EEGLAB SPoC plug-in tutorial

What is the SPoC plug-in for EEGLAB?

The SPoC plug-in is a set of Matlab tools developed by the IDA group of the TU of Berlin, that allows the decomposition of the EEG data using the SSD, SPoC and cSPoC methods. These tools are designed to work within the EEGLAB environment, providing a GUI to decompose the data into different relevant components:

1. SSD - Extracts components with optimized signal-to-noise ratio at a frequency band of interest.
2. SPoC - Extracts components with maximal power covariance with the univariate target function z.
3. cSPoC - Extracts components with maximal envelope correlations from N oscillatory and multivariate datasets. For N>2, the extracted components maximize the pairwise averaged envelope correlations.

All of the tools can also be used from the Matlab command line, providing expert users with the ability to use them in custom scripts.

Requirements

In addition to the requirements of EEGLAB, SPoC plug-in requires the following folders to be stored in the plug-in folder: **SSD ; SPoC ; cSPoC ; utils**. These folders are found in the following repository:

https://github.com/svendaehne/matlab_SPoC.

Download and Installation

1. Go to:

   https://github.com/svendaehne/eeglab_plugins

   push the "Download ZIP" botton and choose where to save the file.

2. Uncompress the downloaded file into the 'plugins' directory of your EEGLAB distribution. You should now have a directory called 'SPoC' containing necessary files.

3. Obtain the following folders from the SPoC repository

   (https://github.com/svendaehne/matlab_SPoC)

   and store them in the 'SPoC' directory: **SSD, SPoC, cSPoC, utils**.

Starting EEGLAB should now automatically recognize and add the plug-in. You should see the following line appear in your Matlab environment window:

EEGLAB: adding "SPoC" v.1.0 (see >> help eegplugin_AD)

Voilà!

Tutorial

Using components

After running each of these decompositions your data will be assigned with components. These components are saved in the ICA components slot (EEG.icaweights).

Read the ICA tutorial

(*http://sccn.ucsd.edu/wiki/Chapter_09:_Decomposing_Data_Using_ICA*)

to learn about the following actions which can be performed using AD components, in the exact same manner:

1. Plotting scalp maps

   (*http://sccn ... #Plotting_2-D_Component_Scalp_Maps*)

2. Plotting components activation

   (*http://sccn ... #Scrolling_through_component_activations*)

3. Studying and removing components

   (*http://sccn ... #Studying_and_removing_ICA_components*)

SSD

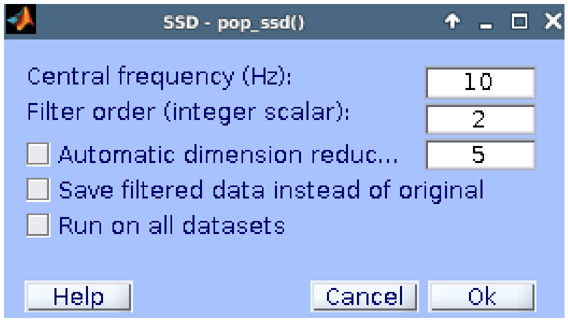
The SSD function extracts components with optimized signal-to-noise ratio at a frequency band of interest.

Preparation

1. Run EEGLAB and load the relevant dataset/s. Data can be either epochs or continuous EEG.

Run

1. Select **Tools** > **SPoC** > **Run SSD**. This calls the function pop_ssd.m.



2. Fill out the parameters form and press **OK**. Here is the parameters description (top down, left to right):

   (a) Central frequency for the filtering processes. The calculation of the cut-off frequencies is as follows:

      i. signal band-pass= $2^{(log2(fc)-0.25)}; 2^{(log2(fc)+0.25)}$.

      ii. noise band-past= $2^{(log2(fc)-0.9)}; 2^{(log2(fc)+0.9)}$.

      iii. noise band-stop= $2^{(log2(fc)-0.35)}; 2^{(log2(fc)+0.35)}$.

      'fc' being the chosen central frequency.

   (b) Filter order used for butterworth bandpass and bandstop filtering. If unsure about it, use the default value of order 2.

