ITS Toolbox:
A Matlab toolbox for the practical computation of Information Dynamics

Version 2.1 – January 2019

Luca Faes

Dept. of Engineering, University of Palermo, Italy

www.lucafaes.net
Luca.faes@unipa.it

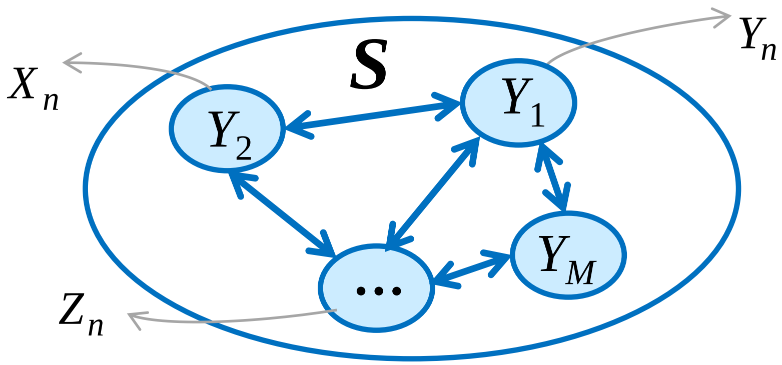
INTRODUCTION

- Network of dynamic processes

Observed dynamical system $\boldsymbol{S}$

M dynamic processes $Y_1, Y_2, \ldots, Y_M$

Measured time series: $Y_{1,n}, Y_{2,n}, \ldots, Y_{M,n}$



- Realization: $N \times M$ data matrix $\boldsymbol{Y} = \begin{bmatrix} Y_{1,1} & \cdots & Y_{M,1} \\ \vdots & \ddots & \vdots \\ Y_{1,N} & \cdots & Y_{M,N} \end{bmatrix}$

- Estimation of Information Dynamics:

(A) Definition of Measures: univariate, bivariate, multivariate

Can be expressed in terms of **conditional entropies**

(B) Approximation of the past history of the processes

Embedding procedures: **uniform, non-uniform**

(C) Computation of conditional entropy

Entropy estimators: **Linear (Gaussian), Binning, Kernel, Nearest Neighbors**

UNIVARIATE SYSTEM ANALYSIS

TARGET SYSTEM: Y_j → Time series: Y_n

j-th column of the data matrix

$$\begin{array}{cc} & \begin{array}{ccc} & j & \end{array} \\ \boldsymbol{Y} = & \begin{bmatrix} \cdots & \boldsymbol{Y}_1 & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \boldsymbol{Y}_N & \cdots \end{bmatrix} \end{array}$$

- Information generated by Y_j: ***Entropy***

$$H_Y = H(Y_n) = -\sum p(y_n)\log p(y_n)$$

*functions for **Entropy**:*

```
its_Elin.m
its_Ebin.m
its_Eker.m
its_Eknn.m
```

- Information Storage in Y_j: ***Mutual Information***

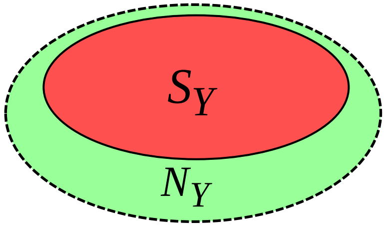
$$S_Y = I(Y_n;Y_n^-) = H(Y_n) - H(Y_n \mid Y_n^-)$$

✓ New Information: ***Conditional Entropy***

$$N_Y = H(Y_n \mid Y_n^-)$$

*functions for **Self Entropy**:*

```
its_SElin.m
its_SEbin.m
its_SEker.m
its_SEknn.m
```



$$H_Y = S_Y + N_Y$$

BIVARIATE SYSTEM ANALYSIS

TARGET SYSTEM: Y_j → Time series: Y_n

j-th column of the data matrix

DRIVER SYSTEM: Y_i → Time series: X_n

i-th column of the data matrix

$$\begin{array}{cc} & \begin{array}{ccccc} & i & & j & \\ & \downarrow & & \downarrow & \end{array} \\ Y = & \begin{bmatrix} \cdots & X_1 & \cdots & Y_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & X_N & \cdots & Y_N & \cdots \end{bmatrix} \end{array}$$

- Information Transfer from X to Y :

  Conditional Mutual Information

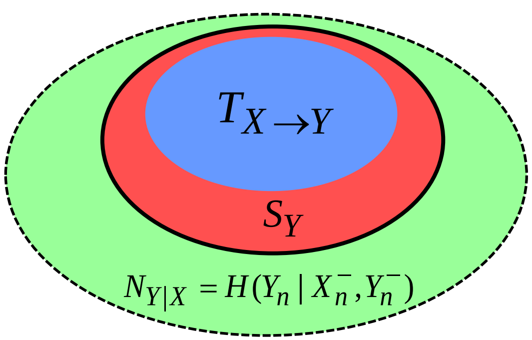
$$T_{X \to Y} = I(Y_n; X_n^- \mid Y_n^-) = H(Y_n \mid Y_n^-) - H(Y_n \mid X_n^-, Y_n^-)$$

