

SOME IMPORTANT CORE FILES:

In the "RD" directory:

ADJ.m:

Used by rd_xxxx files to apply the zero adjustment and maximum/minimum calibration values for data during calibration.

Example: `rh = adj(rh)` -- will use data just entered with ZOOMTOOL
`rh = adj(rh, zero_point, max_adjust, min_adjust)` will use previously found values of these variables.

ADD_COL.m:

Used by rd_rtrv, rd_ascii, rd_asd, rdLABV and rd_ober2 to add additional files' data to the existing eye data arrays.

BIASGEN.m:

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.

CHECKPTS.m:

Sanity check. Makes sure that the data files are associated with the appropriate control point files. Plots the saccade onset and offset points or the cycle beginning and end on top of the data trace (position, velocity or acceleration).

DATACLR.m:

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.m:

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

DOFILT.m:

Called during RD to load filter ('.f') files that store information about what filtering to apply to the data that is being read in.

EDF2BIN.m, EDF2LAB.m:

Preprocessor applications that convert '.ASC' files generated by EyeLink's 'edf2asc.exe' utility to either binary (edf2bin) or ASCII (edf2lab) format. Sorry about the bad naming of these files, but what's done is done.

ENVIRO.m:

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

GLOBALIZ.m, GLOBAL2.m:

Necessary to allow the various functions to share data. You only need to use this if you are reading in a saved MATLAB workspace file without using RD. If you are using RD this is done automatically. (GLOBAL2 is the more up-to-date version, replacing GLOBALIZ)

is_wform.m, is_sactype.m:

Determines if waveform/saccade type read in is valid.

MAKEPTCH.m:

Used by REFIX (but could conceivably be called by the user) to create a smooth patch between the shifted and unshifted portions of the data array.

MOVEDATA.m, SWAP_EM.m, ZAP_EM.m:

Used to change the order of data files and their associated detritus after they have been loaded by RD. Not usually worth messing around with (and possible dangerous).

pick_wfst.m:

Called by fill_scp to select which saccade control points should be loaded if there are several sets in the ".S" file.

RD.m:

Reads in data from either ASCII-, RETRIEVE-, ASYST-, LabVIEW- or ober2-format files. RD will call the appropriate routines.

READBIAS.m:

Used by rd_ascii, rd_rtrv, rd_asd, rdLABV and rd_ober2 to read in biases from a text file.

rd_ascii.m:

Called by RD to read ASCII data. The data must have the same number of columns in every row, or MATLAB will become upset. There should be nothing

but numbers in the data file (i.e. no text headers). The filename suffix must be "txt". (((NOTE: in the unlikely case that anyone ever wants to go slumming around in ASYST, make sure that any ASCII files you are going to use have three blank lines before the beginning of the data. For some reason it was decided that `_this_` was easier than writing the differentiation routines correctly.)))

`rd_asd.m`:

Called by RD to read ASYST native-format data. The filename suffix should be "asd".

`rd_bin.m`:

Called by RD to read headerless binary-format data. The data must be in 'float32' format, and can be stored either contiguously or interleaved. The filename suffix should be "bin".

`rd_dig.m`:

Called by RD to load in the '.dig' files created by our LabVIEW data acquisition software.

`rd_labv.m`:

Called by RD to read in a LabVIEW-generated data file. The filename suffix must be "lab".

`rd_ober2.m`:

Called by RD to read in an ober2 registration file. The filename suffix must be "obr".

`rd_rtrv.m`:

This is the routine to read in RETRIEVE data. It can handle 12- and 16-bit data. It is called by RD. The filename suffix must be "dat".

`rdclear.m`:

Clears all the variables created by RD et al which are not needed once the file has been read in.

`rdscp.m`, `fill_scp.m`:

Called by `rd_xxxx` to load the saccade control points. `rdscp` reads in the points and `fill_scp` places them into the appropriate SCP arrays. (`rdscp` replaces `LOAD_SCP` which is now obsolete.)

`readhdrm.m`, `readhdrd.m`:

Used by `rd_rtrv` to read in the RETRIEVE-format header information for either MacOS (TM) or DOS/WINDOWS (BM), respectively.

`REFIX.m`:

Subtract "local bias" from the selected data trace. REFIX reads adjustments from a "normalization file" named for the file it modifies, with an extension of ".z" REFIX must be called after the initial coil or IR correction (BIAS_ADJ) is performed. RD has been programmed to work in this order. You'd use this if the subject moved after calibration.

In the "UTILS" directory:

ADJDISP.m:

Use when picking a zero point with zoomtool. You can type ADJDISP to plot the data, shifted by the amount you have just selected.

CAL.m:

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.

CALCSHIFT.m:

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

D2PT.m:

Perform a 2-point differentiation with some filtering. The results are identical to those obtained using 2PTDIFF in ASYST. Example: You have an array named "pos", containing eye position data. You want the differentiated result to be named "vel": `vel = D2PT(pos,cutoff,samp_freq);`

EFIT.m, EFITFUN.m:

Calculate an exponential fit to an input vector.

