

***SINOMO* Software Guide**

(February 9, 2011)

Abstract

This material is supplementary to the publication by Echtermeyer et al. [2011]. Building on that paper, this document provides information on usage of our software *SINOMO* (Singular Node Motifs).

Contents

1	Introduction	1
2	File Information	2
3	System Requirements	2
4	Using <i>SINOMO</i>	2
4.1	Command-line version	3
4.2	GUI-version	3
5	File Formats	3
5.1	Input	3
5.2	Output	3
6	Example Networks	5
7	Program Customisation	5
7.1	Manual Parameter Control	5
7.2	Improved Output Plots	5
8	High-Throughput Analyses	6
9	Final Remarks	6
A	Details on <code><input>_analysis.mat</code>	7

1 Introduction

Interactions and connections—be it in sociology or engineering—are often represented as networks, whose studies have improved understanding of underlying features and mechanisms. In many cases, irregularities in structure were identified as vulnerability or as crucial for best case performance. Advanced techniques to detect and specify unusual network-components are thus being developed. One way to characterise complex networks is by their specific connectivity patterns, called *network-motifs* [Milo et al., 2002], which can be identified using *mfinder*.¹ Here we use a different approach, which is to describe networks by *node-motifs*—a combination of local network features. Certain node-motifs, such as highly connected nodes or hubs, have been shown to be important components of networks (e.g. see [Albert et al., 2000, Jeong et al., 2001, Rodrigues

¹<http://www.weizmann.ac.il/mcb/UriAlon/>

and Costa, 2009]). Costa et al. [2009] have presented a technique to detect and specify more complex compound motifs, which are characterised by multiple features in combination. We described improvements to that method and showed how its parameters can be determined automatically [Echtermeyer et al., 2011]. This document describes our implementation *SINOMO* of the enhanced workflow, which can be controlled via a graphical user interface or through the command-line for batch processing.

2 File Information

The following files are supplied:

<code>readme.pdf</code>	this file
<code>sinomo.*</code>	main files of GUI-version
<code>workflow.m</code>	main file of command-line version
<code>!*</code>	directories containing sub-functions for workflow
<code>example_networks/*.csv</code>	example networks in csv-file format

3 System Requirements

Two versions of the code are supplied, which differ in their requirements: The first one requires Matlab (Mathworks Inc, Natick, USA) and allows the user to apply the workflow using a graphical user interface (GUI, Fig. 1). The other one is a command-line utility that either requires Matlab or the free alternative Octave [Eaton, 2002] and it can be easily used to batch process many networks without user interaction.² When using Octave, the freely available packages *econometrics* and *statistics* must be installed.³

Both the GUI- and command-line version make use of the `gs`-command (*Ghostsript*-package).⁴ If this package is not installed, error messages appear, but the analysis is performed correctly. However, the output-plots are split into multiple pdf-files rather than a single one.

Please note that neither version of the code is intended to be fool-proof and that absurd parameters are likely to yield absurd results. Only fundamental checks are performed; if desired, please implement sophisticated check-routines yourself.

4 Using *SINOMO*

The supplied code implements the improved *Beyond the Average*-workflow in two ways:

1. a script version (callable from the command-line) for both Matlab and Octave, and
2. an interactive GUI version (running on Matlab only).

²The code has been tested on Matlab version 7.9.0 [R2009b] and Octave version 3.2.3.

³<http://octave.sourceforge.net/>

⁴<http://www.ghostscript.com/>

The two variants differ with respect to their control, but analysis is performed using the same functions (contained in directories `!*`). Due to differences between Matlab and Octave, some functions contain conditional code that only executes on either of the programs. The correct branch is automatically chosen during execution.

4.1 Command-line version

On Linux, the script version is run via the command

```
matlab -nodisplay -nodesktop -nosplash \<\  
-r "workflow('$filename'); exit;"
```

if Matlab is installed; Octave can be evoked by

```
octave --eval "workflow('$filename'); exit;"
```

where the variable `$filename` has to be replaced by the filename of the csv-file to analyse. (Details on file formats are given in Section 5.) The script outputs are pdf- and mat-files, which are named similar to the input file.

4.2 GUI-version

To use the GUI version, start up Matlab and set the working directory to that containing the main-file `sinomo.m`. Calling the corresponding function `sinomo()` opens a file selection dialog, where the csv-file to analyse must be selected. (Clicking cancel at this point terminates Matlab.) Network statistics are calculated before the main screen with 5 plots appears (Fig. 1). Use the sliders on the top right of the window to change parameters of the "Beyond the Average"-workflow. (Alternatively, values can be entered directly into the text fields or the corresponding +/- buttons.) Plots are updated on any parameter change, if auto-plot updates are enabled (default), and can be saved to a pdf-file. Note that only one instance of the *SINOMO*-GUI runs at a time; to exit the program close its window.

