Software metapaper for submission to the Journal of Open Research Software

To complete this template, please replace the blue text with your own. The paper has three main sections: (1) Overview; (2) Availability; (3) Reuse potential.

Please submit the completed paper to: editor.jors@ubiquitypress.com

---

(1) Overview

Title
DST (Dynamic Synchronization Toolbox): A MATLAB implementation of the dynamic phase-locking pipeline from *Stimulus transformation into motor action: Dynamic graph analysis reveals a posterior-to-anterior shift in brain network communication of older subjects.*

Paper Authors
1. Rosjat, Nils[1,2,*]; $(Lead/corresponding author first)$
2. $Daun, Silvia$[1,2]

Paper Author Affiliations
1. Cognitive Neuroscience, Institute of Neuroscience and Medicine (INM-3), Jülich, Germany
2. Institute of Zoology, University of Cologne, Cologne, Germany

* Contact: n.rosjat@fz-juelich.de

Abstract
The Dynamic Synchronization Toolbox allows the calculation of dynamic graphs based on phase synchronization in experimental data. This enables an analysis of the time-development of network connectivity between multiple recording sites (e.g. in electroencephalography (EEG) or magnetoencephalography (MEG) data) with a high temporal resolution. Optionally, the toolbox offers the possibility to compute several graph metrics (such as cluster dynamics, node degree, HUB nodes) via the Brain Connectivity toolbox.

Keywords
M/EEG; Dynamic graph; phase synchronization; brain networks; MATLAB; Connectivity.

Introduction
The Dynamic Synchronization Toolbox (DST) is an implementation of the pipeline used in our previous article (Rosjat et al. (2021)) to calculate phase connectivity based dynamic graphs. This study dealt with the analysis of EEG data recorded during externally triggered finger movements in young and older subjects. We used this pipeline to get in depth information on the connectivity dynamics during

movement planning and movement execution and especially how these dynamics are changed in healthy aged persons. For this purpose it was necessary to use a *non-static connectivity measure* which can detect instantaneous changes in the coupling of the signals. Compared to other toolboxes that allow connectivity analyses of EEG signals, we focused here on ensuring the highest possible temporal resolution. Other toolboxes such as Dynamic Causal Modeling (DCM, Kiebel et al. (2009)) as a function of the Statistical Parametric Mapping toolbox (SPM, Ashburner et al. (1994)) for instance, require static connectivity and thus do not allow for time course analysis. Also, connectivity approaches, like Granger Causality (Guo et al. (2010)) or Mutual Information (Pereda et al. (2005)), assume stationary signals and thus have a fairly low temporal resolution.

The Brain Connectivity Toolbox (BCT, Rubinov and Sporns (2010)) comes into play after connectivity has been determined to perform further network-related calculations. It is a very powerful toolbox that includes different graph theoretic measures. For this reason, we also use this functionality for the calculation of network properties in our toolbox.

The scripts presented here have been generalized as much as possible to allow application not only to EEG data as in our prior publication, but to a wide variety of data as long as they were transformed to *phase space*. Thereby, this toolbox has a high reuse potential also in other fields of research, e.g Geophysics, Geomorphology, Systems Biology, Ecology or Social Network Sciences, that deal with dynamic connectivity, i.e. interactions of signals at different 'recording sites'.

Implementation and architecture

The toolbox presented here provides a MATLAB implementation of the pipeline for creating and graph theoretically analyzing dynamic networks as has been introduced in Rosjat et al. (2021). The pipeline consists of three major steps (Fig. 1): First, phase-locking values between two measuring sites, e.g. electrodes, are computed relative to a defined baseline period, second, the calculated connectivities are used to define dynamic graphs at the group level by testing for significant increase compared to baseline using t-tests and last, the dynamic graphs are analyzed using graph-theoretic measures from the BCT (Rubinov and Sporns (2010)).

