TimeStudio process manual – Static AOIs analysis

Author: Pär Nyström
Last revision: 2014-02-11 (Peter Johnsson)

Innehåll

1. Purpose

This manual contains step-by-step instructions of how to analyze looking time data from static areas of interest (AOIs) in the TimeStudio framework. The manual is written for novel users of TimeStudio.

This manual assumes that you have downloaded the latest TimeStudio core and the data files mentioned below. If you do not know how to start TimeStudio, please double check that you have followed the steps described in http://timestudioproject.com/getting-started

2. Definitions

Buttons that need to be activated in order to execute a command or fields within the workspace are indicated with quotation marks in the text below. Plugin names are marked with bold. Letters in red circles in figures and text indicate where in TimeStudio a particular action takes place.

3. Dataset

This manual covers how to set up a workflow to analyze static aois in eye tracking studies. In the manual some demo datasets are used to illustrate how and why each step is necessary.
The files needed to follow this manual are:

static_aoi_subj1.txt
static_aoi_subj2.txt
static_aoi_subj3.txt
static_aoi_subj4.txt
static_aoi_subj5.txt

Each file is a recording of one test subject and the files can be downloaded from

http://www.timestudioproject.com/manuals/static_aoi_manual_data files.zip
The data is taken from a large eye tracking study in which several different stimuli was presented to young infants. In this tutorial we will describe how to analyze looking time data from two simultaneously presented movie clips (a preferential looking design). The subjects were 10 month old infants viewing two animated point light walkers that were placed on the left and the right side of the screen. One of the point light walkers was turned upside down (randomized location across trials) and the study investigated whether the infants preferred to look at the upright or inverted walker. An example of the stimuli is also found in the zip file with the data files (BM_s1v1_upL_xvid.mpg). This video is also found online at:
http://timestudioproject.com/manuals/BM_s1v1_upL_xvid.mpg

This study is a replication of Fox and McDaniel (1982) demonstrating that young infants are sensitive to biological motion and prefers to look at up right to inverted point-light humans:
”When a small number of lights are placed on the limbs and joints of a moving human (or

animal), the motions of the lights (biological motion) are sufficient to enable adult observers to perceive immediately the activity of the human. This perception of biological motion has been hypothesized to be an intrinsic capacity of the visual system. The results of this experiment, which demonstrate that infants 4 to 6 months of age exhibit a preference for biological motion patterns, support that hypothesis." (Fox & McDaniel, 1982)

Figure 1. Screenshot of stimuli, upright point light walker to the left, inverted point light walker to the right.

4. Overview

This tutorial demonstrates how TimeStudio can be used to analyze looking times in AOIs using the example specified in section 3. In order to complete the analysis a series of operations need to be performed in TimeStudio.

- Launch MATLAB and TimeStudio
- Add data and define participants
- Read added data files
- Clear existing AOI definitions from memory
- Add new AOI definitions to memory
- Remove trials with insufficient looking time
- Extraction and visualization of looking times
- Calculation and visualization of difference scores between AOIs

When this tutorial is complete the workflow and results windows should look identical to Figure 2, replicating the original findings by Fox and McDaniel (1982).

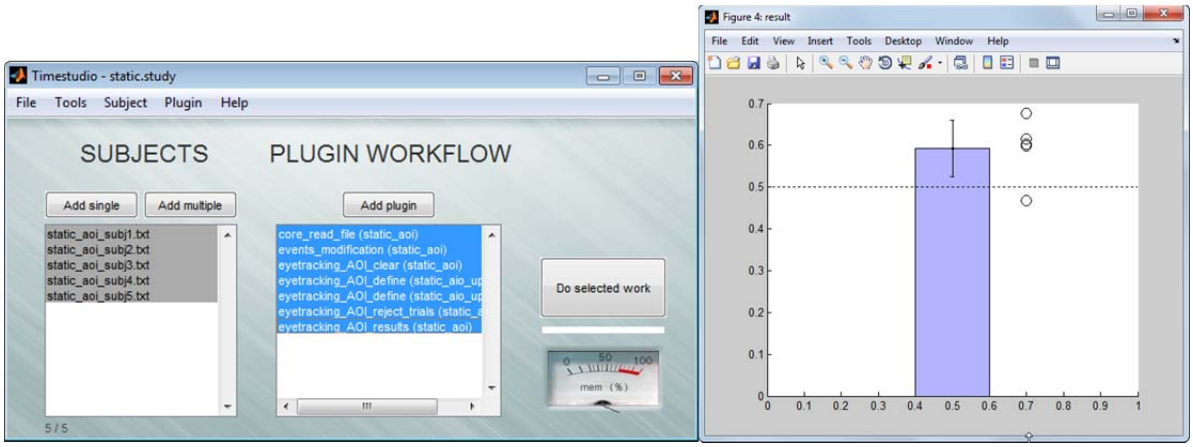

Figure 2. The final state of the workflow and graph demonstrating difference scores between AOIs.

5. Step-by-step instructions

5.1 Launch MATLAB and TimeStudio

1. Start by launching MATLAB.

2. Either add the TimeStudio folder to the MATLAB path, or move into the TimeStudio folder.

3. Start TimeStudio by typing ‘TimeStudio’ in the MATLAB command window and press enter.

When TimeStudio starts the main window is displayed as seen in Figure 3.
The empty field to the left (Figure 3A) is called “subjects” or “list of subjects” and the empty field to the right (Figure 3B) is called “plugin workflow”.

5.2 Add data and create subjects

In order to work with the data that is stored in the data files you need to load them into the Matlab memory. However, before you can load them you must tell TimeStudio where to find them. TimeStudio keeps track of data files by assigning them to subjects in the subject list.

1. Press the “Add multiple subjects” button (Figure 3C) in the main TimeStudio window. This will open a file browser window.

2. Locate the five data files that belong to this demonstration, they should be named “static_aoi_subjx.txt” with x being 1 to 5.

3. In the pop-up file browser window, mark all five files and press the “Open”-button to add all five files and create an individual subject for each file.

This should add five subjects to the subject list (Figure 4A).

Tip1: If each subject have multiple data files, press the ”Add single subject” button and select the data files for one subject. A new subject is then created which contain references to all selected data files.

Tip2: Double click a subject in the subject list! You will then be able to change the name, group, and other properties of the subject. You can also see which data files are assigned to this subject!

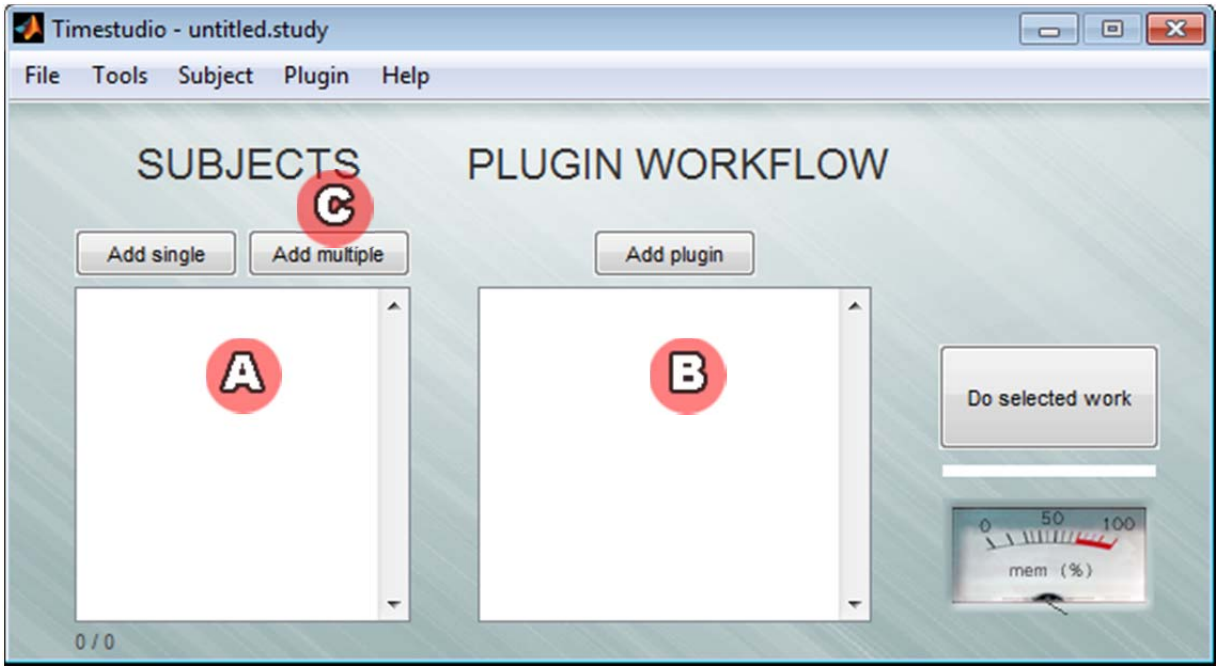

Figure 3. The main window at start up.