*functions for **Transfer Entropy**:*

its_BTElin.m
its_BTEbin.m
its_BTEker.m
its_BTEknn.m

✓ New Information: ***Conditional Entropy***

$$N_{Y|X} = H(Y_n \mid X_n^-, Y_n^-)$$



$$H_Y = S_Y + T_{X \to Y} + N_{Y|X}$$

MULTIVARIATE SYSTEM ANALYSIS

TARGET SYSTEM: Y_j → Time series: Y_n
j-th column of the data matrix

DRIVER SYSTEM: Y_i → Time series: X_n
i*-th* column of the data matrix

OTHER SYSTEMS: $Y\setminus\{Y_i,Y_j\}$ → Time series: $\boldsymbol{Z}_n$
k*-th* columns of the data matrix

$$Y = \begin{array}{c} \begin{array}{ccccc} & \boldsymbol{i} & & j & \\ & \downarrow & & \downarrow & \end{array} \\ \begin{bmatrix} \cdots & X_1 & \cdots & Y_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & X_N & \cdots & Y_N & \cdots \end{bmatrix} \\ \boldsymbol{k} \end{array}$$

- Joint Information Transfer from X,Z to Y :

$$T_{XZ\rightarrow Y} = I(Y_n; X_n^-, Z_n^- \mid Y_n^-) = H(Y_n \mid Y_n^-) - H(Y_n \mid X_n^-, Y_n^-, Z_n^-)$$

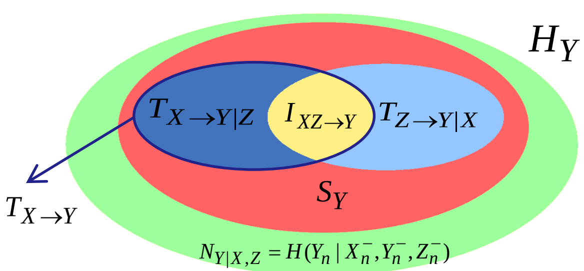
Joint Transfer Entropy:
its_BTElin.m
its_BTEbin.m
its_BTEker.m
its_BTEknn.m

- Conditional Information Transfer from X to Y given Z:

$$T_{X\rightarrow Y|Z} = I(Y_n; X_n^- \mid Y_n^-, Z_n^-) = H(Y_n \mid Y_n^-, Z_n^-) - H(Y_n \mid X_n^-, Y_n^-, Z_n^-)$$

✓ New Information: $N_{Y|X} = H(Y_n \mid X_n^-, Y_n^-, Z_n^-)$

Partial Transfer Entropy:
its_PTElin.m
its_PTEbin.m
its_PTEker.m
its_PTEknn.m



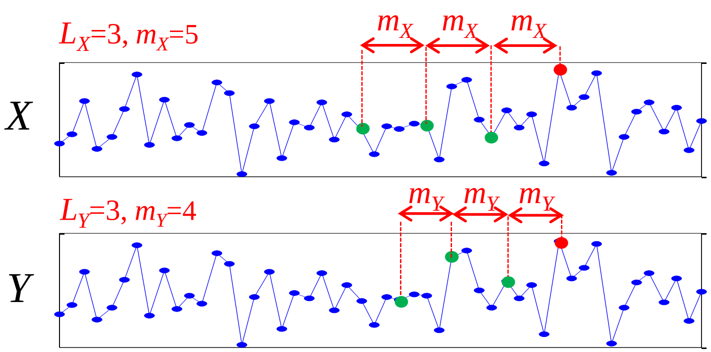
$$H_Y = S_Y + T_{XZ\rightarrow Y} + N_{Y|X,Z}$$

$$T_{XZ\rightarrow Y} = T_{X\rightarrow Y|Z} + T_{Z\rightarrow Y|X} + I_{XZ\rightarrow Y}$$
$$= T_{X\rightarrow Y} + T_{Z\rightarrow Y} - I_{XZ\rightarrow Y}$$

$$I_{XZ\rightarrow Y} = T_{X\rightarrow Y} - T_{X\rightarrow Y|Z}$$

- Uniform embedding (UE):

  Covers the past of each system with predetermined lagged components, uniformly spaced in time



$$X_n^- \approx [X_{n-m_X} \dots X_{n-L_X m_X}]$$

$$Y_n^- \approx [Y_{n-m_Y} \dots Y_{n-L_Y m_Y}]$$

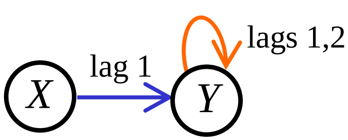
L = embedding dimension
m = embedding lag

Example: L=3, m=1 ➜ $V_n = [X_{n-1}, X_{n-2}, X_{n-3}, Y_{n-1}, Y_{n-2}, Y_{n-3}]$

- Non-Uniform embedding (NUE):

  Approximates the system past through a sequential procedure that selects progressively the lagged components according to a criterion for maximum relevance and minimum redundancy

Example:



➜ $V_n = [\underbrace{X_{n-1}}_{X_n^-}, \underbrace{Y_{n-1}, Y_{n-2}}_{Y_n^-}]$

ESTIMATION: COMPUTATION OF CONDITIONAL ENTROPY

Functions for embedding (common to all estimators):

its_SetLag.m	Sets the indices for embedding
its_buildvectors.m	Given the embedding indices, forms the observation matrix B

