LINC

Learning about Interacting Networks in Climate

Marie Curie Initial Training Networks (ITN)

FP7- PEOPLE - 2011- ITN

Grant Agreement No. 289447

WorkPackage WP3: Natural Climate Variability

Deliverable D3.2

Software package for mechanistic indicators (based on network properties) for identification of physical mechanisms of climate variability

H.A. Dijkstra. Utrecht University, The Netherlands

Release date: 30 November 2013

Status: public

EXECUTIVE SUMMARY

This deliverable involves a software package which was used for the results in Deliverable 3.3. The software is made available on the LINC website (http://www.climatelinc.eu/home/).

Deliverable Identification Sheet

Grant Agreement No.	**PITN-GA-2011-289447**
Acronym	**LINC**
Full title	**Learning about Interacting Networks in Climate**
Project URL	http://climatelinc.eu/
EU Project Officer	Lucia PACILLO

Deliverable	**D3.2 Software package for mechanistic indicators (based on network properties) for identification of physical mechanisms of climate variability**
Work package	**WP3 Natural Climate Variability**

Date of delivery	**Contractual**	M 24	**Actual**	30-11-2013
Status	version. 1.00		final ☑ draft □	
Nature	Prototype □ Report ☑ Dissemination □			
Dissemination Level	Public □ Consortium ☑			

Authors (Partner)	H.A. Dijkstra (UU)			
Responsible Author	H.A. Dijkstra		**Email**	H.A.Dijkstra@uu.nl
	Partner	UU	**Phone**	31-30-2535441

Abstract (for dissemination)	This document contains a short description of the software package associated with Deliverable D3.2 of WP3 of the LINC project.	
Keywords	Natural Climate Variability, Complex Networks	

Version Log			
Issue Date	**Rev No.**	**Author**	**Change(s)**
30-11-2013	001	H.A.Dijkstra	None

SOFTWARE PACKAGE FOR MECHANISTIC INDICATORS (BASED ON NETWORK PROPERTIES) FOR IDENTIFICATION OF PHYSICAL MECHANISMS OF CLIMATE VARIABILITY

ALEXIS TANTET, HENK DIJKSTRA

Contents

1. Introduction

In this document, we summarize some of the tools from statistics and network theory which can be used to study climate variability. A toolbox written in Fortran and Python is provided to conduct such analysis.

Most of the analysis relies on the *Pearson correlation.* It is important to test the significance of these correlations under the null hypothesis of zero-correlation. To account for the non-gaussianity and the persistance of the data, a non-parametric test is used, the *moving block bootstrap* test.

The *Empirical Orthogonal Function* analysis (EOF) is a commonly used method to identify spatial patterns of variability. However, this method often fails to represent regional patterns of variability from global data due to the strong orthogonality constraint. Rotated EOFs can be used to overcome this limitation. However, the justification of the criterium used to rotate the EOFs (usually simplicity

Date: November 28, 2013.

function) is not straightforward and can mislead the interpretation of statistical patterns.

The interaction network approach offer a novel representation of climate data by focusing on the links between grid points. These networks can be built from the correlation matrix. In particular, we show that the communities of a network can be associated with spatial patterns of variability of different spatio-temporal time-scales. Time-series associated to the communities can be calculated which offer an analog to the expansion coefficients associated to EOFs.

Most of the numerical tools presented in this document will refer to provided Fortran source codes and Python scripts.

2. Calculation and test of the correlation matrix

2.1. Pearson correlation matrix.

The sample Pearson correlation gives an estimate of the true correlation. The Pearson correlation coefficient r_{ij} between two time series $(p_i(t_k))_{1\leq k\leq L}$ and $(p_j(t_k))_{1\leq k\leq L}$ is given by:

$$r_{ij} = \frac{\sum_{k=1}^{L} p_i(t_k)p_j(t_k)}{\sqrt{(\sum_{k=1}^{L} p_i^2(t_k))(\sum_{k=1}^{L} p_j^2(t_k)))}}.$$

correlation.f90:pearson_corr_vect: calculates the Pearson correlation of two vectors. This routine relies on BLAS, an optimized linear algebra library.

Because of the limited number of degrees of freedom of the time series, the statistical significance of this estimate should be tested under the null hypothesis of zero-correlation.

2.2. Moving Block Bootstrap test of the correlation matrix.

Climate time series are often non-gaussian and serially correlated. These properties make it difficult to use parametric significance tests like the t-test. This difficulty can be overcome using a non-parametric test such as the Moving Block Bootstrap (MBB). A more thorough presentation of the MBB applied to correlations between climate time series is given in the note_boostrap.pdf document and a general treatment is done in Mudelsee (2010).

correlation.f90:mbb_bivar_pval_field: can be used to estimate the p-values of an estimator using the MBB.

3. Spectral analysis of the correlation matrix

3.1. Empirical Orthogonal Functions (EOF).

An EOF analysis consists in finding an orthogonal basis E such that the variance of the projection of the data

D on the vectors of E is maximized which comes down to solving the eigen-value problem:

$$D^t DE = E\Lambda$$

where D is the dataset, E the basis of eigen-vectors and the Λ a diagonal matrix with corresponding eigen-values. The mean values must be removed from the dataset, such that the matrix product $D^t D$ is in fact the covariance matrix or the correlation matrix if D was normalized by the standard deviations.

