

# Package ‘ihclust’

May 8, 2026

**Type** Package

**Title** Iterative Hierarchical Clustering (IHC)

**Version** 0.1.0

**Description** Provides a set of tools to

- i) identify geographic areas with significant change over time in drug utilization, and
- ii) characterize common change over time patterns among the time series for multiple geographic areas.

For reference, see below:

1. Song, J., Carey, M., Zhu, H., Miao, H., Ram ´irez, J. C., & Wu, H. (2018) <doi:10.1504/IJCDD.2018.10011910>
2. Wu, S., Wu, H. (2013) <doi:10.1186/1471-2105-14-6>
3. Carey, M., Wu, S., Gan, G. & Wu, H. (2016) <doi:10.1016/j.idm.2016.07.001>.

**License** GNU General Public License (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.2

**Depends** R (>= 3.5.0)

**Imports** factoextra, splines, ggplot2, foreach, doParallel, parallel

**NeedsCompilation** no

**Author** Elin Cho [aut, cre],

Yuting Xu [aut],

Jaejoon Song [aut]

**Maintainer** Elin Cho <elincho524@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-04-27 07:20:02 UTC

## Contents

ihclust . . . . .	2
opioidData . . . . .	3
simcurve . . . . .	4
testchange . . . . .	5

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

---

`ihclust`*Iterative Hierarchical Clustering (IHC)*

---

**Description**

This function identifies inhomogeneous clusters using iterative hierarchical clustering (IHC) method.

**Usage**

```
ihclust(  
  data,  
  smooth = TRUE,  
  cor_criteria = 0.75,  
  max_iteration = 100,  
  verbose = TRUE  
)
```

**Arguments**

<code>data</code>	a numeric matrix, each row representing a time-series and each column representing a time point
<code>smooth</code>	if <code>smooth = 'TRUE'</code> , a smooth function is applied before clustering
<code>cor_criteria</code>	pre-specified correlation criteria
<code>max_iteration</code>	maximum number of iterations
<code>verbose</code>	if <code>verbose = 'TRUE'</code> , the result of a progress is printed

**Details**

`ihclust`

The IHC algorithm implements the three steps as outlined below. First, the Initialization step clusters the data using hierarchical clustering. Second, cluster centers are obtained as an average of all the data points in the cluster. The Merging step considers each of the cluster centers (exemplars) as 'new data point', and use the same procedure described in the Initialization step to merge the exemplars into a new set of clusters. Third, the Pruning step streamlines the clusters and removes inconsistencies by reassessing the cluster membership by each data point.

**Value**

Output from the function is a list of three items:

- `Cluster_Label` - the cluster label for each data point
- `Num_Iterations` - total number of iterations
- `Unique_Clusters_in_Iteration` - unique clusters in each iteration

## References

1. Song, J., Carey, M., Zhu, H., Miao, H., Ram´irez, J. C., & Wu, H. (2018). Identifying the dynamic gene regulatory network during latent HIV-1 reactivation using high-dimensional ordinary differential equations. *International Journal of Computational Biology and Drug Design*, 11,135-153. doi: 10.1504/IJCBDD.2018.10011910.
2. Wu, S., & Wu, H. (2013). More powerful significant testing for time course gene expression data using functional principal component analysis approaches. *BMC Bioinformatics*, 14:6.
3. Carey, M., Wu, S., Gan, G. & Wu, H. (2016). Correlation-based iterative clustering methods for time course data: The identification of temporal gene response modules for influenza infection in humans. *Infectious Disease Modeling*, 1, 28-39.

## Examples

```
# This is an example not using the permutation approach

opioid_data_noNA <- opioidData[complete.cases(opioidData), ] #remove NAs

mydata <- as.matrix(opioid_data_noNA[1:500,4:18])

testchange_results <- testchange(data=mydata,perm=FALSE,time=seq(1,15,1))

data_change <- testchange_results$sig.change

clustering_results <- ihclust(data=data_change, smooth = TRUE,

cor_criteria = 0.75, max_iteration = 100, verbose = TRUE)
```

---

opioidData

*Opioid Dispensing Rates*

---

## Description

A dataset containing estimated opioid dispensing rate per 100 persons in United States, 2006-2020.

