Package 'harf'

September 28, 2026

Title Adversarial Random Forests for Omics Synthesis

Version 0.1.0

Description We extend Adversarial Random Forests to a high-dimensional framework. The method partitions the feature space into regions where the assumption of feature independence within tree leaves is more likely to hold. Region-specific adversarial random forest models are trained to capture local dependence structures, while an additional adversarial random forest is fitted to a meta-space representation to model dependencies between regions. New observations are generated by first sampling from the meta-space model and then conditionally sampling from each region-specific model. The proposed methodology is described in Fouodo et al. (2026) <doi:10.64898/2026.09.09.750490>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Imports arf, data.table, stats, ClusterR, matrixStats, pracma, pls, fastPLS, RGCCA, ranger, rsvd, foreach

Suggests testthat (>= 3.0.0), knitr, rmarkdown, checkmate, Rtsne, SingleCellExperiment, corrplot, scater, cowplot, ggplot2, doParallel, pROC, caret

Config/testthat/edition 3

Depends R (>= 3.6.0)

Collate 'single_cell.R' 'kich.R' 'utils.R' 'h_arf.R' 'h_forge.R'

VignetteBuilder knitr, rmarkdown

BugReports https://github.com/bips-hb/harf/issues

LazyData true

URL https://bips-hb.github.io/harf/

NeedsCompilation no

Author Cesaire J. K. Fouodo [aut, cre],
Jan Kapar [aut],
Marvin N. Wright [aut]

Maintainer Cesaire J. K. Fouodo <fouodo@leibniz-bips.de>

Repository CRAN

Date/Publication 2026-09-28 09:00:18 UTC

Contents

`elbow_detection` *ELBOW DETECTION*

Description

ELBOW DETECTION

Usage

```
elbow_detection(scores)
```

Arguments

`scores`	vector

`fast.pca` *PCA built on fast.rsvd*

Description

PCA built on fast.rsvd

Usage

```
fast.pca(X, K = 50)
```

Arguments

`X`	matrix
`K`	number of principal components

Value

projection of X on principal top principal components

`fast.rsvd` *RANDOM PROJECTION SVD*

Description

RANDOM PROJECTION SVD

Usage

```
fast.rsvd(A, K)
```

Arguments

`A`	matrix
`K`	rank

Value

list with components U S and V

`h_arf` *Adversarial random forests for omics synthesis*

Description

This function extends the adversarial random forest (ARF) algorithm to high-dimensional settings. It partitions high-dimensional data into isolated regions and fits ARF models within each region and on a latent space representing region joint distribution to capture within and between region feature dependencies.

Usage

```
h_arf(
  omx_data,
  cli_lab_data = NULL,
  target = NULL,
  feature_ordering = NULL,
  omx_onset_data = NULL,
  num_trees = 10,
  min_node_size = 5,
  correlation_method = "spearman",
  correlation_mat = NULL,
```

```
  num_btwn_pcs = 2,
  num_onset_pcs = 2,
  chunk_size = 10,
  oob = FALSE,
  family = "truncnorm",
  finite_bounds = "no",
  alpha = 0,
  epsilon = 0,
  parallel = FALSE,
  export_cor_mat = FALSE,
  verbose = FALSE
)
```

Arguments

omx_data	A data.frame containing omics data where rows represent samples (e.g. patients or cells) and columns represent features (e.g., gene or protein expressions).
cli_lab_data	A data.frame of clinical or laboratory data. For example, this may be a data.frame where rows represent cell types (for single cell data or additional clinical patient information, e.g. disease status, age, sex, BMI, etc).
target	Optional name of the target variable in cli_lab_data for supervised downstream analysis. If NULL, training is conducted for unsupervised downstream analysis.
feature_ordering	Optional vector of feature names specifying the order of features in the synthesized data. If NULL, features are ordered according to their original order in omx_data, followed by order in clin_lab_data.
omx_onset_data	Optional data.frame of conditional onset omics features. This can be used to condition the synthesis on specific omics features (e.g. expression of a gene of interest). If provided, dimension reduction will be performed on these features and the resulting components will be included as additional meta features for training the meta adversarial model.
num_trees	Number of trees to grow in each Adversarial Random Forest (ARF) model.
min_node_size	Minimum number of samples required to split an internal node in the ARF model.
correlation_method	Methods to compute correlation between features. Options include "pearson", "spearman", and "kendall". Default is "spearman".
correlation_mat	Optional pre-computed correlation matrix between features. If provided, this will be used instead of computing correlations from omx_data. This can be usefull for tuning hyperparameters of the clustering step without having to re-compute the correlation matrix each time.
num_btwn_pcs	Number of principal components to use for between cluster variability. Default is 2.
num_onset_pcs	Number of principal components to use for onset features. Default is 2.

chunk_size	Size of feature chunks to process at a time. Default is 10.
oob	Logical indicating whether to use out-of-bag samples for density estimation. Default is FALSE. Also see forde
family	Distribution to use for density estimation of continuous features. See forde for options.
finite_bounds	Impose finite bounds on all continuous variables? If "local", infinite bounds are set to empirical extrema within leaves. If "global", infinite bounds are set to global empirical extrema. if "no" (the default), infinite bounds are left unchanged. See forde for more details.
alpha	Optional pseudocount for Laplace smoothing in density estimation. See forde for more details.
epsilon	Optional slack parameter on empirical bounds. See forde for more details.
parallel	Logical indicating whether to use parallel processing. Default is TRUE.
export_cor_mat	Logical indicating whether to export the correlation matrix used for clustering. Default is FALSE.
verbose	Logical indicating whether to print progress messages. Default is FALSE.