In the first step, the input data, that has been epoched and transformed to phase space prior application, undergoes a connectivity analysis based on the relative phase-locking value (rPLV), which is defined as follows:

$$PLV_{m,n}(f,t) = \frac{1}{N} \sum_{k=1}^{N} \exp(i(\varphi_{m_k}(f,t) - \varphi_{n_k}(f,t))) \tag{1}$$

$$rPLV_{m,n}(f,t) = \frac{PLV_{m,n}(f,t) - \overline{PLV_{m,n}(f)}}{\overline{PLV_{m,n}(f)}} \tag{2}$$

where t denotes time, f frequency and φ_{m_k} in Eq. (1) the phase of the signal at measuring site m in the k-th trial. N is the total number of trials, i.e. repeating segments of measurements, and i is the imaginary unit. In Eq. (2) the previously calculated PLV is normalized relative to the mean PLV in a predefined baseline period.
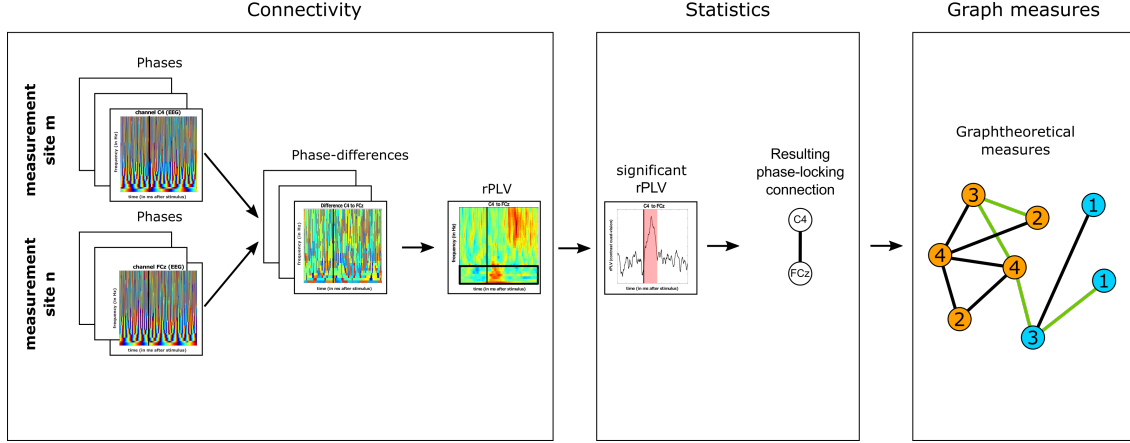
Figure 1: Schematic drawing of the three software parts: connectivity, statistics and graph measures.

In the second step, t-tests are used to define periods of significally increased phase-connectivity for each signal pair. In the toolbox we included the options to compare the rPLV against zero or against an artificially computed baseline signal consisting of random noise with the same mean and standard deviation as the baseline period and of the same length as the test interval. We further implemented three options for dealing with multiple comparisons: (i) no correction (for simple checks), (ii) false discovery rate (FDR) correction for single pairs and (iii) FDR correction pooled over all timepoints and signal pairs. A connection is assigned between a signal pair if the rPLV is significantly greater than zero or the artificial baseline.

Dynamic graphs were then defined as the ordered set $G(t) = \{G_t | t \in [1, \cdots, T]\}$ of binary undirected graphs $G_t = (V, E_t)$, where the graph G is defined by a set of vertices V and edges $E_t : V \times V \to R$, for each point in time t.

In the last step, we added the option to compute several graph measures from the BCT. These are, the node degree (i.e. the number of connections of a node), a Louvain community detection (also called clustering), the node flexibility (i.e. how often the nodes switch between the clusters) and the HUB nodes (i.e. the most influencial nodes) of the networks. We determined all of these measures at all given points in time to obtain their overall dynamics. For a better overview, we added an optimization step for community detection. Here, we always used the previous cluster configuration as the initial condition for the following community detection. Additionally, we post-hoc assigned the label for each community, minimizing the number of label switches between time points, which prevents the same cluster from being assigned different cluster labels at consecutive time points.

