Package: GLBFP (via r-universe)

June 17, 2026

Type Package

Title General Linear Blend Frequency Polygon Density Estimation

Version 0.5.2.9000

Description Implements nonparametric density estimation with Averaged Shifted Histogram (ASH), Linear Blend Frequency Polygon (LBFP), and General Linear Blend Frequency Polygon (GLBFP) estimators. The package provides pointwise and grid-based estimation workflows, sparse-prefix grid-count computation, fixed-grid leave-one-out self-support scores, plotting helpers, and plug-in bandwidth selection.

License GPL (>= 3)

URL https://aureliennicosiaulaval.github.io/GLBFP/, https://github.com/AurelienNicosiaULaval/GLBFP

BugReports https://github.com/AurelienNicosiaULaval/GLBFP/issues

Depends R (>= 4.1.0)

Imports ggplot2, plotly

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

VignetteBuilder knitr, rmarkdown

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev

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

Date/Publication 2026-06-17 15:24:36 UTC

RemoteUrl https://github.com/aureliennicosiaulaval/glbfp

RemoteRef HEAD

RemoteSha 9e91c088b0d478ffcee26b99f8c3e0d3ad3fe31e

Contents

GLBFP-package . . . 2
as.data.frame.glbfp_grid . . . 3
ASH . . . 4
ASH_estimate . . . 5
ashua . . . 6
compute_bi_optim . . . 7
compute_Di . . . 8
compute_G_star . . . 9
G_i . . . 10
GLBFP . . . 11
GLBFP_estimate . . . 12
K_mi . . . 14
LBFP . . . 15
LBFP_estimate . . . 16
lowercase_aliases . . . 17
predict.glbfp_fit . . . 18
summary.glbfp_fit . . . 19

Index **20**

`GLBFP-package` *GLBFP: General Linear Blend Frequency Polygon Density Estimation*

Description

`GLBFP` provides one-point and grid-based density estimators based on ASH, LBFP and GLBFP methodology, with sparse-prefix computation, leave-one-out self-support scores, visualization helpers and bandwidth selection utilities.

Details

Main entry points:

- `ASH()`, `LBFP()`, `GLBFP()`
- `ASH_estimate()`, `LBFP_estimate()`, `GLBFP_estimate()`
- `compute_Di()`
- `compute_bi_optim()`

Lowercase aliases such as glbfp() and glbfp_estimate() are also provided for users who prefer lower-snake-case function names.

Author(s)

Maintainer: Aurélien Nicosia <aurelien.nicosia@mat.ulaval.ca>

See Also

Useful links:

- https://aureliennicosiaulaval.github.io/GLBFP/
- https://github.com/AurelienNicosiaULaval/GLBFP
- Report bugs at https://github.com/AurelienNicosiaULaval/GLBFP/issues

---

`as.data.frame.glbfp_grid` *Convert GLBFP objects to data frames*

Description

Convert GLBFP objects to data frames

Usage

```
## S3 method for class 'glbfp_grid'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S3 method for class 'glbfp_di'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A GLBFP grid object or a `compute_Di()` result.
`row.names`	Optional row names.
`optional`	Passed to `base::as.data.frame()`.
`...`	Additional arguments (unused).

Value

A data frame representation of the object.

ASH *Averaged Shifted Histogram (ASH) estimator at a single point*

Description

Computes the ASH density estimate at point `x`.

Usage

```
ASH(
  x,
  data,
  b = compute_bi_optim(data, m = rep(1, ncol(data))),
  m = rep(1, ncol(data)),
  min_vals = apply(data, 2, min),
  max_vals = apply(data, 2, max)
)

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

Arguments

`x`	Object of class `"ASH"`.
`data`	Numeric matrix or data frame of observations (`n x d`).
`b`	Positive numeric vector of bandwidths (length `d`).
`m`	Positive integer vector of shifts (length `d`).
`min_vals`	Numeric vector of lower grid bounds (length `d`).
`max_vals`	Numeric vector of upper grid bounds (length `d`).
`...`	Additional arguments (unused).