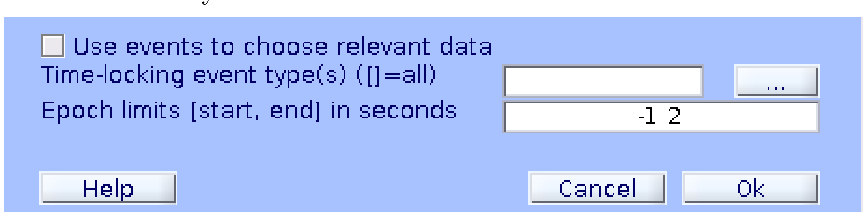
   (c) Automatically subtract components from data (Yes/No).

   (d) Numbers of component to keep (the rest will be subtracted). Only relevant if you marked the checkbox in (c).

   (e) Overwrite the original data with the data filtered around the central frequency given at (a) (Yes/No).

   (f) Run the same calculation on all loaded datasets (Yes/No).

   In case the data is continues and contains event marks, you could also choose to use only event related data:

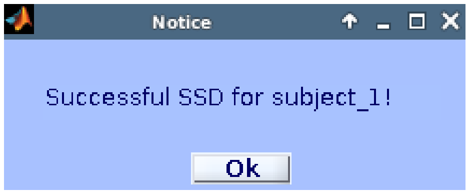


   (a) Mark to use this option.

   (b) The name of the event/s you wish to use.

(c) The limits around the events. Data outside these limits will not be taken into account for the calculation.

3. Wait until the following massage (With your dataset name) appears and press **OK**

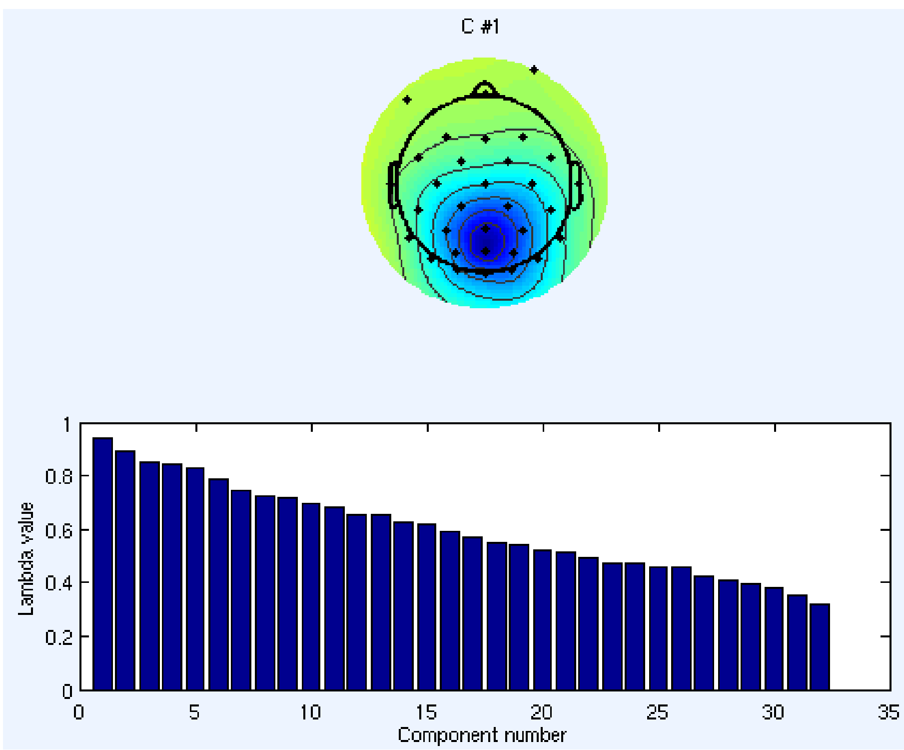


The components spatial filters are now stored under EEG.icaweights while old weights from pervious decompositions are stored under EEG.etc.oldweights.

In the structure EEG.dipfit.model you can find the lambda values.