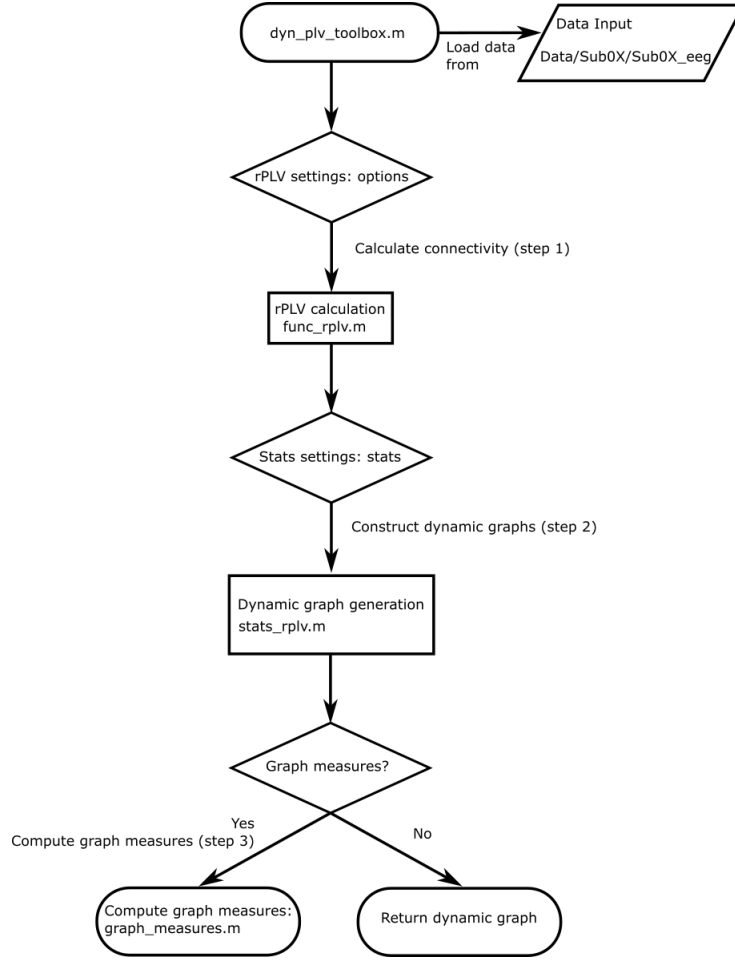A more detailed description of the methods used can be found in Rosjat et al. (2021).

Figure 2: Flowchart depicting the logical sequence of the dynamic graph calculations.

Script architecture

The DST is fully implemented in MATLAB. It includes a main directory consisting of the master function `dynamic_synchronization_toolbox_function.m`, a documentation file `readme.m` and subfolders for data and scripts.

The master script is used to specify various options used for further processing. In this script we define the subject IDs, which will afterwards be loaded from the subfolders of the same name in the subject folder, e.g.:

```
subjects = ['Sub01';'Sub02'];
```

These subject folders are organized in a BIDS (Brain Imaging Data Structure; Gorgolewski et al. (2016); Pernet et al. (2019)) like fashion, i.e. each subject has its own subfolder within the data folder. These subfolders then consist of a mat-file containing the epoched EEG data transformed to phase space and a trial definition file (in case of multiple experimental conditions).

