

# Package ‘sad’

May 9, 2026

**Title** Verify the Scale, Anisotropy and Direction of Weather Forecasts

**Version** 0.1.3

**Description** Implementation of the wavelet-based spatial verification method of Buschow and Friederichs ``SAD: Verifying the Scale, Anisotropy and Direction of precipitation forecasts" (2020, submitted to QJRMS). Forecasts and Observations are transformed by a decimated or redundant dual-tree complex wavelet transform to analyze the spatial scale, degree of anisotropy and preferred direction in each field. These structural attributes are compared by a series of scores. An experimental algorithm for the correction of these errors is included as well.

**License** MIT + file LICENSE

**Imports** emdist

**Depends** dualtrees

**Encoding** UTF-8

**LazyData** false

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Author** Sebastian Buschow [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-4750-361X>>)

**Maintainer** Sebastian Buschow <s6sebusc@uni-bonn.de>

**Repository** CRAN

**Date/Publication** 2020-11-06 16:30:02 UTC

## Contents

getpareto	2
hemd	2
prepare_sad	3
raincols	4
rrain	4
sadcorrect	5
sadforecast	6
sadverif	7
semd	8

**Index** **10**


---

<code>getpareto</code>	<i>Find the pareto set</i>
------------------------	----------------------------

---

**Description**

Determine the set of pareto optimal forecasts in a matrix of scores

**Usage**

```
getpareto(scores)
```

**Arguments**

<code>scores</code>	a matrix of negatively oriented scores where the rows correspond to different forecasts and the columns denote different scores.
---------------------	--

**Details**

The Pareto set contains all those forecasts for which no other forecast is better in every respect. In this function, we assume that all scores are negatively oriented, "better" therefore means lower values.

**Value**

a vector of indices indicating all members of the pareto set.

**Note**

This function becomes very memory hungry if you have more than 1000 forecasts, be careful.

---

<code>hemd</code>	<i>histogram emd</i>
-------------------	----------------------

---

**Description**

Earth Mover's Distance between two histograms, given as vectors

**Usage**

```
hemd(h1, h2, mids = NULL)
```

**Arguments**

<code>h1, h2</code>	vectors of non-negative numbers representing two histograms
<code>mids</code>	the bin mids corresponding to the histograms. Can also be given via the names of h1.

**Value**

the value of the EMD

---

prepare_sad	<i>prepare a sad forecast for verification</i>
-------------	--

---

**Description**

remove small values, apply log-transform, smooth borders, handle boundary conditions

**Usage**

```
prepare_sad(x, xmin = 0.1, log = TRUE, rsm = 0, Nx = NULL,
            Ny = NULL, boundaries = "pad")
```

**Arguments**

x	a list of 2 or more 2D matrices with equal sizes and no missing or infinite values, as required by <code>as.sadforecast</code>
xmin	values smaller than <code>xmin</code> are set to zero
log	logical, do you want to log-transform the data? (recommended for precipitation)
rsm	number of pixels which are linearly smoothed at the edge
Nx	size to which the data is extended in x-direction
Ny	size to which the data is extended in y-direction
boundaries	how to handle the boundary conditions, either "pad", "mirror" or "periodic"

**Details**

the positions within the extended field where the original field resides are output as attributes "px", "py" of the result. The other input parameters are saved as attributes of the result as well.

**Value**

an object of class `sadforecast` which has been prepared in the desired way.

**Examples**

```
data( rrain )
ra <- list( rrain[2,4,,], rrain[3,9,,] )
ra <- prepare_sad( ra, rsm=0, Nx=256, boundaries="mirror", log=FALSE )
plot(ra)
```

---

raincols	<i>rain color scale</i>
----------	-------------------------

---

**Description**

eight shades of blue used in `plot.sadforecast`

**Usage**

```
raincols
```

**Format**

An object of class character of length 8.

---

rrain	<i>Random Rain</i>
-------	--------------------

---

**Description**

Randomly simulated synthetic rain fields with Matern covariances

**Usage**

```
data(rrain)
```

**Format**

A 4x10x128x128 matrix

**Details**

These fields were used in Buschow et al. (2019) <doi:10.5194/gmd-12-3401-2019>. The first array corresponds to the four model configurations from that paper (different roughness `nu` and scale `sc`), the second dimension contains ten realizations for each model.

**Source**

simulated using the 'RandomFields' package, code available at <10.5281/zenodo.3257511>

**Examples**

```
data(rrain)
```

---

sadcorrect	<i>correct structure errors</i>
------------	---------------------------------

---

### Description

use the inverse 'dctwt' to correct errors in scale, anisotropy and direction

### Usage

```
sadcorrect(x, xmin = 0.1, log = TRUE, rsm = 0, Nx = NULL,
           Ny = NULL, J = NULL, boundaries = "pad", direction = TRUE)
```

### Arguments

x	a list of equally sized matrices, the first element is assumed to be the observation
xmin	values smaller than xmin are set to zero
log	logical, do you want to log-transform the data? (recommended for precipitation)
rsm	number of pixels which are linearly smoothed at the edge
Nx	size to which the data is extended in x-direction, has to be a whole power of 2
Ny	size to which the data is extended in y-direction, has to be a whole power of 2
J	largest scale considered
boundaries	how to handle the boundary conditions, either "pad", "mirror" or "periodic"
direction	if TRUE, scale and direction are corrected, otherwise only scale

### Details

The algorithm performs the following steps:

