

Package ‘DataFusionGDM’

May 7, 2026

Type Package

Title Machine Learning for Integrating Partially Overlapped Genetic Datasets

Version 1.3.2

Description Tools to simulate genetic distance matrices, align and compare them via multidimensional scaling (MDS) and Procrustes, and evaluate imputation with the Bootstrapping Evaluation for Structural Missingness Imputation (BESMI) framework. Methods align with Zhu et al. (2025) <[doi:10.3389/fpls.2025.1543956](https://doi.org/10.3389/fpls.2025.1543956)> and the associated software resource Zhu (2025) <[doi:10.26188/28602953](https://doi.org/10.26188/28602953)>.

License GPL-3

URL <https://github.com/jiashuaiz/DataFusion-GDM>

BugReports <https://github.com/jiashuaiz/DataFusion-GDM/issues>

Encoding UTF-8

Depends R (>= 3.6)

Imports ggplot2, vegan, mice, stats, utils

Suggests VIM, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.3.3

NeedsCompilation no

Config/testthat/edition 3

Author Jiashuai Zhu [aut, cre] (ORCID: <<https://orcid.org/0000-0002-9916-9732>>, affiliation: Faculty of Science, The University of Melbourne, Parkville, VIC, Australia; Agriculture Victoria, AgriBio Centre, Bundoora, VIC, Australia), The University of Melbourne [cph], Agriculture Victoria [cph]

Maintainer Jiashuai Zhu <jiashuai.zhu@student.unimelb.edu.au>

Repository CRAN

Date/Publication 2025-11-04 19:20:07 UTC

Contents

apply_procrustes	2
besmi_batch_impute	3
besmi_create_masked_matrices	4
besmi_impute_single_dataset	4
besmi_iterative_imputation	5
besmi_knn_impute	6
besmi_prepare_full_dataset	7
coords_to_distances	7
create_distance_heatmap	8
create_mds_plot	8
export_simulated_gdm	9
perform_mds	9
run_genetic_scenario	10
run_genetic_simulation	11
simulate_genetic_distances	12
visualize_results	14
Index	15

apply_procrustes	<i>Procrustes alignment and mapping back to distances</i>
------------------	---

Description

Procrustes alignment and mapping back to distances

Usage

```
apply_procrustes(X_base, Y_base, Y)
```

Arguments

X_base	Base coordinates for target alignment
Y_base	Base coordinates for source alignment
Y	Full source coordinates to transform

Value

Transformed coordinates matrix

besmi_batch_impute *Run BESMI imputation for a list of dataset paths*

Description

Run BESMI imputation for a list of dataset paths

Usage

```
besmi_batch_impute(  
  dataset_paths,  
  the_method = "lasso.norm",  
  max_iter = 5,  
  imputation_convergence_threshold = 1e-06,  
  propagation_convergence_threshold = 1e-06,  
  distance_metric = "mae",  
  output_dir = file.path(tempdir(), "DataFusionGDM_imputation"),  
  k_filter = NULL,  
  full_dataset_path = NULL  
)
```

Arguments

dataset_paths	Character vector of RDS paths to masked matrices
the_method	Imputation method (e.g., 'lasso.norm' or 'KNN')
max_iter	Maximum iterations for iterative methods
imputation_convergence_threshold	Convergence threshold for imputation metric
propagation_convergence_threshold	Convergence threshold for propagation metric
distance_metric	Distance metric for evaluation ('mae', 'ssd', 'rmse', 'correlation')
output_dir	Output directory for imputed matrices (defaults to a temporary location)
k_filter	Optional numeric filter for k value
full_dataset_path	Optional path to a full matrix RDS used as ground truth

Value

Data frame of metrics for all datasets

besmi_create_masked_matrices

Create masked matrices for BESMI

Description

Create masked matrices for BESMI

Usage

```
besmi_create_masked_matrices(full_matrix, k, seed = NULL)
```

Arguments

full_matrix	Full symmetric matrix
k	Number of populations to mask (as U)
seed	Optional seed for reproducibility

Value

List with masked_matrix, mask_position, group_u, group_s, masked_percentage

besmi_impute_single_dataset

Impute a single dataset from masked matrix path

Description

Impute a single dataset from masked matrix path

Usage

```
besmi_impute_single_dataset(
  input_path,
  method = "lasso.norm",
  max_iterations = 5,
  imputation_convergence_threshold = 0.001,
  propagation_convergence_threshold = 0.001,
  distance_metric = "mae",
  output_dir = file.path(tempdir(), "DataFusionGDM_imputation"),
  full_dataset_path = NULL
)
```

Arguments

input_path	Path to masked matrix RDS
method	Imputation method ('lasso.norm' or 'KNN')
max_iterations	Maximum iterations for iterative methods
imputation_convergence_threshold	Convergence threshold for imputation metric
propagation_convergence_threshold	Convergence threshold for propagation metric
distance_metric	Distance metric name
output_dir	Output directory for results (defaults to a temporary location)
full_dataset_path	Optional path to a full matrix RDS used as ground truth

