

# Package ‘longmixr’

May 8, 2026

**Title** Longitudinal Consensus Clustering with 'flexmix'

**Version** 1.0.0

**Description** An adaption of the consensus clustering approach from 'ConsensusClusterPlus' for longitudinal data. The longitudinal data is clustered with flexible mixture models from 'flexmix', while the consensus matrices are hierarchically clustered as in 'ConsensusClusterPlus'. By using the flexibility from 'flexmix' and 'FactoMineR', one can use mixed data types for the clustering.

**License** GPL (>= 2)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**URL** <https://cellmapslab.github.io/longmixr/>

**BugReports** <https://github.com/cellmapslab/longmixr/issues>

**Depends** R (>= 3.5.0)

**Imports** checkmate, ConsensusClusterPlus, graphics, grDevices, flexmix, StatMatch, stats, utils

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, dplyr, tidyr, ggplot2, ggalluvial, FactoMineR, factoextra, lme4, purrr

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Jonas Hagenberg [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-1849-1106>>),  
Matt Wilkerson [aut, cph],  
Peter Waltman [aut, cph],  
Max Planck Institute of Psychiatry [cph]

**Maintainer** Jonas Hagenberg <jonas\_hagenberg@psych.mpg.de>

**Repository** CRAN

**Date/Publication** 2022-01-13 20:32:42 UTC

## Contents

crosssectional_consensus_cluster . . . . .	2
fake_questionnaire_data . . . . .	3
get_clusters . . . . .	4
longitudinal_consensus_cluster . . . . .	5
plot.lcc . . . . .	7
test_clustering_methods . . . . .	8

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

---

crosssectional\_consensus\_cluster

*Cross-sectional clustering with categorical variables*

---

### Description

This function uses the ConsensusClusterPlus function from the package with the same name with defaults for clustering data with categorical variables. As the distance function, the Gower distance is used.

### Usage

```
crosssectional_consensus_cluster(
  data,
  reps = 1000,
  finalLinkage = "ward.D2",
  innerLinkage = "ward.D2",
  ...
)
```

### Arguments

data	a matrix or data.frame containing variables that should be used for computing the distance. This argument is passed to StatMatch::gower.dist
reps	number of repetitions, same as in ConsensusClusterPlus
finalLinkage	linkage method for final clustering, same as in ConsensusClusterPlus same as in ConsensusClusterPlus
innerLinkage	linkage method for clustering steps, same as in ConsensusClusterPlus
...	other arguments passed to ConsensusClusterPlus, attention: the d argument can <b>not</b> be set as it is directly computed by crosssectional_consensus_cluster

### Details

data can take all input data types that `gower.dist` can handle, i.e. numeric, character/factor, ordered and logical.

**Value**

The output is produced by ConsensusClusterPlus

**Examples**

```
dc <- mtcars
# scale continuous variables
dc <- sapply(mtcars[, 1:7], scale)
# code factor variables
dc <- cbind(as.data.frame(dc),
            vs = as.factor(mtcars$vs),
            am = as.factor(mtcars$am),
            gear = as.factor(mtcars$gear),
            carb = as.factor(mtcars$carb))
cc <- crosssectional_consensus_cluster(
  data = dc,
  reps = 10,
  seed = 1
)
```

---

fake\_questionnaire\_data

*Fake questionnaire data*

---

**Description**

A simulated data set containing observations of 100 individuals at four time points. The data was simulated in two groups (50 individuals each) and contains two questionnaires with five items each, one questionnaire with five continuous variables and one additional cross-sectional continuous variable. In this data set the group variable from the simulation is included. You typically don't have this group variable in your data.

**Usage**

fake\_questionnaire\_data

**Format**

A data frame with 400 rows and 20 variables:

**ID** patient ID

**visit** time point of the observation

**group** to which simulated group the observation belongs to

**age\_visit\_1** age of the patient at time point 1

**single\_continuous\_variable** a cross-sectional continuous variable, i.e. there is only one unique value per individual

**questionnaire\_A\_1** the first item of questionnaire A with categories 1 to 5

**questionnaire\_A\_2** the second item of questionnaire A with categories 1 to 5  
**questionnaire\_A\_3** the third item of questionnaire A with categories 1 to 5  
**questionnaire\_A\_4** the fourth item of questionnaire A with categories 1 to 5  
**questionnaire\_A\_5** the fifth item of questionnaire A with categories 1 to 5  
**questionnaire\_B\_1** the first item of questionnaire B with categories 1 to 5  
**questionnaire\_B\_2** the second item of questionnaire B with categories 1 to 5  
**questionnaire\_B\_3** the third item of questionnaire B with categories 1 to 5  
**questionnaire\_B\_4** the fourth item of questionnaire B with categories 1 to 5  
**questionnaire\_B\_5** the fifth item of questionnaire B with categories 1 to 5  
**questionnaire\_C\_1** the first continuous variable of questionnaire C  
**questionnaire\_C\_2** the second continuous variable of questionnaire C  
**questionnaire\_C\_3** the third continuous variable of questionnaire C  
**questionnaire\_C\_4** the fourth continuous variable of questionnaire C  
**questionnaire\_C\_5** the fifth continuous variable of questionnaire C

### Source

simulated data

---

get_clusters	<i>Extract the cluster assignments</i>
--------------	--

---

### Description

This functions extracts the cluster assignments from an lcc object. One can specify which for which number of clusters the assignments should be returned.