Within the master script, an `options` structure is defined which sets several options for calculating the rPLV in step 1. See Table 1 for all available options. The rPLVs are then computed by calling the function `rplv_func.m` (Fig. 2, step 1).

option	type	description
electrodes	integer list	subset of electrode indices of interest
freqs	integer list	frequency range for time-frequency decomposition
baseline	integer	(begin, end) of time-interval for relative baseline
multiple_conds	boolean	single (false) or multiple (true) conditions
switch_hands	boolean	enables mapping of electrodes to other hemisphere
channels_new	integer list	new order of electrodes for mapped condition
channels_old	integer list	old order of electrodes for mapped condition
contrast	boolean	enabling contrasting conditions
contrast_conds	integer list	indices of two conditions to contrast
avg_freqs	integer list	frequencies of interest for averaging

Table 1: Options for rPLV calculation in step 1.

option	type	description
pid	string	multiple comparisons 'original', 'individual', 'uncorr'
pID_fix	double	fixed p-value for corrected stats
p_fix	double	fixed p-value for uncorrected stats
q_FDR	double	q-value for FDR-correction
comp	string	type of comparison 'baseline' or 'zero'
test_interval_start	integer	start of testinterval in ms
test_interval_end	integer	end of testinterval in ms
baseline_start	integer	start of baseline in ms
baseline_end	integer	end of baseline in ms
task	integer	definition of task by id (contrast appears last)
contrast	boolean	enabling contrasting conditions
time	integer list	sampling timepoints
sampling_rate	integer	sampling rate of the data

Table 2: Options for statistical testing and dynamic graph construction in step 2.

Furthermore, we define the `stats` structure which sets several options for statistical analysis of the rPLV in step 2. See Table 2 for all available options. As described above, we included three options for multiple comparison, which can be set by `stats.pid`. The corresponding thresholds for t-tests and FDR correction can be defined in this structure. Additionally, this structure includes an option `stats.comp` to select the desired statistical comparison, i.e. against an artificial baseline (described above) or against zero. Subsequently the function `stats_rplv.m` is called to compute the significant edges and construct the dynamic graphs (Fig. 2, step 2).

The optional calculation of node degree, community detection, node flexibility and HUB nodes can be altered by adjusting the variable `graph_measures = True or False` (Fig. 2, step 3).

Script application

The first step for using the DST pipeline is to arrange the preprocessed data in a BIDS-like data structure in the Data subfolder. The data set of each subject should be stored in a subfolder `SubID` of the `Data` folder and labelled `SubID_eeg.mat`. Each preprocessed (e.g. transformed to time-frequency phase-space) data set has to be stored in a mat-file with the dimensions [channels × frequencies × time-points × trials]. In order to use the pipeline for multiple experimental conditions an additional file called `SubID_trialselection.mat` with dimensions [Trials × Conditions] giving boolean information about the condition of each trial is needed. In the second step, the user has to set the options for the calculation of the synchronization metric in structures (as defined above) `options`, `stats` and `graphs` as well as a list of subject IDs and the path to the subject data subfolder. In the last step, the main function `dynamic_connectivity_toolbox_function.m` has to be executed with the previously defined options. This will at the end produce the following output variables:

rplv: relative phase-locking value for each dataset in a cell {num, subjects, 1}, each cell stores the rPLV with dimensions [time, channel, channel, conditions].

trials: number of trials in each experimental condition and subject.

rplv_mean: group average of rPLV in [time, channel, channel, conditions].

sig_ti_FDR: significant timepoints after statistics and multiple comparisons as a cell {channel, channel}.

xa: list of significant intervals [intervals, 3] with information about start time-point, stop timepoint and the amount of timepoints to the next interval for each channel pair stored in a cell {channel, channel}.

length: list of length of significant intervals for each channel pair stored in a cell {channel, channel}.

The optional graph metric generates the following additional outputs:

Agg: aggregated graph showing the frequency of all connections over the whole interval in a matrix [channel, channel].

bet: temporal betweenness centrality in a matrix [timepoint, channel].

hub: temporal hub nodes, i.e. nodes with highest betweenness centrality, in a matrix [timepoint, 2].

clusters: clusters assignment for each channel and timepoint [channel, timepoints].

node_flex: node flexibility for each channel stored in a matrix [2, channel].

deg: each channels node degree over time in a matrix [timepoint, channel].