Example: M=2 time series

Uniform embedding with L=2, m=1 → $V_n=[X_{n-1},X_{n-2},Y_{n-1},Y_{n-2}]$

Data matrix → Embedding indices → Observation matrix

$$\begin{bmatrix} X_1 & Y_1 \\ \vdots & \vdots \\ X_N & Y_N \end{bmatrix} \rightarrow Vi = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \end{bmatrix} \rightarrow B = \begin{bmatrix} Y_3 & X_2 & X_1 & Y_2 & Y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_n & X_{n-1} & X_{n-2} & Y_{n-1} & Y_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_N & X_{N-1} & X_{N-2} & Y_{N-1} & Y_{N-2} \end{bmatrix}$$

Functions for computing conditional entropy (estimator-specific):

• ***Linear***	**its_CElin.m**	Conditional Entropy from the Observation Matrix	(Uniform Embedding)
	its_CElinVAR.m	Conditional Entropy from the VAR parameters	
• ***Binning***	**its_CEbin.m**	Conditional Entropy from the Observation Matrix	
	its_NUEbin.m	Conditional Entropy from the Observation Matrix, Non-Uniform Embedding	
• ***Kernel***	**its_CEker.m**	Conditional Entropy from the Observation Matrix	
	its_NUEker.m	Conditional Entropy from the Observation Matrix, Non-Uniform Embedding	
• ***Nearest neighbor***	**its_CEknn.m**	Conditional Entropy from the Observation Matrix	
	its_CMIknn.m	Conditional Mutual Information from the Observation Matrix	
	its_NUEknn.m	Conditional Mutual Information from the Observation Matrix, Non-Uniform Embedding	

ESTIMATORS: LINEAR-MODEL BASED ESTIMATOR

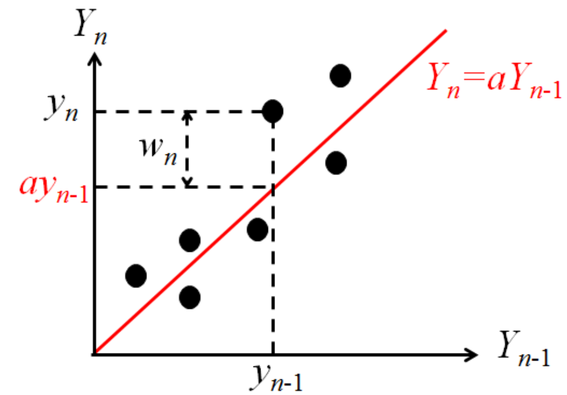
- **Computation based on linear prediction models**

Exploits the analytic relation between (conditional) entropy and (prediction error) variance valid for Gaussian processes, and performs linear regression to find the prediction error variance

$$Y_n = a_1 Y_{n-1} + \cdots a_L Y_{n-L} + W_n$$

$$S_Y = \frac{1}{2}\ln\frac{\sigma(Y_n)}{\sigma(Y_n \mid Y_n^L)} = \frac{1}{2}\ln\frac{\sigma_Y^2}{\sigma_W^2}$$

- Example: L=1 $Y_n^L \cong Y_n^1 = Y_{n-1}$



- ***The estimator uses Uniform Embedding*** $\rightarrow$ $X_n^- \cong X_n^L = [X_{n-1} \ldots X_{n-L}],\; Y_n^- \cong Y_n^L = [Y_{n-1} \ldots Y_{n-L}]$

- ***Analysis parameters:*** Regression order: L (either imposed or selected with optimization criteria)

Main functions:

- **its_Elin.m** System Information, univariate system
- **its_SElin.m** Information Storage, univariate system
- **its_BTElin.m** Information Transfer, bivariate system
- **its_PTElin.m** Conditional Information Transfer, multivariate system

its_SetLag.m
its_buildvectors.m
its_CElin.m
its_CElinVAR.m

Other functions:

- **its_LinReg_Ftest.m** Statistical significance of Information Dynamics, based on Fisher F-test
- **its_FindOrderLin.m** Selection of regression order, based on Akaike or Bayesian information criteria

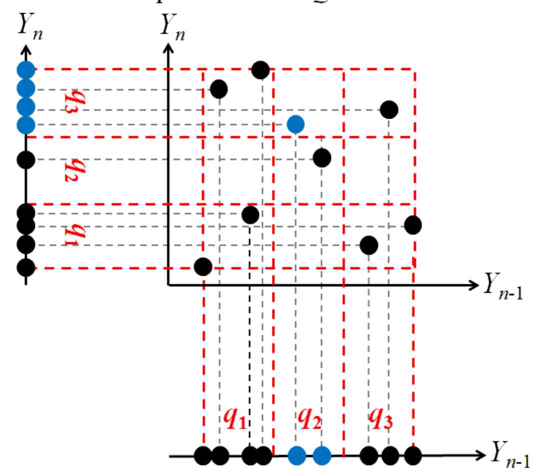
ESTIMATORS: MODEL-FREE ESTIMATOR BASED ON BINNING

- **Computation based on time series quantization**

  Discretize the values of each variable using quantization levels, then estimate the probability as the relative frequency of visitation of the hypercubes in the multidimensional space spanned by the variables

- Example: L=1 $Y_n^L \cong Y_n^1 = Y_{n-1}$

Uniform partition with Q=5 bins



- ***Non-Uniform Embedding is recommended to limit dimensionality***

- ***Analysis parameters:*** Number of quantization levels: c
  Embedding parameters: L, m (for non-uniform embedding, also parameters for procedure termination)

Main functions:

- **its_Ebin.m** System Information, univariate system
- **its_SEbin.m** Information Storage, univariate system
- **its_BTEbin.m** Information Transfer, bivariate system
- **its_PTEbin.m** Conditional Information Transfer, multivariate system