Details

`m` controls the number of shifted histograms used in each dimension. Missing and non-finite values are not accepted; remove or impute them before calling the estimator.

Value

A list with class `c("glbfp_fit", "ASH")` containing: `x`, `estimation`, `b`, `m`, `method`, and `dimension`.

Methods (by generic)

- `print(ASH)`: Print method for object of class `"ASH"`.

References

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley. doi:10.1002/9780470316849.

See Also

`ASH_estimate()`, `LBFP()`, `GLBFP()`, `compute_bi_optim()`

Examples

```
x <- c(200, 30)
b <- c(0.5, 0.5)
m <- c(1, 1)
ASH(x, ashua[, -3], b = b, m = m)
```

ASH_estimate *ASH density estimation on a grid*

Description

Computes ASH density estimates on a regular or user-supplied grid.

Usage

```
ASH_estimate(
  data,
  b = compute_bi_optim(data, m = rep(1, ncol(data))),
  m = rep(1, ncol(data)),
  grid_size = 20,
  grid_points = NULL,
  min_vals = apply(data, 2, min),
  max_vals = apply(data, 2, max)
)

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

## S3 method for class 'ASH_estimate'
plot(x, contour = FALSE, ...)
```

Arguments

`data`	Numeric matrix or data frame of observations (`n x d`).
`b`	Positive numeric vector of bandwidths (length `d`).
`m`	Positive integer vector of shifts (length `d`).
`grid_size`	Integer number of grid points per dimension when `grid_points = NULL`.

grid_points	Optional matrix/data frame of explicit evaluation points.
min_vals	Numeric vector of lower grid bounds (length d).
max_vals	Numeric vector of upper grid bounds (length d).
x	Object from ASH_estimate() to print.
...	Additional arguments (unused).
contour	If TRUE, draw a contour-like 2D representation for 2D data.

Details

When grid_points is NULL, a regular grid is constructed from min_vals to max_vals. Custom grids may be irregular; in that case plotting uses point or scatter representations instead of a surface.

Value

A list with class c("glbfp_grid", "ASH_estimate") containing grid coordinates, densities, and grid metadata.

Methods (by generic)

- print(ASH_estimate): Print method for object of class "ASH_estimate".
- plot(ASH_estimate): Plot method for object of class "ASH_estimate".

See Also

ASH(), LBFP_estimate(), GLBFP_estimate()

Examples

```
b <- c(0.5, 0.5)
# Use a small, representative subset so examples remain fast in checks.
sample_data <- as.matrix(ashua[seq_len(120), -3])
out <- ASH_estimate(sample_data, b = b, m = c(1, 1), grid_size = 10)
out
```

---

ashua *River Ashuapmushuan daily flow and level data*

Description

Daily observations of river flow and level for the Ashuapmushuan river.

Usage

```
ashua
```

Format

A data frame with 4,389 rows and 3 variables:

flow Flow rate in cubic meters per second.

level Water level in meters.

day Day code as integer in `YYYYDDD` format.

Details

Data cover 22 March 1992 to 30 September 2007 with a small number of missing calendar days.

Source

Environment and Climate Change Canada, Historical Hydrometric Data. The exact extraction query still needs to be documented in `data-raw/`.

Examples

```
data(ashua)
summary(ashua)
```

`compute_bi_optim` *Compute bandwidth vector b_i*

Description

Computes a plug-in bandwidth vector used by GLBFP/LBFP/ASH estimators. The function validates numeric inputs, stabilizes near-singular covariance matrices with a small ridge if needed, and returns strictly positive bandwidths.

Usage

```
compute_bi_optim(data, m = rep(1, ncol(data)))
```

Arguments

`data`	A numeric matrix or data frame where rows are observations and columns are variables.
`m`	A positive integer vector of shifts, one value per dimension.