EYEBALLS3D.m:

GUI-based animated 3-D eyeballs that move according to the h/v/t data. Eyeballs3D can also make QuickTime™ movies (Macintosh-only feature).

getnextl.m:

Fetches the next line (delimited by CR and/or LF) from a text buffer.

maket.m:

Create a time vector for a given data array.

Usage: `t = maket(array, sampling frequency);`

NUMSACS.m:

Displays the list of data files selected by PICKDATA, along with the number of saccades associated with that data file.

PAD.m:

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

PFIT.m, PFITFUN.m:

Calculate an polynomial fit to an input vector.

pickpts.m:

Plots the position, velocity/10 and acceleration/1000 on one graph (scaled to fit) and calls ZOOMTOOL. Not particularly useful in general.

PRNT.m:

Automagically print out your data file, N samples to the page.

prnt(array, 'page title', samples/page, time or samp, yAxisLims)

page title: string to appear at top of each page.

samples/page: how many samples will appear on each page.

time or samp: 't' -- use time index for x-axis.

's' -- use sample number for x-axis.

yAxisLims 0 -- all pages use global max/min y-limits.

1 -- each page uses its own max/min y-limit.

procline.m:

Processes a line of text, returning the number of "words" (elements separated by white space) and the list of those words.

readText.m:

Example of how to use "procline," "getnextl" and "stripcom" to read in and process a sample file.

REGRESS.m:

Given X and Y datasets, REGRESS 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.

SHUFFLE1.m, SHUFFLE2.m:

Randomly shuffle an input vector.

stripBad.m:

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

stripcom.m:

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.m:

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

STRTok2.m:

Modified version of MATLAB's original STRTok function. This one actually works correctly, however...

STT.m:

Converts the X-Axis from samples to time.

SUB.m:

Make a subarray from a given array, using time coordinates.

Usage: [subArray] = sub(array, startTime, duration);

SUBP.m:

Plot a subarray versus time. (Calls SUB and TPlot).

Usage: subp(array, startTime, duration)

STT.m:

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

tCONVERT.m:

Convert a pair of time coordinates to sample numbers, or vice versa.

Usage: [out1 out2] = tConvert(in1, in2);

time.m:

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

toASCII.m: Save lh, lv, rh, rv as ASCII files in the EXPORT directory.

tplot.m:

Plot an array versus time.

Usage: tplot(array, startTime, sampling frequency)

tts.m:

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

wherept.m:

Reports the current mouse location in a graph window.

WHERESACS.m:

Tells what file(s) a saccade (or list of saccades) belongs to.

Usage: wheresacs(saccade_vector)

xaxshift.m:

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

In "Utils:Filt"

doFFT.m:

Perform and display an FFT of selected data.

HPF.m, LPF.m, NOTCH.m:

Various linear filters.

MEDFILT.m:

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

MVGAVG.m:

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.m:

This is the front end to the data-preparation file (readbox). Once you have loaded your data, type PICKDATA. Use this before calling any of the analysis routines: TPLLOT3, SACLENS, SAC2, VELPLOT, ADJ, SANITY, CHECKPTS, WHENSLIP or SHOWME. You simply check the boxes for the data you wish to include.

READBOX.m:

This file is called by PICKDATA. It reads the checked boxes and creates the position (and velocity and acceleration) and saccade control point arrays that are used by the analysis routines.

TOGGLE.m:

This is called by READBOX.m and acts to toggle the on/off state (the checkboxes) when determining the files that are to be loaded.

SACSEL.m:

Called by PICKDATA. Allows the user to choose whether to allow loading of files without an associated saccade control points file.

pdclear.m:

Clears the variables no longer needed after PICKDATA finishes.

In the "UTILS:ROTPLOT" directory:

MAKESURF.m:

Stretches a 2-D line into a 3-D surface and plots it as a surface. MAKESURF has several built-in equations including main-sequence and maximum saccadic velocity vs saccade magnitude. As well, you can define your own equations.

In "Graphing"

COLRPICK.m, CPickAct.m:

Graphic tool to determine RGB triplets for a color.

AXISEDIT.m, LINEDIT.m, TEXTEDIT.m, POSEDIT.m:

Used to edit the content/appearance/layout of graph elements.

AEditAct.m, LinEdit4.m, LinEdit5.m, LEditA4.m

LEditA5.m, TEditAct.m, PEditAct.m,

BLEACH.m:

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

DRAGGER.m:

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.

DRAGGY.m:

Called by DRAGGER, not by the user.

DRAW.m:

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.m:

Called by the graphing functions to determine if there are any eligible graph windows currently open

FindMe.m:

Called by the graphing functions to determine if they are already open.

GDRAW.m:

Called by DRAW to do the actual drawing.

GRAPCOPY.m:

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.m:

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".)

LOADFIG.m, SAVEFIG.m:

Used to load/save graph figures.

MYARROW.m:

Called by DRAW. Draws arrows with arrowheads. (This was not easy.)

ROTPLOT.m:

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

ROTBUTT.m:

Called by ROTPLOT to handle button presses.

SETDATA.m:

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

SHRINKY.m

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

scalefig.m:

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.

