# Run Instructions

This guide describes how to run the full processing pipeline for analyzing intra-brain connectivity in parent–infant dyads using fNIRS hyperscanning data and graph-theoretical methods. Each step below corresponds to a specific script in the project.

## Prerequisites

* Python 3.8 or higher
* OS: Windows 10 / Ubuntu 22.04 (tested)
* RAM: ≥ 8GB recommended
* CPU: Multi-core recommended

## Install all Python dependencies:

* pip install -r requirements.txt

Or manually:

* pip install numpy pandas scipy matplotlib seaborn plotly networkx jupyter

## This optional script automates the entire pipeline. Instead of running each step manually, you can run:

* python run\_all.py
* This script will:

1. Convert .mat files
2. Normalize CSVs
3. Compute correlations
4. Extract metrics
5. Run all comparison modules
6. Generate results

To use it, place it in the root directory and ensure folder structure is consistent.

## Input Format

Place all .mat files in a folder (e.g., input\_data/), with the following naming convention:

DyadID\_Role\_Condition.mat

Examples:

* D01\_baby\_free.mat
* D01\_parent\_elicit.mat

## Step-by-Step Execution

1. Convert .mat to .csv

python Scripts/convert\_mat\_to\_csv.py

Input: .mat files (raw fNIRS recordings)

Output: .csv files named by dyad/role/condition

1. Clean and Normalize CSV files

python Scripts/clean\_and\_normalize\_csv.py  
Normalizes all 18 channels using z-score

Ensures consistent format and shape for all files

1. Compute Correlation Matrices

python Scripts/intra\_brain\_connectivity.py

* Computes 18×18 Pearson correlation matrix per file
* Applies thresholds: |r| ≥ 0.3 and p ≤ 0.05
* Saves correlation matrices as .csv

1. Construct Graphs and Extract Metrics  
   python Scripts/extract\_intra\_measures.py
   * Builds binary adjacency graphs
   * Calculates:
   * Global: Mean Degree, Efficiency, Modularity, Clustering, Small-Worldness
   * Local: Node Strength
   * Outputs: one CSV per participant
2. Compare Global Metrics by Condition

python analysis/compare\_conditions.py

* + Aggregates and plots global metrics across conditions
  + Output: CSV summaries + barplots in visualizations/

1. Analyze Dyadic Symmetry

python analysis/compare\_dyadic\_symmetry.py

* Compares parent vs. baby for each dyad
* Computes average absolute difference per metric
* Output: plot + summary CSV

1. Metric Correlation Analysis

* Computes Pearson correlation between graph metrics
* Outputs: correlation matrix CSV + heatmap PNG

1. Role-Based Metric Comparison

python analysis/compare\_roles\_by\_dyad.py

* Compares which role (baby or parent) had higher value per metric
* Output: bar chart + role summary table

1. Node-Level Strength Comparison
2. Baby vs. Parent (Average per node):

python local\_analysis/compare\_strength\_dyadic\_difference.py

1. Strength by Condition

python local\_analysis/compare\_strength\_by\_condition.py

Outputs:

* CSV files per condition
* Bar plots per node and per role

## Interactive Dashboard (HTML)

* After running the pipeline:
* All plots (PNG) are saved to visualizations/
* All summary tables (CSV) are saved to tables/

To explore your results interactively:

1. Open index.html in a browser
2. Use the left dropdown to select analysis section (e.g., Local Strength by Condition)
3. Use the right dropdown to select a specific graph
4. Below each graph, you’ll see:
   * + - A detailed description of the plot
       - An auto-generated conclusion

## Note:

A detailed User Guide and Maintenance Guide are also included in the full project document (Capstone Book) for further reference.