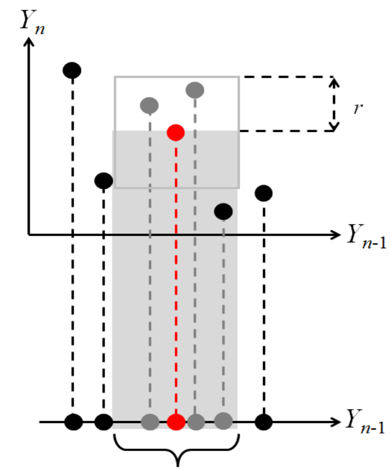
its_SetLag.m
its_buildvectors.m
its_CEbin.m
its_NUEbin.m

Other functions:

- **its_quantization.m** Uniform quantization of the time series

ESTIMATORS: MODEL-FREE ESTIMATOR BASED ON KERNELS

- Example: L=1 $Y_n^L \cong Y_n^1 = Y_{n-1}$



Range search with fixed threshold, r

- **Computation based on kernel density estimation**

  Approximate the probability density at each data point by using kernel functions to weight the distance from the reference point to any other point in the time series;

  If the Heaviside kernel with parameter r is used, the method counts the relative number of points having distance less than r from the reference point, then averages across all points

- ***Non-Uniform Embedding is recommended to limit dimensionality***

- ***Analysis parameters:*** Threshold distance: r (usually a fraction of the SD of the time series)
  Embedding parameters: L, m (for non-uniform embedding, also parameters for procedure termination)

Main functions:

- **its_Eker.m** System Information, univariate system
- **its_SEker.m** Information Storage, univariate system
- **its_BTker.m** Information Transfer, bivariate system
- **its_PTEker.m** Conditional Information Transfer, multivariate system

its_SetLag.m
its_buildvectors.m
its_CEker.m
its_NUEker.m

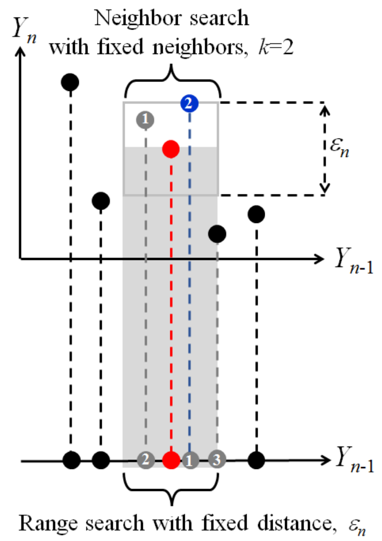
ESTIMATORS: MODEL-FREE ESTIMATOR BASED ON NEAREST NEIGHBORS

- **Computation based on nearest neighbor statistics**

  Study the probability distribution for the distance between an observation and its k-th neighbor

- ***Compensation of the estimation bias*** *in a sum of entropies by looking for neighbors in the higher dimensional spaces, and counting within ranges in the lower dimensional spaces*

- Example: L=1 $Y_n^L \cong Y_n^1 = Y_{n-1}$



- ***Non-Uniform Embedding is recommended to limit dimensionality***

- ***Analysis parameters:*** Number of neighbors: k
  Embedding parameters: L, m (for non-uniform embedding, also parameters for procedure termination)

Main functions:

- **its_Eknn.m** System Information, univariate system
- **its_SEknn.m** Information Storage, univariate system
- **its_BTEknn.m** Information Transfer, bivariate system
- **its_PTEknn.m** Conditional Information Transfer, multivariate system

its_SetLag.m
its_buildvectors.m
its_CMIknn.m
its_NUEknn.m

Other functions:

- **nn_prepare, nn_search, range_search** functions for neighbor search and range search

  Built-in functions, compiled from C++ and imported from the TSTOOL Toolbox [*http://www.physik3.gwdg.de/tstool/*]

COMPUTATION OF INFORMATION DYNAMICS

General procedure for the computation of Information Dynamics

Load data matrix

(normalize time series)

$\mathbf{Y}$

Set indices of time series

Univariate: index of Y
Bivariate: indices of X,Y
Multivariate: indices of X,Y,Z

Y

Compute Entropy Hy
Elin,Ebin,Eker,Eknn

X,Y,Z

Compute Information Dynamics:

a) Set embedding Indices

Uniform Embedding:
SetLag

Non-Uniform Embedding:
SetLag + NUEbin
SetLag + NUEker
SetLag + NUEknn

V_L

b) Compute the desired measure

- Univariate analysis: Information storage S_Y
  SElin, SEbin, Seker, SEknn
- Bivariate analysis: Information Transfer $T_{X\to Y}$
  BTElin, BTEbin, BTEker, BTEknn
- Multivariate analysis: Conditional Information Transfer $T_{X\to Y|Z}$
  PTElin, PTEbin, PTEker, PTEknn

