OMLAB Analysis Tools (OMtools)
last update: 15 June 2011

ABOUT THESE FILES:
This collection of m-files allows you to read and analyze data files generated in the following formats:

- -OMLAB's LabVIEW format
- -OMLAB's RETRIEVE format (versions 1.02 and 1.04)
- -ASCII format (column-oriented data)
- -float-32 binary format (contiguous and interleaved data blocks)
- -ober2 format (NOTE: import filter not updated since 2004)

If you are not using the OMLAB LabVIEW VIs for data acquisition, it is most likely that your data format of choice will be plain ASCII. Benefits of this format are simplicity and portability (almost every program can export data as ASCII).

However, for large datasets the files can quickly become enormous and take a long time to load. Therefore OMtools includes a utility ("txt2bin.m") that will convert a correctly formatted ASCII file into a MATLAB binary file that can be read in much more easily.

You should read the included plain-text files (located in "Documentation"), included with this distribution, to understand how these programs work, and what you need to do to use them most effectively:

- How to RD data
- Cal instructions
- Convert EDF files (if you are using the EyeLink video system)

Global variables and data structures:
If you wish to write your own code read the document "d_struct.txt" for an explanation of the global variables available to you. "d_struct.txt" also contains descriptions of the files used to hold saccade, refix and adjustment data. Most of this information is horribly technical, and of no interest to the casual user. Mostly, it points out what a pain this sort of thing was to put together in the days before MATLAB included data structures. Maybe some day I'll rewrite everything. Maybe not.

****

HOW TO SET UP "OMtools"

New installation

NEW METHOD (MATLAB 2009* and later):

There are several places you can copy the OMtools directory. However, The MathWorks officially supports only placing user-created toolboxes such as OMtools in your own personal "MATLAB" directory which is located in your home directory's "Documents" folder. (See: http://www.mathworks.com/help/techdoc/matlab_env/br8p2lc-1.html). If you choose to install OMtools in this official location you will need to place a copy in the Documents/MATLAB folder of each user who will run MATLAB.

OLD METHOD (prior to MATLAB 2009*):
Copy the "OMtools" directory into the same directory as the "MATLAB" application. For Mac users, this is the directory that contains the file "MATLAB". For PC users, this is the directory that contains the file "MATLAB.exe". (However, the ".exe" part of the name may not be visible, if the option to hide file extensions is active. In this case, you may see two files in the same directory with the name "MATLAB." They will probably both have the MATLAB icon. This is the correct directory.)

---
* I am not sure exactly when the change occurred. Your best bet is to check whether or not there is a /Documents/MATLAB folder, and if it exists, use it.

Updating an existing installation
If you installed OMtools previously, it may be in a location no longer supported by MATLAB. Two common places are in matlabroot (the directory that contains the MATLAB program), or in the MATLAB/Toolbox directory.

The updated directory management for OMtools was designed to work with these locations, but has not been extensively tested. If you have problems, consider moving OMtools to the official location for user-added toolboxes, the "MATLAB" folder in your "Documents" directory.

You should also move the "omprefs" folder, which was originally located alongside OMtools in matlabroot, to its new supported location inside OMtools.

Automatic Startup of OMtools
Make a copy of "STARTUP.m" (in the "OMtools/DATADIRS/to_MLbin" directory).
If you are using a Macintosh, place it in the "Toolbox/Local" directory.
If you are using a PC, the location seems to change frequently. Your best bet is to check the MathWorks web site (www.mathworks.com) and search for "startup.m".

"STARTUP.m" will execute whenever MATLAB is launched, adding several paths to MATLAB's search path. It is vital to add the paths, or you will not be able to run the OMtools files. If you have successfully added "startup.m" you will see the message "Setting OMtools paths..." as MATLAB initializes.

****

Here is a listing, by directory, of the most commonly used OMtools files, and a brief description of their function. For more complete information about any given program, including how to use it, type 'help' (without the quotes) followed by the function name. For information about files not listed here, read the "contents.txt" file from which this was excerpted. (NOTE: there may be files that do not appear in "contents.txt". In most cases the omission is an oversight.)

In the "RD" directory:

rd:
Reads in data from: ASCII, RETRIEVE, ASYST, LabVIEW, raw binary or ober2 file formats. RD will call the appropriate routines. There is a tutorial concerning rd's proper usage in the "documentation" folder.

biasgen:
Interactive program to generate the "adjbias.txt" files that contain the calibration information used by "adj.m" to offset and scale the data as RD reads it in.

cal:
Perform n-pair, asymmetric calibration of data with the assistance of "zoomtool". "cal" is reasonably interactive, prompting the user to perform the necessary actions. Give it a try.

dataclr:
Quickie script that clears all the data structures and associated variables created when RD is used. Use when you want to clear memory before loading in a new file with RD. (It is better to use this than "Clear all" or "Clear" because dataclr will only clear the variables created and used by RD and its subfunctions, leaving your other variables and functions alone.)

