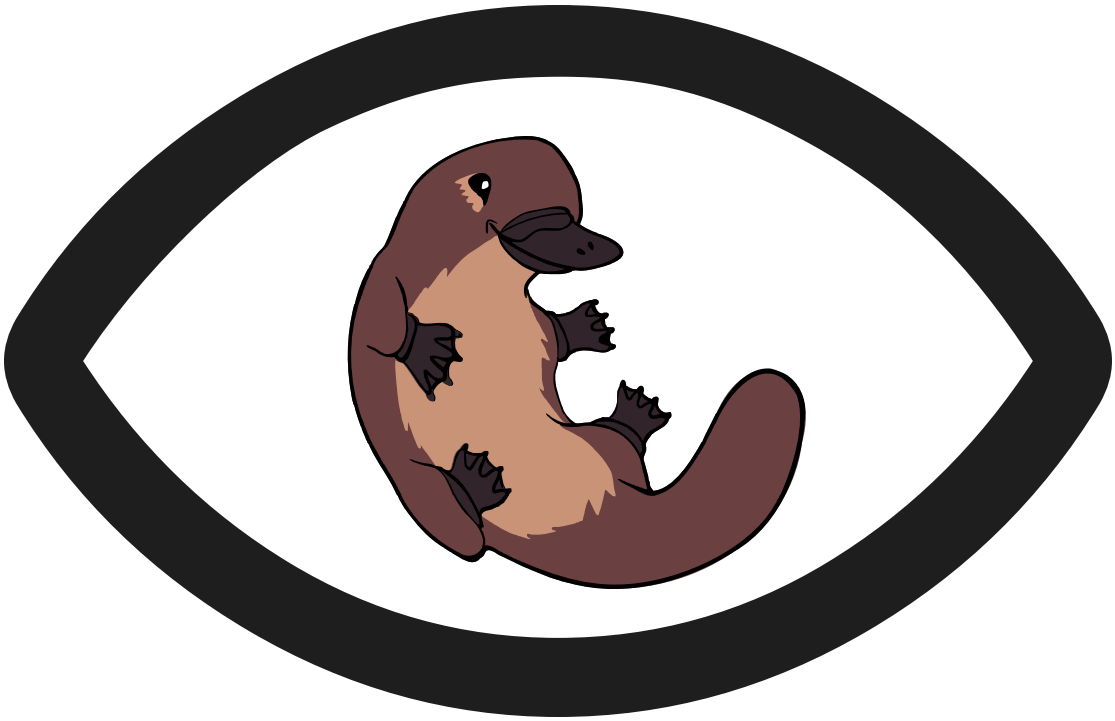


BORIS user guide

v. 9.8.1

Olivier Friard and Marco Gamba

Table of contents

1. User guide of BORIS the Behavioral Observation Research Interactive Software

BORIS is a user-friendly software designed for event logging during video/audio coding and live observations. It is a free and open-source application that can be used on GNU/Linux and Windows operating systems.

The official BORIS web site is **https://www.boris.unito.it**.

This user guide is applicable to the version **9.8.1** of BORIS.

A PDF version of this user guide is available.

2. User guide

2.1 Installation

BORIS can be installed following the instructions on the download section of the BORIS web site

All previous versions of BORIS are available in the Releases section of the GitHub repository.

2.1.1 GNU / Linux

BORIS can run on various Linux distributions including Ubuntu, Debian, Raspberry Pi OS, Chromebook ...).

See the BORIS for Linux page to install BORIS for Linux.

2.1.2 Microsoft-Windows

See the BORIS for Microsoft-Windows page to install BORIS for Windows.

Two versions are available: BORIS Portable and BORIS Setup

2.1.3 macOS

BORIS for macOS is experimental for now. Please report any issue on GitHub.

See the BORIS for macOS page to install BORIS for macOS.

2.2 Starting BORIS

Once BORIS is installed, it can be launched by clicking on its icon.

Warning for Windows users

BORIS does not yet use signed binaries which means that you will need to allow the execution of the downloaded executable. If there is no obvious way to do so, click on "More info" on the error message that shows up and then on "Run anyway".

First launch

The initial launch of BORIS may take some time to display. Please be patient!

The main window of BORIS will appear. Currently, all commands on the toolbar are disabled, except for the Preferences button.

The BORIS main window

If you want to launch BORIS from the source code, refer to the Run BORIS from source code section.

2.3 Create a project

The BORIS project file serves as a container for all project-related information, excluding media files. It encompasses the **ethogram**, **independent variables**, **subjects' definition**, **behavioral coding maps**, **converters**, and **observation** data. To save the project on your local file system, use the "**File**" > **Save Project** or **Save Project As ...** options.

Additionally, you can activate the automatic backup feature in the Preferences section.

> **Very important**
>
> It is **EXTREMELY IMPORTANT** to perform regular backups of your project files to prevent the loss of data. While software can be reinstalled, your data might be irretrievably lost. Consider using an external drive and/or a cloud service for secure backup.

BORIS allows the creation of an unlimited number of projects, but only one project can be opened at a time.

A video tutorial about creating a project is available at this link.

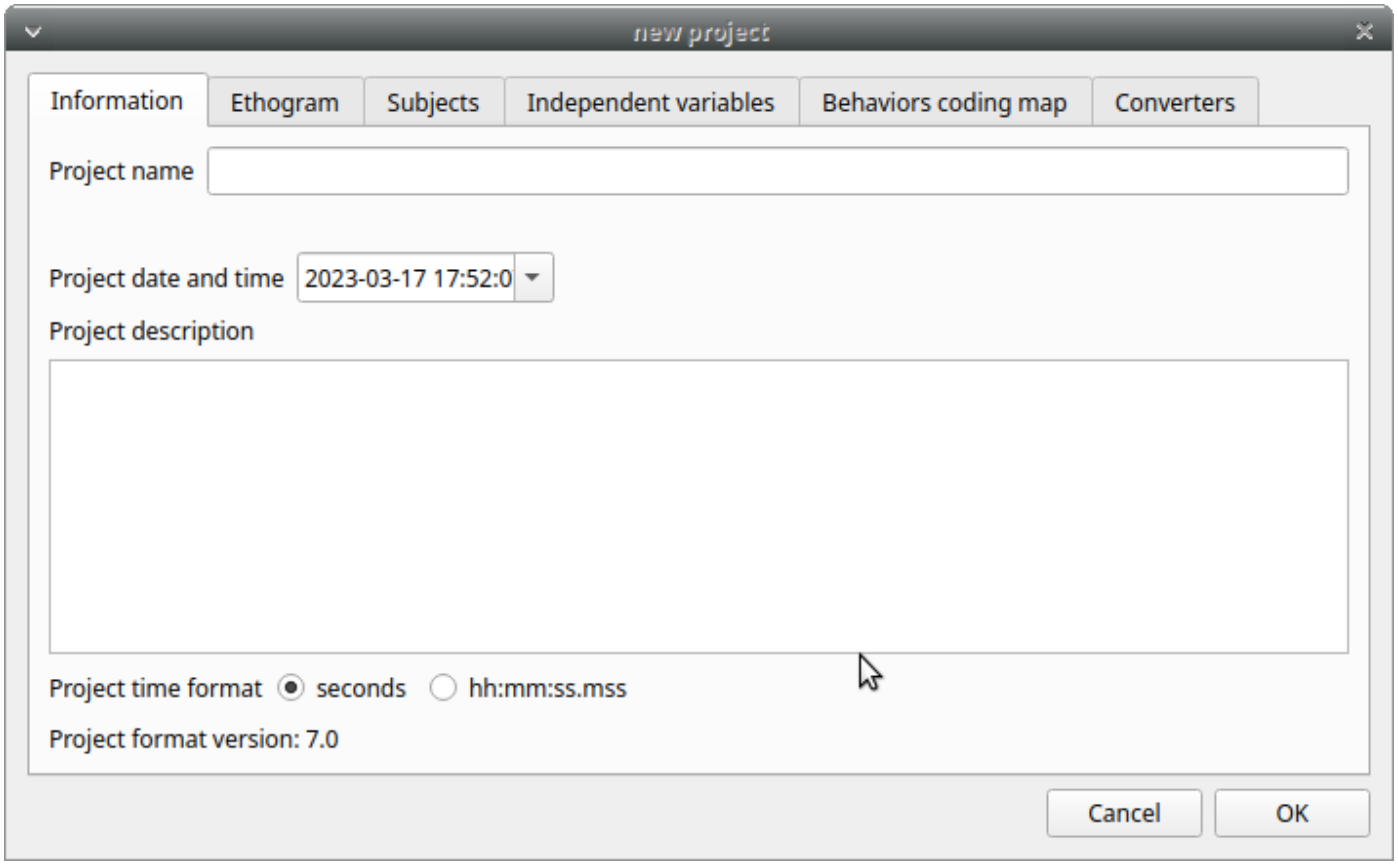
To create a new project, under the menu **File** , select **New project**.

You can determine your project name by writing in the **Project name** field in the **Information** tab. Once the project will be saved, the **Project file path** will show the full path to your project file.

Date will automatically set on the current date and time, but you can alternatively set this info on your media date and time, or whatever you prefer.

Description can host all the relevant information about your project, can be also left empty.

Time format can be alternatively set to **seconds** or to **hh:mm:ss.mss**. This choice can be changed at anytime under **File** > **Preferences**.



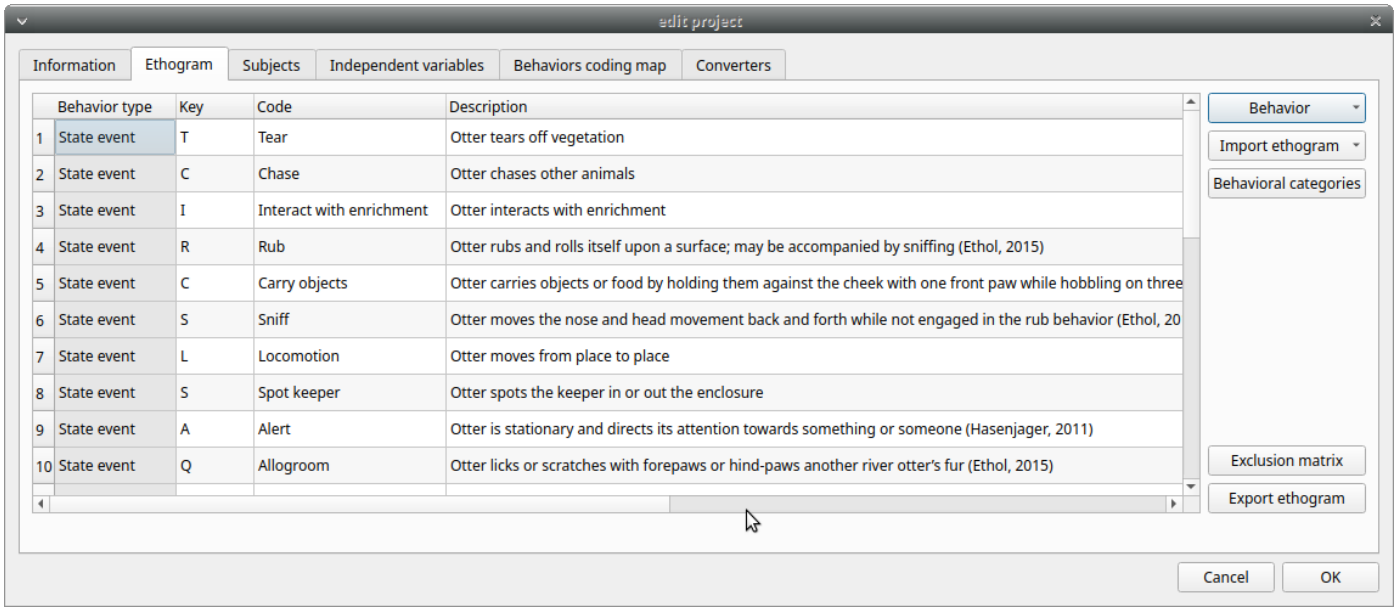
BORIS main window

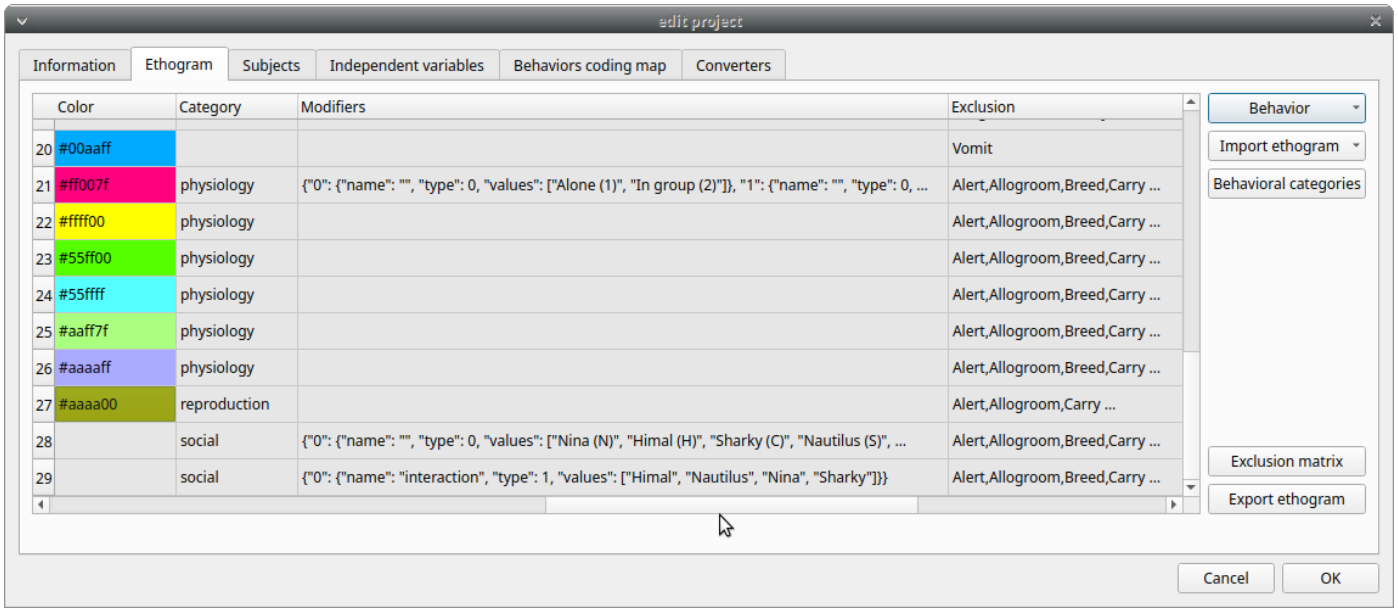
2.3.1 Set an ethogram

See the Wikipedia ethogram definition.

Switching to the **Ethogram tab**, you can alternatively:

- set your ethogram from scratch;
- import an existing ethogram from another BORIS project;
- import an ethogram from a JWatcher global definition file (.gdf).
- import an ethogram from a plain text file or a spreadsheet file (XLSX or ODS)





Set your ethogram from scratch

By clicking on the **Behavior > Add behavior** button, you can add a new row in the **Ethogram** table, and the behavior type will be automatically set to **Point event**.

The cells with gray background can not be directly edited. You must double-click on them and then select a value.

Behavior types

2 types of behaviors can be defined. Double-click on the cell and select the type of behavior:

- **Point event** behavior when the behavior has **no duration**.

  The behavior will be coded by pressing the defined keyboard key (see below) or by double-clicking to the corresponding row in the Ethogram table.

- **State event** behavior when the behavior has a **duration**.

  The behavior start and stop will be coded by pressing the defined keyboard key (see below) or by double-clicking to the corresponding row in the Ethogram table. These behaviors **must** have a start event and a stop event otherwise an **UNPAIRED events** warning will be reported when you will close the observation or during an analysis.

- **Point event with a coding map**

  a **Point event** that can be coded uusing a **coding map**.

- **State event with a coding map**

  a **State event** that can be coded using a **coding map**.

You can switch between the types of behavior at your convenience with a double-click on the **Behavior type** cell. You can also add a **Coding map** to either a **State event** (**State event with coding map**) or a **Point event** (**Point event with coding map**). See the **Coding map** section for details.

An existing behavior can be duplicated using the **Clone behavior** button. Its code have then to be changed. On a selected behavior, click on the **Remove behavior** button to remove. The **Remove all behaviors** button will clear the **Ethogram** table. Both the above-mentioned operations must be confirmed when prompted.

The behavior can be sorted by clicking on the Ethogram table header. They cannot be sorted manually.

Set keys and codes

For each behavior you have to set a keyboard key (**Key** column) that will be then used to code the behavioral events. You can choose whether you want to set a unique key for each behavior or use the same key for more than one behavior. In the case you set the same key for more than a behavior, BORIS will pause your coding and ask which of the behavior you want to record. The keys are **case-sensitive**.

If your project was created with an old version of BORIS (< v.7) you can use the **Convert keys to lower case** to convert all keys to lower case otherwise you will have to code your observation using upper case key.

> If you open a project file created with a version older than v.7 BORIS will ask you to convert the upper case behavior and subject keys to lower case.

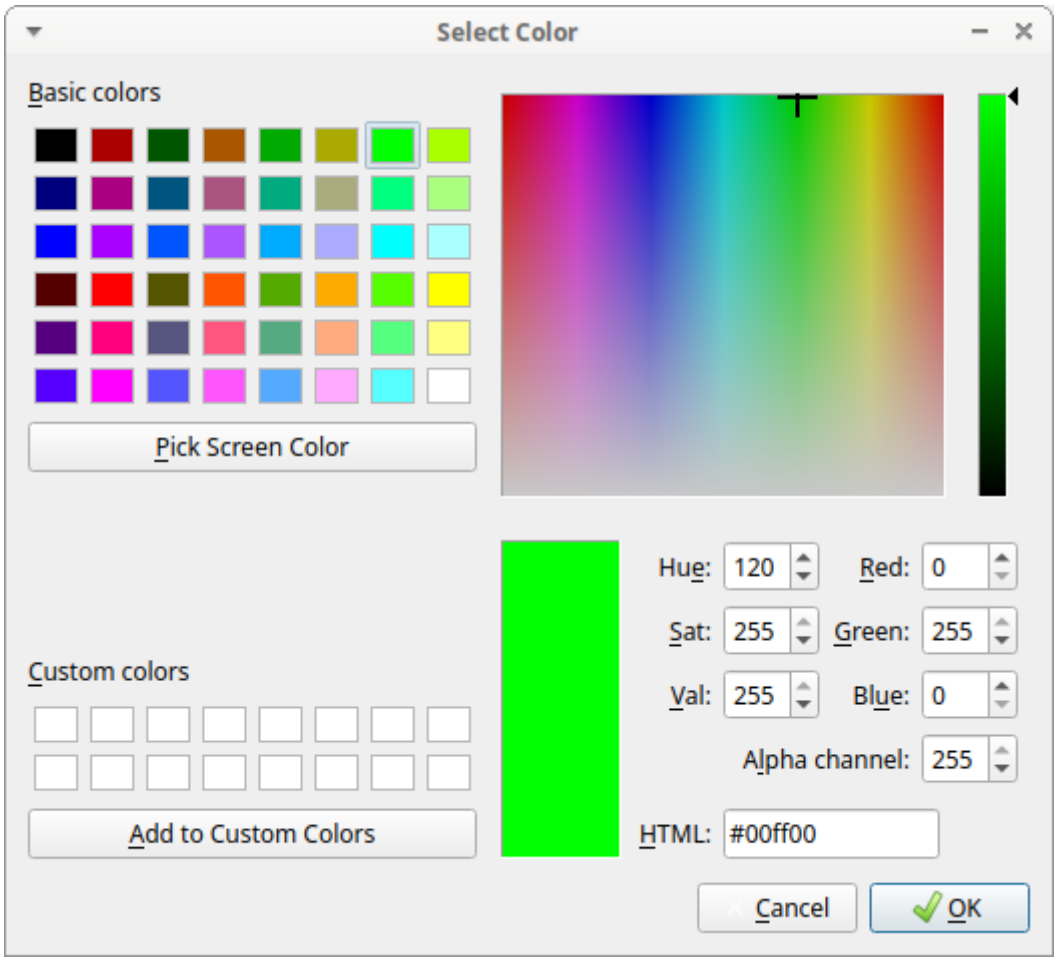
> **Important**
>
> **Do not use the / and * keys! They are reserved for the frame-by-frame mode.**

In the **Code** column, you have to add a unique code for each behavior. Duplicated codes are not accepted and BORIS will warn in red about duplicates on the bottom left of the **Ethogram** tab. The code can be an alphanumeric string (which must not include the pipe character |).

The **Description** of your behavior is optional. The **Description** column can be useful to add information about a specific behavior, its characteristics (e.g. to standardise observation between different users) or to refer to external information (e.g. reference to a previous ethogram).

The columns with a grey background (**Behavior type**, **Color**, **Category**, **Modifiers**, **Exclusion**, **Modifiers coding map**) cannot be edited directly.

The **Color** column allow to select a color for the behavior. This color will be used for plotting events. Double-click on the cell and select the color you want to associate to the behavior.



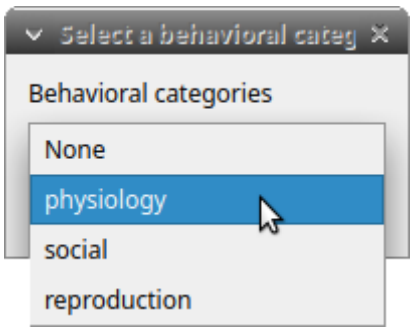
Select the color to associate to the behavior

CATEGORIES OF BEHAVIORS

Defining categories of behaviors can be usefull for the analysis of coded events (for example the time budget analysis).

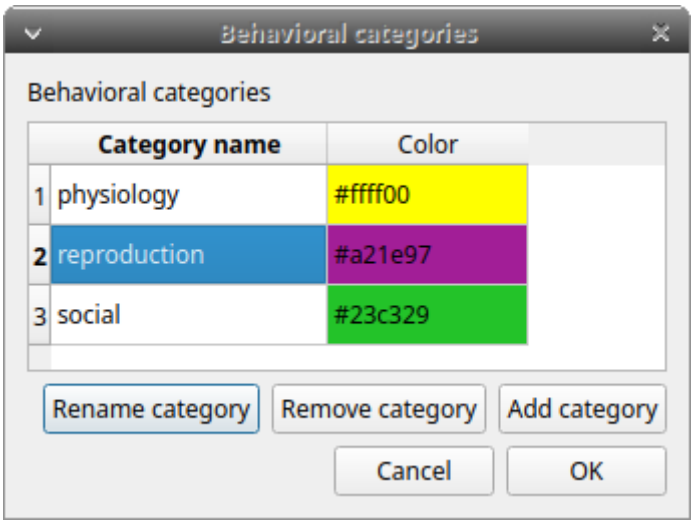
The **Category** column allow you to include the behavior to a predefined behavioral category.

Double-click on the cell and select the behavioral category for the behavior.



Choose a behavioral category for the behavior

To add, remove or rename a behavioral category, click the **Behavioral categories** button. A color can also be associated to a behavioral category.



Behavioral categories manager

Set the modifiers

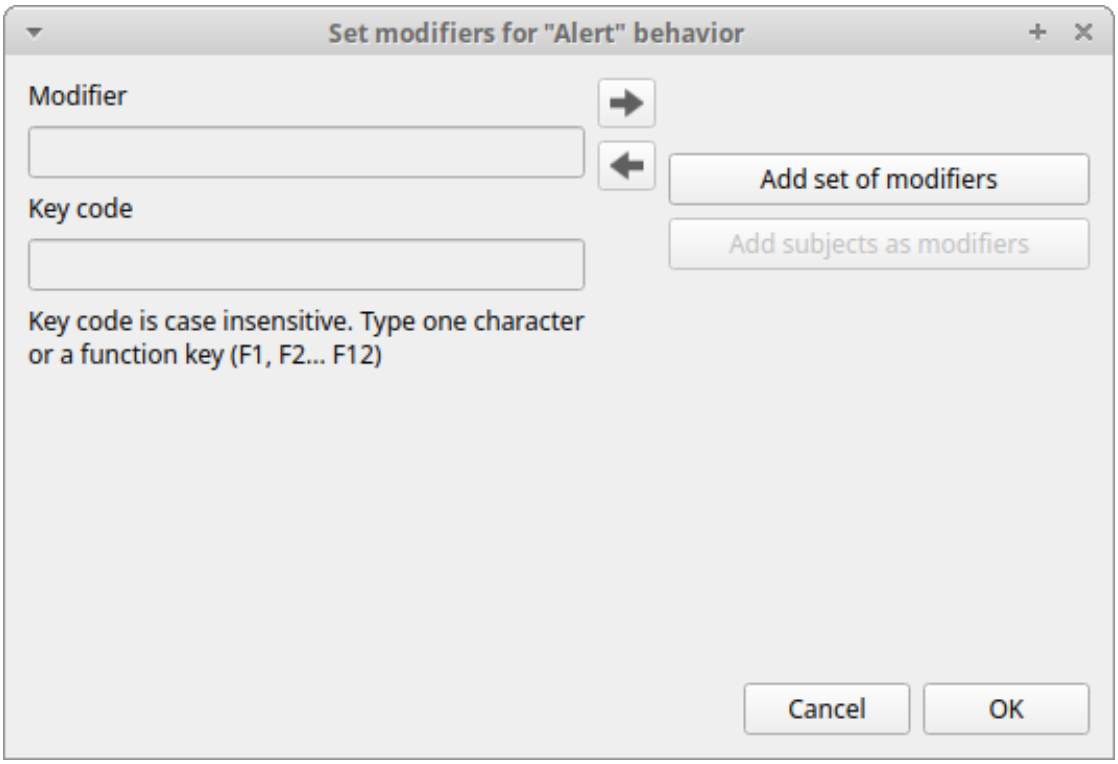
Modifiers can be used to add attributes to a behavior. A single behavior can have two or more modifiers attached (e.g. the behavior **play** may have **solitary** or **social** as modifiers). The use of modifiers can be convenient to significantly reduce the number of keys and simplify the behavioral coding.

4 types of modifiers are available: **Single selection**, **Multiple selection**, **Numeric** and **Value from external data file**:

- the **Single selection** type will allow the observer to select only **one** modifier for the current behavior.
- the **Multiple selection** type will allow the observer to select one or more modifiers for the current behavior.
- the **Numeric** type will allow the observer to input a number. For example a distance of interaction.
- the **Value from external data file** type will save the value of a variable from an external data file.

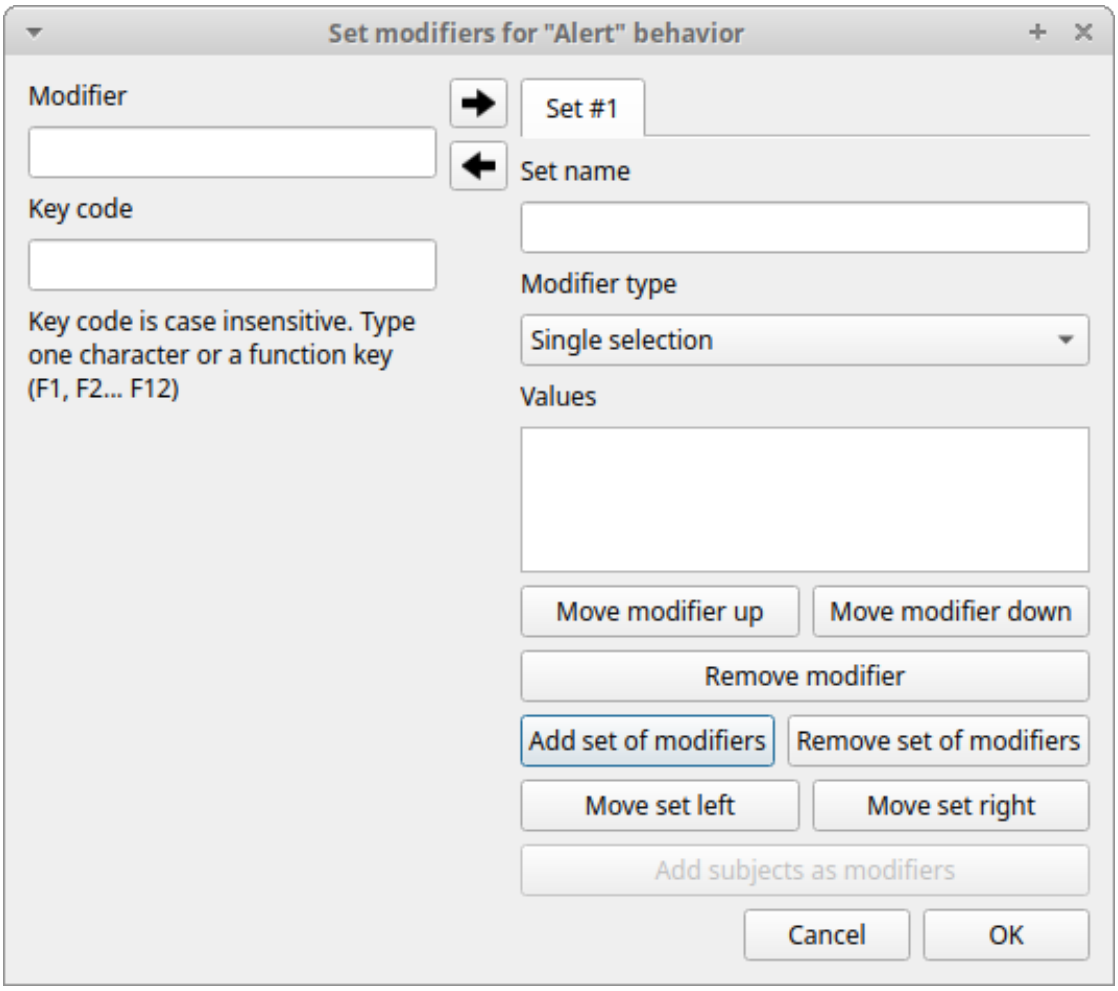
In BORIS modifiers can also be added in different modifier sets (e.g. **play social** may have a modifier set (#1) for **brothers** and another (#2) for **sisters**). In the case of using sets of modifiers, you can select one/more modifier for each set.

To add modifiers to a behavior, you need to double-click the **Modifiers** cell corresponding to the behavior you want to add the modifiers to. The following window will show up:



Modifiers configuration

Click the **Add a set of modifiers** button:



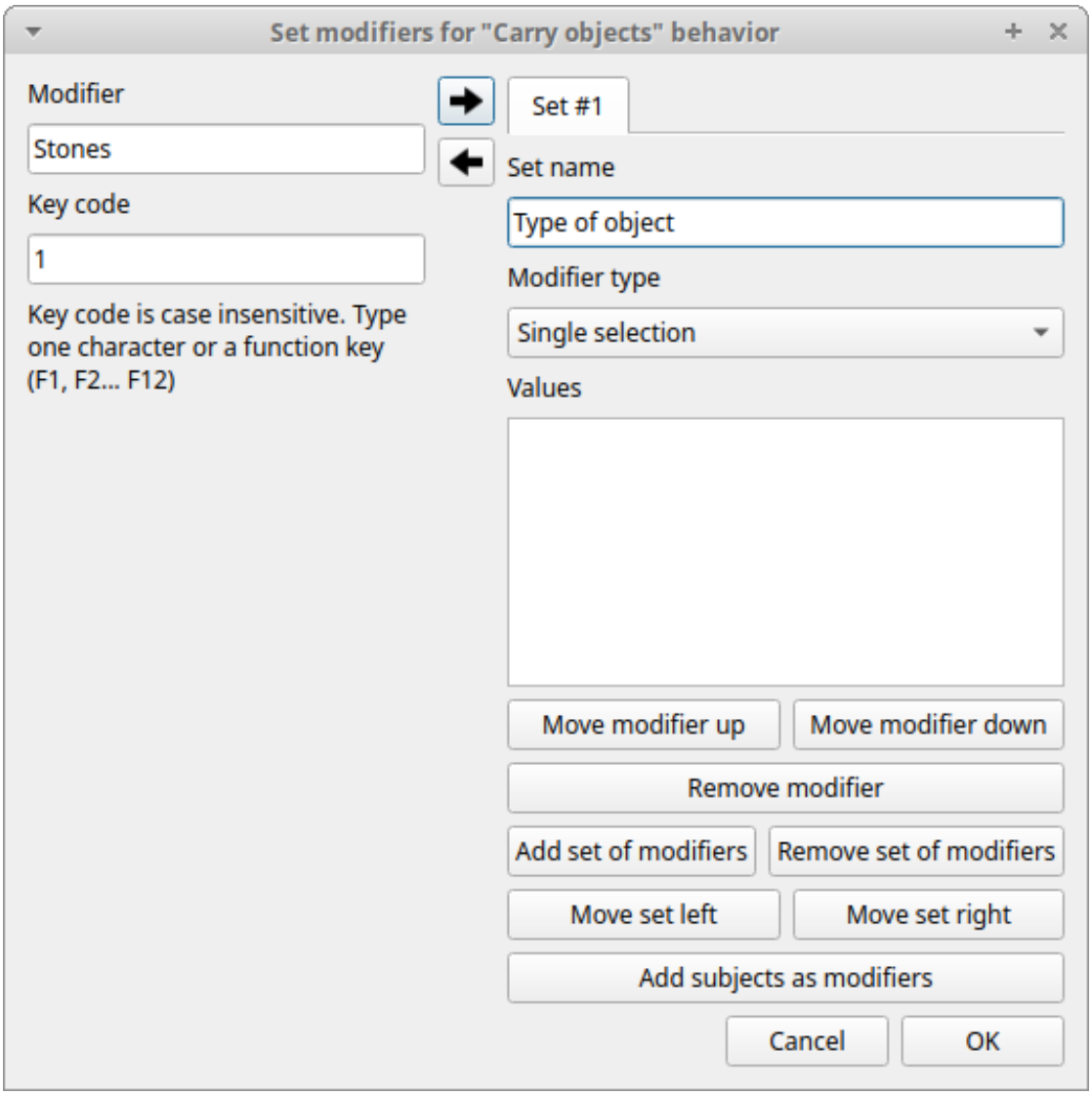
Modifiers configuration

Select the modifier type using the **Modifier type** combo box. You have to choose between **Single selection**, **Multiple selection Numeric** and **Value from external data file**.

SINGLE SELECTION AND MULTIPLE SELECTION MODIFIERS

Set a name for the new modifiers set by typing it in the **Set name** edit box. Setting a modifiers\' set name is not mandatory.

Within a set of modifiers, you can add a modifier by writing the modifier in the **Modifier** edit box. You can choose a shortcut (one character - case sensitive) to this modifier (optional). Then press the **right-arrow** button to add the new modifiers to the set.

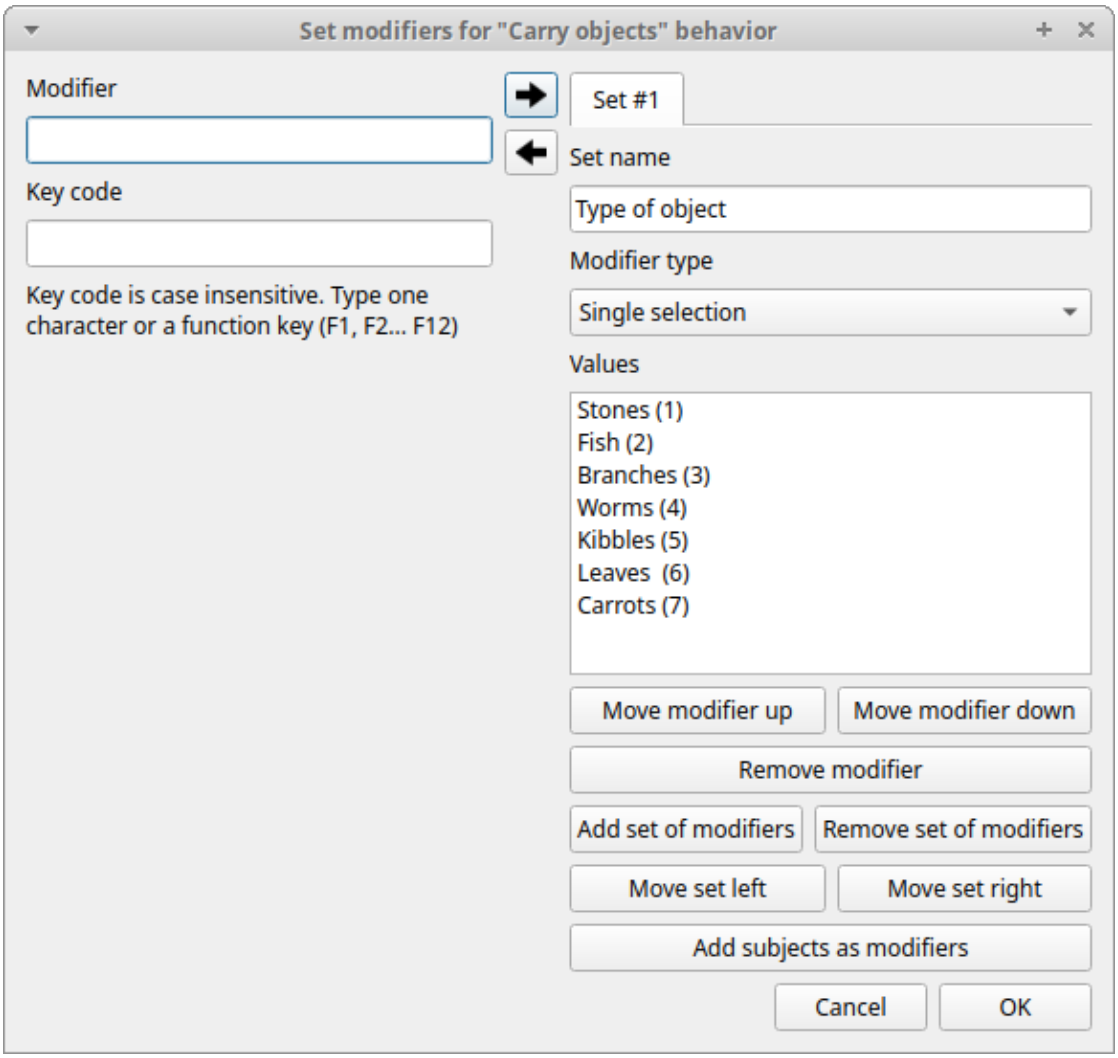


Modifiers configuration

To modify a modifier, select it and press the **left-arrow** button, edit the modifier and press the **right-arrow** button.

A modifier can be removed by pressing the **Remove modifier** button.

After adding all modifiers the window will appear like this:



Modifiers configuration

All defined subjects can be added as modifiers using the **Add subjects as modifiers** button. This can help in case of coding the interactions between subjects for example.

The modifiers can be loaded from a plain text file Use the **Load modifiers from file** button.

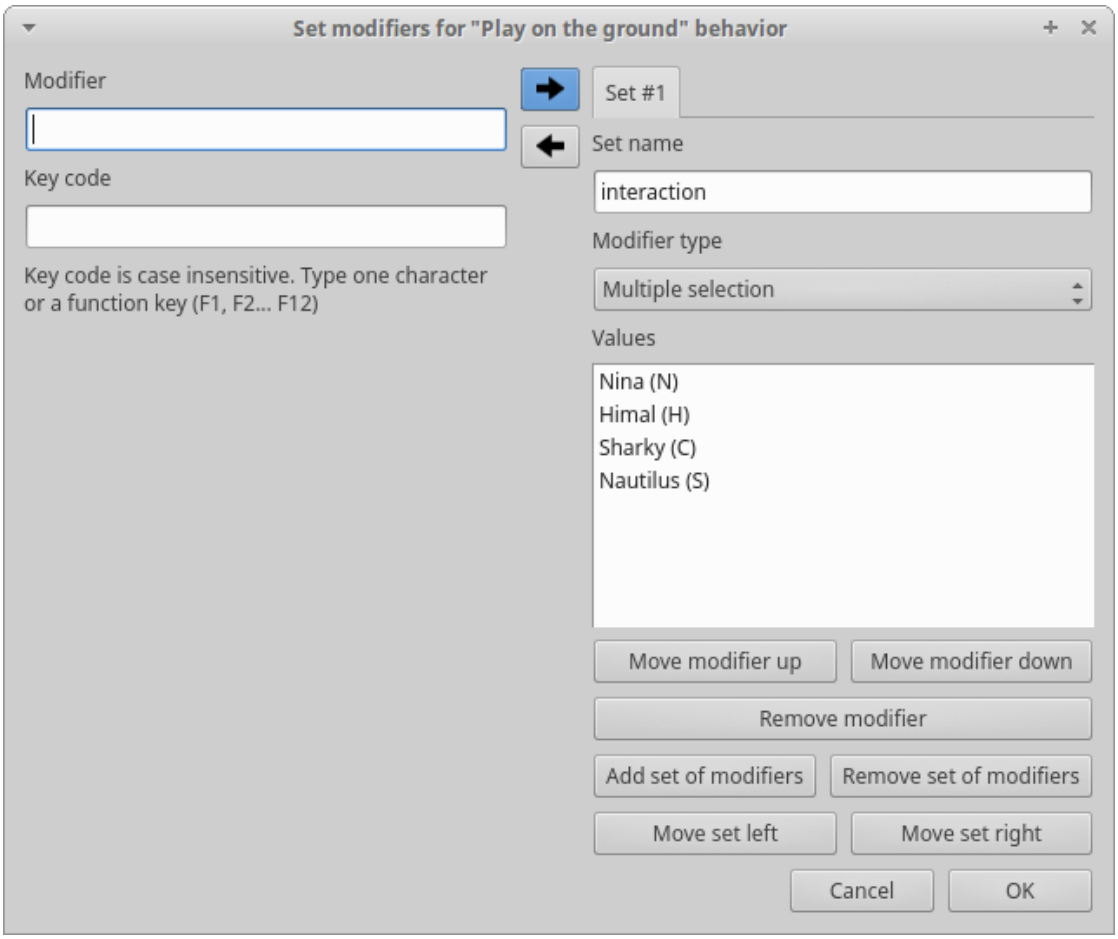
The modifier position into the modifiers\' set can be manually set using the **Move modifier up** and **Move modifier down** buttons. The modifiers can be sorted alphabetically (use the **Sort modifiers** button).

You can add and/or remove sets using the buttons **Add set of modifiers** and **Remove set of modifiers**.

The position of a modifiers\' set can be customized (using the **Move set left** and **Move set right** buttons)

Modifiers can not contain the following characters: **(|),`\~!**

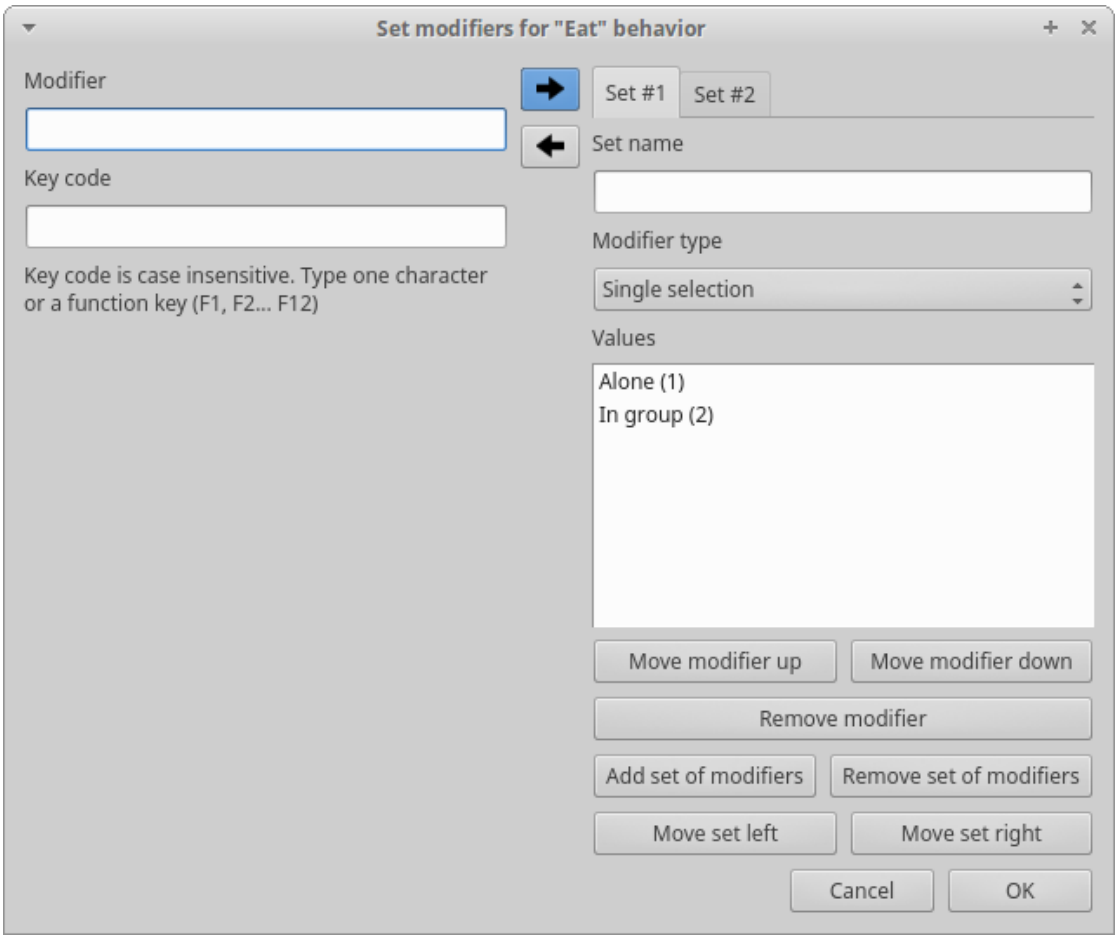
Example of a **multiple selection** modifiers set:



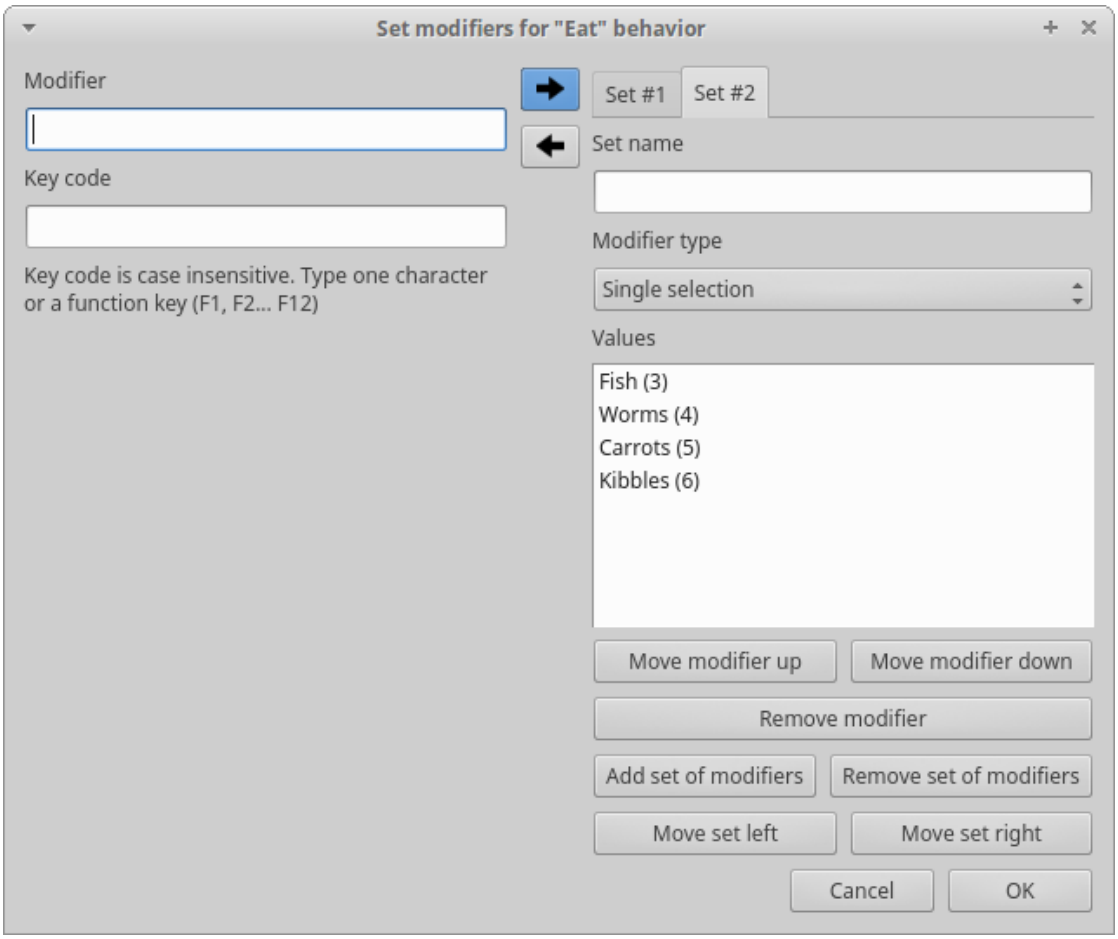
Modifiers configuration

Many values can be selected together.

