Bioelectronic Data Analytics Programming Quiz

Note: Each figure or plot should have the following shown:

Title; X and Y axis labels (incl. units), ticks and ticklabels

Matlab Toolboxes needed: Statistics and Machine Learning, Signal Processing Toolbox, Wavelet (if you don't want to use the provided morlet_wavelet function)

Part 1: Matlab Coding Basics

1. Import example.xls into a cell array in Matlab
2. Find the rows which contain the strings "Bike", "west" and "east"
3. Alphabetically sort all bike brands
4. Calculate mean +/- standard deviation of prices of all bikes from east/NE/SE regions, and of bikes from west/NW/SW regions
5. Plot the 2 means in a bar plot; superimpose the STDs of the means as error bars

Part 2: Signal processing

Create a signal that lasts 10 seconds, sampled at 1000Hz. The deterministic part of the signal is the sum of the following three signals

$$3 * \sin(2 * \pi * f_1 * t), \quad f_1 = 60Hz, t = [0 : 10\ secs]$$

$$5 * \sin(2 * \pi * f_2 * t + \varphi_2), \quad f_2 = 25Hz, \varphi_2 = \pi/2, \quad t = [2 : 6\ secs]$$

$$4 * \sin(2 * \pi * f_3 * t + \varphi_3) * \sin(2 * \pi * f_4 * t + \varphi_4), \quad f_3 = 100Hz, \varphi_3 = \pi,$$

$$f_4 = 250Hz, \varphi_4 = \pi/3, t = [5 : 8\ secs]$$

To this signal, add Gaussian white noise ($\mu = 0$, $\sigma = 1$), to get signal S.

1. Using subplot, plot the four different components of the signal individually, and the resulting signal when everything is added.
2. Compute the 1024-point discrete Fourier transform (DFT) of the final signal and plot the single sided amplitude spectrum vs. frequency. Increase the N-points to a number of your choice and plot the new results.
3. Re-sample the signal at two lower sampling rates (100Hz and 500Hz) and plot the resulting signals, along with the original signal, scaled appropriately to span the same X-axis range.

4. The ECG signal of an anesthetized rat is provided in “ecgsignal.mat” (sampling frequency: 500 Hz). Detect the QRS complex of the ECG (hints: use findpeaks; normal heart rate of a rat is ~ 300 beats/min). Use the detected QRS complexes to compute the inter-beat-interval and heart rate of the ECG signal, in beats per minute.

Part 3: Filters and wavelets

1. Band-pass filter the signal S you created in Part 2, to acquire the beta frequency (20-40Hz) band signal. Plot the beta band signal (red trace) on top of the original signal (grey trace). Use milliseconds as time unit.
2. Calculate the wavelet coefficients of the signal S and display the absolute values of the coefficients (frequency vs time). Use a 2-Hz frequency resolution.
3. Load the signal included in the signals.mat, named lfp1. This is a Local Field Potential (LFP) signal sampled at 500Hz, recorded from area V4, while the subject is performing saccades. Create the frequency vector F, needed as an input to the morlet_wavelet function, with the Matlab function logspace, from 2Hz to 150Hz with 100 generated points. Compute the wavelet coefficients for the first 10 seconds of the signal and plot the results. Modify the colorscale of imagesc accordingly so you can see more details on frequencies above 15 Hz.

4. In the signal1.mat there is also a variable called sacc_end. The variable contains the saccade end times (in seconds). Calculate a Saccade-Triggered Wavelet Average (STWA), for 200 msec before and 300 msec after each saccade ends, using the same frequency vector as in step 2. In order to do that, take this 500 msec snippet of the LFP for each saccade, calculate the wavelet coefficients and store them in a 3-d matrix (saccades x freq x time). Take the average of that for the first dimension and plot the resulting STWA.

Part 4: Regression and Classification

1. Create a regression model for the following data set:

```
% independent variable

x = linspace(0, 3, 101)';


% dependent variable with noise (training data)

y = x.^2 -2*x + randn(101, 1);


figure;ax = gca;plot(x, y);
```

2. Perform binary classification on the following data sets {(X1, y1), (X2, y2), (X3, y3)}:

```
X1 = [mvnrnd([-1; -1], 0.01*eye(2, 2), 25); mvnrnd([0; 0], 0.01*eye(2, 2), 25)];

X1test = [mvnrnd([-1; -1], 0.01*eye(2, 2), 25); mvnrnd([0; 0], 0.01*eye(2, 2), 25)];

y1 = [-ones(25, 1); ones(25, 1)];

y1test = y1;

figure('NumberTitle', 'off', 'Name', 'Data set 1');

scatter(X1(:, 1), X1(:, 2), 12, y1, 'filled');


X2 = [mvnrnd([-1; -1], 0.5*eye(2, 2), 25); mvnrnd([0; 0], 0.5*eye(2, 2), 25)];

X2test = [mvnrnd([-1; -1], 0.5*eye(2, 2), 25); mvnrnd([0; 0], 0.5*eye(2, 2), 25)];

y2 = [-ones(25, 1); ones(25, 1)];

y2test = y2;

figure('NumberTitle', 'off', 'Name', 'Data set 2');

scatter(X2(:, 1), X2(:, 2), 12, y2, 'filled');


X3 = [mvnrnd([-1; -1], 0.5*eye(2, 2), 25); mvnrnd([0; 0], 0.5*eye(2, 2), 25); mvnrnd([1; 1],
0.5*eye(2, 2), 25)];

X3test = [mvnrnd([-1; -1], 0.5*eye(2, 2), 25); mvnrnd([0; 0], 0.5*eye(2, 2), 25); mvnrnd([1; 1],
0.5*eye(2, 2), 25)];

y3 = [-ones(25, 1); ones(25, 1); -ones(25, 1)];

y3test = y3;

figure('NumberTitle', 'off', 'Name', 'Data set 3');

scatter(X3(:, 1), X3(:, 2), 12, y3, 'filled');
```