Quality control

To ensure sufficient quality control, two sample datasets of artificial data each consisting of four conditions were added to the scripts, one exhibiting a high degree of connectivity and the other a low degree of connectivity. The data are created by running the script `sample_data_creation.m` located in the "Data" subfolder. The data consists of two identical subjects with the dimensions [61 channels, 48 frequencies, 800 timepoints, 100 trials]. A set of 27 channels is perfectly phase-locked after artifical stimulation in the time-interval [300:500] for trials [1:25] and [51:75]. In contrast, the remaining channels are phase-locked in the same interval, but for trials [26:50] and [76:100]. We also provide two figures depicting the rPLV for all possible connections: First, the connectivity is shown for all four conditions in the originally created dataset (File *rplv_sampledata_original.png*) and second, there is a version after switching a set of electrodes and merging conditions 1 with 2 and 3 with 4 (File *rplv_sampledata_merged.png*). In the first case the contrast of conditions 1 and 2 will show a huge difference in rPLV between two distinct sets of measurement sites (e.g. left and right hemispheric electrodes). In the second case this difference should disappear since those two groups were swapped in the second condition leading to equal experimental conditions. These figures are located in the folder "Quality control". This makes it possible to check if the definition of the networks was successful. In the following, this data will also be used to ensure the correct integration of the scripts from the BCT.

The data associated with the original article, analyzed with the toolbox, has been made available Rosjat et al. (2020).

(2) Availability

Operating system

DST is a pure MATLAB code, and should function on all operating systems in which MATLAB is supported.

Programming language

MATLAB (developed in vR2018b).

Additional system requirements

None.

Dependencies

The graph analysis is based on significant edges which depend on the Statistic toolbox (Mathworks Inc. (2019)). The optional calculation of dynamic graph metrics makes use of the Brain Connectivity Toolbox (Rubinov and Sporns (2010)).

List of contributors

Nils Rosjat wrote the software and is its current maintainer. Silvia Daun: Conceptualization, Supervision, Project administration, Funding acquisition

Support

Support requests of any kind are preferably to be submitted in GitHub as an issue or sent to the corresponding author Nils Rosjat via email.

Software location:

Archive Jülich-DATA

> **Name:** Dynamic Synchronization Toolbox
> **Persistent identifier:** https://doi.org/10.26165/JUELICH-DATA/BRXHZ9
> **Licence:** BSD 3-Clause
> **Publisher:** Nils Rosjat
> **Version published:** v1.0
> **Date published:** 02/09/21

Code repository Github

> **Name:** Dynamic Synchronization Toolbox
> **Persistent identifier:** https://github.com/nrosjat/dynamic-synchronization-toolbox
> **Licence:** BSD 3-Clause
> **Date published:** 02/09/21

Language
English.

(3) Reuse potential

This toolbox is of great importance for researchers interested in investigating dynamic connectivity patterns in EEG data based on phase-synchronization. Even though the scripts were developed for a specific analysis of EEG data and performed at the electrode level, they could also be useful for researchers interested in performing source-based connectivity analyses or using different modalities such as MEG. Since the only requirement for the input data of the toolbox is that the signals have to be phase-transformed, the analysis can be applied to any signals with expected phase-locking between different recording sites. Additionally, we generalized the scripts as much as possible, not focusing too much on modality specific features, to increase its reuse potential even further. In the current version the connectivity is calculated based on rPLVs. Here, we presented the scripts as they were used for our prior connectivity analysis (Rosjat et al. (2021)) which focused on the use of rPLV for connectivity analysis. However, the connectivity measure can easily be replaced by other methods, e.g. phase-lag index, coherence, corrected imaginary part of phase-locking value, amplitude based connectivity etc., which will be implemented as an option in future versions. Instead of performing group level statistics one might also use different statistical approaches, e.g. phase-locking statistics (Lachaux et al. (1999)), or thresholding techniques, i.e. fixed amount of edges or ratio of strongest connections, to define subject specific individual graphs. This might be of special interest for patient-related research where only few subjects are available or subjects are too diverse to be analyzed on a group level. These techniques can also be applied in the case of continuous recordings, e.g. resting state (M/EEG) data, here, however, the connectivity measure has to be adjusted to account for single-trial connectivity instead of the inter-trial connectivity that was used here.