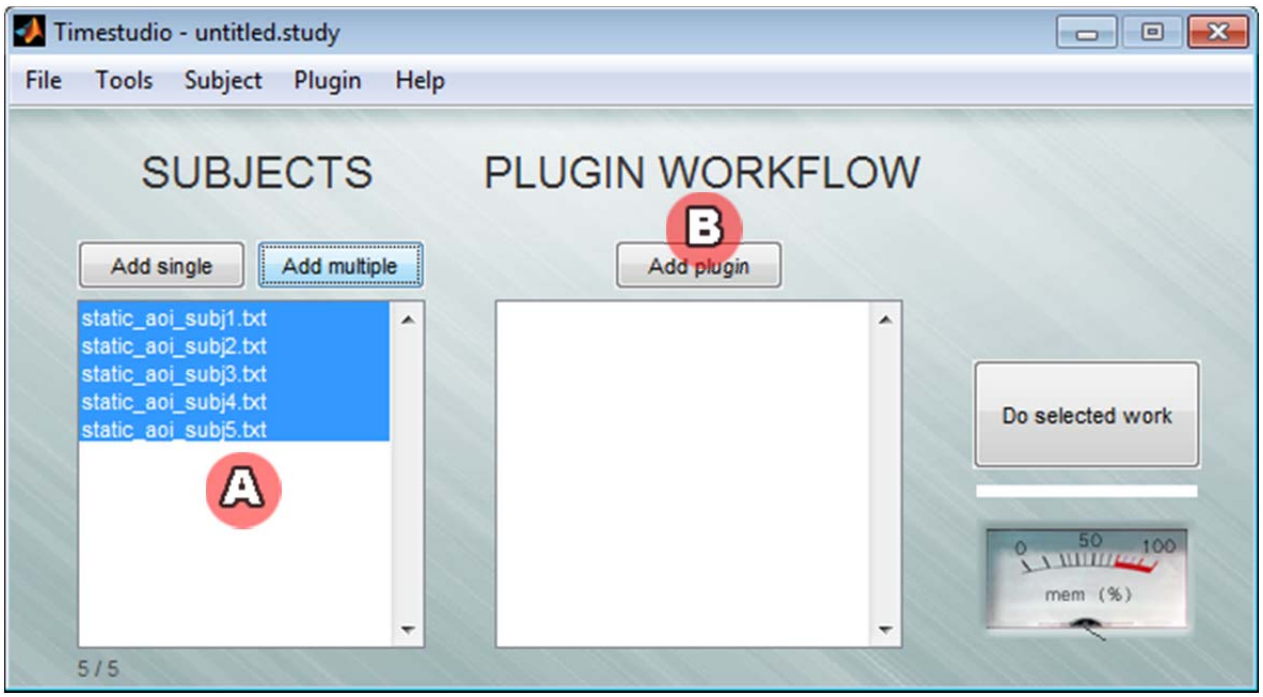

Figure 4. The main window after subjects have been created.

5.3 Read data files into memory

To extract relevant data we need to read the actual gaze data from the data files. To load the data files into memory you need to parse the data files and extract relevant data (such as x and y position of gaze, pupil size, stimuli events etc.). This is the purpose of the first plugin in our workflow. Later we will add other plugins as a chain to perform the whole analysis, but the first plugin is usually one that reads a data file. In our case it will be the plugin called **core_read_file**.

1. Press the "Add plugin" button (Figure 4B) in the TimeStudio main window to bring up the pop up menu for selecting and adding a plugin to the plugin workflow. Hover to "core" and select **core_read_file** from the pop-out menu. This should bring up the window seen in Figure 5.

2. Start with adding a data field. Above A, in Figure 5, enter 'eyetracking'.

3. The next step is to add variables. They are entered by typing the variable name in the "Add variable"-box that is below the A, and pressing enter. In this way enter the variables 'x',' y', and 'pupil', one by one.

As can be seen in the list box B, there are two predefined variables – 'events' and 'time'. These are used and required by almost all plugins and are therefore added by default.

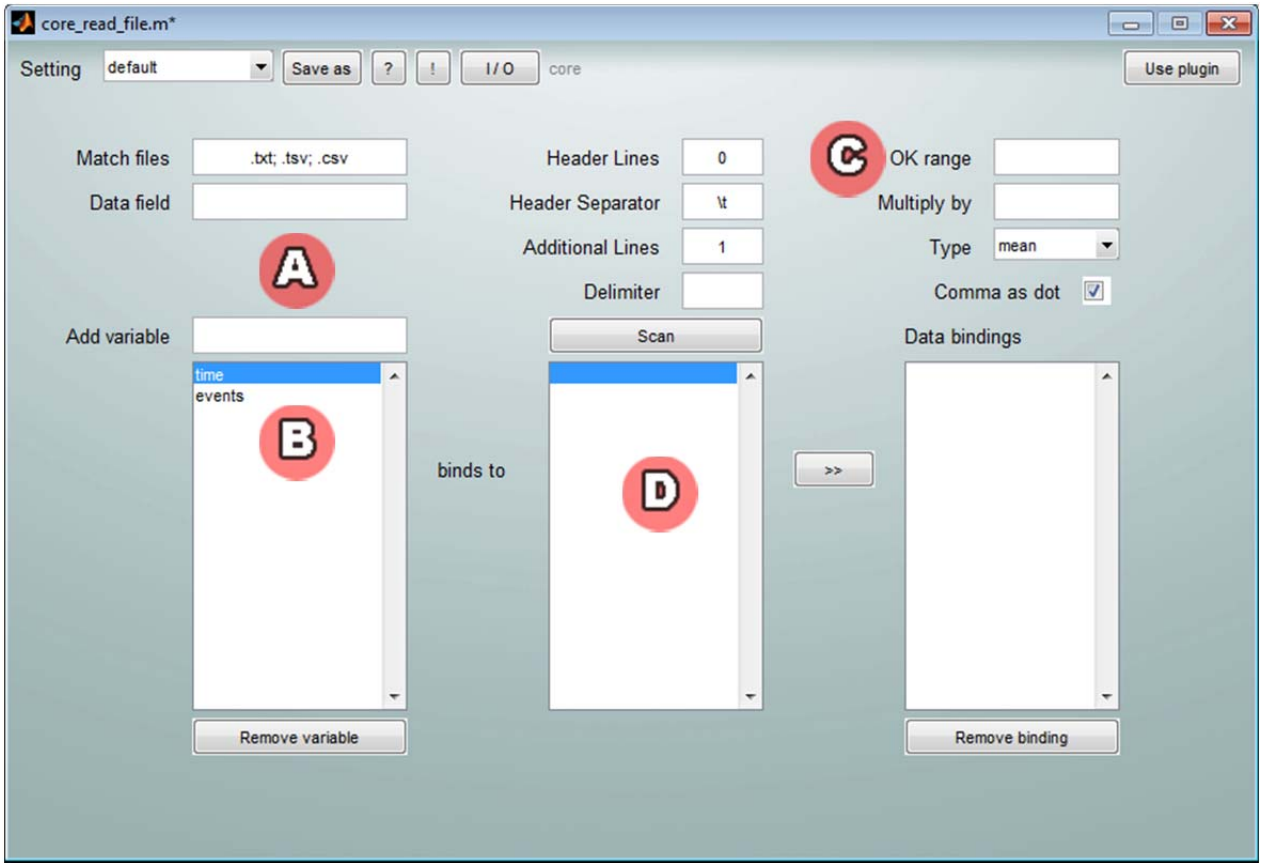

Figure 5. The core_read_file window.

4. After this, the number of header lines in the files to be read should be entered in the "Header Lines"-box (C in Figure 5). In our case, enter '18' in this box, and the

properties (column names, that is) from our input files will be listed in the listbox at D in Figure 5.

5. This enables us to bind our variables to different data columns in the input files. Select ‘x’ in the variable listbox (B in Figure 5), then select ‘GazepointX’ in the properties listbox (D) and finally press the “>>”-button to the left of D the figure. This will create a binding that will be displayed in the “Data bindings”-listbox.

The results so far can be seen in Figure 6.

6. Continue with binding the variable ‘y’ to ‘GazepointY’ in the same way.

7. Next up is to bind ‘time’ to ‘Timestamp’. This is done in the same way as for the other variables except that ‘0.001’ should be entered into the “Multiply by”-box (next to A in Figure 6). This means that the values of Timestamp are multiplied by 0.001, which gives is the time in seconds in this example.

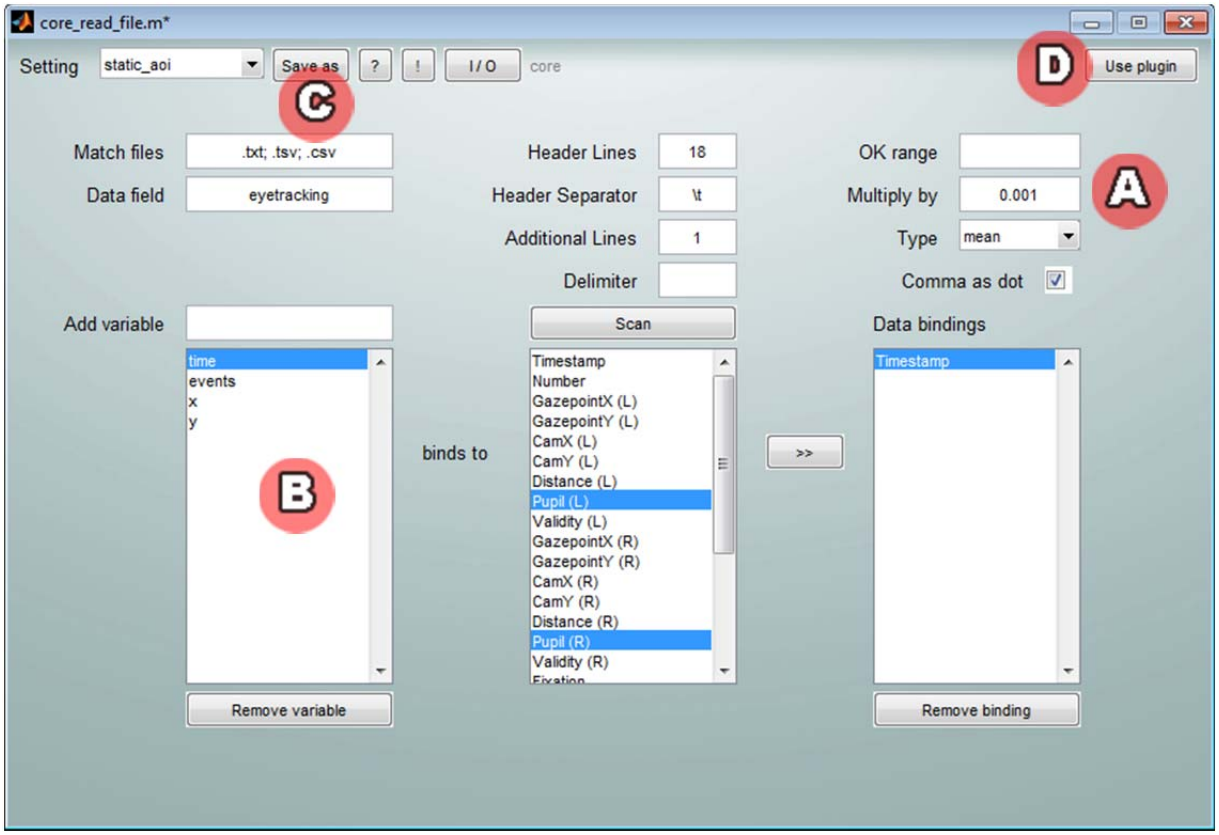

Figure 6. Adding bindings in core_read_file.