Example of 2 sets of modifiers:



Modifiers configuration



Modifiers configuration

NUMERIC MODIFIER

Set a name for the new set by typing it in the **Set name** edit box. Setting a modifiers' set name is not mandatory.

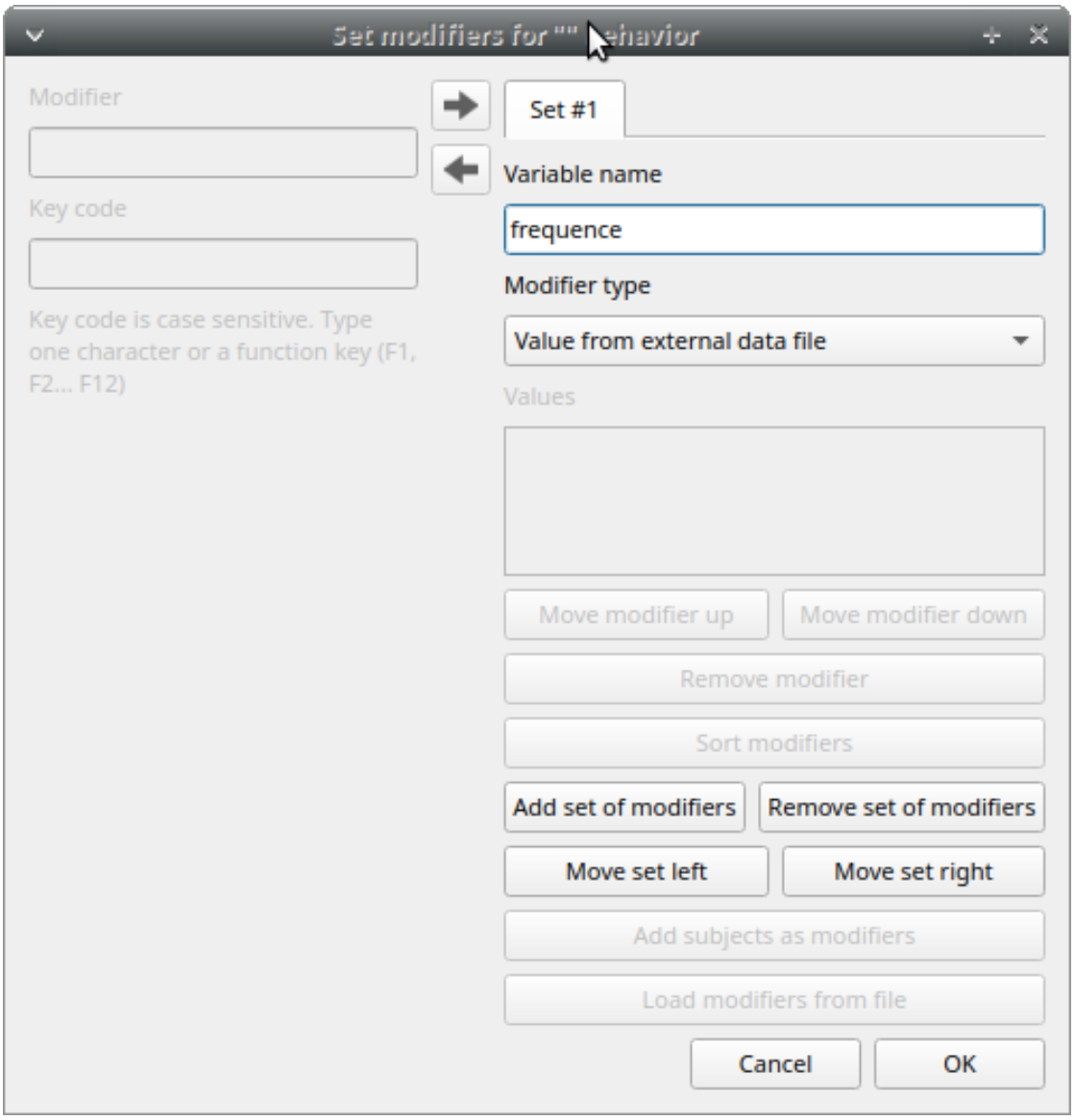
When a **Numeric** modifier will trigger, BORIS will ask the observer for a numeric value.

VALUE FROM EXTERNAL DATA FILE MODIFIER

This modifier can be used to record the value of a variable coming from an external data file (defined during the creation of the observation).

You have to define the variable name in the **Variable name** edit box. This is mandatory and the name of the variable **must** be the same than the variable defined in the observation.

See External data files



modifier value from external data file

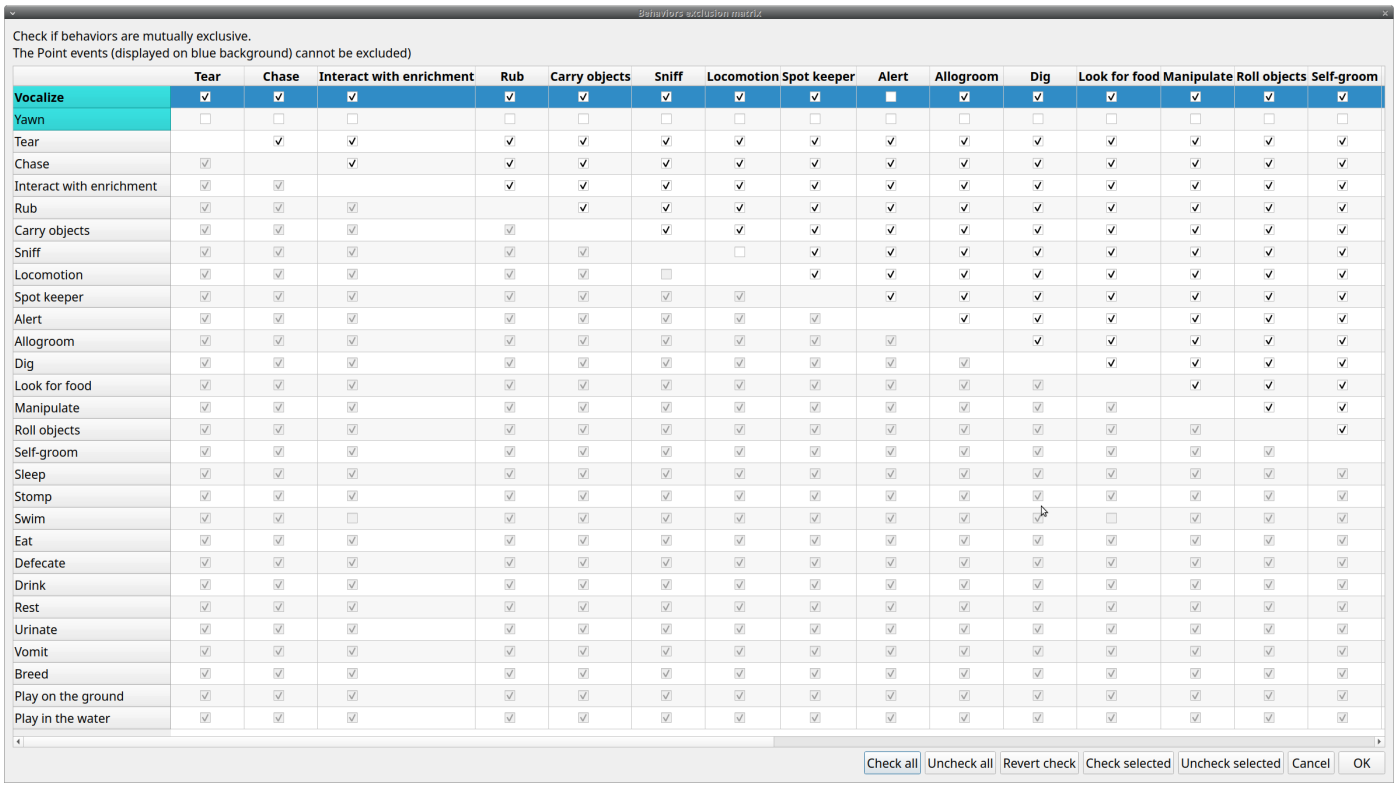
Click **OK** to save modifiers in the **Ethogram** table.

Set the exclusion matrix

The occurrence of an event (State or Point) can exclude the occurrence of a state event. This can be set using the **Exclusion matrix** window, which can be opened clicking on the **Exclusion matrix** button. BORIS will ask for including **Point events** or not and a new **Exclusion matrix** window will open.

Exclusive behavior may be selected by checking on the corresponding checkbox in the automatically-generated matrix. We suggest to work on the **Exclusion matrix** when all the behaviors have been added to your ethogram.

All behaviors can be excluded by a particular behavior by selecting the corresponding entire row (click on the row header of the behavior) and by clicking on the **Check selected** button. You can also uncheck all behaviors by selecting the **Uncheck selected** button.



	Tear	Chase	Interact with enrichment	Rub	Carry objects	Sniff	Locomotion	Spot keeper	Alert	Allogroom	Dig	Look for food	Manipulate	Roll objects	Self-groom
Vocalize	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑
Yawn	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Tear		☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Chase	☑		☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Interact with enrichment	☑	☑		☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Rub	☑	☑	☑		☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Carry objects	☑	☑	☑	☑		☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Sniff	☑	☑	☑	☑	☑		☐	☑	☑	☑	☑	☑	☑	☑	☑
Locomotion	☑	☑	☑	☑	☑	☐		☑	☑	☑	☑	☑	☑	☑	☑
Spot keeper	☑	☑	☑	☑	☑	☑	☑		☑	☑	☑	☑	☑	☑	☑
Alert	☑	☑	☑	☑	☑	☑	☑	☑		☑	☑	☑	☑	☑	☑
Allogroom	☑	☑	☑	☑	☑	☑	☑	☑	☑		☑	☑	☑	☑	☑
Dig	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑		☑	☑	☑	☑
Look for food	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑		☑	☑	☑
Manipulate	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑		☑	☑
Roll objects	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑		☑
Self-groom	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	
Sleep	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Stomp	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Swim	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑
Eat	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Defecate	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Drink	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Rest	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Urinate	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Vomit	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Breed	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Play on the ground	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Play in the water	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑

Example of an exclusion matrix

For example in the previous figure, the **Alert** behavior will exclude the following behaviors: **Allogroom**, **Breed**, **Carry objects**, **Chase** ...

During the observation, the excluding event will stop all the current excluded state events one millisecond before the occurence of the event.

Set the Modifiers coding map

If the behavior is defined as a **Point event with coding map** or a **State event with codinf map** you can associate a **Modifiers coding map** to select the modifiers from a map.

Import an ethogram from an existing project

Behaviors within an ethogram can be imported from an existing BORIS project (.boris) using the **Import ethogram > from a BORIS project** button. BORIS will ask to select a BORIS project file and whether imported behaviors should replace or be appended to the **Ethogram** table. Imported behaviors will retain all the previously defined behavior parameters (namely Behavior type, Key, Code, Description, Modifiers and Exclusion information).

Import an ethogram from a spreadsheet file

Behaviors can be imported from a spreadsheet file using the **Import ethogram > from spreadsheet file (XLSX/ODS)** button.

The first row of your spreadsheet (header) must contain the following labels. The order is not mandatory:

- Behavior code
- Behavior type
- Description
- Key
- Behavioral category
- Excluded behaviors

Behavior code is mandatory, the others fields can be empty.

Optional fields can be added:

- Color
- Modifiers (JSON)

BORIS will ask to select a spreadsheet file (by default: *.xlsx or* .ods) and whether imported behaviors should replace or be appended to the **Ethogram** table. The missing information for the imported behaviours have to be redefined.

Import an ethogram from a plain text file

Behaviors can be imported from a plain text file using the **Import ethogram > from text file** button. The fields must be separated by TAB, comma (,) or semicolomn (;). All rows must contain the same number of fields.

The first row of your plain text file must contain the following labels. The order is not mandatory but respect the case:

- Behavior code
- Behavior type
- Description
- Key
- Behavioral category
- Excluded behaviors

Behavior code is mandatory, the others fields can be empty.

Example of a plain text ethogram definition:

```
Behavior type,Behavior code,Key,Behavioral category,Description,Excluded behaviors
state event,Play,p,,Play on the garden,s
point event,Sleep,s,,Subject is sleeping,p
```

BORIS will ask to select a plain text file (by default: *.txt *.csv *.tsv) and whether imported behaviors should replace or be appended to the **Ethogram** table. The missing information for the behaviours imported from text file have to be redefined.

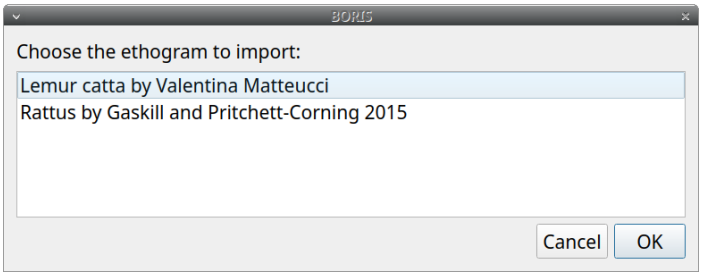
Import an ethogram from a JWatcher global definition file (.gdf)

Behaviors can be imported from a JWatcher global definition file (.gdf) using the **Import ethogram > from JWatcher** button. BORIS will ask to select a JWatcher file (.gdf) and whether imported behaviors should replace or be appended to the **Ethogram** table. Behavior type and exclusion information for the behaviours imported from JWatcher have to be redefined.

Access to the BORIS ethogram repository

This function can be activated by clicking the **Import ethogram > from the BORIS repository** button.

A list of available ethograms will open and an ethogram can be loaded in the current project.

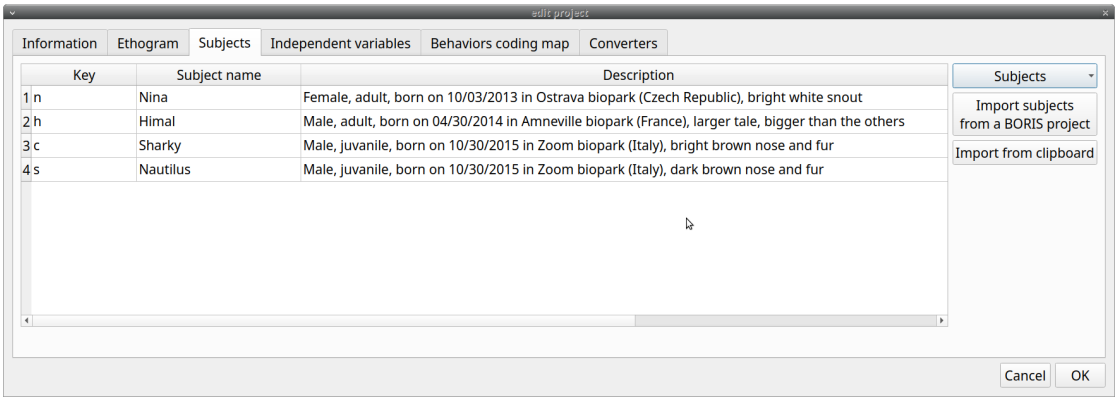


BORIS ethogram repository

Export the ethogram

The entire ethogram can be exported in various formats (TSV, CSV, XLSX, ODS, HTML). See **File** > **Edit project** > **Ethogram tab** > **Export ethogram**

2.3.2 Define the subjects



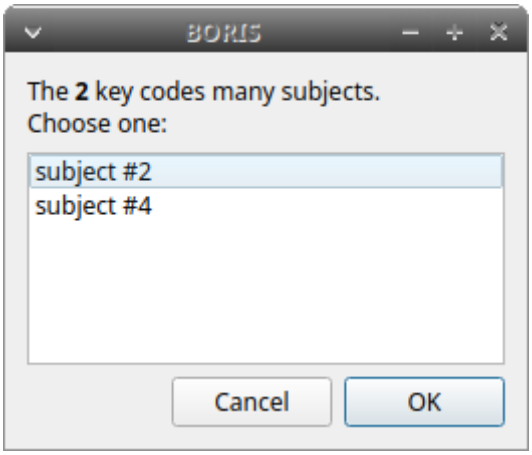
Subjects definition

BORIS allows coding behaviors for different subjects within a single observation. The **Subject** table allows the specification of subjects using a **Key** (e.g., the **k** on your keyboard), **Subject name** (e.g., **Kanzi**), and **Description** (e.g., male, born on October 28, 1980).

With the subjects defined in the previous figure, pressing **n** will set **Nina** as the focal subject for behavioral coding. Pressing **n** again will deselect **Nina** and set the focal subject to **No focal subject**.

The key definition is not mandatory. In this case, you will have to select the current subject from the subjects list with a double-click.

The keys are **case-sensitive** and the same key can be used to select more than one subject. In this case a dialog will show up and will allow to select



Choose a subject

The definition of one or more subjects is not mandatory. Addition, removal and sorting of the subjects follows the same logic of the **Ethogram** table (see Set your ethogram from scratch for info).

Note

If your project was created with a previous version of BORIS (< v.7) you can use the **Convert keys to lower case** to convert all keys to lower case otherwise you will have to code your observation using upper case key.

The subjects can also be imported from an existing BORIS project: use the **Import Subjects from a BORIS project** button.

Import subject from a spreadsheet

The subjects can be imported from a spreadsheet (Google spreadsheet, Microsoft-Excel, LibreOffice Calc).

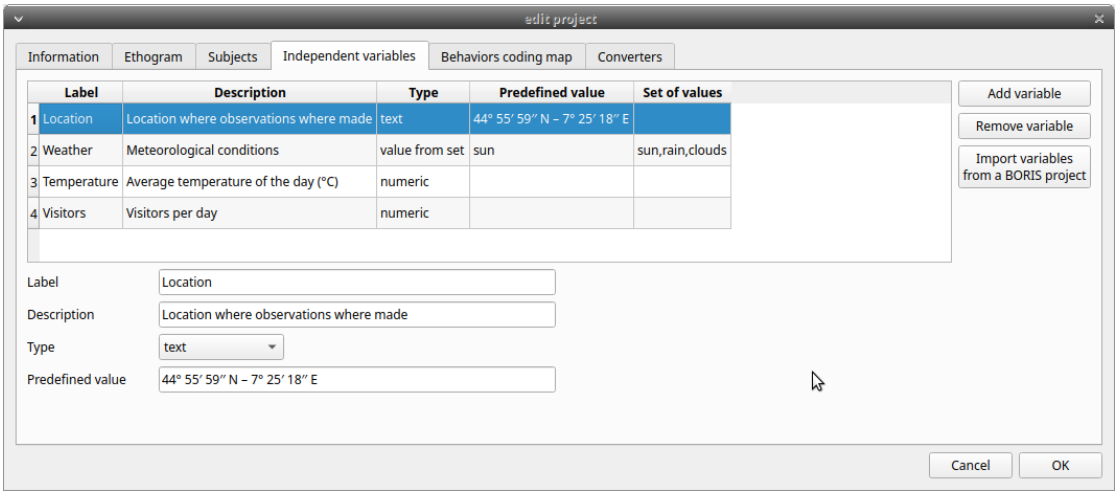
The spreadsheet must contain one subject by row and have to be organized as above:

- 1st column: Subject key (One character - Case sensitive - Optional)
- 2nd column: Subject name (mandatory)
- 3rd column: Description of subject (optional)

Select all cells of your spreadsheet (`^ Ctrl` + `A`), copy to clipboard (`^ Ctrl` + `C`). Click the **Import from clipboard** button.

> **Note**
>
> If you open a project file created with a version older than v.7 BORIS will ask you to convert the upper case behavior and subject keys to lower case.

2.3.3 Define the Independent variables

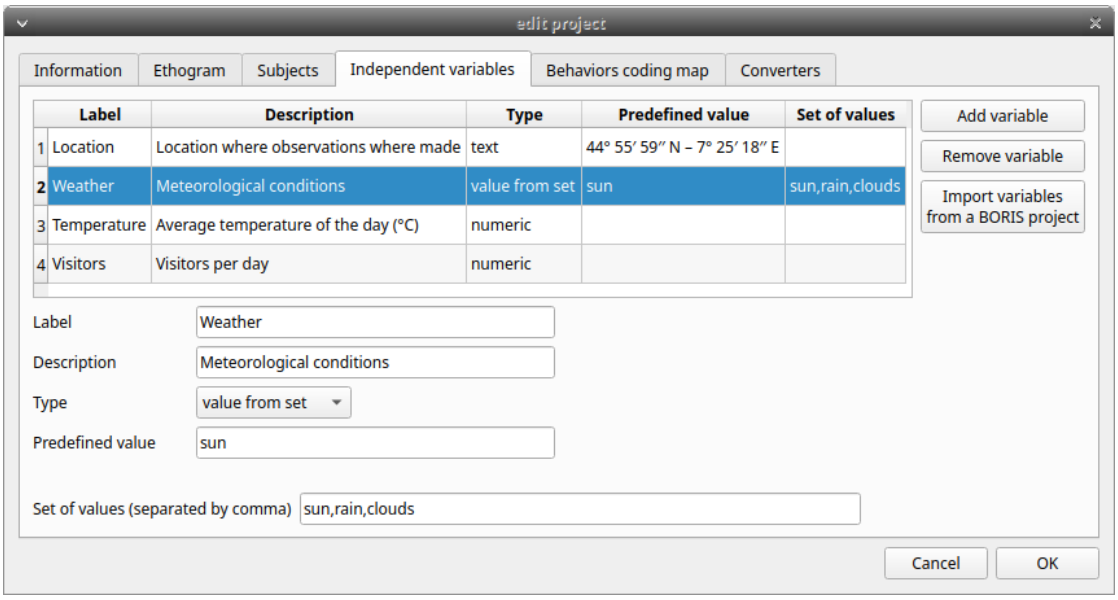


Independent variables

BORIS allows adding information about the observation using **Independent variables**. This can be used to specify factors that may influence the behaviors (e.g. group composition, temperature, weather conditions) but will not change during a single observation within a project. Each independent variable can be defined by a **Label** (e.g. weather), a **Description** (e.g. weather conditions), a **Type** (*text*, *numeric*, *value from set* or *timestamp*).

The values of a set are defined in the **Set of values** column separating the available values with a comma (,). Please note that the first value of the set will be selected by default. It should be useful to define a NA value as first value of every set.

The values for the independent variables will be asked when creating a new observation. Addition, removal and sorting of the independent variables follows the same logic of the **Ethogram** table (see **Set your ethogram from scratch** for info). The independent variables can also be imported from an existing BORIS project using the **Import Variables from a BORIS project**.

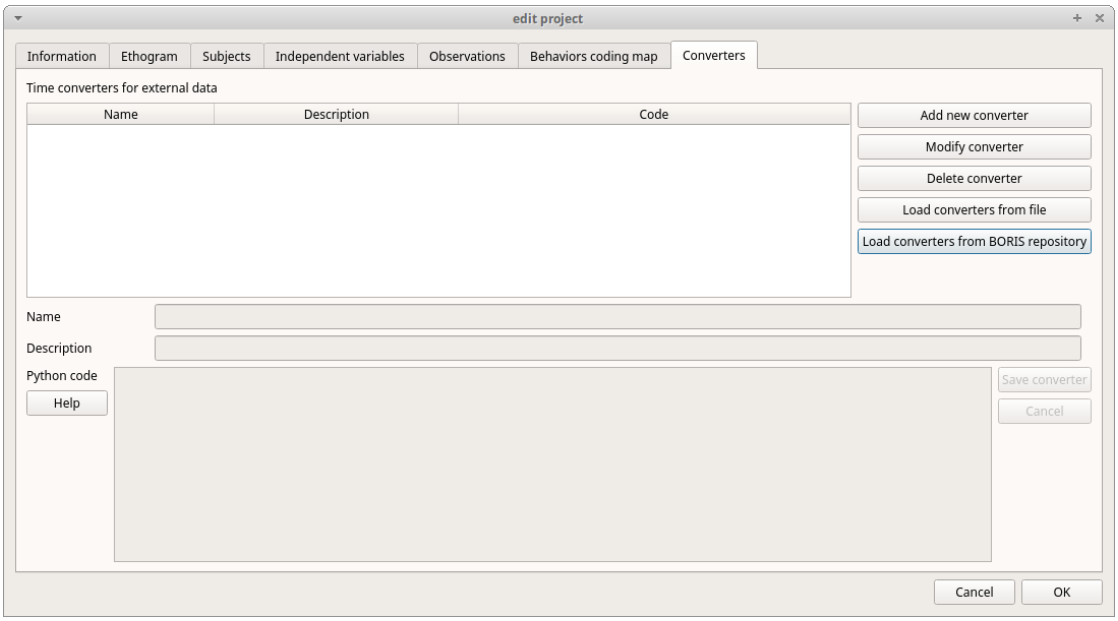


Example of an independent variable (Weather) defined as "set of values"

The predefined value must be contained in the set of value.

2.3.4 Converters' table

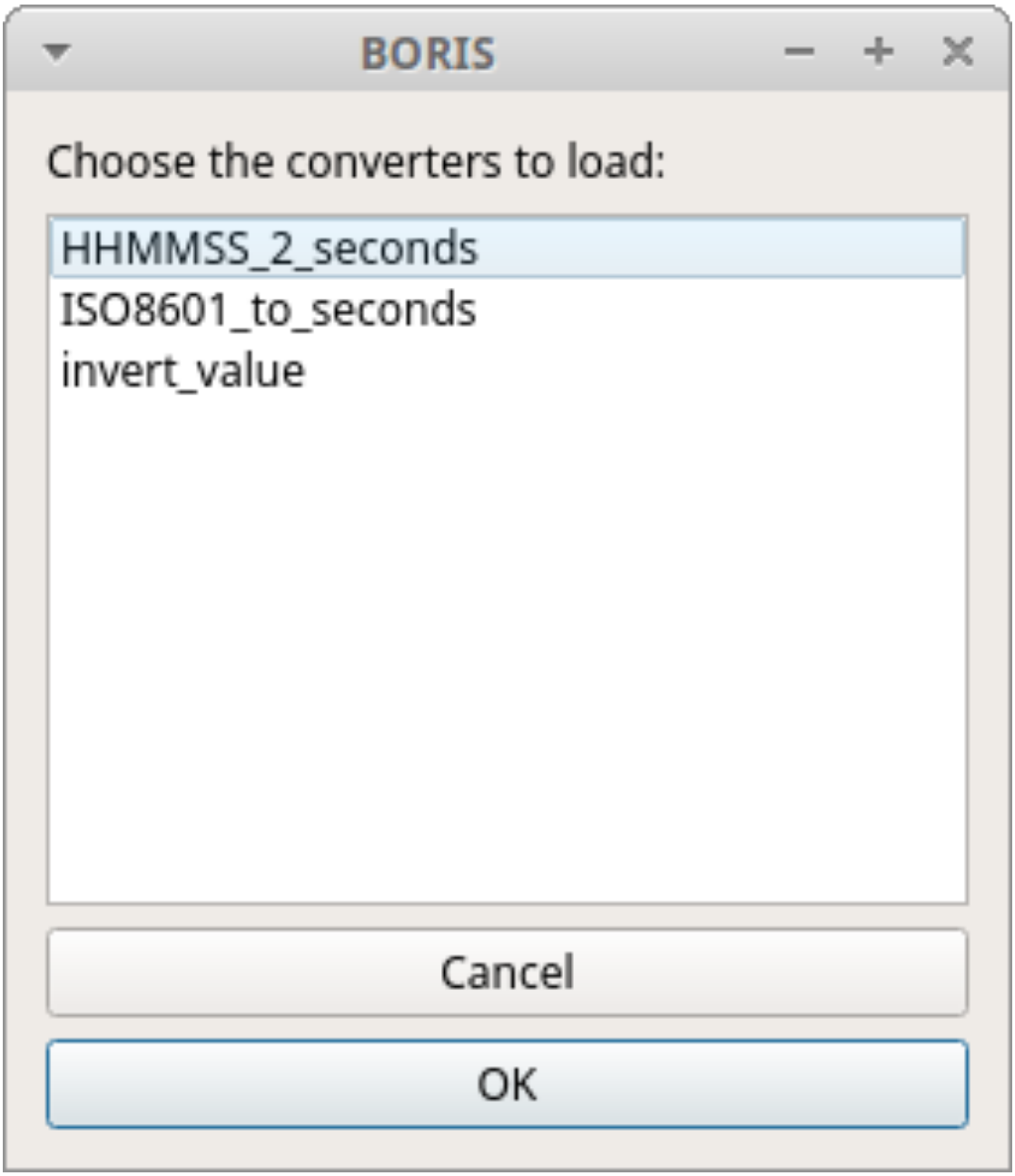
Converters are used for plotting external data when the timestamp values are not expressed in seconds. Converters can be written by the user, loaded from file or loaded from the repository of the BORIS web site (http://www.boris.unito.it/static/converters.json).



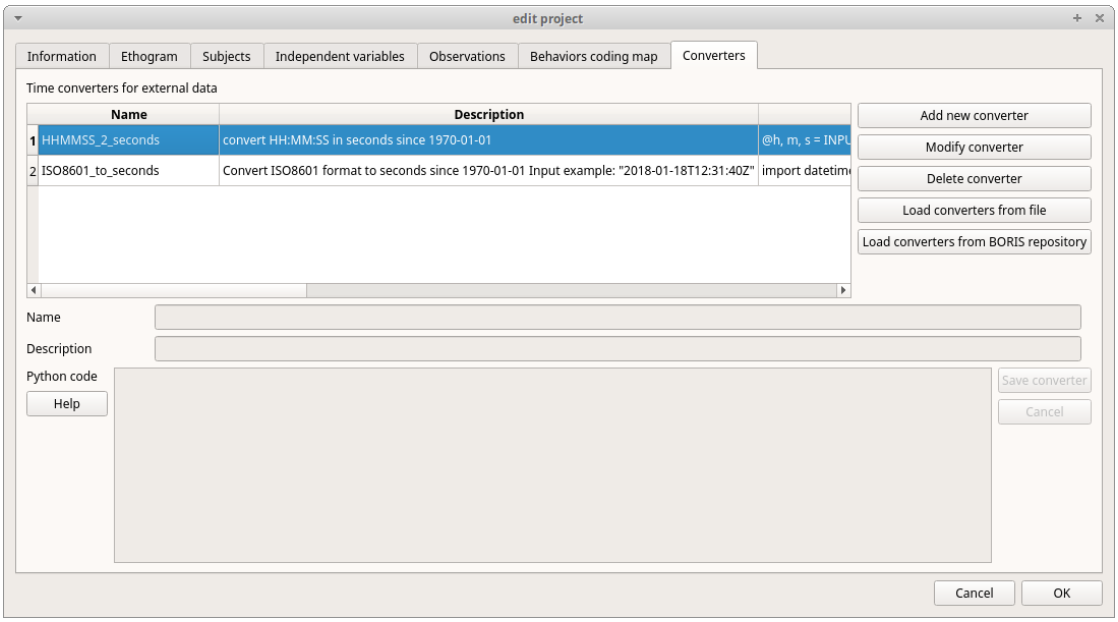
Converters tab

Load converters from BORIS web site

Click **Load converters from BORIS repository** and select the converters to be added to your project.



Converters selection from repository



Converters tab with 2 converters defined

Writing a converter

See "Converters for external data values"

The converters loaded in your project can be then selected for converting timestamp (or other values) in external data file

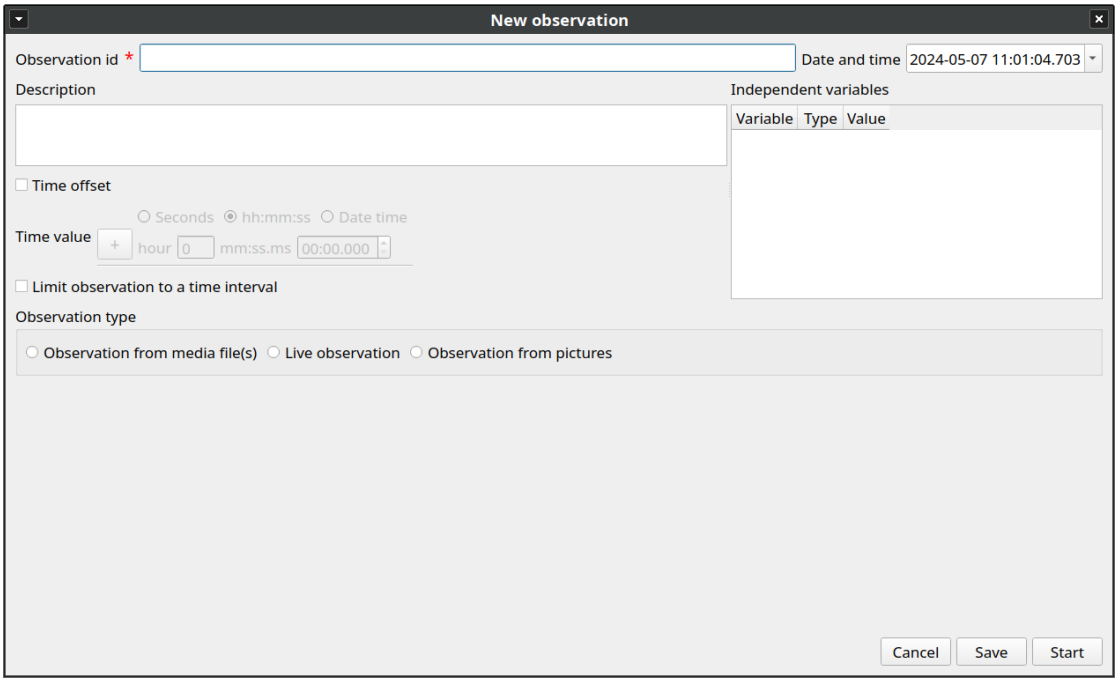
See Converters

2.4 Create a new observation

A video tutorial about making an observation is available at this link.

To create a new observation you must first Create a new project with BORIS or Open an existing project with BORIS.

Clicking on **Observations** > **New observation** will show the **New observation** window.



New observation window

This window allow adding various observation data:

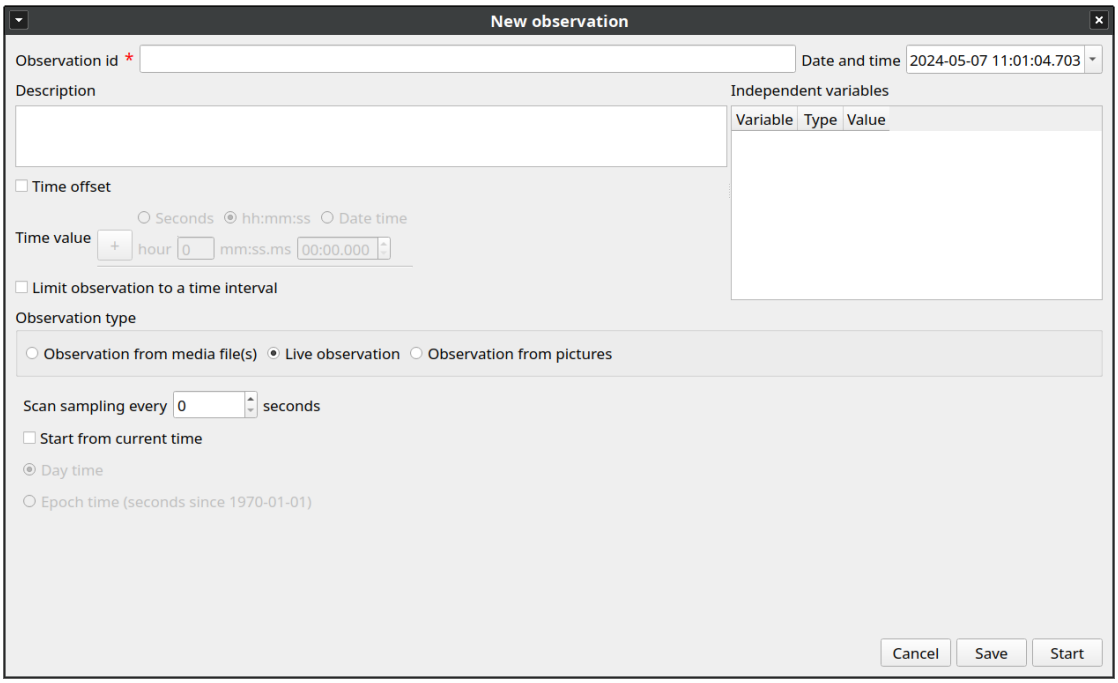
- a mandatory **Observation id** (must be unique across all observations in the open project);
- **Date**, which will be automatically set on the current date and time, but you can alternatively set this info on your media date and time, or whatever you prefer.
- **Description**, which can host all the relevant information about your observation, but can be also left empty.
- **Independent variables** (e.g. to specify factors that may influence the behaviors but will not change during the observation within a project).See the independent variables section for details.
- **Time offset**. BORIS allow specifying a time offset that can be added or subtracted from the media timecode.
- The **Limit observation to a time interval** option can be used to limit the observation to an arbitrary time interval.

You must then indicate if you want to make an observation based on **pre-recorded media (audio / video)** or a **live observation**.

2.4.1 Live observation

During the live observation BORIS will show you a timer that will be used for recording time for coded events.

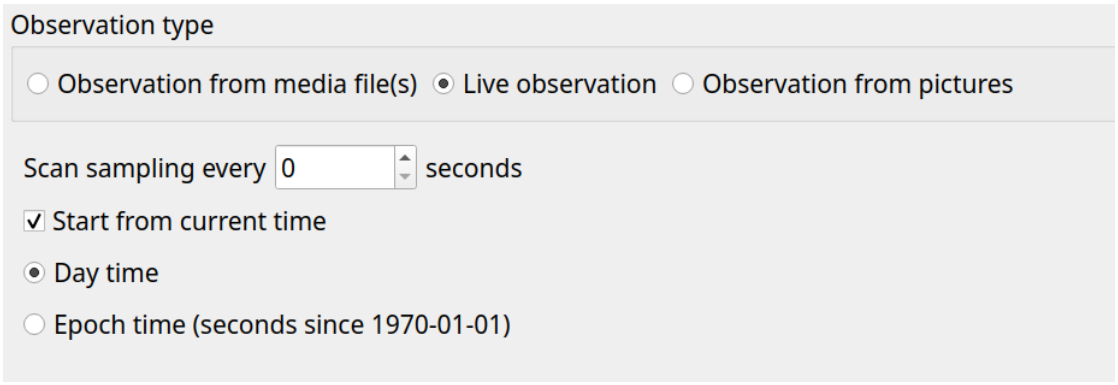
Click on the **Live observation** radio button to create a live observation.



New live observation

Start from current time

If you want that the time starts from the current time you can check the **Start from current time** checkbox.



Set a live observation to start from current time

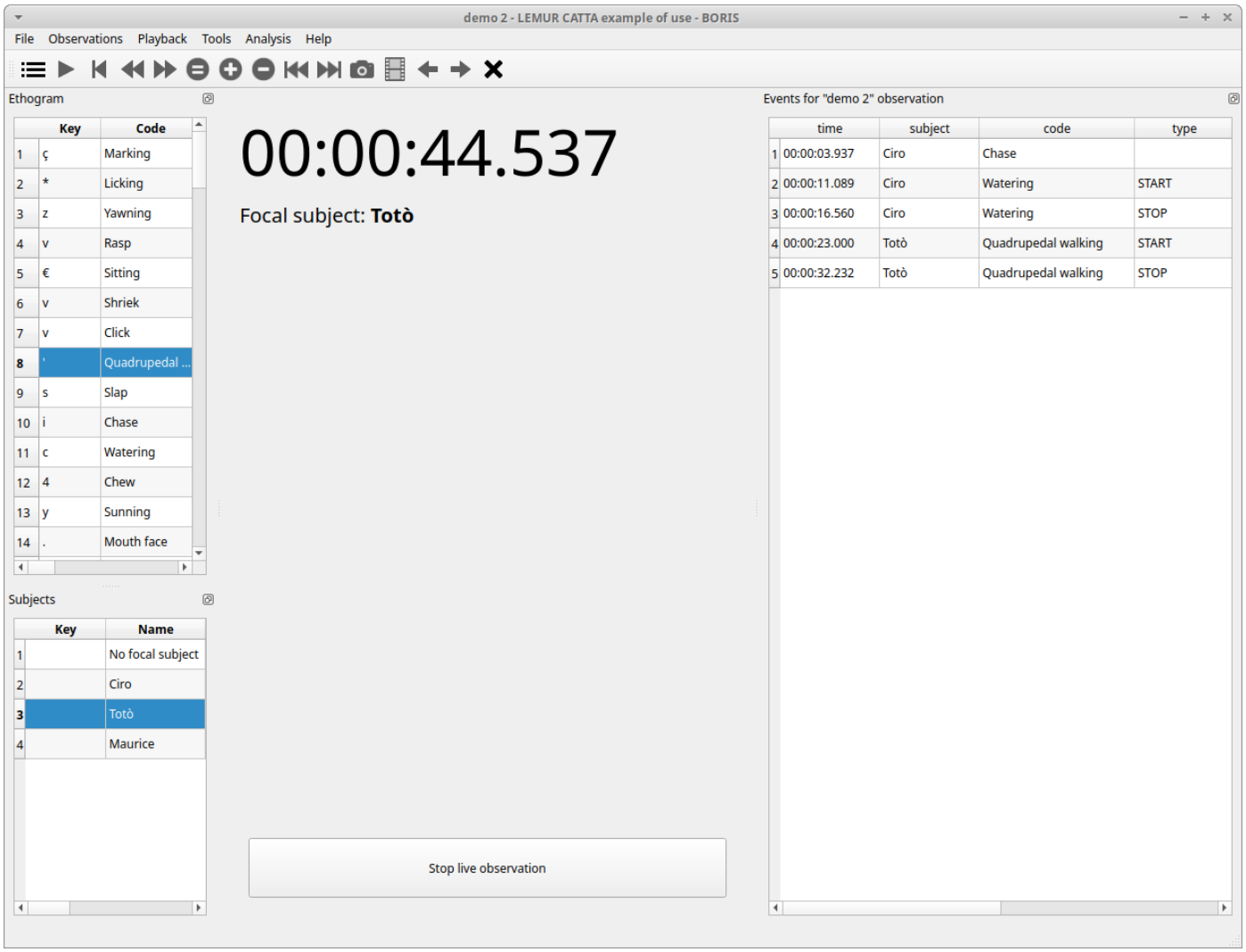
If the **Day time** option is checked the start time will be the computer current time when you will press the **Start** button.

If the **Epoch time** is checked the start time will be the number of seconds since the Jan 1st, 1970 (1970-01-01). See Unix time for details. This option is usefull for long observations (few days) or observations that start before midnight and end after.

Start the observation

Click the **Start** button to begin the live observation or **Save** to save it in the Observations list.

The main window during a live observation will look like this:

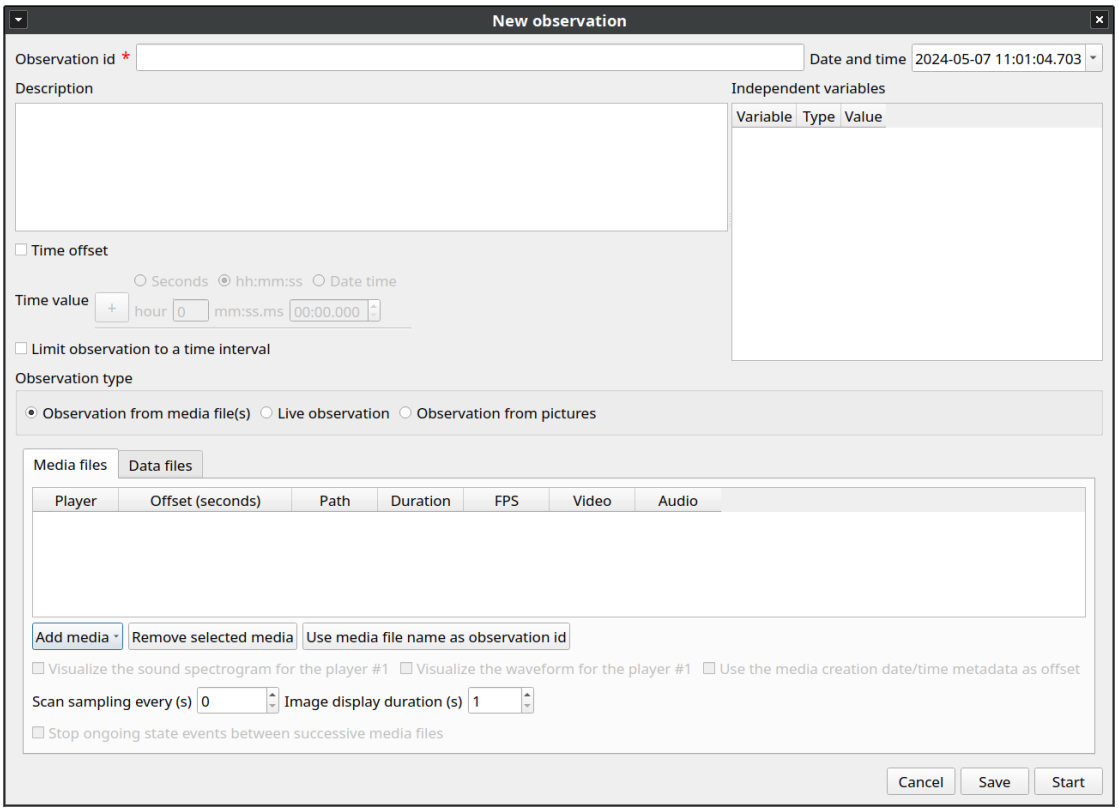


The main window during a live observation

See the Live observations section to start coding.

2.4.2 Observation from media file(s)

Click on the **Observation from media file(s)** radio button to create an observation based on one or more media files.



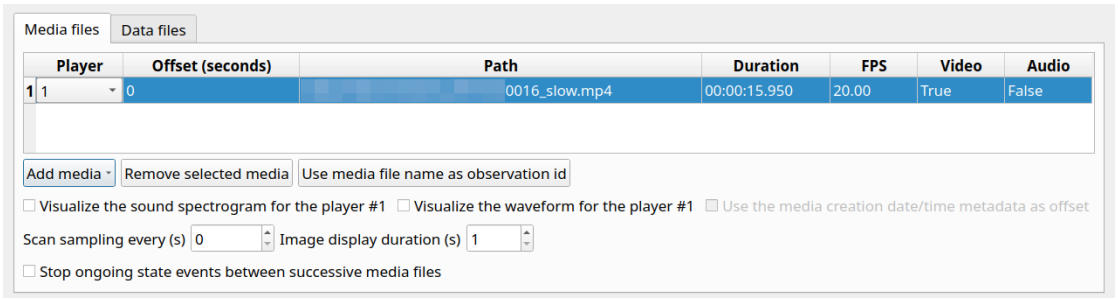
Observation from media files

The **Observation from media file(s)** tab contains 2 tabs: **Media files** and **Data files**.

