

# Package: ccml (via r-universe)

October 13, 2024

**Type** Package

**Title** Consensus Clustering for Different Sample Coverage Data

**Version** 1.4.0

**Description** Consensus clustering, also called meta-clustering or cluster ensembles, has been increasingly used in clinical data. Current consensus clustering methods tend to ensemble a number of different clusters from mathematical replicates with similar sample coverage. As the fact of common variety of sample coverage in the real-world data, a new consensus clustering strategy dealing with such biological replicates is required. This is a two-step consensus clustering package, which is used to input multiple predictive labels with different sample coverage (missing labels).

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.1

**Depends** R (>= 3.5.0)

**Imports** ggplot2, diceR, parallel, tidyr, SNFtool, plyr,  
ConsensusClusterPlus (>= 1.56.0)

**Suggests** spelling, testthat (>= 3.0.0)

**Language** en-US

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Chuanxing Li [aut, cre], Meng Zhou [aut]

**Maintainer** Chuanxing Li <chuan-xing.li@ki.se>

**Date/Publication** 2023-08-30 06:10:02 UTC

**Repository** <https://pulmonomics-lab.r-universe.dev>

**RemoteUrl** <https://github.com/pulmonomics-lab/ccml>

**RemoteRef** HEAD

**RemoteSha** d4fd7fa002c16eb03ec8b1d41dad6feaf9649c7a

## Contents

callNCW . . . . .	2
ccml . . . . .	3
example_data . . . . .	5
plotCompareCW . . . . .	5
randConsensusMatrix . . . . .	6
spectralClusteringAffinity . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

callNCW	<i>Calculate normalized consensus weight(NCW) matrix based on permutation.</i>
---------	--

---

### Description

Calculate normalized consensus weight(NCW) matrix based on permutation.

### Usage

```
callNCW(
  title = "",
  label,
  nperm = 10,
  ncore = 1,
  seedn = 100,
  stability = TRUE,
  plot = NULL
)
```

### Arguments

title	A character value for output directory. Directory is created only if not existed. This title can be an absolute or relative path.
label	A matrix or data frame of input labels, columns=different clustering results and rows are samples.
nperm	A integer value of the permutation numbers, or nperm=10(default), which means nperm*1000 times of permutation.
ncore	A integer value of cores to use, or ncore=1 (default). It's the input core numbers for the parallel computation in this package parallel.
seedn	A numerical value to set the start random seed for reproducible results, or seedn=100 (default). For every 1000 iteration, the seed will +1 to get repeat results.
stability	A logical value. Should estimate the stability of normalized consensus weight based on permutation numbers (default stability=TRUE), or not?
plot	character value. NULL(default) - print to screen, 'pdf', 'png', 'pngBMP' for bitmap png, helpful for large datasets, or 'pdf'. Input for randConsensusMatrix.

**Value**

A matrix of normalized consensus weights.

**Examples**

```
# load data
data(example_data)
label=example_data

# if plot is not NULL, results will be saved in "result_output" directory
title="result_output"

# run ncw
ncw<-callNCW(title=title,label=label,stability=TRUE,nperm=4,ncore=1)
```

---

ccml

*A two-step consensus clustering inputing multiple predictive labels  
with different sample coverages (missing labels)*

---

**Description**

A two-step consensus clustering inputing multiple predictive labels with different sample coverages (missing labels)

**Usage**

```
ccml(
  title,
  label,
  output = "rdata",
  nperm = 10,
  ncore = 1,
  seedn = 100,
  stability = TRUE,
  maxK = 15,
  reps = 1000,
  pItem = 0.9,
  plot = NULL,
  clusterAlg = "spectralClusteringAffinity",
  innerLinkage = "complete",
  ...
)
```