## Usage

```
data(opioidData)
```

## Format

data.frame; columns: fips = FIPS county code, State = State, County = County, X2006-X2020 = estimated opioid dispensing rate per 100 persons in each year

## Source

<https://www.cdc.gov/drugoverdose/rxrate-maps/index.html>

---

simcurve	<i>simcurve</i>
----------	-----------------

---

### Description

This function generates two kinds of datasets. 1. Randomly generates curves with change/no change. 2. Generates true curves assumed from fixed coefficients with some random noise.

### Usage

```
simcurve(numareas = c(300, 300, 300), p = 0.05, type, normerr = 0.1)
```

### Arguments

numareas	number of areas to generate
p	proportion of the areas that have significant change
type	type of curves generated
normerr	standard deviation of the Normal distribution (with mean zero) of which the coefficients are generated

### Details

If type = "random", the function generates curves with change/no change. If type = "fixed", the function generates true curves assumed from fixed coefficients with some random noise. If numareas is not specified, it is assumed as a vector of c(300,300,300). If normerr is not specified, it is assumed as a value of 0.01. It is ignored when type= "random".

### Value

Output from the function is a list of two items:

- data - simulated data
- parameters - parameters used to generate the data

### Examples

```
mydata_ran <- simcurve(numareas = c(300, 300, 300), p=0.01, type="random")
```

```
mydata_fixed <- simcurve(numareas = c(300, 300, 300), p=0.01, type="fixed", normerr = 0.1)
```

---

testchange	<i>testchange</i>
------------	-------------------

---

### Description

This function identifies geographic areas with significant change over time.

### Usage

```
testchange(data, time, perm = FALSE, nperm = 100, numclust = 4, topF = 500)
```

### Arguments

data	a numeric matrix, each row representing a time-series and each column representing a time point
time	defines the time sequence
perm	if perm = 'TRUE', a permutation is performed
nperm	number of permutations
numclust	defines the number of clusters for the parallel processing
topF	number of top F values to be selected when perm = 'FALSE'

### Details

number of permutations of  $\geq 10,000$  is ideal

### Value

Output if perm = 'TRUE' is a list of three items:

- perm.F - F values obtained from permutation tests
- p.values - p-values obtained from permutation tests
- p.adjusted - p-values adjusted by Benjamini-Hochberg method

Output if perm = 'False' is a list of three items:

- obs.F - conventional F-statistic values
- sig.change - areas with significant change over time pattern selected by top F-statistic values
- sel.F - top F-statistic values selected

## References

1. Song, J., Carey, M., Zhu, H., Miao, H., Ramírez, J. C., & Wu, H. (2018). Identifying the dynamic gene regulatory network during latent HIV-1 reactivation using high-dimensional ordinary differential equations. *International Journal of Computational Biology and Drug Design*, 11,135-153. doi: 10.1504/IJCBDD.2018.10011910.
2. Wu, S., & Wu, H. (2013). More powerful significant testing for time course gene expression data using functional principal component analysis approaches. *BMC Bioinformatics*, 14:6.
3. Carey, M., Wu, S., Gan, G. & Wu, H. (2016). Correlation-based iterative clustering methods for time course data: The identification of temporal gene response modules for influenza infection in humans. *Infectious Disease Modeling*, 1, 28-39.

## Examples

```
# This is an example not using the permutation approach

opioid_data_noNA <- opioidData[complete.cases(opioidData), ] #remove NAs

mydata <- as.matrix(opioid_data_noNA[,4:18])

testchange_results <- testchange(data=mydata,perm=FALSE,time=seq(1,15,1))
```

# Index

\* **datasets**

    opioidData, 3

ihclust, 2

opioidData, 3

simcurve, 4

testchange, 5