8. As the final binding we should bind ‘events’ with ‘Description’. Now the “Type” should be set to ‘event’.

9. The final steps are to set the acceptable range for some of the variables. Select ‘x’ in the variable-listbox (at B in Figure 6), and then enter ‘0 1024’ in the “OK range”-box (close to A in Figure 6).

10. Next, select ‘y’ and set the ok range to ‘0 768’, and finish by selecting ‘pupil’ and

setting the range to ‘2 10’.

Now that we have made all the settings required to read our data files we need to save them before using them.

11. Press “Save as” (above C in Figure 6). In the pop-up window enter some sensible name (e.g. ‘static_aoi’), then press “Save”.

12. Press “Use plugin” (at D in Figure 6) to close the settings window and add this plugin with these settings to the plugin workflow.

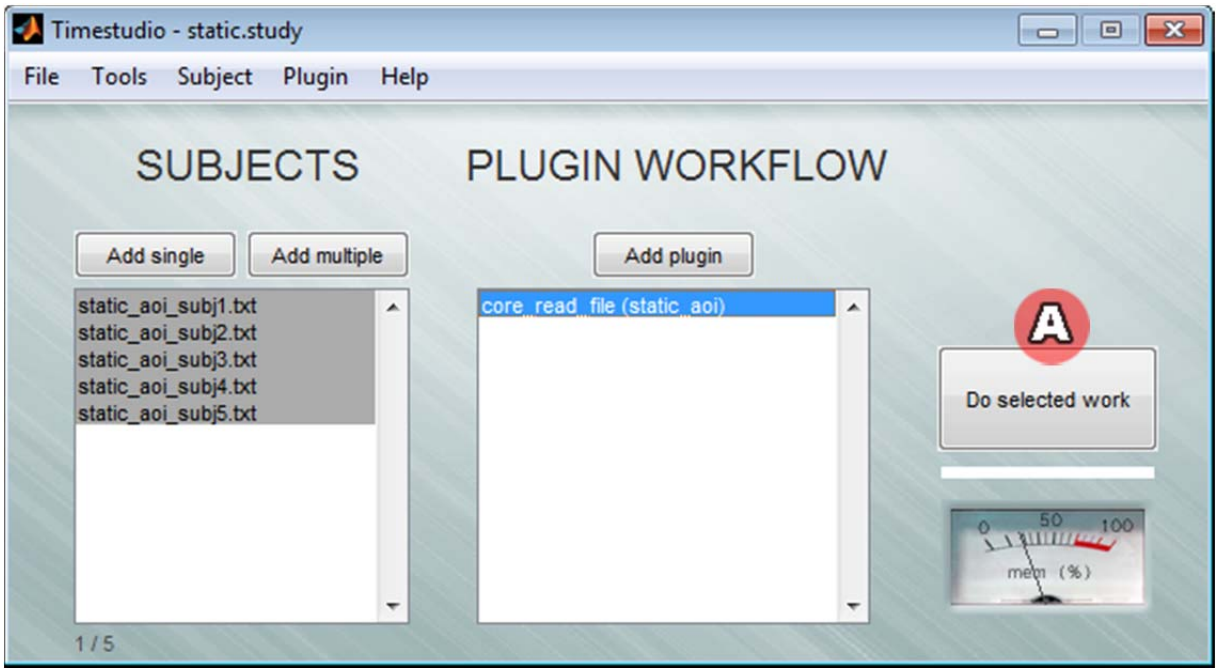

Figure 7. The main window after the first plugin was added.

Back in the main window:

1. Select all five subjects in the subject field, and select **core_read_file** in the plugin workflow (as can be seen in Figure 7).

2. Press the “Do selected work” button (Figure 7A) to read the data from the added files into the subjects.

While TimeStudio is processing a wait bar appears, and additional information is printed in the MATLAB command window.

5.4 Changing the names of events in the dataset

After loading the data into memory it is possible to find events in the datastream that represent when the stimuli was presented to the subjects. Only a few of these events are relevant for our analysis. We will now add a plugin that can rename these events and thereby categorize them into larger units. In our example we have presented four videos with the same video stimuli; two videos with the upright walker on the left side (called BM_s1v1_upL_xvid.mov-@:PointLights: and BM_v1s1_upL_xvid.mov-@:PointLights:),

and two videos with the upright walker on the right side (called BM_s2v2_upR_xvid.mov-@:PointLights: and BM_v2s2_upR_xvid.mov@:PointLights:). We will now group them into events with the more comprehensible names upLeft and upRight, and this can be done with the plugin **events_modification**.

1. As before, press “Add plugin” from the main menu to bring up the pop up for selecting plugins, press “events” and select **events_modification**. This will open up the settings window for the plugin, see Figure 8.

2. Browse (Figure 8A) for the name of the data field to use. The only field visible should be ‘eyetracking’. Select ‘eyetracking’ and press “done”.

3. Then press the “Scan event names” button (Figure 8B), and the previously large empty field named “Event List” (Figure 8C) should now be filled with different names for events that are present in the data set.

4. There are many more events than we need to change, so delete the lines with events that are not of interrest and make sure that the list looks like in Figure 8. The text should be tab separated, and make sure that you have added a new column called “name”.

5. When your list looks OK, press the “Save as” button and save the setting as ‘static_aoi’.

6. Press “Use plugin” to add the plugin to the “Plugin workflow” in the TimeStudio main window.

Back in the main window (Figure 9)

7. Select all subjects and the **events_modification** plugin, then press “Do selected work” button to apply the modifications to the events.

> Tip 4: Since the text is tab separated you can copy the list and paste it into Excel or most other spreadsheet programs. The values will then be aligned in rows and columns and can be easily changed. When you are done you can select all in your spreadsheet and copy it back into the TimeStudio plugin window!

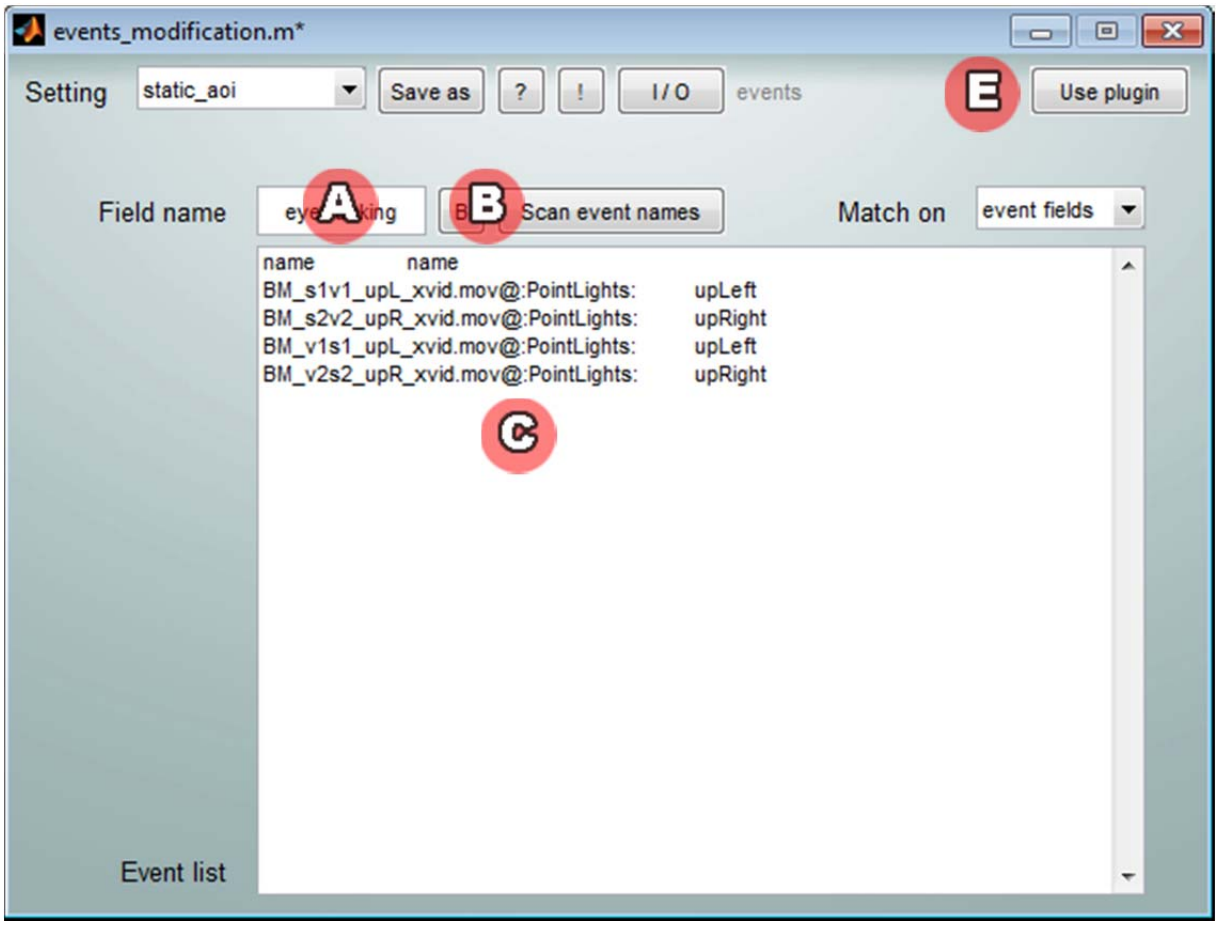

Figure 8. The events_modification plugin which can rename events.

