**Behavioral-Tractography-Toolbox**

Table of Contents

[General GitHub Architecture: 1](#_Toc170745208)

[Scripts 1](#_Toc170745209)

[Functions 2](#_Toc170745210)

[Inputs 3](#_Toc170745211)

[Repository Authors 3](#_Toc170745212)

# General GitHub Architecture:

This GitHub repository provides the complete project structure, enabling direct execution. Three distinct folders comprise this structure: scripts, functions, and data. The scripts folder houses the primary MATLAB files responsible for the project’s core functionality. These scripts rely upon functions stored within the functions folder to execute specific tasks. Finally, the data folder acts as a repository for any requisite MATLAB data files (typically .mat format) that the scripts necessitate for operation. For any inquiries concerning the project structure or if further elucidation is required, please do not hesitate to contact the repository's authors.

# Scripts

In the following section we provide a description of the overall structure and contents of the different Matlab scripts required to perform the analysis described in the main text. The reader is referred to the main text for a general overview of rational justifying each step of the analysis pipeline. Moreover, a precise description of the algorithms employed to implement the analysis is provided as commentary to the Matlab code.

**General description of the contents and purpose of different script:**

* **MLN\_Network\_1:**

This scripts contains the analysis pipeline required for construction and visualization of 3D-Multi-Layer-Networks from time-series EMA data. The Inputs file that is loaded by the script and freely available contains the EMA timeseries data collected in the Typically-Developing-Replication sample described in the main text.

The scripts firstly computes both cross-sectional adjacency matrices using Mixed-Models-Linear-Regression. Then Network-Dimensionality-Reduction (NDR) is performed on such cross-sectional adjacency matrix and results of NDR are tested in terms of correlation of Euclidian-Distance and correlation strength as described in the main text. Subsequently a Multi-Layer Adjacency matrix is constructed considering both cross-sectional and longitudinal correlations between EMA variables. Such Multi-Layer Adjacency matrix is then characterized in all the metrics required for 3D-MLN visualization. For instance, the size of Network nodes reflects the average connection strength of underlying EMA variables. Moreover, Graph-Theory analysis is performed to compute shortest paths connecting EMA variables across temporal layers. Such shortest path information is represented in the arrow settings of the 3D-MLN visualization and is also employed to compute Dynamic-Betweenness-Centrality measures for each EMA-Variable. As described in the main text such Dynamic-BTW estimates are then compared against a distribution of randomly structured networks to identify EMA variables that act as Gateways or Funnels for psychological-contextual interactions, and which are represented in the color-coding of network nodes.

The relevant information required for representation of 3D-MLN is then directly embedded in a JSON file which can be uploaded in a dedicated online visualization platform (https://dev.mlnetwork-diplab.ch/), that is automatically launched at the end of the MLN\_Network\_1 script.

* **MLNetwork web platform for 3D-Multi-Layer-Network Visualization:**

JSON files generated by the MLN\_Network\_1 scripts can be uploaded on the dedicated 3D-MLN web platform for visualization of 3D-Multi-Layer-Networks (https://dev.mlnetwork-diplab.ch/). JSON files can be uploaded manually or via API. The platform allows visualization and 3D manipulation (rotation, translation, zooming) of 3D-Multi-Layer networks. Moreover, the Network can be directly modified by selecting sub-portions of the network or modifying visualization parameters, including color and size of nodes and edges. Such changes can be exported in a JSON file that has an identical format as the originally uploaded one. The platform allows to export high-resolution figures or videos of the 3D-MLN networks. A link to the 3D-MLN representation of the main networks described is available in the main text.

* **Behavioral-Tractography and Behavioral Diffusion Analysis:**

This scripts contains the analysis pipeline required to compute both Behavioral-Tractography (BT) and Behavioral-Diffusion-Analysis (BDA) metrics. Refer to the main text for the description and rationale for the use of these measures. The inputs file that is loaded by the script and freely available contains the 3d-MLN and the cross-sectional coordinates both obtained with the pipeline described above (MLN\_Network\_1).

The first step to compute both BT and BDA, consists in computing the shortest paths connecting variables across temporal layers, as in MLN\_Network\_1. Subsequently trajectory of each path is codified as the 3D coordinates of the variables it traverses. Specifically, each path is decomposed in 4 XYZ coordinates of its TL1-starting-node, TL1-exit-node, TL1-entry-node and TL2-ending-node.

