

# Package ‘graphonmix’

May 8, 2026

**Type** Package

**Title** Generates Mixture Graphs from Dense and Sparse Graphons

**Version** 0.0.1.0

**Maintainer** Sevvandi Kandanaarachchi <sevvandik@gmail.com>

**Description** Generates (U,W) mixture graphs where U is a line graph graphon and W is a dense graphon. Graphons are graph limits and graphon U can be written as sequence of positive numbers adding to 1. Graphs are sampled from U and W and joined randomly to obtain the mixture graph. Given a mixture graph, U can be inferred. Kandanaarachchi and Ong (2025) <[doi:10.48550/arXiv.2505.13864](https://doi.org/10.48550/arXiv.2505.13864)>.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** ggplot2, igraph, imager, stats

**Suggests** gridExtra, knitr, rmarkdown

**VignetteBuilder** knitr

**URL** <https://sevvandi.github.io/graphonmix/>

**NeedsCompilation** no

**Author** Sevvandi Kandanaarachchi [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-0337-0395>>)

**Repository** CRAN

**Date/Publication** 2025-10-09 07:40:02 UTC

## Contents

autoplot.extract_sparse . . . . .	2
create_exp_matrix . . . . .	3
empirical_graphon . . . . .	3
extract_sparse . . . . .	4
generate_star_union . . . . .	5

graph_join . . . . .	5
line_graphon . . . . .	6
plot_graphon . . . . .	7
predict_hubs . . . . .	7
sample_graphon . . . . .	8
sample_mixed_graph . . . . .	8
scale_graphon . . . . .	9

## Index 11

---

autoplot.extract\_sparse

*Plots the output of extract\_sparse function.*

---

### Description

Plots the two lines fitted to the unique, sorted log degrees of the graph.

### Usage

```
## S3 method for class 'extract_sparse'
autoplot(object, ...)
```

### Arguments

object	The output of the function 'extract_sparse'
...	Other arguments currently ignored

### Value

A ggplot object.

### Examples

```
library(igraph)
gr <- sample_pa(10000, power = 1.2, directed = FALSE)
sparse <- extract_sparse(gr)
autoplot(sparse)
```

---

create\_exp\_matrix      *Creates an nxn exponential matrix*

---

**Description**

Creates an nxn matrix where the (i,j)th entry is  $\exp(-(i+j)/\text{scalar})$

**Usage**

```
create_exp_matrix(nrow, scalar)
```

**Arguments**

nrow                    The dimension of the matrix  
scalar                  The scalar in  $\exp(-(i+j)/\text{scalar})$

**Value**

An nxn matrix

**Examples**

```
W <- create_exp_matrix(100, 100)
```

---

empirical\_graphon      *Computes empirical graphon from graph*

---

**Description**

Computes empirical graphon given a graph

**Usage**

```
empirical_graphon(gr, n = NULL)
```

**Arguments**

gr                      A graph  
n                        Dimension of the graphon matrix

**Value**

The empirical graphon

**Examples**

```
library(igraph)
gr <- sample_gnp(1000, p=0.2)
emp <- empirical_graphon(gr, n = 100)
```

---

extract_sparse	<i>Extracts the sparse part from a (U,W) graphon mixture</i>
----------------	--

---

**Description**

This function extracts the sparse component from a (U,W) mixture graph by fitting two lines to the unique sorted log degree values. The first line gives the number of hubs in the sparse part and the remaining line fits the degrees of the dense part.

**Usage**

```
extract_sparse(gr)
```

**Arguments**

<code>gr</code>	The input graph
-----------------	-----------------

**Value**

a list with the following components:

<code>num_hubs</code>	The number of hubs in the sparse component.
<code>phat</code>	The probability vector of the sparse component. This is also known as the mass partition.
<code>segment_sizes</code>	The sizes of the two line segments.
<code>line_equations</code>	The two equations of the lines.
<code>cutoff</code>	The best cut off for the two lines.
<code>models</code>	The models of the fitted lines.
<code>mse</code>	The mean squared error
<code>data</code>	The degree data.

**Examples**

```
library(igraph)
gr <- sample_pa(10000, power = 1.2, directed = FALSE)
sparse <- extract_sparse(gr)
sparse$phat
```

---

generate\_star\_union     *Generates a sparse graph of star graphs*

---

**Description**

Generates a union of star graphs given the weights

**Usage**

```
generate_star_union(wts, n)
```

**Arguments**

wts	The proportion of the hub degrees
n	The number of nodes in the new graph

**Value**

A disjoint union of star graphs

**Examples**

```
library(igraph)
wts <- c(0.5, 0.3, 0.2)
gr <- generate_star_union(wts, n = 100)
gr
```

---

graph\_join     *Joins two graphs*

---

**Description**

Joins two graphs randomly connecting vertices

**Usage**

```
graph_join(gr1, gr2, p = 0.5, option = 2)
```

**Arguments**

gr1	The first graph to join
gr2	The second graph to join
p	The proportion of edges in gr1 to be added as part of the joining
option	Two options. 1 does the disjoint union, 2 does the random edges union.