Details

The returned vector is intended as a starting value for examples and routine workflows. For applied analysis, sensitivity to the bandwidth should still be checked.

The plug-in expression follows the optimal cell-width calculation for multivariate frequency polygons in Carbon and Duchesne (2024).

Near-singular covariance matrices are stabilized with a small ridge term. If this fails, the function returns an error rather than silently producing non-finite bandwidths.

Value

A numeric vector of positive bandwidths with one value per column in `data`.

References

Carbon, M. and Duchesne, T. (2024). Multivariate frequency polygon for stationary random fields. *Annals of the Institute of Statistical Mathematics*, 76(2), 263-287. doi:10.1007/s10463-023-00883-5.

See Also

`ASH()`, `LBFP()`, `GLBFP()`

Examples

```
set.seed(1)
x <- cbind(rnorm(200), rnorm(200))
compute_bi_optim(x, m = c(1, 1))
```

`compute_Di` *Leave-one-out self-support scores for grid density estimators*

Description

Computes the fixed-grid leave-one-out score

$$D_i = 1 - \widehat{f}_{(-i)}(X_i)/\widehat{f}(X_i)$$

for observations in `data`. The grid, bandwidths, and estimator parameters are held fixed when the contribution of observation `i` is removed.

Usage

```
compute_Di(
  data,
  b = compute_bi_optim(data, m = rep(1, ncol(data))),
  m = rep(1, ncol(data)),
  estimator = c("GLBFP", "LBFP", "ASH"),
  min_vals = apply(data, 2, min),
  max_vals = apply(data, 2, max)
)
```

Arguments

`data`	Numeric matrix or data frame of observations (`n x d`).
`b`	Positive numeric vector of bandwidths (length `d`).
`m`	Positive integer vector of shifts (length `d`). Ignored for `estimator = "LBFP"`, where `m = 1`.
`estimator`	Character string. One of `"GLBFP"`, `"LBFP"`, or `"ASH"`.
`min_vals`	Numeric vector of lower grid bounds (length `d`).
`max_vals`	Numeric vector of upper grid bounds (length `d`).

Value

A list with class `"glbfp_di"` containing the score vector `D`, fitted densities, leave-one-out densities, self-weights, and metadata.

Examples

```
x <- as.matrix(ashua[seq_len(80), -3])
b <- c(0.5, 0.5)
out <- compute_Di(x, b = b, m = c(1, 1), estimator = "GLBFP")
summary(out)
```

`compute_G_star` *Compute the $G^* $ bandwidth constant*

Description

Computes the dimension-dependent constant G^* used by `compute_bi_optim()`.

Usage

```
compute_G_star(d)
```

Arguments

`d`	A single positive integer giving the data dimension.

Details

The implemented formula is

$$G^* = 2^{\frac{3(d-4)}{2(4+d)}} \exp\left(\frac{1}{4+d}\right) \frac{\pi^{d/2}}{4+d}.$$

Value

A positive numeric scalar.

See Also

compute_bi_optim(), G_i(), K_mi()

Examples

```
compute_G_star(1)
compute_G_star(2)
```

G_i *Compute the $G(m_i)$ bandwidth constant*

Description

Computes the scalar constant $G(m_i)$ used by compute_bi_optim().

Usage

```
G_i(mi)
```

Arguments

mi	A single positive numeric value. In package estimators, mi corresponds to one component of the integer shift vector m.

Details

The implemented formula is

$$G(m_i) = \frac{1}{12}\left(1 + \frac{1}{2m_i^2}\right).$$

Value

A positive numeric scalar.

See Also

compute_bi_optim(), K_mi(), compute_G_star()

Examples

```
G_i(1)
G_i(2)
```

GLBFP *General Linear Blend Frequency Polygon (GLBFP) estimator at a single point*

Description

Computes the GLBFP density estimate at point `x`.