Click the **Media files** tab and add one or more media files using the **Add media** button. You have 3 options:

- **with absolute path**: the whole media file path will be recorded in the project
- **with relative path**: the media file path will be recorded relatively to the position of the BORIS project file (the directory of the BORIS project file must be included). This option is useful if you have to have to move your BORIS project file on another computer.
- **from directory with absolute path**: all the media file found in the directory will be added to the playlist (the whole media file path will be recorded in the project)
- **from directory with relative path**: all the media file found in the directory will be added to the playlist (the relative media file path will be recorded in the project)

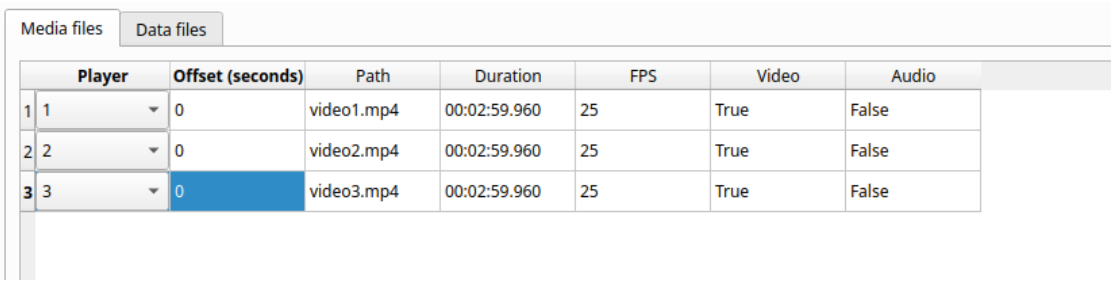
Information about the selected media file will be extracted and displayed in the media list: media file path, media duration, number of frames by second (FPS), the presence of a video stream, the presence of an audio stream.



Media files

You can choose to use the media file name as **Observation id** by clicking the **Use media file name as observation id** button.

The dropdown list in the first column allow you to choose a player (for a maximum of 8). If you want to observer more media files simultaneously you must use consecutive players (starting from 1). See example below:



Media files tab

If you have to synchronize 2 (or more) videos you can use the **Offset column** to indicate when the 2nd player should start. For example if the video loaded in the second player starts 15 seconds after the first video you have to input **15** in the **Offset** cell. If the second video starts before the first player you can set a negative value in the **Offset** cell

If you have to play sequentially many videos you have to select the same player (#1) for all video you have loaded. This means that an event occurring at time t~x~ in the media file queued as second (e.g. second_video.mp4) in the playlist will be scored as happening at time t~1~ + t~x~ (where t~1~ is the duration of the first media file, e.g. first_video.mp4).

The **Remove selected media** button can be used to remove all the selected media files.

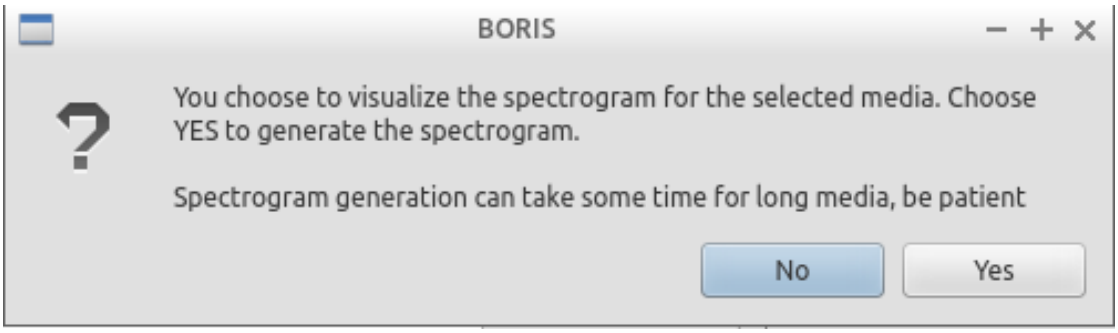
All the media types that can be played by the MPV player can be played in BORIS.

The **Use media file name as observation id** button will set the first media file name as **observation id**

Spectrogram visualization

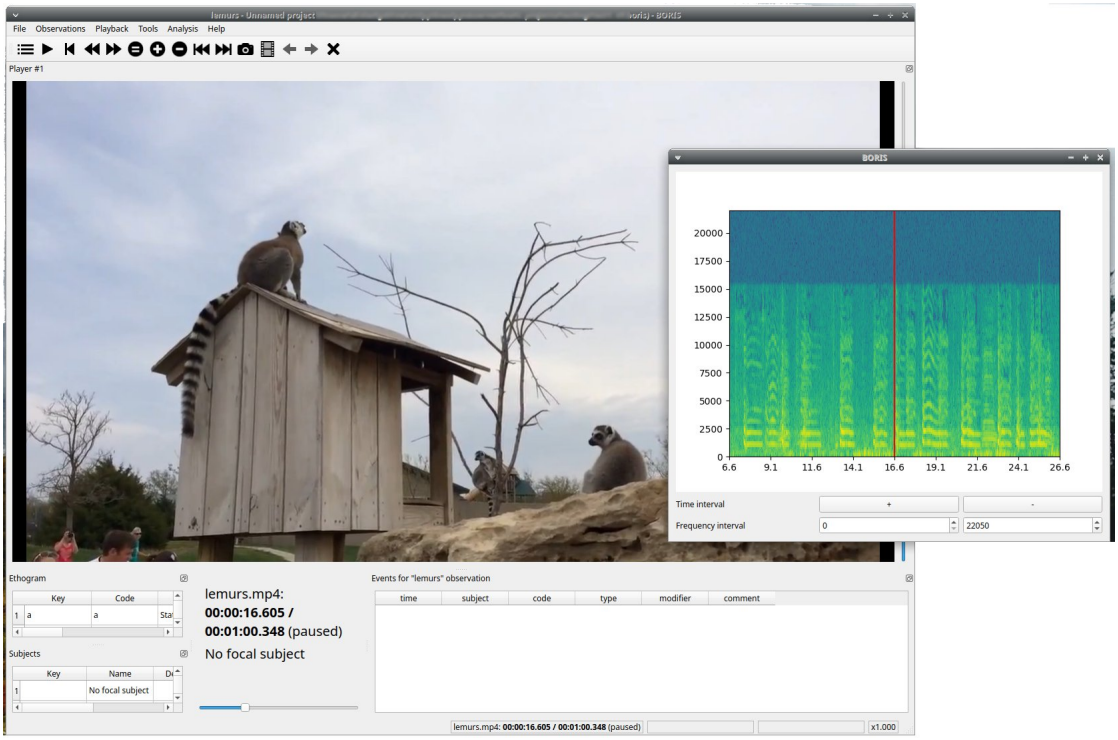
BORIS allow you to visualize the sound spectrogram during the media observation. Activate the **Visualize spectrogram** check box. BORIS will ask you to generate the spectrograms for all media files loaded in the first player.

Please note that the generation of the spectrogram can be long for long duration media files.



Spectrogram generation

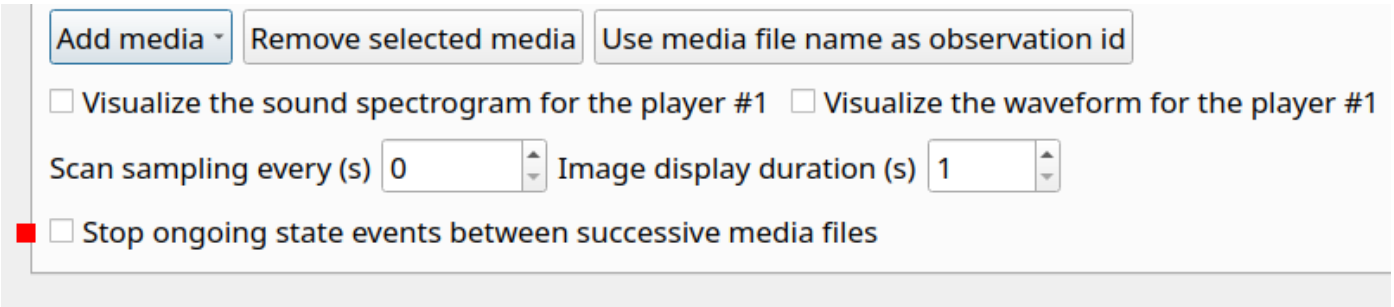
The spectrogram visualization will be synchonized to the media position during the observation.



Spectrogram visualization

Stop all state events between media files

If your media files are not contiguous, you can enable the **Stop ongoing events between successive media files** option to automatically stop state events between media files. This option may be usefull in case of video coding from camera-trap systems, for instance.

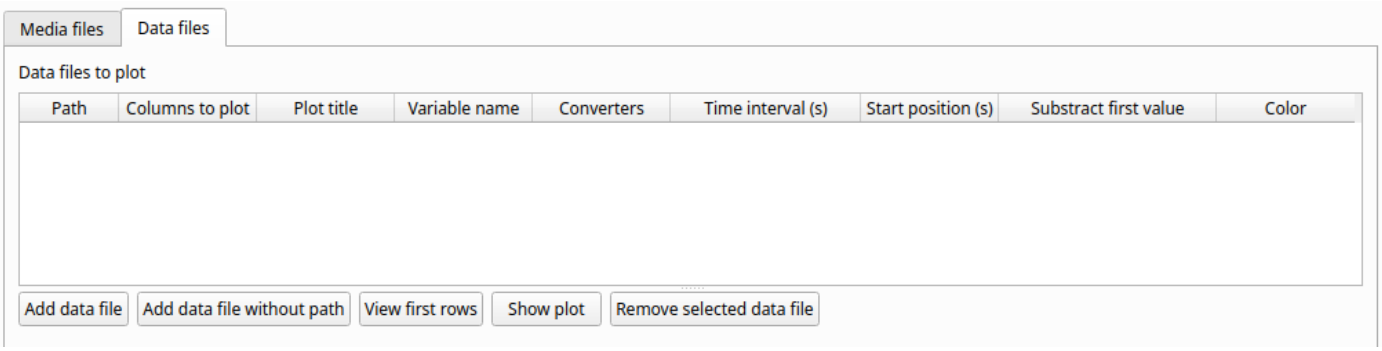


Stop state events between media files

External data files

> **Note**
>
> **At this time only 2 external data can be plotted with your media file**

You can select one or more external data files to be plotted synchronously with your media. Click the **Data files** tab and use the **Add data file** button to select a data file.



External data file table

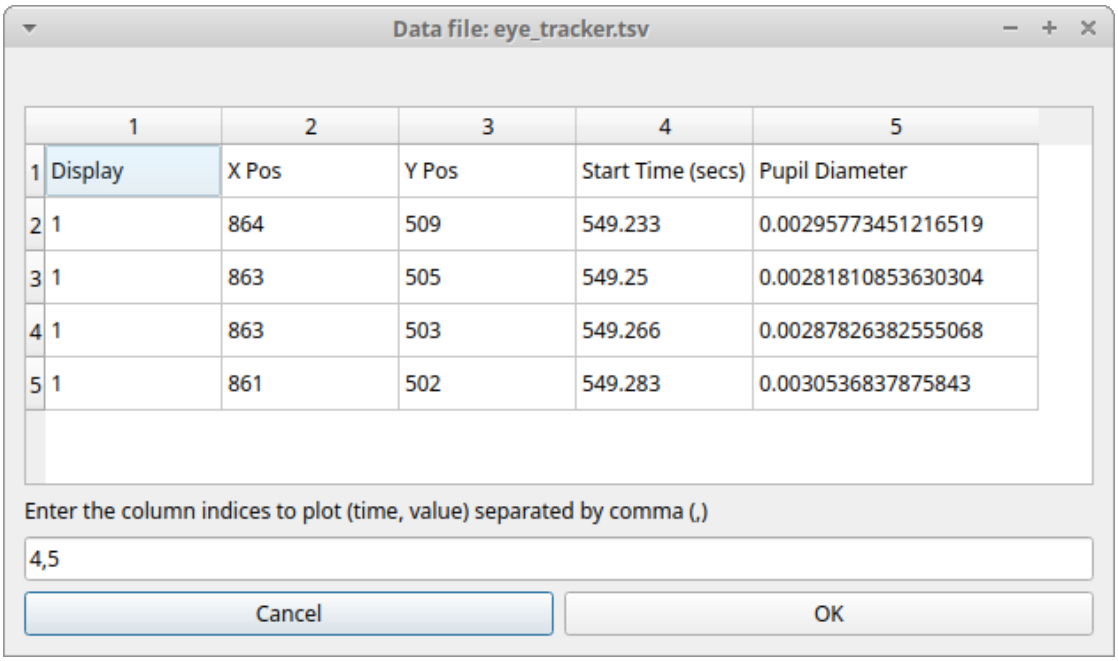
The data files must be plain text files with at least **2 columns** separated by a comma or a TAB character. One column must contain a timestamp that will be used to synchronize the plot with the media. The sampling rate can be variable.

Example of a plain text data file with 5 columns separated by comma (,):

```
Display,X Pos,Y Pos,Start Time (secs),Pupil Diameter
1,864,509,549.233,0.00295773451216519
1,863,505,549.25,0.00281810853630304
1,863,503,549.266,0.00287826382555068
1,861,502,549.283,0.0030536837875843
1,858,501,549.3,0.00308083021081984
1,856,499,549.316,0.00306266942061484
1,854,499,549.333,0.00305776367895305
[...]
```

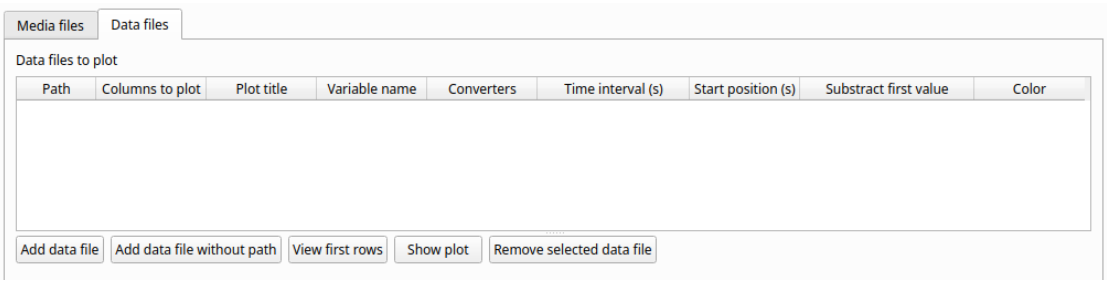
In the above example the 4th column contains the timestamp and the 5th the value to be plotted.

Input the index of the column containing the timestamp and the index of the column containing the value to be plotted. The two indices must be separated by a comma (,). Click **OK** to close the window.



Selection of columns (time, value)

A new row will be added in the data files table.



Data file table

You can modify/complete the following parameters by directly typing in the table cells:

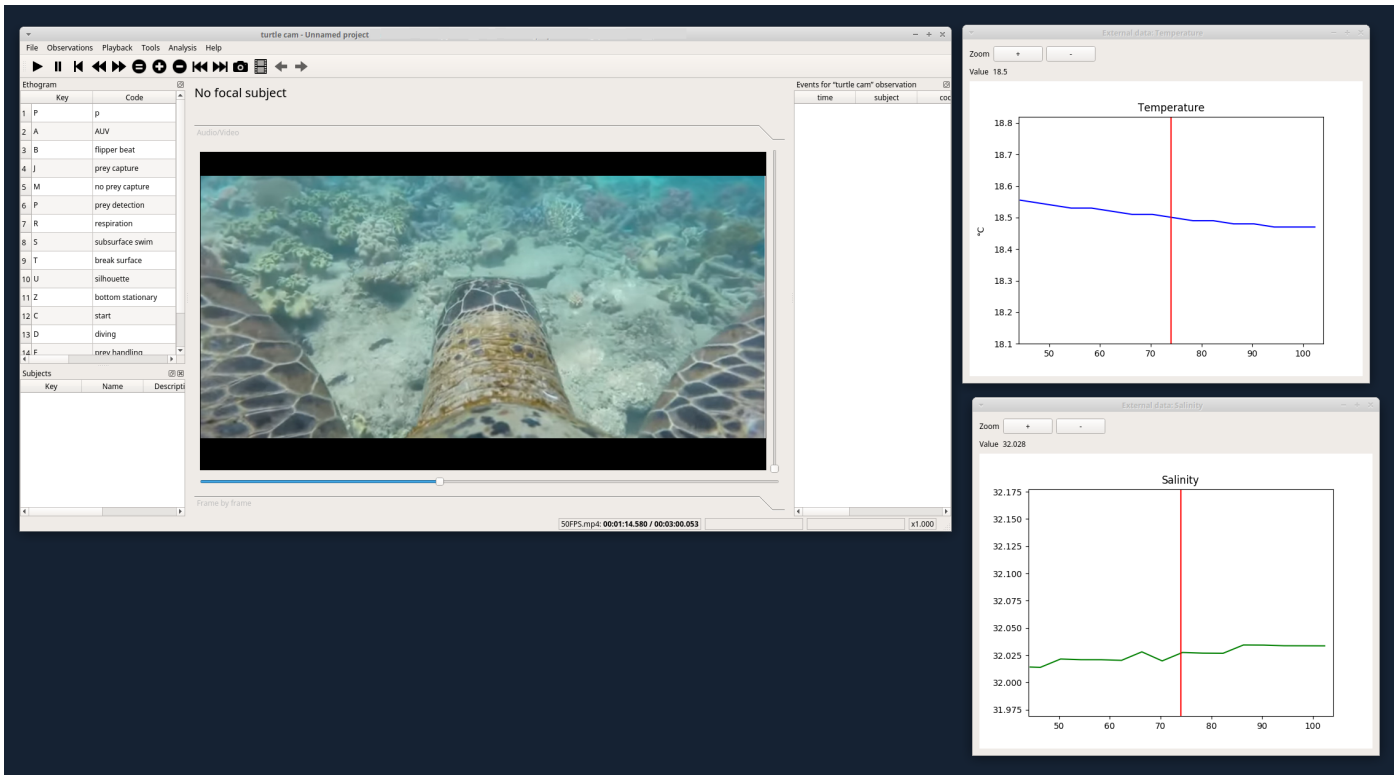
- Columns to plot
- Plot title: the title of the plot
- Variable name
- Converters: Used if the timestamp is not expressed in seconds (see below for details)
- Time interval: The time interval that will be plotted (in seconds)
- Start position: the start position of data for synchronisation with the media (in seconds)
- Substract first value: if the timestamp does not start with a 0 value you can choose to substract the first value to all timestamp values.
- Color: the color of the color

NOTE : if you want to record the value of the plotted variable in a modifier of a behavior (see Value from external data file modifier) the modifier must have the same **variable name**.

You can check if the data from file can be correctly plotted by using the **Show plot** button. If the data are compatible you will see a plot otherwise you will obtain a message with an explanation.

For now only 2 values can be plotted synchronously with your media file. The values can come from the same file or from two different files.

During the observation tha values you have selected in external data files will be plotted synchronously with your media file.



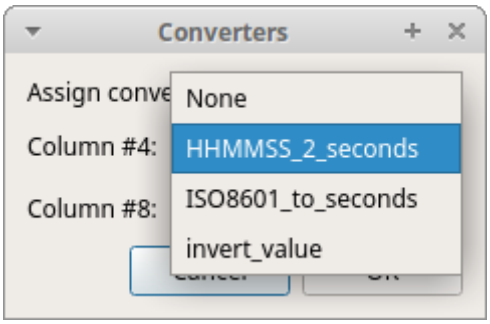
Observation with 2 values plotted from external data files: Temperature and salinity

CONVERTERS

If the values in the timestamp column are not expessed in seconds (like 12.45) but in another format (HH:MM:SS, MM:SS, ISO8601 2018-01-18T12:31:40Z ...) you must use a converter that will convert the current format in seconds.

See the Converters' table in the project configuration.

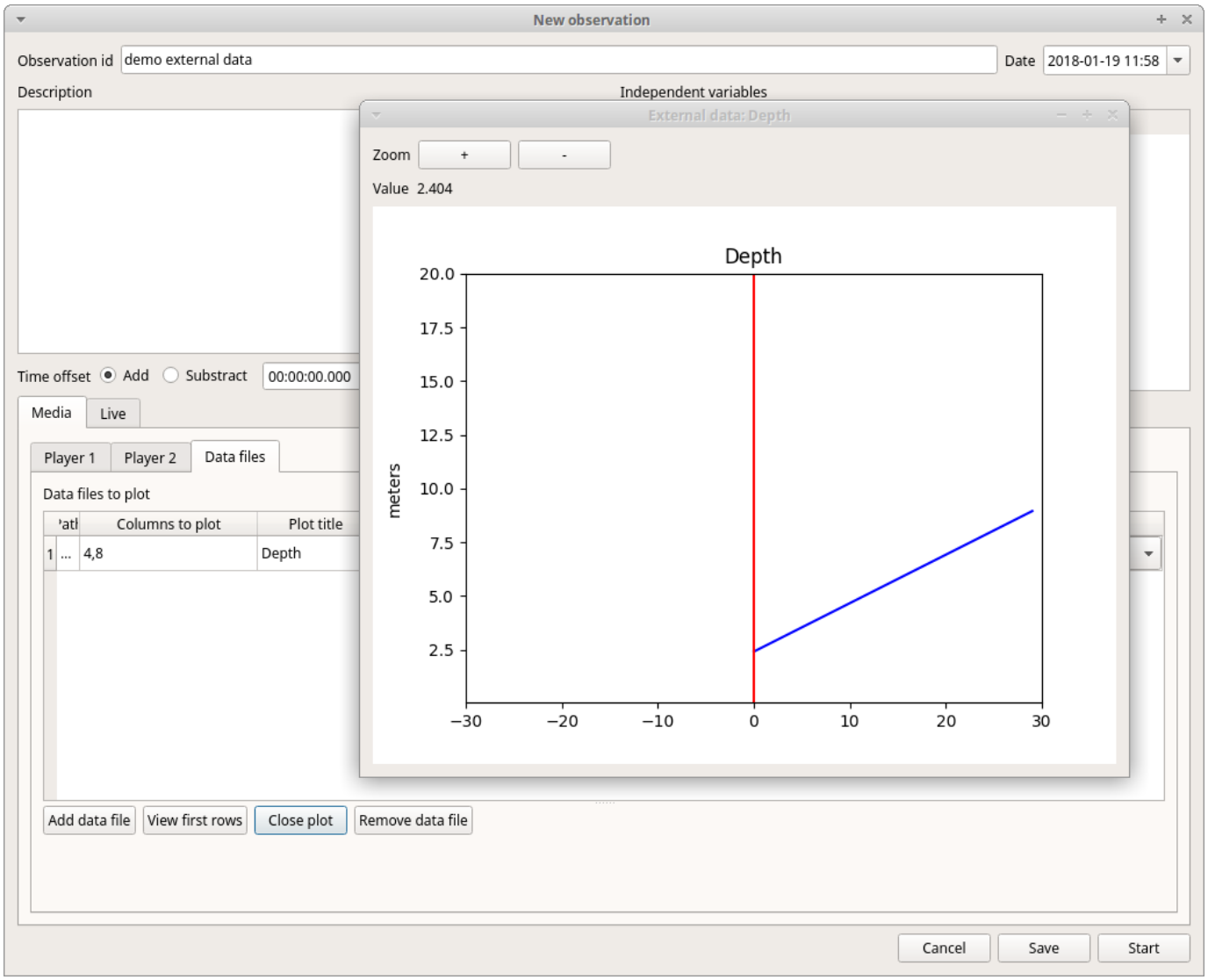
A **double-click** on the converters cell will allow you to select a converter for each column to be plotted



Data files to plot

	Path	Columns to plot	Plot title	Variable name	Converters	Time interval (s)	Start position (s)
1	/home/user/external_data2.csv	4,8	Depth	meters	{'4': 'HHMMSS_2_seconds'}	60	0

Use the **Show plot** button to verify if your external data can be plotted without problem. The **Close plot** button will close the plot window.



Converters can also be used to convert values that are not time value.

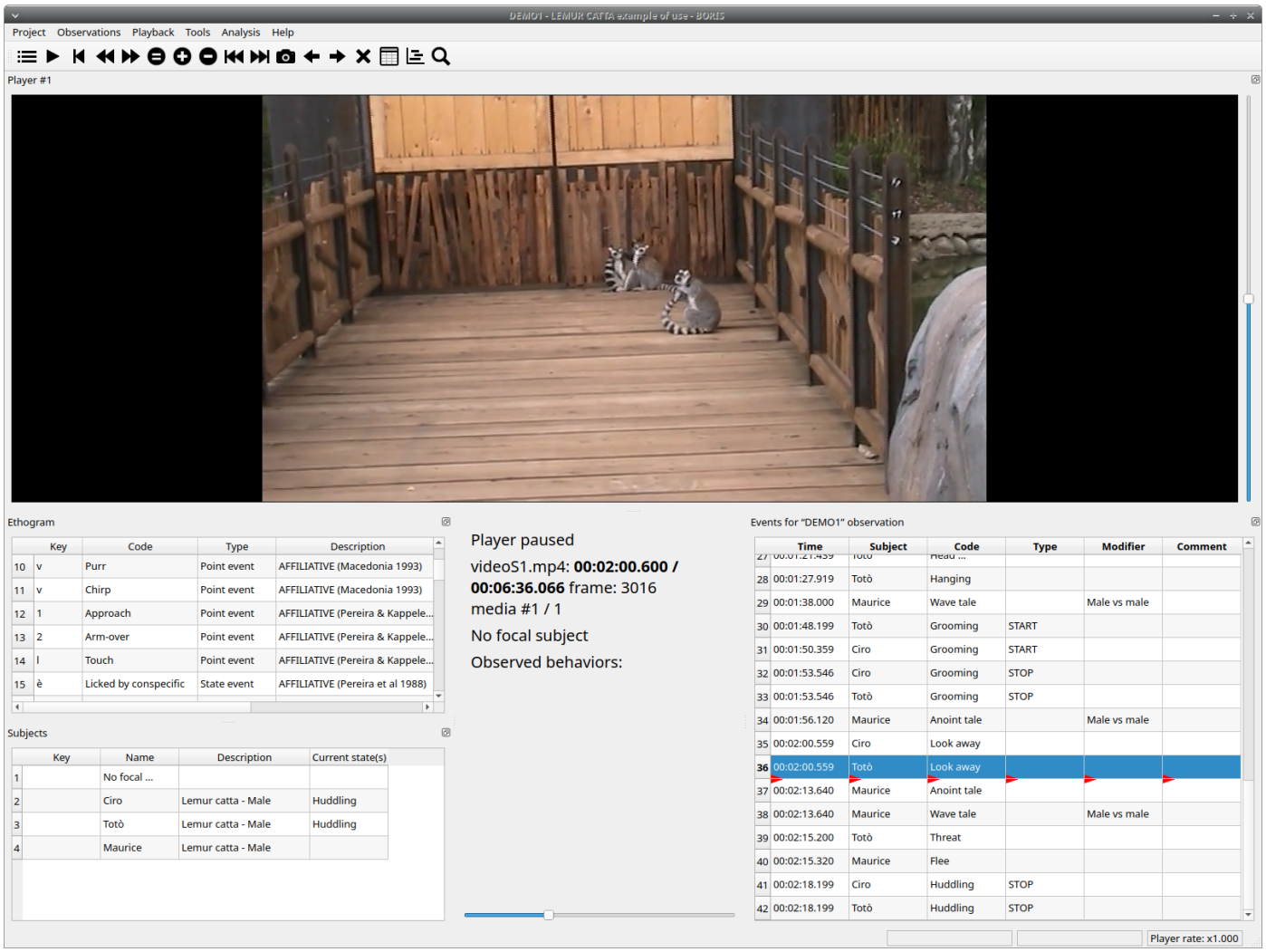
Example of a converter for inverting value:

```
OUTPUT = - float(INPUT)
```

Start the observation

Click the **Start** button to start coding. The **Observation** window will be closed and you'll be transferred to the main **BORIS** window. If you do not want to start the observation click the **Save** button. The observation will be saved in the observations list.

The main window during the observation of a single media file will look like this:

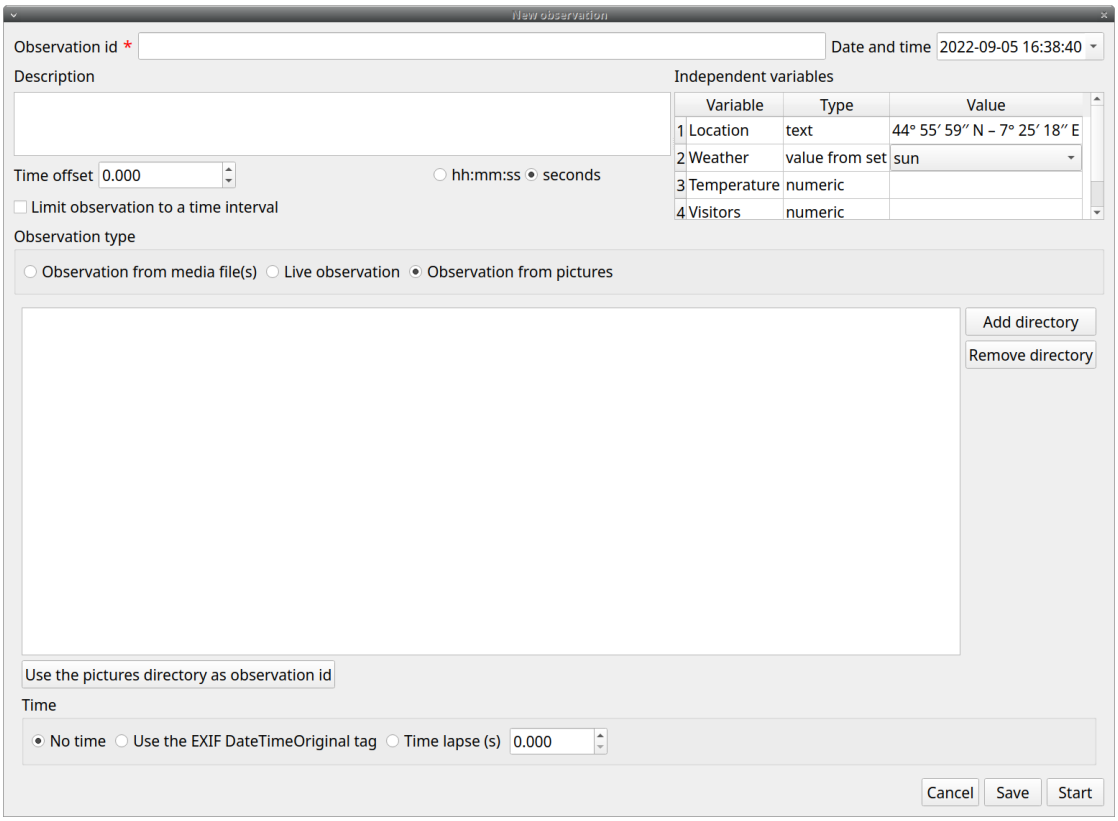


The main window during the observation of one video

See the media coding section to start coding.

2.4.3 Observation from pictures

Click on the **Observation from pictures** radio button to create an observation based on pictures.



Observation from pictures tab

Use the **Add directory** to select a directory containing the pictures you want to code. You can select many directories, in this case the pictures will be browsed in the order of the directories were added.

The **Use the pictures directory as observation id** button will set the directory name as **observation id**

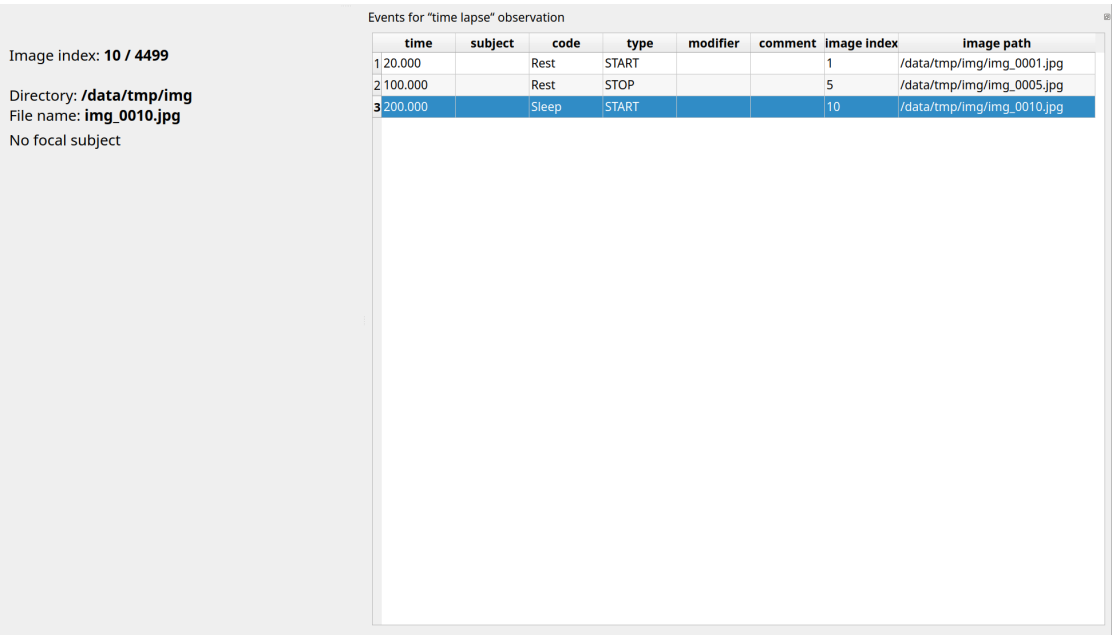
Time

You have 3 option for the coding time:

- No time: no time will be recorded. The image index (the position of image in the directory) and the image file path will be recorded.
- Use the EXIF DateTimeOriginal tag: the time will be extracted from the EXIF tag of the picture file (if any).
- Time lapse: this option will let you define the time interval between the pictures.

Start the observation

Click the **Start** button to start coding. The **Observation** window will be closed and you\'ll be transferred to the main **BORIS** window. If you do not want to start the observation click the **Save** button. The observation will be saved in the observations list.



The main window during the coding of a picture directory

See the media coding section to start coding.

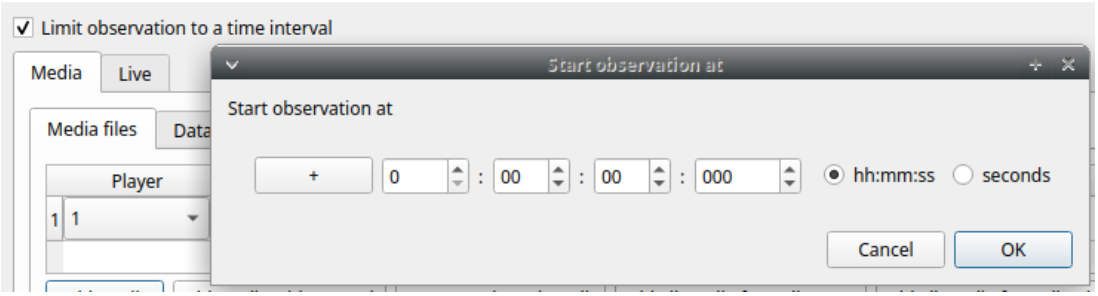
2.4.4 Various options

Scan sampling (Live and media observation)

You can select a time for **Scan sampling** observation. In this case the timer will stop at every time offset you indicated and all the coded events will have the same time value.

Limit observation to a time interval (Live and media observation)

This option can be used to limit the observation to a time interval for live or media based observations.



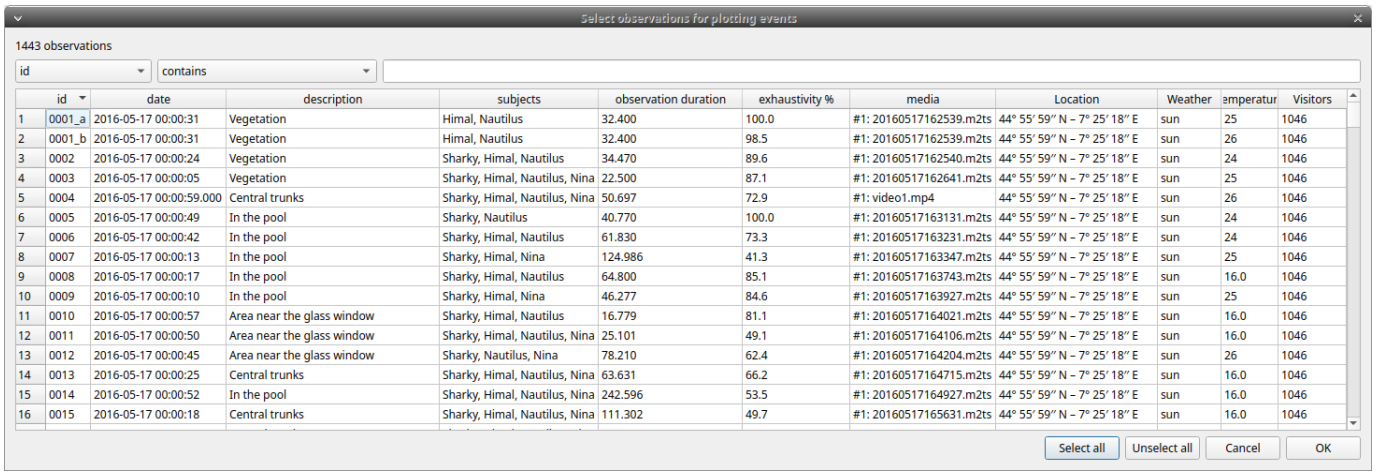
2.5 Managing observations

2.5.1 Observations list

The **Observations** > **Observations list** will show you all the observations contained in the current BORIS project.

The following values are displayed:

- the observation id (**id**)
- the **description** of observation
- the coded subjects (**subjects**)
- the **observation duration** (as the difference between the last recorded event and the first one)
- the percent of **exhaustivity** of the coding (as the sum of the length of the coded events divided by the observation duration)
- the **media** file path, **LIVE** in case of live observation, the pictures directory path in case of observation from pictures
- the values of the independent variables (if defined)

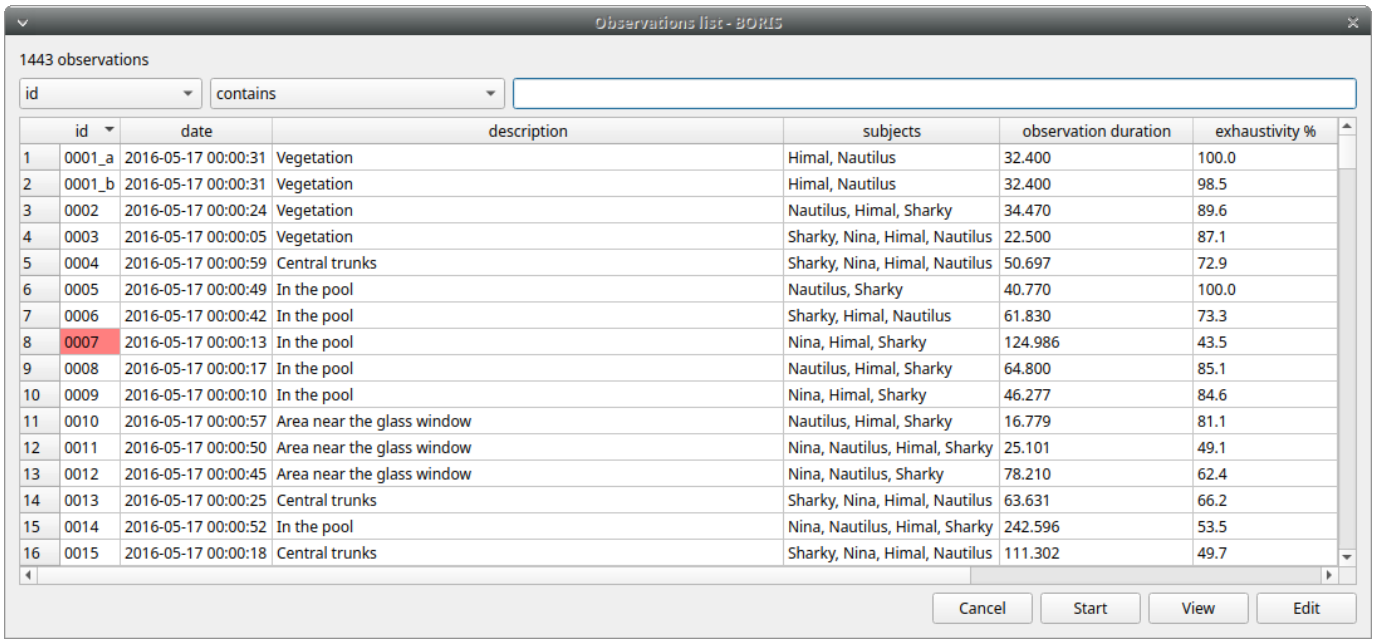


	id	date	description	subjects	observation duration	exhaustivity %	media	Location	Weather	emperatur	Visitors
1	0001_a	2016-05-17 00:00:31	Vegetation	Himal, Nautilus	32.400	100.0	#1: 20160517162539.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25	1046
2	0001_b	2016-05-17 00:00:31	Vegetation	Himal, Nautilus	32.400	98.5	#1: 20160517162539.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
3	0002	2016-05-17 00:00:24	Vegetation	Sharky, Himal, Nautilus	34.470	89.6	#1: 20160517162540.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
4	0003	2016-05-17 00:00:05	Vegetation	Sharky, Himal, Nautilus, Nina	22.500	87.1	#1: 20160517162641.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25	1046
5	0004	2016-05-17 00:00:59.000	Central trunks	Sharky, Himal, Nautilus, Nina	50.697	72.9	#1: video1.mp4	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
6	0005	2016-05-17 00:00:49	In the pool	Sharky, Nautilus	40.770	100.0	#1: 20160517163131.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
7	0006	2016-05-17 00:00:42	In the pool	Sharky, Himal, Nautilus	61.830	73.3	#1: 20160517163231.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
8	0007	2016-05-17 00:00:13	In the pool	Sharky, Himal, Nina	124.986	41.3	#1: 20160517163347.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25	1046
9	0008	2016-05-17 00:00:17	In the pool	Sharky, Himal, Nautilus	64.800	85.1	#1: 20160517163743.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046
10	0009	2016-05-17 00:00:10	In the pool	Sharky, Himal, Nina	46.277	84.6	#1: 20160517163927.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25	1046
11	0010	2016-05-17 00:00:57	Area near the glass window	Sharky, Himal, Nautilus	16.779	81.1	#1: 20160517164021.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046
12	0011	2016-05-17 00:00:50	Area near the glass window	Sharky, Himal, Nautilus, Nina	25.101	49.1	#1: 20160517164106.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046
13	0012	2016-05-17 00:00:45	Area near the glass window	Sharky, Nautilus, Nina	78.210	62.4	#1: 20160517164204.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
14	0013	2016-05-17 00:00:25	Central trunks	Sharky, Himal, Nautilus, Nina	63.631	66.2	#1: 20160517164715.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046
15	0014	2016-05-17 00:00:52	In the pool	Sharky, Himal, Nautilus, Nina	242.596	53.5	#1: 20160517164927.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046
16	0015	2016-05-17 00:00:18	Central trunks	Sharky, Himal, Nautilus, Nina	111.302	49.7	#1: 20160517165631.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046

The observations can be sorted by clicking in the desired column header (alphabetic order ascending or descending).

Checking the observations

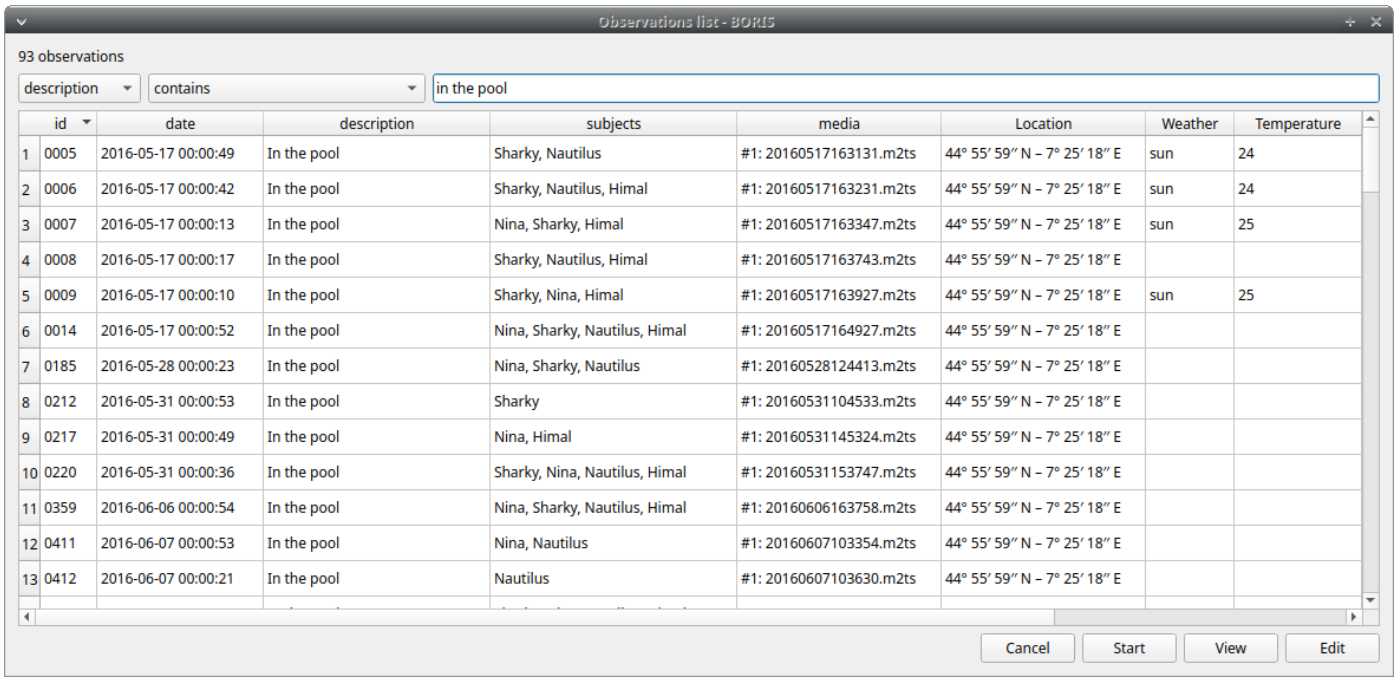
The status of observation is displayed in the first column (**id**). If the background of this column is **red** the observations has one or more UNPAIRED state events. These UNPAIRED observations will not be analyzed. See Fix unpaired state events for details.



Filtering the observations

The observations list can be filtered selecting a field and a condition in the drop-list boxes.