This toolbox provides a pipeline beginning at the level of phase-transformed data, constructing dynamic connectivity networks using statistical analysis and ending

with the application of certain graph metrics from the BCT (Rubinov and Sporns (2010)). The format of the generated output makes it easy to incorporate even more sophisticated graph metrics if needed. Therefore, this toolbox provides a good starting point for researchers studying phase-connectivity in a wide range of signals since it provides a pipeline for their entire analysis.

Funding statement

Competing interests

The authors declare that they have no competing interests.

References

Ashburner, J., Barnes, G., Chen, C.-C., Daunizeau, J., Flandin, G., Friston, K., Glauche, V., Henson, R., Hutton, C., Kiebel, S. et al. (1994), 'Statistical parametric mapping'.

Gorgolewski, K. J., Auer, T., Calhoun, V. D., Craddock, R. C., Das, S., Duff, E. P., Flandin, G., Ghosh, S. S., Glatard, T., Halchenko, Y. O. et al. (2016), 'The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments', *Scientific data* **3**(1), 1–9.

Guo, S., Ladroue, C. and Feng, J. (2010), 'Granger Causality: Theory and Applications', *Frontiers in Computational and Systems Biology* pp. 83–111.
URL: *http://link.springer.com/chapter/10.1007/978-1-84996-196-7$_5$*

Kiebel, S. J., Garrido, M. I., Moran, R., Chen, C. C. and Friston, K. J. (2009), 'Dynamic causal modeling for EEG and MEG', *Hum Brain Mapp* **30**(6), 1866–1876.
URL: *http://www.ncbi.nlm.nih.gov/pubmed/19360734*

Lachaux, J. P., Rodriguez, E., Martinerie, J. and Varela, F. J. (1999), 'Measuring phase synchrony in brain signals.', *Human brain mapping* **8**(4), 194–208.
URL: *http://www.ncbi.nlm.nih.gov/pubmed/10619414*

Mathworks Inc. (2019), 'Matlab Statistics and Machine Learning Toolbox'.
URL: *https://www.mathworks.com/help/stats.*

Pereda, E., Quiroga, R. Q. and Bhattacharya, J. (2005), 'Nonlinear multivariate analysis of neurophysiological signals', *Progress in Neurobiology* **77**(1-2), 1–37.

Pernet, C. R., Appelhoff, S., Gorgolewski, K. J., Flandin, G., Phillips, C., Delorme, A. and Oostenveld, R. (2019), 'Eeg-bids, an extension to the brain imaging data structure for electroencephalography', *Scientific data* **6**(1), 1–5.

Rosjat, N., Wang, B. A., Liu, L., Fink, G. R. and Daun, S. (2020), 'Data for: Stimulus transformation into motor action: dynamic graph analysis reveals a posterior-to-anterior shift in brain network communication of older subjects'.
URL: *https://doi.org/10.26165/JUELICH-DATA/T1PKNZ*

Rosjat, N., Wang, B. A., Liu, L., Fink, G. R. and Daun, S. (2021), 'Stimulus transformation into motor action: Dynamic graph analysis reveals a posterior-to-anterior shift in brain network communication of older subjects', *Human Brain Mapping* **42**(5), 1547–1563.
URL: *https://onlinelibrary.wiley.com/doi/10.1002/hbm.25313*

Rubinov, M. and Sporns, O. (2010), 'Complex network measures of brain connectivity: Uses and interpretations', *NeuroImage* **52**(3), 1059–1069.
URL: *http://dx.doi.org/10.1016/j.neuroimage.2009.10.003*