Value

Data frame of per-iteration metrics

besmi_iterative_imputation
Iterative imputation with MICE (tails-chain)

Description

Iterative imputation with MICE (tails-chain)

Usage

```
besmi_iterative_imputation(  
  M_input,  
  M_mask,  
  M_real = NULL,  
  method = "lasso.norm",  
  max_iterations = 5,  
  imputation_convergence_threshold = 0.001,  
  propagation_convergence_threshold = 0.001,  
  distance_metric = "mae",  
  k = NA,  
  bs_i = NA  
)
```

Arguments

M_input	Matrix with NAs to impute
M_mask	Logical mask matrix (TRUE indicates masked positions)
M_real	Optional ground truth matrix
method	MICE method (e.g., 'lasso.norm')
max_iterations	Max outer iterations
imputation_convergence_threshold	Threshold for imputation distance
propagation_convergence_threshold	Threshold for propagation distance
distance_metric	Distance metric name
k	Dataset parameter k (for logging)
bs_i	Bootstrap index (for logging)

Value

List with final_matrix, metrics, tails_chain

besmi_knn_impute	<i>KNN imputation sweep (uses VIM::kNN)</i>
------------------	---

Description

KNN imputation sweep (uses VIM::kNN)

Usage

```
besmi_knn_impute(
  M_input,
  M_mask,
  M_real = NULL,
  distance_metric = "mae",
  k = NA,
  bs_i = NA
)
```

Arguments

M_input	Matrix with NAs
M_mask	Logical mask matrix
M_real	Optional ground truth
distance_metric	Distance metric name
k	Dataset parameter k
bs_i	Bootstrap index

Value

List with final_matrix, metrics, tails_chain

besmi_prepare_full_dataset

Prepare full GDM dataset from CSV or RData

Description

Prepare full GDM dataset from CSV or RData

Usage

besmi_prepare_full_dataset(input_path)

Arguments

input_path Path to CSV or RData file containing the full distance matrix

Value

Symmetric numeric matrix

coords_to_distances

Convert coordinate matrix to distance matrix

Description

Convert coordinate matrix to distance matrix

Usage

coords_to_distances(coords)

Arguments

coords Numeric coordinate matrix

Value

Symmetric distance matrix

`create_distance_heatmap`*Create a heatmap of genetic distances (ggplot2)*

Description

Returns a ggplot heatmap of the distance matrix using ggplot2 only (no Bioconductor dependencies).

Usage

```
create_distance_heatmap(dist_matrix, pop_info)
```

Arguments

<code>dist_matrix</code>	Symmetric numeric distance matrix with row/column names
<code>pop_info</code>	Data frame with at least Population and MajorGroup columns

Value

A ggplot object

`create_mds_plot`*Create MDS plot of genetic distances*

Description

Create MDS plot of genetic distances

Usage

```
create_mds_plot(dist_matrix, pop_info)
```

Arguments

<code>dist_matrix</code>	Symmetric numeric distance matrix
<code>pop_info</code>	Data frame with metadata columns

Value

A ggplot object

export_simulated_gdm *Export a simulated GDM to CSV*

Description

Export a simulated GDM to CSV

Usage

```
export_simulated_gdm(  
  output_file = tempfile("gdm_", fileext = ".csv"),  
  scenario = "default",  
  n_pops = 30,  
  verbose = TRUE,  
  seed = NULL  
)
```

Arguments

output_file	Output CSV filename (defaults to a session-scoped temporary path)
scenario	Scenario name
n_pops	Number of populations
verbose	Verbose output
seed	Optional seed forwarded to run_genetic_scenario()

Value

Invisibly, the normalized path to the written CSV

Examples

```
tmp <- export_simulated_gdm(verbose = FALSE)  
if (file.exists(tmp)) unlink(tmp)
```

perform_mds *Perform MDS on a pair of distance matrices*

Description

Perform MDS on a pair of distance matrices

Usage

```
perform_mds(A, B)
```

Arguments

A	First distance matrix
B	Second distance matrix

Value

A list with coordinates X, Y, optimal dimension d_opt, and variance info

run_genetic_scenario *Run simulation with predefined biological scenarios*

Description

Run simulation with predefined biological scenarios

Usage

```
run_genetic_scenario(
  scenario = "default",
  n_pops = 30,
  output_file = NULL,
  seed = NULL,
  verbose = TRUE
)
```

Arguments

scenario	Scenario name: 'default', 'island', 'stepping_stone', 'admixture', 'ancient_divergence', 'simple'
n_pops	Number of populations
output_file	Optional CSV path to write the distance matrix
seed	Optional seed forwarded to run_genetic_simulation()
verbose	Print diagnostic information

Value

Same structure as run_genetic_simulation()

`run_genetic_simulation`*Run a high-level genetic simulation with configurable model*