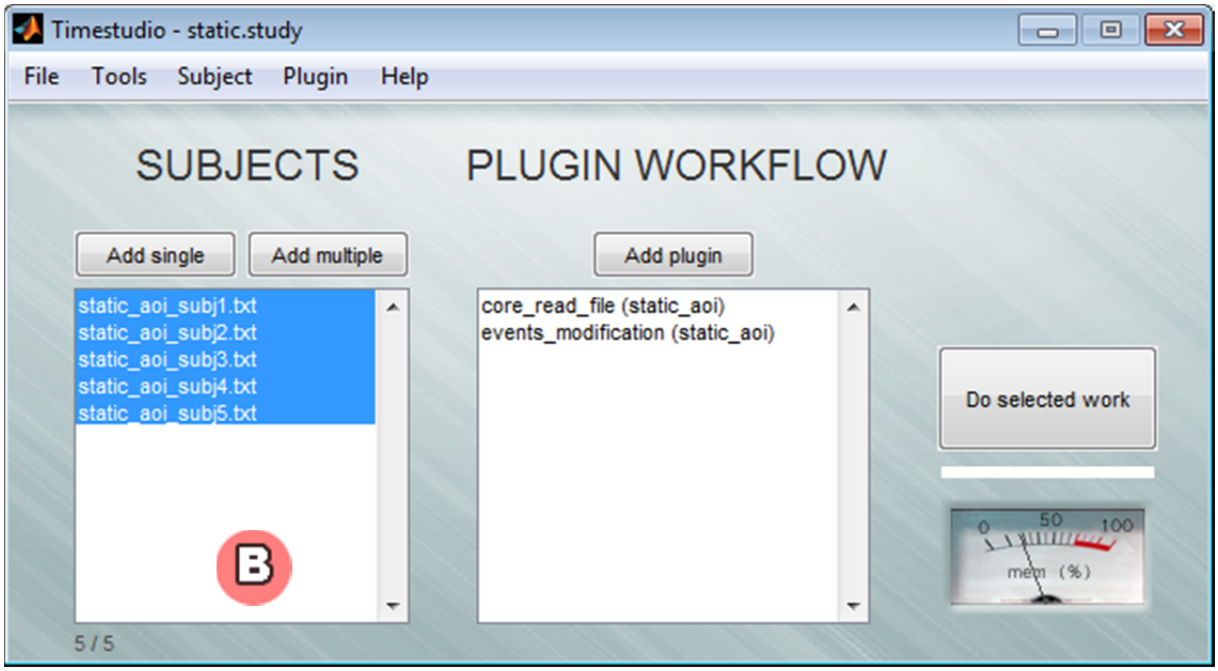

Figure 9. The main window after adding events_modification

Your subjects will now have events that are called upLeft and upRight that keeps information on the stimuli was presented.

Until now we have only made sure that we have data to work with in memory and adjusted so that we have events that represent our different videos. We will now continue with working with areas of interest.

5.5 Cleaning up before analysis

In eye tracking studies it is common to define interesting areas on a monitor. These areas are called areas of interest (AOIs), and it is possible to define multiple AOIs in TimeStudio with a variety of behaviors. When areas are defined they will stay in memory until they are explicitly cleared by using the plugin **eyetracking_AOI_clear**. If you only run the whole workflow once you do not need to clear the AOIs, but it is good practice to always include **eyetracking_AOI_clear** before your AOI definitions. That way you do not risk that any old AOI definitions remain in memory.

1. Press “Add plugin” from the main menu to bring up the pop up for selecting plugins, press “eyetracking” and select **eyetracking_AOI_clear** (Figure 10).

2. This plugins has no parameters to set. Save it as ‘static_aoi’ and click “Use plugin” to add the plugin to the workflow in the main window.

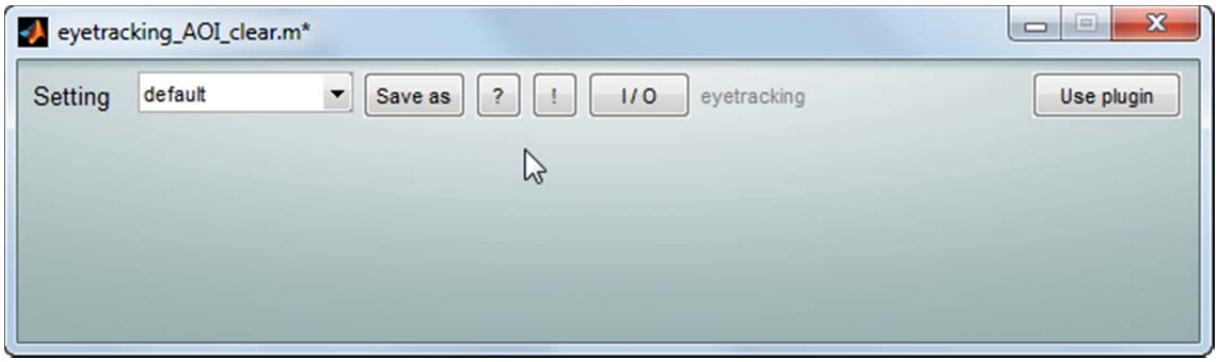

Figure 10. The eyetracking_AOI_clear plugin which will clear existing AOI definitions.

5.6 Defining Areas Of Interest for the event upRight

To define AOIs the plugin **eyetracking_AOI_define** should be used.

1. In the usual way, press “Add plugin” from the main menu to bring up the pop up for selecting plugins, press “eyetracking” and select **eyetracking_AOI_define**.

This plugin will open two windows: one window with the plugin settings Figure 11) and a preview window (not shown here). The **eyetracking_AOI_define** is one of the most complex plugins in TimeStudio and it can be used in many different contexts. We will describe the most common way to define static AOIs here (please see the manual about dynamic AOIs for defining more complex behaviors).

The first step when defining AOIs is to select the time periods when the AOIs should be applied to the gaze data. This is equal to splitting up the gaze time series into trials with a given starting time and ending time relative to specific events.

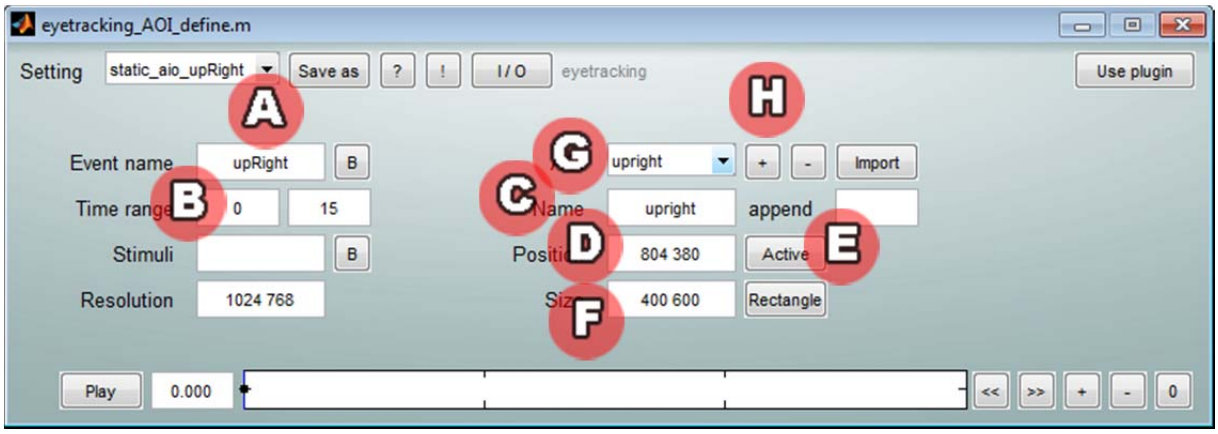

Figure 11. The eyetracking_AOI_define setting window

2. To select the events, press Figure 11A and choose the event name called ‘upRight’. Next, set the starting and ending time (Figure 11B) of the trials in relation to the event time. In this example we set the starting time to 0 and ending time to 15, giving us trials with a length of 15 seconds. It is possible to use both positive and negative values in the time range, but the ending time should always be larger than the starting time.

When the plugin window is opened a default AOI is already created. The properties of AOIs can be set in the right of the plugin window.

3. Change the name of the default AOI to ‘upright’ by changing the name in Figure 11C and pressing the enter key.

4. Change the position of the AOI by changing the field in Figure 11D. The position consists of two values with a white space in between. The first value is the x position of the center of the AOI, and the second value is the y position. When you press the enter key you should see that the preview window changes the position of a red rectangle which is the AOI that we are changing. In this example we have used the position 804 380.

5. The red color indicates that the AOI is inactive at this time point. Make the AOI active by pressing Figure 11E. You can press this button several times to toggle between active and inactive mode, but make sure that you leave it in active mode with a yellow color of the AOI.

6. Set the size of the AOI by setting the values in Figure 11F. The first number is the total width of the AOI and the second number is the total height of the AOI. In this example we have used the size 400 600.

Now the first AOI is defined correctly.

7. Add another AOI by pressing “+” in Figure 11H. You can choose which AOI to modify by selecting the wanted AOI in the dropbox Figure 11G. Name the new AOI ‘inverted’ and give it the position 230, 380 and size 400, 600. Make sure it is active.

Now the second AOI is defined correctly.

8. Add another AOI by pressing Figure 11H. Name it ‘screen’ and give it the position 512, 384 and size 1024, 768. Make sure it is active.

9. Save the settings by clicking the button “Save as” in the top row of the plugin window. Save the settings as ‘static_aois_upRight’. Add the plugin with these settings to the workflow by pressing “Use plugin”.

> Tip: When you save a setting you cannot use white spaces or arithmetic operators such as the minus sign. However, it is OK to use underscore!