REFERENCES

- **Theory and generalities about estimation**
  - L Faes, A Porta, 'Conditional entropy-based evaluation of information dynamics in physiological systems', in *Directed Information Measures in Neuroscience*, R Vicente, M Wibral, J Lizier (eds), Springer-Verlag; **2014**, pp. 61-86
- **Theory and linear-model based estimation**
  - L Faes, A Porta, G Nollo, 'Information decomposition in bivariate systems: theory and application to cardiorespiratory dynamics', *Entropy*, special issue on "Entropy and Cardiac Physics", **2015**, 17:277-303.
  - L Faes, A Porta, G Nollo, M Javorka, 'Information decomposition in multivariate systems: definitions, implementation and application to cardiovascular networks', Entropy, special issue on Multivariate entropy measures and their applications, **2017**, 19(1), 5.
- **Comparison of Entropy measures and estimators**
  - W Xiong, L Faes, P Ch Ivanov, 'Entropy measures, entropy estimators and their performance in quantifying complex dynamics: effects of artifacts, nonstationarity and long-range correlations', Phys. Rev. E, **2017**; 95:062114 (37 pages).
- **Nearest neighbor estimation and non-uniform embedding**
  - L Faes, D Kugiumtzis, A Montalto, G Nollo, D Marinazzo, 'Estimating the decomposition of predictive information in multivariate systems', *Phys. Rev. E* **2015**; 91:032904 (16 pages)
- **Binning estimation and non-uniform embedding**
  - L Faes, D Marinazzo, A Montalto, G Nollo, 'Lag-specific transfer entropy as a tool to assess cardiovascular and cardiorespiratory information transfer', IEEE Trans Biomed Eng **2014**; 61(10):2556-2568.
  - L Faes, G Nollo, A Porta: 'Non-uniform multivariate embedding to assess the information transfer in cardiovascular and cardiorespiratory variability series', Comput Biol Med **2012**; 42:290-297.
  - L Faes, G Nollo, A Porta: 'Information-based detection of nonlinear Granger causality in multivariate processes via a nonuniform embedding technique', Phys Rev E; **2011**; 83(5 Pt 1):051112.
- **Implementation for Transfer Entropy**
  - A Montalto, L Faes, D. Marinazzo, 'MuTE: a MATLAB toolbox to compare established and novel estimators of the multivariate transfer entropy', PLOS ONE **2014**; 9(10):e109462 (13 pages).
- **Applications**
  - L Faes, D Marinazzo, F Jurysta, G Nollo, 'Linear and nonlinear analysis of brain-heart and brain-brain interactions during sleep', Phys. Meas. **2015**; 36:683-698.
  - L Faes, G Nollo, F Jurysta, D Marinazzo, 'Information dynamics of brain-heart physiological networks during sleep', New J Phys **2014**; 16:105005 (20 pages).
  - L Faes, A Porta, G Rossato, A Adami, D Tonon, A Corica, G Nollo: 'Investigating the mechanisms of cardiovascular and cerebrovascular regulation in orthostatic syncope through an information decomposition strategy', Autonomic Neurosci **2013**; 178:76-82.
  - L Faes, G Nollo, A Porta: 'Mechanisms of causal interaction between short-term heart period and arterial pressure oscillations during orthostatic challenge', J Appl Physiol **2013**;114:1657-1667.

EXAMPLES: SIMULATIONS 1-2

- **Simulated VAR process: cardiorespiratory dynamics**

$$X_n = a_1 X_{n-1} + a_2 X_{n-2} + U_n$$

$$Y_n = b_1 Y_{n-1} + b_2 Y_{n-2} + C\, X_{n-2} + V_n$$

→ a_1, a_2 Respiratory oscillation (HF)

→ b_1, b_2 Slow cardiac oscillation (LF)

C cardiorespiratory coupling

Transfer function poles:

ρ_x=0.9, f_x=0.3 Hz

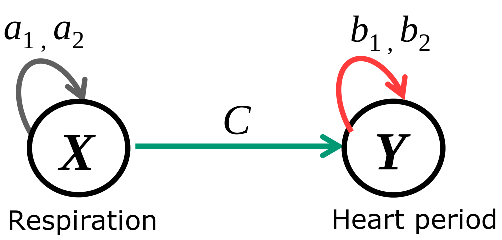
ρ_y=0.8, f_y=0.1 Hz

→ *Parameters:*

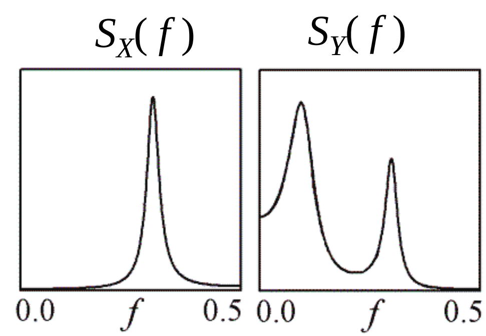
a_1=-0.556, a_2=-0.81

b_1=1.294, b_2=-0.64,

❖ **Process Graph**



❖ **Spectra** (C=1)



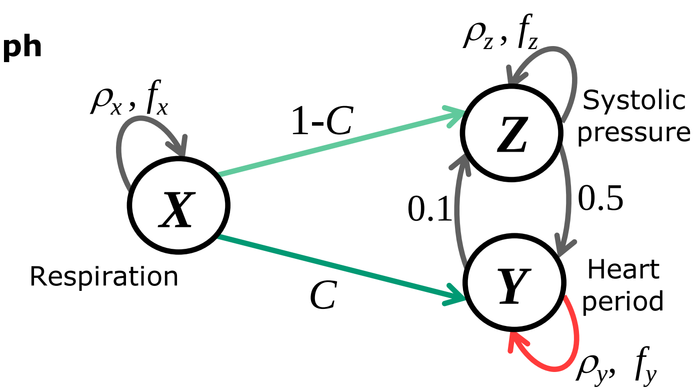
- Simulation scripts:
  - **Example_simu1_AR1.m** *Information Dynamics for X only*
  - **Example_simu2_AR2.m** *Information Dynamics for both X and Y*