### Usage

```
get_clusters(cluster_solution, number_clusters = NULL)
```

### Arguments

**cluster\_solution**  
 an lcc object

**number\_clusters**  
 default is NULL to return all assignments. Otherwise specify a numeric vector with the number of clusters for which the assignments should be returned, e.g. 2:4

**Value**

a data.frame with an ID column (the name of the ID column was specified by the user when calling the longitudinal\_consensus\_cluster) function and one column with cluster assignments for every specified number of clusters. Only the assignments included in number\_clusters are returned in the form of columns with the names assignment\_num\_clus\_x

**Examples**

```
# not run
set.seed(5)
test_data <- data.frame(patient_id = rep(1:10, each = 4),
  visit = rep(1:4, 10),
  var_1 = c(rnorm(20, -1), rnorm(20, 3)) +
  rep(seq(from = 0, to = 1.5, length.out = 4), 10),
  var_2 = c(rnorm(20, 0.5, 1.5), rnorm(20, -2, 0.3)) +
  rep(seq(from = 1.5, to = 0, length.out = 4), 10))
model_list <- list(flexmix::FLXMRmgcv(as.formula("var_1 ~ .")),
  flexmix::FLXMRmgcv(as.formula("var_2 ~ .")))
clustering <- longitudinal_consensus_cluster(
  data = test_data,
  id_column = "patient_id",
  max_k = 2,
  reps = 3,
  model_list = model_list,
  flexmix_formula = as.formula("~s(visit, k = 4) | patient_id"))
cluster_assignments <- get_clusters(clustering, number_clusters = 2)
# end not run
```

---

longitudinal\_consensus\_cluster

*Longitudinal consensus clustering with flexmix*

---

**Description**

This function performs longitudinal clustering with flexmix. To get robust results, the data is subsampled and the clustering is performed on this subsample. The results are combined in a consensus matrix and a final hierarchical clustering step performed on this matrix. In this, it follows the approach from the ConsensusClusterPlus package.

**Usage**

```
longitudinal_consensus_cluster(
  data = NULL,
  id_column = NULL,
  max_k = 3,
  reps = 10,
  p_item = 0.8,
  model_list = NULL,
```

```

flexmix_formula = as.formula("~s(visit, k = 4) | patient_id"),
title = "untitled_consensus_cluster",
final_linkage = c("average", "ward.D", "ward.D2", "single", "complete", "mcquitty",
  "median", "centroid"),
seed = 3794,
verbose = FALSE
)

```

### Arguments

<code>data</code>	a <code>data.frame</code> with one or several observations per subject. It needs to contain one column that specifies to which subject the entry (row) belongs to. This ID column is specified in <code>id_column</code> . Otherwise, there are no restrictions on the column names, as the model is specified in <code>flexmix_formula</code> .
<code>id_column</code>	name (character vector) of the ID column in <code>data</code> to identify all observations of one subject
<code>max_k</code>	maximum number of clusters, default is 3
<code>reps</code>	number of repetitions, default is 10
<code>p_item</code>	fraction of samples contained in subsampled sample, default is 0.8
<code>model_list</code>	either one <code>flexmix</code> driver or a list of <code>flexmix</code> drivers of class <code>FLXMR</code>
<code>flexmix_formula</code>	a formula object that describes the <code>flexmix</code> model relative to the formula in the <code>flexmix</code> drivers (the dot in the <code>flexmix</code> drivers is replaced, see the example). That means that you usually only specify the right-hand side of the formula here. However, this is not enforced or checked to give you more flexibility over the <code>flexmix</code> interface
<code>title</code>	name of the clustering; used if <code>writeTable = TRUE</code>
<code>final_linkage</code>	linkage used for the last hierarchical clustering step on the consensus matrix; has to be <code>average</code> , <code>ward.D</code> , <code>ward.D2</code> , <code>single</code> , <code>complete</code> , <code>mcquitty</code> , <code>median</code> or <code>centroid</code> . The default is <code>average</code>
<code>seed</code>	seed for reproducibility
<code>verbose</code>	boolean if status messages should be displayed. Default is <code>FALSE</code>

### Details

The data types `longitudinal_consensus_cluster` can handle depends on how the `flexmix` models are set up, in principle all data types are supported for which there is a `flexmix` driver with the desired outcome variable.

If you follow the dimension reduction approach outlined in `vignette("Example clustering analysis", package = "longmixr")`, the input data types depend on what FAMD from the `FactoMineR` package can handle. FAMD accepts numeric variables and treats all other variables as factor variables which it can handle as well.