Value

An isoARF object containing the fitted adversarial models, and clustering information.

References

- Watson et al. (2023). Adversarial Random Forests. Proceedings of the International Conference on Machine Learning (PMLR 206). https://proceedings.mlr.press/v206/watson23a.html
- Fouodo et al. (2026). Adversarial random forests for omics synthesis. bioRxiv preprint doi:10.64898/2026.09.09.750490

See Also

h_forge, arf::adversarial_rf, arf::forde

Examples

```
data(single_cell)
harf_model <- h_arf(
  omx_data = single_cell[ , - which(colnames(single_cell)  == "cell_type")],
  cli_lab_data = data.frame(cell_type = single_cell$cell_type)
)
# Unconditional sampling from harf_model
set.seed(123)
synth_single_cell <- h_forge(
 harf_obj = harf_model,
 n_synth = nrow(single_cell)
 )
 # Conditional resampling from harf_model
 set.seed(142)
```

```
lung_single_cell <- h_forge(
    harf_obj = harf_model,
    n_synth = sum(single_cell$cell_type == "lung"),
    evidence = data.frame(cell_type = "lung")
   )
```

---

h_forge *Adversarial random forests for omics synthesis*

Description

This function uses a high-dimensional ARF model to generate synthetic data.

Usage

```
h_forge(
  harf_obj,
  n_synth,
  evidence = NULL,
  omx_onset_data = NULL,
  evidence_row_mode = c("separate", "or"),
  round = TRUE,
  sample_NAs = FALSE,
  nomatch = c("force", "na"),
  verbose = TRUE,
  stepsize = 0,
  parallel = FALSE
)
```

Arguments

harf_obj	A pre-trained harf model.
n_synth	Number of synthetic samples to generate.
evidence	Optional set of conditioning events. This will be further passed to the forde function in each isolated regions. See forde for details.
omx_onset_data	Optional data.frame of conditional onset omics features.
evidence_row_mode	Interpretation of rows in multi-row evidence. See forde for details.
round	Round continuous variables to their respective maximum precision in the real data set? See forde for details.
sample_NAs	Sample NAs respecting the probability for missing values in the original data? See forde for details.
nomatch	What to do if no leaf matches a condition in evidence? Options are to force sampling from a random leaf ("force") or return NA ("na"). The default is "force".

verbose	What to do if no leaf matches a condition in evidence? See forde for details.
stepsize	How many rows of evidence should be handled at each step? See forde for details.
parallel	Compute in parallel? See forde for details.

Value

A data.table containing the generated synthetic omics data.

Author(s)

Césaire Fouodo

References

- Watson et al. (2023). Adversarial Random Forests. Proceedings of the International Conference on Machine Learning (PMLR 206). https://proceedings.mlr.press/v206/watson23a.html
- Fouodo et al. (2026). Adversarial random forests for omics synthesis. bioRxiv preprint doi:10.64898/2026.09.09.750490

See Also

forge for details on the forging process.

Examples

```
data(single_cell)
harf_model <- h_arf(
  omx_data = single_cell[ , - which(colnames(single_cell)  == "cell_type")],
  cli_lab_data = data.frame(cell_type = single_cell$cell_type)
)
# Unconditional sampling from harf_model
set.seed(123)
synth_single_cell <- h_forge(
 harf_obj = harf_model,
 n_synth = nrow(single_cell)
 )
 # Conditional resampling from harf_model
 set.seed(142)
 lung_single_cell <- h_forge(
     harf_obj = harf_model,
     n_synth = sum(single_cell$cell_type == "lung"),
     evidence = data.frame(cell_type = "lung")
    )
```

kich *Example HARF dataset*

Description

A small example of the TCGA-KICH dataset used to demonstrate functions in the harf package.

Usage

```
kich
```

Format

A data frame.

Source

Data derived from The Cancer Genome Atlas Kidney Chromophobe Collection (TCGA-KICH).

single_cell *Example HARF dataset*

Description

A small example of single cell dataset used to demonstrate and test functions in the `harf` package.

Usage

```
data(single_cell)
```

Format

An object of class `data.frame` with 1652 rows and 81 columns.

Source

Data derived from the RNAseqDB repository maintained by Memorial Sloan Kettering Cancer Center (https://github.com/mskcc/RNAseqDB).

Index