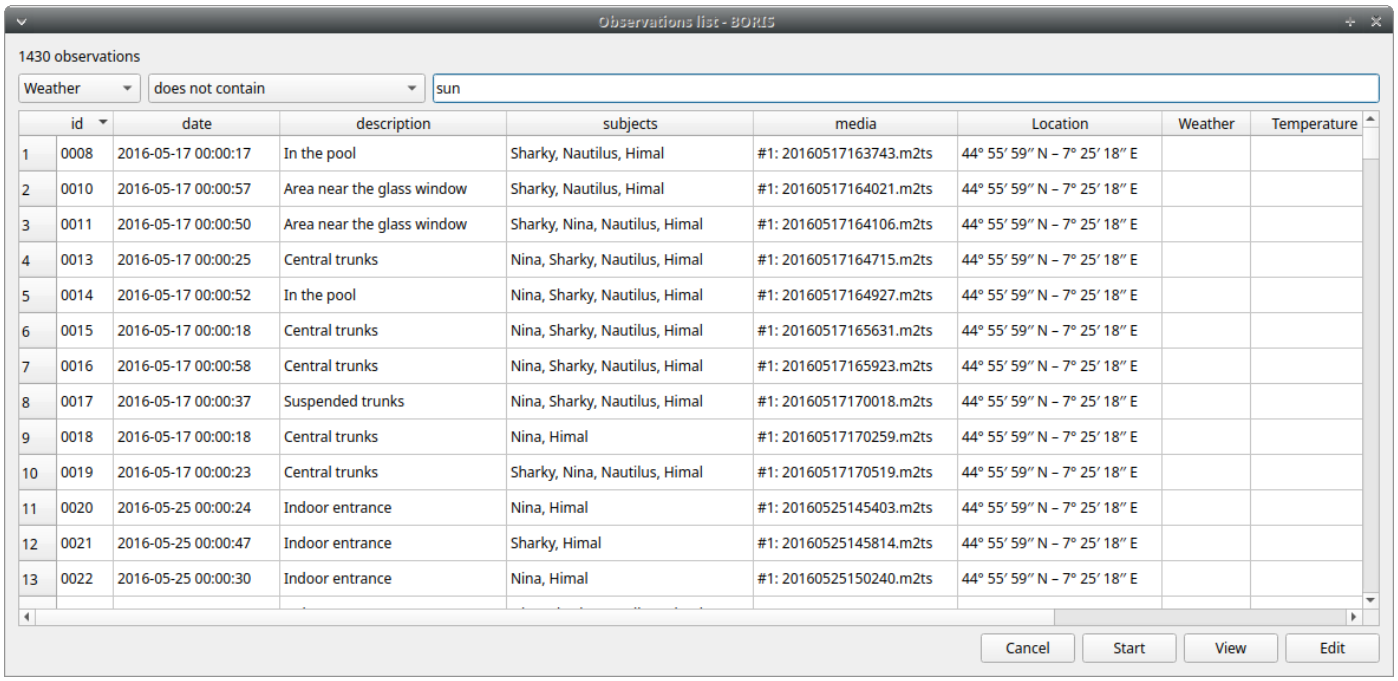
In the following example observations are filtered: only observations with **description** containing the **In the pool** subject are shown:



Observations list

Observations can be filtered with **Independent variables** values.

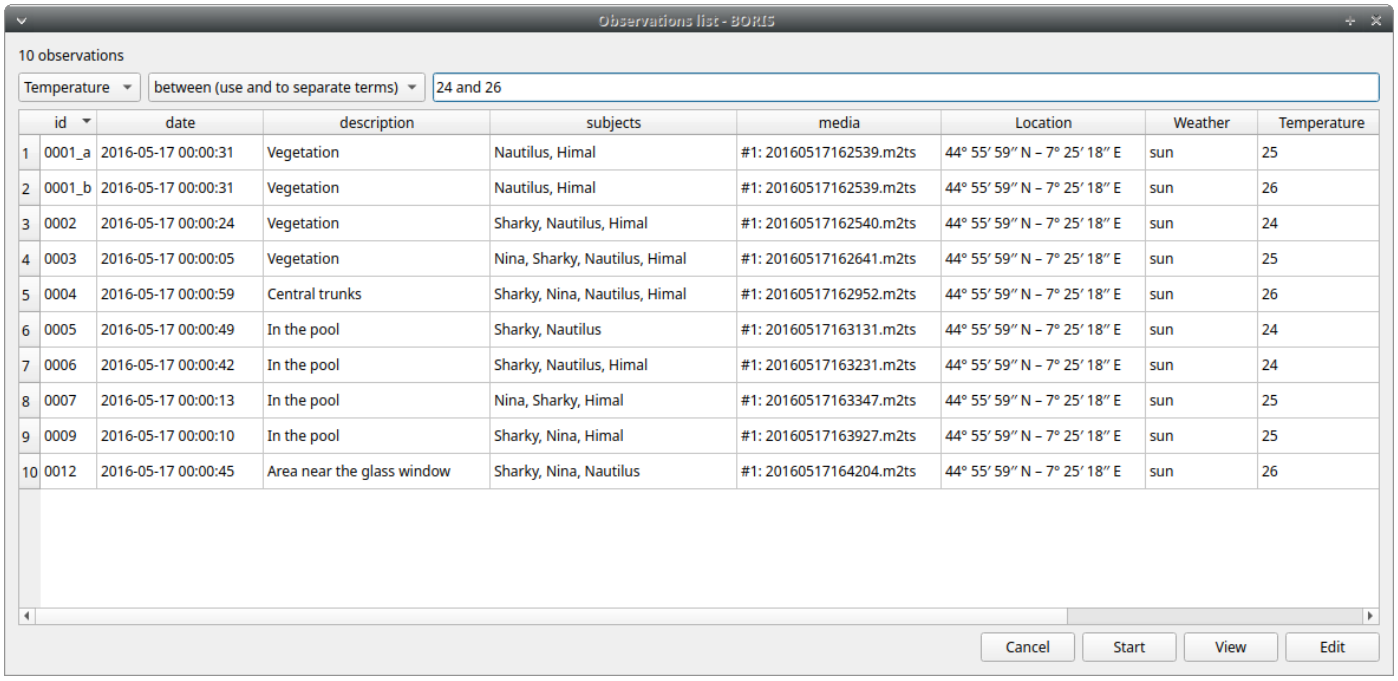
The following example displays only the observations that do not contain "Sunny" in the **Weather** independent variable :



	id	date	description	subjects	media	Location	Weather	Temperature
1	0008	2016-05-17 00:00:17	In the pool	Sharky, Nautilus, Himal	#1: 20160517163743.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
2	0010	2016-05-17 00:00:57	Area near the glass window	Sharky, Nautilus, Himal	#1: 20160517164021.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
3	0011	2016-05-17 00:00:50	Area near the glass window	Sharky, Nina, Nautilus, Himal	#1: 20160517164106.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
4	0013	2016-05-17 00:00:25	Central trunks	Nina, Sharky, Nautilus, Himal	#1: 20160517164715.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
5	0014	2016-05-17 00:00:52	In the pool	Nina, Sharky, Nautilus, Himal	#1: 20160517164927.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
6	0015	2016-05-17 00:00:18	Central trunks	Nina, Sharky, Nautilus, Himal	#1: 20160517165631.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
7	0016	2016-05-17 00:00:58	Central trunks	Nina, Sharky, Nautilus, Himal	#1: 20160517165923.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
8	0017	2016-05-17 00:00:37	Suspended trunks	Nina, Sharky, Nautilus, Himal	#1: 20160517170018.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
9	0018	2016-05-17 00:00:18	Central trunks	Nina, Himal	#1: 20160517170259.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
10	0019	2016-05-17 00:00:23	Central trunks	Sharky, Nina, Nautilus, Himal	#1: 20160517170519.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
11	0020	2016-05-25 00:00:24	Indoor entrance	Nina, Himal	#1: 20160525145403.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
12	0021	2016-05-25 00:00:47	Indoor entrance	Sharky, Himal	#1: 20160525145814.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		
13	0022	2016-05-25 00:00:30	Indoor entrance	Nina, Himal	#1: 20160525150240.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E		

Observations list

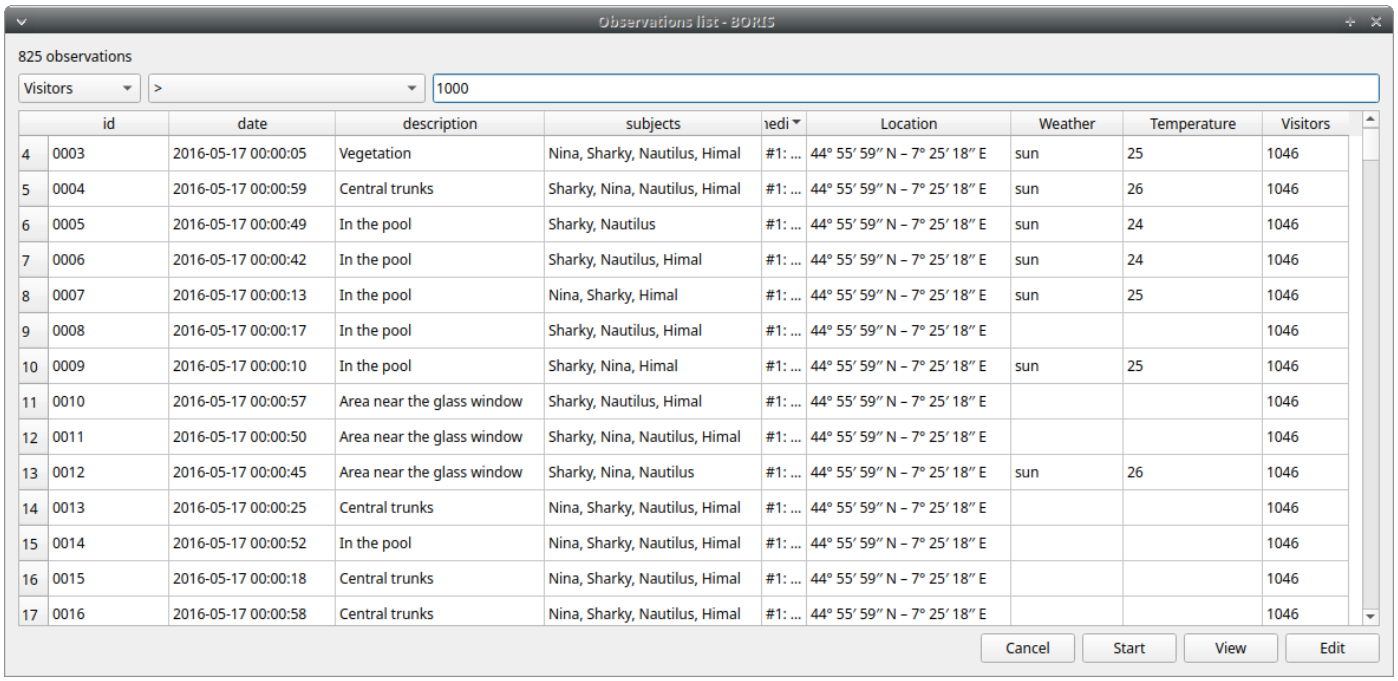
Observations with a value of **Temperature** independent variable between 18 and 22:



	id	date	description	subjects	media	Location	Weather	Temperature
1	0001_a	2016-05-17 00:00:31	Vegetation	Nautilus, Himal	#1: 20160517162539.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25
2	0001_b	2016-05-17 00:00:31	Vegetation	Nautilus, Himal	#1: 20160517162539.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26
3	0002	2016-05-17 00:00:24	Vegetation	Sharky, Nautilus, Himal	#1: 20160517162540.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24
4	0003	2016-05-17 00:00:05	Vegetation	Nina, Sharky, Nautilus, Himal	#1: 20160517162641.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25
5	0004	2016-05-17 00:00:59	Central trunks	Sharky, Nina, Nautilus, Himal	#1: 20160517162952.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26
6	0005	2016-05-17 00:00:49	In the pool	Sharky, Nautilus	#1: 20160517163131.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24
7	0006	2016-05-17 00:00:42	In the pool	Sharky, Nautilus, Himal	#1: 20160517163231.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24
8	0007	2016-05-17 00:00:13	In the pool	Nina, Sharky, Himal	#1: 20160517163347.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25
9	0009	2016-05-17 00:00:10	In the pool	Sharky, Nina, Himal	#1: 20160517163927.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25
10	0012	2016-05-17 00:00:45	Area near the glass window	Sharky, Nina, Nautilus	#1: 20160517164204.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26

Observations list

Observations with a value of **Visitors** independent variable greater than 1000:



	id	date	description	subjects	medi	Location	Weather	Temperature	Visitors
4	0003	2016-05-17 00:00:05	Vegetation	Nina, Sharky, Nautilus, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25	1046
5	0004	2016-05-17 00:00:59	Central trunks	Sharky, Nina, Nautilus, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
6	0005	2016-05-17 00:00:49	In the pool	Sharky, Nautilus	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
7	0006	2016-05-17 00:00:42	In the pool	Sharky, Nautilus, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
8	0007	2016-05-17 00:00:13	In the pool	Nina, Sharky, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25	1046
9	0008	2016-05-17 00:00:17	In the pool	Sharky, Nautilus, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E			1046
10	0009	2016-05-17 00:00:10	In the pool	Sharky, Nina, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25	1046
11	0010	2016-05-17 00:00:57	Area near the glass window	Sharky, Nautilus, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E			1046
12	0011	2016-05-17 00:00:50	Area near the glass window	Sharky, Nina, Nautilus, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E			1046
13	0012	2016-05-17 00:00:45	Area near the glass window	Sharky, Nina, Nautilus	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
14	0013	2016-05-17 00:00:25	Central trunks	Nina, Sharky, Nautilus, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E			1046
15	0014	2016-05-17 00:00:52	In the pool	Nina, Sharky, Nautilus, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E			1046
16	0015	2016-05-17 00:00:18	Central trunks	Nina, Sharky, Nautilus, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E			1046
17	0016	2016-05-17 00:00:58	Central trunks	Nina, Sharky, Nautilus, Himal	#1: ...	44° 55′ 59″ N – 7° 25′ 18″ E			1046

Observations list

2.5.2 Delete observations from project

The observations can be deleted from the project using the following procedure:

Observations > **Remove observations**

Select the observations you want to delete.

Click the **OK** button and confirm the deletion.

Please note that the deletion is irreversible, the deleted observations can not be restored.

It's a good idea to back up your project before proceeding with removing observations.

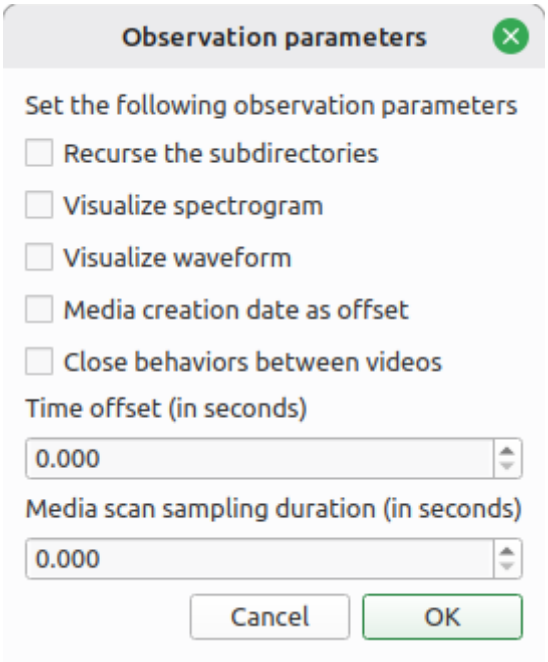
2.5.3 Create observations in bulk

Observations from media file can be created from a directory of media file:

Observations > **Create observations**

Choose the directory

Select the parameters



Parameters for creating observations

The ID of each created observation will be the path to the media file.

2.5.4 Import observations

The **Observations** > **Import observations** option allows to import observations. Two formats are available for importing observations:

From a BORIS project file

Choose the BORIS project file and then the observations to import. BORIS will check if observations with same id are already existing in the current project. BORIS will also check if behaviors and/or subjects used in the imported observations are not defined in the current project.

From a spreadsheet file

Observations can be imported from:

- OpenDocument (ODS)
- Microsoft-Excel (XLSX)

Choose the spreadsheet file

2.5.5 Export a list of observations

This option allow to export the selected observation as a list in various formats (CSV, TSV, ODS, XLSX, HTML):

Observations > **Export observations list**

The dataframe will contains the following columns:

- Observation id
- Date
- Description
- Subjects
- Media files/Live observation
- independents variables

2.6 Coding

When looking at the BORIS main window, the window title bar shows the **Observation id** - **Project name** - **BORIS**. The media (the first in the queue) will be loaded in the media player and paused.

On macOS, the video is not embedded into the main window of BORIS so it is **VERY IMPORTANT to click on the BORIS main window** to score as usual.

2.6.1 Media based coding

The toolbar

The BORIS toolbar

List of observations

Play (become **Pause** when media is played)

Rewind reset your media at the beginning

Fast backward jumps for n seconds backward in your media (See preferences)

Fast forward jumps for n seconds forward in your media (See preferences)

Set the playback speed to 1x

Increase the playback speed (See preferences)

Decrease the playback speed (See preferences)

Jump to the previous media file

Jump to the next media file

Take a snapshot of current video or frame

Move on frame back

Move one frame forward

Close current observation

Real time plot of events

Time budget of the current observation

Plot events of the current observation

Plot the time budget of the current observation

Geometric measurements

Find in events

Explore project

Preferences

The toolbar can be resized using the **Preferences** > **Interface** option.

The media can also be controlled by special keyboard keys:

`⇞ Page Up` Switch to the next media

`⇟ Page Down` switch to the previous media

`↑ Up` Jump forward in the current media

`↓ Down` Jump backward in the current media

`⇱ Home` Increase the playback speed (See general preferences to set the step value)

`⇲ End` Decrease the playback speed (See general preferences to set the step value)

`⌫ Backspace` Set the playback speed to 1x

`← Left` Go to the previous frame

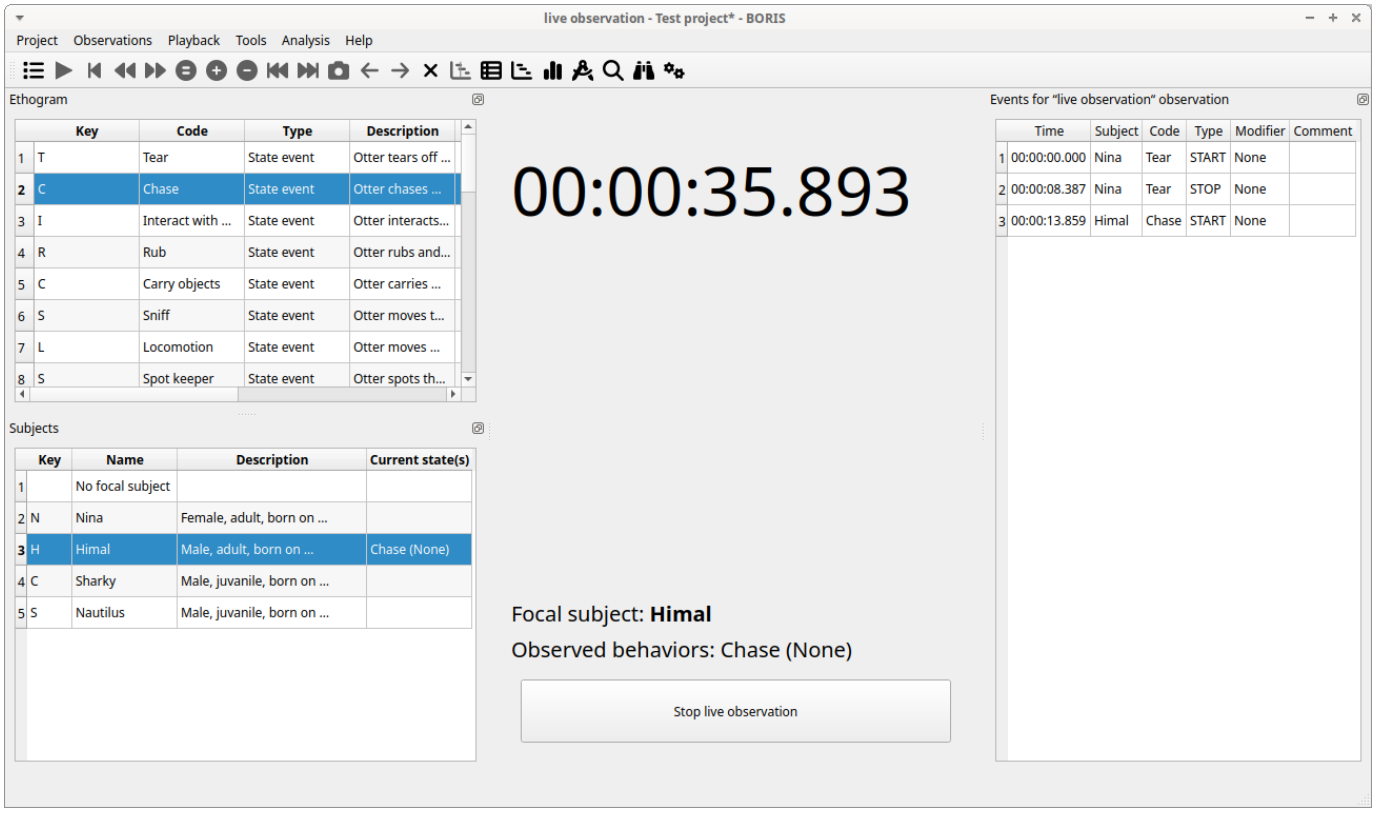
`→ Right` Go to the next frame

2.6.2 Live observations

During a live observation the media control toolbar is disabled.

Press the **Start live observation** button to start your observation. If some events are already coded BORIS will ask you for deleting them.

A timer will be displayed. The events will be recorded in the events widget.



2.6.3 **Ethogram** table in the main window

Ethogram

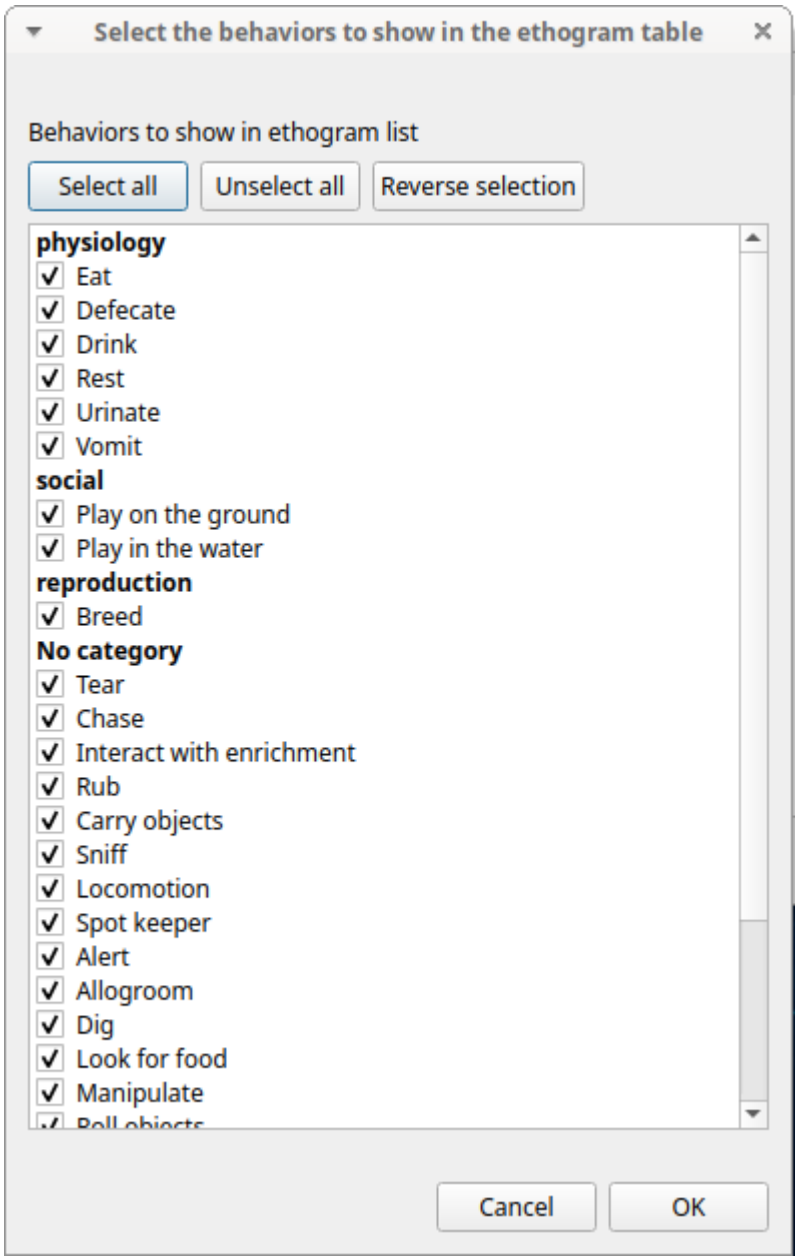
	Key	Code	Type	Description	Color	Category	Modifiers	Excluded
1	T	Tear	State event	Otter tears off vegetation			{'0': {'name': '', 'type': 0, ...	Alert,Allogroom,Breed,Carry ...
2	C	Chase	State event	Otter chases other animals			{'0': {'name': '', 'type': 0, ...	Alert,Allogroom,Breed,Carry ...
3	I	Interact with ...	State event	Otter interacts with enrichment			{'0': {'name': '', 'type': 0, ...	Alert,Allogroom,Breed,Carry ...
4	R	Rub	State event	Otter rubs and rolls itself upon a surface; may be ...			{'0': {'name': '', 'type': 0, ...	Alert,Allogroom,Breed,Carry ...
5	C	Carry objects	State event	Otter carries objects or food by holding them ...			{'0': {'name': '', 'type': 0, ...	Alert,Allogroom,Breed,Chase,Defecate,Dig,Drink,Eat,In...
6	S	Sniff	State event	Otter moves the nose and head movement back ...			{'0': {'name': '', 'type': 0, ...	Alert,Allogroom,Breed,Carry ...
7	L	Locomotion	State event	Otter moves from place to place			{'0': {'name': '', 'type': 0, ...	Alert,Allogroom,Breed,Carry ...
8	S	Spot keeper	State event	Otter spots the keeper in or out the enclosure			{'0': {'name': '', 'type': 0, ...	Alert,Allogroom,Breed,Carry ...
9	A	Alert	State event	Otter is stationary and directs its attention toward...				Allogroom,Breed,Carry ...
10	Q	Allogroom	State event	Otter licks or scratches with forepaws or hind-paw...				Alert,Breed,Carry ...
11	D	Dig	State event	Otter uses front legs to move sand, stones on the ...				Alert,Allogroom,Breed,Carry ...

The **Ethogram** table provide the user with the list of behaviors defined in the **Ethogram**. It can be used to record an event by double clicking on the corresponding row. The **Key** column indicates the keyboard key assigned to each behavior (if any). Pressing a key will record the corresponding behavior (that will appear in the **Events** table).

The behaviors shown in the ethogram widget can be filtered:

Right-click on ethogram widget > **Filter behaviors**

Check/Uncheck single behaviors or double-click on the behavioral category.



2.6.4 **Subjects** table in the main window

Subjects

	Key	Name	Description	Current state(s)
1		No focal subject		
2	N	Nina	Female, adult, born on 10/03/2013 in Ostrava biopark (Czech Republic), bright white snout	
3	H	Himal	Male, adult, born on 04/30/2014 in Amneville biopark (France), larger tale, bigger than the others	
4	C	Sharky	Male, juvanile, born on 10/30/2015 in Zoom biopark (Italy), bright brown nose and fur	
5	S	Nautilus	Male, juvanile, born on 10/30/2015 in Zoom biopark (Italy), dark brown nose and fur	

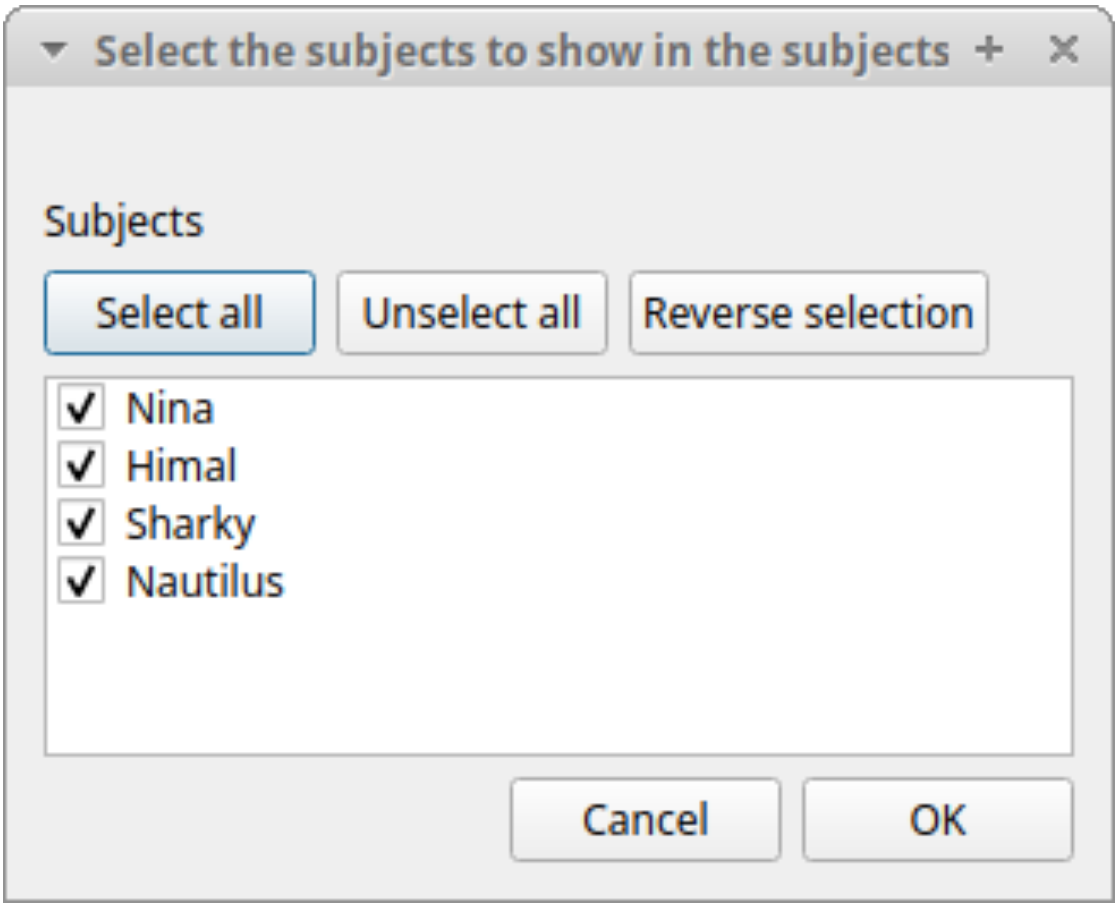
The subjects table in main window

The **Subjects** widget provide the user with the list of subjects defined in the **Subject** tab in the **Project** window. It can be used to add information about the focal subject on the recorded behaviors by double clicking on the corresponding row. When a subject is selected his/her name appears above the media player. The **Key** column indicates the keyboard key assigned to each subject (if any).

The subjects shown in the subjects widget can be filtered:

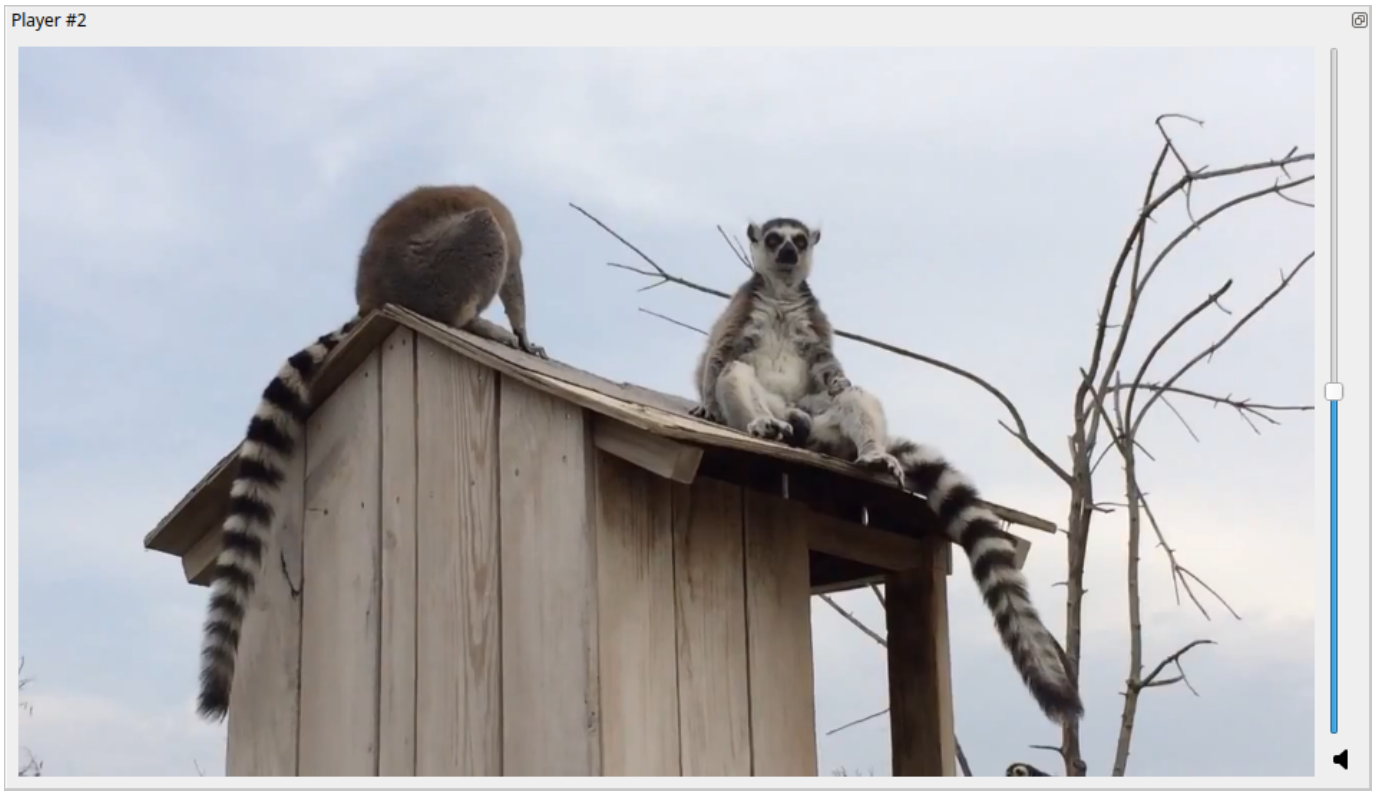
Right-click on subjects widget > **Filter subjects**

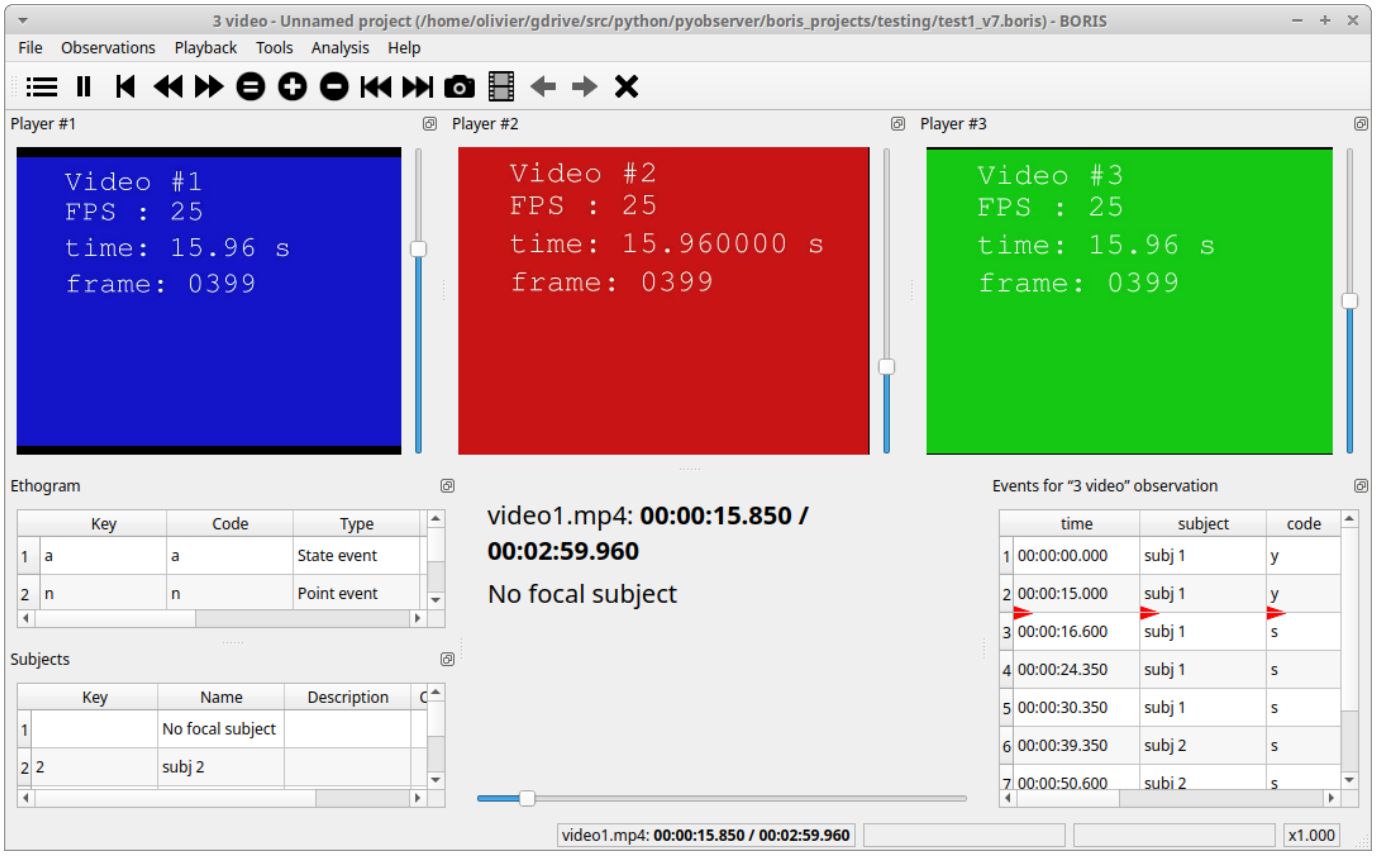
Check/Uncheck the subjects to show/hide them on the subjects' table.



Filter subjects in subjects table

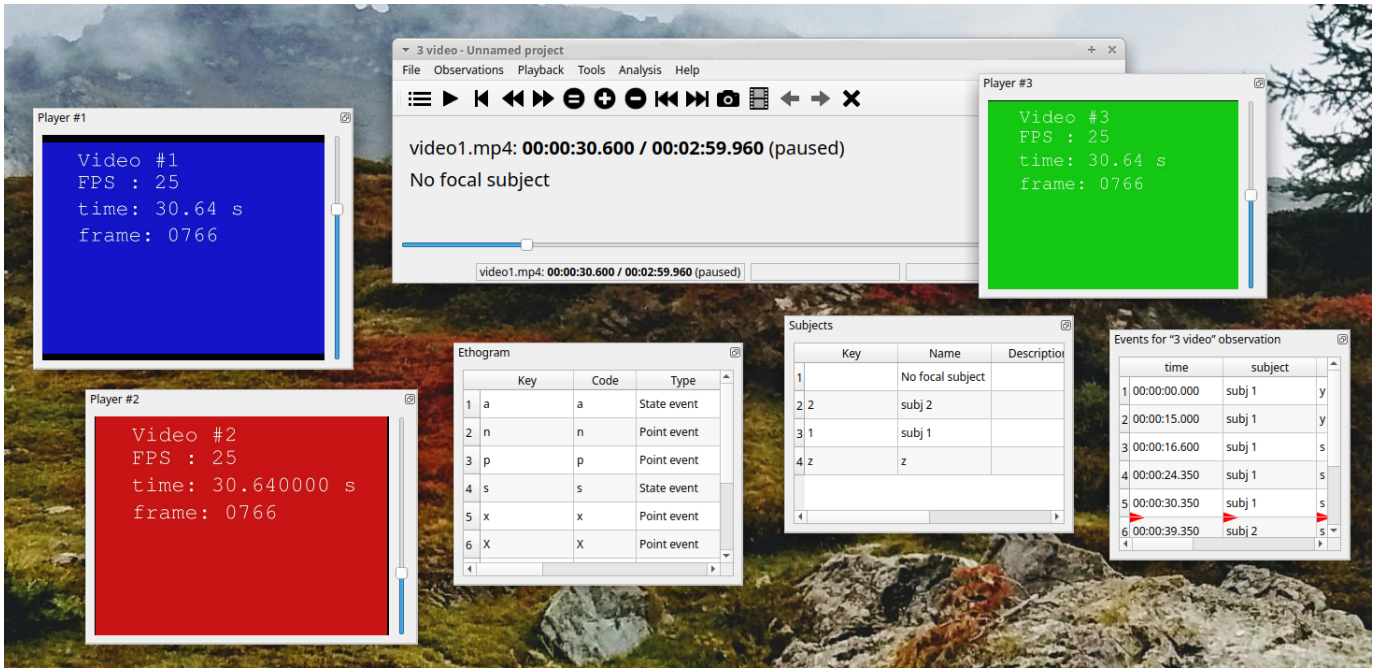
2.6.5 The **media player**





You can adjust the media position with the horizontal slider. Each media player is equipped with its own audio volume control, which can be adjusted using the vertical slider located on the right side of the player and a mute/unmute button for easy sound management.

The arrangement of different widgets can be personalized to suit your preferences. In the screenshot provided, all widgets are detached from the main program window. This feature is particularly useful when using a multi-monitor setup.



2.6.6 The events table

The **events table** shows all the recorded behaviors (events).

The displayed parameters (organized in columns) depend of the type of the observation:

Observation from media file

Events for "0004" observation

	Time	Frame index	Subject	Code	Type	Modifier	Comment
1	00:00:08.253	207	Himal	Locomotion	START	Run	
2	00:00:08.329	209	Nautilus	Locomotion	START	Run	
3	00:00:10.400	260	Sharky	Swim	START		
4	00:00:11.779	295	Himal	Locomotion	STOP	Run	
5	00:00:12.778	320	Nina	Alert	START		
6	00:00:13.789	345	Nautilus	Locomotion	STOP	Run	
7	00:00:13.790	345	Nautilus	Locomotion	START	Walk	
8	00:00:14.348	359	Himal	Locomotion	START	Jump	
9	00:00:14.660	367	Nina	Alert	STOP		
10	00:00:14.865	372	Nautilus	Locomotion	STOP	Walk	
11	00:00:14.866	372	Nautilus	Locomotion	START	Jump	
12	00:00:15.001	375	Nina	Rest	START		
13	00:00:16.467	412	Himal	Locomotion	STOP	Jump	

The following paramters are displayed:

- **Time**, the time at which the event occurred;
- **Frame index** the frame index corresponding to the event;
- **Subject**, the focal subject (if any);
- **Code**, the behavior code;
- **Type**, in case of a **state event** indicates whether the time corresponds to the start or to the stop. Empty for a **point event**;
- **Modifier**, indicates the modifier(s) that was(ere) selected (if any);
- **Comment**, is an open field where the user can add notes.

A tracking cursor (red triangle) will visualize the current event. This cursor can be positioned above the current event, see tracking cursor position option in **Preferences** window.

A double-click on a row will reposition the media player to the moment of the corresponding event. See Time offset for media reposition in **Preferences** window to customize the time offset for media repositioning.

Live observation

Events for "live observation" observation

	Time	Subject	Code	Type	Modifier	Comment
1	00:00:01.248	Nina	Alert	START		
2	00:00:15.664	Himal	Manipulate	START		vegetables
3	00:00:22.857	Himal	Manipulate	STOP		
4	00:00:24.841	Himal	Roll objects	START		
5	00:00:29.985	Himal	Roll objects	STOP		
6	00:00:38.009	Nina	Alert	STOP		
7	00:00:41.577	Sharky	Stomp	START		
8	00:00:49.105	Sharky	Stomp	STOP		

The following paramters are displayed:

- **Time**, the time at which the event occurred;
- **Subject**, the focal subject (if any);
- **Code**, the behavior code;
- **Type**, in case of a **state event** indicates whether the time corresponds to the start or to the stop. Empty for a **point event**;
- **Modifier**, indicates the modifier(s) that was(ere) selected (if any);
- **Comment**, is an open field where the user can add notes.

Observation from pictures

Events for "cametra trap" observation

	Time	Subject	Code	Type	Modifier	Comment	Image index	Image path
1	NA		Coleoptera		None|Bostrichoidea|Cerambycidae|Lagriinae|Anthaxia|None|Cetonia aurata|None|None|None		1	/data/[illegible]/08010001.JPG
2	NA		Diptera		Brachycera|None|Conopidae|Eristalinae|Chrysotoxum|None|Cheilosia impressa|None|None|None		2	/data/[illegible]/08010002.JPG
3	NA		Hymenoptera		None|Apoidea|Cephidae|Arginae|Anthidiellum|Bombus s.s.|Bombus distinguendus|None|None|None		3	/data/[illegible]/08010003.JPG
4	NA		Lepidoptera		Macroheterocera|None|Geometridae|Dismorphiinae|Albulina|None|Anarta melanopa|None|None|None		4	/data/[illegible]/08010004.JPG
5	NA		Others		flower_on		5	/data/[illegible]/08010005.JPG
6	NA		Hymenoptera		None|Cephoidea|Argidae|Cephinae|Andrena|Ashtonipsithyrus|Bombus bohemicus|None|None|None		6	/data/[illegible]/08010006.JPG

The following paramters are displayed:

- **Time**, the time at which the event occurred;
- **Subject**, the focal subject (if any);
- **Code**, the behavior code;
- **Type**, in case of a **state event** indicates whether the time corresponds to the start or to the stop. Empty for a **point event**;
- **Modifier**, indicates the modifier(s) that was(ere) selected (if any);
- **Comment**, is an open field where the user can add notes;
- **Image index**, the image index (in the directory) corresponding to the event,
- **Image path**, the path of the image corresponding to the event (can be relative or absolute).

To simplify the **events table** the relevant behaviors and subjects can be filtered see Filter events

2.6.7 Events

Recording an event

An event is a unique combination of a **time**, a **subject** and a **behavior**. If the subject is not set it will be **No focal subject**.

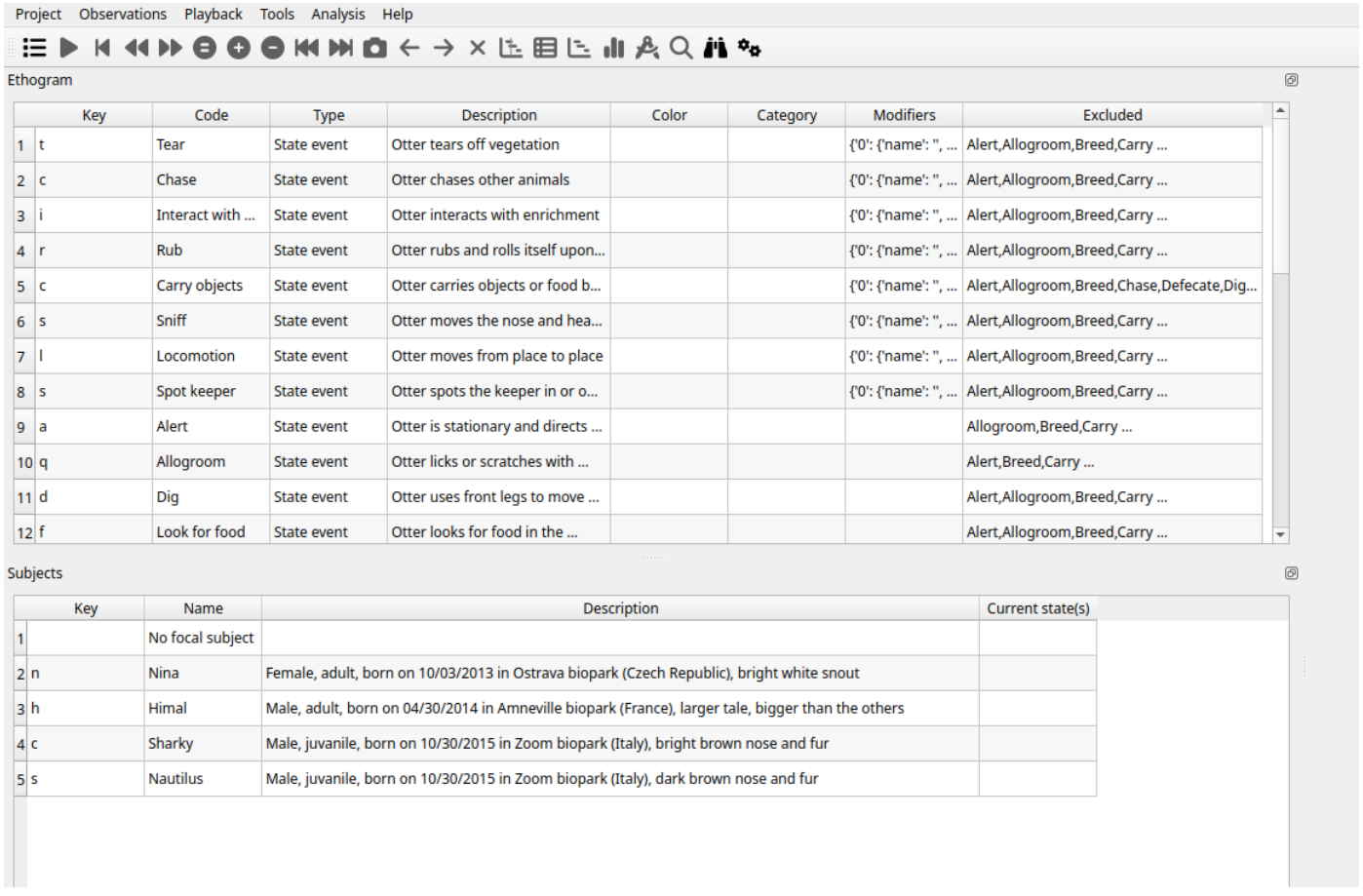
Once ready to begin your coding, you can start the media player using the **Play** button or the **Space bar**.