5 File Formats

5.1 Input

The only input-file to the workflow is a csv-file that contains the adjacency matrix $A = (a_{ij})$ of the network to analyse. Elements in each of A 's rows are separated by commas; and each line of the csv-file corresponds to one of A 's rows. Internally, network-nodes are identified by unique numbers $1, 2, 3, \dots$, corresponding to their row-/column-index in A .

5.2 Output

For each input file `<input>.csv`, the workflow creates two output files named `<input>_analysis.mat` and `<input>.bw_%4.2f_w_%i_k_%i.pdf`. The mat-file stores all network-nodes' statistics, their mapping to the PCA-plane, estimated probabilities, the number of outliers w and motif-groups k , alongside with

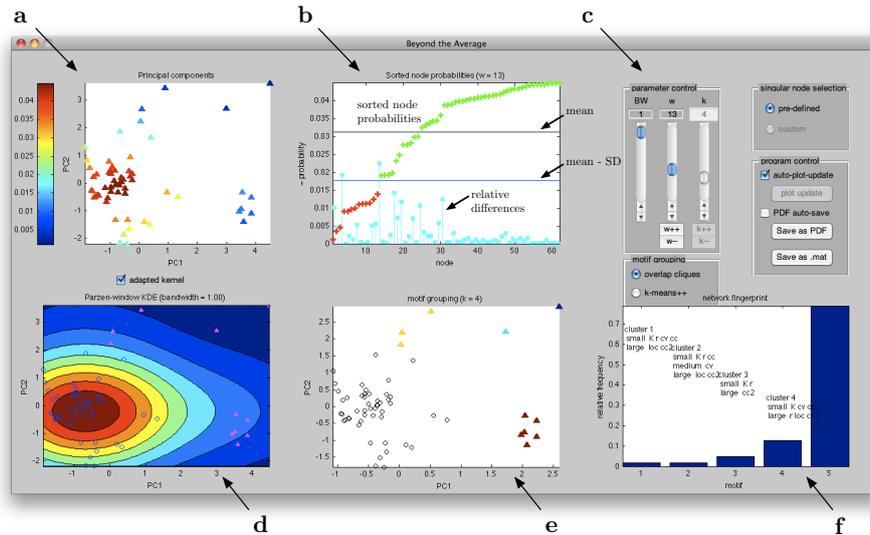


Figure 1: Graphical user interface for the BtA-workflow: **a** Nodes mapped to PCA-plane where their probability is coded by colour. The title of the plot informs about the percentage of variance in the 6-dimensional data is accounted for by the 2 principal components used. **b** Sorted node probabilities and relative differences. Red and green colour indicates singular and regular nodes, respectively. Mean probability indicated by black line; blue line marks mean minus one standard deviation. Stems (cyan) indicate relative differences between their two adjacent probabilities. **c** Manual workflow-parameter control and options for result export. Note, that the number of motif-groups k can only be altered if motif grouping is performed using k -means++. By default, changed settings show immediate effect in all plots (a,b,d-f). **d** Contour plot of PDF with reduced feature vectors superimposed, whose colour indicates whether they are classified regular or singular. The tick-box above the plot controls whether the Gaussian kernel is reshaped according to the standard deviation along each PC-axis (box ticked) or not. **e** PCA-plane (rescaled by standard deviations) showing differently coloured motif groups. **f** Bar plot showing the relative frequency for each motif-region. A brief characterisation of each motif is given above its bar. All plots and all data used for display can be stored to a file, by pressing the corresponding button (upper right).

cluster-assignments and other information, which may be useful for further processing. For details on stored variables please refer to Appendix A.

All plots that are generated by the GUI-version are stored as a pdf-file; likewise for the command-line version. The output file-name informs about the input-file and all relevant parameters to replicate contained results.

6 Example Networks

To verify that *SINOMO* works on your system, we supply example networks as csv-files, which can be found in the folder *example_networks*. In detail, the smallest network *ER_50.csv* is an Erdős-Rényi random network with 50 nodes [Erdős and Rényi, 1959]. Analysing the remaining networks *mac95.csv*, *celegans131.csv*, and *celegans277.csv* takes longer as these have 95, 131, and 277 nodes, respectively. These files represent neural connectivity of the Macaque cortex (one hemisphere) [Kötter, 2004, Kaiser and Hilgetag, 2006] and in *C. elegans*; consisting of 131 frontal neurons and all 277 neurons, respectively [Choe et al., 2004, Kaiser and Hilgetag, 2006]. When applying *SINOMO* to any of these networks, expect processing times of up to 30 seconds; no error messages should appear in the console.

7 Program Customisation

Depending on your needs and computing environment, you might want to choose to adapt certain parts of the program. The following paragraphs make suggestions about changes we found to be particularly useful.

