

# Package ‘unbalhaar’

May 8, 2026

**Type** Package

**Title** Function Estimation via Unbalanced Haar Wavelets

**Version** 2.1

**Date** 2022-04-19

**Author** Piotr Fryzlewicz

**Maintainer** Piotr Fryzlewicz <p.fryzlewicz@lse.ac.uk>

**Description** Top-down and bottom-up algorithms  
for nonparametric function estimation in Gaussian noise using  
Unbalanced Haar wavelets.

**License** GPL-2

**LazyLoad** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-04-19 12:00:02 UTC

## Contents

unbalhaar-package	2
best.unbal.haar	3
best.unbal.haar.bu	4
hard.thresh	5
hard.thresh.bu	6
inner.prod.iter	7
inner.prod.max	7
inner.prod.max.p	8
med	9
reconstr	10
reconstr.bu	10
uh	11
uh.bu	12
unbal.haar.vector	13

<b>Index</b>	<b>14</b>
--------------	-----------

---

unbalhaar-package      *Function estimation via Unbalanced Haar wavelets*

---

## Description

The package implements top-down and bottom-up algorithms for nonparametric function estimation in Gaussian noise using Unbalanced Haar wavelets.

## Details

Package: unbalhaar  
Type: Package  
Version: 2.0  
Date: 2010-08-09  
License: GPL-2  
LazyLoad: yes

The main functions of the package are [uh](#) and [uh.bu](#).

## Author(s)

Piotr Fryzlewicz

Maintainer: Piotr Fryzlewicz <p.fryzlewicz@lse.ac.uk>

## References

P. Fryzlewicz (2007) “Unbalanced Haar technique for nonparametric function estimation”. *Journal of the American Statistical Association*, 102, 1318-1327.

## Examples

```
x <- c(rep(0, 100), rep(1, 200)) + rnorm(300)
est.topdown <- uh(x)
est.bottomup <- uh.bu(x)
```

---

best.unbal.haar	<i>Best top-down Unbalanced Haar decomposition</i>
-----------------	--

---

**Description**

The function finds the “best” top-down Unbalanced Haar (UH) decomposition of the input vector  $x$ , according to a selection rule (*criterion*) which specifies which UH vector gets chosen at each scale and location.

**Usage**

```
best.unbal.haar(x, criterion = inner.prod.max)
```

**Arguments**

<code>x</code>	a vector
<code>criterion</code>	a function which takes a vector of length $n$ and returns an integer between 1 and $n-1$

**Value**

<code>tree</code>	A list of $J$ matrices, where $J$ represents the number of “scales”. Each matrix is of size $5 \times J$ (the number of UH coefficients at a given scale). Each column (= vector of length 5) contains an Unbalanced Haar coefficient in the following format: 1st component - an index of the coefficient; 2nd component - the value of the coefficient; 3rd component - time point where the corresponding UH vector starts; 4th component - last time point before the breakpoint of the UH vector; 5th component - end point of the UH vector.
<code>smooth</code>	the “smooth” component of $x$ , equal to $\text{sum}(x) / \text{sqrt}(n)$ , where $n$ is the length of $x$

**Author(s)**

Piotr Fryzlewicz

**See Also**

[inner.prod.max](#), [inner.prod.max.p](#), [best.unbal.haar.bu](#)

**Examples**

```
best.unbal.haar(rnorm(100), inner.prod.max.p)
```

---

best.unbal.haar.bu      *Best bottom-up Unbalanced Haar decomposition*

---

### Description

The function finds the “best” bottom-up Unbalanced Haar (UH) decomposition of the input vector  $x$ .

