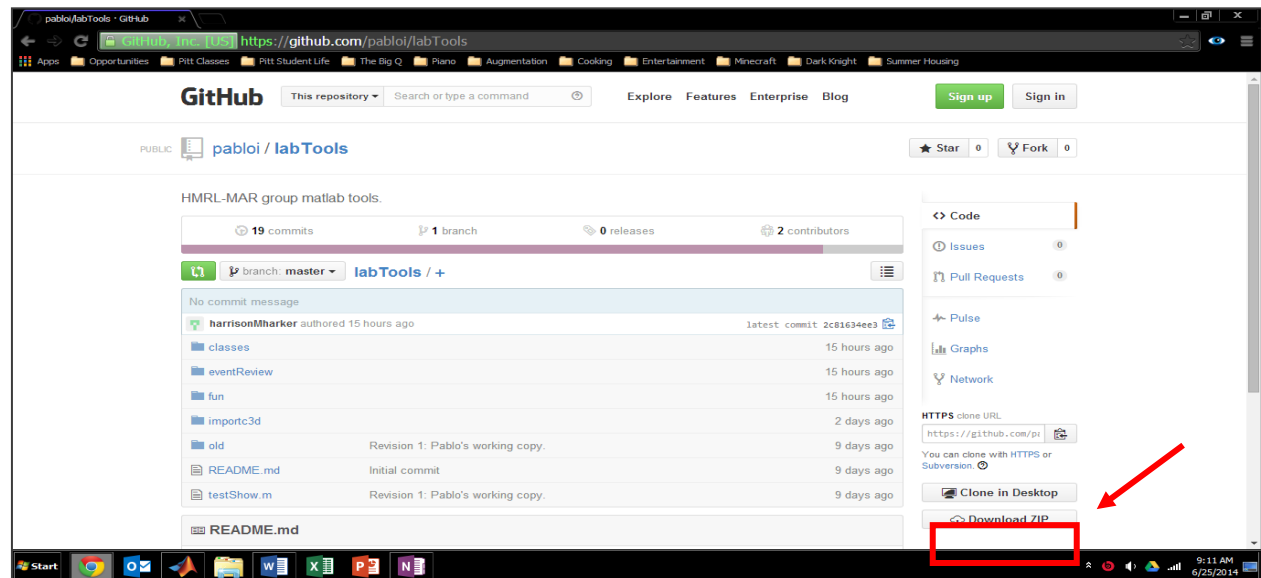# LabTools User Guide

Before using labTools, make sure that the entire labTools package was downloaded. To download, either copy the labTools folder from the server to your personal computer or go to

https://github.com/pabloi/labTools.git

to download the most recent version.

You should see at least four folders: classes, eventReview, fun, and importc3d. These folders and the subfolders they contain should be added to your MATLAB® path (see appendix if unsure how to do this).
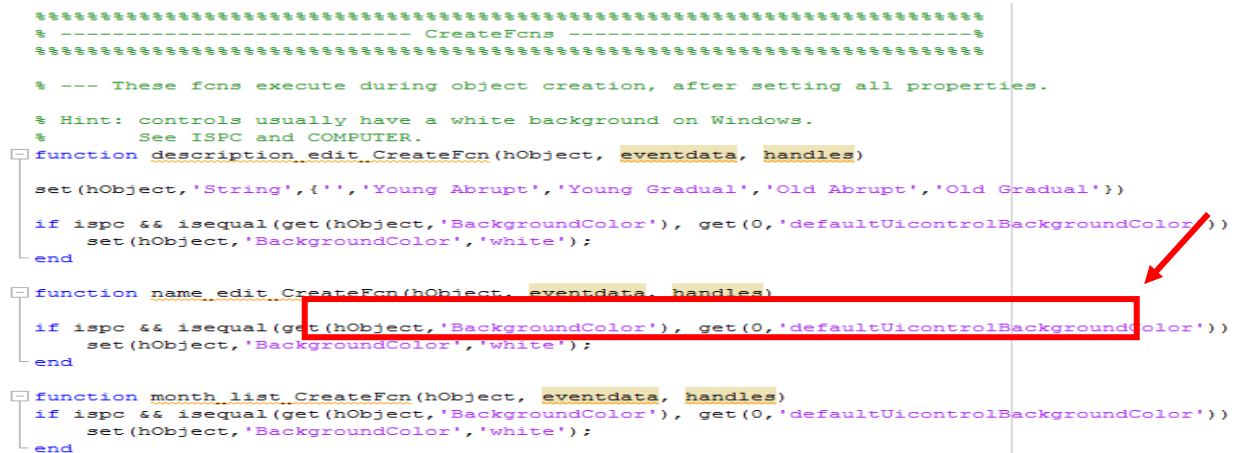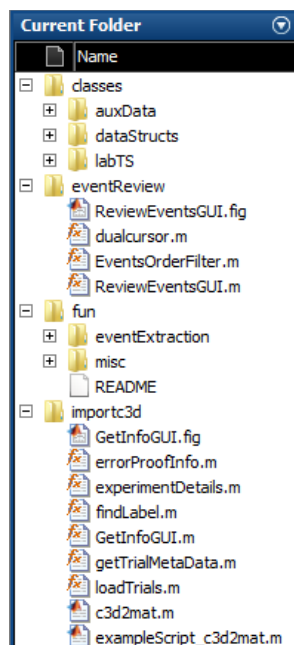
Additionally, you should download the Biomechanics Toolkit and add those folders/subfolders to your MATLAB® path as well, if you have not already done so (see appendix).