An **event** can be recorded by:

- pressing the predefined **key** of the keyboard corresponding to the behavior to record.
- double-clicking to the corresponding row in the **Ethogram** table.
- using the **Coding pad** (See coding pad).

The **focal subject** can be selected by:

- pressing the predefined **key** of the keyboard corresponding to the subject to select.
- double-clicking to the corresponding row in the **Subects** table.
- using the **Subject pad** (See subject pad).

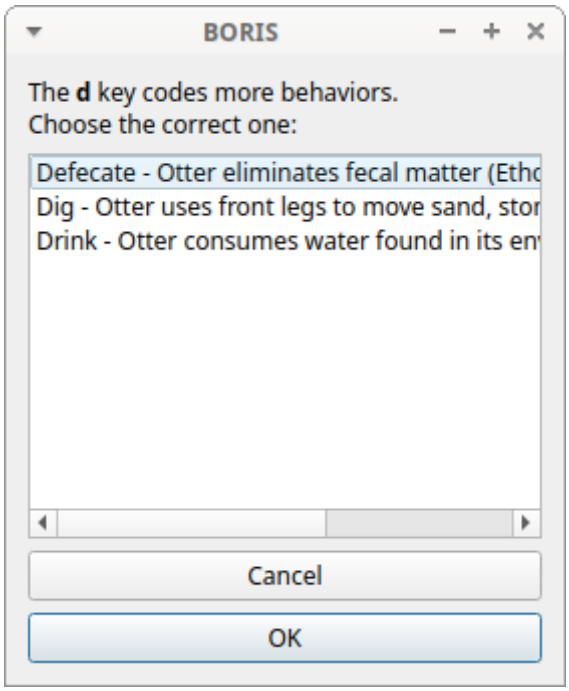


	Key	Code	Type	Description	Color	Category	Modifiers	Excluded
1	t	Tear	State event	Otter tears off vegetation			{'0': {'name': '', ...	Alert,Allogroom,Breed,Carry ...
2	c	Chase	State event	Otter chases other animals			{'0': {'name': '', ...	Alert,Allogroom,Breed,Carry ...
3	i	Interact with ...	State event	Otter interacts with enrichment			{'0': {'name': '', ...	Alert,Allogroom,Breed,Carry ...
4	r	Rub	State event	Otter rubs and rolls itself upon...			{'0': {'name': '', ...	Alert,Allogroom,Breed,Carry ...
5	c	Carry objects	State event	Otter carries objects or food b...			{'0': {'name': '', ...	Alert,Allogroom,Breed,Chase,Defecate,Dig...
6	s	Sniff	State event	Otter moves the nose and hea...			{'0': {'name': '', ...	Alert,Allogroom,Breed,Carry ...
7	l	Locomotion	State event	Otter moves from place to place			{'0': {'name': '', ...	Alert,Allogroom,Breed,Carry ...
8	s	Spot keeper	State event	Otter spots the keeper in or o...			{'0': {'name': '', ...	Alert,Allogroom,Breed,Carry ...
9	a	Alert	State event	Otter is stationary and directs ...				Allogroom,Breed,Carry ...
10	q	Allogroom	State event	Otter licks or scratches with ...				Alert,Breed,Carry ...
11	d	Dig	State event	Otter uses front legs to move ...				Alert,Allogroom,Breed,Carry ...
12	f	Look for food	State event	Otter looks for food in the ...				Alert,Allogroom,Breed,Carry ...

	Key	Name	Description	Current state(s)
1		No focal subject		
2	n	Nina	Female, adult, born on 10/03/2013 in Ostrava biopark (Czech Republic), bright white snout	
3	h	Himal	Male, adult, born on 04/30/2014 in Amneville biopark (France), larger tale, bigger than the others	
4	c	Sharky	Male, juvanile, born on 10/30/2015 in Zoom biopark (Italy), bright brown nose and fur	
5	s	Nautilus	Male, juvanile, born on 10/30/2015 in Zoom biopark (Italy), dark brown nose and fur	

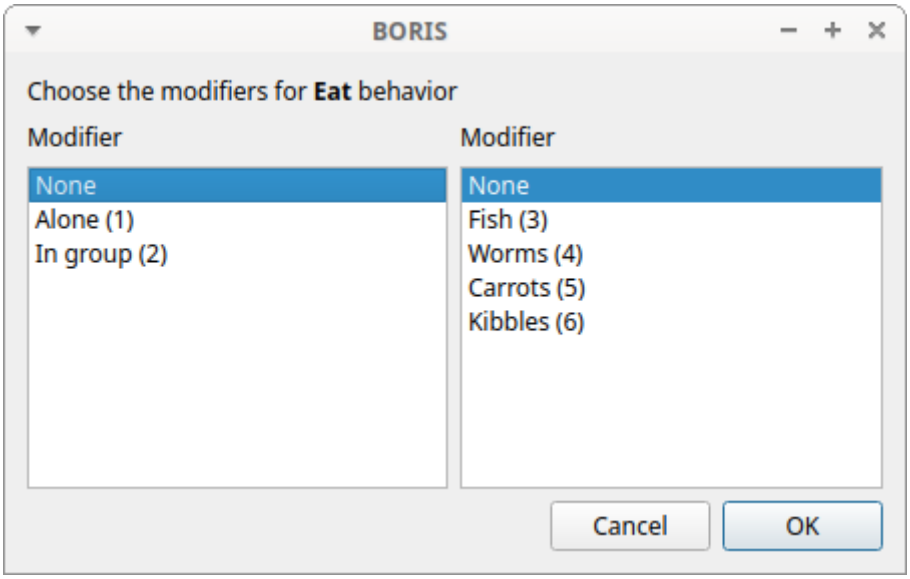
Ethogram and subjects widgets

If the pressed key defines a single event, the corresponding event will be recorded directly in the **Events** table. In the case you have specified the same key for two (or more) events (e.g. key **d** in the figure below), BORIS will prompt you for the desired behavior.



Ask for a behavior

In the case you have specified modifiers (one or more sets), BORIS will prompt you for the desired modifier(s) if any (e.g. **ball** or **opponent** in the figure below). You can select the modifiers using the mouse or the keyboard (1 , 2 , 3 , 4 , 5 or 6 key)



Ask for modifiers

If no keys are defined for the modifier selection, you can type the first character of the modifier and use the **Up arrow** and **Down arrow** keyboard keys to select the correct modifier.

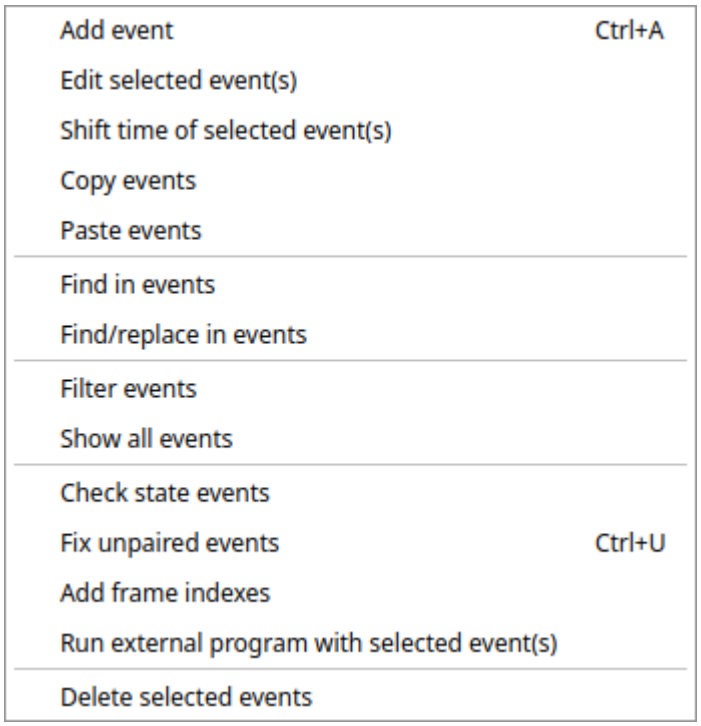
In the case your behavior type is a **Point event with coding map** or a **State event with coding map**, BORIS will show the **Coding map** window and will allow selecting the desired area(s). In case you click a part of the map in which two (or more) areas overlap, the corresponding codes will be recorded.

A recorded event can be edited (once selected) using the **Observations** > **Edit event** menu option. The resulting *Edit event parameters* allows modifying every parameter (e.g. time, subject, code, modifiers, and comment).

The **Observations** > **Add event** menu option allows adding a new event by specifying its time and the other parameters.

The Events table context menu

Some functions are available in the Events table context menu. Righ-click on the Events table and the menu will pop-up.



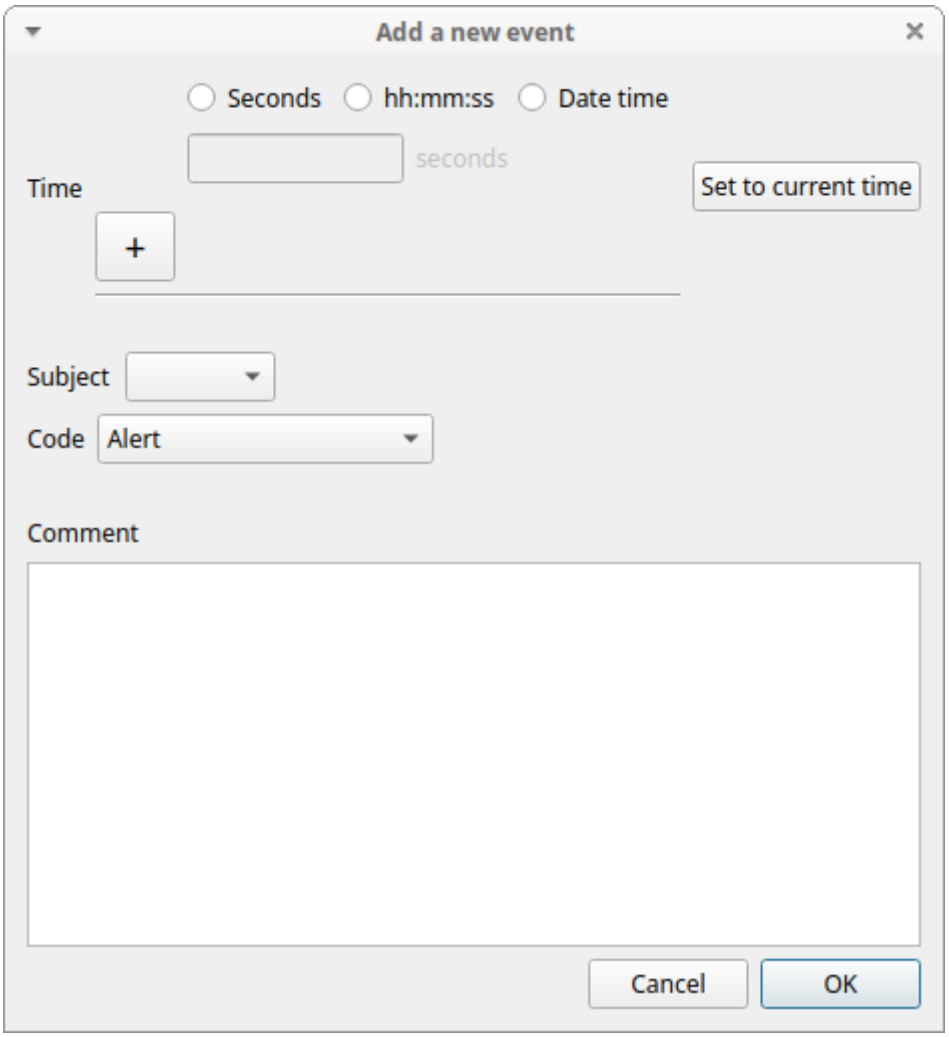
The various functions available in the menu are described below. The same functions and others are available in the **Observations** menu.

Undo an even recording

A wrong event can be removed from the events list using the **Undo** function (`^ Ctrl` + `Z`). You can go back till 25 events recorded events.

Add event

This option allows adding a new event by specifying its time and the other parameters.



Add a new event

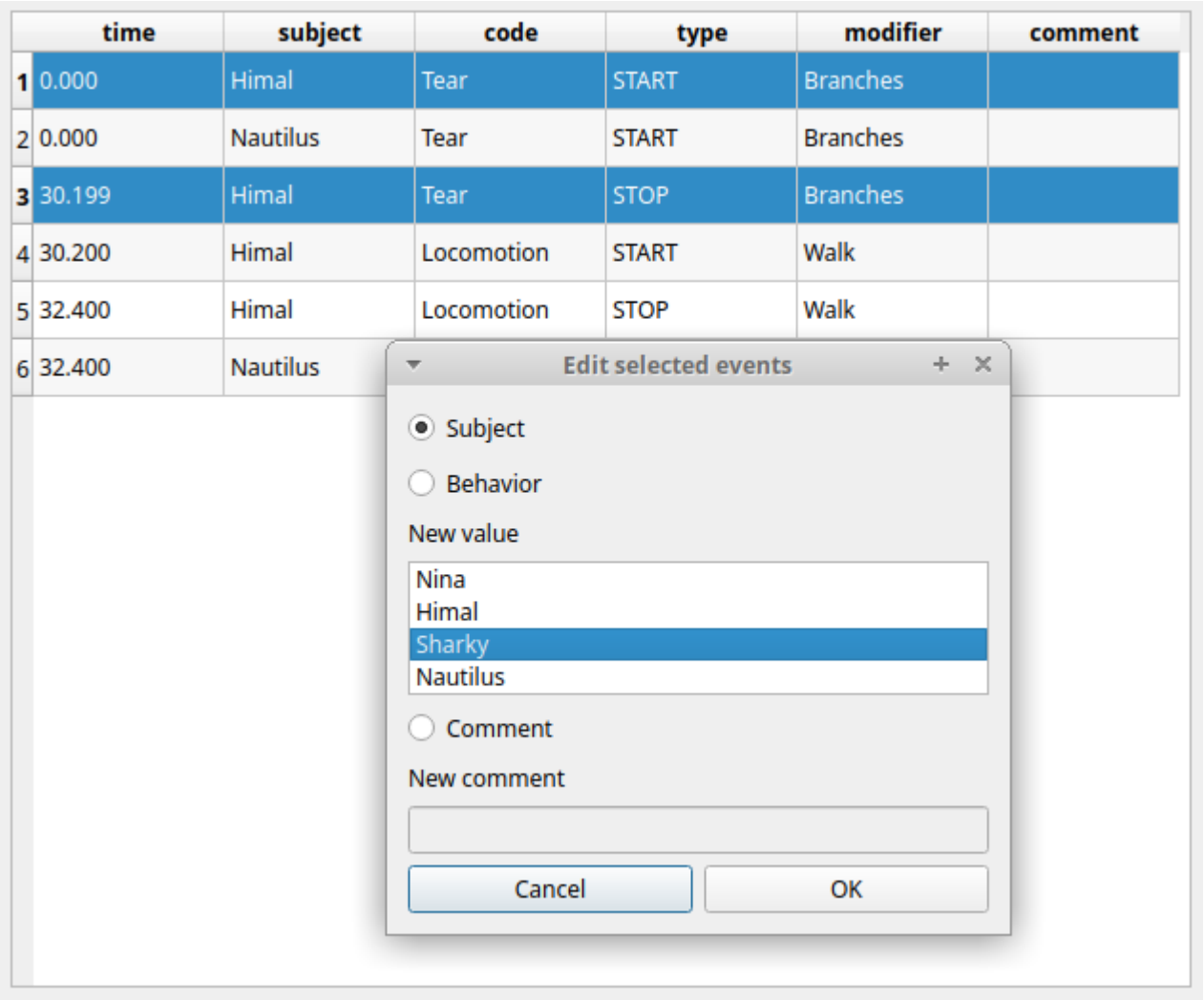
Select a time format and imput the time value.

Select the **subject** from the drop-down menu or leave empty for **No focal subject**.

Select the **behavior** from the drop-down menu.

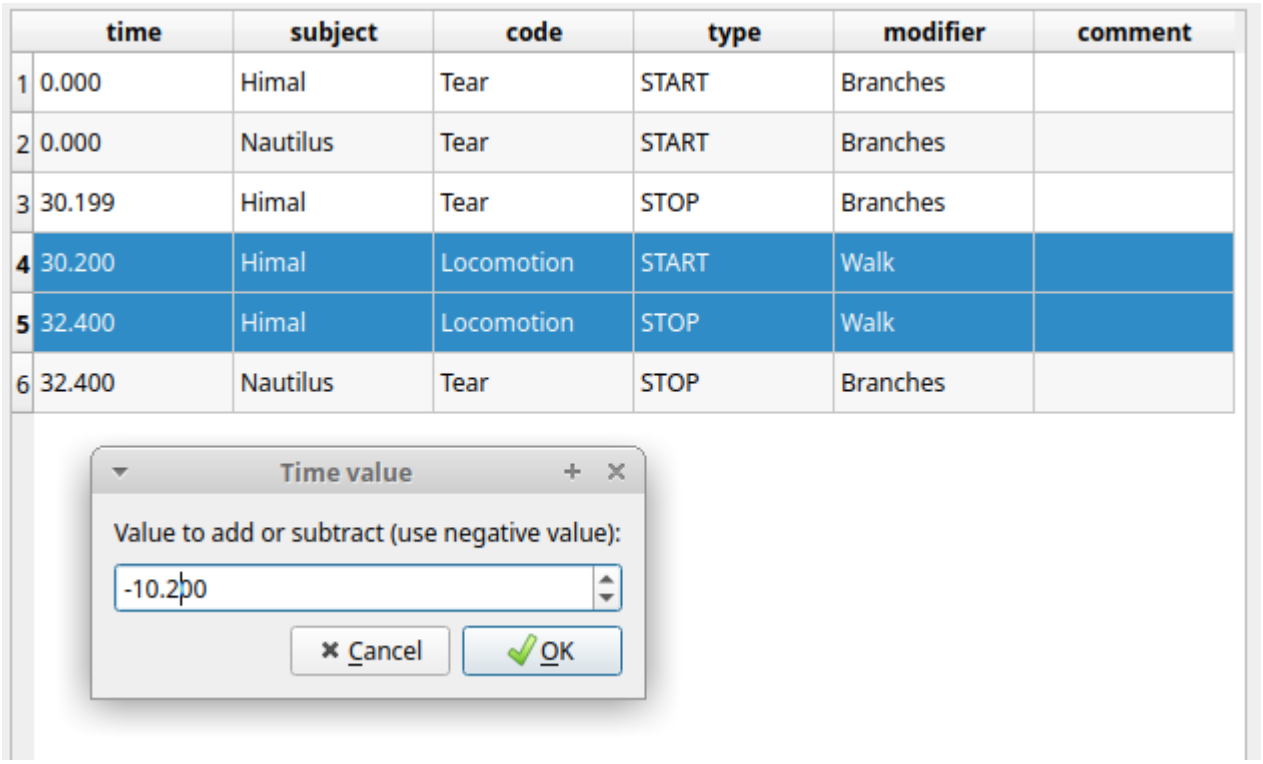
Edit selected event(s)

This option allows to edit the selected event(s). When many events are selected you have to choose the field to edit between **Subject**, **Behavior** and **Comment**. In this case the new value will apply to all selected events.



Edit time of selected event(s)

This option allows to add or subtract a time value (in seconds) to all selected events. For subtracting a value use a negative value.



	time	subject	code	type	modifier	comment
1	0.000	Himal	Tear	START	Branches	
2	0.000	Nautilus	Tear	START	Branches	
3	30.199	Himal	Tear	STOP	Branches	
4	30.200	Himal	Locomotion	START	Walk	
5	32.400	Himal	Locomotion	STOP	Walk	
6	32.400	Nautilus	Tear	STOP	Branches	

Copy events

This option allows to copy the selected events in the clipboard. The clipboard will contain the values of the selected events (except the **type** field) separated by a **<TAB>** character. The copied values are: **Time**, **Subject**, **Behavior**, **Modifier(s)**, **Frame index**

Example of clipboard content:

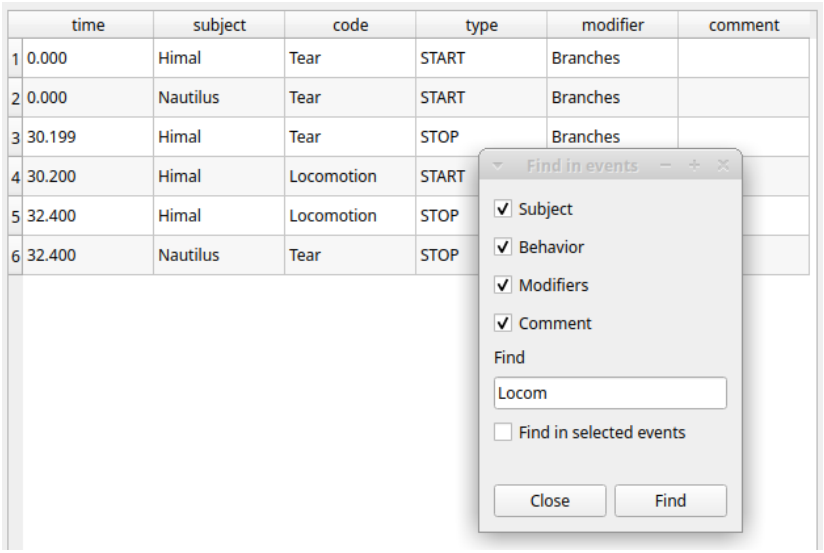
```
8.253   Himal   Locomotion  Run     207
8.329   Nautilus    Locomotion  Run     209
10.400  Sharky  Swim            260
11.778  Himal   Locomotion  Run     295
12.778  Nina    Alert           320
13.788  Nautilus    Locomotion  Run     345
13.789  Nautilus    Locomotion  Walk        345
14.348  Himal   Locomotion  Jump        359
14.660  Nina    Alert           367
14.865  Nautilus    Locomotion  Walk        372
14.865  Nautilus    Locomotion  Jump        372
15.000  Nina    Rest            375
16.466  Himal   Locomotion  Jump        412
16.467  Himal   Alert           412
23.600  Nautilus    Locomotion  Jump        590
23.600  Nautilus    Rest            590
24.228  Nautilus    Rest            606
24.407  Himal   Alert           611
24.917  Himal   Locomotion  Walk        623
39.682  Nautilus    Locomotion  Run     992
40.549  Nina    Rest            1014
42.313  Nautilus    Locomotion  Run     1058
42.314  Nautilus    Rest            1058
44.759  Himal   Locomotion  Walk        1119
44.761  Himal   Allogroom           1119
48.219  Nautilus    Rest            1206
48.363  Himal   Allogroom           1209
48.365  Himal   Locomotion  Walk        1209
49.274  Himal   Locomotion  Walk        1232
49.274  Himal   Drink           1232
50.408  Himal   Drink           1261
50.408  Himal   Swim            1261
58.851  Sharky  Swim            1472
58.950  Himal   Swim            1474
```

Paste events

This option allows to paste the clipboard content into the events table. The clipboard must respect the format described in the previous section: 5 columns separated by a **<TAB>** character.

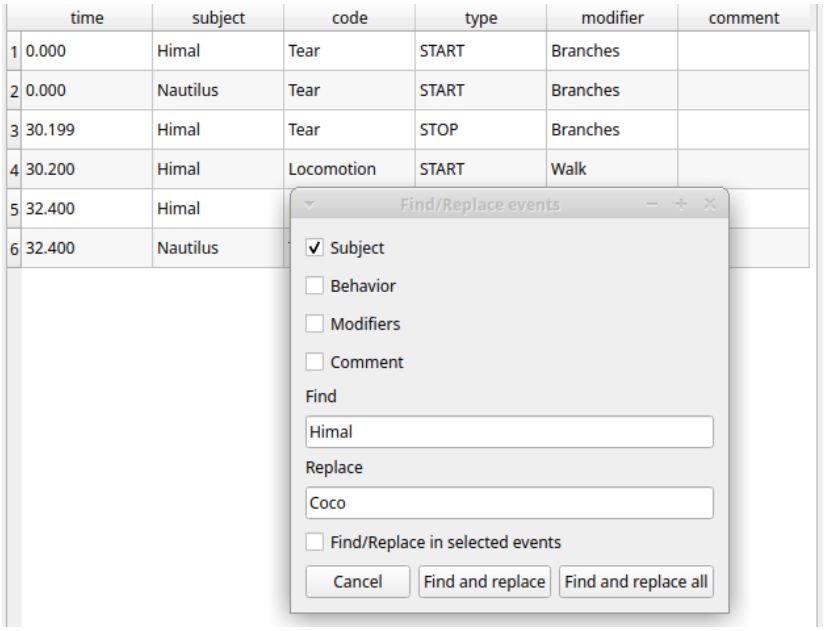
Find in events

This option allows to search for a string in the various field of events. Select the fields to be searched. The find/replace operation can be restricted to the selected events.



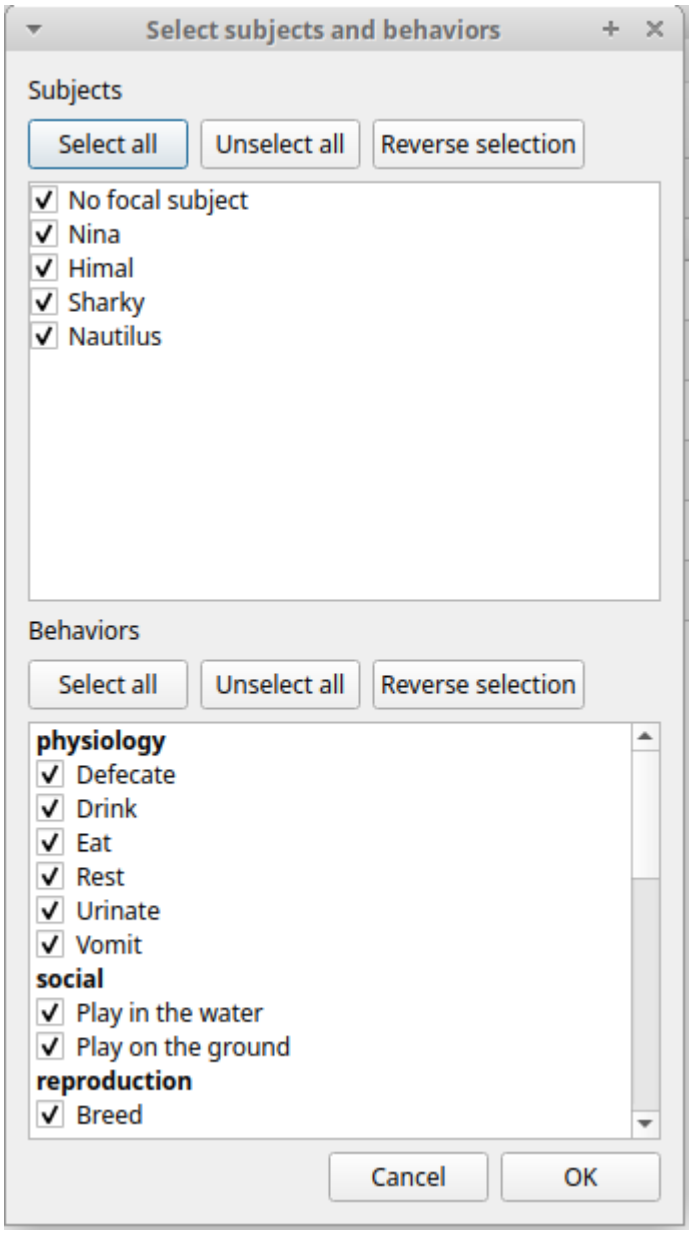
Find/Replace in events

This option allows to search for a string and replace it by a new value in the various field of events. Select the fields to be searched. The find operation can be restricted to the selected events.



Filter events

This option allows to filter the events by field value (Subject and Behavior).



Show all events

This option reverts the previous one and allows to visualize all coded events.

Check state events

This option allows to check if the **state events** are **PAIRED**, if they have a **START** and a **STOP** occurences.

Delete selected events

This option allows to delete the selected events. This operation is irreversible!

Delete all events

This option is not present in the context menu but only in the main menu (**Observations** > **Delete all events**).

This option allows to delete all then events in the current observation. This operation is irreversible!

Fix unpaired state

You can use the **Fix unpaired events** function to fix the **state events** without a STOP event.

Observations > **Fix unpaired events** (keyboard shortcut: `^ Ctrl` + `U`)

The program will ask for a time at which insert the STOP events for all unpaired **state events**

This function can be run on a set of selected observations (when no observation is open). In this case the STOP events will be inserted at the end of observation.

Add frame indexes

This function can be used for the observations from a video. The frame index corresponding to the coded events will be added in the events table.

Run external program with selected events

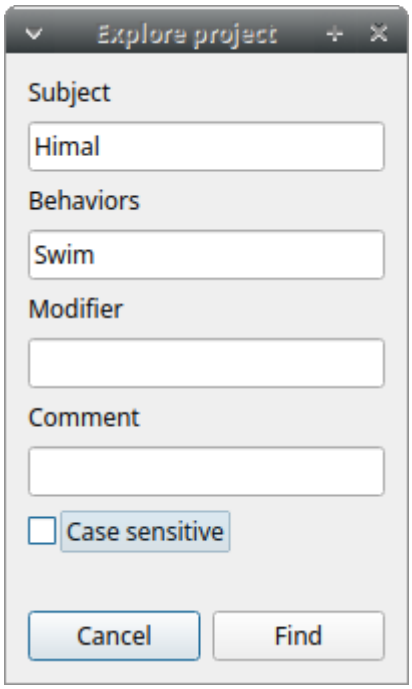
This function is not yet implemented.

Explore project

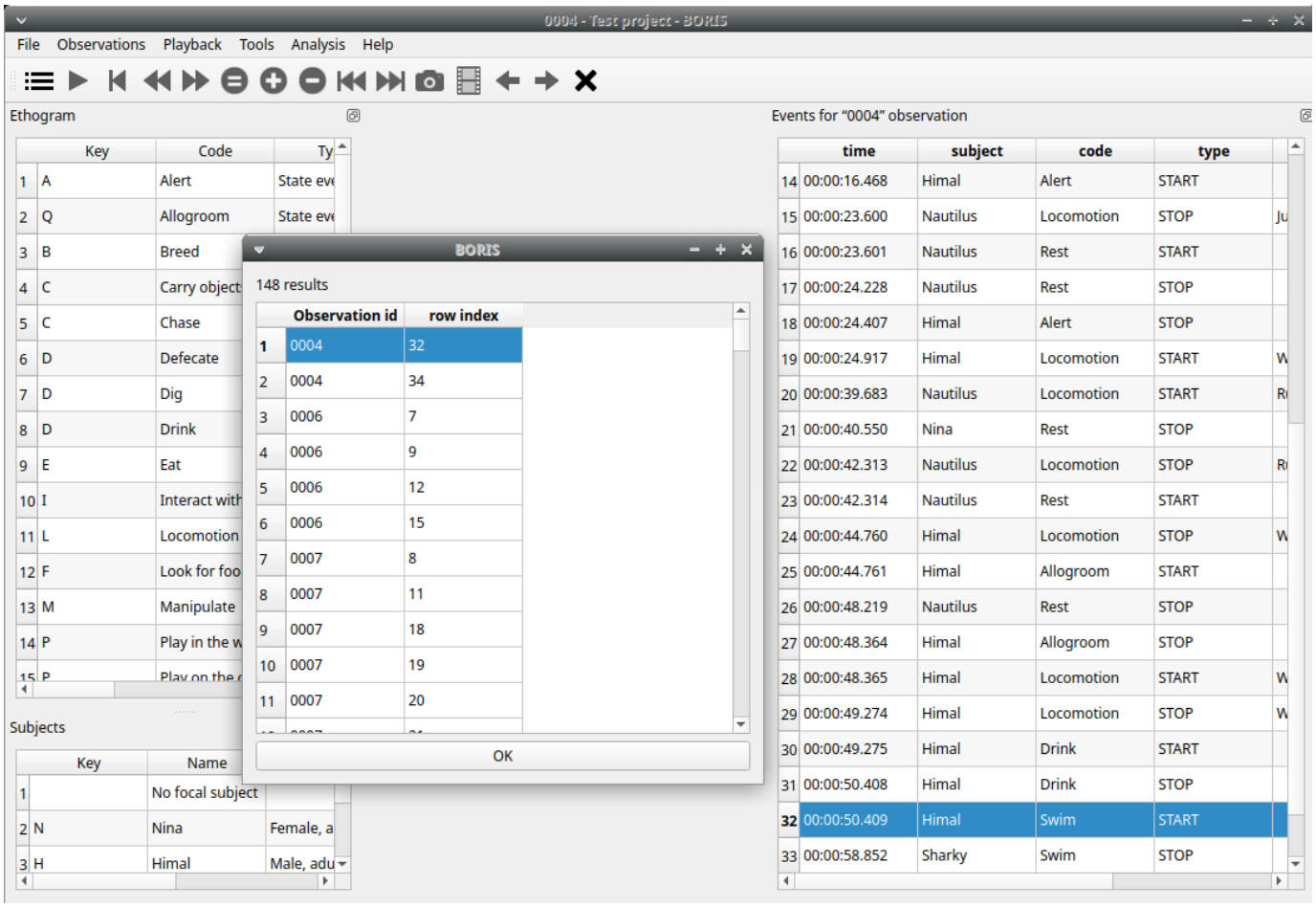
You can search information in various fields in all observations in the current project (Observations > Explore project).

The searchable fields are: **subject**, **behavior**, **modifier** and **comment**.

If more than one field is searched a logic **AND** will apply.



The events that were found are listed in a table. By double-clicking on the row the corresponding observation will be opened and the visualization will be scrolled to the row corresponding to the event.



Frame-by-frame mode

You can switch between the media player and the frame-by-frame mode using the **arrow buttons** in the toolbar:

In frame-by-frame mode the video will stop playing and the user will visualize the video frame by frame.

> **Note**
>
> Some video files should be re-encoded to be used in frame-by-frame mode. Otherwise the extracted frames are not reliable or it will not be possible to move backward.

You can move between frames by using the arrow keys in the toolbar (on the right) or by using keyboard special keys:

`← Left` Go to the **previous frame**

`→ Right` Go to the **next frame**

`⇞ Page Up` Switch to the **next media**

`⇞ Page Up` Switch to the **previous media**

`↑ Up` **Jump forward** in the current media

`↓ Down` **Jump backward** in the current media

If you have a numeric keypad you can use the following keys in alternative:

- The key **/** will allow you to view the previous frame
- The key ***** will allow you to view the next frame

To return in the media player mode press the **Play**  button in the toolbar.

2.7 Export events

The coded events can be exported in various formats.

2.7.1 Export events in tabular format

Observations > **Export events** > **Tabular events**

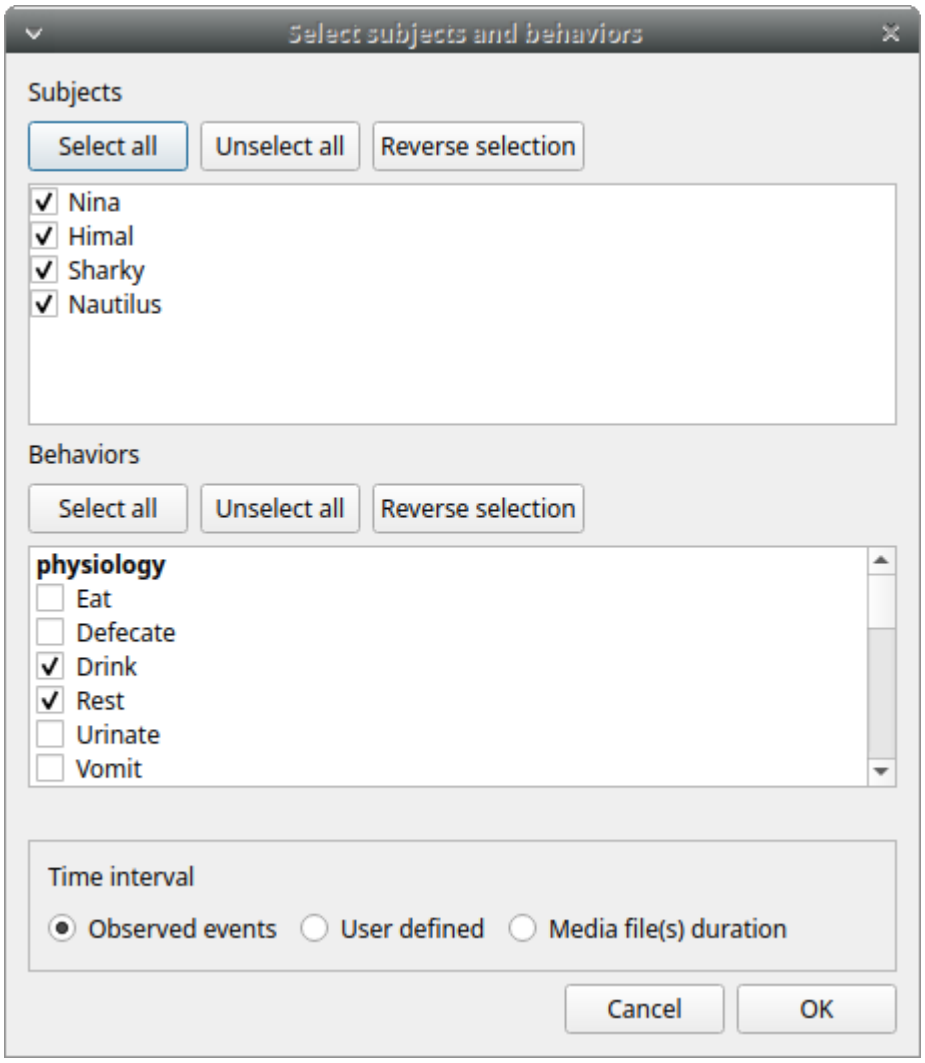
This function exports the events of selected observations in one or many files. Various formats are available:

- Plain text in tabular format
  - **Tab Separated Values** (TSV)
  - **Comma Separated Values** (CSV)
  - **Hyper Text Markup language** (HTML)
- Spreadsheet files
  - **OpenDocument** (ODS)
  - **Microsoft Excel** (XLSX, XLS)
- **Pandas dataframe** (to be loaded in Python with the pickle module)
- **R dataframe** (to be loaded in R with readRDS function)

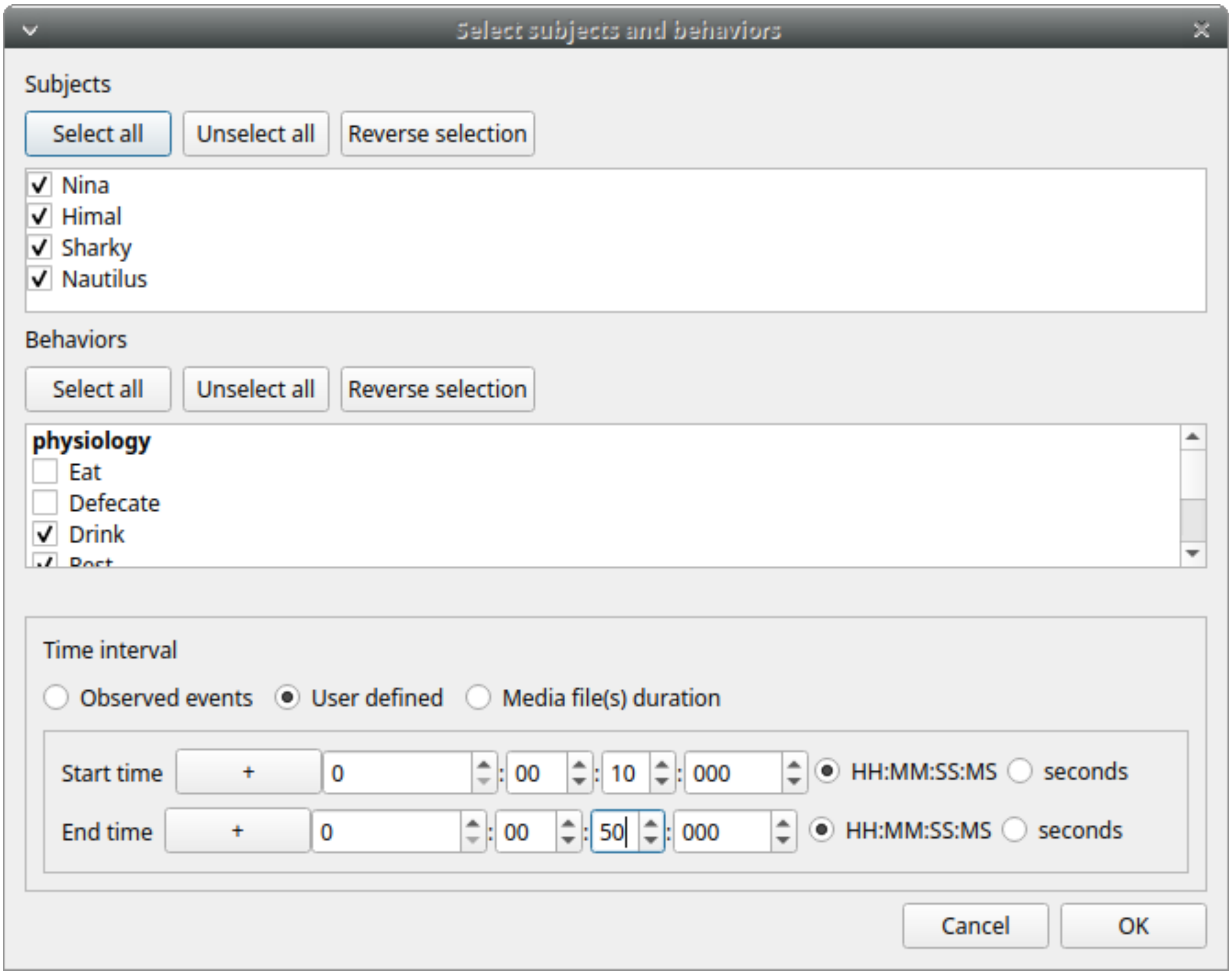
If many observations are selected BORIS will ask for a directory to save the various files. For the spreadsheet format (XLSX and ODS) the events can be exported on various worksheet in a single workbook. All these formats are suitable for further analysis.

Select the subjects, the behaviors and the time interval.

Set the time interval to the **Observed events**



Select a **User defined** time interval.



Example of output of tabular events

Observation id	Observation date	Description	Observation duration	Observation type	Source	Media duration (s)	FPS	Location	Weather	Temperature	Visitors	Subject	Behavior	Behavioral category	Modifier #1	Behavior type	Time	Media file name	Image index	Image file path	Comment
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Himal	Tear		Branches	START	0.000	20160517162540.m2ts	NA	NA	
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Sharky	Tear		Branches	START	0.000	20160517162540.m2ts	NA	NA	
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Nautilus	Tear		Branches	START	1.359	20160517162540.m2ts	NA	NA	
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Nautilus	Tear		Branches	STOP	25.776	20160517162540.m2ts	NA	NA	
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Nautilus	Carry objects		Branches	START	25.777	20160517162540.m2ts	NA	NA	
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Nautilus	Carry objects		Branches	STOP	27.732	20160517162540.m2ts	NA	NA	
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Sharky	Tear		Branches	STOP	30.688	20160517162540.m2ts	NA	NA	
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Sharky	Locomotion		Walk	START	30.689	20160517162540.m2ts	NA	NA	
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Sharky	Locomotion		Walk	STOP	31.819	20160517162540.m2ts	NA	NA	
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Himal	Tear		Branches	STOP	33.898	20160517162540.m2ts	NA	NA	
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Himal	Locomotion			START	33.899	20160517162540.m2ts	NA	NA	
0002	2016-05-17 00:00:24	Vegetation	34.47	Media file(s)	player #1:20160517162540.m2ts	36.000	25.000	4° 55′ 59″ N – 7° 25′ 18″	sun	24	1046	Himal	Locomotion			STOP	34.470	20160517162540.m2ts	NA	NA	

2.7.2 Export aggregated events

Observations > **Export events Aggregated events**

This function will export the events corresponding to the selected subjects and the selected behaviors of the selected observations.

Various formats are available:

- Plain text in tabular format
  - **Tab Separated Values** (TSV)
  - **Comma Separated Values** (CSV)
  - **Hyper Text Markup language** (HTML)
- Spreadsheet files
  - **OpenDocument** (ODS)
  - **Microsoft Excel** (XLSX, XLS)
- **SQL format** for populating a SQL database
- **SDIS format** for analysis with the GSEQ program available at http://www2.gsu.edu/~psyrab/gseq
- **Pandas dataframe** (to be loaded in Python with the pickle module)
- **R dataframe** (to be loaded in R with readRDS function)

If two or more observations are selected you can choose to group all results in one file. If you do not want to group results BORIS will ask for a directory to save the various files (the observation id will be used as file name).

The **State events** are paired and in this case the event duration is available.

An arbitrary time interval can be selected (check the **Limit to time interval** option). In this case the ongoing events will be started at start time and stopped at end time in the export file.