### Usage

```
best.unbal.haar.bu(x, stretch = length(x))
```

### Arguments

<code>x</code>	a vector
<code>stretch</code>	at each iteration, only the first <code>1:stretch</code> elements of the current input vector (whose length decreases by one with each iteration) get scanned in the search for the worst-fitting fine-scale Unbalanced Haar wavelet

### Value

<code>detail</code>	A matrix of size $3 \times n-1$ , where $n$ is the length of $x$ , containing the detail coefficients of $x$ in the order they were chosen. Each column corresponds to a single coefficient and contains, from top to bottom: location of the coefficient, the associated weight, and the value of the coefficient.
<code>smooth</code>	the “smooth” component of $x$ , equal to $\text{sum}(x) / \text{sqrt}(n)$ , where $n$ is the length of $x$

### Author(s)

Piotr Fryzlewicz

### See Also

[best.unbal.haar](#)

### Examples

```
best.unbal.haar.bu(rnorm(100))
```

---

hard.thresh	<i>Hard thresholding of a top-down Unbalanced Haar decomposition</i>
-------------	--

---

**Description**

Presented with an object returned by `best.unbal.haar`, the function sets to zero those Unbalanced Haar coefficients which fall below a certain threshold `sigma`.

**Usage**

```
hard.thresh(buh, sigma = 1)
```

**Arguments**

<code>buh</code>	an object returned by <code>best.unbal.haar</code> containing the decomposition to be thresholded
<code>sigma</code>	the threshold (a positive scalar)

**Value**

a thresholded object, of the same class as `buh`

**Author(s)**

Piotr Fryzlewicz

**See Also**

[best.unbal.haar](#), [hard.thresh.bu](#)

**Examples**

```
x <- rnorm(1000)
x.uh <- best.unbal.haar(x)
x.uh.th <- hard.thresh(x.uh)
x.uh.th.r <- reconstr(x.uh.th)
ts.plot(x.uh.th.r)
```

---

hard.thresh.bu	<i>Hard thresholding of a bottom-up Unbalanced Haar decomposition</i>
----------------	---

---

### Description

Presented with an object returned by `best.unbal.haar.bu`, the function sets to zero those Unbalanced Haar coefficients which fall below a certain threshold `sigma`.

### Usage

```
hard.thresh.bu(buh.bu, sigma = 1)
```

### Arguments

<code>buh.bu</code>	an object returned by <code>best.unbal.haar.bu</code> containing the decomposition to be thresholded
<code>sigma</code>	the threshold (a positive scalar)

### Value

a thresholded object, of the same class as `buh.bu`

### Author(s)

Piotr Fryzlewicz

### See Also

[best.unbal.haar.bu](#), [hard.thresh](#)

### Examples

```
x <- rnorm(1000)
x.uh <- best.unbal.haar.bu(x)
x.uh.th <- hard.thresh.bu(x.uh)
x.uh.th.r <- reconstr.bu(x.uh.th)
ts.plot(x.uh.th.r)
```

---

inner.prod.iter	<i>Inner products with Unbalanced Haar wavelets</i>
-----------------	---

---

**Description**

For an input vector of length  $n$ , the function computes inner products between the input vector and all possible  $n-1$  Unbalanced Haar vectors of length  $n$ .

**Usage**

```
inner.prod.iter(x)
```

**Arguments**

`x` a vector of length  $n$

**Details**

The computation is iterative and is performed in computational time  $O(n)$ .

**Value**

a vector of length  $n-1$ , containing inner products between `x` and consecutive Unbalanced Haar wavelets of length  $n$

**Author(s)**

Piotr Fryzlewicz

**Examples**

```
inner.prod.iter(rnorm(100))
```

---

inner.prod.max	<i>Unbalanced Haar wavelet which maximises the inner product</i>
----------------	--

---

**Description**

The function finds the Unbalanced Haar vector which yields the largest (in absolute value) inner product with the input vector.

**Usage**

```
inner.prod.max(x)
```

**Arguments**

x                    a vector

**Value**

The index where `abs(inner.prod.iter(x))` is maximised. If two or more maxima are found, the med of their locations is returned.