WYSIWYG.m:

"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.m:

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.

YANKDATA2.m:

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

ZOOMIN.m, ZOOMOUT.m:

Zoom in or out when viewing a 3-D graph. Called by buttons in ROTPLOT but are also standalone files.

In the "ANALYSIS" directory:

AKSHIFT.m:

Convert between Total mm of R&R and Null-Angle shift.

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

AVG_PLO3.m, SD_PLOT3.m:

When you use TPL0T3 to move forwards and backwards in time, you generate an array of the standard deviation and average of the position, velocity and accelerations for the times viewed. These m_files will simply print a graph of the SD or average values at these times (relative to the saccades).

CYCSAC.m:

Calculate and plot the duration of a saccadic cycle (measured from onset of foveation saccade to onset of the next foveation saccade) versus the amplitude of the braking saccade that occurred during that cycle.

DETECTFOVS.m:

Updated version of 'findFovP' called by NAFX.m to find and calculate foveation information.

DIST2CONV.m:

Convert between Distance and Convergence Angle.

Usage: `out = dist2conv(dist_or_conv, ipd, inputValue);`

EYEMOVIE.m:

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)

FindFovP.m:

Used by NFF.m to find the foveation periods for a data file.

FPLEN.m:

Calculate the duration of the foveation period for a data file. This probably doesn't work very well.

MAINSEQ.m:

Calculate and plot Duration v Amplitude, and Maximum Saccadic Velocity vs Amplitude.

NAFLOOP.m, NAFLOOP.m:

Iterative solution of tau matrix for NAF/NAFX. Find the tau for each variation in position/velocity limits that results in an NAFx (extended pos,vel limits) equivalent to the NAF obtained for the default pos,vel limit.

NAFX.m, NAFX_GUI.m:

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

PD2ANG.m:

Convert between PD and Angle for Small Angles (<30D).

Usage: out = pd2ang(pd_or_ang, inputValue, conv);

SACLENS.m:

SACLENS will plot the time length of the saccade versus the distance the eye travelled from the end of the saccade to the peak of the slow phase.

SAC2.m:

SAC2 will plot the time length of the saccade versus the position, velocity and acceleration at user-selectable times before the onset of the saccades.

SHOWME.m, DRAWME.m:

Plot an eye position record and show the points at times before the saccade onset. Use PICKDATA to select exactly one record to be displayed. Then

type SHOWME, which invokes DRAWME at each timestep you wish to view.

SNEL2LOG.m:

Convert between LogMAR and Snellen acuities.

skewU.m:

Calculate the skewness of a list of saccades.

TAUTEST.m;

See how well a tau matrix works when plugged into the NAF/NAFP for a given eye movement data file.

TPLOT3.m:

Phase plane display of time progression of eye data with respect to time of saccade onset. Use this function once you have created a file of control point information for the data file you wish to analyze. (The control points include saccade onset, saccade offset and peak of the slow phase. Of these, only saccade onset is used in TPLOT3.) Example: You must call TPLOT3 with position, velocity and acceleration data -- TPLOT3(pos, vel, acc)

TBUTT3.m, UPDATE3.m:

TBUTT is invoked by TPLOT3 to handle most of the control button scripts.

UPDATE3 is the companion to TPLOT3 and is called to update the phase plane as the time before saccade onset is changed.

VA2NAF.m:

Convert between Snellen acuity and NAF.

VA2SNEL.m:

Convert between Visual Acuity decimal and Snellen fraction.

VELPLOT.m:

VELPLOT will plot the difference in velocity just before and just after a braking saccade fires. This is plotted against the size of the braking saccade.

WHENSLIP.m, WHENFOV.m:

Finds the times when eye velocity is just inside the velocity slip/foveation radius. Calculates the average time and plots a histogram.

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.m:

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

drawCirc.m:

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

drawRad.m:

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

FOVRAD.m, SLIPRAD.m:

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

FSBOX.m, FSACUBE.m:

Draw a box enclosing the foveal and slip radii (FSBOX) on a 2-D phase plane. Draw a 3-D cube enclosing the foveal, slip and acceleration radii (FSACUBE) on the current 3-D phase plane.

LA.m:

Draw a text label on a plot.

In the DATADIRS directory:

OMPATH.m:

Adds all the paths that OMtools needs.

omdir.m:

Go to the OMtools directory.

DATADIR.m:

Go to the root data directory.

SETROOT.m:

Read DATAROOT.TXT and set the root data directory to the location stored in it.

DATAROOT.TXT:

Holds the location of the root data directory.

In the ZOOMTOOL directory:

<<There are around seventy files in this directory. If you are not modifying ZOOMTOOL, the main tool in this directory, you don't need to know about any of the others. If you are really interested, read the ZOOMTOOL documentation.>>

ZOOMTOOL.m:

This is an interactive graphics display. You have two sets of cursors and some relatively self-explanatory control buttons that let you move through the displayed data. You first have MATLAB plot a 2-D graph and then you type "ZOOMTOOL". ZOOMTOOL can be used to pick out control points, or to pick bias/calibration adjustment values. Really, it has hundreds of uses!