the Fast ECM toolbox

1. Download & Install, Command line

The latest version of the code is at `github.com/amwink/bias/tree/master/matlab/fastECM`. Downloading the bias repository `github.com/amwink/bias/archive/master.zip` is the easiest way to install `fastECM` and the easy-to-use graphical user interface `fegui`. In Linux you can do this:

with git:

```
$ git clone https://github.com/amwink/bias.git
```

or, with wget:

```
$ wget -c --no-check-certificate github.com/amwink/bias/archive/master.zip
$ unzip bias-master.zip
$ ln -fs bias-master bias
```

and continue in MatLab:

```
$ matlab -nodesktop -nosplash
>> addpath(genpath('bias'))
>> fegui
```

Example NIfTI images are available at `https://bias.googlecode.com/files/fastECM.zip`. Please make sure not to use the code from this archive as it is deprecated. It is best to extract the images and then delete the rest:

```
$ cd bias
$ wget https://bias.googlecode.com/files/fastECM.zip
$ unzip fastECM.zip
$ mv fastECM/*nii* matlab/fastECM
$ rm -rf fastECM*
```

With these files in the fastECM directory, the call

```
>> fastECM
```

with no arguments will produce a demo analysis, using the input fMRI file, the parenchymal mask and the brain region atlas provided, to show the syntax and the output. The minimal number of arguments is 1, and

```
>> fastECM('fmri4d.nii.gz')
```

produces the simplest analysis possible: one ECM map, no masking, voxelwise centrality. The rest of the arguments are explained via the GUI.

2 Graphical user interface

The GUI has 2 tabs called 'Files' (*see Fig 1a*) and 'Settings' (*see Fig 1b*). The left side of the Files tab contains four buttons: one to start the fECM computation, one to add input files, one for specifying an atlas (to do regional centrality analysis) and one for specifying a mask. The right side shows the list of input files to be analysed.

The input file dialog supports the addition of NIfTI and Analyze files to the list (also compressed) by selecting one/more files, and can also read text files containing paths to multiple image files. The dialogs for mask and atlas selection expect a single NIfTI file. Adding compressed images to the list may take a while as the integrity check of each image requires decompression of the file. Both single-file options are cleared (no mask/atlas file selected) with a right-button mouse click. Using an atlas produces a file `fastECM_tseries.mat` with regional time series before computation.

A left-button mouse click on the input file list ask for the name of a text file to which the list is then saved. A right click empties the input file list.

The left side of the Settings tab also shows the button to start the fECM computation at the top. Below are 3 tick boxes for (1) writing a 'node power' map, a continuous extension of degree centrality, (2) writing a -uniformly distributed- ECM ranks map and (3) writing a N(0,1) distributed map, computed from the ranks by inverse transform sampling.

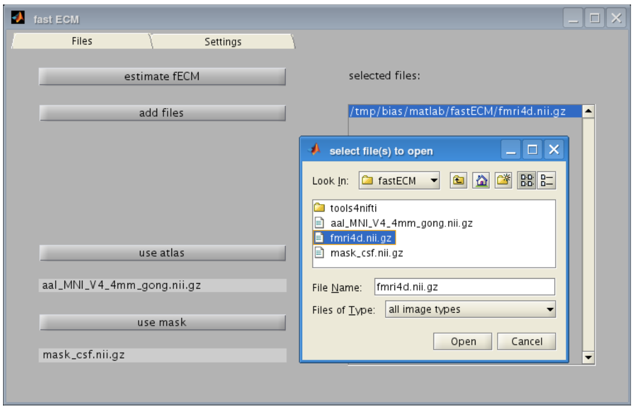
Below these tick boxes is a slider / text box combination to set the maximal number of iterations which is an integer between 1 and 100. The actual number of iterations is between 10 and 20 for a typical dataset; the main use of this setting is to end the program if the algorithm does not converge.

In the text box below the iterations text box the number of 'dynamics' can be set. Dynamic ECM are time series of maps computed in a sliding window.

If the tick box at the bottom is set, the connectivity matrix is explicitly computed, also for voxelwise ECM. This is a *computationally expensive* option that is not necessary when an atlas is used: in atlas mode the matrix is always computed.