Then K-means clustering is applied to these coordinates, in order to dissect “bundles” of psychological-contextual pathways that shared similar 3D trajectories. The optimal number of Clusters/Bundles was determined using a consensus procedure across all samples, described in supplementary analysis 1. To estimate BDA at the level of each symptom, we averaged the trajectories that each symptom mediates within each BT bundle.

The second part of the script describes the analysis pipeline for visualization of BT results. A JSON is generated for each BT bundle, which can be uploaded in a dedicated online visualization platform (https://dev.mlnetwork-diplab.ch/), that is automatically launched at the end of the script.

# Functions

The scripts within this repository rely on a complement of internal and external functions to achieve their functionality.

External functions, essential for successful execution but not included within the repository, require separate download and installation. Links to the relevant download pages for these external functions are provided for user convenience. These external functions typically originate from established toolboxes or resources.

Internal functions were developed by the project's authors specifically for use within these scripts.

Both internal and external functions contribute to various aspects of the scripts' functionality, encompassing network analysis computations, data manipulation, and output formatting. The list of all functions, including detailed descriptions and source attributions (for external functions), can be found below in the documentation.

1. **BTW\_SOURCE\_TARGET\_DIR**

This function calculates the betweenness centrality for all nodes within a network. However, unlike standard betweenness centrality measures, it exclusively considers longitudinal shortest paths.

1. **Cross\_corr\_c**

This function utilizes a mixed-effects model framework to populate an adjacency matrix representing cross-sectional network connectivity between EMA items.

1. **MIXED\_CORR\_MATRIX\_c**

This function, similar to Cross\_corr\_c, employs a mixed-effects model to populate an adjacency matrix, but in this case, it focuses both on cross-sectional and longitudinal network connections. By incorporating longitudinal information, the function provides a representation of how network connectivity evolves over time.

1. **jUpperTriMatToVec**

This function serves to transform the upper triangular portion of a matrix into a vector. The upper triangular portion encompasses elements above the main diagonal of the matrix. This transformation can be useful for specific data analysis tasks that require a vectorized representation of the upper triangular elements.

1. **Contraire\_jupperT**

This function performs the inverse operation of jUpperTriMatToVec. Given a vector, it reconstructs the corresponding upper triangular matrix. This functionality is particularly relevant when the analysis workflow requires manipulating data in both matrix and vector formats.

1. **Poseweight**

This function filters an adjacency matrix, retaining only positive connections. Adjacency matrices represent network connections, where positive values indicate a link between nodes. By isolating positive connections, the function facilitates the analysis of directed or weighted networks.

1. **Brain Connectivity Toolbox**

The scripts rely on functionalities provided by the Brain Connectivity Toolbox (BCT). This toolbox is a comprehensive suite of functions specifically designed for network analysis in neuroscience research. Users are required to download BCT prior to running the scripts. Additional information and download instructions can be found at: <https://sites.google.com/site/bctnet/home>

1. **fdr\_bh**

This function implements the Benjamini-Hochberg (1995) procedure for controlling the false discovery rate (FDR) in a set of hypothesis tests. The FDR is a statistical concept used to adjust for multiple comparisons and reduce the risk of identifying false positives. The function can be accessed through the MATLAB Central File Exchange: <https://ch.mathworks.com/matlabcentral/fileexchange/27418-fdr_bh>

1. **Prettyjson**

This function enhances the readability and formatting of JSON data structures. Additional information and access to the function can be found at: <https://ch.mathworks.com/matlabcentral/fileexchange/72667-prettyjson-m>

# Inputs

This section details the essential input files necessary for successful script execution. Please ensure all these files are present in the designated data folder before running the scripts.

1. **Coord\_1 (Nodes 2D Coordinates):** This file provides the two-dimensional (2D) spatial coordinates for each node within the network..
2. **EMA\_data (Ecological Momentary Assessment Data):** This file houses the Ecological Momentary Assessment (EMA) data employed by the scripts throughout the analysis pipeline. EMA data typically captures real-time experiences and self-reported information from participants in their natural environments.
3. **Matrix\_1 (Multi-Layer-Network):** This file contains a representation of the network's connectivity structure both at cross and longitudinal level.
4. **Names\_3\_v4 (Nodes Names):** This file provides a list of unique names or identifiers associated with each node within the network.

# Repository Authors

Corrado Sandini

corrado.sandini@unige.ch

Andrea Imparato

andrea.imparato@unige.ch