**Value**

The joined graph

**Examples**

```
W <- create_exp_matrix(100, 100)
# create the sparse part - a disjoint set of stars
wts <- c(0.5, 0.3, 0.2)
grdense <- sample_graphon(W, 100)
grsparse <- generate_star_union(wts, 200)
gr <- graph_join(grdense, grsparse, opt = 2)
```

---

line\_graphon

*Creates a line graphon from a sequence of probabilities*

---

**Description**

Creates a line graphon, which is a disjoint clique graphon from a sequence of probability values

**Usage**

```
line_graphon(probs)
```

**Arguments**

probs            The list of probabilities starting from the largest.

**Value**

The line graphon

**Examples**

```
library(ggplot2)
wts <- c(0.5, 0.3, 0.2)
U <- line_graphon(wts)
plot_graphon(U)
```

---

plot_graphon	<i>Plots graphon</i>
--------------	----------------------

---

**Description**

Plots graphon

**Usage**

```
plot_graphon(W, cols = c("white", "black"))
```

**Arguments**

W	A graphon given by an nxn matrix
cols	Colors, by default white and black

**Value**

A ggplot object.

**Examples**

```
W <- create_exp_matrix(100, 100)
plot_graphon(W)
```

---

predict_hubs	<i>Predicts the degree of hubs of an new graph</i>
--------------	--

---

**Description**

Predicts the degree of hubs of an unseen graph given a graph generated from the same process.

**Usage**

```
predict_hubs(gr, n, k = NULL)
```

**Arguments**

gr	The input graph
n	The number of nodes in the new graph
k	The number of hubs. Default is NULL

**Value**

A vector of hub degrees

**Examples**

```
library(igraph)
gr <- sample_pa(10000, power = 1.2, directed = FALSE)
predict_hubs(gr, n = 11000)
```

---

sample_graphon	<i>Generates a graph given a graphon</i>
----------------	--

---

**Description**

Generates a graph given a dense graphon  $W$ .

**Usage**

```
sample_graphon(W, n)
```

**Arguments**

$W$	A graphon given by a matrix
$n$	The number of nodes of the sampled graph

**Value**

A graph sampled from the graphon  $W$  with  $n$  nodes

**Examples**

```
library(igraph)
W <- matrix(0.2, nrow = 100, ncol = 100)
gr <- sample_graphon(W, n = 100)
```

---

sample_mixed_graph	<i>Generate a (U,W) mixture graph</i>
--------------------	---------------------------------------

---

**Description**

Generate a  $(U,W)$  mixture graph from a dense graphon  $W$  and a mass partition corresponding to a line graph graphon  $U$ .

**Usage**

```
sample_mixed_graph(W, wts, nd, ns, p = 0.5, option = 2)
```

**Arguments**

W	The dense graphon. This is a symmetric matrix with values in [0,1]
wts	The degree proportions of the hub degrees. Need to add up to 1. This is the mass partition corresponding to the line graph graphon U.
nd	The number of nodes in the dense part of the graph
ns	The number of nodes in the sparse part of the graph
p	The nodes to be added as a proportion of the edges in the dense part
option	Graph joining option. If option == 1 then a disjoint union is considered. If option == 2 the two graphs are joined randomly with the number of edges specified by p.

**Value**

A graph sampled from the (U,W) mixture.

**Examples**

```
library(igraph)
W <- matrix(0.1, nrow = 100, ncol = 100)
wts <- c(0.5, 0.3, 0.2)
ns <- 200
nd <- 100
p <- 0.5
gr <- sample_mixed_graph(W, wts, nd, ns, p, option = 2)
gr
```

---

scale\_graphon

*Scales a graphon to an nxn matrix*

---

**Description**

Scales a graphon to an nxn matrix suitable for large adjacency matrices

**Usage**

```
scale_graphon(W, n)
```

**Arguments**

W	A graphon given as a symmetric square matrix
n	The dimension of the output matrix

**Value**

Scaled nxn graphon

**Examples**

```
library(igraph)
gr <- sample_gnp(1000, p=0.2)
adj <- as_adjacency_matrix(gr)
W <- scale_graphon(adj, 100)
```

# Index

`autoplot.extract_sparse`, 2

`create_exp_matrix`, 3

`empirical_graphon`, 3

`extract_sparse`, 4

`generate_star_union`, 5

`graph_join`, 5

`line_graphon`, 6

`plot_graphon`, 7

`predict_hubs`, 7

`sample_graphon`, 8

`sample_mixed_graph`, 8

`scale_graphon`, 9