When the full connectivity matrix is computed, .i.e., in atlas mode or when the 'use whole matrix' box is ticked, the connectivity matrix of each dynamic is written as a slice of an image. A 'backbone' is also computed from each matrix: a minimal spanning tree with the strongest connections added up to an average degree of $\sqrt{|\text{regions}|}$. From this backbone, node-wise betweenness, module membership, path length and clustering are computed using Brain Connectivity Toolbox functions. These computations are done in the 1st iteration of every dynamic. The results are written to a spreadsheet-readable XML file `fastECMstats.xml`, where each node-based measure on a separate sheet. Each sheet has one column per atlas region and one row per dynamic (*see Fig 1c*).

The right side shows the log of the output. A left-button mouse click on the output log asks for the name of a text file to which the log is then saved. A right click clears he log. The log is also cleared each time the fECM computation button is pressed.



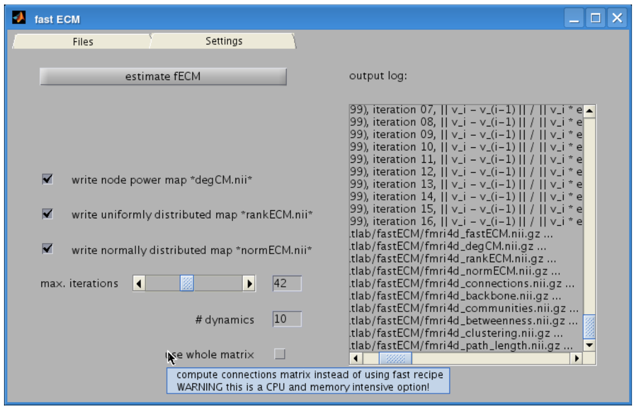
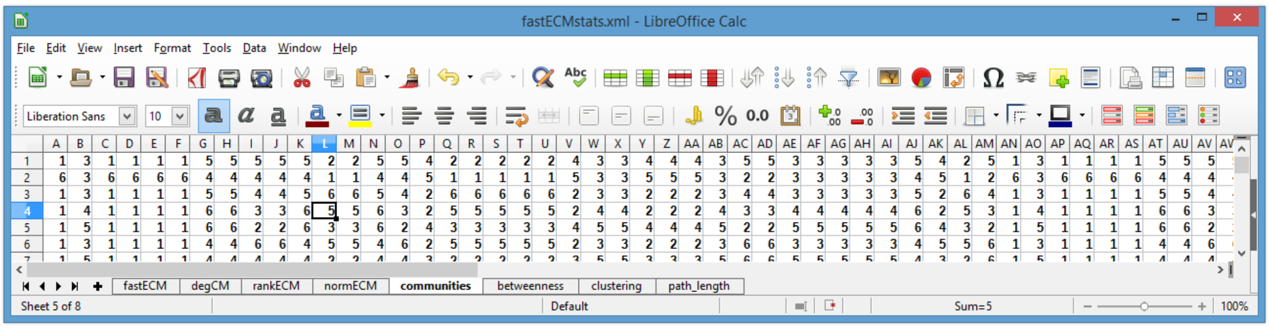


Fig 1a: the Files tab

Fig 1b: the Settings tab

Fig 1c: spreadsheet XML with atlas mode



References

Wink, A.M. et al. (2012) “Fast eigenvector centrality mapping of voxel-wise connectivity in functional magnetic resonance imaging: implementation, validation, and interpretation”, Brain connectivity 2(5): 265-74
http://dx.doi.org/10.1089/brain.2012.0087

Binnewijzend, M.A.A. et al. (2014) “Brain network alterations in Alzheimer's disease measured by Eigenvector centrality in fMRI are related to cognition and CSF biomarkers”, Human Brain Mapping 35(5): 2382-93
http://dx.doi.org/10.1002/hbm.22335

Schoonheim, M.S. et al. (2014) “Changes in functional network centrality underlie cognitive dysfunction and physical disability in multiple sclerosis”, Multiple Sclerosis Journal 20(8): 1058-65
http://dx.doi.org/10.1177/1352458513516892