Plot

For SSD and SPoC you can plot the lambda values of the decomposition. Select **Plot** > **SPoC** > **Lambda spectrum**.



clicking on a blue bar shows the scalp plot of the relevant component.

SPoC

The SPoC function extracts components with maximal power covariance with the univariate target function z.

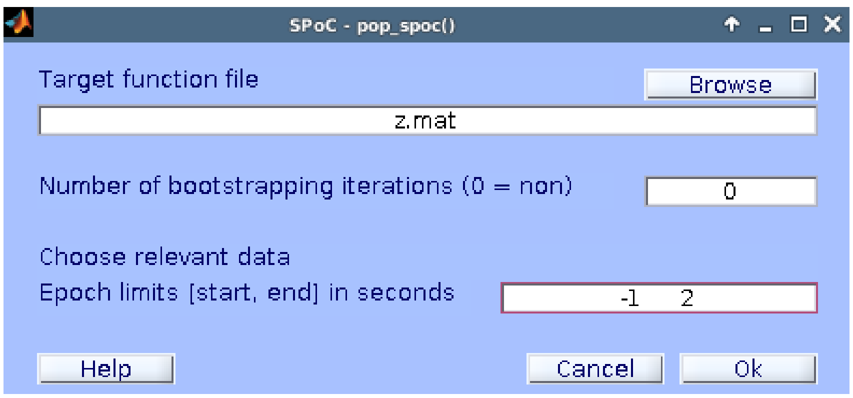
Preparation

To get the best results we recommend applying SPoC analysis only after filtering and reducing data dimension using SSD. To do both, select **Tools** > **SPoC** > **Run SSD**, mark the "**Dimension reduction**" and "**Save filtered data instead of original**" checkboxes and choose the relevant central frequency and the number of components you wish to keep (=dimension degree).

1. Save the relevant target function z as a vector in a .mat file.
2. Run EEGLAB and load the relevant dataset. Data can be either epochs or continuous EEG. In the continues case data will be segmented based on its events. Make sure that the number of epochs or events is equal to the length of your target function z vector.

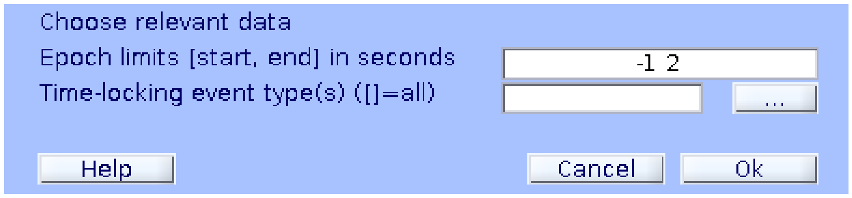
Run

1. Select **Tools** > **SPoC** > **Run SPoC**. This calls the function pop_spoc.m.



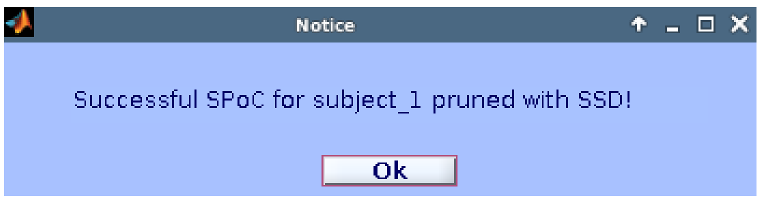
2. Fill out the parameters form and press **OK**. Here is the parameters description (top down, left to right):
   (a) Target function z file path. You could use the browse option to find the file in your folders.
   (b) Number of bootstrapping iterations used to calculate p-values of the results.
   (c) The relevant data limits for the calculation. Data outside these limits will not be taken into account for the calculation.

In case the data is continues, you could still use SPoC based on event marks:



   (a) The limits around the events. Data outside these limits will not be taken into account for the calculation.
   (b) The name of the event/s you wish to use.

