Manual Scripts

Written during Research Project I
Nora Hagopian

Department of Integrative Neurophysiology
Vrije Universiteit Amsterdam

Section I

The first section of this manual contains scripts that were written in the starting phases of this project. These scripts are the starting blocks of the project, which act as a foundation for the scripts written to compute the final results. You can find these final scripts in the second section of the manual.

prepareData.m

Prepares the data from the available EEG files for Brainstorm and further scripts.

% get Headmodel using **getHeadmodel.m**

% get Channels with file path. Prepares for: **getChannels.m**

% get Atlas using **getAtlas.m** & source-space timeseries downsampled to Atlas regions

% "scalpsignal" : timeseries of every channel in scalp-space

getHeadmodel.m

This function returns a struct of the headmodel file and variable with the location of 15.000 vertices, given the path to the headmodel file.
—> Used in: ***prepareData.m***

% "path" : path to the headmodel file [input]

% "headmodelstruc" : struct with information regarding the headmodel [output]

% "vertices" : 15.000 vertices with X, Y, Z coordinates [output]

```
function [headmodelstruc, vertices] = getHeadmodel(path)
```

getChannels.m

This function returns a struct of all channels and variables containing fields of channels of interest.
—> Used in: ***DataAnalyses.m, ValidateSourceReconstruct.m, regionCorrIncrease.m***

% Channels in channel file .loc contain the coordinates of every electrode channel

% "allchannels" : channels with coordinates in .Loc [input]

% "channelnames" : channels of interest stated by name in cell [input]

% "channelsofinterest" : struct of channels of interest with channel names, numbers and X, Y, Z location in separate struct fields [output]

```
function [channelsofinterest] = getChannels(allchannels, channelnames)
```

getAtlas.m

This function returns 3 variables: two variables containing the vertices and seeds of the atlas and one variable containing the source-reconstructed timeseries spatially downsampled to the atlas regions.
—> Used in: ***prepareData.m***, ***getConnectivitiesScalpRegion.m***

% “path” : path with atlas output (acquired as atlas file from Brainstorm) [input]

% “atlasstruct” : structure with information regarding the atlas given by “path”

% “scouts” : regions with .vertices (in the region) and .seed (midpoint of the region) [output]

% “seeds” : vertices that are the middle points of all the regions [output]

% “sourcesignal” : source-space timeseries per region (averaged vertices) [output]

```
function [sourcesignal, scouts, seeds] = getAtlas(path)
```

DataAnalyses.m

Main script for the initial data analyses through calling other scripts to compute and plot the results.

% run preparations before following with this script: **prepareData.m**

% use **computedistance.m** to compute the Euclidian distance of a chosen electrode to the seeds of all the atlas regions.

% plot channel signals next to and overlying the (averaged) closest region signal with: **signalplot.m** type **1**
[uses: getChannels.m, computedistance.m]

% plot from 1 channel the Z-scored signal overlying the (averaged) closest region signal including Correlation between the signals, with: **signalplot.m** type **2**
[uses: getChannels.m, computedistance.m]

% plot from channel and second neighbors the Z-scored signal overlapping the (averaged) closest region signal including Correlation between the signals, with: **signalplot.m** type **3**
[uses: getChannels.m, computedistance.m]

% plot from 1 channel the Z-scored signal overlapping the (averaged) closest, second and third region signals including Correlation between them, with: **signalplot.m** type **4**
[uses: getChannels.m, computedistance.m]

% to inspect the influence of Volume Conduction, plot timeseries of different neighboring and further away channels and regions & overlay different regions, with: **signalplot.m** type **5**
[uses: getChannels.m, computedistance.m]

% plot power-spectrum of sensor-signals from all channels, with: **timefreqAnalyses.m***

% plot power-spectrum of sensor-signals from all regions, with: **timefreqAnalyses.m***

% plot timeseries, amplitude envelopes and power-spectra of one channel with it's closest region in one Figure **
[uses: getChannels.m, computedistance.m]

* **timefreqAnalyses.m** was written in an early stage and does not deliver a desired outcome. Later in the project, the **powerspectMethod.m** script was applied to compute correct power-spectra. For this reason, **timefreqAnalyses.m** will not be further explained in this manual.
** The method used here to compute the power-spectra is outdated, please look at **powerspectMethod.m** (in Section II of this manual) for a correct version to apply in a similar manner.