5.7 Defining Areas Of Interest for the event upLeft

To define AOIs for the other event, when the upright pointlight walker is presented on the left side of the screen, we add another instance of the **eyetracking_AOI_define.**

1. Press “Add plugin” from the main menu to bring up the pop up for selecting plugins, press “eyetracking” and select **eyetracking_AOI_define**.

2. When the plugin window appears, select the ‘static_aois_upRight’-setting in Figure 12A. This will bring up the settings we just defined for the “upRight” event. To clone the settings press Figure 12B and save as ‘static_aois_upLeft’. We can now change the values of just a few fields.

3. Start by changing the event name in Figure 12C by pressing the “B” button and selecting ‘upLeft’.

4. Then change the name of the “upright” AOI to “inverted” and the “inverted” AOI to “upright”. In practice this swaps the positions of the AOIS. You could also do this by changing the position for the AOIs.

5. Add the plugin with these settings by pressing “Use plugin” (Figure 12E).

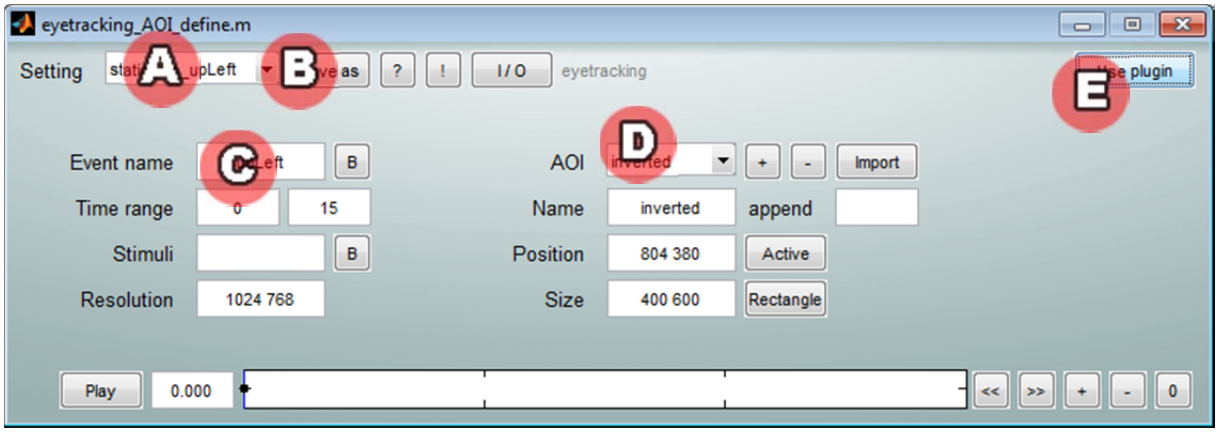

Figure 12. Cloning the settings to make minor changes.

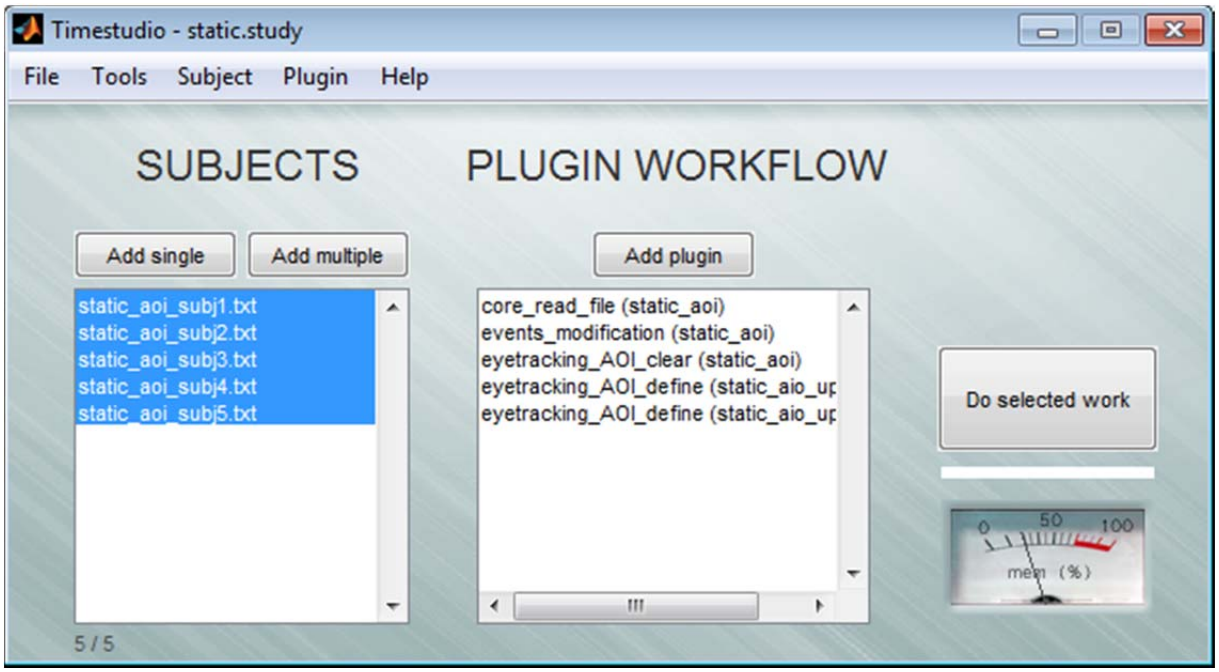

Figure 13. The main window after eyetracking_AOI_clear and two eyetracking_AOI_define has been added.

Now the main window should look as in Figure 13.

5.8 Rejecting bad trials

Some of the trials may contain bad data or perhaps no data at all. To reject specific trials there is a plugin called **eyetracking_AOI_reject_trials**. This plugin can be used after the AOIs have been defined. In this example we want to reject trials with less than 25% looking time at the screen (during the 15 seconds that the trials last).

1. Add the plugin **eyetracking_AOI_reject_trials** so that the plugin window is opened.

2. Press the “B”-button in the “Aois to check”-box at Figure 14A, and select ‘screen’.

3. Enter ‘25%’ in the field Figure 14B.

4. Save the setting as ‘static_aoi’ by pressing “Save as” and press “Use plugin” to add the plugin to the workflow.

In the main window,

5. Select the last three plugins in the workflow (AOI definitions and rejection) and press “Do selected work” (Figure 15).

TimeStudio will now calculate the gaze behavior in the AOIs and reject trials with less looking time than 25%.

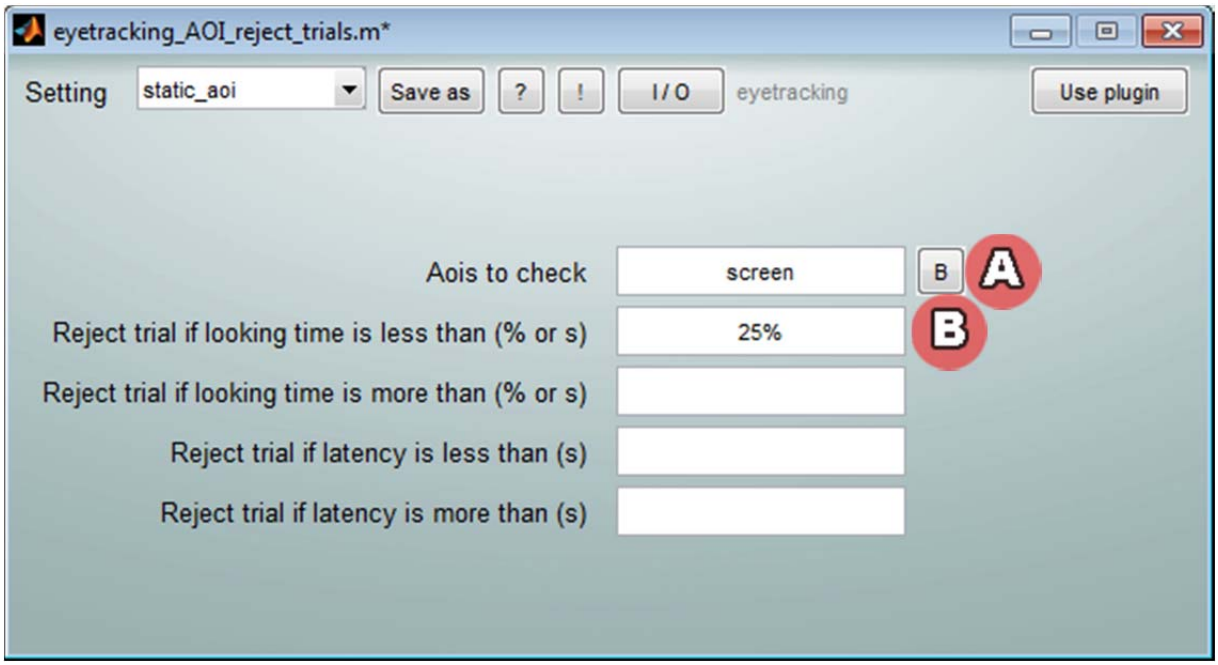

Figure 14. The eyetracking_AOI_reject_trials plugin that will reject trials that have too little looking time.

