# CICA tutorial

Jeffrey Durieux, MSc

6/28/2021

## Introduction

This document contains a brief tutorial on how to use Clusterwise Independent Component Analysis (CICA) in R. This method enables the clustering of subjects based on similarities and differences in resting state networks (RSNs). A paper about this procedure is currently under review but you can find more information on this method by reading my master thesis: https://hdl.handle.net/1887/35077.

A CRAN version of the software package is also currently under review. However, you can already download my package via GitHub: https://github.com/jeffreydurieux/CICA. New additions and options will be added to this GitHub page during the summer of 2021.

## CICA model and loss function

In order to cluster subjects based on similarities and differences in underlying RSNs, the C-ICA model assumes that data of $I$ subjects $\mathbf{X}_i$ $(i = 1, \ldots, I)$ fall apart into $R$ mutually exclusive clusters, with each $\mathbf{X}_i$ containing the (rs-)fMRI data ($V$ voxels $\times$ $T$ time points) of subject $i$. Thus, C-ICA decomposes each $\mathbf{X}_i$ as:

$$\mathbf{X}_i = \sum_{r=1}^{R} p_{ir} \mathbf{S}^r \mathbf{A}_i^T + \mathbf{E}_i \tag{1}$$

where the elements $p_{ir}$ denote the entries from the binary partition matrix $\mathbf{P}$ $(I \times R)$ which equal 1 when subject $i$ is assigned to cluster $r$ and 0 otherwise. $\mathbf{A}_i$ $(T \times Q$, with $Q$ being the number of components, which is kept equal across subjects and clusters) denotes the mixing matrix (i.e., time courses) for subject $i$ and $\mathbf{S}^r$ $(V \times Q)$ represents a matrix where the columns contain the independent components or RSNs patterns for cluster $r$ $(r = 1, \ldots, R)$. Note that the cluster specific RSNs patterns in $\mathbf{S}^r$ are common group RSNs patterns that represent the data $\mathbf{X}_i$ for all subjects $i$ belonging to the cluster in question. The common components $\mathbf{S}^r$ are assumed to be non-normally distributed and mutually statistical (spatially) independent. Additionally, the model contains an error term $\mathbf{E}_i$ for each data block $i$.

The aim of C-ICA is to estimate a partitioning matrix $\mathbf{P}$, subject specific mixing matrices $\mathbf{A}_i$ and cluster specific independent component matrices or RSNs matrices $\mathbf{S}^r$. The estimation is done by minimizing the following loss function:

$$L = \sum_{i=1}^{I} ||\mathbf{X}_i - \sum_{r=1}^{R} p_{ir} \mathbf{S}^r \mathbf{A}_i||^2 \tag{2}$$

The current version of the CICA package (version 0.1.0) uses an alternating least squares type of algorithm. This procedure has problems with local minima so it is strongly advised to use a multiple random starts procedure.

## Download and install the CICA R package

The development version of CICA can be downloaded from my GitHub repository. This is easily done using the devtools package in R:

```
devtools::install_github(repo = 'jeffreydurieux/CICA')
```

This packages contains a very small simulated example data set (`ExampleData`). The data consists of a list object with 9 matrices (each refer to a subject). The matrices ($\mathbf{X}_i$) have a size of 100 voxels $\times$ 50 time points. Note that the true data generating model contains 3 equally sized clusters and 5 underlying components per cluster.

In order to apply CICA the following function should be used in R:

```
library(CICA)
```

```
## Loading required package: ica
```

```
## Loading required package: NMFN
```

```
output <- CICA(DataList = ExampleData, nStarts = 30, nComp = 5, nClus = 3,
               scale = TRUE, center = TRUE, verbose = FALSE)
```

Here a CICA model with the true underlying components and clusters is fitted to the data and 30 random starts were performed in order to avoid local minima. Additionally the argument `scale` was set to `TRUE` in order to have an equal sum of squares per data matrix. `center` was set to `TRUE` in order to mean center the voxels. If a user wants to view to loss function values per random start printed out to the console one could set the argument `verbose` equal to `TRUE`.

The output of the CICA function is a list object if class `CICA`. This list contains the estimated clustering, cluster specific component matrices and subject specific time course matrices. Additionally, information about the loss function value of the most optimal start and all random starts are provided:

```
str(output)
```

```
## List of 5
##  $ P         : int [1:9] 3 3 3 1 1 1 2 2 2
##  $ Sr        :List of 3
##   ..$ : num [1:1000, 1:5] -0.556 0.629 0.679 -0.449 0.975 ...
##   ..$ : num [1:1000, 1:5] 0.27 -0.512 -0.942 -1.119 -0.713 ...
##   ..$ : num [1:1000, 1:5] 0.3124 -0.4576 0.7688 0.0275 0.782 ...
##  $ Ais       :List of 9
##   ..$ : num [1:50, 1:5] -0.1097 -0.0153 0.0552 0.051 -0.027 ...
##   ..$ : num [1:50, 1:5] -0.0058 -0.0626 0.00204 -0.05388 -0.06644 ...
##   ..$ : num [1:50, 1:5] -0.02884 -0.07546 -0.04914 0.15512 -0.00759 ...
##   ..$ : num [1:50, 1:5] -0.03582 -0.02168 0.07399 -0.00139 -0.00349 ...
##   ..$ : num [1:50, 1:5] -0.06742 0.00167 -0.01524 -0.05269 0.02898 ...
##   ..$ : num [1:50, 1:5] 0.1336 0.0102 0.0354 0.0094 -0.0425 ...
##   ..$ : num [1:50, 1:5] 0.0241 0.013 -0.0498 0.084 0.0518 ...
##   ..$ : num [1:50, 1:5] -0.0192 0.0283 0.0298 0.0261 0.0105 ...
##   ..$ : num [1:50, 1:5] 0.0487 0.0748 -0.0363 0.1551 -0.1251 ...
##  $ Loss      : num 36.3
##  $ LossStarts: num [1:30] 36.3 36.3 36.3 36.3 36.3 ...
##  - attr(*, "class")= chr "CICA"
```

The current version of the CICA package contains a S3 summary function. This functions prints out a neat summary of the estimated clustering:

```
summary(output)
```

```
## Partitioning matrix P:
##
##       Cluster 1 Cluster 2 Cluster 3
##  [1,]         0         0         1
##  [2,]         0         0         1
##  [3,]         0         0         1
##  [4,]         1         0         0
##  [5,]         1         0         0
##  [6,]         1         0         0
##  [7,]         0         1         0
##  [8,]         0         1         0
##  [9,]         0         1         0
##
## Tabulation of clustering:
##
## Cluster 1 Cluster 2 Cluster 3
##         3         3         3
##
## Loss function value of optimal solution is:  36.27277
```

## Final remarks

The first version of the CICA package contains the CICA method using random starts. Future releases will contain the option to provide rational starts that will help the algorithm to converge to an optimal solution. The rational starts are based on the procedure from Durieux and Wilderjans (2019). For a tutorial on this procedure check out this short tutorial on my website http://jeffreydurieux.com/tutorial/P1_tutorial.pdf

Hope you like this very short tutorial and if you have any questions or suggestions please contact me via: j.durieux@fsw.leidenuniv.nl

## References

Durieux, Jeffrey, and Tom F. Wilderjans. 2019. "Partitioning Subjects Based on High-Dimensional fMRI Data: Comparison of Several Clustering Methods and Studying the Influence of ICA Data Reduction in Big Data." *Behaviormetrika* 46 (2): 271–311. https://doi.org/10.1007/s41237-019-00086-4.