**HELP FILE FOR THE USAGE OF MODIFIED N-BIC**

The following help file contains instructions for usage of Modified N-BiC, including prerequisites, required files, input parameters and final outputs.

**Prerequisites**

* MATLAB R2022A. (Tested on this version of MATLAB)

**Required function files**

* Main script file “Modified\_N\_BiC.m” (for calling other functions and displaying the output).
* *Sort\_load.p* is a function which sorts the data matrix according to four criterias mentioned in [1].
* *mintersect.m* is a function to perform intersection between multiple sets.

(MATLAB file exchange)

* *mat\_create.p*: this function can be used for creating a simulated dataset with certain specifications.
* *disp\_mat.p:* this function is used for displaying a matrix of interest using imagesc with certain specifications.
* *mat\_normalize:* function to normalize a matrix of interest to the range of 0-1.
* *Search\_Bics.p*: this function is used to search and evaluate the biclusters (using multiple other functions).( For custom loading matrix)
* *Search\_Bics\_sim.p:*  this function is used to search and evaluate the biclusters (For simulated matrix)
* *consensus.p*: calculates consensus score between ground truth and extracted biclusters. [1]
* *F1cons.p*: calculates F1 index between extracted bicluster and ground truth bicluster.
* *F1ind.p*: calculates F1 index between biclusters in the validation process.
* *Msr.p*: calculates mean square residue (MSR) of a bicluster. [2]

**Input parameters for the functions**

(Sample parameters for each of the functions are given in the main script)

The following parameters 1- 9 are stored in a data structure called Vars.

1. **Components {C1,C2,C3,…}-** Each one represents a component/ (column number) where the data matrix must be embedded. The user may define as many components as required. The components are stored in struct data structure called Vars.
2. **Subjects {S1,S2,S3,...}-** Each one represents a list of subjects pertaining to a bicluster. The user may define as many subjects as required. The subjects are stored in a struct data structure called Vars.
3. **Embedding -** Contains the embedding factor for embedding a bicluster in the data matrix.
4. **Start\_c -** Starting component number
5. **End\_c -** Ending component number
6. **Noc -** Number of components.
7. **ToS -** Number of subjects.
8. **Subs -** Minimum number of subjects required in a bicluster.
9. **Overlap -** Overlap value to be used for validation.

**Function call 1 (With Simulated Matrix creation)**

| [A1,ABics] = mat\_create(Vars); |
| --- |

**Output:**

**A1 -**  simulated data matrix returned by the function.

**ABics -** Ground truth biclusters.

**Input:**

**Vars -** struct data structure previously defined.

If the loading coefficients matrix is available, the previous step can be skipped.

The matrix should be assigned to the variable A1.

**Function call 2**

| [Sor\_Extreme,Sor\_Absolute,Sor\_Quartile,Sor\_posneg] =Sort\_load(A1,Vars.Noc,Vars.ToS); |
| --- |

**Output:**

**Sor\_Extreme -** Matrix sorted as per positive or negative method.

**Sor\_Absolute** - Matrix sorted as per absolute method.

**Sor\_Quartile** - Matrix sorted as per quartile method.

**Sor\_posneg** - Matrix sorted as per positive and negative method.

**Input:**

**A1 -** data matrix (simulated / loading matrix).

**Vars.Noc -** Number of components.

**Vars.ToS -** Total number of subjects.

**Function call 3**

| dispmat(Sorted\_mat,'Components','Subjects','Sorted Matrix Showing Biclusters') |
| --- |

**Output:**

Sorted matrix visualized using imagesc.

**Input:**

**Sorted\_mat -** Sorted matrix returned from any of the four methods.

**‘Components’** - x axis title

**‘Subjects’-** y axis title

**‘Sorted Matrix showing Biclusters’** - Title of the image.

**Function call 4**

| [BicMat, AvMSR, Bics,CS] = Search\_BICS(Sorted\_mat,Vars);  % This function call is used for custom loading matrix  [BicMat,AvMSR,Bics] = Search\_BICS\_sim(Sorted\_mat,Vars,ABics);  % This function call is used for simulated loading matrix |
| --- |

**Output:**

**BicMat -** Matrix with locations of possible biclusters.

**AvMSR -** Average MSR of the biclusters.

**Bics -** Table containing subjects and the corresponding components.

**CS -** Consensus score.

**Input:**

**Sorted\_mat -** Sorted matrix.

**ABics -** Ground truth biclusters.

**Vars -** struct data structure previously defined.

**Function call 5**

| dispmat(BicMat,'Components','Subjects','Matrix Showing Possible Biclusters'); |
| --- |

**Output:**

BicMat visualized using imagesc.

**Input:**

**BicMat -** Matrix with locations of possible biclusters.

**‘Components’** - x axis title

**‘Subjects’-** y axis title

**‘Matrix Showing Possible Biclusters’ -** Title of the image.

**Additional Instructions:**

* If you want to use a custom loading matrix, assign the loading matrix to the variable A1, change the parameters of interest, and choose option 1 when the script is run.
* If you want to simulate a loading matrix , change the parameters of interest and choose option 2 when the script is run.
* The script can be executed with the matrix of interest being assigned to variable A1 (Loading matrix for example).
* Please modify the input parameters as it befits the problem.

**References:**

1. Rahaman, M.A., Mathalon, D., Lee, H.J., et al.: N-BIC: A method for multi-component and symptom biclustering of structural MRI data: Application to schizophrenia. IEEE Transactions on Biomedical Engineering. 67, 110–121 (2020).
2. Pontes, B., Giráldez, R., Aguilar-Ruiz, J.S.: Biclustering on Expression Data: A Review. Journal of Biomedical Informatics. 57, 163–180 (2015).

