

# Getting Started

The BRAPH 2 Team

January 4, 2025

This tutorial provides a concise introduction to the BRAPH-2 software framework, which enables comprehensive analyses of brain connectivity using conventional graph theory, advanced multilayer network, and deep learning approaches. BRAPH-2 has been designed following an object-oriented paradigm and supports workflows involving structural, functional, and connectivity data derived from multimodal neuroimaging techniques (e.g., MRI, dMRI, fMRI). It offers a set of ready-made pipelines to compute graph measures, perform group-level comparisons, and apply classification or regression tasks with deep learning, all accessible via an intuitive graphical user interface (GUI) or through MATLAB scripts. BRAPH-2 also facilitates more advanced network neuroscience investigations by enabling the construction and analysis of multilayer graphs, permitting the user to probe how multiple types of data interact across layers (e.g., combining anatomical and functional information). To ensure reproducibility and expandability, BRAPH-2 integrates a “lock in” mechanism that saves and freezes data, parameters, and random seeds along the analysis pipeline. Moreover, developers can extend BRAPH-2 by adding new graph measures, neural network architectures, or analysis pipelines. In this tutorial, we outline the major steps for installing and running BRAPH-2, preparing datasets, creating and executing pipelines, and interpreting results, as well as point to developer resources for deeper customization.

## Contents

<a href="#">Introduction</a>	2
<a href="#">Installation and Setup</a>	2
<a href="#">Using the Graphical User Interface (GUI)</a>	2
<a href="#">Launching BRAPH-2</a>	2
<a href="#">Pipeline Steps</a>	3
<a href="#">Basic Tutorial Example</a>	3
<a href="#">Single-Layer Connectivity Analysis</a>	3
<a href="#">Multilayer and Deep Learning Pipelines</a>	4
<a href="#">Automating Analyses with Scripts</a>	4
<a href="#">Extending BRAPH-2</a>	4
<a href="#">Further Information</a>	5
<a href="#">BRAPH-2 GitHub and Discussion Forum</a>	5
<a href="#">User Tutorials for Pipelines</a>	5
<a href="#">Developer Tutorials for Extending BRAPH-2</a>	5

## *Introduction*

BRAPH-2 is an open-source MATLAB-based toolbox for brain connectivity analysis using graph-theoretical approaches. It extends upon the previous version (BRAPH 1.0) by introducing multilayer graphs, deep learning pipelines, improved performance (including GPU support), and a fully modular, object-oriented software design.

A detailed description of the scientific background and capabilities of BRAPH 2.0 is provided in the article:

*BRAPH 2.0: Brain Connectivity Analysis with Multilayer Graphs and Deep Learning* by Chang et al.

Below, we present a concise introductory tutorial that takes you through the essential steps of installing BRAPH-2, loading your data, running standard analyses, and interpreting your results.

## *Installation and Setup*

Requirements:

- MATLAB (R2022a or later).
- Git (optional) for cloning the repository.

Downloading BRAPH-2:

1. Visit the GitHub repository:

<https://github.com/braph-software/BRAPH-2>

2. Download the ZIP or clone it via Git.
3. Extract (or clone) the files into a folder of your choice.

## *Using the Graphical User Interface (GUI)*

BRAPH-2 provides an intuitive GUI to perform analyses with minimal or no coding.

## *Launching BRAPH-2*

1. In MATLAB, navigate to the main BRAPH-2 folder.
2. Type:

```
braph2
```

3. The main BRAPH-2 window will pop up, displaying different pipeline categories (e.g., structural, functional, connectivity, deep learning, etc.).

### *Pipeline Steps*

Each pipeline consists of a series of sequential steps to guide your analysis. For example, a typical graph analysis pipeline might include the following steps:

1. **Load Atlas:** Import or select a brain atlas (ROI definitions).
2. **Load Group(s):** Load subjects' data (structural, functional, or connectivity matrices).
3. **Analyze:** Build graphs (optionally threshold them) and compute graph measures.
4. **Compare (optional):** If you have multiple groups, run permutation tests or other statistical comparisons.
5. **Visualize and Export:** Review measures, visualize results (e.g., brain views, adjacency matrices), and export them.

### *Basic Tutorial Example*

#### *Single-Layer Connectivity Analysis*

1. **Run the Connectivity Pipeline:**

In the BRAPH-2 main window, pick a pipeline under the “Connectivity” tab (e.g., *Pipeline Comparison Connectivity WU*).

2. **Load Atlas:**

Choose your brain atlas file (e.g., a .xls, .txt, or provided template).