computedistance.m

This function returns the Euclidian distance computed between chosen electrodes and every Region seed. Returns structure containing closest atlas region to channels of interest.
—> *used in:* ***DataAnalyses.m****,* ***ValidateSourceReconstruct.m****,* ***regionCorrIncrease.m***

% Euclidian Distance (d) between p and q: d(p,q) = sqrt((qx - px)^2 + (qy - py)^2 + (qz - pz)^2)
% For all channels of interest compute Euclidian distance to seed of all regions

% "channelsofinterest" : output of **getChannels.m** function (see in Manual) [input]

% "seeds" : output of **getAtlas.m** function (see in Manual) [input]

% "scouts" : output of **getAtlas.m** function (see in Manual) [input]

% "vertices" : output of **getHeadmodel.m** function (see in Manual) [input]

% "extended" : is an alternative input, where if extended is "1", we also compute 2nd and 3d closest distance [input]

% "closestRegion" : structure containing selected channel names and numbers with additional field of closest (or with 2nd and 3rd closest) region per channel [output]

```
function [closestRegion] = computedistance(channelsofinterest, seeds, scouts, vertices, extended)
```

signalplot.m

Plots displaying the results of 6 different data analyses, see ***DataAnalyses.m*** *for further specification on the analyses*
—> *used in:* ***DataAnalyses.m***

% "scalpsignal" : timeseries of every channel in scalp-space (computed in **prepareData.m**) [input]

% "sourcesignal" : output of **getAtlas.m** function (see in Manual) [input]

% "timeaxes" : variabele containing time-points for the plots concerning the data [input]

% "closestRegion" : output of **computedistance.m** function (see in Manual) [input]

% "type" : used to indicate which of the 6 different analyses are wished to execute [input]

% "fftScalp", "psdScalp", "fftSource", "psdSource": variables used to compute power-spectra, only used in type 6 [input]

```
function [] = signalplot(scalpsignal, sourcesignal, timeaxes, closestRegion, type, fftScalp, psdScalp, fftSource, psdSource)
```

Section II

The second section of this manual contains specifications on the scripts written towards the end of the project. These scripts were used to run the final analyses and obtain the final results. The main outline of the scripts are based on the produced figures for the paper.

ValidateSourceReconstruct.m

Figure 3 (in thesis): Source-reconstruction control. Show intuitive validation of signal after source-reconstruction by plots.
—> uses: ***getChannels.m, computedistance.m,*** *sh_convertToEEG.m (script provided by S. Houtman), sh_computePowerI.m (by S. Houtman)*

% A) Plot typical source-region signal
% "channelnames" : give name(s) of channel for the plot, input needs to be cell

% B) Plot typical power-spectrum of source-region
% "SignalSource" : take closest source-region of "channelnames" for power spectrum plot
% convert source signal to EEGLAB format
% get power spectrum
% plot power spectrum of typical source signal

% C) Plot amplitude envelopes of channel with closest source-region
% "channelnames" : define channels of interest by name, input needs to be cell
% get signals z-scored
% calculate Amplitude envelopes with absolute Hilbert transform
% plot
% calculate correlation
% plot the rho on the graph

% D) Plot power spectra of channel with closest source-region
% "SignalSource" : take z-scored closest source-region for power spectrum plot
% convert source signal to EEGLAB format
% get power spectrum
% "SignalScalp" : take z-scored scalp signal for power spectrum plot
% convert scalp signal to EEGLAB format
% plot power spectrum of scalp and source signal
% calculate correlation
% plot the rho on the graph

getConnectivitiesScalpRegion.m

Figure 4 (in thesis): Signal unmixing from scalp- to source space shown by Histogram plot.
—> uses: sh_rdir.m (by S. Houtman), ***getAtlas.m, computeWPLI.m***

% Get scalp- and region-signals of all subjects
% index all recordings in the folder with clean files
% get Atlas using getAtlas.m & source-space timeseries downsampled to Atlas regions

% Compute mean all-to-all electrode connectivity
% "meanconnectivitiesScalp" : the averaged all-to-all inter-channel connectivity, measured with the weighted phase lag index (wPLI), over all subject (n = 118)