  (var_simulations.m, var_filter.m, var_spectra.m)

EXAMPLES: SIMULATION 3

- **Simulated VAR process: cardiovascular and cardiorespiratory dynamics**
  - *Respiratory flow* $\rightarrow$ $X_n = 2r_x \cos(2\pi f_x) \cdot X_{n-1} - r_x^2 \cdot X_{n-2} + U_n$
  - *Heart period:* $\rightarrow$ $Y_n = 2r_y \cos(2\pi f_y) \cdot Y_{n-1} - r_y^2 \cdot Y_{n-2} + 0.5 \cdot Z_{n-1} + C \cdot X_{n-1} + V_n$
  - *Systolic pressure:* $\rightarrow$ $Z_n = 2r_z \cos(2\pi f_z) \cdot Z_{n-1} - r_z^2 \cdot Z_{n-2} + (1-C) \cdot X_{n-2} + 0.1 \cdot Y_{n-2} + W_n$

❖ **Process Graph**



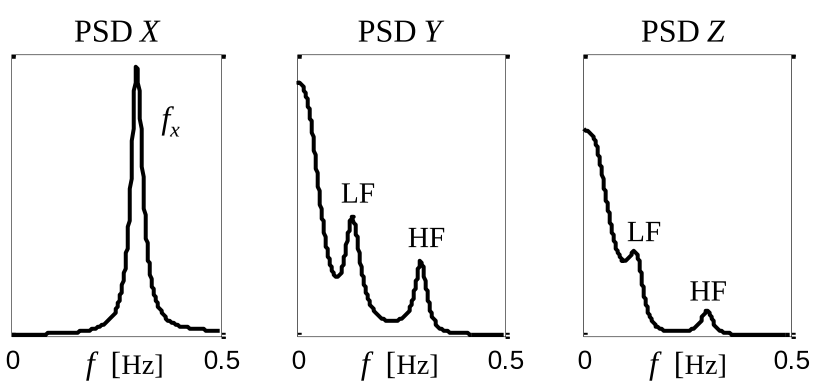
❖ **Spectra** *Parameters:*

ρ_x=0.9, f_x=0.3 Hz (HF)

ρ_y=0.8, f_y=0.1 Hz (LF)

ρ_z=0.8, f_z=0.1 Hz (LF)

C=0.5



- Simulation script:
  - **Example_simu3_AR3.m (var_simulations.m, var_filter.m, var_spectra.m)**

EXAMPLES: SIMULATION 4

- **Order-two autoregressive process (AR2):**

$$X_n = 2\rho\cos(2\pi f)X_{n-1} - \rho^2 X_{n-2} + U_n$$

The input white noise is colored with a 1-pole filter, where ρ *and* f *are the pole modulus and frequency*

This simulation compares the performance of linear, binning, kernel and nearest neighbor estimators in quantifying Entropy (information), Conditional Entropy (new information) and Self Entropy (information storage) for a simple linear autoregressive process

After choosing an estimator, results are displayed showing the theoretical values of the measures, and the distribution of the estimated values over several replications of the process, as a function of the parameter ρ *determining the dynamical complexity of the process.*

- Simulation script:
  - **Example_simu4_cmpEst.m**

    (var_filter.m)

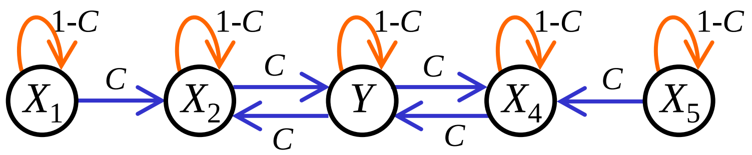
EXAMPLES: SIMULATION 5

- **Simulated Coupled Nonlinear Henon systems:**

$$X_{i,n} = 1.4 - X_{i,n-1}^2 + 0.3X_{i,n-2} \quad , \quad i = 1, M$$

$$X_{i,n} = 1.4 - \left[0.5C(X_{i-1,n-1} + X_{i+1,n-1}) + (1-C)X_{i,n-1}\right]^2 + 0.3X_{i,n-2} \quad , i = 2, \ldots, M-1$$

Process graph (M=5):



This simulation shows the necessity of using non-uniform embedding for quantifying the information transfer in non-linear systems composed by many subsystems where model-free estimators are more appropriate;

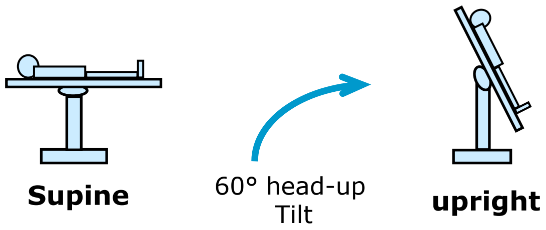
In the script, the nearest neighbor or kernel estimators are employed, and simulation parameters to test are the number of nodes, the coupling strength, and the data length