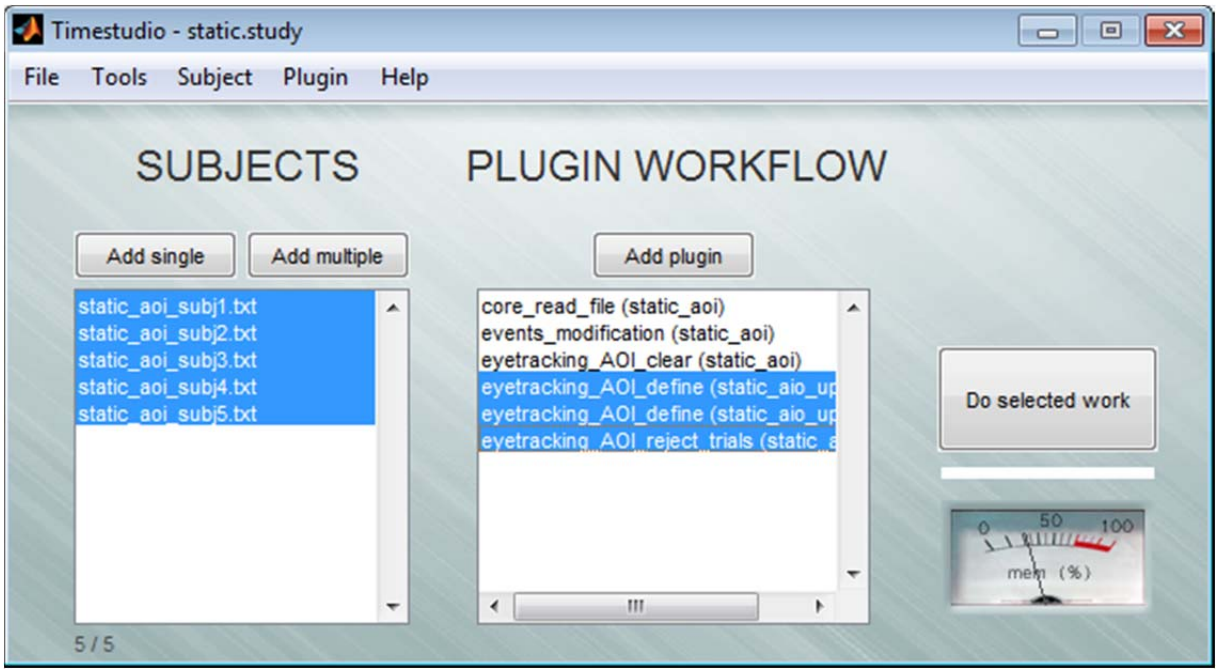

Figure 15. The main window after eyetracking_AOI_reject has been added.

5.9 Getting the results

When the AOIs are defined and we have calculated the gaze behavior in the AOIs it is possible to extract the relevant results. The actual measure depend on the context and may include looking times in different AOIs, latency to first visit in an AOI, the number of visits in AOIs etc. All these measures are available to extract in the plugin **eyetracking_AOI_results**, but in this example we will only calculate the mean looking time in seconds.

1. Open the plugin window by clicking “Add plugin” and select

eyetracking_AOI_results. The plugin window should look like Figure 16.

It is possible to get the results only for specific AOIs.

2. Click button Figure 16A and select the AOIs that are called ‘inverted’ and ‘upright’.

It is also possible to make a calculation over all trials within a subject (mean / median / min / max / range). This value is used for plotting the results, and will appear in a results text field later.

3. Select ‘Mean of trials’ in dropdown Figure 16B.

4. Then select what kind of measure we want to use for every trial (looking time / latency / visits etc.) in dropdown Figure 16C. In this example we use ‘AOI looking time (sec)’ which tells the plugin to extract the absolute looking time within each trial (our trial length is 15 seconds, so max value in each trial is 15).

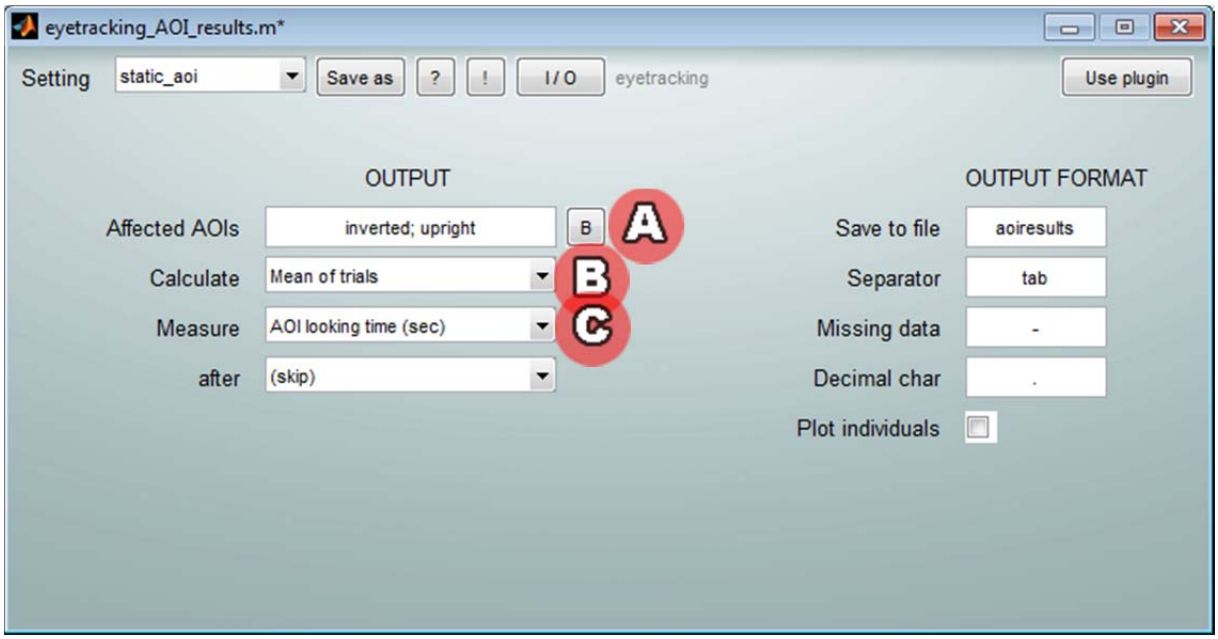

Figure 16. The eyetracking_AOI_results plugin which extracts the relevant data from your AOIs.

5. Save the settings as usual by pressing “Save as” and give it the name ‘static_aoi’. Then add the plugin to the workflow by pressing “Use plugin”.

Back in the main window (Figure 17),

6. Run the results plugin by having all subjects in the subject list selected and only the **eyetracking_AOI_results** plugin selected in the workflow list. Then press “Do selected work” to calculate the results.

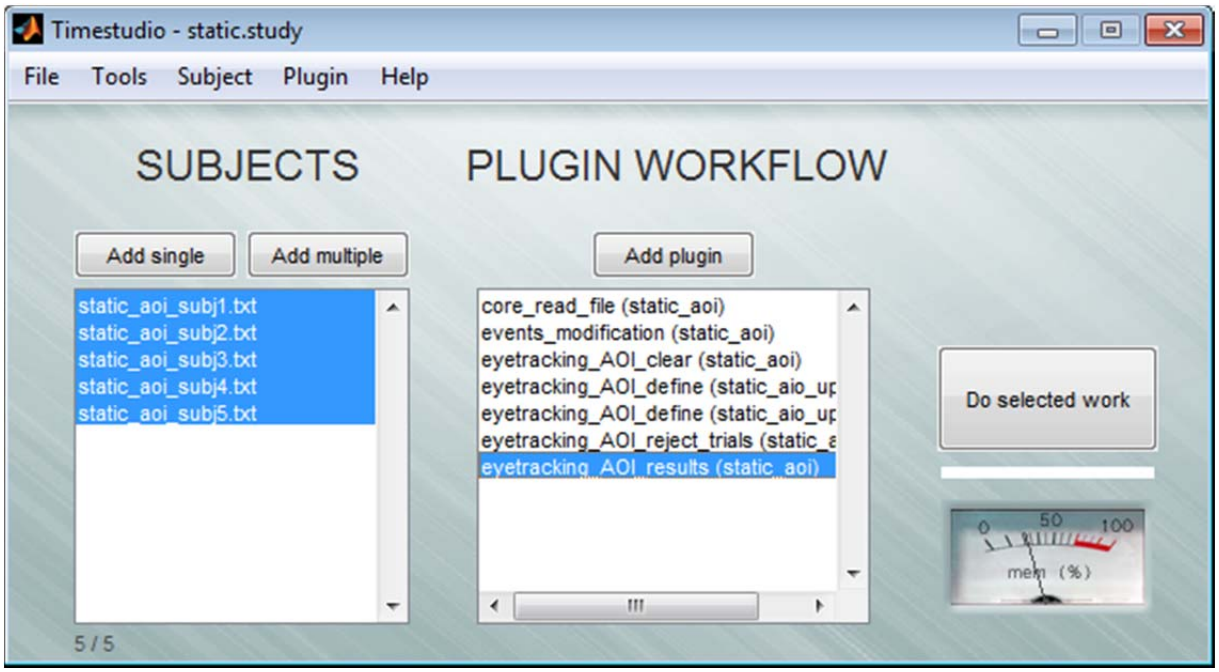

Figure 17. The main window after eyetracking_AOI_results is added.

The results plugin may take some time to finish, but when it is done it will create three new windows (Figure 18, Figure 19, and Figure 20). The top window (Figure 18) is a bar plot showing the mean looking time in each AOI. Error bars represent the standard error. Beside the error bars there are also circles representing the mean value of each individual subject. By looking at this plot you can get a feeling for differences and distributions of the AOI looking times. In our example we can see that the subjects prefer to look at the upright pointlight walker (mean looking time around 7 sec over all trials) rather than the inverted pointlight walker (mean looking time slightly less than 5 sec over all trials).

