

# Package: somhca (via r-universe)

January 24, 2025

**Type** Package

**Title** Self-Organising Maps Coupled with Hierarchical Cluster Analysis

**Version** 0.1.3

**Description** Implements self-organising maps combined with hierarchical cluster analysis (SOM-HCA) for clustering and visualization of high-dimensional data. The package includes functions to estimate the optimal map size based on various quality measures and subsequently generates a model with the selected dimensions. It also performs hierarchical clustering on the map nodes to group similar units. Documentation about the SOM-HCA method is provided in Pastorelli et al. (2024) [doi:10.1002/xrs.3388](https://doi.org/10.1002/xrs.3388).

**License** MIT + file LICENSE

**Encoding** UTF-8

**Depends** dplyr, kohonen, aweSOM, maptree, RColorBrewer

**RoxygenNote** 7.3.2

**Config/pak/sysreqs** make libicu-dev libx11-dev zlib1g-dev

**Repository** <https://gianluca-pastorelli.r-universe.dev>

**RemoteUrl** <https://github.com/gianluca-pastorelli/somhca>

**RemoteRef** HEAD

**RemoteSha** 3d2555cc9000042b57b3e8818a13319355f2e6d5

## Contents

clusterSOM . . . . .	2
finalSOM . . . . .	3
generatePlot . . . . .	3
getClusterData . . . . .	4
optimalSOM . . . . .	5
readMatrix . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

`clusterSOM`*Perform Clustering on SOM Nodes*

---

### Description

Groups similar nodes of the SOM using hierarchical clustering and the KGS penalty function to determine the optimal number of clusters.

### Usage

```
clusterSOM(model, plot_result = TRUE, file_path = NULL)
```

### Arguments

<code>model</code>	The trained SOM model object.
<code>plot_result</code>	A logical value indicating whether to plot the clustering result. Default is 'TRUE'.
<code>file_path</code>	An optional string specifying the path to a CSV file. If provided, clusters are assigned to the observations in the original dataset, and the updated data is stored in a package environment as 'DataAndClusters'.

### Value

A plot of the clusters on the SOM grid (if 'plot\_result = TRUE'). If 'file\_path' is specified, the clustered dataset is stored in a package environment for retrieval.

### Examples

```
# Create a toy matrix with 9 columns and 100 rows
data <- matrix(rnorm(900), ncol = 9, nrow = 100) # 900 random numbers, 100 rows, 9 columns

# Run the finalSOM function with the mock data
model <- finalSOM(data, dimension = 6, iterations = 700)

# Perform clustering using the mock model
clusterSOM(model, plot_result = TRUE)

# Load the toy data from the package's inst/extdata/ directory, perform
# clustering and retrieve the clustered dataset
file_path <- system.file("extdata", "toy_data.csv", package = "somhca")
clusterSOM(model, plot_result = FALSE, file_path)
getClusterData()
```

---

`finalSOM`*Train Final SOM Model*

---

**Description**

Re-trains the SOM using a specified optimal grid size and number of iterations.

**Usage**

```
finalSOM(data, dimension, iterations)
```

**Arguments**

<code>data</code>	The preprocessed data matrix containing the input data for SOM training.
<code>dimension</code>	An integer specifying the dimension of the square SOM grid (e.g., 5 results in a 5x5 grid).
<code>iterations</code>	An integer defining the number of iterations for training the SOM model. Use a large value, e.g., 500 or higher, for improved training (an error message could suggest that reducing the number of iterations might be necessary).

**Value**

A trained SOM model object.

**Examples**

```
# Create a toy matrix with 9 columns and 100 rows
data <- matrix(rnorm(900), ncol = 9, nrow = 100) # 900 random numbers, 100 rows, 9 columns

# Run the finalSOM function with the mock data
myFinalSOM <- finalSOM(data, dimension = 6, iterations = 700)
```

---

`generatePlot`*Generate SOM Visualization Plots*

---

**Description**

Creates various types of plots to visualize and evaluate the trained SOM model.