3. **Load Groups:**

Import each group of subjects (e.g., controls, patients). Each subject should have a connectivity matrix (WU, BUD, or BUT formats).

4. **Run the Analysis:**

Select graph measures (e.g., degree, clustering).

Click Analyze to compute measure values per subject and per group.

5. **Group Comparison:**

Optionally set up statistical comparisons (e.g., 1000 permutations with FDR correction).

Click Compare to execute.

## 6. Visualize & Export:

Examine adjacency matrices, 3D brain plots, bar charts, etc.

Save the pipeline (\*.b2), results, and figures for reproducibility.

## *Multilayer and Deep Learning Pipelines*

BRAPH-2 also offers:

- **Multilayer Graph Analysis:** Combine multiple modalities (e.g., structural and functional) into a *multiplex* or *multilayer* graph.
- **Deep Learning Classification/Regression Pipelines:** Use adjacency matrices or graph measures as input to train neural networks and perform classification (e.g., patient vs. control) or regression (e.g., predict age).

These pipelines follow similar steps but involve constructing a *supra-adjacency* matrix (multilayer) or generating input features for neural networks.

## *Automating Analyses with Scripts*

If you prefer or need batch processing:

1. Open the examples folder in BRAPH-2 to see sample scripts.
2. Copy and adapt them for your data.
3. Run in MATLAB to process large datasets or integrate with HPC clusters.

## *Extending BRAPH-2*

Thanks to its object-oriented architecture:

- You can create new pipelines by extending existing ones.
- Add new metrics by subclassing the Measure class.
- Add new neural network architectures by extending the NNBase class.

All changes can be compiled using the command:

```
braph2genesis
```

## *Further Information*

BRAPH-2 provides a unified environment to explore brain connectivity using modern graph-theoretical methods, multilayer modeling, and deep learning. With a simple GUI for non-programmers and rich scripting/development options for advanced users, it accommodates a wide range of neuroscience research needs.

For additional learning resources and reference materials, please see the following.

### *BRAPH-2 GitHub and Discussion Forum*

BRAPH-2 GitHub:

<https://github.com/braph-software/BRAPH-2>

Discussion forum:

<https://github.com/braph-software/BRAPH-2/discussions>

### *User Tutorials for Pipelines*

<https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/pipelines>

Explore ready-made pipelines for structural, functional, connectivity, and deep learning analyses.

Learn how to load brain atlases, import data, compute graph measures, compare groups, and visualize/export your results.

### *Developer Tutorials for Extending BRAPH-2*

<https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers>

**Adapting a Pipeline Script** [https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev\\_pipeline](https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev_pipeline): Learn how to modify an existing pipeline, customize parameters (e.g., default number of permutations), or add new pipeline steps.

**Adapt an Example Script** [https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev\\_script](https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev_script): See how to turn a GUI-based analysis into an automated script for batch processing or cluster/HPC applications.

**Implement, Import, and Export Groups of Subjects** <https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/>

**dev\_subject:** Discover how to introduce new Subject types (e.g., new data modalities), handle group-level imports, and export results.

**Implement a New Ensemble Analysis** [https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev\\_analysis\\_ensemble](https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev_analysis_ensemble): Create advanced ensemble analyses in which a measure is calculated for each subject, enabling subject-level comparisons.

**Implement a New Group Analysis** [https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev\\_analysis\\_group](https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev_analysis_group): Create analyses where measures are computed at the group level (useful for structural pipelines and group-average connectivity).

Implementing new graphs and measures:

**Implement a New Graph** [https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev\\_graph](https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev_graph): Learn the steps to create and register novel graph types (e.g., new adjacency representations, directed/undirected, multiplex).

**Implement a New Measure** [https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev\\_measure](https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev_measure): Extend BRAPH-2 with original graph-theoretical metrics (e.g., community detection, centrality, resilience) by defining measure properties and calculations.

Deep learning tutorials:

**Implement a New Neural Network Classifier** [https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev\\_neural\\_networks/dev\\_nn\\_classifier](https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev_neural_networks/dev_nn_classifier): Build custom classification architectures, integrating them directly into the BRAPH-2 pipelines.

**Implement a New Neural Network Regressor** [https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev\\_neural\\_networks/dev\\_nn\\_regressor](https://github.com/braph-software/BRAPH-2/tree/develop/tutorials/developers/dev_neural_networks/dev_nn_regressor): Add regression pipelines for tasks such as age prediction or continuous clinical measures.