After data is collected in the lab, Vicon can export the data files in the 'c3d' format. These are the files that the codes work with and that you will want to have saved on your computer. The purpose of labTools is to convert these .c3d files into a form that is easier to work with within MATLAB®. Before any data can be processed, there are a few setup steps to perform first:

1. Type 'c3d2mat' into the MATLAB® command window. You should see the GetInfoGUI appear. The very first drop-down list is where the experiment description is chosen. Notice that if one of the descriptions are selected, the fields in the 'Trial Info' section of the GUI automatically fill in. Chances are, these pre-defined experiments aren't the exact same ones you will be performing/processing. To add your own experiment types to this list, follow steps 2-4. You can exit out of the GUI now (ignore the error message that appears).

2. Type 'edit GetInfoGUI' into the command window. Now use ctrl+f with 'CreateFcns' or just scroll down to the part of the code that contains the 'CreateFcns'. The first function should be the 'description_edit_CreateFcn'. Here you will want to add the experiment descriptions that you will be using. Simply replace or append your experiment descriptions to the set(hObject,'String',...) command. Make sure that the first element in the cell is a blank string (i.e. ''). Make a record of the descriptions you set and then save and close out of GetInfoGUI.m.

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ------------------------- CreateFcns -----------------------------%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- These fcns execute during object creation, after setting all properties.

% Hint: controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
function description_edit_CreateFcn(hObject, eventdata, handles)

set(hObject,'String',{'','Young Abrupt','Young Gradual','Old Abrupt','Old Gradual'})

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function name_edit_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function month_list_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

3. Now, type 'edit experimentDetails' into the command window. This is where you will define the condition numbers, names, and descriptions along with the expected trial numbers that belong to each condition (note that all of these can be edited in the GUI, so don't worry about small variations). First you will need to add a 'case' for each of the experiment descriptions you added in step 2. If two descriptions follow the same paradigm, then include both descriptions for a single case.

```matlab
switch expDescrip
    case {'Old Abrupt','Young Abrupt'}
```

Second, add the condition numbers to the 'for condition =' command and edit the number of conditions in the 'set' command.

ex1:
```
%condition numbers
for cond = 1:10
    eval(['set(handles.condition',num2str(cond),',''string'',''',num2str(cond),''')'])
end
set(handles.numofconds,'string','10')
```

ex2:
```
%condition numbers
i=1;
for cond = [1 2 4:10]
    eval(['set(handles.condition',num2str(i),',''string'',''',num2str(cond),''')'])
    i=i+1;
end
set(handles.numofconds,'string','9')
```

Finally, the names, descriptions, trial numbers, and over ground specifier for each condition can be set. (Note that the numbers in the set commands DO NOT correspond to the condition number. The set command numbers should always be 1,2,3,4... whereas the conditions are allowed to skip a number.) The *condNames* should be simple and (if possible) the same as what other students in the lab are using to name their conditions. The *descriptions* can contain whatever information seems important to someone who is unfamiliar with what you did, such as the number of strides and possibly the speeds the belts were set at. The *trialnums* should be the expected numbers for that condition (multiple numbers can be set as '1:5', '1,2,3' or '4 5' but not '1-5'). The *OGcheck* values should then be set to 1 for any condition that is overground walking.