**Arguments**

title	A character value for output directory. Directory is created only if not existed. This title can be an absolute or relative path. Input for <code>callNCW</code> , <code>plotCompareCW</code> , <code>ConsensusClusterPlus::ConsensusClusterPlus</code> , <code>ConsensusClusterPlus::calcICL</code>
label	A matrix or data frame of input labels or a character value of input file name, columns=different clustering results and rows are samples. label could be imported as <code>'rdata'</code> , <code>'rda'</code> , or <code>'csv'</code> . Input for <code>callNCW</code> , <code>plotCompareCW</code>
output	A character value for output format, or "rdata"(default) as save to <code>.rdata</code> when both output and plot are not NULL, others will return to workspace.
nperm	A integer value of the permutation numbers, or <code>nperm=10</code> (default), which means <code>nperm*1000</code> times of permutation. Input for <code>callNCW</code>
ncore	A integer value of cores to use, or <code>ncore=1</code> (default). It's the input core numbers for the parallel computation in this package parallel. Input for <code>callNCW</code>
seedn	A numerical value to set the start random seed for reproducible results, or <code>seedn=100</code> (default). For every 1000 iteration, the seed will +1 to get repeat results. Input for <code>callNCW</code> , <code>ConsensusClusterPlus::ConsensusClusterPlus</code>
stability	A logical value. Should estimate the stability of normalized consensus weight based on permutation numbers (default <code>stability=TRUE</code> ), or not? Input for <code>callNCW</code>
maxK	integer value. maximum cluster number to evaluate. Input for <code>ConsensusClusterPlus::ConsensusClusterPlus</code> for the consensus clustering based on normalized consensus weights.
reps	integer value. number of subsamples. Input for <code>ConsensusClusterPlus::ConsensusClusterPlus</code>
pItem	numerical value. proportion of items to sample. Input for <code>ConsensusClusterPlus::ConsensusClusterPlus</code>
plot	character value. NULL(default) - print to screen, <code>'pdf'</code> , <code>'png'</code> , <code>'pngBMP'</code> for bitmap png, helpful for large datasets. Input for <code>ConsensusClusterPlus::ConsensusClusterPlus</code> , <code>ConsensusClusterPlus::calcICL</code> , <code>callNCW</code> , <code>plotCompareCW</code>
clusterAlg	character value. cluster algorithm. <code>'spectralClusteringAffinity'</code> for spectral clustering of similarity/affinity matrix(default), other methods for clustering of distance matrix, <code>'hc'</code> heirarchical ( <code>hclust</code> ), <code>'pam'</code> for partitioning around medoids, <code>'km'</code> for k-means upon data matrix, <code>'kmdist'</code> for k-means upon distance matrices (former <code>km</code> option), or a function that returns a clustering. Input for <code>ConsensusClusterPlus::ConsensusClusterPlus</code> .
innerLinkage	heirarchical linkage method for subsampling, or "complete"(default). Input for <code>ConsensusClusterPlus::ConsensusClusterPlus</code>
...	Other input arguments for <code>ConsensusClusterPlus::ConsensusClusterPlus</code>

**Value**

A list of three items

- `ncw` - A matrix of normalized consensus weights. Output from `callNCW`.
- `fcluster` - A list of length `maxK`. Each element is a list containing `consensusMatrix` (numerical matrix), `consensusTree` (`hclust`), `consensusClass` (consensus class assignments). `ConsensusClusterPlus` also produces images. Output from `ConsensusClusterPlus::ConsensusClusterPlus`
- `icl` a list of two elements `clusterConsensus` and `itemConsensus` corresponding to cluster-consensus and item-consensus. Output from `ConsensusClusterPlus::ConsensusClusterPlus`

**Examples**

```

# load data
data(example_data)
label=example_data

# if plot is not NULL, results will be saved in "result_output" directory
title="result_output"

# not estimate stability of permutation numbers.
res_1=ccml(title=title,label=label,nperm = 3,ncore=1,stability=FALSE,maxK=5,pItem=0.8)

# other methods for clustering of distance matrix
res_2<-ccml(title=title,label=label,nperm = 10,ncore=1,stability=TRUE,maxK=3,
            pItem=0.9,clusterAlg = "hc")

# set the start random seed
res_3<-ccml(title=title,label=label,output=FALSE,nperm = 5,ncore=1,seedn=150,stability=TRUE,maxK=3,
            pItem=0.9)

```

---

example_data	<i>The input data for example</i>
--------------	-----------------------------------

---

**Description**

In this matrix, columns represent the results of different clustering results and rows are samples.