1. remove values below xmin
2. if log=TRUE log-transform all fields
3. set all fields to zero mean, unit variance
4. apply dctwt to all fields
5. loop over forecasts and scales. If direction=TRUE loop over the six directions. Multiply forecast energy at each location by the ratio of total observed energy to total forecast energy at that scale (and possibly direction)
6. apply idtwt to all forecasts
7. reset means and variance of the forecasts to their original values
8. if log=TRUE invert the log-transform
9. return the list of corrected fields

### Value

an object of class sadforecast

**Examples**

```
data(rrain)
ra  <- as.sadforecast( list( rrain[2,1,,], rrain[3,1,,], rrain[3,2,,], rrain[3,3,,] ) )
ra_c <- sadcorrect( ra, rsm=10 )
plot(ra_c)
```

---

sadforecast	<i>class for a list of forecasts</i>
-------------	--------------------------------------

---

**Description**

check that a list of forecasts fulfills all requirements to be verified by our method

**Usage**

```
as.sadforecast(x)

## S3 method for class 'sadforecast'
plot(x, mfrow = NULL, col = NULL, ...)
```

**Arguments**

x	a list of 2 or more 2D matrices with equal sizes and no missing or infinite values
mfrow	vector with the number of rows and columns you would like in the plot
col	color scale for the plot
...	further arguments passed to image

**Details**

as.sadforecast does nothing except check that everything is as it should be, add the attributes that can be changed by prepare\_sad and provide a method for quick plots of the data.

**Value**

an object of class sadforecast

**Examples**

```
data( rrain )
ra <- list( rrain[1,1,,], rrain[4,5,,], rrain[2,7,,] )
ra <- as.sadforecast(ra)
plot(ra)
```

---

sadverif                      *dual-tree verification*

---

## Description

verify the scale, anisotropy and direction of a number of forecasts

## Usage

```
sadverif(x, dec = TRUE, xmin = 0.1, log = TRUE, a = 1, nbr = 33,
         rsm = 0, Nx = NULL, Ny = NULL, J = NULL, boundaries = "pad",
         return_specs = FALSE)
```

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

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

## Arguments

x	a list of equally sized matrices, the first element is assumed to be the observation
dec	logical, do you want to use the decimated transform
xmin	values smaller than xmin are set to zero
log	logical, do you want to log-transform the data? (recommended for precipitation)
a	relative weight of directional errors compared to scale errors in semdd
nbr	number of breaks for the scale histograms, has no effect if dec=TRUE
rsm	number of pixels which are linearly smoothed at the edge
Nx	size to which the data is extended in x-direction
Ny	size to which the data is extended in y-direction
J	largest scale considered
boundaries	how to handle the boundary conditions, either "pad", "mirror" or "periodic"
return_specs	if TRUE, the spatial mean spectra are returned as well
...	further arguments, currently ignored.
object	object of class sadverif

## Details

each element of x is transformed via dtcwt from the 'dualtrees' package. Scores and centres based on the mean spectra are calculated. If dec=FALSE, scale histograms and the corresponding score hemd are calculated as well.

**Value**

an object of class `sadverif`, containing the following elements

**settings** a dataframe containing the parameters that were originally passed to `dtverif`

**centres** a matrix containing the anisotropy  $\rho$ , angle  $\phi$  and central scale  $z$  derived from the mean spectra. Rain area and sum are included as well.

**detscores** a matrix containing the differences in centre components, the direction/anisotropy score  $dxy$ , the emd between direction-averaged spectra (`semd`) and the emd between the directional spectra (`semdd`). If `dec=FALSE`, the emd between the scale histograms, `hemd`, is included as well.

**time** the time the calculation took in seconds

if there is more than one forecast, the ensemble scores `SpEn` and (if available), `hemd` are computed as well, treating all forecasts as members of the ensemble to be verified.

**References**

Selesnick, I.W., R.G. Baraniuk, and N.C. Kingsbury (2005) <doi:10.1109/MSP.2005.1550194>  
 Buschow et al. (2019) <doi:10.5194/gmd-12-3401-2019> Buschow and Friederichs (2020) <doi:10.5194/ascmo-6-13-2020>

**Examples**

```
oldpar <- par(no.readonly=TRUE)
on.exit(par(oldpar))
data(rrain)
ra <- as.sadforecast( list( rrain[1,1,,], rrain[1,2,,], rrain[2,1,,], rrain[3,1,,] ) )
plot(ra)
verif <- sadverif( ra, log=FALSE, xmin=0 )
summary(verif)
par( mfrow=c(2,2) )
plot( verif )
```

---

semd

*spectral emd*


---

**Description**

earth mover's distance between dual-tree wavelet spectra

**Usage**

```
semd(dt1, dt2, a = 1, dir = TRUE)
```

**Arguments**

<code>dt1, dt2</code>	forecast and observed spectrum
<code>a</code>	ratio between scale- and directional component
<code>dir</code>	whether or not to include direction information

**Value**

a single value, the emd. If *dir*=FALSE, the value is signed, indicating the direction of the scale error.

# Index

## \* datasets

raincols, 4

rrain, 4

as.sadforecast (sadforecast), 6

getpareto, 2

hemd, 2

plot.sadforecast (sadforecast), 6

plot.sadverif (sadverif), 7

prepare\_sad, 3

raincols, 4

rrain, 4

sadcorrect, 5

sadforecast, 6

sadverif, 7

send, 8

summary.sadverif (sadverif), 7