**Usage**

```
generatePlot(model, plot_type, data = NULL)
```

**Arguments**

model	The trained SOM model object.
plot_type	An integer specifying the type of plot to generate. Options are: <ol style="list-style-type: none"> <li>1 Training progress plot (changes during training).</li> <li>2 Node count plot (number of samples mapped to each node) for assessing map quality.</li> <li>3 U-matrix plot (visualizing similarities between neighboring nodes).</li> <li>4 Weight vector plot (patterns in the distributions of variables).</li> <li>5 Kohonen heatmaps for all variables in the dataset (distribution of single variables across the map).</li> </ol>
data	The preprocessed data matrix containing the input data. Required only for 'plot_type = 5'.

**Value**

A plot or a series of plots is generated and displayed based on the specified type.

**Examples**

```
# Create a toy matrix with 9 columns and 100 rows
data <- matrix(rnorm(900), ncol = 9, nrow = 100) # 900 random numbers, 100 rows, 9 columns

# Assign column names to the data matrix
colnames(data) <- paste("Var", 1:ncol(data), sep = "_")

# Run the finalSOM function with the mock data
model <- finalSOM(data, dimension = 6, iterations = 700)

# Generate plots using the mock model
generatePlot(model, plot_type = 2)
generatePlot(model, plot_type = 5, data)
```

---

getClusterData	<i>Retrieve Clustered Data</i>
----------------	--------------------------------

---

**Description**

Access the dataset with cluster assignments stored by 'clusterSOM'.

**Usage**

```
getClusterData()
```

**Value**

A data frame with the clustered dataset.

---

`optimalSOM`*Estimate Optimal SOM Grid Size*

---

### Description

Computes the optimal grid size for training a SOM using various quality measures and heuristic approaches.

### Usage

```
optimalSOM(data, method = "A", increments, iterations)
```

### Arguments

<code>data</code>	The preprocessed data matrix containing the input data for SOM training.
<code>method</code>	A character string indicating the method for estimating the maximum grid dimension. Options are:  "A" Uses the heuristic formula by Vesanto et al. (default). "B" Applies an alternative heuristic approach. <b>numeric</b> Manually specified maximum dimension.
<code>increments</code>	An integer specifying the step size for increasing grid dimensions. For example, set increments to 2 or 5 to increment the grid size by 2 or 5 rows/columns at each step. Smaller increments lead to more granular searches but may increase computation time; larger increments risk errors if they exceed the estimated maximum SOM grid dimensions.
<code>iterations</code>	An integer defining the number of iterations for SOM training. A lower value, such as less than 500, helps reduce computation time. If the process takes too long or an error occurs, try reducing the number of iterations for quicker results.

### Value

A data frame summarizing quality measures and their associated optimal grid dimensions. Use these results to select the most suitable grid size for your SOM.

### Examples

```
# Create a toy matrix with 9 columns and 100 rows
data <- matrix(rnorm(900), ncol = 9, nrow = 100) # 900 random numbers, 100 rows, 9 columns

# Run the optimalSOM function with the mock data
myOptimalSOM <- optimalSOM(data, method = "A", increments = 2, iterations = 300)
```

---

`readMatrix`*Read a CSV File and Convert to a Matrix*

---

**Description**

Reads data from a CSV file, optionally removes row headings, and applies specified normalization methods before converting the data to a matrix. In the original dataset, rows represent observations (e.g., samples), columns represent variables (e.g., features), and all cells (except for column headers and, in case, row headers) only contain numeric values.

**Usage**

```
readMatrix(file_path, remove_row_headings = FALSE, scaling = "no")
```

**Arguments**

`file_path` A string specifying the path to the CSV file.

`remove_row_headings` A logical value. If 'TRUE', removes the first column of the dataset. This is useful when the first column contains non-numeric identifiers (e.g., sample names) that should be excluded from the analysis. Default is 'FALSE'.

`scaling` A string specifying the scaling method. Options are:  
"no" No scaling is applied (default).  
"SimpleFeature" Each column is divided by its maximum value.  
"MinMax" Each column is scaled to range [0, 1].  
"ZScore" Each column is Z-score standardized.

**Value**

A matrix with the processed data.

**Examples**

```
# Load the toy data from the package's inst/extdata/ directory
file_path <- system.file("extdata", "toy_data.csv", package = "somhca")

# Run the readMatrix function with the mock data
myMatrix <- readMatrix(file_path, TRUE, "MinMax")
```

# Index

clusterSOM, 2

finalSOM, 3

generatePlot, 3

getClusterData, 4

optimalSOM, 5

readMatrix, 6