Description

Simulate Misaligned Spatial Data

Usage

```
simulate_misaligned_data(
  seed = 2,
  dist_covariates_x = c("normal", "poisson", "binomial"),
  dist_covariates_y = c("normal", "poisson", "binomial"),
  dist_y = "poisson",
```

```
  x_intercepts = rep(0, 3),
  y_intercepts = rep(0, 3),
  rho_x = 0.6,
  rho_y = 0.6,
  x_correlation = 0.5,
  y_correlation = 0.5,
  beta0_y = NULL,
  beta_x = NULL,
  beta_y = NULL,
  diff_pops = TRUE,
  xy_cov_cor = FALSE
)
```

Arguments

`seed`	Random seed (default = 2)
`dist_covariates_x`	Vector specifying distribution type for each synthetic X-grid covariate ('poisson', 'binomial', or 'normal')
`dist_covariates_y`	Vector specifying distribution type for each synthetic Y-grid covariate ('poisson', 'binomial', or 'normal')
`dist_y`	Distribution type for synthetic outcome variable (one of 'poisson', 'binomial', or 'normal')
`x_intercepts`	Intercepts for X covariates
`y_intercepts`	Intercepts for Y covariates
`rho_x`	Spatial correlation parameter for X-grid covariates (0 to 1 with higher values yielding more spatial correlation, default = 0.6)
`rho_y`	Spatial correlation parameter for Y-grid covariates and outcome (0 to 1 with higher values yielding more spatial correlation, default = 0.6)
`x_correlation`	Between-variable correlation for all pairs of X-grid covariates (default = 0.5)
`y_correlation`	Between-variable correlation for all pairs of Y-grid covariates (default = 0.5)
`beta0_y`	Intercept for outcome model
`beta_x`	Outcome model coefficients for X-grid covariates
`beta_y`	Outcome model coefficients for Y-grid covariates
`diff_pops`	Logical, indicating whether the atoms should be generated with different population sizes (diff_pops = TRUE) or a common population size (diff_pops = FALSE)
`xy_cov_cor`	Logical, indicating whether the atom-level spatial random effects for X-grid and Y-grid covariates should be correlated (xy_cov_cor = TRUE) or not. When set to TRUE, the x_correlation and rho_x parameters are used to generate all covariates (separate correlation parameters are not allowed for X-grid and Y-grid covariates).

Value

List containing gridy, gridx, atoms, and true_params

`summary.abrm` *Summary method for abrm objects*

Description

Summary method for abrm objects

Usage

```
## S3 method for class 'abrm'
summary(object, ...)
```

Arguments

`object`	An abrm object
`...`	Additional arguments (unused)

Value

Invisibly returns the input object `object`. The function is called for its side effect of printing the ABRM model summary including detailed parameter estimates.

`summary.abrm_comparison` *Summary method for abrm_comparison objects*

Description

Summary method for abrm_comparison objects

Usage

```
## S3 method for class 'abrm_comparison'
summary(object, ...)
```

Arguments

`object`	An object of class "abrm_comparison"
`...`	Additional arguments (ignored)

Value

Invisibly returns the input object `object`. The function is called for its side effect of printing the method comparison summary including combined comparison results across all simulations.

`summary.misaligned_data` *Summary method for misaligned_data objects*

Description

Summary method for misaligned_data objects

Usage

```
## S3 method for class 'misaligned_data'
summary(object, ...)
```

Arguments

`object`	A misaligned_data object
`...`	Additional arguments (unused)

Value

Invisibly returns the input object `object`. The function is called for its side effect of printing the misaligned data summary including grid information and true parameter values (beta_x and beta_y).

`summary.sensitivity_analysis` *Summary method for sensitivity_analysis objects*

Description

Summary method for sensitivity_analysis objects

Usage

```
## S3 method for class 'sensitivity_analysis'
summary(object, ...)
```

Arguments

`object`	A sensitivity_analysis object
`...`	Additional arguments (unused)

Value

Invisibly returns the input object `object`. The function is called for its side effect of printing the sensitivity analysis summary including comprehensive results across all simulation scenarios.

Index