3. Wait until the following massage (With your dataset name) appears and press **OK**



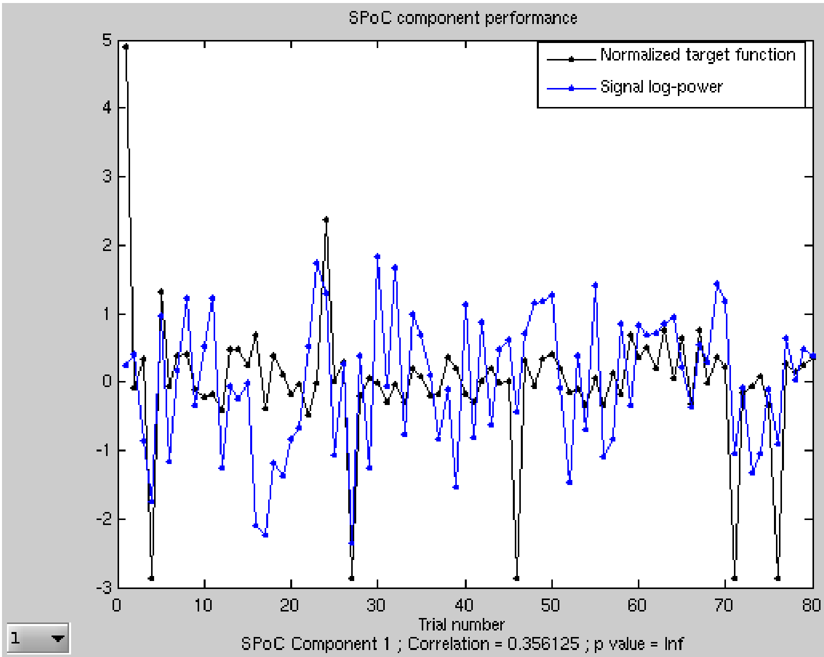
The components spatial filters are now stored under EEG.icaweights while old weights from pervious decompositions are stored under EEG.etc.oldweights.

In the structure EEG.dipfit.model you can find the lambda values, p values and the SPoC signal function of each component.

Plot

See the plot section of SSD for lambda spectrum plot.

Select **Plot** > **SPoC** > **SPoC results**.



Choose any component from the drop list to show its power function next to the target function.

cSPoC

The cSPoC function extracts components with maximal envelope correlations from N oscillatory and multivariate datasets. For N>2, the extracted components maximize the pairwise averaged envelope correlations.

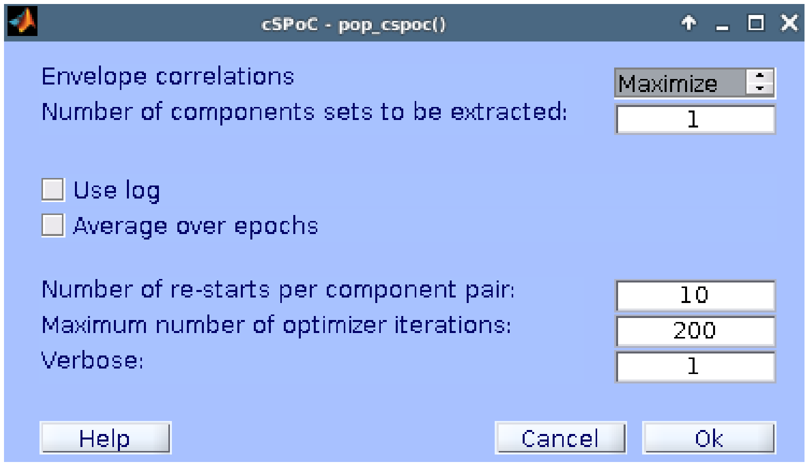
Preparation

To get the best results we recommend applying cSPoC analysis only after filtering and reducing data dimension using SSD. To do both, select **Tools > SPoC > Run SSD**, mark the "**Dimension reduction**", "**Save filtered data instead of original**" and "**Run on all datasets**" checkboxes and choose the relevant central frequency and the number of components you wish to keep (=dimension degree).

1. Run EEGLAB and load all relevant datasets. Any loaded dataset will be taken into account. Make sure that all datasets are segmented to the same number of epochs.