7.1 Manual Parameter Control

By default, both the command-line and the GUI-version of the workflow choose parameters automatically according to the mechanisms we described [Echtermeyer et al., 2011]. Using the GUI, settings can be altered using the slider- and button-controls on the upper right. The command-line version also allows to choose some or all parameters manually by assigning values to the corresponding variables *bandwidths*, *ws*, and *ks* at the beginning of the file *workflow.m*. If multiple values are assigned (i.e. a vector) all of its values are used successively in any combination with the remaining parameters. The default setting of a parameter is chosen, if the parameter list is defined empty.

7.2 Improved Output Plots

By default, plots saved as a pdf-file appear side-centred with a significant margin, which can be reduced if the *pdfcrop* utility is installed.⁵ To enable its use, edit the file *save_plot.m* in the *!dataHandling* directory and comment out the corresponding line in the *save_and_crop*-function that evokes the command.

⁵<http://pdfcrop.sourceforge.net/>

8 High-Throughput Analyses

The command-line version of the supplied code is suitable for large scale data-analysis. It is mostly written such that Matlab/Octave makes use of small scale parallelisation on multi-core CPUs, which benefits run-time. Computer-clusters or similar architectures can give additional speed-up, which can be achieved in two ways:

1. When analysing many networks, total run-time is reduced by applying the workflow in parallel. This approach involves distributing data and programs, evoking calculations, and collecting results.
2. For every single network, the computational bottleneck of the workflow is the calculation of local measures for all network nodes. In order to reduce the run-time of this step, different measure can be evaluated on different compute nodes, which makes analyses of very large networks feasible.

For the first alternative, the distribution-, evocation-, and collection-step can be automated using a generic parallelisation-tool presented by Ribeiro et al. [2009]: *Adapa* (Automatic Data Parallelism). *Adapa* is available freely and can be used in combination with our tool.⁶ The second approach, however, requires appropriate modification of the code. We have corresponding implementations and facilities; please contact us if you are interested in collaborations.

9 Final Remarks

Although implemented with care, software is seldomly free of bugs. We perform systematic testing after any change of the code, but errors may still remain. If you experience any problems, please let us know. Also, if you use this software for your research, please cite the corresponding paper [Echtermeyer et al., 2011] in any work you publish.

⁶<http://www.dcc.fc.up.pt/adapa/>

A Details on `<input>_analysis.mat`

Following variables are stored in the file `<input>_analysis.mat`:

<code>no_of_nodes</code>	number of (non-isolated) network nodes
<code>w</code>	number of singular nodes
<code>k</code>	number of motif groups
<code>statistics</code>	values of local measures (column) for each node (row = feature-vector)
<code>stats_description</code>	descriptive text-label for <code>statistics</code> -columns
<code>PCA_projection</code>	reduced feature-vectors (according to PCA)
<code>probabilities</code>	estimated probabilities
<code>sorted_index</code>	ranking of nodes according to probability (node with lowest probability first)
<code>assignments</code>	motif-group where singular node belongs to
<code>noOfpointsInCluster</code>	number of members in each motif group

References

- R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–82, 2000.
- Y. Choe, B. H. McCormick, and W. Koh. Network connectivity analysis on the temporally augmented C. elegans web: A pilot study. In *Society of Neuroscience Abstracts*, page 30:921.9, Washington, DC, 2004. Society for Neuroscience.
- L. D. F. Costa, F. A. Rodrigues, C. C. Hilgetag, and M. Kaiser. Beyond the average: Detecting global singular nodes from local features in complex networks. *Europhysics Letters*, 87(July):18008, 2009.
- J. W. Eaton. *GNU Octave Manual*. Limited, Network Theory, 2002.
- C. Echtermeyer, L. da Fontoura Costa, F. A. Rodrigues, and M. Kaiser. Automatic Network Fingerprinting through Single-Node Motifs. *PLoS ONE*, 6: e15765, 2011.
- P. Erdős and A. Rényi. On Random Graphs I. *Publ. Math. (Debrecen)*, 6:290–7, 1959.
- H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41–2, 2001.
- M. Kaiser and C. C. Hilgetag. Nonoptimal Component Placement, but Short Processing Paths, due to Long-Distance Projections in Neural Systems. *PLoS computational biology*, 2(7):e95, 2006.
- R. Kötter. Online Retrieval, Processing, and Visualization of Primate Connectivity Data From the CoCoMac Database. *Neuroinformatics*, 2:127–44, 2004.
- R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298 (5594):824–7, 2002.
- P. Ribeiro, J. Simonotto, M. Kaiser, and F. Silva. Parallel calculation of multi-electrode array correlation networks. *Journal of Neuroscience Methods*, 184: 357–64, 2009.
- F. A. Rodrigues and L. D. F. Costa. Protein lethality investigated in terms of long range dynamical interactions. *Molecular BioSystems*, 5(4):385–90, 2009.