Usage

```
GLBFP(
  x,
  data,
  b = compute_bi_optim(data, m = rep(1, ncol(data))),
  m = rep(1, ncol(data)),
  min_vals = apply(data, 2, min),
  max_vals = apply(data, 2, max)
)

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

Arguments

`x`	Object returned by `GLBFP()`.
`data`	Numeric matrix or data frame of observations (`n x d`).
`b`	Positive numeric vector of bandwidths (length `d`).
`m`	Positive integer vector of shifts (length `d`).
`min_vals`	Numeric vector of lower grid bounds (length `d`).
`max_vals`	Numeric vector of upper grid bounds (length `d`).
`...`	Additional arguments (unused).

Details

`GLBFP()` generalizes the linear blend frequency polygon workflow through the positive integer shift vector `m`. Missing and non-finite values are not accepted; remove or impute them before calling the estimator.

Value

A list with class `c("glbfp_fit", "GLBFP")` containing: `x`, `estimation`, `sd`, `IC`, `b`, `m`, `method`, and `dimension`.

Methods (by generic)

- `print(GLBFP)`: Print method for object of class `"GLBFP"`.

References

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley. doi:10.1002/9780470316849.

The complete methodological citation for GLBFP has not yet been verified in this repository. Add it before using this help page as publication text.

See Also

`GLBFP_estimate()`, `ASH()`, `LBFP()`, `compute_bi_optim()`

Examples

```
x <- c(200, 30)
b <- c(0.5, 0.5)
m <- c(1, 1)
GLBFP(x, ashua[, -3], b = b, m = m)
```

`GLBFP_estimate` *GLBFP density estimation on a grid*

Description

Computes GLBFP density estimates on a regular or user-supplied grid.

Usage

```
GLBFP_estimate(
  data,
  b = compute_bi_optim(data, m = rep(1, ncol(data))),
  m = rep(1, ncol(data)),
  grid_size = 20,
  grid_points = NULL,
  min_vals = apply(data, 2, min),
  max_vals = apply(data, 2, max)
)

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

## S3 method for class 'GLBFP_estimate'
plot(x, contour = FALSE, ...)
```

Arguments

data	Numeric matrix or data frame of observations (n x d).
b	Positive numeric vector of bandwidths (length d).
m	Positive integer vector of shifts (length d).
grid_size	Integer number of grid points per dimension when grid_points = NULL.
grid_points	Optional matrix/data frame of explicit evaluation points.
min_vals	Numeric vector of lower grid bounds (length d).
max_vals	Numeric vector of upper grid bounds (length d).
x	Object returned by GLBFP_estimate().
...	Additional arguments (unused).
contour	If TRUE, draw a contour-like 2D representation for 2D data.

Details

When grid_points is NULL, a regular grid is constructed from min_vals to max_vals. Custom grids may be irregular; in that case plotting uses point or scatter representations instead of a surface.

Value

A list with class c("glbfp_grid", "GLBFP_estimate") containing grid coordinates, densities, uncertainty estimates, and grid metadata.

Methods (by generic)

- print(GLBFP_estimate): Print method for object of class "GLBFP_estimate".
- plot(GLBFP_estimate): Plot method for object of class "GLBFP_estimate".

See Also

GLBFP(), ASH_estimate(), LBFP_estimate()

Examples

```
b <- c(0.5, 0.5)
# Use a small, representative subset so examples remain fast in checks.
sample_data <- as.matrix(ashua[seq_len(120), -3])
out <- GLBFP_estimate(sample_data, b = b, m = c(1, 1), grid_size = 10)
out
```

K_mi *Compute the $K(m\_i)$ bandwidth constant*

Description

Computes the scalar constant $K(m_i)$ used by `compute_bi_optim()`.

Usage