4. Type 'c3d2mat' again and select each of your descriptions to make sure the fields populate correctly. Fix any errors that you encounter.

5. The last thing you want to make sure to do before processing any of your data is to check that the marker labels from Vicon will be recognized by the processing functions in labTools. Certain markers must have the correct labels in order for the codes to work, and these are set in the 'findLabel' function. Type 'edit findLabel' into the command window to see which markers these are, and add any labels that aren't included to the appropriate case.

You are now ready to process your first subject!

Step 1. c3d2mat

This is the first step in processing any subject. Run c3d2mat and this time fill in all the information in the GUI. Note that if no folder is selected in either the '.c3d file location' or the 'Subject file save location', the current folder is assumed. Once the information is filled in, hit 'OK' and MATALB® will ask if there are any observations for individual trials. These are things like 'left ankle marker fell off around step 100'. Type 'y' for yes or 'n' for no. If you type 'y', then you will be able to select each trial

for which you have an observation. When complete, simply hit 'Done' at the end of the menu. Now MATLAB® will go to work, so just pay attention to any of the warnings that pop up until the script is finished running.

Once this script is finished, there should be three variables in the workspace and three files saved in the subject save file location.

Let's start with the workspace variables. The 'data' variable is a cell that is the length of the number of trials processed. Within each of the cells, all of the data (marker data, GRF data, gait events, angles, EMG data, belt speed data, etc.) for an individual trial can be accessed. There is also a 'meta data' field in every cell that contains general information about that specific trial. The expMD variable contains information that applies to the experiment as a whole such as the date, the experimenter, and the conditions tested. Finally, the subData variable contains information specific to the person who was tested such as height, weight, age, and gender.

The three files that were saved in the save folder location should be named (Subject ID)RAW.mat, (Subject ID).mat, and (Subject ID)info.mat. The RAW data file is just that, it contains only the raw data collected from VICON. The regular .mat file contains the raw data but has been processed, and therefore has events, angles, processed EMGs, and calculated belt speeds. The info file contains the information entered in the GetInfoGUI and is what would be needed to load that subject as an existing subject in the GUI (more about this later).

Step 2. ReviewEventsGUI

When you type 'ReviewEventsGUI' in the command window, a blank version of the figure shown below should pop up. The first time the event reviewer is run, you will want to make sure that there are no major problems (like drop-outs or mislabeling) with the important markers. An easy way to do this is by checking the angle data since angles are composed of at least two markers and therefore more markers can be reviewed at a time. Viewing the max time window also makes this process easier. If you do find a problematic trial, re-processed that trial in Vicon, replace the old .c3d with the new, run c3d2mat again, but this time press 'Load Existing' and look for the (SubjectID)info.mat file. The information should all be the same, so you can just hit 'OK' and a new subject file will be generated.
Now that the marker data is correct, the events can be reviewed.
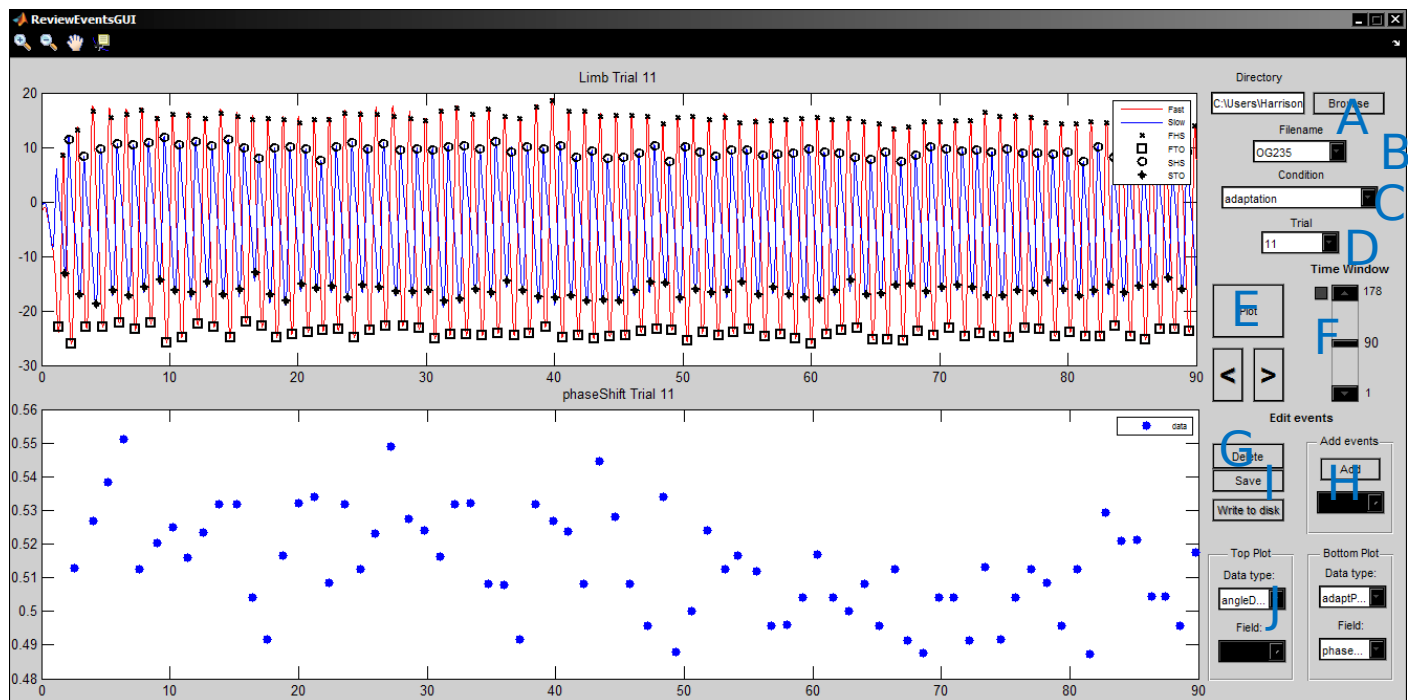
The ReviewEventsGUI:

