PLAN

1. Create / load data signals
2. Import data in the database
3. Analyze

Let us create simple trial data (N=10) with amplitude from 0 to 1. Each trial has a step-signal composed of three blocks 100 points of zeroes, 100 points of ones, and 100 points of zeroes that are stacked within each trial with a random offset from 0 to 100 samples.

```
% MAKE ARTIFICIAL SIGNALS ------------------------------------------------
% 10 trials
nTest    = [zeros(1,100),ones(1,100),zeros(1,100)];
nTrial   = 10;
nSignal  = nan(400,nTrial);
nEvent   = nan(1,nTrial);
for i = 1:size(nSignal,2)
   nEvent(i) = ceil(rand*100);
   nSignal(nEvent(i)+(0:299),i) = nTest;
end
```

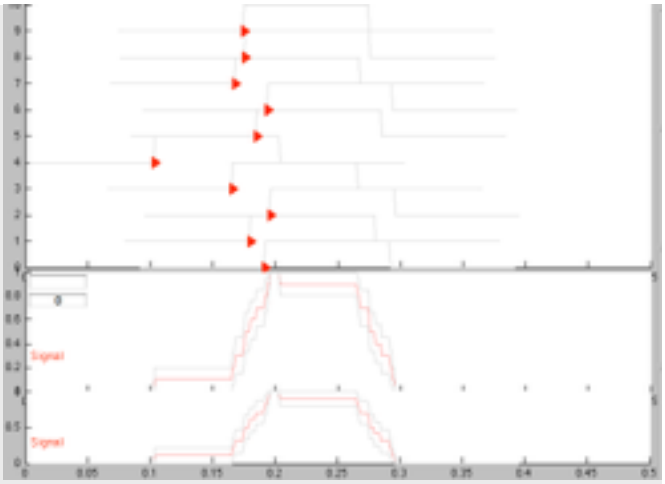
```
>> whos nSignal
  Name        Size            Bytes  Class     Attributes

  nSignal     400x10          32000  double
```

These data signals and events are now inserted into dbData structure with the following code.

```
% POPULATE DB ------------------------------------------------------------
dbData      = dbCreate; % create empty database structure
nRate       = 1000;     % sampling rate
idSession   = setSession('sSubject','Subject1','sPrefix','S1','sSession','TEST1');
% IMPORT SIGNAL
idTrial     = setSignal(nSignal,qry('sTable','Test','sSignal','Signal'),1:nTrial,...
   'nRate'      ,nRate,...
   'sUnit'      ,'sec',...
   'idSession'  ,idSession,...
   'idTrialType',1,...
   'tSync'      ,0);
% IMPORT EVENT
idEvent = setEvent((nEvent+100)/nRate,idTrial,qry('sTable','Test','sSignal','Signal'),
1);
```

```
>> dbData.Test
ans =
   Signal: {1x10 cell}
>> dbData.Test.Signal
ans =
  Columns 1 through 4
    [400x1 single]    [400x1 single]    [400x1 single]    [400x1 single]
  Columns 5 through 8
    [400x1 single]    [400x1 single]    [400x1 single]    [400x1 single]
  Columns 9 through 10
    [400x1 single]    [400x1 single]
```

Fig. Sequential view of single trials. The onsets that are randomly generated with nEvent(i) = ceil(rand*100); and then inserted with offset nEvent+100 are marked with RED triangles. Notice they coincide with the onset of [1] data block.



Now that the data signals and the trial events are imported various analysis can be performed. For example, let us average the signals and plot the confidence interval, which should be zero in this case.

```
% MAKE AVERAGE REPORT ------------------------------------------------------
tEvent   = getEvent(idTrial,qry('sTable','Test','sSignal','Signal'),'on',1);
tLim     = [-.4 .4];
tPeriod  = [tEvent'+tLim(1), tEvent'+tLim(2)];
nData = getSignal(idTrial,qry('sTable','Test','sSignal','Signal'),tPeriod,...
   'bPlot'  ,1, ...
   'bPlotCI',1, ...
   'nTail'  ,2, ...
   'nProb'  ,.95, ...
   'cData'  ,[0 0 0], ...
   'tLim'   ,tLim);
ylabel('Test:Signal (au)')

whos nData
  Name        Size            Bytes  Class     Attributes

  nData     801x10            32040  single
```

This code generates the following figure that we can troubleshoot for possible errors.

Fig. The average is perfectly aligned on the requested event, the block of 100 points with Fs=1kHz is not smeared neither at the onset or offset. The average amplitude is exactly on [1] and the confidence interval is [0].