Run

1. Select **Tools > SPoC > Run cSPoC**. This calls the function pop_cspoc.m.



2. Fill out the parameters form and press **OK**. Here is the parameters description (top down, left to right):

   (a) Maximizing or minimizing the envelopes correlation.

   (b) Number of envelope-correlated components per dataset to be extracted.

   (c) Optimize correlations of log-envelopes rather then envelopes (Yes/No).

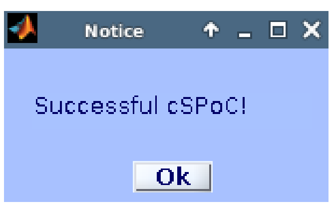
   (d) When optimizing the correlations, average the source envelopes within epochs (Yes/No).

   (e) Number of re-starts per component pair.

   (f) Maximum number of optimizer iterations.

   (g) Level of detail in command line output.

3. Wait until the following massage appears and press **OK**

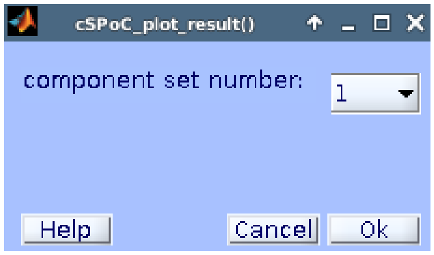


The components spatial filters are now stored under EEG.icaweights while old weights from pervious decompositions are stored under EEG.etc.oldweights.

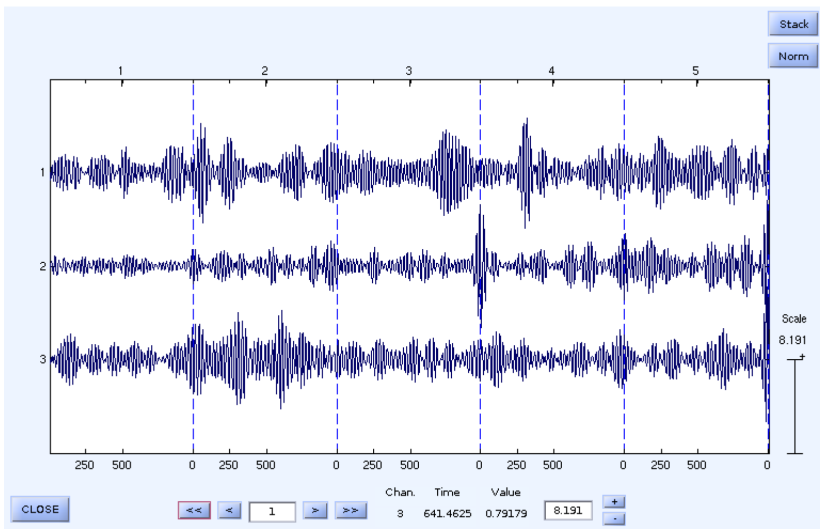
In the structure EEG.dipfit.model you can find all the correlation r values.

Plot

Select **Plot** > **SPoC** > **cSPoC results**.



In the pop-up window, select which component set you would like to plot.



We recommend increasing the displayed number of epochs using **Setting** > **Time range to display** > **Number of epoch(s)**.

References

[1] Nikulin VV, Nolte G, Curio G. A novel method for reliable and fast extraction of neuronal EEG/MEG oscillations on the basis of spatio-spectral decomposition. NeuroImage, 2011, 55: 1528-1535.

[2] S. Dähne, F. C. Meinecke, S. Haufe, J. Höhne, M. Tangermann, K. R. Müller, V. V. Nikulin, "SPoC: a novel framework for relating the amplitude of neuronal oscillations to behaviorally relevant parameters", NeuroImage, 86(0):111-122, 2014

[3] S. Dähne, V. V. Nikulin, D. Ramirez, P. J. Schreier, K. R. Müller, S. Haufe, "Finding brain oscillations with power dependencies in neuroimaging data", NeuroImage, 96:334-348, 2014

[4] S. Haufe, S. Dähne, V. V. Nikulin, "Dimensionality reduction for the analysis of brain oscillations" NeuroImage 101:583-597 2014