

# Package ‘Rwclust’

January 20, 2025

**Title** Random Walk Clustering on Weighted Graphs

**Version** 0.1.0

**Author** Carson Sprock [aut, cre]

**Maintainer** Carson Sprock <csprock@gmail.com>

**Description**

Implements the random walk clustering algorithm for weighted graphs as found in Harel and Koren (2001) <[https://link.springer.com/chapter/10.1007/3-540-45294-X\\_3](https://link.springer.com/chapter/10.1007/3-540-45294-X_3)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Suggests** igraph, knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Imports** checkmate, Matrix

**Depends** R (>= 3.5.0)

**LazyData** true

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-07-24 21:50:06 UTC

## Contents

adjacency . . . . .	2
apply_similarity . . . . .	2
compute_similarities . . . . .	3
compute_transition_matrix . . . . .	4
create_weight_matrix . . . . .	4
example1 . . . . .	5
example2 . . . . .	5
new_rwclust . . . . .	6

plot.rwclust . . . . .	6
run_main_loop . . . . .	7
rwclust . . . . .	7
update_weights . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

adjacency	<i>Generic helper for extracting adjacency matrix from rwclust object.</i>
-----------	--

---

### Description

Generic helper for extracting adjacency matrix from rwclust object.

### Usage

```
adjacency(x)

## Default S3 method:
adjacency(x)

## S3 method for class 'rwclust'
adjacency(x)
```

### Arguments

x                    rwclust object

### Value

Matrix object containing the adjacency matrix of the after the final iteration

---

apply_similarity	<i>Apply similarity function to rows of a matrix</i>
------------------	--

---

### Description

Apply similarity function to rows of a matrix

### Usage

```
apply_similarity(idx, mat, similarity, ...)
```

**Arguments**

<code>idx</code>	vector of length two containing row indices
<code>mat</code>	a matrix
<code>similarity</code>	similarity function to apply
<code>...</code>	additional parameters to be passed to the similarity function

**Value**

a scalar

---

`compute_similarities` *Apply similarity function over edges of graph*

---

**Description**

Apply similarity function over edges of graph

**Usage**

```
compute_similarities(edgelist, mat, similarity, ...)
```

**Arguments**

<code>edgelist</code>	3-column dataframe
<code>mat</code>	a matrix
<code>similarity</code>	the similarity function to apply
<code>...</code>	other parameters to pass to the similarity function

**Value**

a vector containing updated weights

compute\_transition\_matrix

*Compute transition matrix*

---

### Description

Compute transition matrix

### Usage

```
compute_transition_matrix(x)
```

### Arguments

x                    sparseMatrix or denseMatrix

### Value

transition matrix

---

create\_weight\_matrix    *Construct sparse matrix from weighted edgelist*

---

### Description

Takes the weights from compute\_kernel and creates weighted adjacency matrix

### Usage

```
create_weight_matrix(edgelist, weights, ...)
```

### Arguments

edgelist            a dataframe with two columns  
weights            a vector of weights  
...                other parameters to be passed to Matrix::sparseMatrix()

### Value

sparseMatrix

---

`example1`*Example Graph 1*

---

**Description**

First demonstration test graph used in the original.

**Usage**

```
example1
```

**Format**

A data frame with three columns representing a weighted graph. Each row represents an edge with a weight:

**from** An integer vertex id

**to** An integer vertex id

**weight** A double representing the edge weight

**Examples**

```
data(example1, package="Rwclust")
```

---

`example2`*Example Graph 2*

---

**Description**

Second demonstration test graph used in the original paper.

**Usage**

```
example2
```

**Format**

A data frame with three columns representing a weighted graph. Each row represents an edge with a weight.

**from** An integer vertex id

**to** An integer vertex id

**weight** A double representing the edge weight

**Examples**

```
data(example2, package="Rwclust")
```

---

new_rwclust	<i>rwclust class constructor</i>
-------------	----------------------------------

---

**Description**

Returns a object of class "rwclust" for use with generic summary and plotting functions.

**Usage**

```
new_rwclust(x)
```

**Arguments**

x                    output of run\_main\_loop function

**See Also**

[run\\_main\\_loop\(\)](#)

---

plot.rwclust	<i>Generic plotting for rwclust object</i>
--------------	--

---

**Description**

Generic function for plotting the distribution of weights. Calls hist under the hood.

**Usage**

```
## S3 method for class 'rwclust'
plot(x, cutoff = NULL, ...)
```

**Arguments**

x                    rwclust object  
 cutoff              optional numeric, will plot the cutoff value as a vertical line  
 ...                  additional graphical parameters passed to the hist function

---

run_main_loop	<i>Execute main algorithm loop</i>
---------------	------------------------------------

---

**Description**

Execute main algorithm loop

**Usage**

```
run_main_loop(M, edgelist, similarity, k, iter)
```

**Arguments**

M	transition matrix
edgelist	dataframe edgelist
similarity	a similarity function
k	integer, length of longest walk
iter	number of iterations

**Value**

list

---

rwclust	<i>Sharpen the edge weights of a weighted graph.</i>
---------	--

---

**Description**

Sharpens the weights of a weighted graph for later pruning.

**Usage**

```
rwclust(x, iter = 5, k = 3, similarity = "hk")
```

```
## S3 method for class 'data.frame'
```

```
rwclust(x, iter = 5, k = 3, similarity = "hk")
```

```
## S3 method for class 'matrix'
```

```
rwclust(x, iter = 5, k = 3, similarity = "hk")
```

**Arguments**

<code>x</code>	matrix or dataframe with three columns <ol style="list-style-type: none"> <li>1. vertex label (integer)</li> <li>2. vertex label (integer)</li> <li>3. edge weights (float)</li> </ol>
<code>iter</code>	integer, number of iterations
<code>k</code>	integer, maximum length of random walk
<code>similarity</code>	string, the name of the similarity metric used to update weights

**Value**

list

**weights** A vector of the updated edge weights

**adj** Updated adjacency matrix containing updated weights

**Details**

Internally, the edgelist passed to `rwclust` is converted into a transition matrix, whose powers are used to compute the probability of reaching a vertex  $u$  from vertex  $v$  in  $k$  steps for all  $v$  and  $u$ . New edge weights are computed using the similarity between these "walk probabilities" for each pair of vertices. The intuition is that vertices who have similar neighborhoods in terms of random walk reachability are similar to each other.

The returned weights can be used for clustering by deleting edges with weights below a certain threshold. The connected components of the resulting graph form the clusters.

**References**

Harel, David, and Yehuda Koren. "On clustering using random walks." International Conference on Foundations of Software Technology and Theoretical Computer Science. Springer, Berlin, Heidelberg, 2001.

---

update_weights	<i>Update edge weights</i>
----------------	----------------------------

---

**Description**

Update edge weights

**Usage**

```
update_weights(M, edgelist, similarity, k)
```



**Arguments**

M	matrix
edgelist	dataframe representing weighted edgelist
similarity	a similarity function
k	integer, length of longest walk

**Value**

list

# Index

## \* datasets

example1, [5](#)

example2, [5](#)

adjacency, [2](#)

apply\_similarity, [2](#)

compute\_similarities, [3](#)

compute\_transition\_matrix, [4](#)

create\_weight\_matrix, [4](#)

example1, [5](#)

example2, [5](#)

new\_rwclust, [6](#)

plot\_rwclust, [6](#)

run\_main\_loop, [7](#)

run\_main\_loop(), [6](#)

rwclust, [7](#)

update\_weights, [8](#)