% Compute mean all-to-all region connectivity
% “meanconnectivitiesRegion” : the averaged average wPLI between all-to-all source-region signals over all subjects (n = 118)

% Plot histograms of connectivities
% the averaged all-to-all inter-channel connectivity, measured with wPLI, over all subject (n = 118)
% The averaged average wPLI between all-to-all source-region signals over all subjects (n = 118)

% Get average of the connectivity distribution
% indicates inter-channel and inter-region functional connectivity

computeWPLI.m

Compute functional connectivity by weighted phase lag index (wPLI) method.
—> *used in:* ***regionCorrIncrease.m**, **getVertexConnect.m***

% “Signal” : (multiple) scalp OR source signals [input]

% “nChannels” : get number of channels

% “phaseSignal” : compute phase of the signal (Hilbert)

% “wpli” : compute weighted phase lag index (matrix) [output]

```
function wpli = computeWPLI(Signal)
```

regionCorrIncrease.m

Figure 5 (in thesis): Stretched source-regions uncorrelated with nearby channels. Plot three Regions with an incresing size and their closest channels.
—> *uses:* ***getChannels.m, computedistance.m***

% “channelnames” : define channels of interest by name, input needs to be a cell
% “channelsofinterest” : structure containing the channels defined in “channelnames"
% “closestRegion” : structure containing closest atlas region to channels of interest

% Z-score the chosen scalp signals and their closest region signals

% Plot timeseries of a SMALL region with its closest channel
% calculate correlation
% plot the rho on the graph

% Plot timeseries of a MEDIUM region with its closest channel
% calculate correlation
% plot the rho on the graph

% Plot timeseries of a LARGE region with its closest channel
% calculate correlation
% plot the rho on the graph

getVertexConnect.m

Figure 6 (in thesis): Stretched source-regions are an incorrect representation of neuronal activity. Get the inter-vertex connectivity of a region extracted from two atlases for comparison.
—> *uses:* ***computeWPLI.m***

% Inter-vertex connectivity of region from the 68-regions atlas
% "**scout**" : the file (extracted from Brainstorm) containing the vertices (and seed) of the region of interest from the 68-regions atlas *[open this file in MATLAB with the name "scout"]*
% "region" : vertices (and seed) of the region of interest from the 68-regions atlas
% "vertices" : vertices from "region"
% "**SourceData**" : the file with all the timeseries of all vertices of the chosen subject *[open this file in MATLAB with the name "SourceData"]*
% "sources" : the timeseries of all the vertices
% "connectMat" : connectivity matrix (intra-region connectivity by computing the inter-vertex wPLI)
% "meanConnect" : mean of the connectivity matrices

% Inter-vertex connectivity of region from the 136-region atlas
% "**scout2**" : the file (extracted from Brainstorm) containing the vertices (and seed) of the region of interest from the 136-regions atlas *[open this file in MATLAB with the name "scout2"]*
% "region2" : vertices (and seed) of the region of interest from the 136-regions atlas
% "vertRegion1", "vertRegion2", "vertRegion3", "vertRegion4" : the vertices from the divided regions that correspond with the region (of interest) extracted from 68-region atlas
% "sources1", "source2", "sources3", "sources4" : the timeseries of all the vertices per sub-region
% "cMat1", "cMat2", "cMat3", "cMat4" : connectivity matrixces (inter-vertex wPLI) per sub-region
% "maenMat1", "maenMat2", "maenMat3", "maenMat4" : mean of the connectivity matrices
% "meanMatAll" : merges the separate sub-region mean connectivities

% Plot Histograms
% plot Histogram of inter-vertex connectivity of the region extracted from 68-regions atlas
% plot Histogram of inter-vertex connectivity of the region extracted from 136-regions atlas

sourceReconstruct.m *

Compute Source-reconstruction to get timeseries of all sources (without Brainstorm)

powerspectMethod.m **

Compute power-spectrum of signal(s)

***** It is optional to use **sourceReconstruct.m**, but the sources can also be reconstructed in Brainstorm.
****** It is optional to use **powerspectMethod.m**, but it might not work perfectly. The other scripts that compute power-spectra, include the correct code for this computation.