NEMO ERP Pattern Segmentation Tutorial:
ERP pattern extraction with microstate segmentation
Robert Frank & Gwen Frishkoff

1. OVERVIEW

This tutorial will make use of the following matlab toolbox, which was written by Robert Frank:
*NEMO_ERP_Pattern_Segmentation*. The toolbox can be downloaded from the NEMO wiki
(http://nemo.nic.uoregon.edu). An overview of the analysis methods and their matlab implementation
can be found in "NEMO_ERP_Pattern_Segmentation_Description.pdf," which is also available for
download from the NEMO wiki. Questions concerning this tutorial should be directed to Gwen
Frishkoff (gfrishkoff@gsu.edu) or Robert Frank (rmfrank@mac.com).

2. RUNNING NEMO ERP MICROSTATE ANALYSIS DEMO SCRIPT

For the purposes of this tutorial, we have provided an initialization script or "meta-script." To illustrate
how this initialization-script works, the test dataset, *Sim_ERP_Data.raw*, is provided with this tutorial
The dataset contains 150 time sample (@250Hz), 129 electrode ERPs for 2 experimental conditions and
20 subjects. See Ref [4] for a complete description of this dataset. Note that the parameter values in the
tPCAsix_Gavg_Sim_ERP_Data.m script are specific to the tPCA of the Sim_ERP_Data.raw data. By
specifying new parameter values, the tPCAsix_Gavg_Sim_ERP_Data.m initialization-script may serve
as a template when you are ready to make a new initialization-script to apply to your own data.

To run this script, simply navigate to your Working_Folder and locate the tutorial “initialization-
scripts,” which are located in the NEMO_ERP_Pattern_Segmentation > Support > Tutorial
> RunTime subfolder. To run the meta-script, simply type in the name of the script (minus '.m')
at the command line, as follows:

```matlab
>> NemoErpPatternSegmentation_Demo
```
APPENDIX

There are three basic steps in running NEMO_ERP_Pattern_Segmentation:

**Step 1**: Initialize parameters for input dataset & create class constructor

**Step 2**: Call class constructor

**Step 3**: Initialize Additional Class Properties (Set Experimental / Data Attributes)

**STEP 1. Initialize parameters for input dataset**

To generate microstate boundaries for a particular ERP observation (i.e., for a single subject or grand average and for a single experiment condition), you must specify 12 parameters (see below). MATLAB will then create a **class instance**, import the ERP data file and compute the microstates using a default configuration based upon the segment border probability algorithm. Additional class properties that characterize the experimental session may be initialized using MATLAB’s standard interface for accessing property name / property value pairs after class object instantiation as described in Step 2.

*Note: Error checking parameter values is a work in progress. Specifying parameter values that index outside the range of cells, components or data intervals will generate an error and halt program execution.*

Lab_ID = 'Simulated_Lab';
Experiment_ID = 'Simulated_Experiment';
Session_ID = 'Simulated_Session';
Electrode_Montage_ID = 'HCGSN-128';
Subject_Group_ID = 'Simulated_Adults';
Subject_ID = 'Simulated_Subject_001-020';
Subject_File_ID = 'Sim_ERP_Data.raw';
Cell Import_Index = [1, 2];
DiffCell Import_Index = [];
Subject Import_Index = (1:20);
ERP Import_Onset_Latency = [];
ERP Import_Offset_Latency = [];

**STEP 2. Call Class Constructor**

Sim_ERP_Data_Microstates = NemoErpPatternSegmentation(Lab_ID, ...
...
Experiment_ID, Session_ID, Subject_Group_ID, ...
...
Electrode_Montage_ID, Subject_ID, Subject_File_ID, ...
...
Cell Import_Index, DiffCell Import_Index, Subject Import_Index, ...
...
ERP Import Onset_Latency, ERP Import Offset_Latency);
STEP 3. Initialize Additional Class Properties (Set Experimental / Data Attributes)

set(Sim_ERP_Data_Microstates, ...
... 'Cell_Event_Type', {'Sim_Stimulus_ONSET', 'Sim_Stimulus_ONSET'}, ..., ...
... 'Cell_Label_Descriptor', {'Sim_Cond_001', 'Sim_Cond_002'}, ..., ...
... 'Cell_Event_Modality', {'Sim_Visual', 'Sim_Visual'}, ..., ...
... 'Cell_Stimulus_Type', {'Sim_Word', 'Sim_Word'}, ..., ...
... 'Experiment_Info', 'Test_Pattern_Segmentation');

Note: You can set class properties using MATLAB’s set / get interface and property_name, property_value pairs: set(class instance name, property name 001, property value 001, property name 002, property value 002, ...).