- Simulation script:
  - **Example_simu5_UEvsNUE.m**

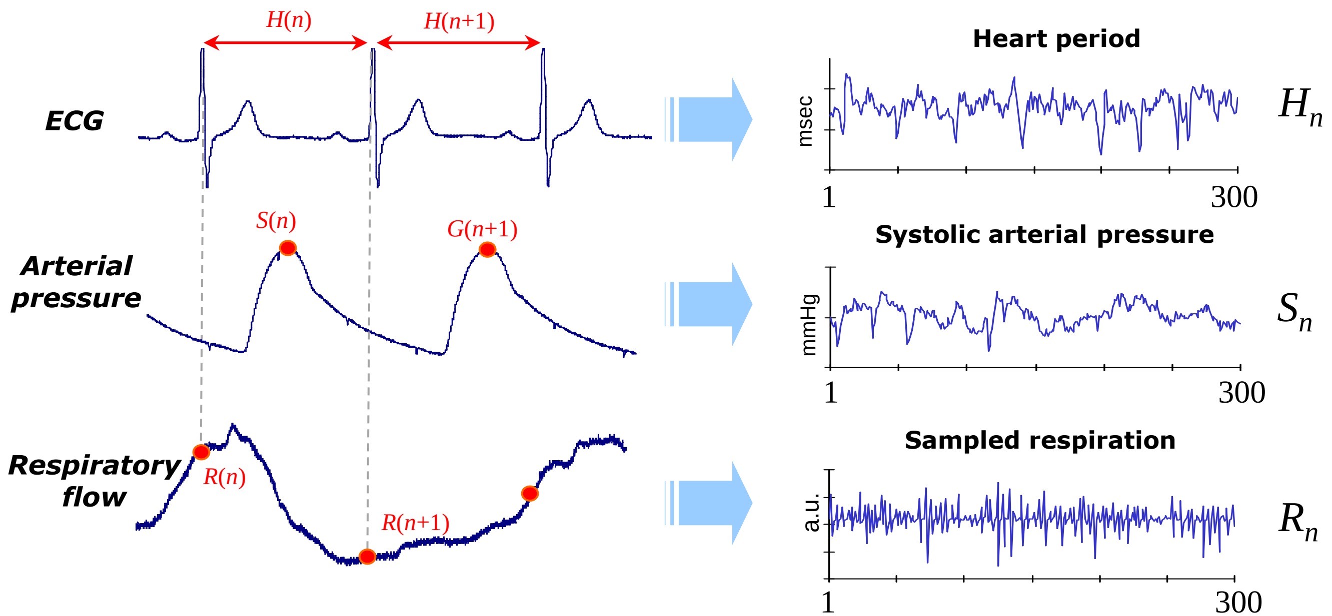
  (sim_coupledhenonmaps2.m)

EXAMPLES: APPLICATION TO CARDIOVASCULAR VARIABILITY

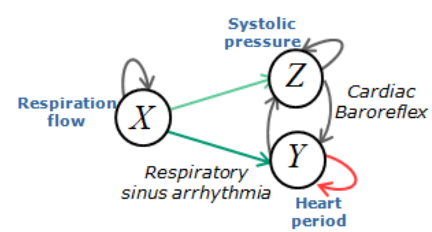
- **Experimental protocol:**
  - ✓ Young healthy subject
  - ✓ Head-up tilt test



- **Cardiac system** Y
- **Vascular system** Z
- **Resp. system** X



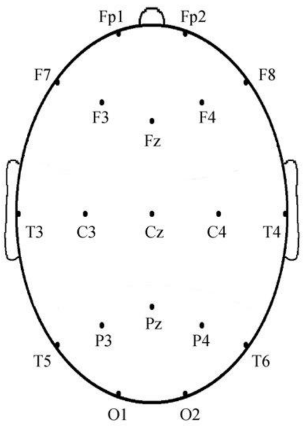
- Application scripts: **Example_RR.m** *Univariate analysis of heart period, UE, all estimators*
  - **Example_Cardio_lin.m** *Linear, full analysis, order selection*
  - **Example_Cardio_bin.m** *Binning, full analysis*
  - **Example_Cardio_ker.m** *Kernel, full analysis*
  - **Example_Cardio_knn.m** *Nearest neighbors, full analysis*



- Data: `supine.prn`
  `upright.prn`

EXAMPLES: APPLICATION TO EEG

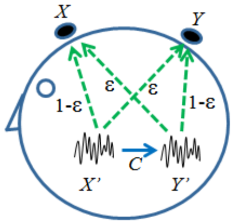
- **Experimental protocol:**
  - ✓ Young healthy subject
  - ✓ Acquisition of EEG signals with eyes closed in the relaxed awake state
  - ✓ Acquisition: international 10-20 system (128 Hz)
  - ✓ Pre-processing:
    - bandpass filter (0.3-40 Hz)
    - selection of 8 sec window



- **Information transfer in the presence of volume conduction**

Implementation of a compensation for instantaneous causality effects in the computation of BTE and PTE

L Faes, G Nollo, A Porta: 'Compensated transfer entropy as a tool for reliably estimating information transfer in physiological time series', Entropy; special issue on "Transfer Entropy", 2013; 15(1):198-219.