datstat:
Tells you what files are loaded into memory and what files have been selected for current use.

edf2bin:
Preprocessor applications that convert '.ASC' files generated by EyeLink's 'edf2asc.exe' utility to binary (edf2bin) format.

enviro:
GUI-based utility that reads from, and writes to, the file "enviroG.mat"
Allows the user to change enviromental variables:
 -showGraphs -- should RD display graphs of the loaded data?

```
 -doLoadSacs  -- should RD display the saccade event data?
 -doScaling   -- should RD apply the adjust bias?
 -doRefix     -- should RD apply the refix data?
 -doFiltering -- should RD apply the filtering data file?
 -doUnfolding -- should RD apply the unfolding data file?
 -doHeadAdj   -- leave set to zero
 -debugme     -- should RD clear non data-related variables upon finish?
 -export      -- save analysis data to "EXPORT" directory?
 -useTimeAxis -- Plot versus TIME, rather than SAMPLE


------------------------------
```

In the "UTILS" directory:

bridgenan:
Replace NaN points in an array with a linear fit line between non-NaN edge points.

d2pt:
Perform a 2-point differentiation with some filtering.
Example: Given an array named "pos", containing eye position data, calculate the differentiated result and name it "vel":

```
vel = d2pt(pos,cutoff,samp_freq);
```

deblink:
Remove blinks in data taken from analog out port of the EyeLink video system. Blinks are determined using position, velocity and acceleration criteria.

desacc:
Interactive, GUI-driven program to remove saccades from a data record. User can set position, velocity and acceleration criteria.

efit, efitfun:
Calculate an exponential fit to an input vector.

hv2r:
Combine horizontal and vertical position segments (of equal length) into a radial representation of eye position.

isdigit:
find the elements of a string that are digits.

isnull:
checks if input 1) is empty, 2) is all NaN or 3) is all zero.

maket:

Create a time vector for a given data array.
Usage: t = maket(array, sampling frequency);

nameclean:
allows for proper printing of strings with carets or underscores.

pad:
Pads input vector with "NaN" (Not A Number) elements to fit a given length.

pause2:
improved "pause" functions, with accuracy to around 10 milliseconds.

pfit, pfitfun:
Calculate an polynomial fit to an input vector.

regress2:
Given X and Y datasets, REGRESS2 will calculate a linear regression and r-squared value, and optionally will plot the result.
Usage: [slope, intercept, r_squared] = regress(x_vector, y_vector, plot/noplot); where an argument of zero for 'plot/noplot' will suppress the graphical output.

stimgen:
simple command-line program to build stimuli composed of linear segments.

stretch:
world's simplest interpolation program: simply repeats each point the specified number of times.

stripbad:
[newX, newY] = stripbad(x, y, bad). The "bad" list contains the indices of the points to be stripped from x and y. It can also be called with just two arguments as follows: [newX] = stripbad(x, bad).

stripcom:
Strips "comments", i.e., all text following a set of delimiters: ! @ # $ % ^ & *
Returns the text up to the first occurrence of one of these characters.

stripnan:
Strips "NaN" (Not A Number) elements from input array.

strtok2:
Modified version of MATLAB's original STRTOK function. _This_ one actually works correctly, however...

stt:
Converts a plot's X-axis units from sample number to time.

sub:
Make a subarray from a given array, using time coordinates.
Usage: [subArray] = sub(array, startTime, duration);

time:
Display the date and time, or save it to a character array.

tts:
Converts a plot's X-axis units from time to sample number.

wherept:
Reports the current mouse location in a graph window.

xaxshift:
Shifts a plot's X-axis coordinates by a specified amount.

zeronans:
Convert all the NaNs in an array to zeros.

In "Utils:Filt"

dofft:
Perform and display an FFT of selected data.

hpf, lpf, notch:
Various linear filters.

medfilt:
Apply a median filter (3, 5, 7, 9 or 11 pt) on an array.

mvgavg:
Perfrom a moving window average (3, 5, 7, 9 or 11 pt) on an array.
Optionally you can specify a windowing function: 'h' = Hamming.
Example: a_filt = mvgavg(a, n, 'h'); where n = 3, 5, 7, 9 or 11.

In "Utils:Pickdata"

pickdata:
Used when you wish to load multiple data files and select from them for analysis. Pickdata will concatenate the data into large arrays. This was written in the days before MATLAB had proper data structures, and therefore is a bit of a relic necessary when using any of these analysis routines:

TPLOT3, SACLENS, SAC2, VELPLOT, ADJ, SANITY, CHECKPTS, WHENSLIP or SHOWME.
You simply check the boxes for the data you wish to include.