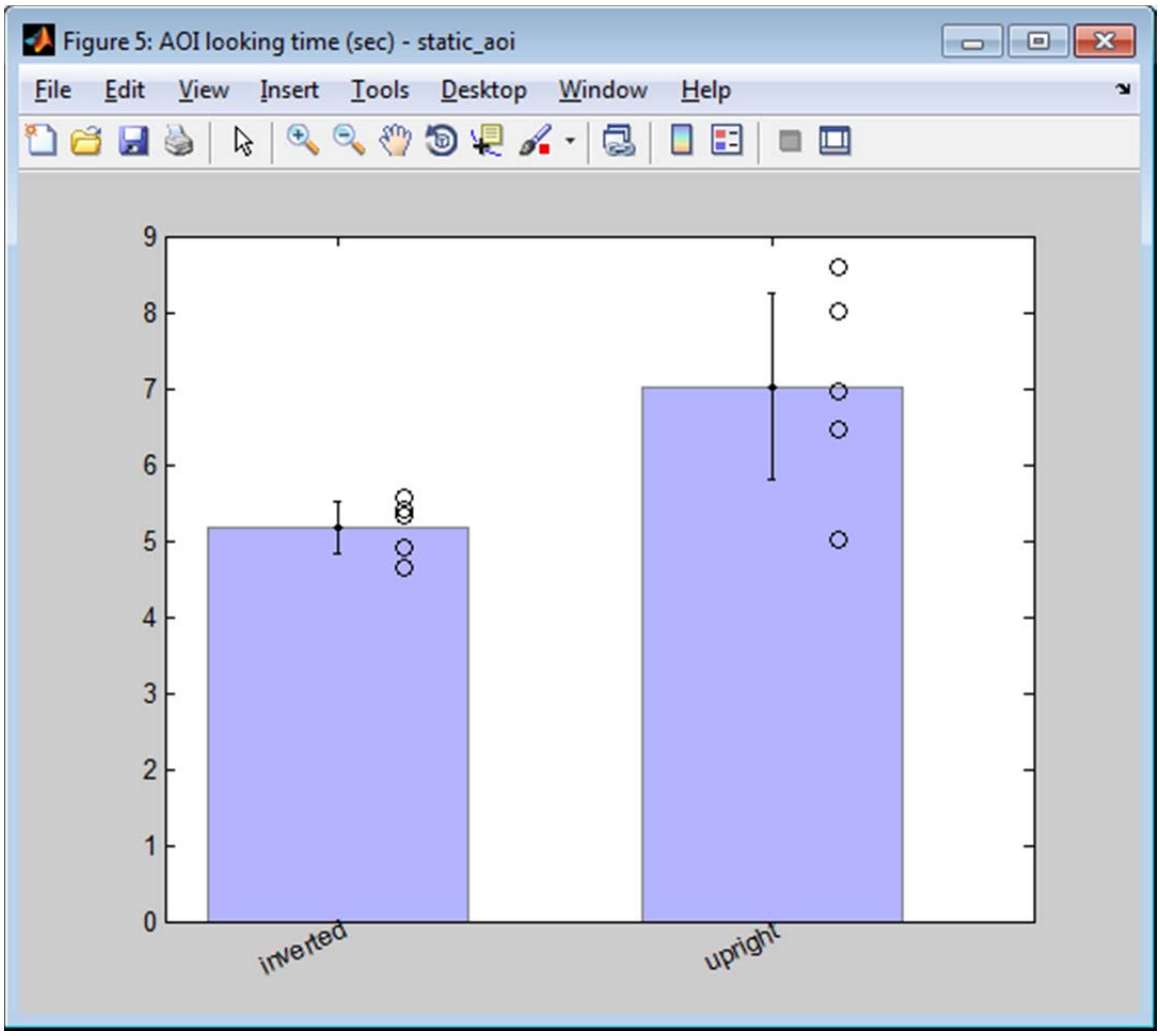

Figure 18. Results plot from eyetracking_AOI_results.

If you close the plot window or move it to the side you will see a window (Figure 19) with a text field with the data that was represented in the plot (together with the number of trials for each subject and AOI). You can click in this text field and copy the content by pressing Ctrl+A followed by Ctrl+C. You can now paste the data into a spreadsheet or statistical program such as Excel, Statistica, SPSS, etc., for further analysis of this data. However, to get the value for each individual trial you need to move or close this window to reveal the last results window (Figure 20).

The last result window contains the values for each AOI for every individual trial for every subject. Just like the previous window you can copy and paste it into a spreadsheet or statistical program by clicking in the text field and pressing Ctrl+A followed by Ctrl+C. You can then do further analysis of your data by removing outliers and doing the statistical test of your choice.

In many cases it is sufficient to stop here and calculate the statistics in a third party software. However, it is possible to go one step further and calculate difference scores or ratios between AOIs, which is shown in the next step.

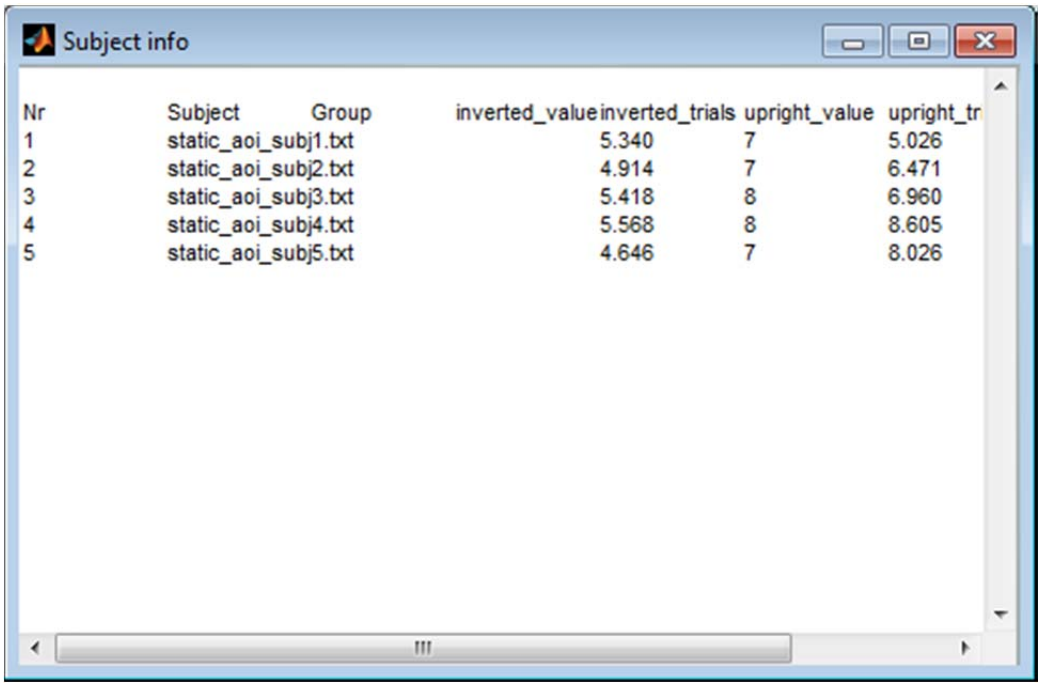

Figure 19. Subject summary from eyetracking_AOI_results.

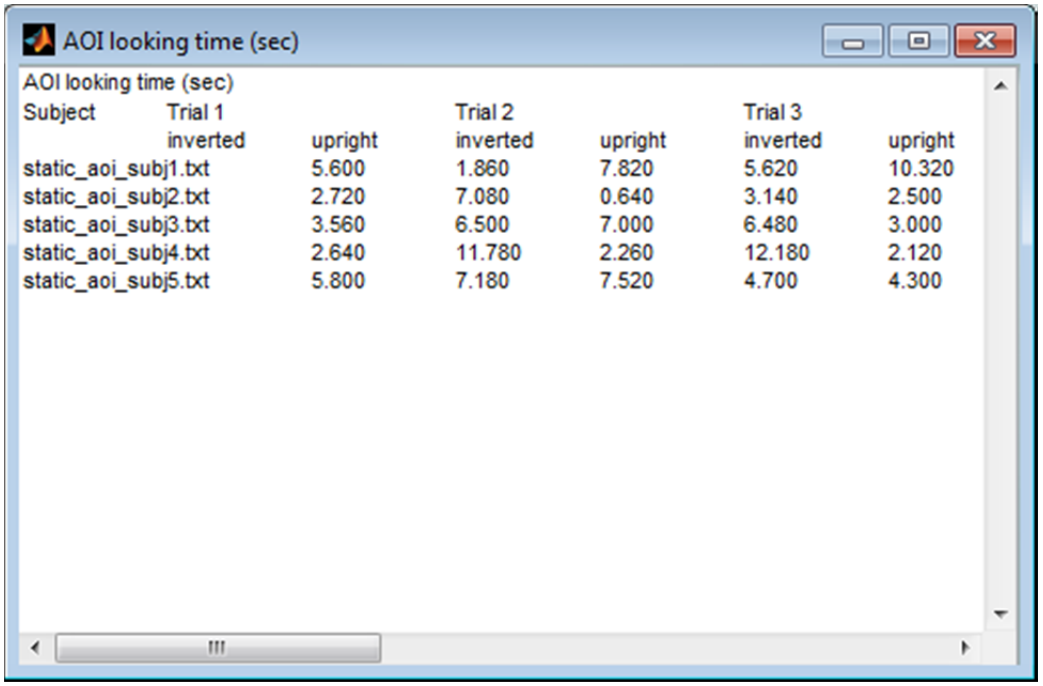

Figure 20. Trial information from eyetracking_AOI_results.

5.10 Creating ratios between AOIs

To calculate ratios or difference scores between AOIs in TimeStudio we use the **eyetracking_AOI_diffscores** plugin.

1. Open the plugin window (Figure 21) by clicking on “Add plugin” and adding it from the menu.

2. Press Figure 21A to select the nominator value. If several AOIs are selected then

the sum of the AOI values will be used. In this example we select the “upright” AOI to be the nominator.

3. Next, set Figure 21B to be “divided by” to get a ratio rather than a difference score. Then press Figure 21C and select “inverted” and “upright” to be the denominator. The last setting is Figure 21D, which should be set to “in individual trials” since both AOIs are present in the same trials.

> TIP: The order of plugins in your workflow may change the output results. Also, some plugins require data that comes from other plugins. For example, eyetracking_AOI_diffscores requires that the plugin eyetracking_AOI_results has been used before.