```
K_mi(mi)
```

Arguments

`mi`	A single numeric value greater than 0.5. In package estimators, `mi` corresponds to one component of the integer shift vector `m`.

Details

The implemented formula is

$$K(m_i) = \sqrt{\frac{1}{6} + \frac{1}{12m_i^2}} + \frac{4m_i^2 - 1}{6\sqrt{2}m_i} \log\left(\frac{\sqrt{3} + \sqrt{4m_i^2 + 2}}{\sqrt{4m_i^2 - 1}}\right).$$

Value

A positive numeric scalar.

See Also

`compute_bi_optim()`, `G_i()`, `compute_G_star()`

Examples

```
K_mi(1)
K_mi(2)
```

LBFP *Linear Blend Frequency Polygon (LBFP) estimator at a single point*

Description

Computes the LBFP density estimate at point `x`.

Usage

```
LBFP(
  x,
  data,
  b = compute_bi_optim(data, m = rep(1, ncol(data))),
  min_vals = apply(data, 2, min),
  max_vals = apply(data, 2, max)
)

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

Arguments

`x`	Object from `LBFP()`.
`data`	Numeric matrix or data frame of observations (`n x d`).
`b`	Positive numeric vector of bandwidths (length `d`).
`min_vals`	Numeric vector of lower grid bounds (length `d`).
`max_vals`	Numeric vector of upper grid bounds (length `d`).
`...`	Additional arguments (unused).

Details

The estimate is obtained by linear blending of neighboring histogram bin heights. Missing and non-finite values are not accepted; remove or impute them before calling the estimator.

Value

A list with class `c("glbfp_fit", "LBFP")` containing: `x`, `estimation`, `sd`, `IC`, `b`, `method`, and `dimension`.

Methods (by generic)

- `print(LBFP)`: Print method for object of class `"LBFP"`.

References

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley. doi:10.1002/9780470316849.

Terrell, G. R., and Scott, D. W. (1985). Oversmoothed Nonparametric Density Estimates. *Journal of the American Statistical Association*, 80(389), 209-214. doi:10.1080/01621459.1985.10477163.

See Also

LBFP_estimate(), ASH(), GLBFP(), compute_bi_optim()

Examples

```
x <- c(200, 30)
b <- c(0.5, 0.5)
LBFP(x, ashua[, -3], b = b)
```

LBFP_estimate *LBFP density estimation on a grid*

Description

Computes LBFP density estimates on a regular or user-supplied grid.

Usage

```
LBFP_estimate(
  data,
  b = compute_bi_optim(data, m = rep(1, ncol(data))),
  grid_size = 20,
  grid_points = NULL,
  min_vals = apply(data, 2, min),
  max_vals = apply(data, 2, max)
)

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

## S3 method for class 'LBFP_estimate'
plot(x, contour = FALSE, ...)
```

Arguments

data	Numeric matrix or data frame of observations (n x d).
b	Positive numeric vector of bandwidths (length d).
grid_size	Integer number of grid points per dimension when grid_points = NULL.

grid_points	Optional matrix/data frame of explicit evaluation points.
min_vals	Numeric vector of lower grid bounds (length d).
max_vals	Numeric vector of upper grid bounds (length d).
x	Object returned by LBFP_estimate().
...	Additional arguments (unused).
contour	If TRUE, draw a contour-like 2D representation for 2D data.

Details

When grid_points is NULL, a regular grid is constructed from min_vals to max_vals. Custom grids may be irregular; in that case plotting uses point or scatter representations instead of a surface.

Value

A list with class c("glbfp_grid", "LBFP_estimate") containing grid coordinates, densities, uncertainty estimates, and grid metadata.

Methods (by generic)

- print(LBFP_estimate): Print method for object of class "LBFP_estimate".
- plot(LBFP_estimate): Plot method for object of class "LBFP_estimate".