In "Graphing"
colrpick:
Graphic tool to determine RGB triplets for a color.

axisedit, linedit, textedit, posedit:
Used to edit the content/appearance/layout of graph elements.

bleach:
Converts an open Figure so that all lines/text/objects, etc will appear in black and white.

dragger:
Turns object dragging (lines, text, etc.) on or off. Double-click on the object you wish to drag. If the cursor turns to crosshairs, the object can be dragged. Keep the mouse button down and move the mouse to where you want the object to move. The screen will update dynamically throughout this process. Dragger communicates with the drawing program (DRAW.m) so that only one or the other can be active at any time.

draw:
Allows the user to draw lines, rectangles, arrows and circles on a plot. Place the pointer where you want the object to start. Click the button and move the mouse. A guide line will appear on the screen between the start point and the present point. When you release the button, the object will be drawn.

findhotw:
Find the foremost editable (i.e. non-control) window.

findme:
Look for an open window with a given string in its 'Tag' field.

findwind:
Look for an open window with a given string in its 'Name' field.

grapcopy:
Allows the user to copy a graph and all its associated elements (lines, surfaces, patches, text, axis labels, title) from one figure to another. Particularly useful for going from full plots to subplots and vice versa.

killobj:
Removes objects (including entire plots!) from a figure window. If you wish to delete an object, click on it and type 'killobj' You will be asked to confirm if you wish to delete the object. Be careful, because it is possible to delete a plot axis if you don't read killobj's confirmation message. ("Do you really want to delete this axis object" means "Do you want to erase this graph".)

linebacker:
(Attempt to) send the selected line behind all the other lines in a plot.

objcopy:
Copy an object and reproduce it in another set of axes.

rotplot:
Allows the user to interactively view 3-D data from any view angle and distance.

scalefig:
Scale figure axes down. Usage: scalefig(x), where 'x' is the percent of the figure window that the figure axis should use. Hint: try 0.75 to start. For SINGLE-AXIS figures only.

setdata:
Substitutes given data for current data in an already-plotted line.

shrinky:
Decimate the data in a figure, decreasing the file size. Decimation factors of 2-10 are OK for printing figures.

wysiwyg:
"What You See Is What You Get." Attempts to modify on-screen plots to resemble the way they will appear when they are printed. Alternatively forces plots to print the way they appear on the screen.

yankdata:
Click on a data object such as a line or set of point. YANKDATA will make a copy of the x,y or z data contained in this object.
Usage: a   = yankdata('x');   will put the x data into 'a'
                [a,b] = yankdataa('xy'); will put x data into 'a', and y data into 'b'
                You can select any combination of 'x','y' and 'z'.
        If you do not specify 'x', 'y', or 'z' YANKDATA will prompt you for the data you wish to extract.

yankdataa2:
Same as YANKDATA, except that it only extracts the data between the figure's current X-axis limits.

In the "ANALYSIS" directory:
akshift:
Convert between Total mm of R&R and Null-Angle shift.

```
Usage: out = akshift(mm_or_ang, inputValue, rr);

calcshift:
Calculates the shift necessary to make the midway point between the vertical
cursors equal to zero.

dist2conv:
Convert between Distance and Convergence Angle.
Usage: out = dist2conv(dist_or_conv, ipd, inputValue);

eyemovie:
Shows an animation of eye movement in near-real time.
usage: eyemovie( 'direction', start, stop, speedFactor )
         direction:   'h' -- horizontal only
                      'v' -- vertical only
                      'b' -- both directions
         start, stop: enter as time or sample number
         speedFactor: default is 1, higher values cause movie to run slower
                      (You may need to experiment to find what works for you)

nafx:
Command line and GUI-based function to perform eXpanded Nystagmus Acuity
Function calculations.  Can also display points that meet the position and/or
velocity criteria.

pd2ang:
Convert between PD and Angle for Small Angles (<30D).
Usage: out = pd2ang(pd_or_ang, inputValue, conv);

snel2log:
Convert between LogMAR and Snellen acuities.

va2nafx:
Convert between Snellen acuity and NAFX.

va2snel:
Convert between Visual Acuity decimal and Snellen fraction.


------------------------------


In the "LABELS" directory:
<<<There are zillions of files here.  Most of them are simple one- or two-liners
that call the "drawXXX.m" primitives.>>>

drawbox:
Plot a foveation window on top of an existing phase plane.
```

drawcirc:
Plot an ellipse/circle on top of an existing vertical vs horizontal plot.

drawrad:
Plot a foveal radius on top of an existing position vs time plot.

fovrad, sliprad:
Draw a foveal radius (+/- 0.5 deg) or a slip radius (+/- 4 deg/sec) on the current plot.

la:
Draw a text label on a plot.

In the DATADIRS directory:
ompath:
Adds all the paths that OMtools needs.

omdir:
Go to the OMtools directory.

datadir:
Go to the root data directory.

In the ZOOMTOOL directory:

zoomtool:
Zoomtool is an interactive graphics display, that provides two sets of cursors and some relatively self-explanatory control buttons for displaying segments of data, zooming in on points of interest and displaying their x- and y-values, and placing them in memory. Zoomtool works on single-axis, 2-D plots that have their data points equally spaced along the x-axis. To use, plot your data and then type "zoomtool".