

```
function sleep_Montage(handles, range)
% Stephanie Greer and Jared Saletin
% Walker Lab, UC Berekeley 2011

%% set up
EEG = handles.EEG;
scaleSpace = handles.curScale;
hideCh = handles.hideChans;
h = handles.axes1;
scaleCh = handles.scaleChans;

%electrode names that should be plotted. The order is from the bottom of
%the axes to the top of the axes.
electrodes = {'EMG3'; 'EMG2'; 'EMG1'; 'O1'; 'C4'; 'O2'; 'C3'; 'LOC'; 'ROC'};
%colors for each electrode. The order and length must match the electrode
%list.
colors = 'rrrbbbbkk';

%% Plot all of the electrode data on the main axes (handles.axes1)
hold(h, 'off')
%Loop through each electrode in the above list
for e = 1:length(electrodes)
    %find data for that electrode by name
    for c = 1:length(EEG.chanlocs)

        %This removes anything after a - in an electrode name (i.e.
        %"ROC-A1" becomes "ROC")
        cleanElect = strtok(EEG.chanlocs(c).labels, '-');

        %Checks to see if this electrode should be plotted
        if(strcmp(electrodes{e}, cleanElect) == 1 && ~ismember(electrodes{e}, hideCh))

            %Makes sure the current rage is legal.
            if(range(1) > 0 && range(end) < length(EEG.data))

                %extracts the data to be plotted and multiples by -1 so that
                %the data is plotted inverted.
                centerData = (EEG.data(c, range))*-1;

                %plots the data for this electrode at the appropriate y
                %axis location.
                plotLines = plot(h, range, centerData + e*scaleSpace, colors(e));

                %Adds a tag to each plot line for use with event marking
                for i = 1:length(plotLines)
                    set(plotLines(i), 'Tag', sprintf('scale:%d;chan:%d',e*scaleSpace,c))
                end
                hold(h, 'on')

            % Here the scale lines are plotted if the current electrode
            % is in the scaleCh list.
            if(ismember(electrodes{e}, scaleCh))
                plot(h, range, zeros(size(centerData)) + e*scaleSpace - 50, 'r')
                plot(h, range, zeros(size(centerData)) + e*scaleSpace - 25, 'r')
                plot(h, range, zeros(size(centerData)) + e*scaleSpace, 'r--')
                plot(h, range, zeros(size(centerData)) + e*scaleSpace + 25, 'r')
                plot(h, range, zeros(size(centerData)) + e*scaleSpace + 50, 'r')
```

```
        end
    end
end
end

%% This section checks for any events in the current data range and plots them

%checks for events
if(isfield(handles.stageData, 'events'))
    ylimVal = ylim(h);
    labels = fields(handles.stageData.events);

    %loops through all the labels in the events struct
    for l = 1:length(labels)
        %gets the data from the current label
        cur = eval(['handles.stageData.events.', labels{l}, ';'']);

        %checks for any events that are in the current range
        curEvents = cur(cur(:, 1) >= range(1) & cur(:, 1) <= range(end), :);

        %plots all the events in the current range
        for c = 1:size(curEvents, 1)
            m = plot(h, [curEvents(c, 1), curEvents(c, 1)], ylimVal, 'k', 'LineWidth', 2);
            set(m, 'Tag', labels{l});
            if(gca ~= h)
                axes(h)
            end
            text(curEvents(c, 1) + 5, 10, labels{l})
        end
    end
end

%% This section fixes the axes labels and ranges

% Y axis - uses electrode list as labels
maxY = scaleSpace*(length(electrodes) + 1);
set(h, 'Ylim', [0, maxY], 'YTick', scaleSpace:scaleSpace:(maxY - scaleSpace), 'YTickLabel',
electrodes)

% X axis
srate = EEG.srate;
newX = [range(1), range(end)];
xlim(h, newX)
set(h, 'XTick', newX(1):1*srate:newX(2), 'XTickLabel', ((newX(1) - 1)/srate):1:((newX(2) -
1)/srate), 'XGrid', 'on')

%% This section plot lights on and lights off lines

onT = etime(datevec(handles.stageData.lightsON), datevec(handles.stageData.recStart))*srate;
offT = etime(datevec(handles.stageData.lightsOFF), datevec(handles.stageData.recStart))*srate;
plot(h, [onT, onT], [0, maxY], 'r', 'LineWidth', 2);
plot(h, [offT, offT], [0, maxY], 'g', 'LineWidth', 2);
```