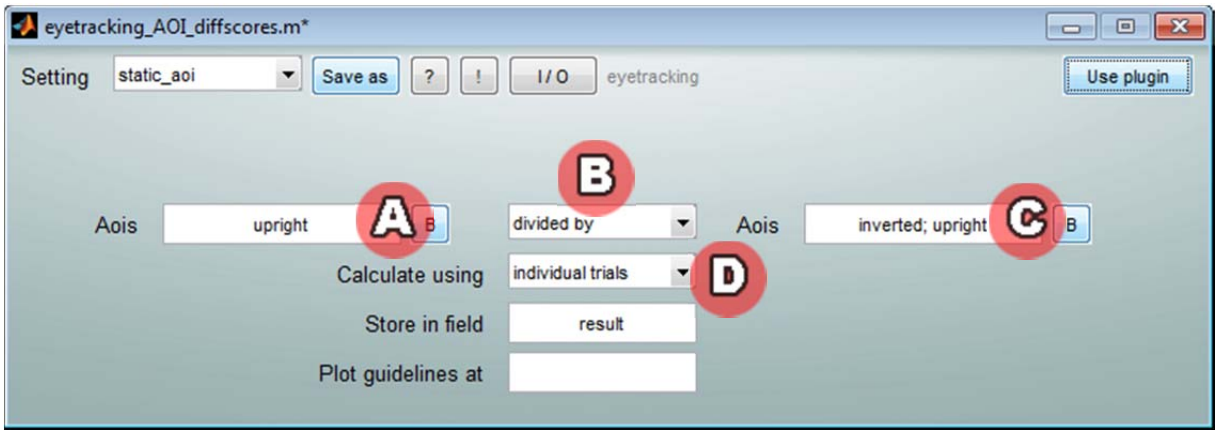

Figure 21. The eyetracking_AOI_diffscores plugin will calculate difference scores and ratios between AOIs or groups of AOIs.

4. Save the settings as ‘static_aois’ and add the plugin to the workflow by pressing “Use plugin”.

Back in the main window it is now possible to select all subjects and all plugins in the workflow to batch process the whole analysis. In case you want to add more subjects to the analysis you can just add them to the subject list and run the whole workflow again.

5. To run the whole workflow press “Do selected work” (Figure 21).

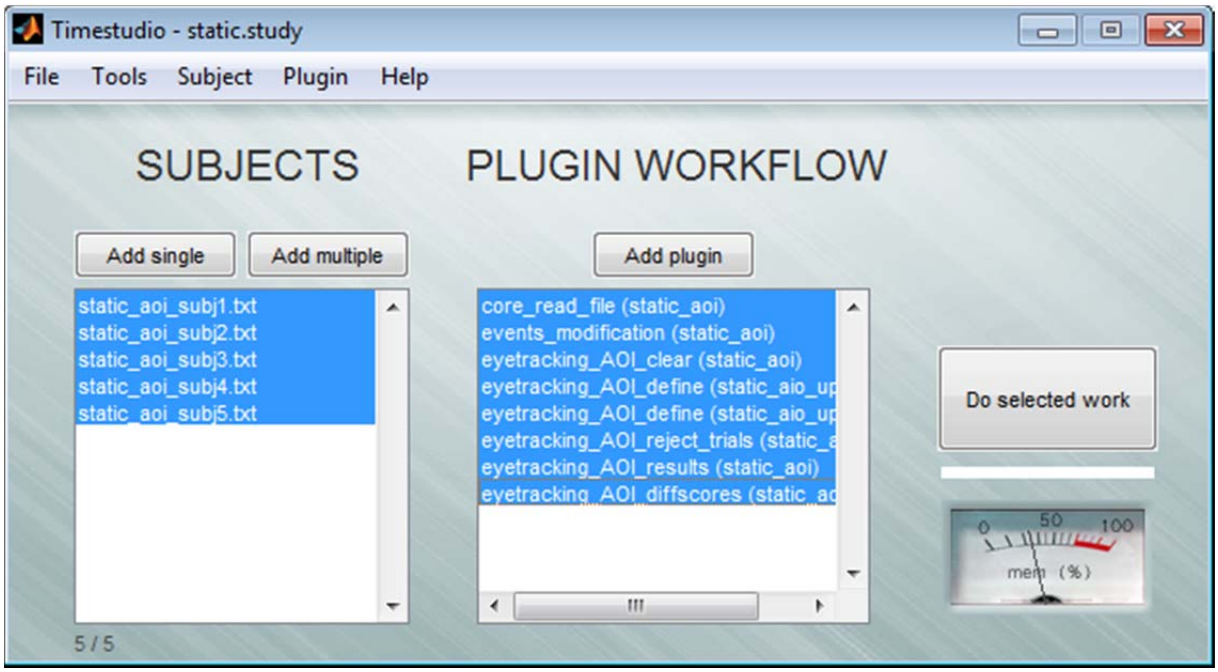

Figure 22. The main window when all plugins have been added.

After running the whole workflow a new plot for the ratios will be created by **eyetracking_AOI_diffscores** showing a bar with the mean ratio (Figure 23). The error bars are standard error and empty dots represent the ratio for each individual subject. The dotted line at 0.5 represent equal preference for upright and inverted point light displays, but since the error bars do not cross this line it is an indication that there is a significant difference between the looking times of the AOIs.

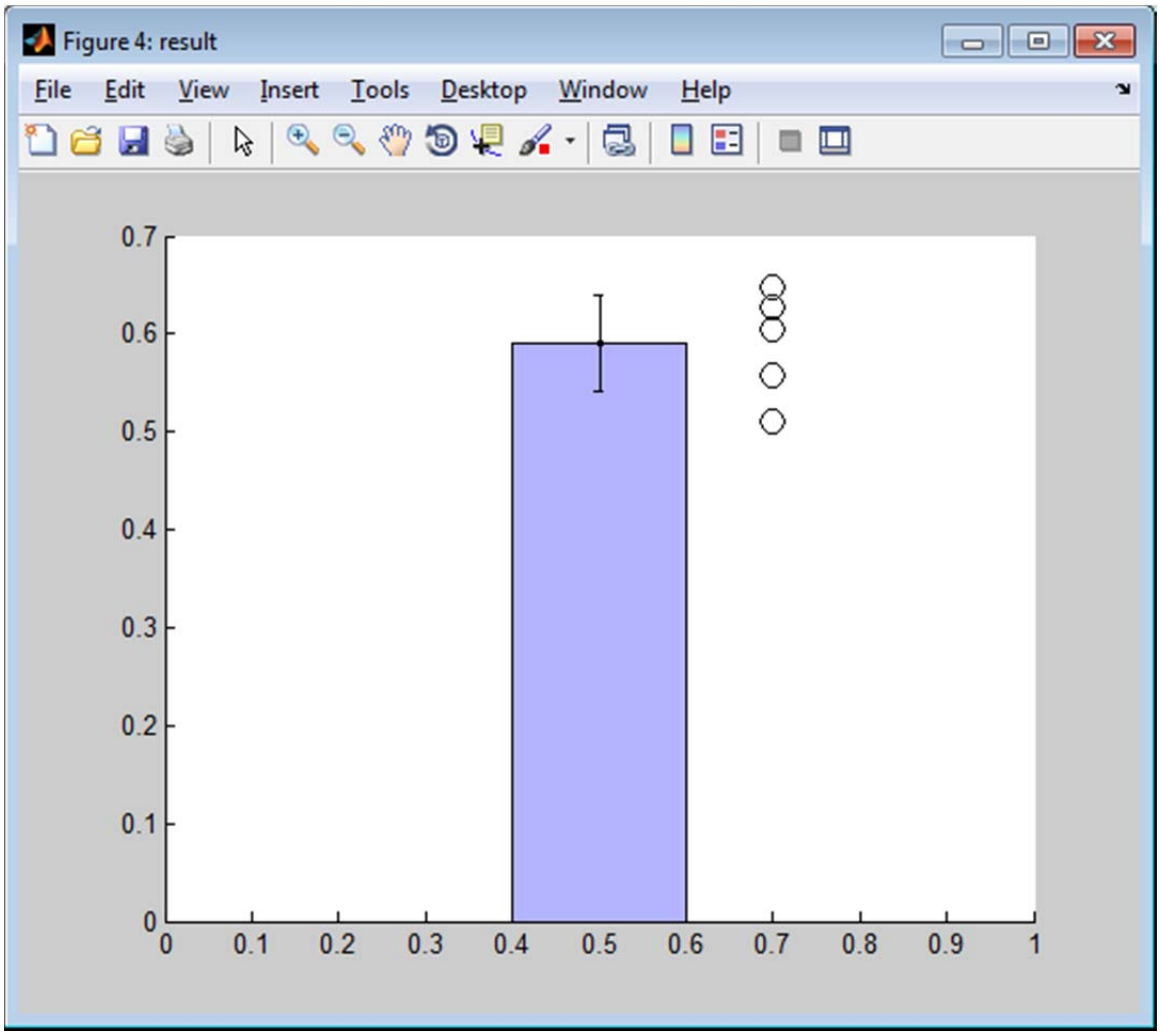

Figure 23. The resulting plot from eyetracking_AOI_diffscores.

6. Final comments and remarks

Worth mentioning is that during this demonstration all plugins has been applied one by one as they were added. It is also possible to first add all plugins (still in this particular order) to the study and then applying them all at once by selecting all subjects and all plugins and then pressing the “Do selected work” button.

Also, by changing the settings it is possible to define the AOIs in other ways but still get the same result. This example just shows one of the possible ways to use these plugins. Other manuals will cover more advanced uses of AOIs in TimeStudio.

7. References

Fox, R. & McDaniel, C. (1982) The perception of biological motion by human infants. *Science*. 208 (2571), 486-487.

8. Acknowledgements

The authors want to thank the following people for comments and suggestions for this manual and for testing the workflow: Gustaf Gredebäck, Terje Falck-Ytter, Joshua Jovrud , Tommie Forslund, Christine Fawsett and Ida Hensler. The manual was funded by grants ERC-StG-312292-CACTUS and the Swedish Research Council in partnership with FAS, FORMAS and VINNOVA [Cross-disciplinary research programme concerning children's and young people's mental health; grant number 259 - 2012-24].