**Author(s)**

Piotr Fryzlewicz

**See Also**

[inner.prod.iter](#), [med](#), [inner.prod.max.p](#)

**Examples**

```
inner.prod.max(c(rep(0, 100), rep(1, 200)))
```

---

inner.prod.max.p            *Unbalanced Haar wavelet which maximises the inner product*

---

**Description**

The function finds the Unbalanced Haar vector which yields the largest (in absolute value) inner product with the input vector, amongst those Unbalanced Haar vectors whose breakpoint is located between  $100(1-p)\%$  and  $100p\%$  of their support.

**Usage**

```
inner.prod.max.p(x, p = 0.8)
```

**Arguments**

x                    a vector  
p                    a scalar in (0.5, 1]

**Value**

The index where `abs(inner.prod.iter(x))` is maximised on the subinterval  $(1+\text{floor}((1-p)*n)):\text{ceiling}(p*n)$ , where  $n$  is the length of  $x$ . If two or more maxima are found, the med of their locations is returned.

**Author(s)**

Piotr Fryzlewicz

**See Also**

[inner.prod.iter](#), [med](#), [inner.prod.max](#)

**Examples**

```
inner.prod.max.p(c(rep(0, 100), rep(1, 200)), .55)
```

---

med

*Median*

---

**Description**

The function computes the median of a vector. Unlike `median`, it is guaranteed to return a value which is a component of the input vector.

**Usage**

```
med(x)
```

**Arguments**

x                    a vector

**Value**

a scalar defined as `quantile(x, .5, type=3)[[1]]`

**Author(s)**

Piotr Fryzlewicz

**Examples**

```
med(1:4)
median(1:4)
```

reconstr

*Reconstruct a top-down Unbalanced Haar decomposition*

---

**Description**

Reconstructs a vector from its top-down Unbalanced Haar decomposition stored in an object returned by `best.unbal.haar` or `hard.thresh`.

**Usage**

```
reconstr(buh)
```

**Arguments**

`buh` an object of the type returned by `best.unbal.haar` and `hard.thresh`

**Value**

the inverse Unbalanced Haar transform of `buh`

**Author(s)**

Piotr Fryzlewicz

**See Also**

[best.unbal.haar](#), [hard.thresh](#), [reconstr.bu](#)

**Examples**

```
x <- rnorm(1000)
x.uh <- best.unbal.haar(x)
x.uh.th <- hard.thresh(x.uh)
x.uh.th.r <- reconstr(x.uh.th)
ts.plot(x.uh.th.r)
```

---

reconstr.bu*Reconstruct a bottom-up Unbalanced Haar decomposition*

---

**Description**

Reconstructs a vector from its bottom-up Unbalanced Haar decomposition stored in an object returned by `best.unbal.haar.bu` or `hard.thresh.bu`.

**Usage**

```
reconstr.bu(buh.bu)
```

**Arguments**

buh.bu                    an object of the type returned by `best.unbal.haar.bu` and `hard.thresh.bu`

**Value**

the inverse Unbalanced Haar transform of `buh.bu`

**Author(s)**

Piotr Fryzlewicz

**See Also**

[best.unbal.haar.bu](#), [hard.thresh.bu](#), [reconstr](#)

**Examples**

```
x <- rnorm(1000)
x.uh <- best.unbal.haar.bu(x)
x.uh.th <- hard.thresh.bu(x.uh)
x.uh.th.r <- reconstr.bu(x.uh.th)
ts.plot(x.uh.th.r)
```

---

uh

*Denoising via top-down Unbalanced Haar*

---

**Description**

Given an input vector of the form “signal + iid Gaussian noise”, the function estimates the noise level via Median Absolute Deviation, finds the best top-down Unbalanced Haar decomposition (according to the selection rule `criterion`), thresholds it with the universal threshold, and performs the inverse Unbalanced Haar transform to yield an estimate of the signal.

**Usage**

```
uh(x, criterion = inner.prod.max)
```

**Arguments**

x                        a vector of the form “signal + iid Gaussian noise”  
criterion                a function which takes a vector of length  $n$  and returns an integer between 1 and  $n-1$

**Value**

an estimate of the signal

**Author(s)**

Piotr Fryzlewicz

**References**

P. Fryzlewicz (2007) “Unbalanced Haar technique for nonparametric function estimation”. *Journal of the American Statistical Association*, 102, 1318-1327.