A. Select the directory where the files of the subject under review are located (if it is not the present working directory)

B. Select the correct file from the dropdown list. (It should be the plain subject ID file and not the RAW or info files)

C. The condition can be specified, but if looking at a subject for the first time then it is a good idea to just start from the beginning (default)

D. The trial within the condition can also be specified, but once again don't change this if looking at the subject for the first time.

E. The plot button is actually rarely used, but is sometimes needed to refresh the plots such as when zooming in/out on the graphs. The back/next buttons can be used to advance frames within a trial, and will automatically advance to the next trial/condition when reaching the end.

F. Choose the number of seconds you want to look at. If you always want to look at the full trial, click the check box near the top arrow button. This will ensure the full trial is visible even when changing which trial is being shown. Note that a smaller time window cannot be viewed as long as the box is checked.

G. Hitting 'Delete' will being up a crosshair. This can be placed over an event and then clicked to delete it. Delete as many events as desired and then hit the return (enter) key to refresh the plot.

H. Hitting 'Add' will activate a drop-down list which allows you to choose which event you would like to add. Once an event is chosen, a crosshair will appear which allows you to add the event by clicking on the graph (hitting return not necessary).

I. If you are confident with the events after deleting/adding wherever necessary, you can hit 'save' which activates the 'Write to disk' button. Clicking this button will overwrite the subject file with a new one that has the updated events. A new file will also be generated called (Subject

ID)params.mat. This file contains the experiment and subject data and a data variable that only contains the adaptation parameters.

J.   The Top Plot and Bottom Plot drop down lists allow you to choose what data you want to view.

Step 3. Plot adaptation parameter time course

The subject is now processed and should be ready to be analyzed. There are some basic data viewing functions built-in to the adaptationData object (which is stored in the (SubjectID)params.mat file). Load the param.mat file and you should see a variable called adaptData in the workspace. To see which parameters are available to be plotted, type something along the lines of 'labels=adaptData.data.labels'. Now, if you type 'adaptData.plotParamTimeCourse(*specify a parameter or a cell of parameters here*)' A colorful plot of the parameters entered will appear. If you would like to see the data divided up by trial instead of by condition, use .plotParamTrialTimeCourse   instead.

If you are interested in plotting the average values of different parameters early and/or late into each condition to compare them, you can use the function plotParamByCondition, with the same syntax as before. This will result in a bar plot of the average values of the parameters in the first 3/5 strides of each condition, and   the   last   20   strides   of   each   condition.