Because physical spatial patterns of variability exhibit a coherent spatial signature, one would expect such spatial signature to be described by the EOFs. However, these patterns need not to be orthogonal, in particular regional patterns of variability.

eof_analysis.py: The first part of this Python script shows how to use Numpy to find the EOFs of a dataset.

3.2. **Rotated EOF.** To overcome this orthogonality constraint, the EOFs or at least the dominant EOFs can be rotated which means that the rotated EOFs will be linear combinations of the the standard EOFs. However, a criterium has to be defined in order to find the rotation matrix. A commonly used criterium is Kaiser's normalized varimax. This criterium can be seen as a simplicity function minimizing the spatial variance of the squared amplitude of the EOFs.

eof_analysis.py: The second part of this Python script shows how to use the varimax function to find rotated EOFs.

However, the justification of the criterium used to rotate the EOFs (usually simplicity function) is not straightforward and can mislead the interpretation of statistical patterns.

4. Network Construction and Analysis

The script of the construction and analysis of a climate network as presented below is attached to the present document as:

network.py

4.1. **Construction of the adjacency matrix.** Interaction networks allow to work on the same mathematical object as the EOFs, namely, the correlation matrix, but with a different mathematical framework where variables correspond to nodes and their interaction to links. First of all, a matrix describing the topology of the network, namely, the adjacency matrix, must be built from the correlation matrix. A simple way to do so is to construct an undirected and unweighted network by thresholding the correlation matrix. A link between two nodes is defined as follow: node (grid-point) i is linked to node j if and only if $|r_{ij}| > \tau$ where τ is a chosen threshold. The elements of the adjacency matrix A of the network are thus defined as:

$$a_{ij} = \Theta(|r_{ij}| - \tau) - \delta_{ij}$$

where Θ and δ are the Heaviside and Kroenecker distributions, respectively. Thus $a_{ij} = 1$ if node i is connected to node j, $a_{ij} = 1$ otherwise.

The choice of the threshold τ is not straightforward. First of all, one might choose a minimum threshold in order to keep only significant correlations. Secondly, one could choose a threshold heuristically in order to keep only strong correlations more likely to represent physical connections. For exemple, in the study of climate variability, the threshold should allow for known teleconnections for which a physical mechanism is known to be represented in the network. Finally, in order to avoid defining a threshold, a weighted network can be built in which the information of the intensity of the correlation is kept, such that weak correlations are still represented but negligible.

Once the network is built, many of its properties can be analyzed. The degree centrality and the community structure of the network appear to be particularly relevant to the study of climate variability.

4.2. **Degree.** The degree of a node is simply its number of links with other nodes of the network. The degree d_i of node i can be calculated from the adjacency matrix as follow:

$$d_i = \sum_{j=1}^{n} a_{ij}$$

As such, nodes with high degree are nodes which are highly correlated to many other nodes. These nodes are thus likely to belong to spatial patterns of variability influencing a large part of the domain. The representation of the degree is usually similar to the first EOF but other modes can also be apparent. However, it is not possible to distinguish the different spatial patterns from the degree. For this matter, one can use algorithms to detect the community structure of the network.

4.3. **Community detection.** Communities are groups of nodes tightly connected with each other, sparsely connected to the rest of the network. Since nodes influenced by the same spatial pattern of variability tend to be connected with each other and not to nodes influenced by distinct patterns of variability, these communities should be related to spatial patterns of variability. If each community was a fully disconnected component of the network, one could find these components analytically by calculating the eigen-vectors of the graph laplacian. However, this is usually not the case and numerical algorithms must be used. Many algorithm have been proposed in the literature, some also rely on the graph Laplacian, some of aggregating or dividing algorithm trying to maximize a quality function such as the modularity. In our case, the Infomap algorithm from Rosvall (2008) offered good results. It uses the Louvain algorithm (aggregating) to find partitions which minimize the code length required to code the motion of random walkers inside and between communities. Moreover, this algorithm is multi-scale, meaning that it can detect hierarchical communities.

The *network.py* file uses the *Infomap* executable to detect the communities which should be compiled from source first. The Infomap executable reads the graph from disk as Pajek file which must be saved using the *write_pajek* function from the *iograph.py* module. The result of the Infomap algorithm is written in a tree file which can be read using the *read_tree* function from the *iograph.py* module.

4.4. **First neighbours of the communities.** Once the community structure known, it is interesting to look at the connection between the communities and the rest of the network, first to verify the quality of the community itself and secondly to look at links between communities which can be regarded as teleconnections. One way to do so is to look at the first-neighbours map of a community which describe the fraction of nodes belonging to a given community any node in the network is connected to. It is defined as

$$FN_{n_i \to c_j} = \frac{1}{N_{c_j}} \sum_{n_j \in c_j} a_{ij}$$

where c_j is the set of N_{c_j} nodes belonging to community j and $FN_{n_i \to c_j}$ the fraction of nodes in community c_j node n_i is connected to.

4.5. **Community time-series.** It would be interesting to have a time series associated to each community just as expansion coefficients are associated to EOFs. One way to do so is to spatially average the time series of every node of the community which is equivalent to projecting the dataset on a community vector C_j with element $C_{ij} = 1./N$ if node n_i belongs to community c_j, $C_{ij} = 0$ otherwise. Thus, these community time series are to communities what expansion coefficients are to EOFs. As an example, the correlation between these community time series and any other index can be calculated to study the relationship between this community and the index.