**Usage**

```
example_data
```

**Format**

A matrix with 10 rows and 5 columns.

---

plotCompareCW	<i>Plot of original consensus weights vs. normalized consensus weights grouping by the number of co-appeared percent of clustering(non-missing).</i>
---------------	--

---

**Description**

Plot of original consensus weights vs. normalized consensus weights grouping by the number of co-appeared percent of clustering(non-missing).

**Usage**

```
plotCompareCW(title, label, ncw, plot = NULL)
```

**Arguments**

title	A character value for output directory.
label	A matrix or data frame of input labels, columns=different clustering results and rows are samples.
ncw	A matrix of normalized consensus weights with sample-by-sample as the order of sample(rows) in label.
plot	character value. NULL(default) - print to screen, 'pdf', 'png', 'pngBMP' for bitmap png, helpful for large datasets, or 'pdf'.

**Value**

A ggplot point in PDF format with x-axis: original consensus weights; y-axis: normalized consensus weights; color: percent of co-appeared of clustering; size: number of duplicates sample .

**Examples**

```
# load data
data(example_data)
label=example_data

# if plot is not NULL, results will be saved in "result_output" directory
title="result_output"

ncw<-callNCW(title=title,label=label,stability=TRUE)
plotCompareCW(title=title,label=label,ncw=ncw)
```

---

randConsensusMatrix	<i>Calculate consensus weight matrix based on the permuted input label matrix. Internal function used by callNCW</i>
---------------------	--

---

**Description**

Calculate consensus weight matrix based on the permuted input label matrix. Internal function used by callNCW

**Usage**

```
randConsensusMatrix(
  l.seed,
  l.label = label,
  l.ns = ns,
  l.nc = nc,
  l.nv = nv,
  l.index = index,
  l.pair.ind = pair.ind,
  l.ppath = ppath,
  l.plot = plot
)
```

**Arguments**

<code>l.seed</code>	A numerical value to set the random seed for reproducible results, 1000 random label matrix will be generated based on this seed number.
<code>l.label</code>	A matrix or data frame of input labels, columns=different clustering results and rows are samples.
<code>l.ns</code>	A integer value of number of samples, <code>=nrow(l.label)</code>
<code>l.nc</code>	A integer value of number of samples, <code>=ncol(l.label)</code>
<code>l.nv</code>	A integer vector of the number of non missing values for each column in <code>l.label</code>
<code>l.index</code>	A list of index with length of <code>l.nc</code> of non missing values for each column in <code>l.label</code>
<code>l.pair.ind</code>	A n-by-2 index matrix of array indices of upper triangular of <code>l.label</code> with non missing values
<code>l.ppath</code>	A character value for output directory.
<code>l.plot</code>	character value. NULL(default) - print to screen, 'pdf', 'png', 'pngBMP' for bitmap png, helpful for large datasets, or 'pdf'.

**Value**

A character of finished seed.

Write a binary file of 1000 random consensus weight matrix(as a vector n-by-1, `n = nrow(l.pair.ind)`) with the seed `l.seed`, output file name: `paste0("s",l.seed,"rcw")`.

---

spectralClusteringAffinity

*Perform spectral clustering algorithms for an affinity matrix, using SNFtool::spectralClustering.*

---

**Description**

Perform spectral clustering algorithms for an affinity matrix, using SNFtool::spectralClustering.

**Usage**

```
spectralClusteringAffinity(affi_matrix, k, type = 3)
```

**Arguments**

<code>affi_matrix</code>	A numerical similarity or affinity matrix.
<code>k</code>	A number value of clusters
<code>type</code>	The variants of spectral clustering to use. See <code>SNFtool::spectralClustering</code>

**Value**

A vector consisting of cluster labels of each sample.



# Index

## \* datasets

example\_data, 5

callNCW, 2

ccml, 3

example\_data, 5

plotCompareCW, 5

randConsensusMatrix, 6

spectralClusteringAffinity, 7