Source code documentation for ‘Characterizing unidirectional couplings between point processes and flows’ - Ralph G. Andrzejak and Thomas Kreuz

8. December 2011.

The source code allows you to calculate the measure L introduced in Andrzejak RG, Kreuz T: Characterizing unidirectional couplings between point processes and flows. EPL, 96 (2011) 50012. Please cite this paper if you use this code to calculate the measure L. In order to understand the following documentation, you should at first read the paper. In case you need a reprint or have any questions, please contact the authors at ralphandrzejak at yahoo dot de.

Below we use the example of the coupled Hindmarsh-Rose dynamics studied in [1] to illustrate the code. In particular we have:

Signal1: A flow of the dynamics X
Signal2: A flow of a replica surrogate of the dynamics X
Signal3: A point process derived from dynamics Y.

All signals are of length 200T (see again [1] for the definition of T). The dynamics X is driving dynamics Y with epsilon = 0.1442. The notation for the following source code parameters corresponds to the one used in [1].

`q`: window length
`s`: step size
`W`: Theiler window
`Q`: total duration,
`N`: Number of windows
`k`: Number of nearest neighbours
`delta_t`: sampling step

Remark on time unit convention: The time step between subsequent entries in the matlab arrays of flow signals corresponds to `1*delta_t`. In other words, by going from the element with index i to the element with index i+1, we make a step of `delta_t`. Accordingly, the parameters `q`, `s`, and `Q` are specified in units of `delta_t`. The parameters `q`, `s` and multiples thereof can directly be used as array indices, and `Q` is the length of the flow arrays. Furthermore, the spike times of the point process signals are also specified in units of `delta_t` and therefore one should use `delta_t=1` as input for the function ‘`AndrzejakKreuzDpointprocess`’ below. Accordingly, the length of the flow signals is 400.000 samples. And the spike times are bounded by 0 and 400.000. `N` and `W` are in units of time windows.

To run the code copy: `AndrzejakKreuzExampleData.mat`, `AndrzejakKreuzExample.m`, `AndrzejakKreuzL.m`, `AndrzejakKreuzDflow.m`, `AndrzejakKreuzDpointprocess.m`, `f_SPIKE_ISI_distance_new.m` into some directory and call `AndrzejakKreuzExample.m` Please look through the code and read the following short documentation of the main functions.

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

A distance matrix from a flow is calculated using:

```
function DF=AndrzejakKreuzDflow(flow,Q,q,s,W)
```

Here `flow` is a 1-D vector of length `Q` containing the sample values of the flow. The output `DF` is a `(N,N)` distance matrix. All other parameters are explained above.

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

A distance matrix from a point process is calculated using

```
function DP=AndrzejakKreuzDpointprocess(spiketimes,Q,q,s,W,dt,dchoice)
```

`spiketimes is` a 1-D vector of containing the times of spikes. The length of the vector is determined by the number of spikes.

`dchoice`: In reference [1] we used the ISI distance. With this code you can also use the SPIKE distance (see references [2-5] for more information on these distances.). Depending on the value of `dchoice`, you get as output is the either ISI distance, the SPIKE distance or both:

If you set
`dchoice=1: DP` is a 2-D array of size `(N,N)` containing the SPIKE distance.
`dchoice=2: DP` is a 2-D array of size `(N,N)` containing the ISI distance.
`dchoice=2: DP` is a 3-D array of size `(2,N,N)` containing the SPIKE distance in `DP(1,:,:)` and the ISI distance in `DP(2,:,:)`.

All other parameters are explained above.

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

The main function is:

```
function L = AndrzejakKreuzL(DArray,k,W)
```

`DArray` is a 3-D array of size `(N,N,ND)`. It contains `ND` distance matrices of size `(N,N)`. In our example we have 3 signals. Accordingly, `ND=3`. For your own data you can use any number of `ND>1` distance matrices (At some point limited by the memory of your computer). You do not have to specify `ND`. It will be detected automatically from the size of `DArray`. The individual distance matrices can be obtained from flows and/or point processes. The order used in our example is arbitrary. Whatever order you choose, will also be used for the output:

`L`: a 2-D array of size `(ND,ND)`. In element `L(a,b)` we have L(a|b). For example `L(2,3)`contains L(signal2 | signal3) = L(replica surrogate of flow X | point process derived from dynamics Y). The diagonal is set to zero. For the present example, since X is driving Y, we get a high value for L(signal 1 | signal 3) = L(flow X | point process derived from dynamics Y). Lower values are obtained for the other non-diagonal entries.

▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪

The following optimisation can be carried out depending on your matlab version:

In `AndrzejakKreuzL.m` line 45 you can replace

```
[dummy, Hindexes(:,:,counter3)] = sort(Hdistances(:,:,counter3));
```

by

```
[~, Hindexes(:,:,counter3)] = sort(Hdistances(:,:,counter3));
```

In `AndrzejakKreuzDpointprocess` line 22 you can replace

```
[dummy,isi,time_profiles]=f_SPIKE_ISI_distance_new(spikemat,0,duration
,dt,mod(dchoice,4));
```

by

```
[~,isi,time_profiles]=f_SPIKE_ISI_distance_new(spikemat,0,duration,dt,
mod(dchoice,4));
```

▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪

References:

[1] Andrzejak RG, Kreuz T: Characterizing unidirectional couplings between point processes and flows. EPL, 96 (2011) 50012.
[2] Kreuz T, Chicharro D, Greschner M, Andrzejak RG: Time-resolved and time-scale adaptive measures of spike train synchrony. J Neurosci Methods, 195 (2011) 92.
[3] Kreuz T, Chicharro D, Andrzejak RG, Haas JS, Abarbanel HDI: Measuring multiple spike train synchrony. J Neurosci Methods 183 (2009) 287.
[4] Kreuz T, Haas JS, Morelli A, Abarbanel HDI, Politi A: Measuring spike train synchrony. J Neurosci Methods 165 (2007) 151
[5] http://www.fi.isc.cnr.it/users/thomas.kreuz/Source-Code/Spike-Sync.html