You might be also interested in comparing several subjects, or even several populations in this way. You can do this by calling the static function adaptationData.plotGroupedSubjects. First you need to have the files that contain the adaptData variables for each subject (param.mat) available in your current directory. Once that is done, you can call:
adaptationData.plotGroupedSubjects({'Sub1file.mat','Sub2file.mat','Sub3file.mat',... },{'parameter1','parameter2',...})
Note that the first cell array is an array of strings, each one pointing to the name of one of the files containing adaptation data for a subject. The second cell array is an array of strings with the names of the parameters you wish to plot.

If you wish to compare populations (two or more groups of several subjects each), you can use the same function substituting the first argument. In this case, the first argument needs to be a cell array of cell arrays of strings, for example:
{{'Group1Sub1file','Group1Sub2file',....},{'Group2Sub1file','Group2Sub2file',...}, {'Group3Sub1file',...},...}
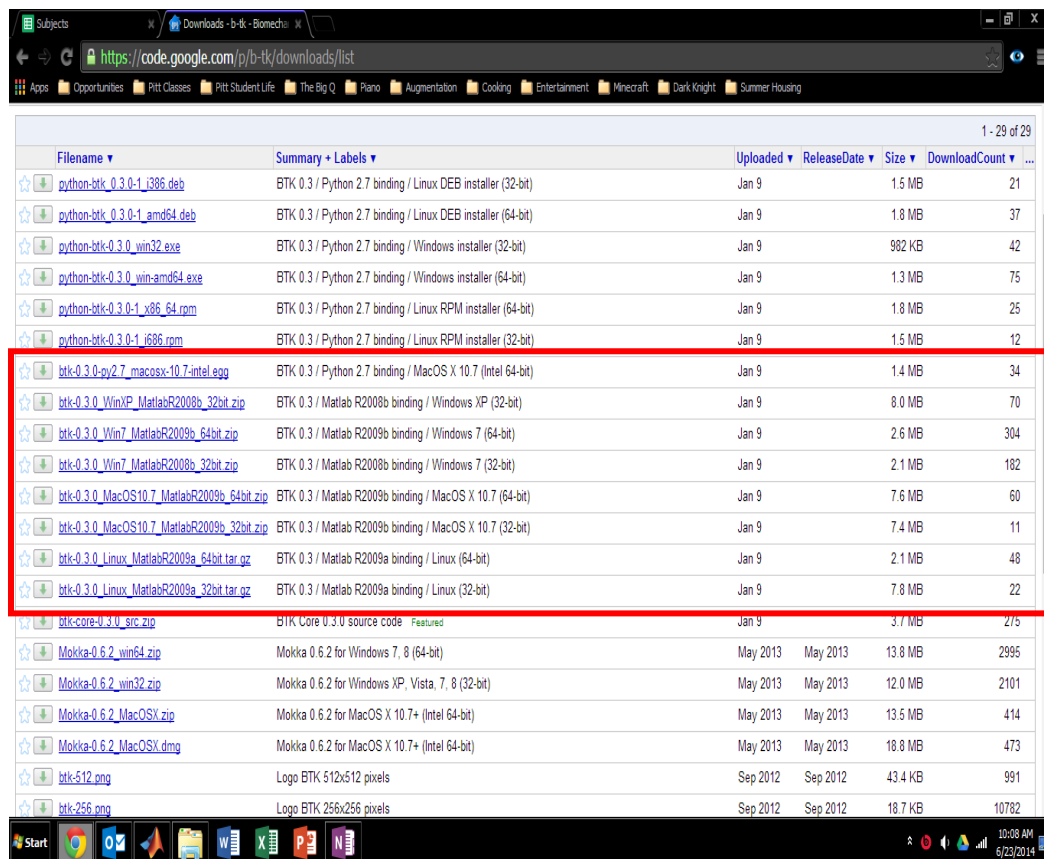
Appendix

Adding folders to the MATLAB® path:
Go here for a how-to:
http://www.mathworks.com/help/matlab/matlab_env/add-remove-or-reorder-folders-on-the-search-path.html
Make sure to choose 'add with subfolders...' as opposed to just 'add folder...' and to save changes before closing.

Downloading the Biomechanics Toolkit
Go to the btk website: https://code.google.com/p/b-tk/downloads/list
and then download one of the following downloads that corresponds to your operating system:

Unzip the folder that was downloaded and then add the folder (with subfolders) to your matlab path.