See Also

LBFP(), ASH_estimate(), GLBFP_estimate()

Examples

```
b <- c(0.5, 0.5)
out <- LBFP_estimate(ashua[, -3], b = b, grid_size = 15)
out
plot(out, contour = TRUE)
```

---

lowercase_aliases *Lowercase aliases for the public API*

Description

These aliases follow common R naming style while preserving the original uppercase function names used in earlier versions of the package.

Usage

```
ash(...)

lbfp(...)

glbfp(...)

ash_estimate(...)

lbfp_estimate(...)

glbfp_estimate(...)

compute_di(...)
```

Arguments

`...`	Arguments passed to the corresponding uppercase function.

Value

The same object returned by the corresponding uppercase function.

`predict.glbfp_fit` *Predict from GLBFP fit objects*

Description

Prediction helper for fitted GLBFP objects.

Usage

```
## S3 method for class 'glbfp_fit'
predict(object, newdata = NULL, ...)
```

Arguments

`object`	A fitted object of class `"glbfp_fit"` or `"glbfp_grid"`.
`newdata`	Optional matrix/data frame with points where prediction is requested. For `"glbfp_fit"`, `newdata` is not supported.
`...`	Additional arguments (unused).

Value

Numeric vector of predicted densities.

`summary.glbfp_fit` *Summarize GLBFP fit objects*

Description

Summarizes objects returned by `ASH()`, `LBFP()`, `GLBFP()` and their grid counterparts.

Usage

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

Arguments

`object`	A fitted object of class `"glbfp_fit"` or `"glbfp_grid"`.
`...`	Additional arguments (unused).

Value

A list with class `"summary.glbfp_fit"` or `"summary.glbfp_grid"`.

Index

∗ **datasets**
ashua, 6

as.data.frame.glbfp_di
*(*as.data.frame.glbfp_grid*)*, 3
as.data.frame.glbfp_grid, 3
ASH, 4
ash *(*lowercase_aliases*)*, 17
ASH(), *2*, *6*, *8*, *12*, *16*, *19*
ASH_estimate, 5
ash_estimate *(*lowercase_aliases*)*, 17
ASH_estimate(), *2*, *5*, *6*, *13*, *17*
ashua, 6

base::as.data.frame(), *3*

compute_bi_optim, 7
compute_bi_optim(), *2*, *5*, *9*, *10*, *12*, *14*, *16*
compute_Di, 8
compute_di *(*lowercase_aliases*)*, 17
compute_Di(), *2*
compute_G_star, 9
compute_G_star(), *10*, *14*

G_i, 10
G_i(), *10*, *14*
GLBFP, 11
glbfp *(*lowercase_aliases*)*, 17
GLBFP(), *2*, *5*, *8*, *11*, *13*, *16*, *19*
glbfp(), *2*
GLBFP-package, 2
GLBFP_estimate, 12
glbfp_estimate *(*lowercase_aliases*)*, 17
GLBFP_estimate(), *2*, *6*, *12*, *13*, *17*
glbfp_estimate(), *2*

K_mi, 14
K_mi(), *10*

LBFP, 15
lbfp *(*lowercase_aliases*)*, 17
LBFP(), *2*, *5*, *8*, *12*, *15*, *17*, *19*
LBFP_estimate, 16
lbfp_estimate *(*lowercase_aliases*)*, 17
LBFP_estimate(), *2*, *6*, *13*, *16*, *17*
lowercase_aliases, 17

plot.ASH_estimate *(*ASH_estimate*)*, 5
plot.GLBFP_estimate *(*GLBFP_estimate*)*, 12
plot.LBFP_estimate *(*LBFP_estimate*)*, 16
predict.glbfp_fit, 18
print.ASH *(*ASH*)*, 4
print.ASH_estimate *(*ASH_estimate*)*, 5
print.GLBFP *(*GLBFP*)*, 11
print.GLBFP_estimate *(*GLBFP_estimate*)*, 12
print.LBFP *(*LBFP*)*, 15
print.LBFP_estimate *(*LBFP_estimate*)*, 16

summary.glbfp_fit, 19