STEP 4. Compute Microstate Boundaries

The MATLAB statements for computing microstate boundaries with the NemoErpPatternSegmentation class, for each of the segmentation algorithms, are listed below. These examples assume that the class instance, Sim_ERP_Data_Microstates, has first been instantiated by calling the class constructor.

To invoke a function or algorithm that the NemoErpPatternSegmentation class implements, which is called a class method, use “dot” notation:

class_instance_name.class_method_name(required input arguments). In many cases you will not have to explicitly pass arguments when calling a class method, since the class’s properties, i.e. its data, are passed implicitly to the method by the MATLAB runtime engine.

The NemoErpPatternSegmentation class implements four alternative algorithms for computing an ERP’s microstate boundaries. Furthermore, you can specify whether the microstate boundaries are determined from the grand average ERP, (average subject-specific ERPs than extract microstate segmentation metrics) or subject-specific ERPs (extract microstate segmentation metrics from subject-specific ERPs than average).

Before invoking a specific microstate algorithm, you will need to specify three class properties that indentify the algorithm and set its operational parameters.

- **Dimension_Flag**: Specifies the dimensionality of the potential centroid coordinates as either 2D (XY coordinate pairs) or 3D (XYZ coordinate triples). Permitted values are 2 or 3. Note that this property is used only be the segment border probability and centroid dissimilarity algorithms.

- **Averaging_Protocol**: Specifies whether microstate boundaries are determined from grand average ERP data, (average subject-specific ERPs than extract microstate segmentation metrics) or subject-specific ERPs (extract microstate segmentation metrics from subject-specific ERPs than average).
than average). Permitted values are "AverageThanExtract" or "ExtractThanAverage".

- **Microstate_Algorithm**: Permitted values are @SegmentBorderProbability, @CentroidDissimilarity, @GlobalMapDissimilarity and @GlobalFieldPower.

The NemoErpPatternSegmentation class has five user-accessible methods for computing an ERP’s microstate boundaries and extracting the microstate statistics. These methods are called in the order listed below, except where indicated, only after the Dimension_Flag, Averaging_Protocol and Microstate_Algorithm properties have been set.

- **ComputePotentialCentroids**: Computes the 2D or 3D coordinate locations of each ERP topography’s positive and negative centroids, one topography per ERP sample, as a function of time. This method is needed for the SegmentBorderProbability and CentroidDissimilarity algorithms only.

- **ComputeTopographicCorrCoeff**: Computes the global field power of each ERP topography as a function of time. This global field power function is used to compute the Pearson Product-Moment Correlation Coefficient (PPMCC) between successive ERP topography pairs, which in turn determines the global map dissimilarity. The ComputeTopographicCorrCoeff method is needed for the GlobalMapDissimilarity and GlobalFieldPower algorithms only.

- **ComputeMicrostateBoundaries**: Determines the microstate boundaries of the specified ERP in accordance with the settings of the Dimension_Flag, Averaging_Protocol and Microstate_Algorithm properties.

- **PlotMBF**: Plots the algorithm-specific microstate boundary function at each ERP sample point. The microstate boundary function is used by its corresponding algorithm to predict the location of microstate borders.

- **ComputeMicrostateStatistics**: Verifies that the prospective microstates have sufficient duration and computes the microstate statistics for each microstate, subject and condition.
Segment Border Probability
To compute microstate boundaries via the Segment Border Probability algorithm and 2D centroid coordinates, type at the MATLAB command prompt:

```matlab
Sim_ERP_Data_Microstates.Dimension_Flag = 2;
Sim_ERP_Data_Microstates.Microstate_Algorithm = @SegmentBorderProbability;
Sim_ERP_Data_Microstates.ComputePotentialCentroids;
Sim_ERP_Data_Microstates.ComputeMicrostateBoundaries;
Sim_ERP_Data_Microstates.PlotMBF;
Sim_ERP_Data_Microstates.ComputeMicrostateStatistics;
```

To compute microstate boundaries via the Segment Border Probability algorithm and 3D centroid coordinates, type at the MATLAB command prompt:

```matlab
Sim_ERP_Data_Microstates.Dimension_Flag = 3;
Sim_ERP_Data_Microstates.Microstate_Algorithm = @SegmentBorderProbability;
Sim_ERP_Data_Microstates.ComputePotentialCentroids;
Sim_ERP_Data_Microstates.ComputeMicrostateBoundaries;
Sim_ERP_Data_Microstates.PlotMBF;
Sim_ERP_Data_Microstates.ComputeMicrostateStatistics;
```

Centroid Dissimilarity
To compute microstate boundaries via the Centroid Dissimilarity algorithm and 2D centroid coordinates, type at the MATLAB command prompt:

```matlab
Sim_ERP_Data_Microstates.Dimension_Flag = 2;
Sim_ERP_Data_Microstates.Microstate_Algorithm = @CentroidDissimilarity;
Sim_ERP_Data_Microstates.ComputePotentialCentroids;
Sim_ERP_Data_Microstates.ComputeMicrostateBoundaries;
Sim_ERP_Data_Microstates.PlotMBF;
Sim_ERP_Data_Microstates.ComputeMicrostateStatistics;
```

