Source code documentation for:

[1] Marc G. Leguia, Cristina G. B. Martínez, Irene Malvestio, Adrià Tauste Campo, Rodrigo Rocamora, Zoran Levnajić, and Ralph G. Andrzejak: "Inferring directed networks using a rank-based connectivity measure" Phys. Rev. E 99, 012319.

Relevant literature:

[2] Daniel Chicharro and Ralph G. Andrzejak. "Reliable detection of directional couplings using rank statistics". Phys. Rev. E 80, 026217.

[3] Petroula Laiou and Ralph G. Andrzejak. "Coupling strength versus coupling impact in nonidentical bidirectionally coupled dynamics" Phys. Rev. E 95, 012210.

[4] Lorenz, Edward Norton*. "Deterministic nonperiodic flow". Journal of the Atmospheric Sciences.* ***20*** *(2): 130–141*

The source code allows you to calculate the measure L first introduced in [2] with the improvements introduced in [1]. Please cite Ref. [1] if you use this code.

In order to understand the following documentation, you should at first read Ref. [1]. The codes use the same notation for mathematical symbols as in this paper. Additional comments can be found throughout the source code. In case you have any questions, please contact the authors at mgrauleg@gmail.com. We on purpose we do not include the ones analyzing the EEG signals since we do not have the permission of sharing that data.

To get started please copy all the source codes in the same directory. To run the main program call *GrauLeguiaPRE2019example(N,resets,seed,epsilon,rho,noise,homo)*. If the function has no input parameters, the default parameters are N=16 nodes of a homogeneous Lorenz with b=28, 200 dynamical resets, link density \rho=0.1, coupling strength \variepsilon=0.1, noise = 0.

How you call it:

N: number of nodes

resets: number of dynamical resets from which we average.

seed: Seed for the generation of Adjacency matrix

epsilon: coupling strength

rho: link density of the Adjacency matrix

Homo: if 1 homogeneous Lorenz with b=28, if 0 heterogeneous Lorenz with b\in[28,48]

To compute the matrix of the pairwise L between all signals we use LMultiBig.m. The function:

L = LMultiBig (Datarec,m,tau,krec,W)

Datarec: Time series form which you want to compute the pairwise L. Columns should contain the time series.

m: embedding dimension used for the reconstruction of the signal.

tau: time delay used for the reconstruction of the signal.

krec: nearest neighbours from which L is computed (can be a vector)

W: Theiler correction

The default parameters we use are: m=5, tau=5,krec=5,W=15. These values are taken from [3] to avoid any in-sample parameter optimization.

To compute solve the coupled network of Lorenz attractors we use the Runge-kutta of 4th order function

[x, times] = IntegratorNetnoise(stepkind,deriv,N,dt,inicond, a1,eps,noise)

stepkind: If==1 euler method, if==0 Runge-kutta of 4th order

deriv: @deriv the ODE of coupled Lorenz attractors

N: number of nodes

dt: time step

inicond: initial conditions

a1: rho value of the Lorenz attractor.

eps: Matrix of the coupling strength

noise: amplitude of the Gaussian noise

OUTPUT

x: 2-D vector (Number of points, 3*N) of the time series for each component and node. Each column represent the time series of the x,y,z component in order for each of the N nodes of the system.

times: time of the derivation in a.u.

Finally, the network of the Lorenz oscillators is:

dx = LorenzLorenzODENetdifR(x,a1,eps)

x: the initial conditions for the whole system in an array.

a1: array of rho values for each Lorenz in the system. The rest of coefficents of the lorenz attractor are set to \sigma=10 and \beta=8/3

eps: Matrix of couplings strengths.

OUTPUT

dx: an array of all components of the Lorenz attractor for all the nodes of the system