- Application script: **Example_EEG.m**
- Data: `EEG_EyesClosed.mat`
- Additional functions: *Non-uniform embedding with initial embedding vector passed as input*

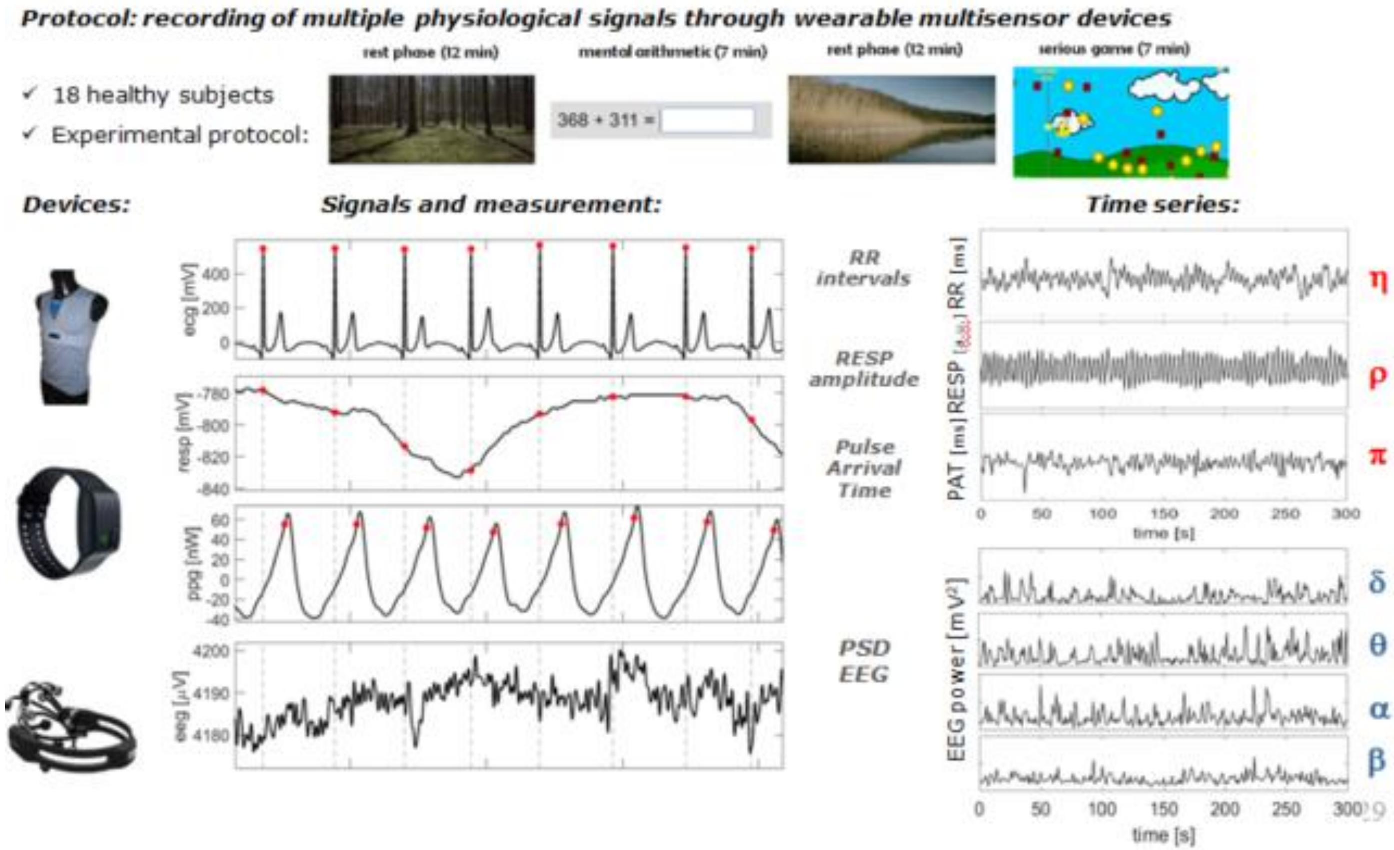
•its_NUEknn_Vstart.m

•its_BTEknn_0.m
•its_PTEknn_0.m *Small modification to allow conditioning to lag-zero terms*

(AR_filter.m) *Autoregressive high-pass (de-trending) filter*

EXAMPLES: APPLICATION TO BRAIN-BODY INTERACTIONS

- **Experimental protocol and time series:**



- Data: `BrainBodyStress.mat`
- Application scripts: **`Example_BrainBodyStress_1.m`**

  Maps of mutual information between body and brain, and of the information transfer along the two directions

  `Example_BrainBodyStress_2.m`

  Computation of information storage, information transfer (total+conditional) for a given scalp location

New functions – added in 2019 (v.2.1)

- In the computation of the bivariate TE, the driver process can be multivariate (index i passed as a vector), but the target process must be univariate (index j is scalar). To overcome this limitation, the following functions are added to the toolbox:
  - **its_MIknn_V.m**
  - **its_CMIknn_V.m**
  - **its_NUEknn_V.m**

    Extension to multivariate target process of the computation of mutual information and conditional mutual information, and of the non-uniform embedding procedure (nearest neighbor estimator)

  - **its_CEknn_V.m** *Extension to multivariate target process of the Conditional Entropy for the nearest neighbor estimator*

  - **its_BTElin_V.m** *Extension to multivariate target process of the bivariate Transfer Entropy for the linear estimator*
  - **its_BTEknn_V.m** *Extension to multivariate target process of the bivariate Transfer Entropy for the nearest neighbor estimator*

  - **its_JBTEknn_VS.m** *For the nearest neighbor estimator, computes the joint transfer entropy from two (possibly multivariate) source processes to a scalar target process, and computes also the terms of its decomposition (individual and conditional transfer entropies from one source to target) through distance projection.*