# Backbone detection algorithms

For further inquiries, please contact matthieu.nadini@gmail.com

This repository aims to convey several backbone detection algorithms in an unique script, thereby allowing users to easily analyze the same dataset in multiple ways, according with their preferences/needs. At the moment, we list three different approaches. The repository covers undirected networks only.

- The evolving activity-driven model (EADM) is a method that detect the backbone network against time-varying individual properties. The EADM proposes an approach based on the configuration model which offers reliable estimates when the size of the network is hundreds of nodes and beyond. We realize two different versions of the same method: the 'EADM_I' uses a supervised method, the V-optimal histograms [1], implemented as a part of the Visor tool [2], to determine the optimal interval partition, while the 'EADM_BB' uses an unsupervised method, the Bayesian Block representation [3], to determine the interval partition. For small networks, it is possible to test the accuracy of these methods by considering 'EADM_I_Accuracy', and 'EADM_BB_Accuracy', respectively. and the 'EADM_Accuracy'. The paper will appear soon in Nonlinear Dynamics.

- The temporal fitness model (TFM) is the most accurate method to reconstruct the backbone network of small networks, but it becomes demanding as the size of the network increases. We suggest to use it for networks having at most thousands nodes. More details in [4].

- The statistically validated network (SVN) was the first method able to filter the backbone network considering temporal patterns. It is particular useful when individual properties are constant in time and for large networks. More details in [5].

In order to run the code, we use a freely available dataset from the Sociopattern project [6] as an example. These simple steps have to be followed.

1. Download the files and place yourself in the directory with the python file Filtering_methods_u.py

2. Type python (the version 2.7 is required)

3. Import all the functions in the script (you may have to manually install some packages), type: from Filtering_methods_u import *

4. Run the code with default options, type: Backbone_detection()

Aside from this example, we explain in the following how to run the code with the option of your choices. We list the input parameters, their purposes and how to modify them:

- **path_in**: path to the directory containing the dataset to be analyzed. Default: path_in = ‘DATASET/Example_PrimarySchool/’.
- **name_file**: name of the file in which the edgelist and its timestamp are listed. Default: name_file = ‘primaryschool.csv’.
- **column_time**: it indicates the column containing the timestamp. Only the first three columns in the file will be considered. Default: column_time=0.
- **sep**: how the columns in the edgelist file are separated. Default: sep=‘\t’.
- **dt**: time step which determines the evolution of the temporal network. It must be a list. Default: dt=[20., 60.*1., 60.*5., 60.*15.]. These resolutions correspond to 20 seconds, 1 minute, 5 minutes, and 15 minutes, respectively.
- **multiedges**: if you want to retain multiple connections happening within the same time step, write ‘yes’, otherwise ‘no’ to allow only one link per time step. It must be a list and its length should be equal to dt. The TFM does not work if multiedges are present. Default: multiedges = [‘no’].
- **directory_out**: it determines where all output file will be stored. Default: directory_out = ‘Output’.
- **remove_nights**: if you want to remove time steps in which no connections happen, write ’yes’, otherwise ’no’ to consider the whole observation window. Default: remove_nights=‘yes’.
- **alpha**: value of the significance threshold. All links having a p-value lower than the significance threshold will be included in the backbone network. It must be a list. Default: alpha = [0.01].
- **Bonferroni_corr**: write ‘yes’ if you want to correct alpha according to the Bonferroni correction, otherwise write ’no’ and the significance threshold is set to be equal to alpha. Default: Bonferroni_corr = ‘yes’.
- **model**: choose what filtering approach you want to apply to the temporal dataset. Options are: ‘EADM_BB’, ‘EADM_I’, ‘EADM_BB_Accuracy’, ‘EADM_I_Accuracy’, ‘TFM’, and ’SVN’. Default: model = ‘EADM’. Default: model = ‘EADM_BB’.

For example, if we would like to analyze the network using the TFM model, we have to type Backbone_detection(model = ‘TFM’).

The output files are listed in the following:

- **links_*dt[i]_model*.txt**: it stores the time step chosen (first column) and the number of significant links detected (second column). *dt[i]* is a string equal to i element of the list dt, which is the input parameter determining the length of the time step. *model* is a string equal to the respective input parameter.

- **edgelist_*dt[i]_alpha[j]_model*.txt**: it returns the edgelist representing all significant links in the network. *dt[i]* is a string equal to i element of the list dt, which is the input parameter determining the length of the time step. *alpha[j]* is a string equal to j element of the list alpha, which is the input parameter determining the threshold for selecting significance links. *model* is a string equal to the respective input parameter.
- **relative_error_*dt[i]_model*.txt**: it computes the accuracy of the method in describing the overall network evolution through the relative error. *dt[i]* is a string equal to i element of the list dt, which is the input parameter determining the length of the time step. *model* is a string equal to the respective input parameter. The accuracy of the approach can be computed only for 'EADM_BB_Accuracy', 'EADM_I_Accuracy', and 'TFM'.
- If the models 'EADM_I' or 'EADM_I_Accuracy' are used, then a file a.txt is generated. Please do not consider it since it is useful only as a supplementary file. You may delete it.

The directory **oksPublic** does not have to be deleted. It is useful for the 'EADM_I' or 'EADM_I_Accuracy' methodologies.

## References

[1] Hosagrahar Visvesvaraya Jagadish, Nick Koudas, S Muthukrishnan, Viswanath Poosala, Kenneth C Sevcik, and Torsten Suel. Optimal histograms with quality guarantees. In *VLDB*, volume 98, pages 24–27, 1998.

[2] Giovanni Mahlknecht, Michael H Bohlen, Anton Dignös, and Johann Gamper. Visor: Visualizing summaries of ordered data. *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, page 40, 2017.

[3] Jeffrey D Scargle, Jay P Norris, Brad Jackson, and James Chiang. Studies in astronomical time series analysis. vi. bayesian block representations. *The Astrophysical Journal*, 764(2):167, 2013.

[4] Teruyoshi Kobayashi, Taro Takaguchi, and Alain Barrat. The structured backbone of temporal social ties. *Nature communications*, 10(1):220, 2019.

[5] Ming-Xia Li, Vasyl Palchykov, Kaski Kimmo Jiang, Zhi-Qiang, János Kertész, Salvatore Miccichè, Michele Tumminello, Wei-Xing Zhou, and Rosario N. Mantegna. Statistically validated mobile communication networks: the evolution of motifs in European and Chinese data. *New Journal of Physics*, 16(8):083038, 2014.

[6] `www.sociopatterns.org`.