**See Also**

[uh.bu](#), [best.unbal.haar](#), [inner.prod.max](#), [inner.prod.max.p](#), [hard.thresh](#), [reconstr](#)

**Examples**

```
x <- c(rep(0, 100), rep(1, 200)) + rnorm(300)
est <- uh(x)
```

---

uh.bu

*Denoising via bottom-up Unbalanced Haar*

---

**Description**

Given an input vector of the form “signal + iid Gaussian noise”, the function estimates the noise level via Median Absolute Deviation, finds the best bottom-up Unbalanced Haar decomposition, thresholds it with the universal threshold, and performs the inverse Unbalanced Haar transform to yield an estimate of the signal.

**Usage**

```
uh.bu(x, stretch = length(x))
```

**Arguments**

x	a vector of the form “signal + iid Gaussian noise”
stretch	at each iteration, only the first 1:stretch elements of the current input vector (whose length decreases by one with each iteration) get scanned in the search for the worst-fitting fine-scale Unbalanced Haar wavelet

**Value**

an estimate of the signal

**Author(s)**

Piotr Fryzlewicz

**References**

P. Fryzlewicz (2007) “Unbalanced Haar technique for nonparametric function estimation”. *Journal of the American Statistical Association*, 102, 1318-1327.

**See Also**

[uh](#), [best.unbal.haar.bu](#), [hard.thresh.bu](#), [reconstr.bu](#)

**Examples**

```
x <- c(rep(0, 100), rep(1, 200)) + rnorm(300)
est <- uh.bu(x)
```

---

unbal.haar.vector      *Unbalanced Haar vector*

---

**Description**

Computes the non-zero part of an Unbalanced Haar vector with a given start-, break- and end-point.

**Usage**

```
unbal.haar.vector(a)
```

**Arguments**

**a**                    a three-component vector of integers such that  $a[1] \leq a[2] < a[3]$ . The three components specify, respectively, the start point, the time point just before the breakpoint, and the endpoint of the desired Unbalanced Haar vector.

**Value**

the non-zero part of the corresponding Unbalanced Haar vector

**Author(s)**

Piotr Fryzlewicz

**Examples**

```
unbal.haar.vector(c(1, 1, 2))
unbal.haar.vector(c(2, 5, 12))
```

# Index

## \* **math**

- best.unbal.haar, [3](#)
- best.unbal.haar.bu, [4](#)
- hard.thresh, [5](#)
- hard.thresh.bu, [6](#)
- inner.prod.iter, [7](#)
- inner.prod.max, [7](#)
- inner.prod.max.p, [8](#)
- med, [9](#)
- reconstr, [10](#)
- reconstr.bu, [10](#)
- uh, [11](#)
- uh.bu, [12](#)
- unbal.haar.vector, [13](#)

## \* **package**

- unbalhaar-package, [2](#)

best.unbal.haar, [3](#), [4](#), [5](#), [10](#), [12](#)  
best.unbal.haar.bu, [3](#), [4](#), [6](#), [11](#), [13](#)

hard.thresh, [5](#), [6](#), [10](#), [12](#)  
hard.thresh.bu, [5](#), [6](#), [11](#), [13](#)

inner.prod.iter, [7](#), [8](#), [9](#)  
inner.prod.max, [3](#), [7](#), [9](#), [12](#)  
inner.prod.max.p, [3](#), [8](#), [8](#), [12](#)

med, [8](#), [9](#), [9](#)

reconstr, [10](#), [11](#), [12](#)  
reconstr.bu, [10](#), [10](#), [13](#)

uh, [2](#), [11](#), [13](#)  
uh.bu, [2](#), [12](#), [12](#)  
unbal.haar.vector, [13](#)  
unbalhaar (unbalhaar-package), [2](#)  
unbalhaar-package, [2](#)