Description

Run a high-level genetic simulation with configurable model

Usage

```
run_genetic_simulation(  
  n_pops = 30,  
  n_major_groups = 4,  
  n_subgroups = 8,  
  model = "mixed",  
  geo_dims = NULL,  
  isolation_factor = NULL,  
  genetic_dims = NULL,  
  group_separation = 15,  
  subgroup_separation = NULL,  
  pop_dispersion = 0.5,  
  admixture_prob = 0.15,  
  bottleneck_prob = 0.1,  
  use_subgroups = TRUE,  
  use_genetic_dims = NULL,  
  use_admixture = TRUE,  
  use_bottlenecks = TRUE,  
  use_isolation_by_distance = NULL,  
  use_nonlinear = TRUE,  
  use_noise = TRUE,  
  seed = NULL,  
  output_file = NULL,  
  verbose = TRUE  
)
```

Arguments

<code>n_pops</code>	Number of populations
<code>n_major_groups</code>	Number of major groups
<code>n_subgroups</code>	Number of subgroups
<code>model</code>	One of "mixed", "geography", "genetics", or "custom"
<code>geo_dims</code>	Geographic dimensions (overrides default based on model if set)
<code>isolation_factor</code>	Geography-genetics balance (overrides default based on model if set)
<code>genetic_dims</code>	Genetic dimensions (overrides default based on model if set)

group_separation	Separation between major groups
subgroup_separation	Separation between subgroups (default: group_separation/3 when NULL)
pop_dispersion	Within-subgroup dispersion
admixture_prob	Proportion of admixed populations
bottleneck_prob	Proportion of bottlenecked populations
use_subgroups	Whether to create subgroups
use_genetic_dims	Whether to include genetic dimensions
use_admixture	Whether to include admixture
use_bottlenecks	Whether to include bottlenecks
use_isolation_by_distance	Whether to weight geographic distance
use_nonlinear	Whether to apply nonlinear transformation
use_noise	Whether to add noise
seed	Optional seed forwarded to simulate_genetic_distances()
output_file	Optional CSV file path to write the distance matrix
verbose	Print diagnostics

Value

List with results and plots (functions to print plots)

simulate_genetic_distances

Simulate genetic distances using realistic population structure

Description

Generates a synthetic genetic distance matrix and metadata using hierarchical population structure, admixture and bottleneck options.

Usage

```
simulate_genetic_distances(
  n_pops = 50,
  n_major_groups = 5,
  n_subgroups = 12,
  geo_dims = 2,
  genetic_dims = 2,
  group_separation = 15,
```

```

    subgroup_separation = 5,
    pop_dispersion = 0.5,
    isolation_factor = 0.8,
    admixture_prob = 0.1,
    bottleneck_prob = 0.05,
    noise_level = 0.1,
    nonlinear_factor = 0.7,
    use_subgroups = TRUE,
    use_genetic_dims = TRUE,
    use_admixture = TRUE,
    use_bottlenecks = TRUE,
    use_isolation_by_distance = TRUE,
    use_nonlinear = TRUE,
    use_noise = TRUE,
    seed = NULL,
    verbose = TRUE
)

```

Arguments

n_pops	Number of populations
n_major_groups	Number of major groups
n_subgroups	Number of subgroups
geo_dims	Geographic dimensions
genetic_dims	Additional genetic drift dimensions
group_separation	Separation between major groups
subgroup_separation	Separation between subgroups
pop_dispersion	Within-subgroup dispersion
isolation_factor	Weight for geography in isolation-by-distance model (0-1)
admixture_prob	Proportion of admixed populations
bottleneck_prob	Proportion of bottlenecked populations
noise_level	Noise level in transformation
nonlinear_factor	Nonlinearity factor in transformation
use_subgroups	Whether to create subgroups
use_genetic_dims	Whether to include genetic dimensions
use_admixture	Whether to include admixture
use_bottlenecks	Whether to include bottlenecks

use_isolation_by_distance	Whether to weight geographic distance
use_nonlinear	Whether to apply nonlinear transformation
use_noise	Whether to add noise
seed	Optional seed for reproducibility (NULL leaves the RNG state unchanged)
verbose	Print diagnostics

Value

A list with `distance_matrix`, `population_info`, `position_matrix`, and `parameters`.

visualize_results	<i>Create plotting handles for simulation results</i>
-------------------	---

Description

Create plotting handles for simulation results

Usage

```
visualize_results(sim_results)
```

Arguments

`sim_results` A list returned by `simulate_genetic_distances()` or `run_genetic_simulation()`

Value

A list with `heatmap` and `mds` functions that print plots when called

Index

[apply_procrustes](#), 2

[besmi_batch_impute](#), 3

[besmi_create_masked_matrices](#), 4

[besmi_impute_single_dataset](#), 4

[besmi_iterative_imputation](#), 5

[besmi_knn_impute](#), 6

[besmi_prepare_full_dataset](#), 7

[coords_to_distances](#), 7

[create_distance_heatmap](#), 8

[create_mds_plot](#), 8

[export_simulated_gdm](#), 9

[perform_mds](#), 9

[run_genetic_scenario](#), 10

[run_genetic_simulation](#), 11

[simulate_genetic_distances](#), 12

[visualize_results](#), 14