To compute microstate boundaries via the Centroid Dissimilarity algorithm and 3D centroid coordinates, type at the MATLAB command prompt:

```matlab
Sim_ERP_Data_Microstates.Dimension_Flag = 3;
Sim_ERP_Data_Microstates.Microstate_Algorithm = @CentroidDissimilarity;
Sim_ERP_Data_Microstates.ComputePotentialCentroids;
Sim_ERP_Data_Microstates.ComputeMicrostateBoundaries;
Sim_ERP_Data_Microstates.PlotMBF;
Sim_ERP_Data_Microstates.ComputeMicrostateStatistics;
```

Global Map Dissimilarity
To compute microstate boundaries via the Global Map Dissimilarity algorithm, type at the MATLAB command prompt:
Sim_ERP_Data_Microstates.Microstate_Algorithm = @GlobalMapDissimilarity;
Sim_ERP_Data_Microstates.ComputeTopographicCorrCoeff;
Sim_ERP_Data_Microstates.ComputeMicrostateBoundaries;
Sim_ERP_Data_Microstates.PlotMBF;
Sim_ERP_Data_Microstates.ComputeMicrostateStatistics;

Global Field Power
To compute microstate boundaries via the Global Field Power algorithm, type at the MATLAB command prompt:

Sim_ERP_Data_Microstates.Microstate_Algorithm = @GlobalFieldPower;
Sim_ERP_Data_Microstates.ComputeTopographicCorrCoeff;
Sim_ERP_Data_Microstates.ComputeMicrostateBoundaries;
Sim_ERP_Data_Microstates.PlotMBF;
Sim_ERP_Data_Microstates.ComputeMicrostateStatistics;

STEP 5. Querying Microstate Spatial and Temporal Properties in MATLAB.
To see the list of class properties in the MATLAB command window that you can access, type properties(class instance name) at the MATLAB command prompt:

properties(Sim_ERP_Data_Microstates)

Another way to view and interact with the object is to change your screen from the Command window to the Workspace Browser, and then double-click on the class instance. Empty properties, symbolized with [], may typically be assigned values via the standard MATLAB set / get property-name / property-value interface. For example, the properties Experiment_Info, Cell_Label_Descriptor, Cell_Event_Type, Cell_Event_Modality and Cell_Stimulus_Type are set as shown below.

set(Sim_ERP_Data_Microstates,
   'Cell_Stimulus_Type', {'Sim_Stimulus_ONSET', 'Sim_Stimulus_ONSET'}, ...
   'Cell_Label_Descriptor', {'Sim_Cond_001', 'Sim_Cond_002'}, ...
   'Cell_Event_Modality', {'Sim_Visual', 'Sim_Visual'}, ...
   'Cell_Label_Descriptor', {'Related', 'Unrelated'}, ...
   'Experiment_Info', 'Test_Pattern_Segmentation');

The class definition files for the parent NemoErpDataInfo class and NemoErpPatternSegmentation subclass, NemoErpDataInfo.m and NemoErpPatternSegmentation.m, respectively, are in their corresponding @ directories.

The methods of each class, also contained in their respective @ directories, are implemented as MATLAB function m files. One important note is that these two classes are defined as handle classes, which means that property data is passed to their methods by reference, not value, so an output argument to return data from the function workspace to the class workspace is not necessary.
The NemoErpPatternSegmentation class computes the following statistics for each microstate of the grand average (average across all subjects and conditions), subject average (average across all subjects within each condition) and subject-specific ERPs:

- Microstate latency
- Microstate duration
- Microstate duration mean and standard deviation
- Microstate PPMCC mean and standard deviation
- Microstate topography mean and standard deviation
- Microstate potential centroid coordinates mean and standard deviation
- Microstate global field power mean and standard deviation

The values of these statistics for the grand average, subject average and subject-specific ERPs are stored, respectively, in their eponymously named class properties. To see a list of the property labels, type `properties(class instance name)` at the MATLAB command prompt. The class properties containing the extracted microstate statistics are highlighted in red below.

```matlab
properties(Sim_ERP_Data_Microstates);
```

