

# *Calculating Different Measures*

*The BRAPH 2 Developers*

*September 24, 2023*

This is the developer tutorial for adapting a script for calculating different graph measures. In this tutorial, we will explain how to edit an example script to calculate different graph measures in braph2genesis. Here, we use. EXAMPLE\_ST\_MP\_WU as an example to show how to edit this script to calculate different graph measures for structural data using a multiplex weighted undirected graph.

## *Contents*

<i>Atlas Loading</i>	2
<i>Data Loading</i>	2
<i>Group Analysis</i>	3
<i>Measure Calculation</i>	3
<i>Group Comparison</i>	4

## Atlas Loading

We started with implementing the brain atlas. Here we use the function `ImporterBrainAtlasXLS` as an example. The function allows users to import the atlas from an XLS/XLSX file.

**Code 1: Brain Atlas Importer.** The header section of the pipeline `EXAMPLE_ST_MP_WU`. It defines the importer for Atlas loading.

---

```

1 %% iheader!
2 % EXAMPLE_ST_MP_WU
3 % Script example pipeline ST MP WU
4
5 clear variables %#ok<*NASGU>
6
7 %% Load BrainAtlas
8 im_ba = ImporterBrainAtlasXLS( ... ①
9     'FILE', [fileparts(which('example_ST_MP_WU')) filesep 'Example data
10     ST_MP_XLS' filesep 'atlas.xlsx'], ...
11     'WAITBAR', true ...
12 );
13 ba = im_ba.get('BA');
```

---

① First, the script loads the brain atlas from the excel file `atlas.xlsx`

## Data Loading

This step is to load the data of two groups. In this section, we use `ImporterGroupSubjectST_MP_XLS` as an example. Users can define the path to the group data in XLS/XLSX files contained in a folder.

**Code 2: Group Subject Data Importer.** The group data importer section provides the code for loading group subject data.

---

```

1 im_gr1 = ImporterGroupSubjectST_MP_XLS( ... ①
2     'DIRECTORY', [fileparts(which('SubjectST_MP')) filesep 'Example data
3     ST_MP_XLS' filesep 'ST_MP_Group_1_XLS'], ...
4     'BA', ba, ...
5     'WAITBAR', true ...
6 );
7 gr1 = im_gr1.get('GR');
8
9 im_gr2 = ImporterGroupSubjectST_MP_XLS( ... ②
10    'DIRECTORY', [fileparts(which('SubjectST_MP')) filesep 'Example data
11    ST_MP_XLS' filesep 'ST_MP_Group_2_XLS'], ...
12    'BA', ba, ...
13    'WAITBAR', true ...
14 );
15 gr2 = im_gr2.get('GR');
```

---

① `ImporterGroupSubjectST_MP_XLS` imports a group of subjects with structural data and their covariates (optional) from another XLS/XLSX file.

② imports a second group of subjects with structural data and their covariates (optional) from another XLS/XLSX file.

## Group Analysis

This step is to initialize the group data analysis. Users are allowed to specify the correlation type and `NEGATIVE_WEIGHT_RULE` in this section, the script will perform the group analysis automatically for each group loaded from previous step.

**Code 3: Group Subject Data Analysis.** The group data analysis provides code for initialization of group data analysis.

---

```

1 a_WU1 = AnalyzeGroup_ST_MP_WU( ... ①
2     'GR', gr1, ...
3     'CORRELATION_RULE', Correlation.PEARSON ...
4 );
5
6 a_WU2 = AnalyzeGroup_ST_MP_WU( ... ①
7     'TEMPLATE', a_WU1, ...
8     'GR', gr2 ...
9 );

```

---

① creation of the group analysis with the loaded atlas and groups data.

## Measure Calculation

This step is to calculate graph measures with data loaded in the previous step. Here we use overlapping strength as an example.

**Code 4: Group Subject Data Analysis.** The group data analysis provides code for initialization of group data analysis.

---

```

1 g_WU1 = a_WU1.memorize('G'); ①
2 ovstrength_WU2 = g_WU2.get('MEASURE', 'OverlappingS').get('M'); ②
3 ovstrength_av_WU2 = g_WU2.get('MEASURE', 'OverlappingSAv').get('M'); ③
4
5 g_WU2 = a_WU2.get('G');
6 ovstrength_WU2 = g_WU2.get('MEASURE', 'OverlappingS').get('M'); ②
7 ovstrength_av_WU2 = g_WU2.get('MEASURE', 'OverlappingSAv').get('M'); ③

```

---

① memorize in case there are measures with non-default rules.

② The function calculates graph measure. In this case, overlapping strength is calculated.

③ Other measures can also be calculated by changing the measure name. For example, `OverlappingS` calculates overlapping strength and `OverlappingSAv` calculates overlapping strength average

## Group Comparison

The last step is to perform group comparison. CompareGroup contains the results of a group-based comparison for a given measure. Specifically, it contains the one-tailed and two-tailed p-values and the 95% confidence interval.

**Code 5: Group Subject Data Analysis.** The group data analysis provides code for initialization of group data analysis.

---

```

1 % comparison
2 c_WU = CompareGroup( ... ①
3     'P', 10, ...
4     'A1', a_WU1, ...
5     'A2', a_WU2, ...
6     'WAITBAR', true, ...
7     'VERBOSE', false, ...
8     'MEMORIZE', true ...
9 );
10
11 ovstrength_WU_diff = c_WU.get('COMPARISON', 'OverlappingS').get('DIFF'); ②
12 ovstrength_WU_p1 = c_WU.get('COMPARISON', 'OverlappingS').get('P1'); ③
13 ovstrength_WU_p2 = c_WU.get('COMPARISON', 'OverlappingS').get('P2'); ③
14 ovstrength_WU_cil = c_WU.get('COMPARISON', 'OverlappingS').get('CIL'); ④
15 ovstrength_WU_ciu = c_WU.get('COMPARISON', 'OverlappingS').get('CIU'); ⑤

```

---

① Creation of group comparison based on previous group analysis.

② DIFF calculates the difference of group comparison.

③ P1 calculates the one-tailed p-value.

③ P2 calculates the two-tailed p-value.

④ CIL calculates the lower value of the 95% confidence interval.

⑤ CIU calculates the upper value of the 95% confidence interval.