**Value**

An object (list) of class lcc with length maxk. The first entry general\_information contains the entries:

consensus\_matrices a list of all consensus matrices (for all specified clusters)

cluster\_assignments a data.frame with an ID column named after id\_column and a column for every specified number

call the call/all arguments how longitudinal\_consensus\_cluster was called

The other entries correspond to the number of specified clusters (e.g. the second entry corresponds to 2 specified clusters) and each contains a list with the following entries:

consensus\_matrix the consensus matrix

consensus\_tree the result of the hierarchical clustering on the consensus matrix

consensus\_class the resulting class for every observation

found\_flexmix\_clusters a vector of the actual found number of clusters by flexmix (which can deviate from the specified number)

**Examples**

```
set.seed(5)
test_data <- data.frame(patient_id = rep(1:10, each = 4),
  visit = rep(1:4, 10),
  var_1 = c(rnorm(20, -1), rnorm(20, 3)) +
  rep(seq(from = 0, to = 1.5, length.out = 4), 10),
  var_2 = c(rnorm(20, 0.5, 1.5), rnorm(20, -2, 0.3)) +
  rep(seq(from = 1.5, to = 0, length.out = 4), 10))
model_list <- list(flexmix::FLXMRmgcv(as.formula("var_1 ~ .")),
  flexmix::FLXMRmgcv(as.formula("var_2 ~ .")))
clustering <- longitudinal_consensus_cluster(
  data = test_data,
  id_column = "patient_id",
  max_k = 2,
  reps = 3,
  model_list = model_list,
  flexmix_formula = as.formula("~s(visit, k = 4) | patient_id"))
# not run
# plot(clustering)
# end not run
```

---

plot.lcc

*Plot a longitudinal consensus clustering*


---

**Description**

Plot a longitudinal consensus clustering

**Usage**

```
## S3 method for class 'lcc'
plot(x, color_palette = NULL, ...)
```

**Arguments**

`x` lcc object (output from [longitudinal\\_consensus\\_cluster](#))  
`color_palette` optional character vector of colors for consensus matrix  
`...` additional parameters for plotting; currently not used

**Value**

Plots the following plots:

consensus matrix legend	the legend for the following consensus matrix plots
consensus matrix plot	for every specified number of clusters, a heatmap of the consensus matrix and the result of the fit
consensus CDF	a line plot of the CDFs for all different specified numbers of clusters
Delta area	elbow plot of the difference in the CDFs between the different numbers of clusters
tracking plot	cluster assignment of the subjects throughout the different cluster solutions
item-consensus	for every item (subject), calculate the average consensus value with all items that are assigned to
cluster-consensus	every bar represents the average pair-wise item-consensus within one consensus cluster

---

test\_clustering\_methods

*Try out different linkage methods*

---

**Description**

In the final step, the consensus clustering performs a hierarchical clustering step on the consensus cluster. This function tries out different linkage methods and returns the corresponding clusterings. The outputs can be plotted like the results from [longitudinal\\_consensus\\_cluster](#).

**Usage**

```
test_clustering_methods(
  results,
  use_methods = c("average", "ward.D", "ward.D2", "single", "complete", "mcquitty",
    "median", "centroid")
)
```

**Arguments**

results	clustering result of class lcc
use_methods	character vector of one or several items of average, ward.D, ward.D2, single, complete, mcquitty, median or centroid

**Value**

a list of elements, each element of class lcc. The entries are named after the used linkage method.

**Examples**

```
set.seed(5)
test_data <- data.frame(patient_id = rep(1:10, each = 4),
  visit = rep(1:4, 10),
  var_1 = c(rnorm(20, -1), rnorm(20, 3)) +
  rep(seq(from = 0, to = 1.5, length.out = 4), 10),
  var_2 = c(rnorm(20, 0.5, 1.5), rnorm(20, -2, 0.3)) +
  rep(seq(from = 1.5, to = 0, length.out = 4), 10))
model_list <- list(flexmix::FLXMRmgcv(as.formula("var_1 ~ .")),
  flexmix::FLXMRmgcv(as.formula("var_2 ~ .")))
clustering <- longitudinal_consensus_cluster(
  data = test_data,
  id_column = "patient_id",
  max_k = 2,
  reps = 3,
  model_list = model_list,
  flexmix_formula = as.formula("~s(visit, k = 4) | patient_id"))

clustering_linkage <- test_clustering_methods(results = clustering,
  use_methods = c("average", "single"))
# not run
# plot(clustering_linkage[["single"]])
# end not run
```

# Index

## \* datasets

- fake\_questionnaire\_data, 3
- crosssectional\_consensus\_cluster, 2
- fake\_questionnaire\_data, 3
- get\_clusters, 4
- gower.dist, 2
- longitudinal\_consensus\_cluster, 5, 8
- plot.lcc, 7
- test\_clustering\_methods, 8