Properties for class NemoErpPatternSegmentation:
- Dimension_Flag
- Time_Zero_Offset
- Averaging_Protocol
- Microstate_Algorithm
- Microstate_Cell_ID
- Microstate_Subj_ID
- Electrode_XY_Coord
- Electrode_XYZ_Coord
- Electrode_Distance_Stats
- Global_Field_Power_Fcn
- Centroid_Dissimilarity_Fcn
- Global_Map_Dissimilarity_Fcn
- Segment_Border_Probability_Fcn
- Exploratory_Diff_CondStats
- Confirmatory_Diff_Cond_Stats
- Number_Microstate_Regions
- Microstate_Region_Indices
- Subject_Microstate_Latency
- Subject_Microstate_Duration
- Subject_Microstate_Mean_Duration
- Subject_Microstate_StdDev_Duration
- Subject_Microstate_Mean_PPMCC
- Subject_Microstate_StdDev_PPMCC
- Subject_Microstate_Mean_Topography
- Subject_Microstate_StdDev_Topography
Subject_Microstate_Mean_Centroid_Coord
Subject_Microstate_StdDev_Centroid_Coord
Subject_Microstate_Mean_Global_Field_Power
Subject_Microstate_StdDev_Global_Field_Power
Grand_Avg_Microstate_Latency
Grand_Avg_Microstate_Duration
Grand_Avg_Microstate_Mean_Duration
Grand_Avg_Microstate_StdDev_Duration
Grand_Avg_Microstate_Mean_PPMCC
Grand_Avg_Microstate_StdDev_PPMCC
Grand_Avg_Microstate_Mean_Topography
Grand_Avg_Microstate_StdDev_Topography
Grand_Avg_Microstate_Mean_Centroid_Coord
Grand_Avg_Microstate_StdDev_Centroid_Coord
Grand_Avg_Microstate_Mean_Global_Field_Power
Grand_Avg_Microstate_StdDev_Global_Field_Power
Subject_Avg_Microstate_Latency
Subject_Avg_Microstate_Duration
Subject_Avg_Microstate_Mean_Duration
Subject_Avg_Microstate_StdDev_Duration
Subject_Avg_Microstate_Mean_PPMCC
Subject_Avg_Microstate_StdDev_PPMCC
Subject_Avg_Microstate_Mean_Topography
Subject_Avg_Microstate_StdDev_Topography
Subject_Avg_Microstate_Mean_Centroid_Coord
Subject_Avg_Microstate_StdDev_Centroid_Coord
Subject_Avg_Microstate_Mean_Global_Field_Power
Subject_Avg_Microstate_StdDev_Global_Field_Power
Lab_ID
Session_ID
Experiment_ID
Subject_Group_ID
Electrode_Montage_ID
Subject_ID
Subject_File_ID
Cell_Import_Index
Segment_Import_Index
DiffCell_Import_Index
ERP_Onset_Import_Latency
ERP_Offset_Import_Latency
Cell_Event_Type
Cell_Stimulus_Type
Cell_Event_Modality
Cell_Label_Descriptor

The class properties store their microstate statistics data as MATLAB cell arrays of cell arrays
(nested cell arrays). This allows for the flexible storage of different data types. So to access a statistic you query its class property using “dot” notation to specify the property and “{}{}” notation to reference its appropriate cell. With the “{}{}” indexing, the first “{}” specifies the ERP’s cell and subject index; the second “{}” identifies the microstate. The following examples illustrate the referencing notations for the grand average, subject average and subject-specific microstate statistics. These examples assume that the class instance,

\texttt{Sim\_ERP\_Data\_Microstates},

has first been instantiated by calling the class constructor.

\textbf{Grand Average}

To query the statistics of the k-th microstate of the grand average ERP, type at the MATLAB command prompt:

\texttt{Sim\_ERP\_Data\_Microstates.Grand\_Average\_Microstate\_Stat\_Label\{1, 1\}\{k\}}

For example, the duration of the 2nd microstate of the grand average ERP is given by

\texttt{Sim\_ERP\_Data\_Microstates.Grand\_Average\_Microstate\_Duration\{1, 1\}\{2\}}

\textbf{Subject Average}

To query the statistics of the k-th microstate of the j-th cell’s subject average ERP, type at the MATLAB command prompt:

\texttt{Sim\_ERP\_Data\_Microstates.Subject\_Average\_Microstate\_Stat\_Label\{1, j\}\{k\}}

For example, the duration of the 2nd microstate of the 3rd cell’s subject average ERP is given by

\texttt{Sim\_ERP\_Data\_Microstates.Subject\_Average\_Microstate\_Duration\{1, 3\}\{2\}}

\textbf{Subject Specific}

To query the statistics of the k-th microstate of the j-th cell’s i-th subject-specific ERP, type at the MATLAB command prompt:

\texttt{Sim\_ERP\_Data\_Microstates.Subject\_Microstate\_Stat\_Label\{i, j\}\{k\}}

For example, the duration of the 2nd microstate of the 3rd cell’s 4th subject-specific ERP is given by

\texttt{Sim\_ERP\_Data\_Microstates.Subject\_Microstate\_Duration\{4, 3\}}