The following fields are available in the output:

- Observation id
- Observation date and time
- Observation description
- Observation type (Media file / Live / Pictures)
- Source (for media file and pictures)
- Total duration (in seconds, the duration of observation in base of the selected time interval)
- Media duration(s) (in seconds, for media file observation)
- FPS (frame/s, for video file, number of images per second)
- Independent variables (one column by variable9)
- Subject name
- Observation duration by subject
- Behavior
- Behavioral category (if any)
- Modifier(s) of behavior (one column by madifier)
- Behavior type (STATE / POINT)
- Start (seconds)
- Stop (seconds)
- Duration (seconds, duration of the event for STATE events)
- Media file name (for media file observation, media in which the event occurs)
- Image index start (for observations from pictures, index of the image where the event starts)
- Image index stop (for observations from pictures, index of the image where the event stops)
- Image file path start (for observations from pictures, path of the image where the event stops)
- Image file path stop (for observations from pictures, path of the image where the event stops)
- Comment start
- Comment stop

Example of table export of aggregated events (TSV, CSV, XLSX, ODS, HTML)

Observation id	Observation date	Description	Observation type	Source	Total duration	Media duration (s)	FPS (frame/s)	Location	Weather	Temperature	Visitors
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0004	2016-05-17 00:00:59	Central trunks	Media file	player #1:20160517162952.m2ts	50.697	59.040	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
0005	2016-05-17 00:00:49	In the pool	Media file	player #1:20160517163131.m2ts	40.770	40.800	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
0005	2016-05-17 00:00:49	In the pool	Media file	player #1:20160517163131.m2ts	40.770	40.800	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
0006	2016-05-17 00:00:42	In the pool	Media file	player #1:20160517163231.m2ts	61.830	61.920	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
0006	2016-05-17 00:00:42	In the pool	Media file	player #1:20160517163231.m2ts	61.830	61.920	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
0006	2016-05-17 00:00:42	In the pool	Media file	player #1:20160517163231.m2ts	61.830	61.920	25.000	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046

Subject	Observation duration by subject by observation	Behavior	Behavioral category	Modifier #1	Behavior type	Start (s)	Stop (s)	Duration (s)	Media file name	Image index start	Image index stop	Image file path start	Image file path stop	Comment start	Comment stop
Himal	47.613	Locomotion	Not defined	Run	STATE	8.253	11.779	3.526	20160517162952.m2ts	NA	NA				
Nautilus	24.431	Locomotion	Not defined	Run	STATE	8.329	13.789	5.460	20160517162952.m2ts	NA	NA				
Sharky	48.452	Swim	Not defined		STATE	10.400	58.852	48.452	20160517162952.m2ts	NA	NA				
Nina	27.431	Alert	Not defined		STATE	12.778	14.660	1.882	20160517162952.m2ts	NA	NA				
Nautilus	24.431	Locomotion	Not defined	Walk	STATE	13.790	14.865	1.075	20160517162952.m2ts	NA	NA				
Himal	47.613	Locomotion	Not defined	Jump	STATE	14.348	16.467	2.119	20160517162952.m2ts	NA	NA				
Nautilus	24.431	Locomotion	Not defined	Jump	STATE	14.866	23.600	8.734	20160517162952.m2ts	NA	NA				
Nina	27.431	Rest	physiology		STATE	15.001	40.550	25.549	20160517162952.m2ts	NA	NA				
Himal	47.613	Alert	Not defined		STATE	16.468	24.407	7.939	20160517162952.m2ts	NA	NA				
Nautilus	24.431	Rest	physiology		STATE	23.601	24.228	0.627	20160517162952.m2ts	NA	NA				
Himal	47.613	Locomotion	Not defined	Walk	STATE	24.917	44.760	19.843	20160517162952.m2ts	NA	NA				
Nautilus	24.431	Locomotion	Not defined	Run	STATE	39.683	42.313	2.630	20160517162952.m2ts	NA	NA				
Nautilus	24.431	Rest	physiology		STATE	42.314	48.219	5.905	20160517162952.m2ts	NA	NA				
Himal	47.613	Allogroom	Not defined		STATE	44.761	48.364	3.603	20160517162952.m2ts	NA	NA				
Himal	47.613	Locomotion	Not defined	Walk	STATE	48.365	49.274	0.909	20160517162952.m2ts	NA	NA				
Himal	47.613	Drink	physiology		STATE	49.275	50.408	1.133	20160517162952.m2ts	NA	NA				
Himal	47.613	Swim	Not defined		STATE	50.409	58.950	8.541	20160517162952.m2ts	NA	NA				
Sharky	40.77		Manipulate	Not defined	STATE	0.000	40.770	40.770	20160517163131.m2ts	NA	NA				
Nautilus	40.77		Manipulate	Not defined	STATE	0.000	40.770	40.770	20160517163131.m2ts	NA	NA				
Himal	44.368	Manipulate	Not defined		STATE	0.000	40.760	40.760	20160517163231.m2ts	NA	NA				
Nautilus	45.037	Manipulate	Not defined		STATE	0.000	17.448	17.448	20160517163231.m2ts	NA	NA				
Sharky	46.537	Manipulate	Not defined		STATE	1.824	48.361	46.537	20160517163231.m2ts	NA	NA				

Example of SQL export of aggregated events:

```
CREATE TABLE aggregated_events (id INTEGER PRIMARY KEY ASC, observation TEXT, subject TEXT, behavior TEXT, type TEXT, modifiers TEXT, start FLOAT, stop FLOAT,
comment TEXT, comment_stop TEXT, image_index_start INTEGER,image_index_stop INTEGER,image_path_start TEXT,image_path_stop TEXT);
INSERT INTO "aggregated_events" VALUES(1,'0001_a','Himal','Tear','STATE','Branches',0.0,30.199,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(2,'0001_a','Himal','Locomotion','STATE','Walk',30.2,32.4,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(3,'0001_a','Nautilus','Tear','STATE','Branches',0.0,32.4,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(4,'0001_b','Himal','Tear','STATE','Branches',0.0,30.199,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(5,'0001_b','Himal','Locomotion','STATE','Walk',30.2,32.4,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(6,'0001_b','Nautilus','Tear','STATE','Branches',0.0,31.4,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(7,'0002','Himal','Tear','STATE','Branches',0.0,33.898,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(8,'0002','Himal','Locomotion','STATE','',33.899,34.47,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(9,'0002','Sharky','Tear','STATE','Branches',0.0,30.688,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(10,'0002','Sharky','Locomotion','STATE','Walk',30.689,31.819,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(11,'0002','Nautilus','Tear','STATE','Branches',1.359,25.776,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(12,'0002','Nautilus','Carry objects','STATE','Branches',25.777,27.732,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(13,'0003','Nina','Locomotion','STATE','Walk',21.626,22.5,'','','NA','NA',NULL,NULL);
INSERT INTO "aggregated_events" VALUES(14,'0003','Nina','Manipulate','STATE','',0.0,21.625,'','','NA','NA',NULL,NULL);
```

2.7.3 Export events as behavioral sequences

Observations > **Export events as behavioral sequences**

Behavioral strings can be used with the **Behatrix** program: Behatrix

Example:

```
# observation id: demo#1
# observation description:
# Media file name: video1.mp4, video2.mp4


Subject #1:
eat|jump|eat|jump

Subject #2:
eat|rest|jump|eat|jump
```

2.7.4 Export events as Praat TextGrid

Observations > **Export events as Praat TextGrid**

Example:

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 4.3
xmax = 113.988
tiers? <exists>
size = 2
item []:
    item [1]:
        class = "IntervalTier"
        name = "Subject #1"
        xmin = 4.3
        xmax = 10.0
        intervals: size = 1
        intervals [1]:
            xmin = 4.3
            xmax = 10.0
            text = "eat"
    item [2]:
        class = "IntervalTier"
        name = "Subject #2"
        xmin = 26.6
        xmax = 113.988
```

```
        intervals: size = 1
        intervals [1]:
            xmin = 26.6
            xmax = 113.988
            text = "eat"
```

2.7.5 Export events for analysis with JWatcher

Observations > **Export events for analysis with JWatcher**

JWatcher is a powerful tool for the quantitative analysis of behavior.

The events coded with BORIS can be exported to be analyzed with JWatcher.

Click **Observations** > **Export events** > **for analysis with JWatcher** to export the coded events.

BORIS will ask for selecting a directory. After this, for each combination of selected observation and selected subject the following files will be created:

- the Focal Data File (.dat)
- the Focal Analysis Master File (.faf)
- the Focal Master File (.fmf)

These files can be used to analyze your observations with JWatcher.

2.7.6 Export events as Behaviors Binary Table

Observations > **Export events as Behaviors Binary Table**

A time interval will be asked to the user (in seconds). The observation will be checked every n seconds and the presence (1, absence: 0) of the selected behaviors will be exported in a table for each selected subjects.

Example for a time interval of 1 second:

time	Alert	Drink	Locomotion	Swim
0.0	0	1	0	0
1.0	0	1	0	0
2.0	0	1	0	0
3.0	0	1	0	0
4.0	0	1	0	0
5.0	0	1	0	0
6.0	0	1	0	0
7.0	0	1	0	0
8.0	0	1	0	0
9.0	1	0	0	0
10.0	1	0	0	0
11.0	0	0	1	0
12.0	1	0	0	0
13.0	1	0	0	0
14.0	1	0	0	0
15.0	1	0	0	0
16.0	1	0	0	0
17.0	1	0	0	0
18.0	0	0	1	0
19.0	0	0	1	0
20.0	0	0	1	0
21.0	0	0	1	0
22.0	1	0	0	0
23.0	0	0	0	0
24.0	0	0	0	0
25.0	0	0	0	0
26.0	0	0	0	0
27.0	0	0	0	0
28.0	0	0	0	0
29.0	0	0	0	0
30.0	0	0	0	0
31.0	0	0	0	0
32.0	0	0	0	0
33.0	0	0	0	1
34.0	0	0	0	1
35.0	0	0	0	1
36.0	0	0	0	1
37.0	0	0	0	1
38.0	0	0	0	1
39.0	0	0	0	1
40.0	0	0	0	1
41.0	0	0	0	1
42.0	0	0	0	1

```
43.0    0         0        0           1
44.0    0         0        0           1
45.0    0         0        0           1
46.0    0         0        0           1
47.0    0         0        0           1
48.0    0         0        0           1
49.0    0         0        0           1
```

2.7.7 Extract clips from media files corresponding to coded events

Sequences of media file corresponding to coded events can be extracted from media files:

1. Click on **Observations** > **Extract clips from media files** option.
2. Choose the observation(s).
3. Select the subjects and events to be extracted. Modifiers con be included
4. Select the time interval around the events (in seconds, the default value is 0).
5. Select the tracks to include in clips:
    - Video and audio
    - Only video
    - Only audio
6. Select a destination directory that will contain the extracted clips.

The time offset will be substracted from the starting time of event and added to the stopping time. All the extracted clips will be saved in the selected directory followind the file name format:

{observation id}_{player}_{subject}_{behavior}_{start time}-{stop time}

If modifiers are included:

{observation id}_{player}_{subject}_{behavior}_{start time}-{stop time}_{modifiers concatenated with +}

2.7.8 Extract frames corresponding to coded events

The frames corresponding to coded events can be extracted and saved as images.

1. Click on **Observations** > **Extract frames from media files** option.
2. Choose the observation(s).
3. Select the subjects and events to be extracted. Modifiers con be included
4. Select the time interval around the events (in seconds, the default value is 0).
5. Select the image format:
    - JPG - small size / low quality
    - PNG - big size / high quality
6. Select a destination directory that will contain the extracted frames.

2.7.9 Export transitions matrix

3 transitions matrix outputs are available: The matrix of frequencies of transitions, the matrix of frequencies of transition after each behavior and the matrix of number of transitions.

Matrix of frequencies of transitions

This matrix contains the frequencies of total transitions. The sum of all frequencies must be 1.

Example of frequencies of transitions matrix:

```
eat   sleep     walk
eat         0.0   0.286    0.143
sleep     0.143     0.0    0.143
walk      0.286     0.0      0.0
```

In this matrix you can see that the **eat** behavior precedes the **sleep** behavior with a frequency of **0.286** of the total number of transitions.

Matrix of frequencies of transitions after behavior

This matrix contains the frequencies of transitions after each behavior. The sum of each row must be 1.

Example:

```
eat    sleep     walk
eat      0.0    0.667    0.333
sleep    0.5      0.0      0.5
walk     1.0      0.0      0.0
```

In this example you can see that **sleep** follows **eat** with a frequency of **0.667** and **walk** follows with a frequency of **0.333**.

Matrix of number of transitions

This matrix contains the number of transitions after each behavior.

Example:

```
eat   sleep   walk
eat       0       2      1
sleep     1       0      1
walk      2       0      0
```

2.8 Playback menu

2.8.1 Jump

Jump forward

Allow to jump forward in the current media file. See **File** > **Preferences** for setting the jump value.

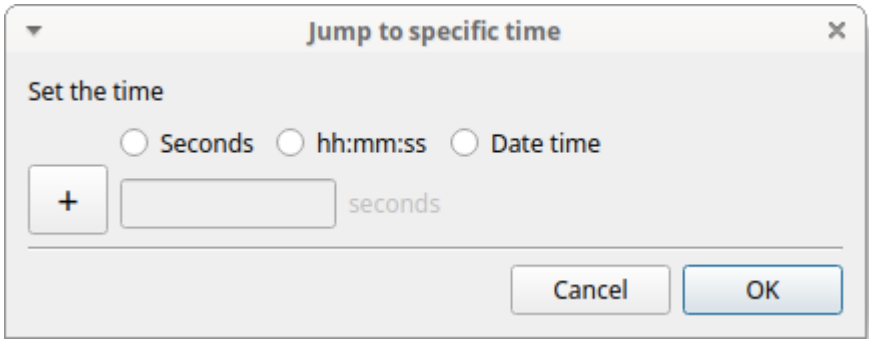
Jump backward

Allow to jump backward in the current media file. See **File** > **Preferences** for setting the jump value.

Jump to specific time

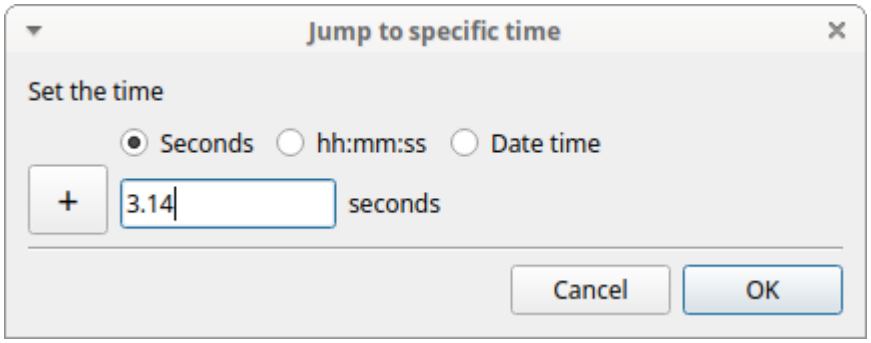
Allow to go to a specific time in the current media file.

The time selection widget will pop-up:

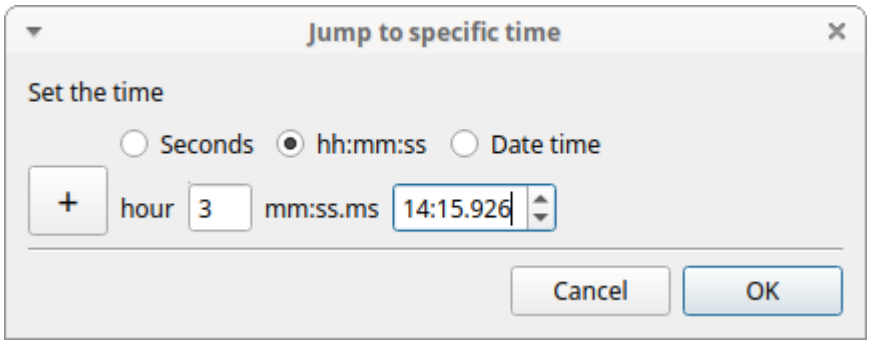


3 formats are available to select the time:

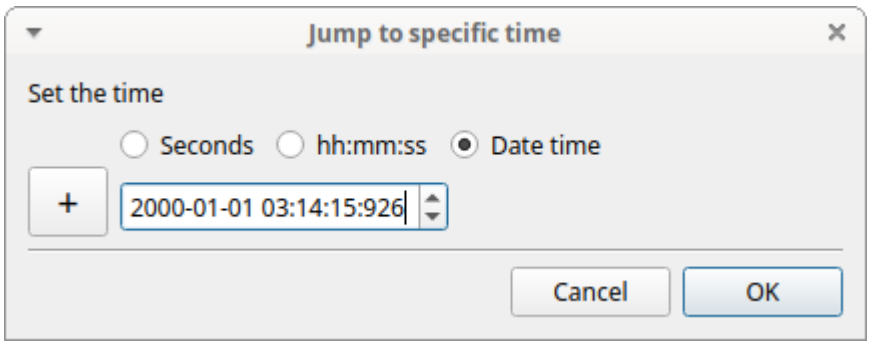
- Decimal seconds:



- HH:MM:SS:ZZZ format (ZZZ indicates the milliseconds):



• A date-time format (YYYY-MM-DD hh:mm:ss.zzz):



2.8.2 Zoom level

Click the media player you want to set the zoom level.

Using the keyboard

Zoom in `^ Ctrl` + `+` or `^ Ctrl` + Mouse wheel up

Zoom out `^ Ctrl` + `-` or `^ Ctrl` + Mouse wheel down

Reset zoom level `^ Ctrl` + `0` or by clicking the mouse right button on the video.

Using the mouse

Zoom in Double click on left mouse button

Zoom out Double click on right mouse button

2.8.3 Pan video

Click the media player you want to pan.

Using the keyboard

Pan Left `^ Ctrl` + `← Left`

Pan Right `^ Ctrl` + `→ Right`

Pan Down `^ Ctrl` + `↑ Up`

Pan Up `^ Ctrl` + `↓ Down`

Using the mouse

Pan Up: **Mouse Wheel up** (the video moves down)

Pan Down: **Mouse Wheel down** (the video moves up)

Pan Left: ⇧ Shift + **Mouse Wheel Up** (the video moves to the right)

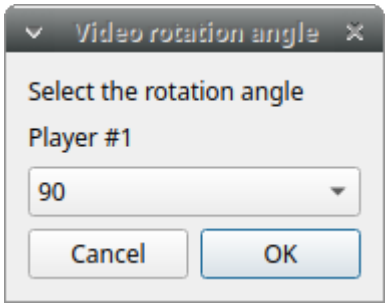
Pan Right: ⇧ Shift + **Mouse Wheel Down** (the video moves to the left)

Reset Pan and zoom: ⇧ Shift + **Left mouse button**

The zoom level can also be set using the menu **Playback** > **Zoom level**

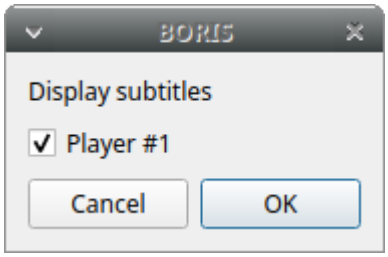
2.8.4 Rotate video

Select the video rotation angle for each player using the menu **Playback** > **Rotate video**. The available rotation angles are: 0, 90, 180 and 270.



2.8.5 Display subtitles

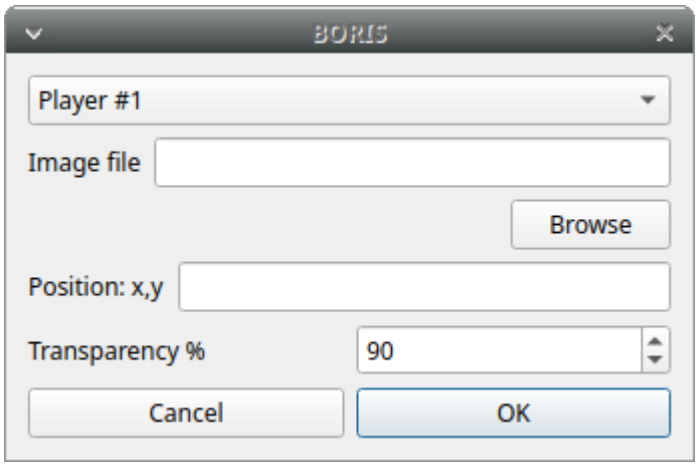
Select to display or hide the subtitles using the menu **Playback** > **Display subtitles**. The subtitles file must have exactly the same name of the video file except for the extension and be placed in the same directory.



2.8.6 Image overlay on video

Select an image overlay to be displayed on the video **Playback** > **Image overlay on video** > **Add**. If the selected image does not have a transparent background the transparency can be set from 0 (full transparency) to 255 (no transparency).

The image must be in PNG format, if the image is smaller than the video resolution the image position can be set from the top-left corner (x: horizontally, y: vertically).

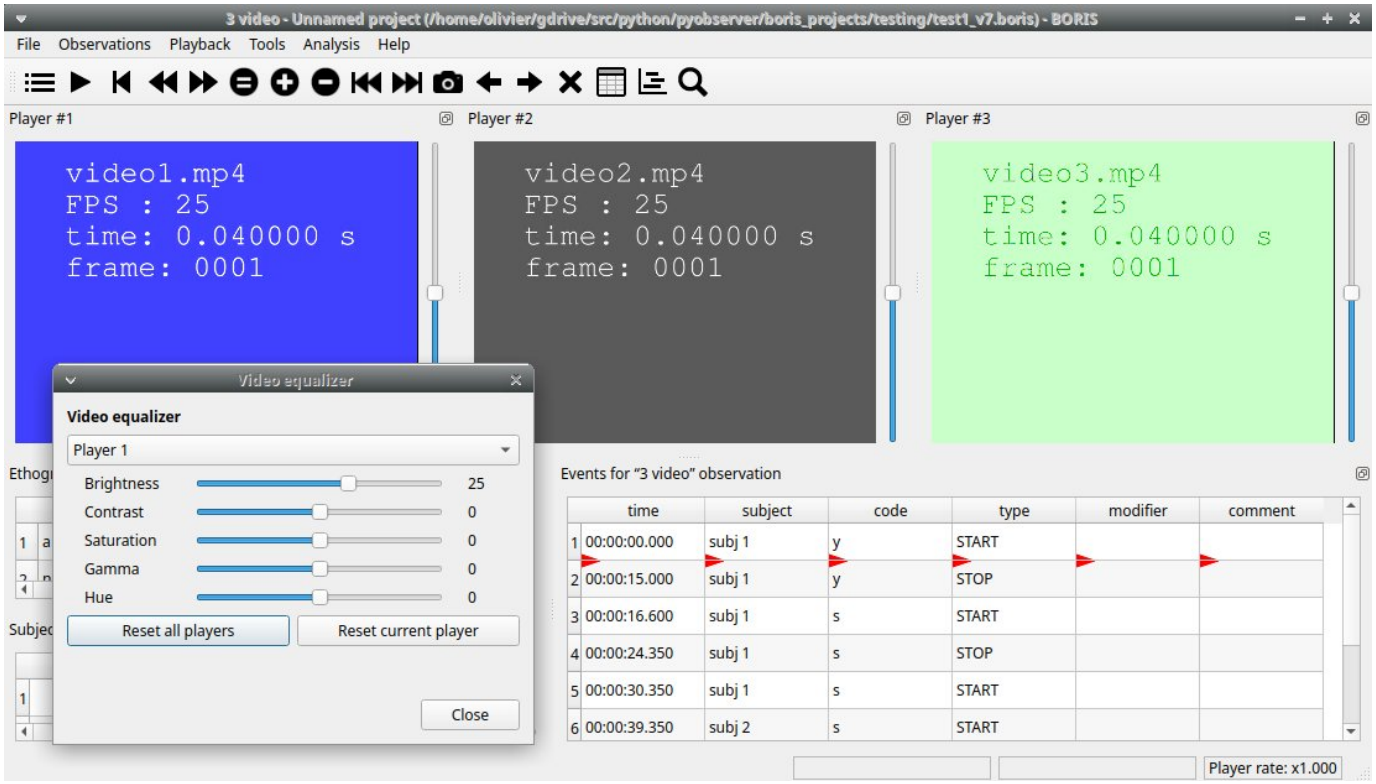


Select **> Playback > Image overlay on video > Remove** to remove the image overlay.

2.8.7 Video equalizer

Playback > **Video equalizer**

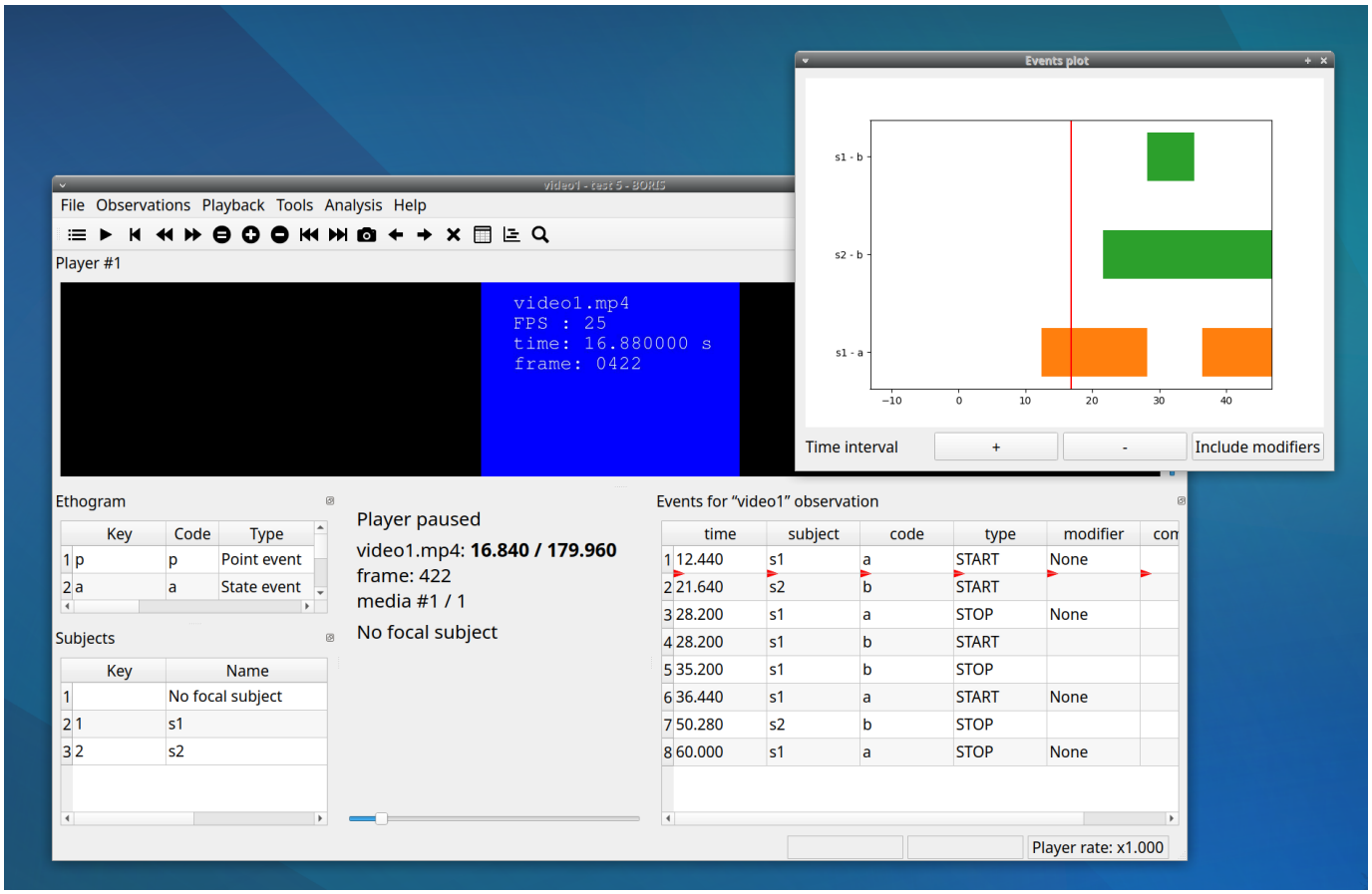
Using this function the **brightness**, the **contrast**, the **saturation**, the **gamma** and the **Hue** can be set for each player.



2.9 Tools

2.9.1 Plot events in real-time

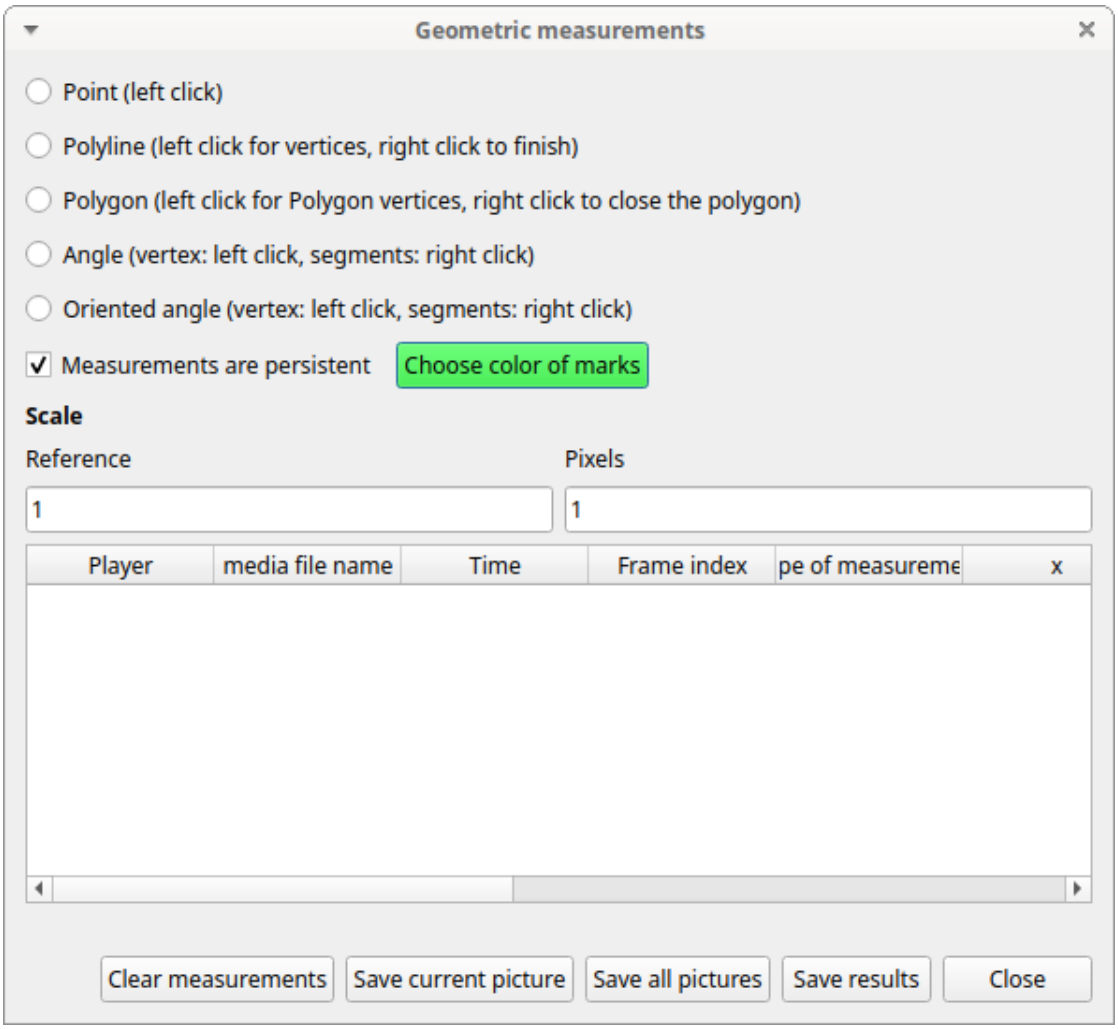
This function can be activated with **Tools** > **Plot event in real time**.



2.9.2 Geometric measurements

Some geometric measurements can be done: **distances**, **areas** and **angles** can be measured and **point positions** recorded.

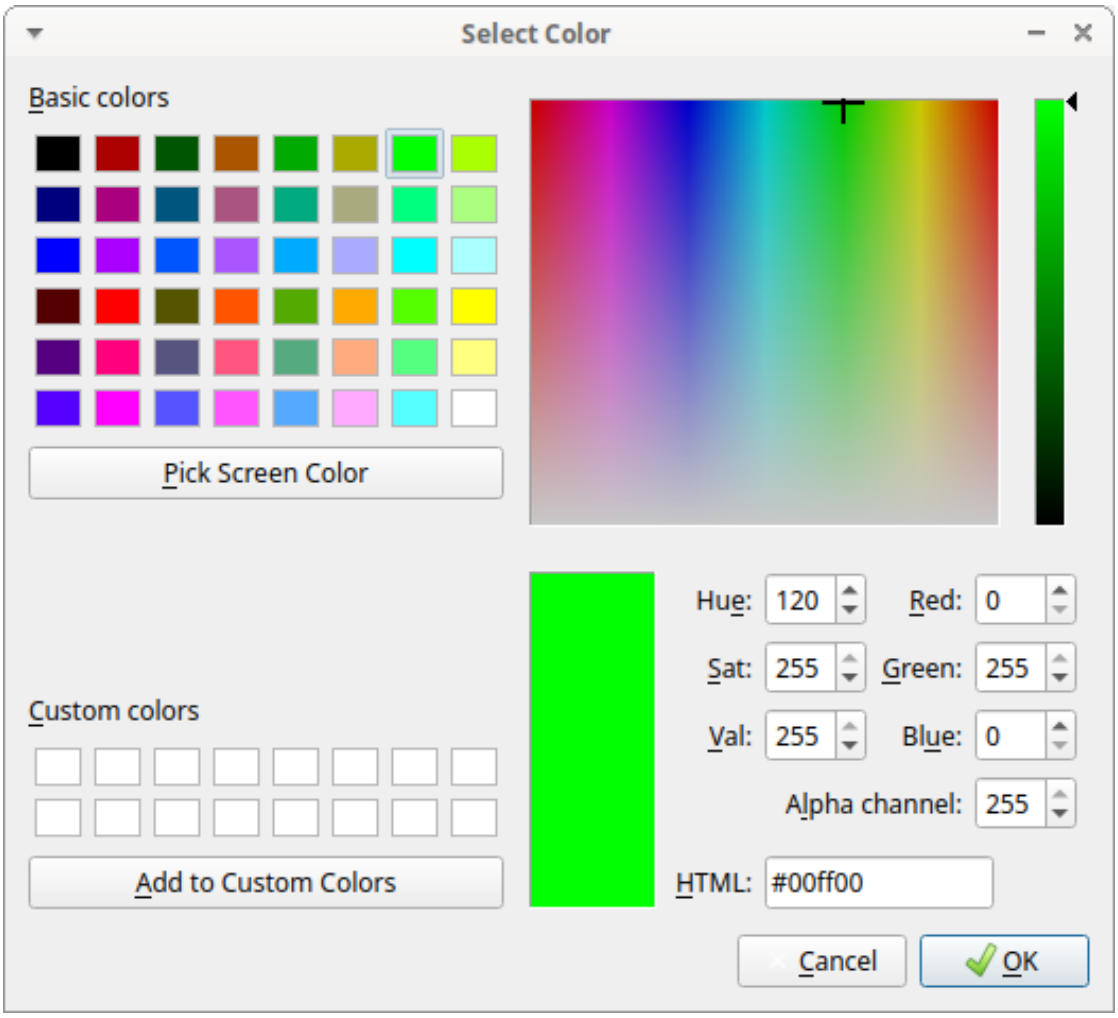
Click on **Tools** > **Geometric measurements** to activate the measurements.



The geometric measurements window

Mark color

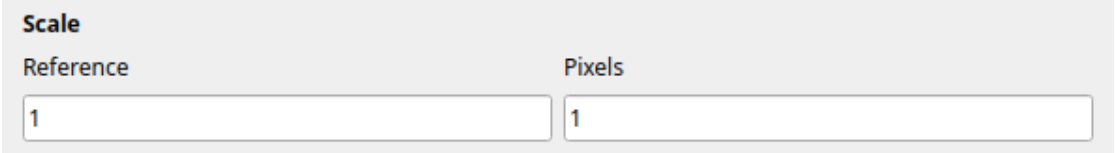
Use the **Choose color of marks** button to select a color. All marks will be drawn with the selected color. The color transparency can be set using the **Alpha channel** vale (0 for 100% transparent, 255 for a solid color).



The color selection window

Setting the scale

For distance and area measurements you can set a scale in order to have results of measurements in a real unit (like centimeters, meters etc).

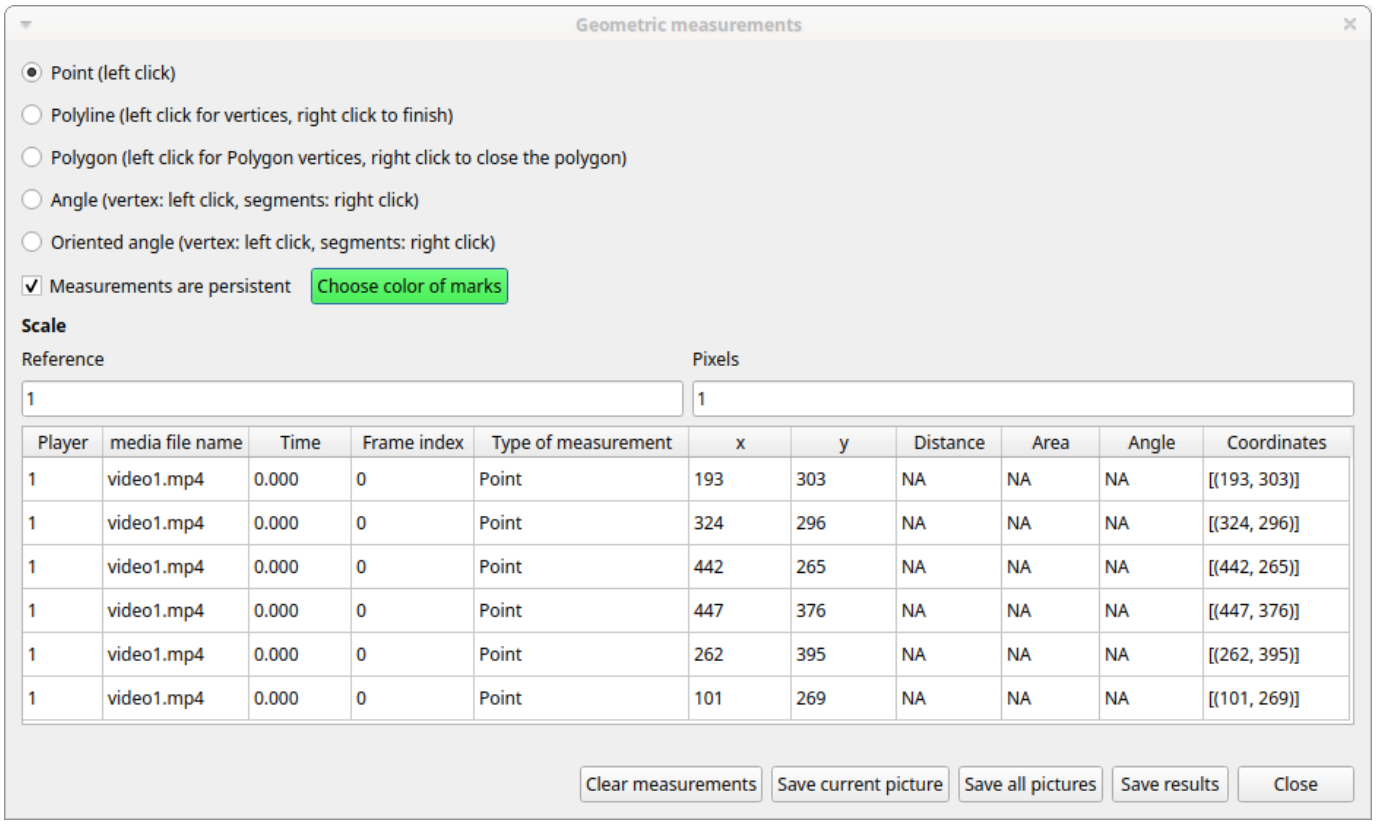


Setting the scale

1. measure a reference object (that have a known size) on the frame (with the distance tool. See next chapter for details) and set the pixel distance in the **Pixel** text box.
2. Set the real size of the reference object in the **Reference** text box (must be a number without unit).

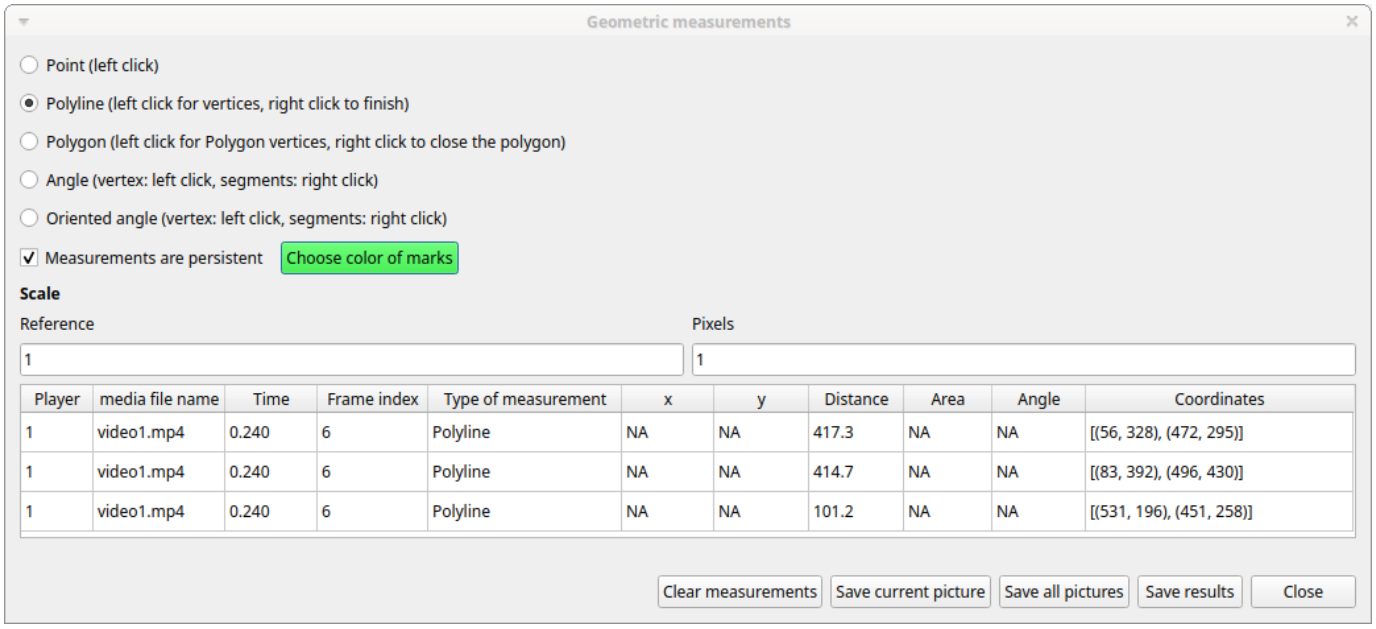
Point

Select the **Point** radio button. Click the left mouse button on the video/image to record the position of the clicked pixel.



Distance measurements

Select the **Distance** radio button. Click the left mouse button on the frame bitmap to set the start of the segment that will be measured. A circle with a cross will be drawn. Click the right mouse button to set the end. A red circle with a cross will be drawn. The distance between the two selected points will be available in the text area of the **Measurements window**.



Area measurements

Select the **Area** radio button. Click the left mouse button on the frame bitmap to set the area vertices. Circles with a cross will be drawn. Click the right mouse button to close the area. The area of the drawn polygon will be available in the text area of the **Measurements window**.

Angle measurements

Select the **Angle** radio button. Click the left mouse button on the frame bitmap to set the angle vertex. A red circle with a cross will be drawn. Click the right mouse button to set the two segments. Circles with a cross will be drawn. The angle between the two drawn segments will be available in the text area of the **Measurements window**.

Persistent measurements

If the **Measurements are persistent** checkbox is checked the measurement schemes will be available on all frames otherwise they will be deleted between frames.

The marks selected on other frames will be drawn in red.

2.9.3 Coding pad

During observation a coding pad with the available behaviors can be displayed (**Tools** > **Coding pad**). This **Coding pad** allows the user to code using a touch-screen or by clicking on the buttons. When the **Coding pad** is displayed you can continue to code using the keyboard or the ethogram.

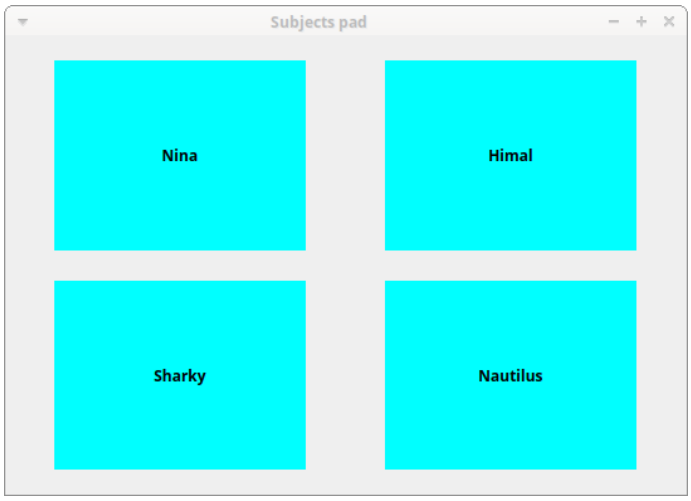
The button size can be increased or decreased.

The button color can be set for every behavior, for every behavioral category or to no color.

See the drop-down list in the upper-left corner of the Coding pad window.

2.9.4 Subjects pad

A pad with all defined subjects (or filtered subjects) can be displayed during the observation (**Tools** > **Subjects pad**). This **Subjects pad** allows the user to select the focal subject using a touch-screen or by clicking on the buttons. When the **Subjects pad** is displayed you can continue to select the focal subject using the keyboard or the subjects list.



2.9.5 Converters for external data values

Converters can be written using the Python 3 programming language.

The **INPUT** variable will be loaded with the original value of the external data file (for example 01:22:32).

The **OUPUT** variable must contain the converted value in seconds (the dot must be used for decimal separator).

Example of a code to convert HH.MM:SS format in seconds:

```
h, m, s = INPUT.split(':')
OUTPUT = int(h) * 3600 + int(m) * 60 + int(s)
```

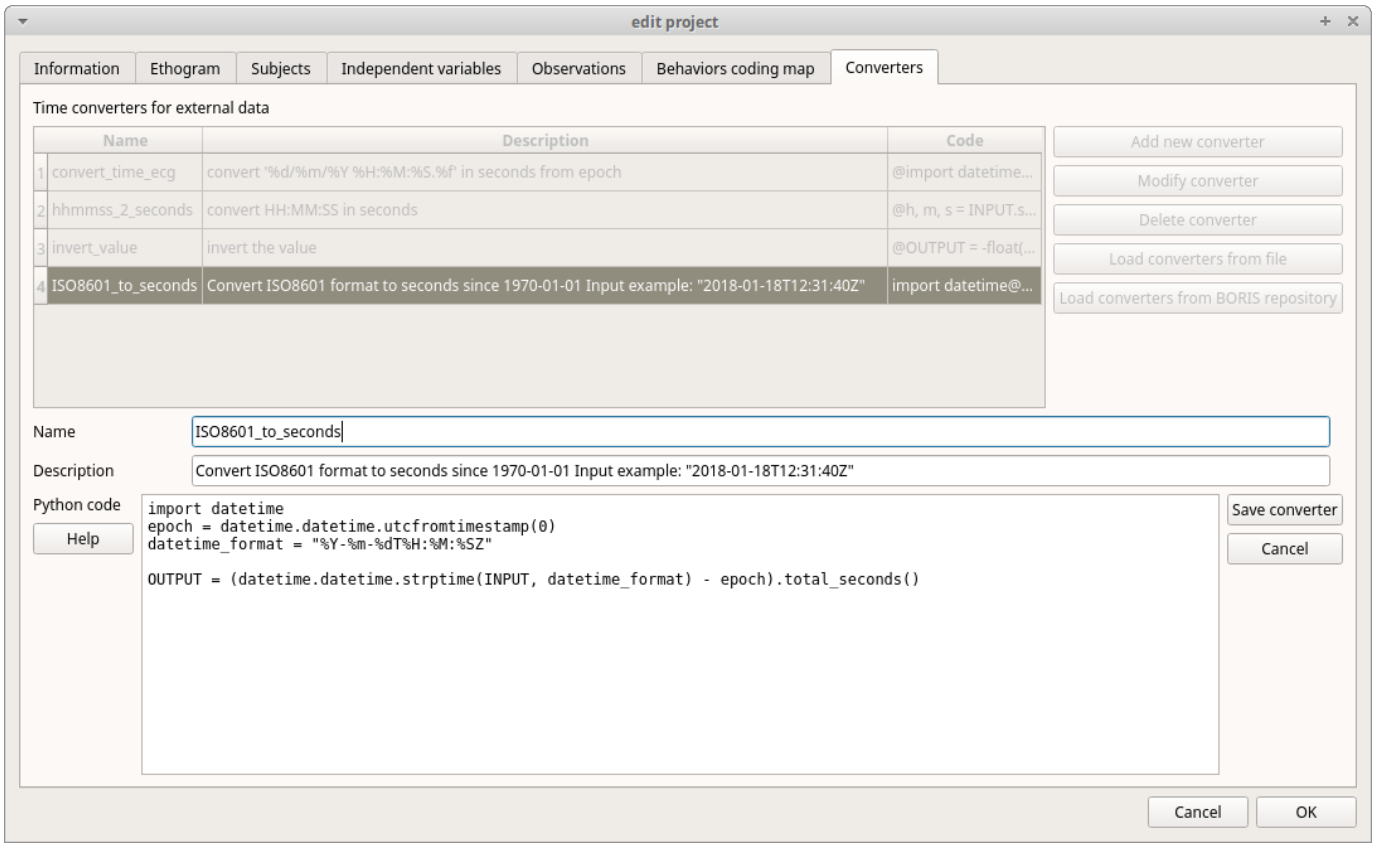
The Python function **strptime()** from the **datetime** module can be useful for converting time values: https://docs.python.org/3/library/datetime.html#strftime-strptime-behavior

Example of a code to a date in ISO8601 format in seconds using the strptime() function:

```
import datetime
epoch = datetime.datetime.utcfromtimestamp(0)
datetime_format = "%Y-%m-%dT%H:%M:%SZ"

OUTPUT = (datetime.datetime.strptime(INPUT, datetime_format) - epoch).total_seconds()
```

File > **Edit project** > **Converters**



2.9.6 Transitions flow diagram

BORIS can generate DOT scripts and flow diagrams from the transitions matrices (See Observations > Create transition matrix for obtaining the transitions matrices).

DOT script (Graphviz language)

Tools > Transitions flow diagram > Create transitions DOT script

Choose one ore more transitions matrix files and BORIS will create the relative DOT script file(s).

The DOT script files can then be used with Graphviz (Graph Visualization Software) or WebGraphviz (Graphviz in the Browser) to generate flow diagram of transitions.

See DOT (graph description language) for details.

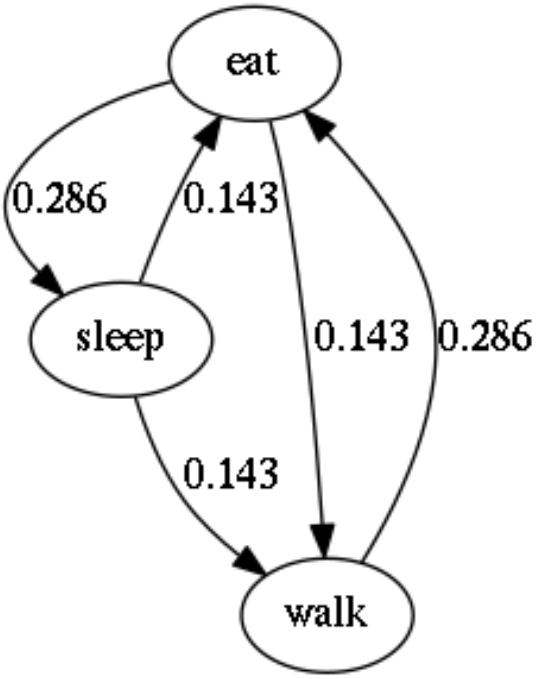
Flow diagram

If Graphviz (Graph Visualization Software) is installed on your system (and the **dot** program available in the path) BORIS can generate flow diagram (PNG format) from a transitions matrix file.

Tools > Transitions flow diagram > Create transitions flow diagram

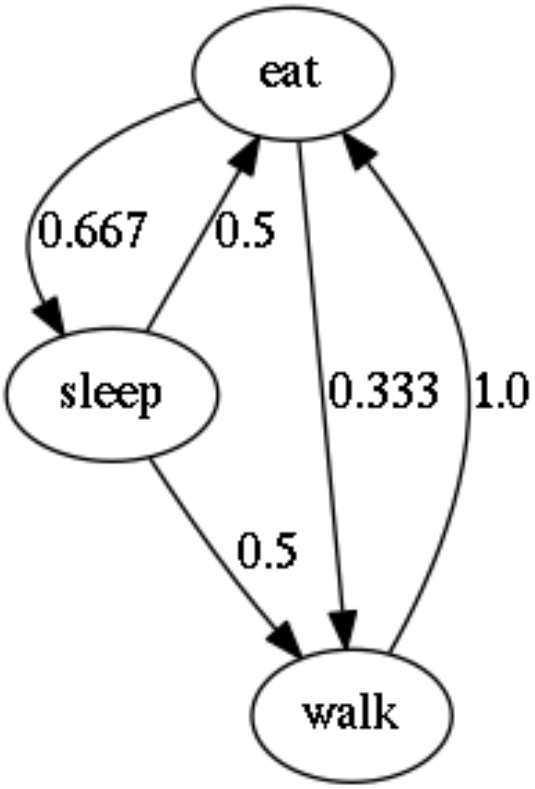
Choose one ore more transition matrix files and BORIS will create the relative flow diagram.

Flow diagram of frequencies of transitions



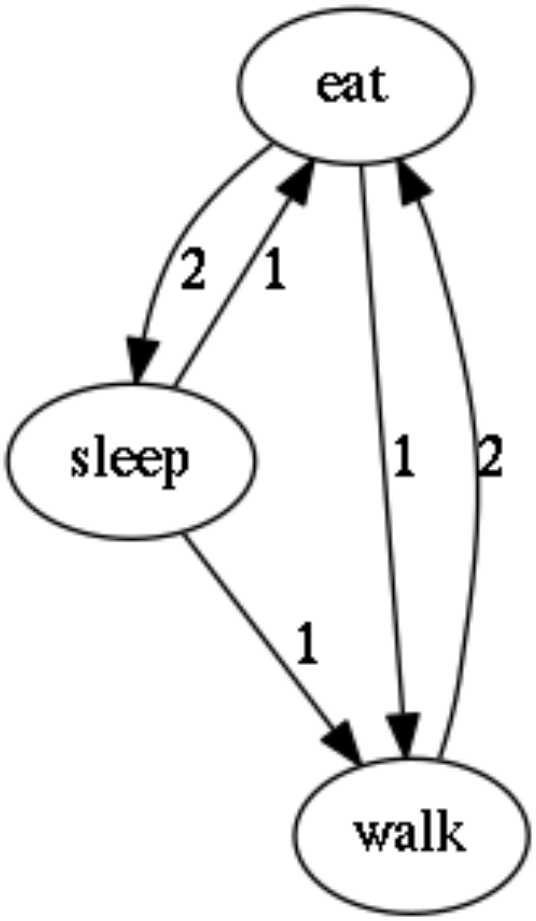
Frequencies of total transitions (the frequencies are plotted on the edges)

Flow diagram of frequencies of transitions after behavior



Frequencies of transitions after behavior (the frequencies are plotted on the edges)

Flow diagram of number of transitions



Number of transitions (the frequencies are plotted on the edges)

Re-encoding and resizing a video file

BORIS can re-encode and resize your video files in order to reduce the size of the files and have a smooth coding (specially with two video files playing together). The re-encoding and resizing operations are done with the embedded ffmpeg program with high quality parameters (bitrate 2000k).

Select the files you want re-encode and resize and select the horizontal resolution in pixels (the default is 1024). The aspect ratio will be maintained.

You can continue to use BORIS during the re-encoding/resizing operation.

The re-encoded/resized video files are renamed by adding the re-encoded.avi extension to the original files.

Rotating a video file

BORIS can rotate your video files in order to code them using the right view. The rotating operation is done with the embedded ffmpeg program using the same quality parameters then the original video.

Select the files you want rotate and select the rotation angle between: **Rotate 90 clockwise**, **Rotate 90 counter clockwise** and **Rotate 180**.

The aspect ratio will be maintained.

You can continue to use BORIS during the rotation operation.

The rotated video files are renamed by adding the **rotated\<ANGLE>** to the original file name.

2.10 Coding map

A coding map is a bitmap image with user-defined clickable areas that will help to code for behaviors or modifiers for a behavior.

2 types of coding maps are available:

- Behaviors coding map
- Modifiers coding map

2.10.1 The Behaviors coding map

BORIS allows creating a **Behaviors coding map** using the **Map creator** tool (**Tools** > **Create a coding map** > **for behaviors).**

A **Behaviors coding map** can be created only if you have defined behaviors in your ethogram.

Creating a Behaviors coding map

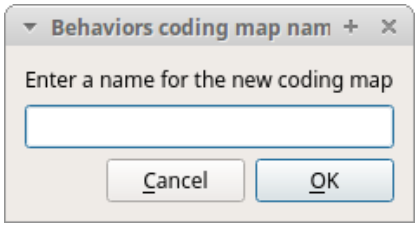
To create a new **Behaviors coding map** launch the **Behaviors coding map creator**

Tools > **Create a coding map** > **for behaviors).**

A new window will open

File > **New behaviors coding map**

Enter a name for the new **Behaviors coding map**

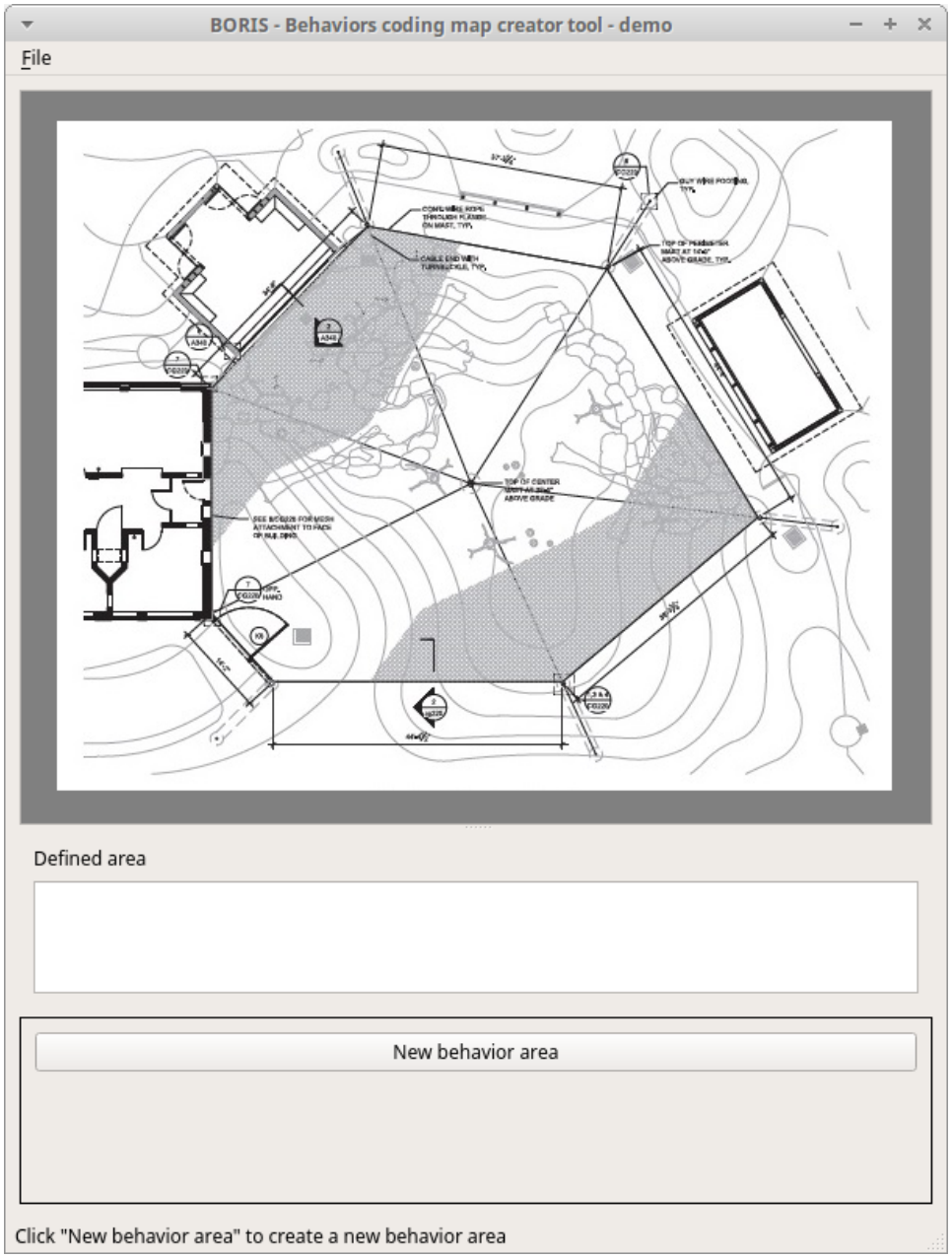


Loading a bitmap for a behaviors coding map

Click the **Load bitmap** button in the bottom of the window and select a bitmap image (PNG and JPEG formats are accepted).

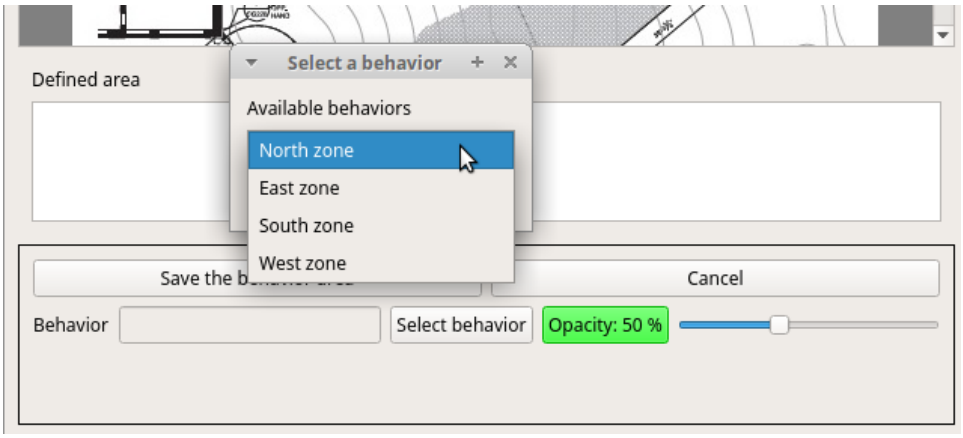
If the size of your bitmap image is bigger than 640 x 640 pixels BORIS will resize it to 640 x 640 pixels keeping the aspect ratio and store the resized version in the coding map file.

The bitmap will be displayed



Adding areas corresponding the behaviors

Click the **New behavior area** button in the bottom of the window and select a behavior by clicking on the **Select behavior** button.



The available behaviors are taken from the ethogram of the current project.

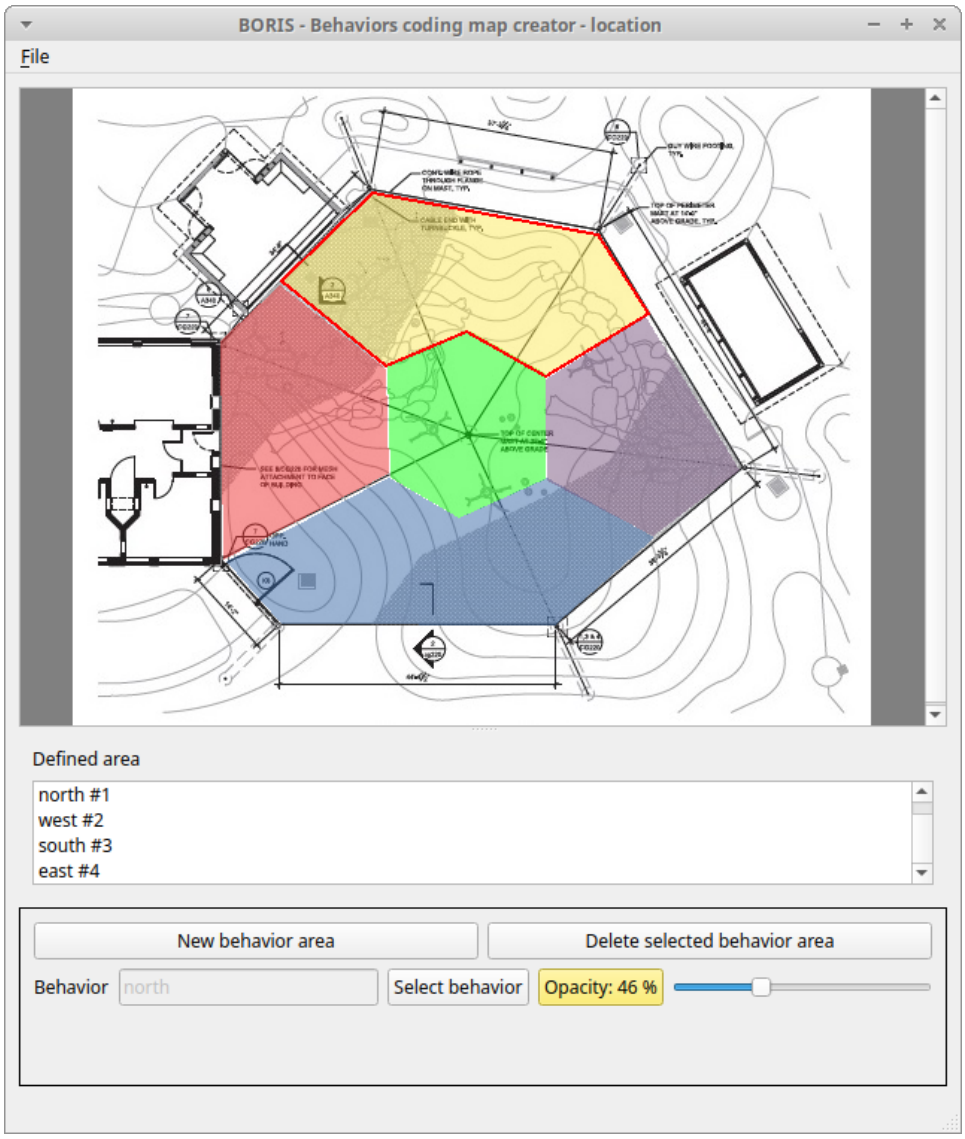
Click on the bitmap to define the vertex on the area that will code the selected behavior. Close the area by clicking again on the first point.

The color of the new area can be changed using the **Opacity** button. The opacity can be changed (from 0 to 100%) using the slider.

Save the behavior area by clicking on the **Save the behavior area** button

The area will be added to the **Defined area** list

You can add more area and also add more than one area for a same behavior. Two or more areas can overlap. In this case all corresponding behaviors will be triggered.



Add the Behaviors coding map to the current project

File > **Add coding map to project**

The coding map will be added to the current project

You can add a **Behaviors coding map** to the current project from a file containing the coding map:

(**File** > **Edit project** > **Behaviors coding map** > **Add a behaviors coding map**)

Saving the Behaviors coding map

Saving the **Behaviors coding map** will create a file containing the **Behaviors coding map** including the bitmap image.

File > **Save the current Behaviors coding map**

The file containing the **Behaviors coding map** can be then reloaded in the **Behaviors coding map creator** or added to a BORIS project (**File** > **Edit project** > **Behaviors coding map** > **Add a behaviors coding map**)

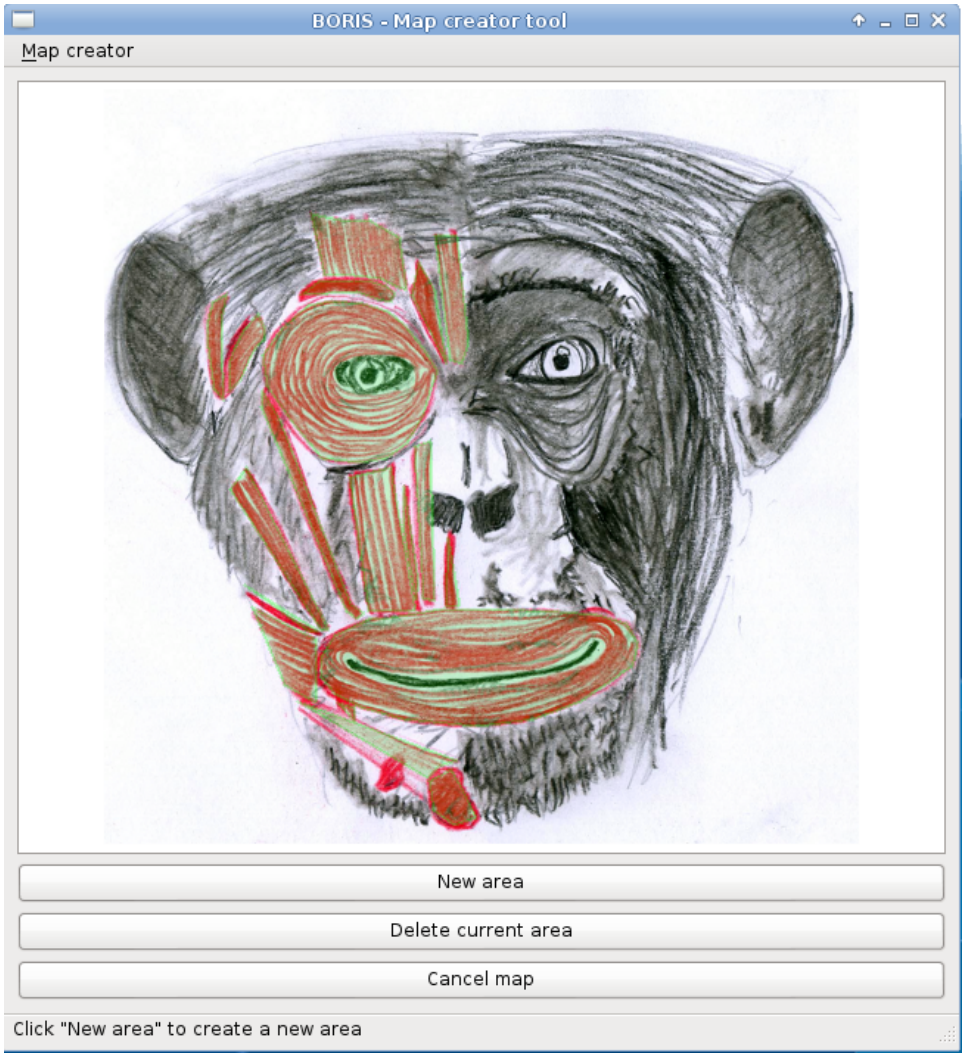
2.10.2 The Modifiers coding map

BORIS allows creating a modifiers coding map using the **Modifiers Map creator** tool (**Tools** > **Create a coding map** > **for modifiers**.) Clickable areas may correspond to specific modifiers that can be meaningful for the behavioral coding. Facial expression is the case we thought to when developing this function.

Creating a modifiers coding map

Loading a bitmap for a modifiers coding map

To create a new **Modifiers coding map**, launch the **Modifiers Map creator** tool (**Tools** > **Create a coding map** > **for modifiers).** The BORIS main window will be replaced by the **Modifiers Map creator** window. Click on **Modifiers Map creator** > **New Modifiers map** and enter a name for the new map in the edit box. You have to load a bitmap image (JPEG or PNG) using the **Load bitmap** button. The loaded image will be displayed.



If the size of your bitmap image is bigger than 640 x 640 pixels BORIS will resize it to 640 x 640 pixels keeping the aspect ratio and store the resized version in the coding map file.

Adding areas corresponding to the modifiers

To create clickable areas on a coding map, you have to click on the **New area** button and enter an **Area code** in the edit box. The new area can now be defined by clicking on the image. The drawing tool allows defining a irregular polygon (a plane shape with straight sides, which does not have all sides equal and all angles equal) by clicking to determine subsequent vertices. It can be convex or concave. Straight sides must not cross each other. Once selected an area can be deleted using the **Delete area** button. When an area is closed and its name has been defined in the **Area code** field, it can be saved by using the **Save area** button. The areas can partially overlap each other. See the **Using a Coding map** section for more details. Once

all areas are added the entire map can be saved using the **Save map** option menu (**Map creator** > **Save map**). The map is now saved in its own file (.boris_map) which is NOT part of the BORIS project. A map can be edited at anytime by opening the map file from the **Open map** menu option (**Map creator** > **Open map**).

Adding a modifiers coding map to your project

Creating a Coding map is not automatically adding the map to your project. The Coding map have to be added to your project by selecting the corresponding **Behavior type** (**Point event with coding map**, **State event with coding map**). BORIS will ask to select the file name containing the coding map (.boris_map) and load the coding map in the project. The coding map name will appear in the **Coding map** column and will be saved in the BORIS project file.

> **Important**
>
> If you later modify your coding map you must reload the new version in your BORIS project.

2.11 Analysis and plot

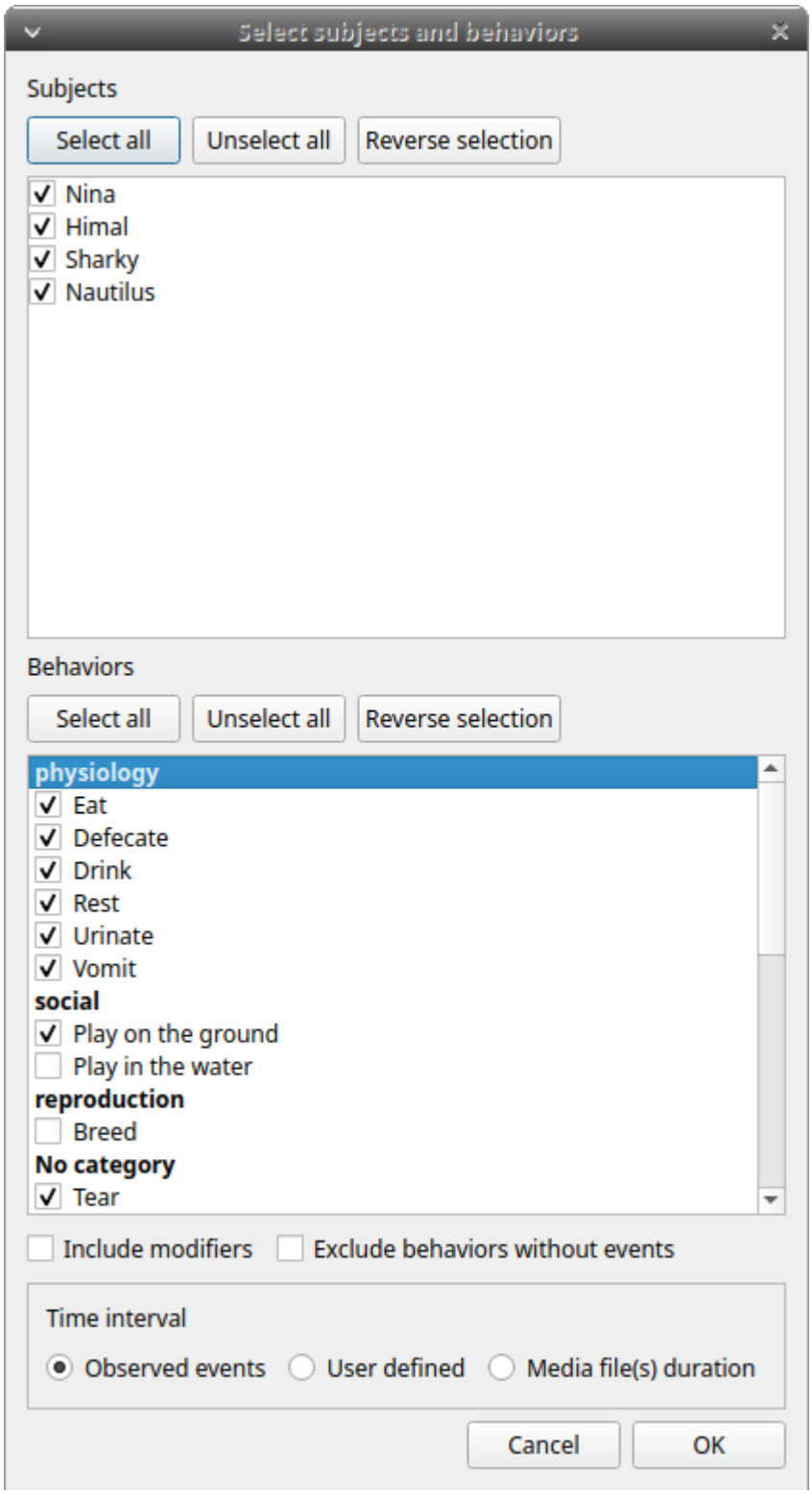
2.11.1 Time budget analysis

The **Time budget analysis** can be done by behavior (including or not the modifiers) or by category of behaviors. Choose the option from the **Analysis** menu.

The **Time budget analysis** can be done on one or more observations. If you select more than one observation you must then choose for a global time budget analysis that will contain all selected observations or a time budget analysis for every single observation.

Choose **Yes** to group all observations in one time budget analysis

The **Analysis** > **Time budget** menu option shows the time budget for the events of the selected observations. Select the subjects and behaviors you want to include in the time budget analysis:



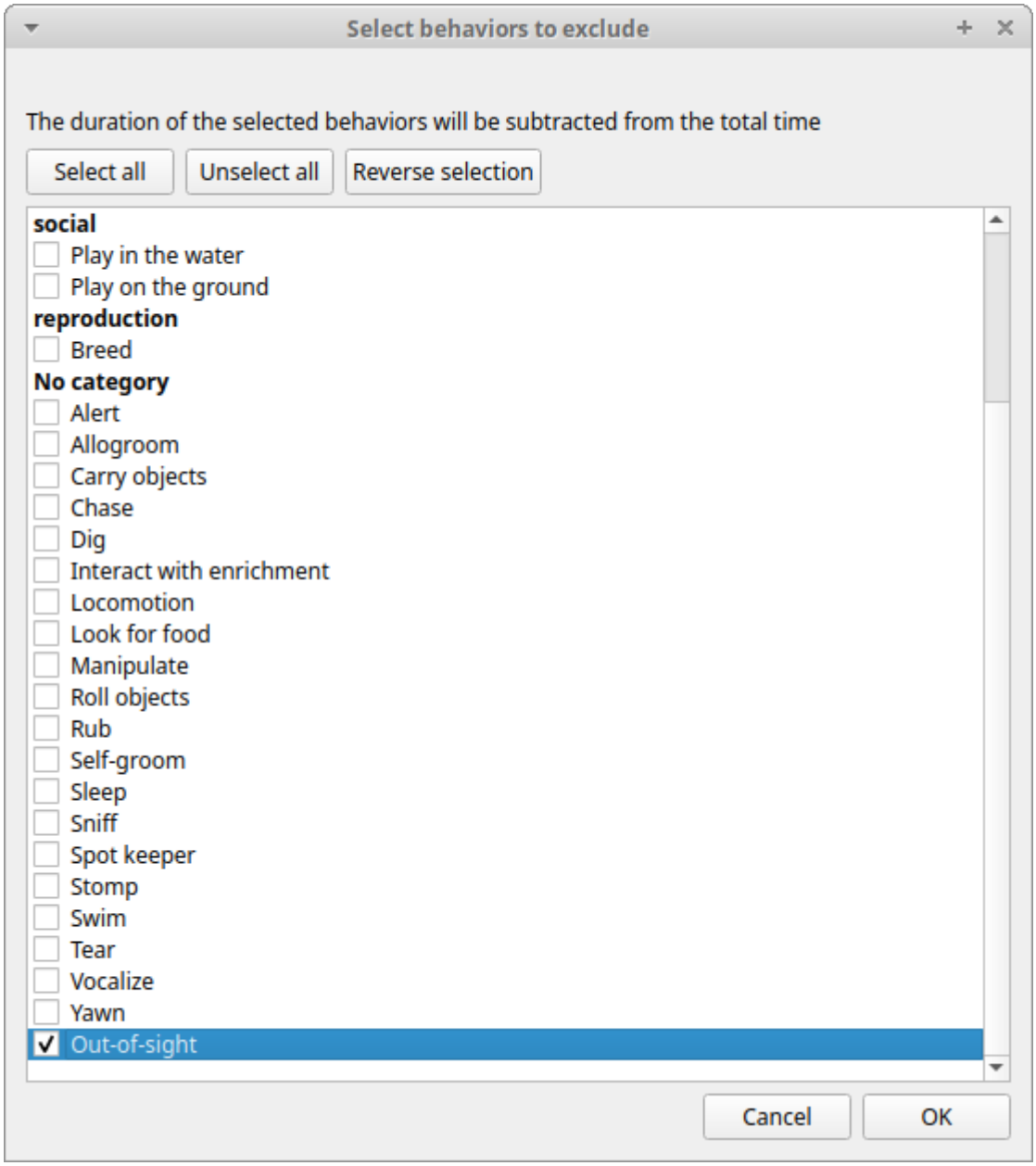
All behaviors can be selected or unselected by clicking on the Category (bold).

You can choose to include or not the behavior modifiers in the Time budget analysis and to exclude behaviors without coded events.

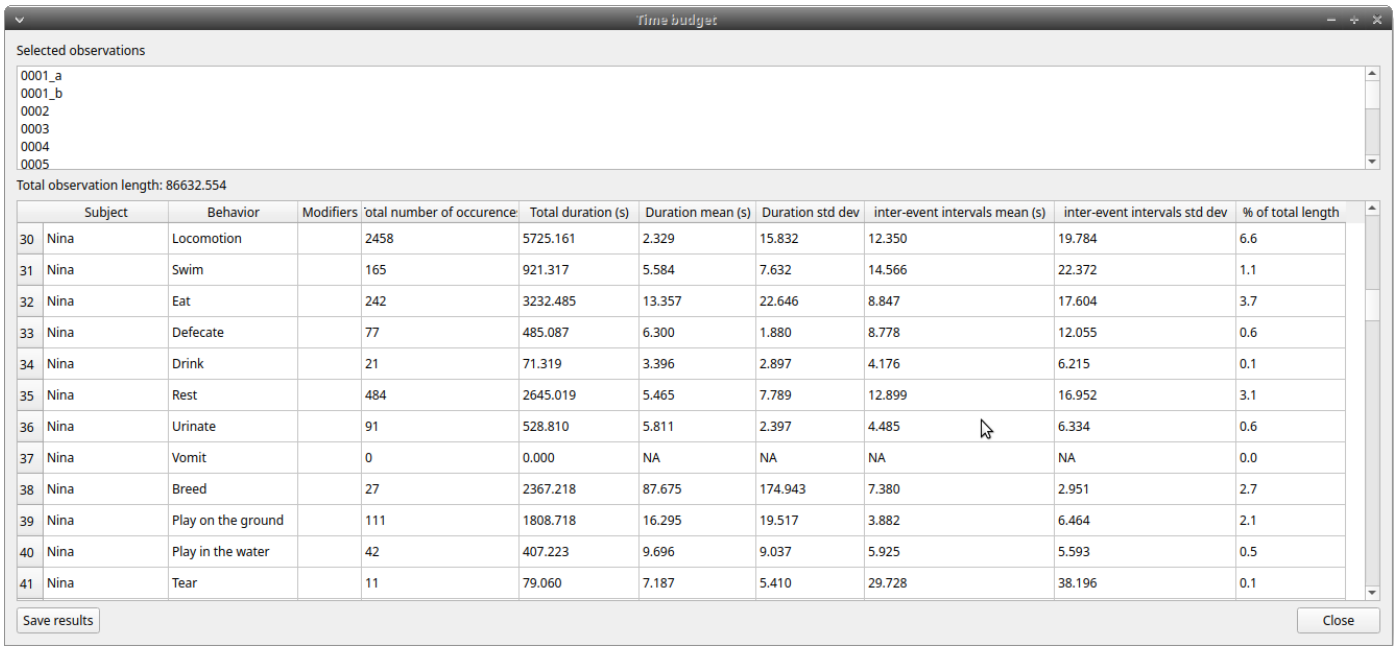
The Time budget analysis can be restricted to a part of the observation:

- Full observation(s): the analysis will be made on the full observation length.
- Limit to time interval: use the **Start time** and **End time** boxes to select starting time and ending time.
- Limit to observed events: the analysis will be made from the first observed event to the last observed event.

The last dialog window will allow you to subtract the duration of one or more behaviors from the total duration of the observation. This can help you if you have defined a "out-of-sight" code in your ethogram for example:



The results contain for each subject and behavior the **total nuber of occurrences**, the **total duration** (for the behaviors defined as state events), The **duration mean** (for the behaviors defined as state events), the **standard deviation of duration**, the **inter-events intervals duration mean**, th **standard deviation of the inter-events intervals duration** and the **percent of total duration of observation(s)**. All duration times are expressed in seconds (s).



	Subject	Behavior	Modifiers	otal number of occurence	Total duration (s)	Duration mean (s)	Duration std dev	inter-event intervals mean (s)	inter-event intervals std dev	% of total length
30	Nina	Locomotion		2458	5725.161	2.329	15.832	12.350	19.784	6.6
31	Nina	Swim		165	921.317	5.584	7.632	14.566	22.372	1.1
32	Nina	Eat		242	3232.485	13.357	22.646	8.847	17.604	3.7
33	Nina	Defecate		77	485.087	6.300	1.880	8.778	12.055	0.6
34	Nina	Drink		21	71.319	3.396	2.897	4.176	6.215	0.1
35	Nina	Rest		484	2645.019	5.465	7.789	12.899	16.952	3.1
36	Nina	Urinate		91	528.810	5.811	2.397	4.485	6.334	0.6
37	Nina	Vomit		0	0.000	NA	NA	NA	NA	0.0
38	Nina	Breed		27	2367.218	87.675	174.943	7.380	2.951	2.7
39	Nina	Play on the ground		111	1808.718	16.295	19.517	3.882	6.464	2.1
40	Nina	Play in the water		42	407.223	9.696	9.037	5.925	5.593	0.5
41	Nina	Tear		11	79.060	7.187	5.410	29.728	38.196	0.1

Results of the time budget analysis

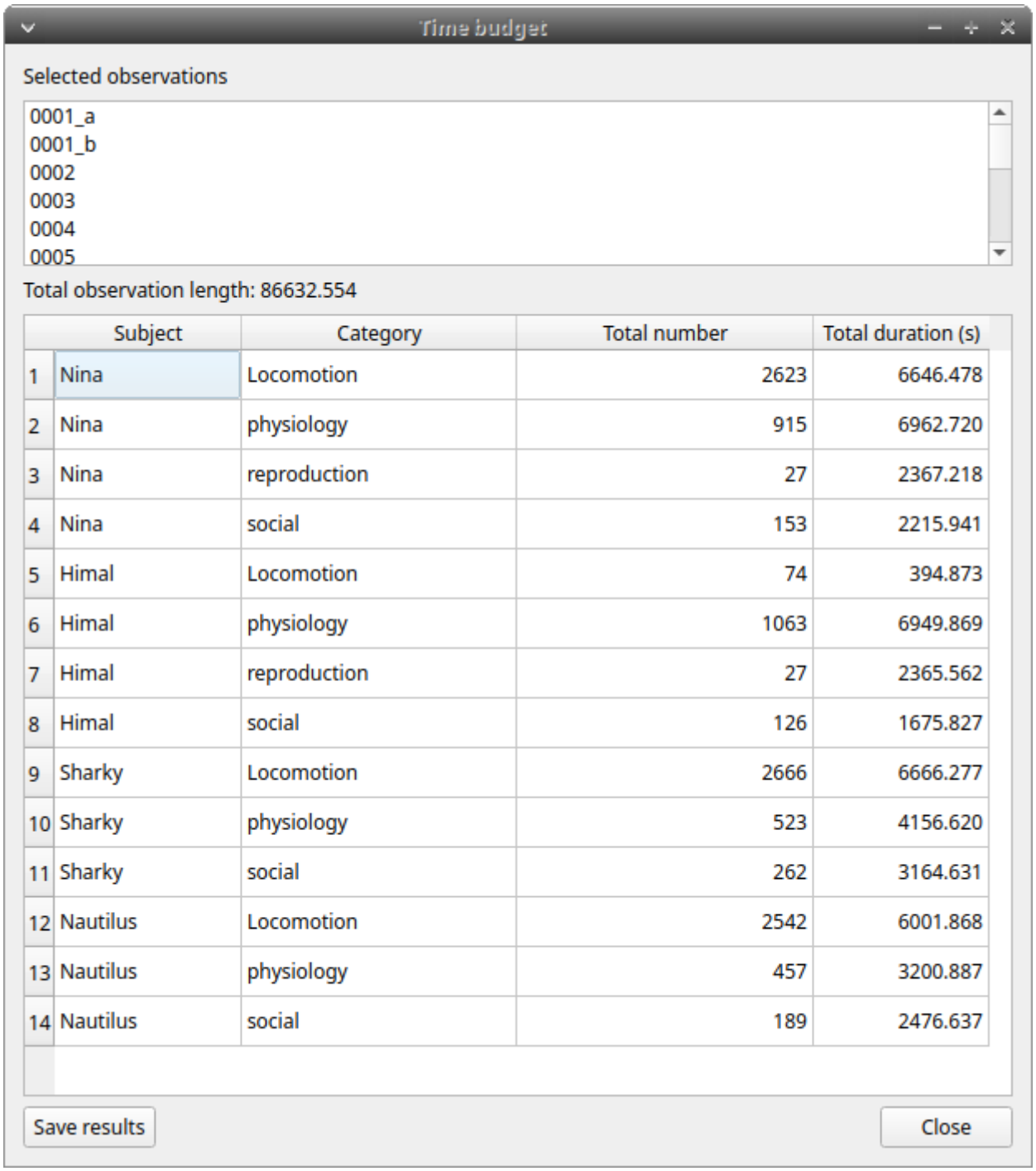
The time budget results can be saved in various formats for further analysis:

- Plain text in tabular format
  - **Tab Separated Values** (TSV)
  - **Comma Separated Values** (CSV)
  - **Hyper Text Markup language** (HTML)
- Spreadsheet files
  - **OpenDocument** (ODS)
  - **Microsoft Excel** (XLSX, XLS)
- **Pandas dataframe** (to be loaded in Python with the pickle module)
- **R dataframe** (to be loaded in R with readRDS function)

> **Important**
>
> If a STATE behavior has an odd number of coded events, BORIS will report **UNPAIRED** instead of results.

2.11.2 Time budget by behavioral category

The **Time budget by behavioral category** is similar to the **Time budget analysis** except that the behaviors are grouped into **behavioral categories**.

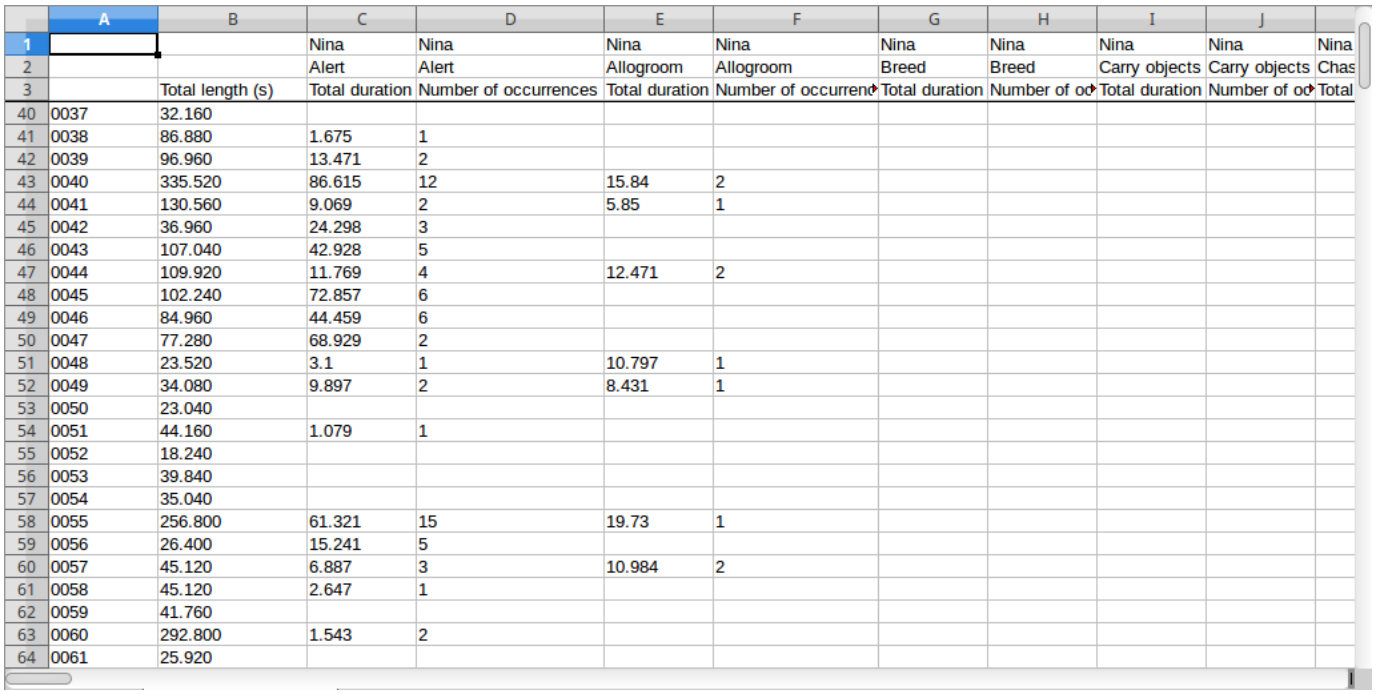


	Subject	Category	Total number	Total duration (s)
1	Nina	Locomotion	2623	6646.478
2	Nina	physiology	915	6962.720
3	Nina	reproduction	27	2367.218
4	Nina	social	153	2215.941
5	Himal	Locomotion	74	394.873
6	Himal	physiology	1063	6949.869
7	Himal	reproduction	27	2365.562
8	Himal	social	126	1675.827
9	Sharky	Locomotion	2666	6666.277
10	Sharky	physiology	523	4156.620
11	Sharky	social	262	3164.631
12	Nautilus	Locomotion	2542	6001.868
13	Nautilus	physiology	457	3200.887
14	Nautilus	social	189	2476.637

Results of a time budget by behavioral category analysis

2.11.3 Synthetic time budget

The synthetic time budget is similar to time budget but with fewer parameters and a different organization of results. Results of all selected observations are organized in columns on a single page. Two parameters are provided for now: **number of occurrences** and **total duration** (for the behaviors defined as state events)



	A	B	C	D	E	F	G	H	I	J	
1			Nina	Nina	Nina	Nina	Nina	Nina	Nina	Nina	Nina
2			Alert	Alert	Allogroom	Allogroom	Breed	Breed	Carry objects	Carry objects	Chas
3		Total length (s)	Total duration	Number of occurrences	Total duration	Number of occurrenc	Total duration	Number of oc	Total duration	Number of oc	Total
40	0037	32.160									
41	0038	86.880	1.675	1							
42	0039	96.960	13.471	2							
43	0040	335.520	86.615	12	15.84	2					
44	0041	130.560	9.069	2	5.85	1					
45	0042	36.960	24.298	3							
46	0043	107.040	42.928	5							
47	0044	109.920	11.769	4	12.471	2					
48	0045	102.240	72.857	6							
49	0046	84.960	44.459	6							
50	0047	77.280	68.929	2							
51	0048	23.520	3.1	1	10.797	1					
52	0049	34.080	9.897	2	8.431	1					
53	0050	23.040									
54	0051	44.160	1.079	1							
55	0052	18.240									
56	0053	39.840									
57	0054	35.040									
58	0055	256.800	61.321	15	19.73	1					
59	0056	26.400	15.241	5							
60	0057	45.120	6.887	3	10.984	2					
61	0058	45.120	2.647	1							
62	0059	41.760									
63	0060	292.800	1.543	2							
64	0061	25.920									

All duration times are expressed in seconds (s).

The time budget results can be saved in various formats for further analysis:

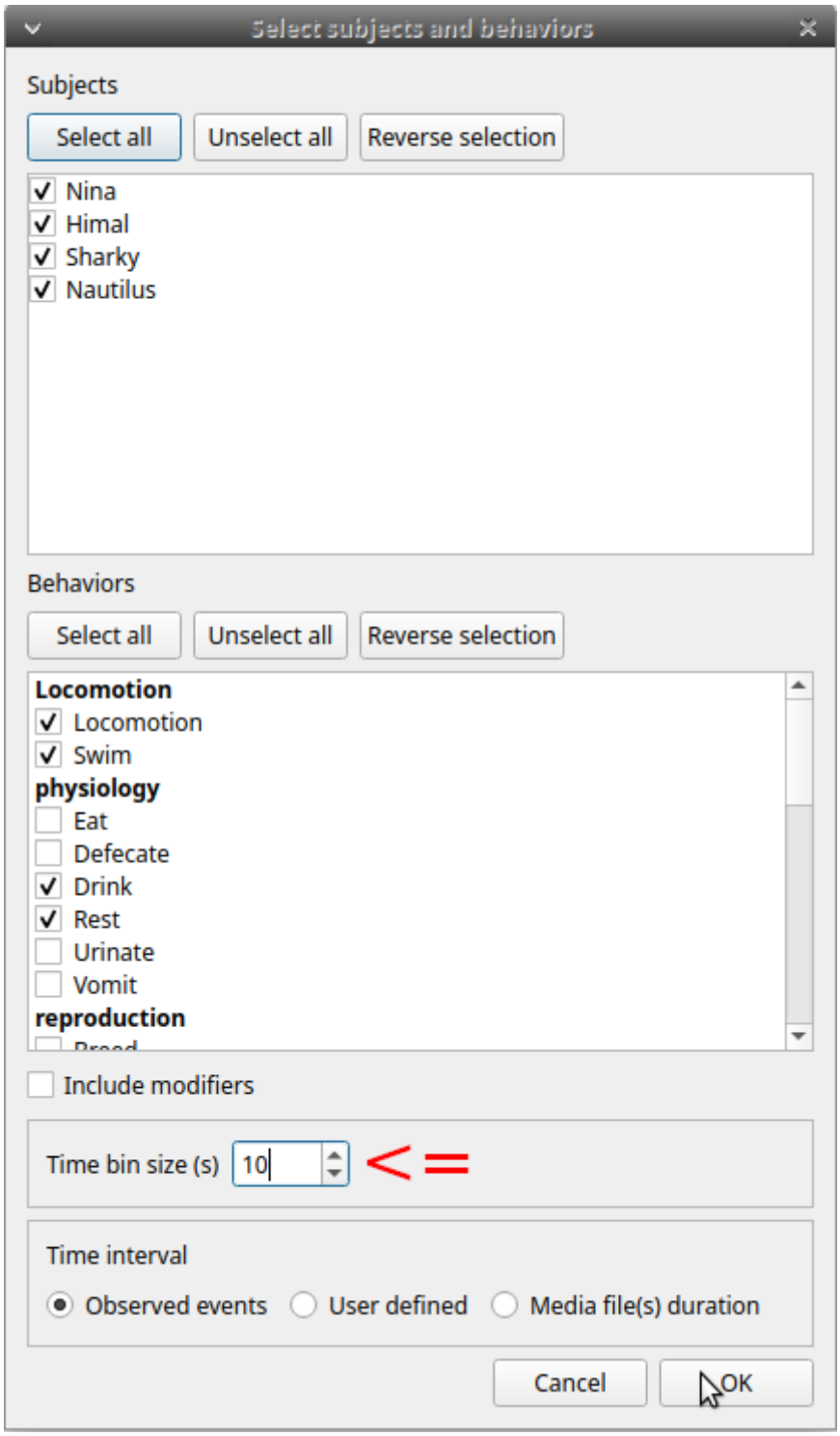
- Plain text in tabular format
  - **Tab Separated Values** (TSV)
  - **Comma Separated Values** (CSV)
  - **Hyper Text Markup language** (HTML)
- Spreadsheet files
  - **OpenDocument** (ODS)
  - **Microsoft Excel** (XLSX, XLS)

2.11.4 Synthetic time budget with time bin

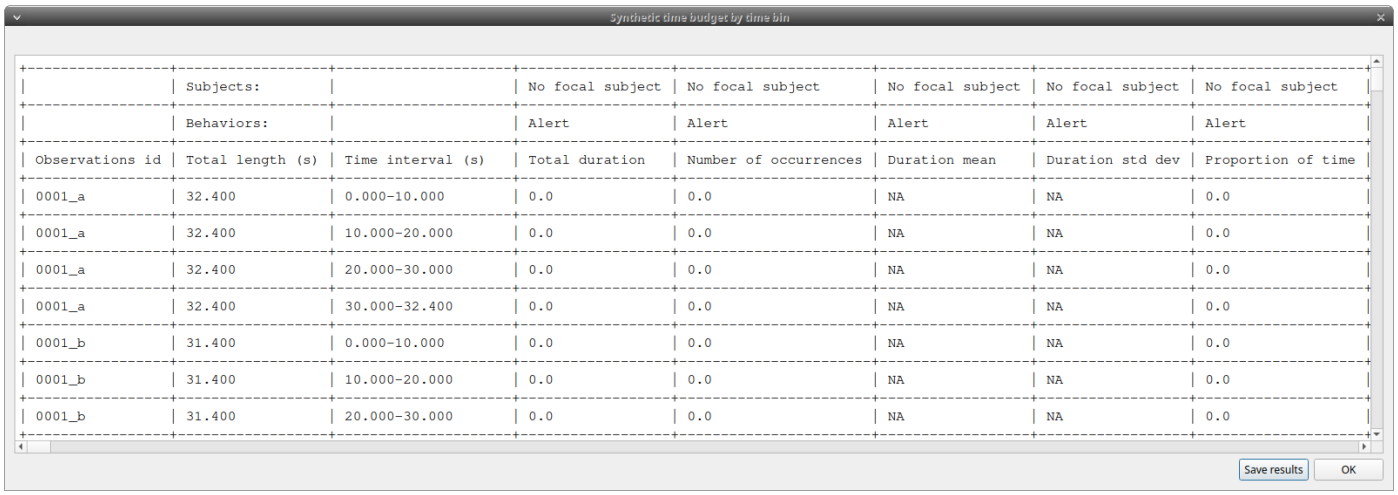
The **synthetic time budget with time bin** is similar to the **Synthetic time budget** but the results are divised in time bin.

Analysis > **Synthetic time budget with time bin**

Choose a time bin size (in seconds)



Time bin size of 10 seconds



	Subjects:		No focal subject	No focal subject	No focal subject	No focal subject	No focal subject
	Behaviors:		Alert	Alert	Alert	Alert	Alert
Observations id	Total length (s)	Time interval (s)	Total duration	Number of occurrences	Duration mean	Duration std dev	Proportion of time
0001_a	32.400	0.000-10.000	0.0	0.0	NA	NA	0.0
0001_a	32.400	10.000-20.000	0.0	0.0	NA	NA	0.0
0001_a	32.400	20.000-30.000	0.0	0.0	NA	NA	0.0
0001_a	32.400	30.000-32.400	0.0	0.0	NA	NA	0.0
0001_b	31.400	0.000-10.000	0.0	0.0	NA	NA	0.0
0001_b	31.400	10.000-20.000	0.0	0.0	NA	NA	0.0
0001_b	31.400	20.000-30.000	0.0	0.0	NA	NA	0.0

Results of a Synthetic time budget with time bin of 10 seconds

The **time budget with time bin** results can be saved in various formats for further analysis:

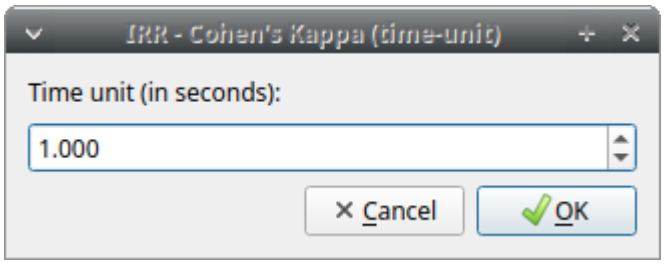
- Plain text in tabular format
  - **Tab Separated Values** (TSV)
  - **Comma Separated Values** (CSV)
  - **Hyper Text Markup language** (HTML)
- Spreadsheet files
  - **OpenDocument** (ODS)
  - **Microsoft Excel** (XLSX, XLS)

2.11.5 Inter-rater reliability

The Cohen's kappa coefficient can be calculated (**Analysis** > **Inter-rater reliability** > **Cohen\'s kappa**).

Cohen's kappa on Wikipedia

After selecting 2 observations and a time window (in seconds) for the analysis (the default value is 10 seconds) the Cohen's kappa will be displayed in the results window.



Implementation of IRR Cohen's Kappa in BORIS

If a time window of n seconds is set the 2 selected observations will be checked every n seconds for agreement/disagreement from the first event to the last event of the 2 observations . In case of a point event the presence of a corresponding event in the other observation will be verified using a time window of n seconds centered on the point event.

A IRR Cohen's Kappa analysis is available in the GSEQ program (http://www2.gsu.edu/~psyrab/gseq). For this the coded events can be exported as aggregated events in SDIS format. See export aggregated events.

2.11.6 Similarities

Needleman-Wunsch similarity

2.11.7 Co-occurence

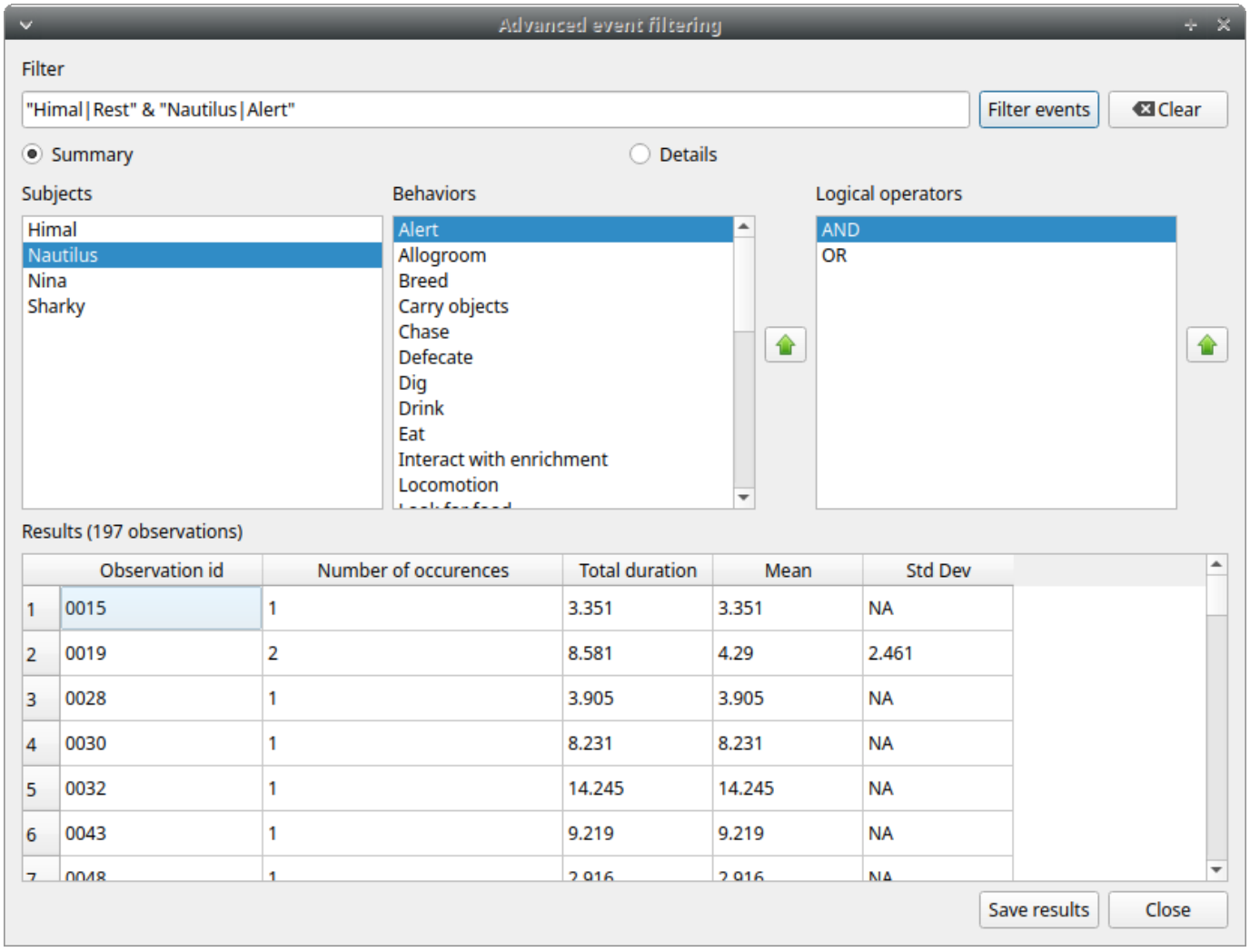
This function allow to determine the co-occurence of 2 behaviors.

2.11.8 Advanced event filtering

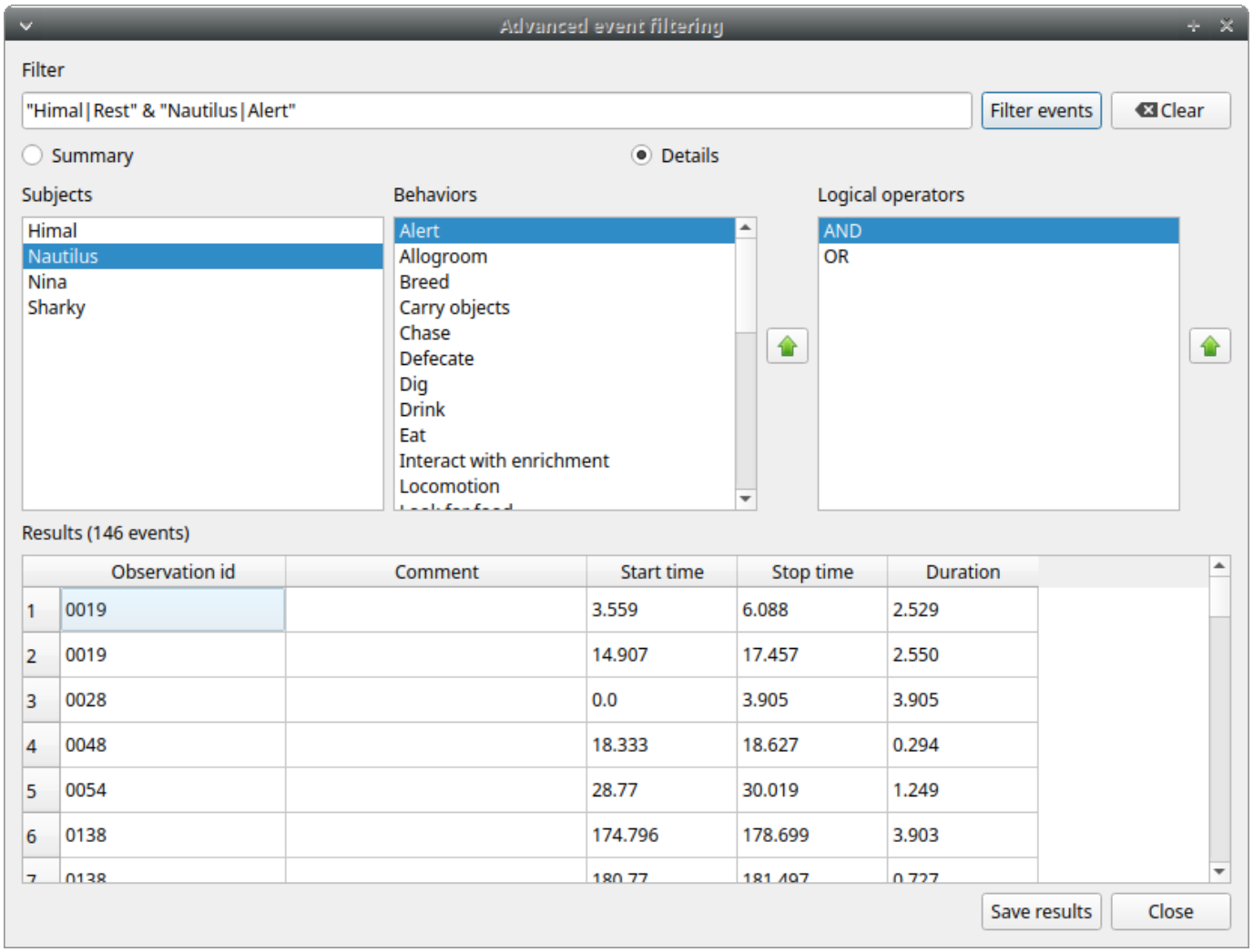
This function allows to filter events from one or more observations by selecting subjects, behaviors and logical operators.

To use the filter, select a subject, select a behavior and click on the button with the green arrow on the side of the behaviors list. The tuple subject/behavior will be added in the **filter text edit**. A complex filter query can be constructed by adding parenthesis and logical operator **&** (AND) or | (OR) for combining various subjects and behaviors.

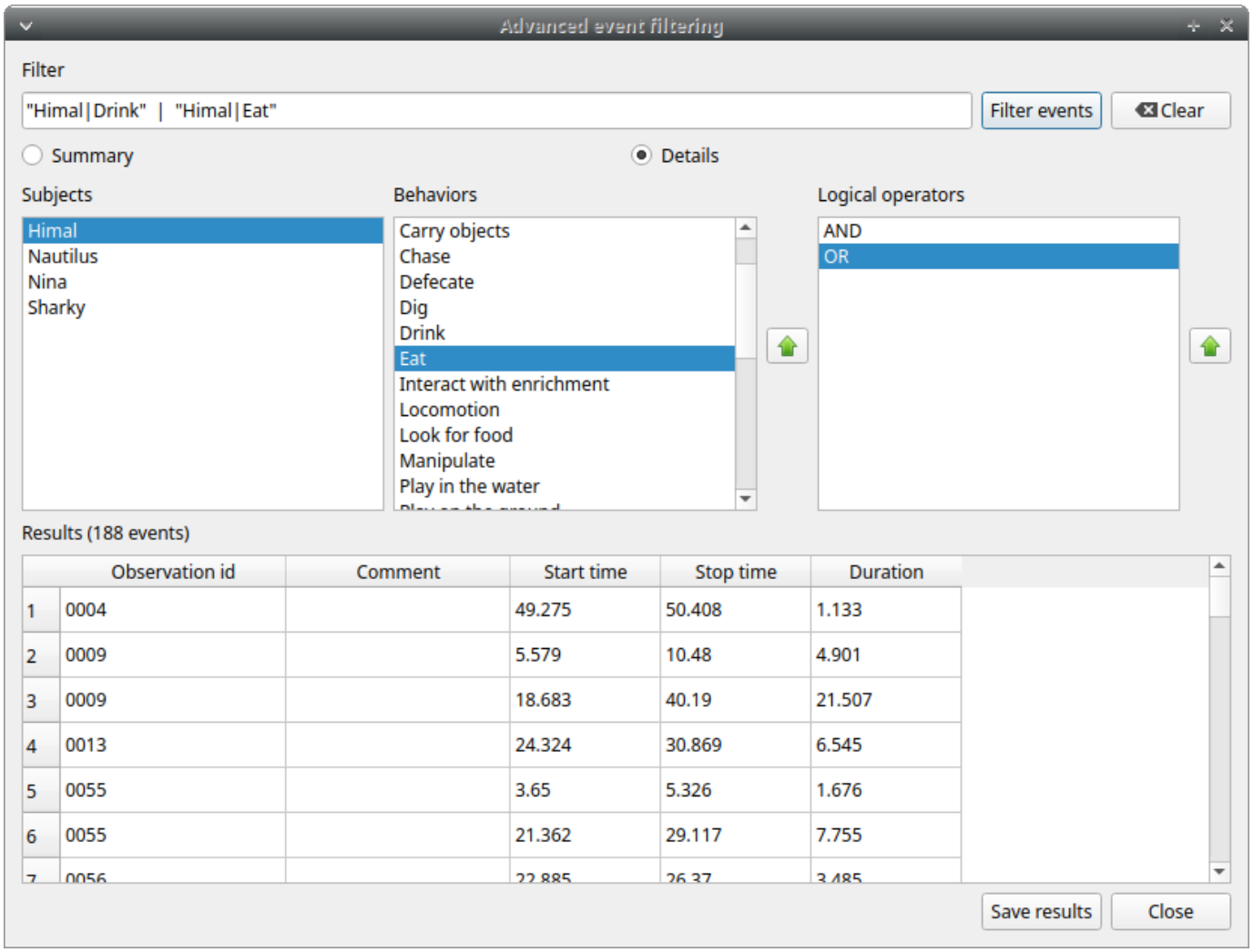
Example of a summarized output showing the occurences of Himal resting while Nautilus in alert:



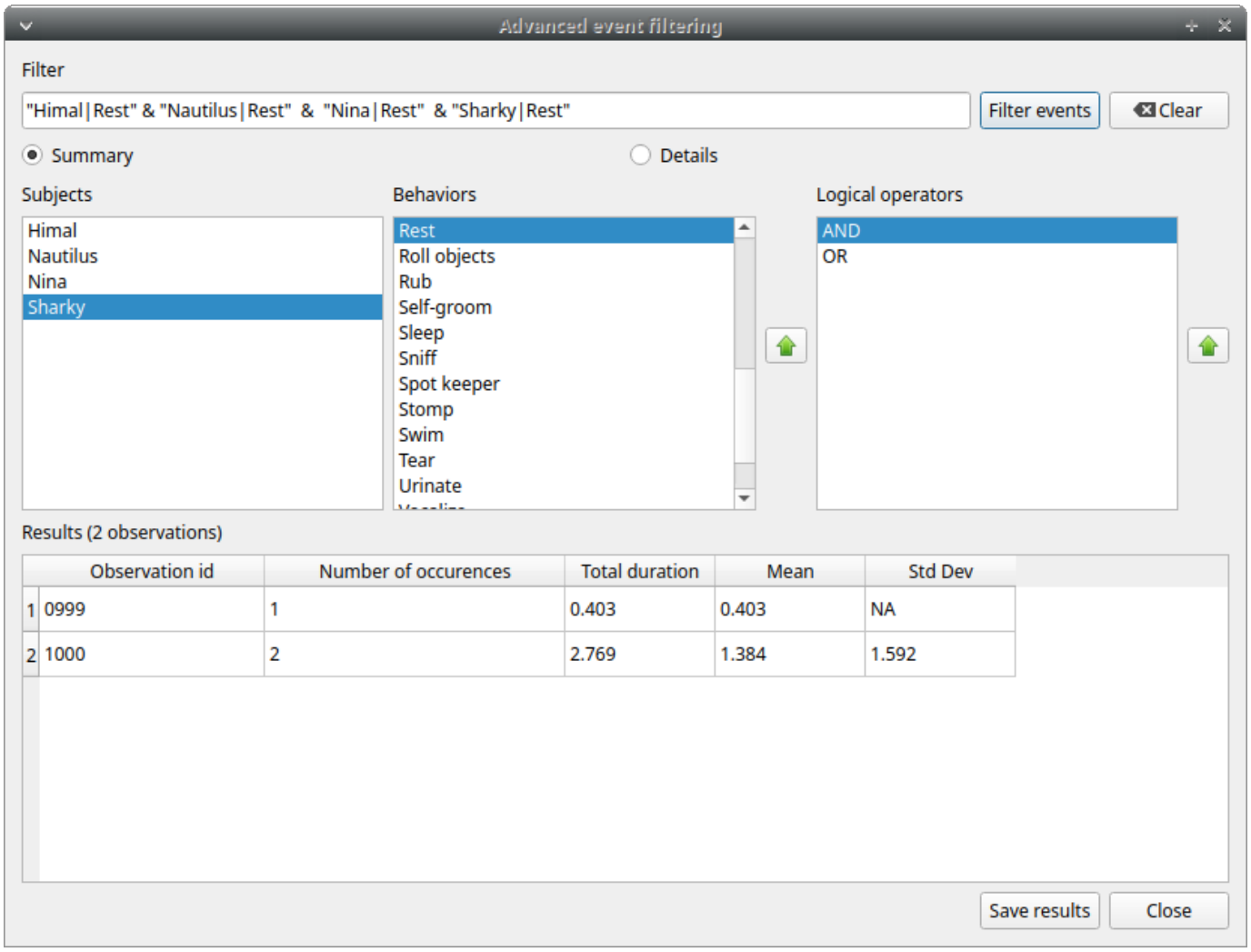
Example of a detailed output showing the overlaping intervals while Himal rests and Nautilus is in alert:



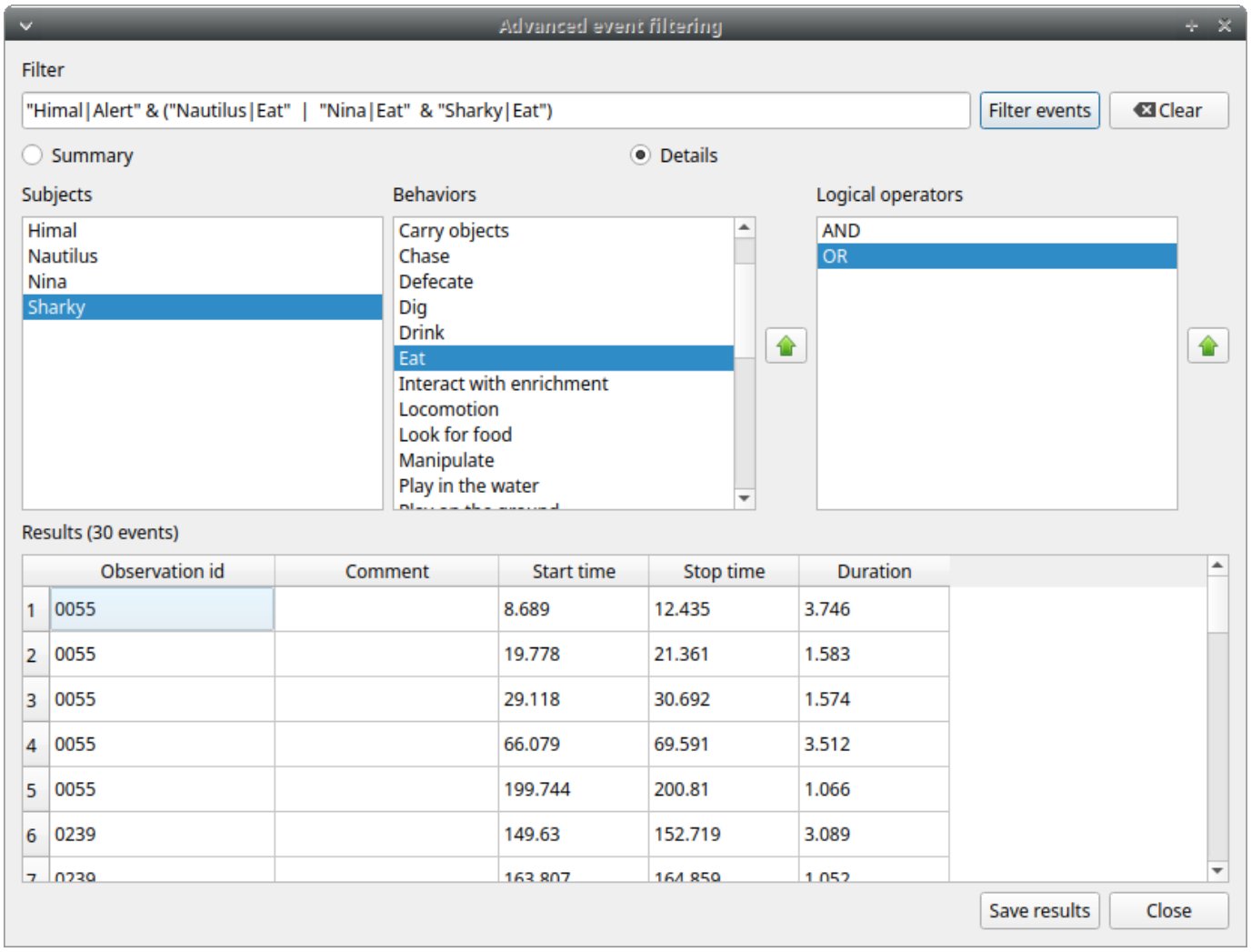
The same subject can be used many times in the query with OR or AND (in case of non exclusive behaviors):



An unlimited number of conditions can be used:



Parenthesis can be used to group logical conditions into block(s):



The results can be saved in a Tab Separted Values (TSV) file using the **Save results** button. Other formats will be added in future.

2.11.9 Latency

The latency will analyze the time between one or more markers (arbitrary behaviors(s)) and other behaviors.

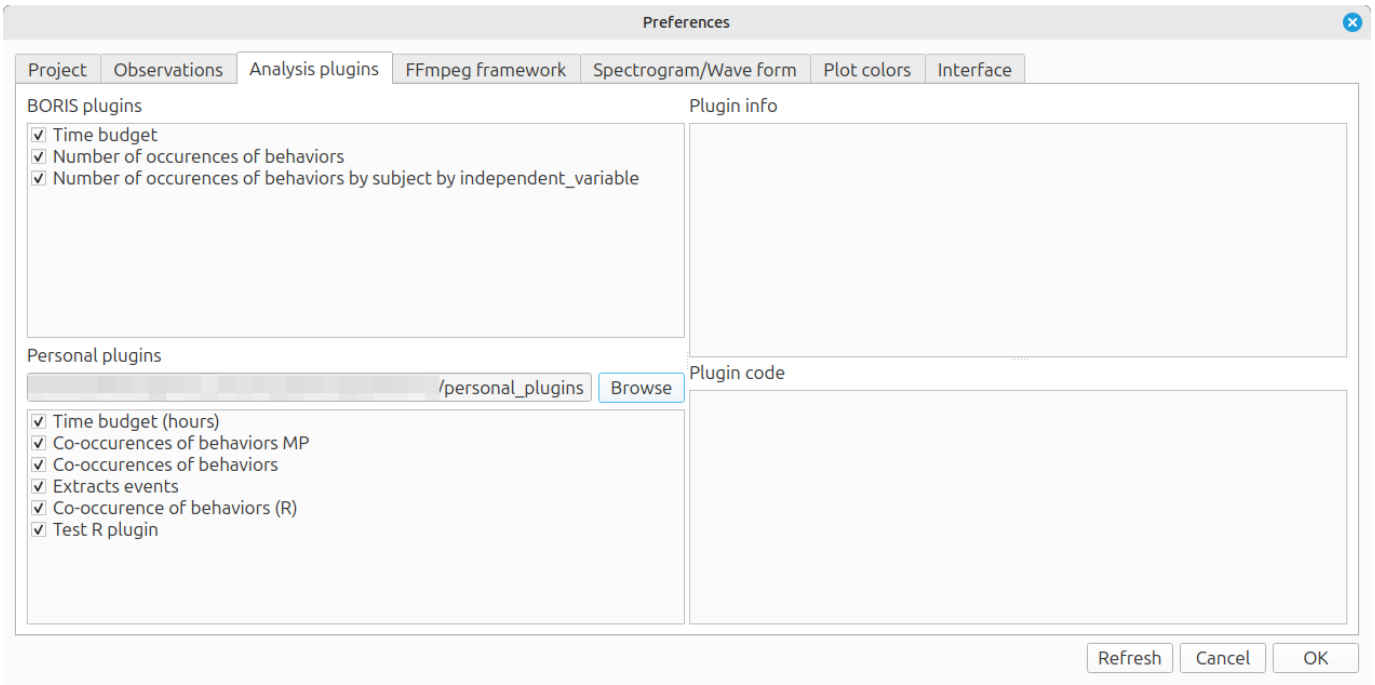
2.11.10 Plugins

Starting from version 9, you can create plugins to analyze coded data. BORIS supports two programming languages for writing plugins: **Python** and **R**.

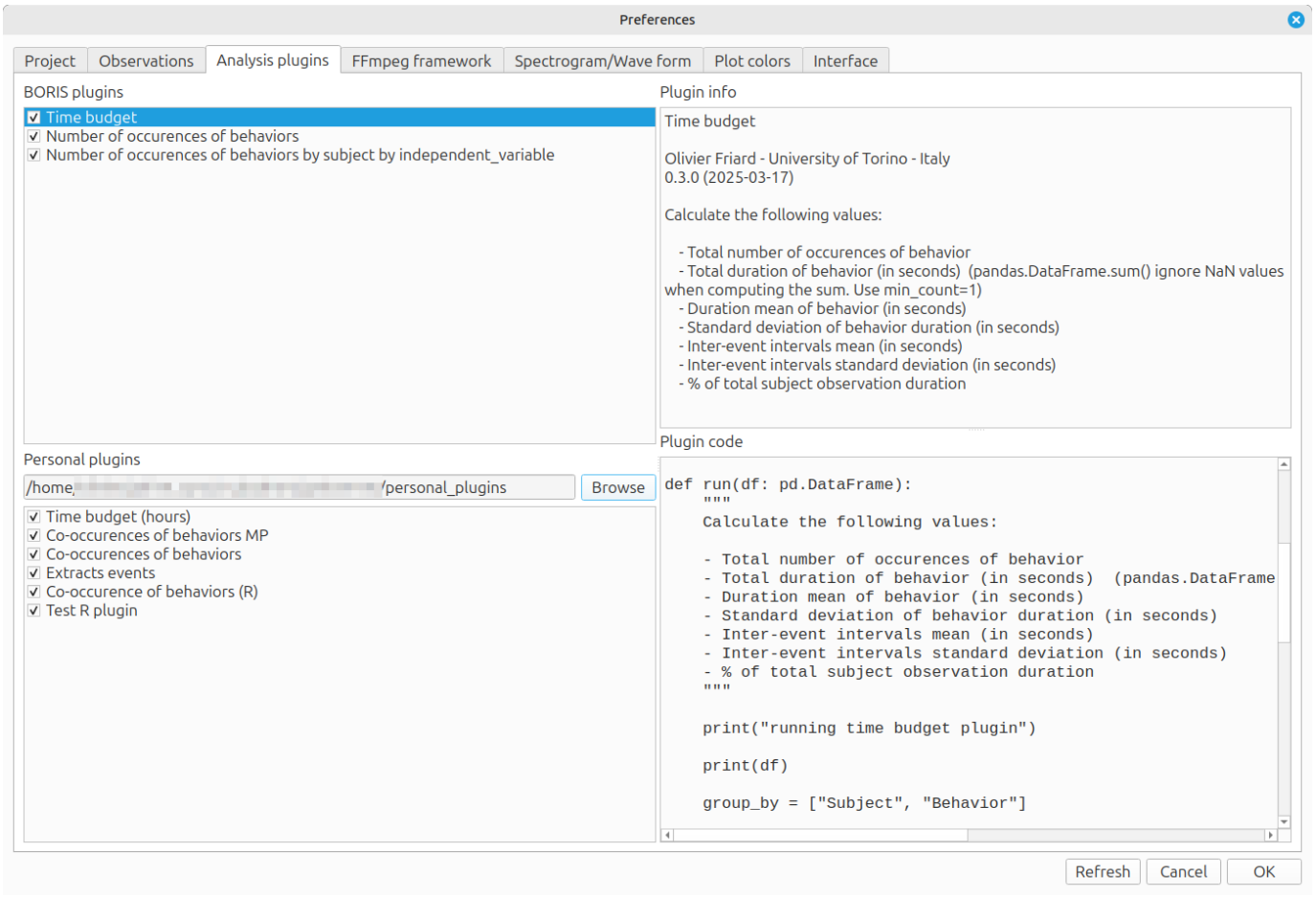
> **Important**
>
> Use this function for testing purposes, as it is currently experimental and may be subject to changes in the future.

Some plugins are built into BORIS (see the **BORIS plugins** list), and you can also create your own custom plugin using Python and Pandas or R.

Go to **Preferences** > **Analysis plugins**, then choose the plugins you'd like to enable.



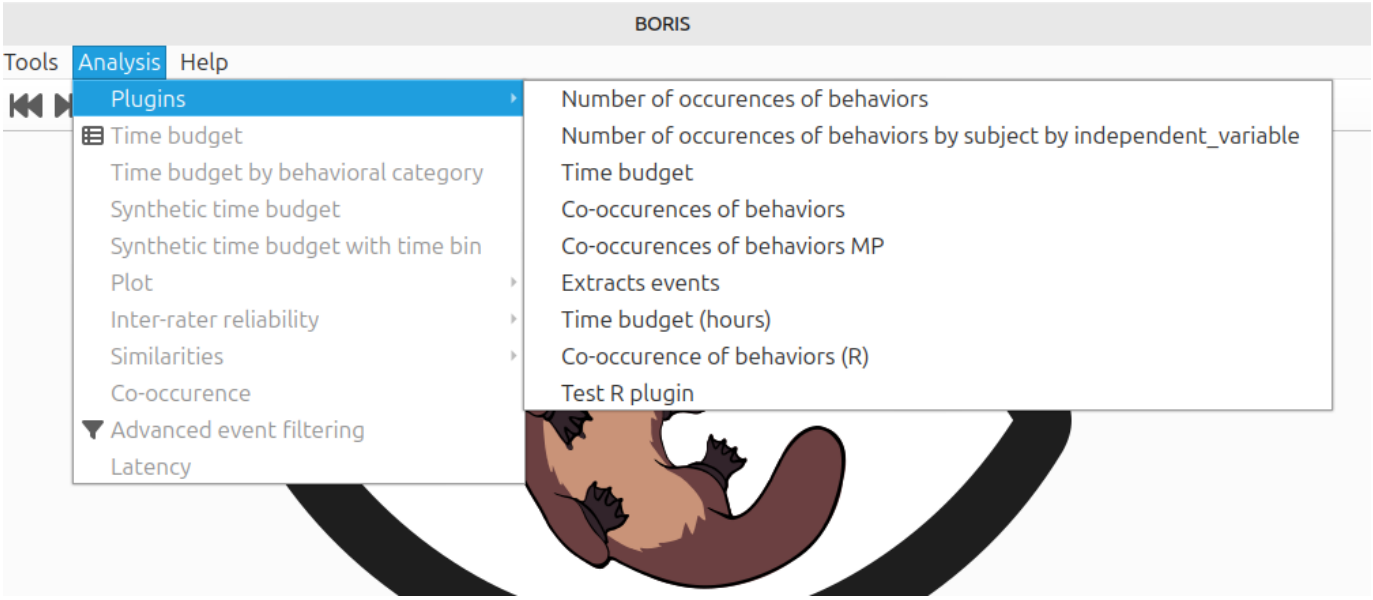
You can view more information by clicking on the plugin name.



You can find the plugin code in the boris/analysis_plugins directory.

How to use an Analysis plugin

Go to **Analysis** > **Plugins**



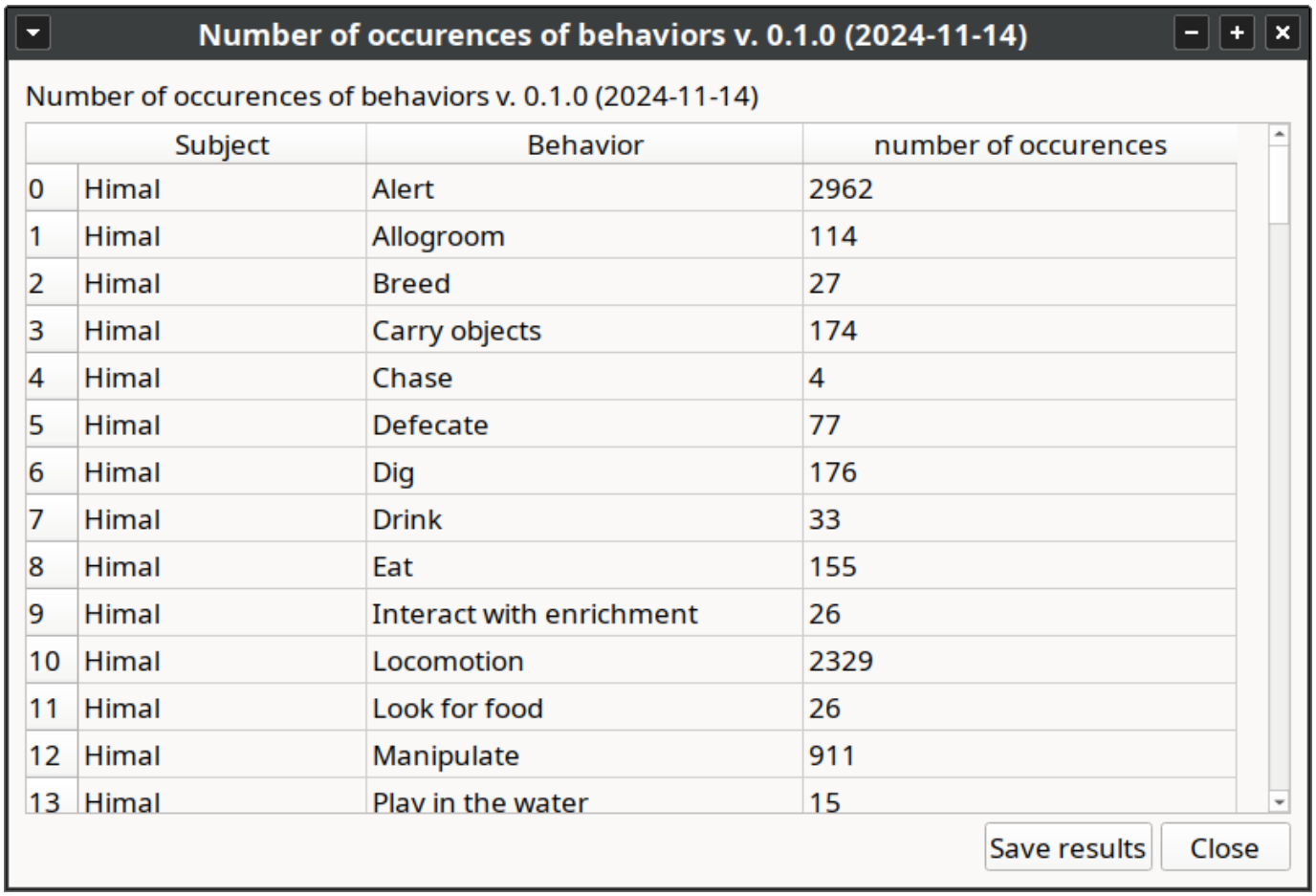
All plugins are listed (the BORIS plugins and your personal plugins)

Select the plugin you want to use

Select the observations to analyze

Select the subjects and the behaviors

You should obtain a window with the results of the plugin analysis



	Subject	Behavior	number of occurences
0	Himal	Alert	2962
1	Himal	Allogroom	114
2	Himal	Breed	27
3	Himal	Carry objects	174
4	Himal	Chase	4
5	Himal	Defecate	77
6	Himal	Dig	176
7	Himal	Drink	33
8	Himal	Eat	155
9	Himal	Interact with enrichment	26
10	Himal	Locomotion	2329
11	Himal	Look for food	26
12	Himal	Manipulate	911
13	Himal	Play in the water	15

The results can be saved in various formats (TSV, CSV, ODS, XLSX, Pandas dataframe, R dataframe and HTML)

Anatomy of an Analysis plugin

The DataFrame passed to the plugin includes the following columns:

```
Column                                          Dtype
 ------                                          -----
Observation id                                  object
independent variables                           object
...
Subject                                         object
Observation duration by subject by observation  int64
Behavior                                        object
Behavioral category                             object
Behavior modifiers                              object
...
Behavior type                                   object
Start (s)                                       object
Stop (s)                                        object
Duration (s)                                    object
Comment start                                   object
Comment stop                                    object
```

The DataFrame will include a column for each independent variable defined in your project.

The DataFrame will include a column for each behavior modifier set defined in your project.

Here is an example of the DataFrame structure, including 4 independent variables and various behavior modifiers:

```
Data columns (total 27 columns):
#   Column                                      Dtype
---  ------                                      -----
0   Observation id                              object
1   independent variable 'Location'             object
2   independent variable 'Weather'              object
3   independent variable 'Temperature'          object
4   independent variable 'Number of visitors'   object
```

```
5   Subject                                          object
6   Observation duration by subject by observation   int64
7   Behavior                                         object
8   Behavioral category                              object
9   (Carry objects, set #1)                          object
10  (Chase, set #1)                                  float64
11  (Eat, set #1)                                    float64
12  (Eat, set #2)                                    float64
13  (Interact with enrichment, set #1)               float64
14  (Locomotion, set #1)                             object
15  (Play in the water, interaction)                 object
16  (Play on the ground, set #1)                     object
17  (Rub, set #1)                                    object
18  (Sniff, set #1)                                  object
19  (Spot keeper, set #1)                            float64
20  (Tear, set #1)                                   object
21  Behavior type                                    object
22  Start (s)                                        object
23  Stop (s)                                         object
24  Duration (s)                                     object
25  Comment start                                    object
26  Comment stop                                     object
```

PYTHON PLUGIN

A BORIS Python plugin is a Python script consisting of one function **run**.

The **run** function takes a Pandas DataFrame as its sole argument and must return:

- a **Pandas DataFrame**.
- a **string** (str)
- a **Pandas DataFrame** and a **string** (str) as a tuple.

For example:

```
def run(df: pd.DataFrame) -> pd.DataFrame:
   ...
   ...
   return elaborated_df
```

or:

```
def run(df: pd.DataFrame) -> str:
   ...
   ...
   return results
```

or:

```
def run(df: pd.DataFrame) -> tuple[pd.DataFrame, str]:
   ...
   ...
   return (elaborated_df, results)
```

The plugin code must define the following global variables:

```
__plugin_name__ = "PLUGIN NAME"
__version__ = "x.y.z"
__version_date__ = "YYYY-MM-DD"
__author__ = "AUTHOR - INSTITUTION"
```

Example of Python plugin for determining the number of occurences of the selected behaviors for each selected subjects:

```
"""
BORIS plugin

number of occurences of behaviors
"""

import pandas as pd

__version__ = "0.3.0"
__version_date__ = "2025-03-17"
__plugin_name__ = "Number of occurences of behaviors"
__author__ = "Olivier Friard - University of Torino - Italy"


def run(df: pd.DataFrame) -> pd.DataFrame:
    """
    Calculate the number of occurrences of behaviors by subject.
```

```
    """

    df_results: pd.DataFrame = df.groupby(["Subject", "Behavior"])["Behavior"].count().reset_index(name="number of occurences")

    return df_results
```

Visualize the plugin code on GitHub: number_of_occurences.py

You can modify the **run** function to implement your custom logic.

R PLUGIN

The plugin written with R must contain a **run** function. The **run** function takes a dataframe as its sole argument and must return a dataframe:

```
run <- function(df) {...
   ...
   ...
   return(elaborated_df)
}
```

The plugin code must define the following global variables:

```
plugin_name <- 'PLUGIN NAME'
version <- 'x.y.z'
version_date <- 'YYYY-MM-DD'
author  <- 'AUTHOR - INSTITUTION'
description <- "..."
```

Example of a R plugin for determining the number of co-occurences of behaviors:

```
# plugin.R
# converted from the co-occurences.py BORIS plugin by ChatGPT


plugin_name <- 'Co-occurence of behaviors (R)'
version <- '0.0.1'
version_date <- '2025-06-12'
author  <- 'Olivier Friard - University of Torino - Italy'
description <- "Co-occurence of behaviors (R)"


run <- function(df) {
# Number of combinations
total_combinations <- choose(nrow(df), 2)
cat("total_combinations =", total_combinations, "\n")

cooccurrences <- list()

# All combinations
comb_indices <- combn(nrow(df), 2)

for (k in 1:ncol(comb_indices)) {
    i <- comb_indices[1, k]
    j <- comb_indices[2, k]

    row1 <- df[i, ]
    row2 <- df[j, ]

    # skip if behaviors are the same
    if (row1$Behavior == row2$Behavior) next

    # Check overlap
    if (!(row1[["Stop (s)"]] <= row2[["Start (s)"]] || row2[["Stop (s)"]] <= row1[["Start (s)"]])) {
    pair <- sort(c(as.character(row1$Behavior), as.character(row2$Behavior)))
    key <- paste(pair, collapse = " | ")

    if (key %in% names(cooccurrences)) {
        cooccurrences[[key]] <- cooccurrences[[key]] + 1
    } else {
        cooccurrences[[key]] <- 1
    }
    }
}

# Convert list to dataframe
if (length(cooccurrences) > 0) {
    cooc_df <- data.frame(
    Pair = names(cooccurrences),
    Count = unlist(cooccurrences),
    row.names = NULL
    )
} else {
    cooc_df <- data.frame()
}
```

```
return(cooc_df)
}
```

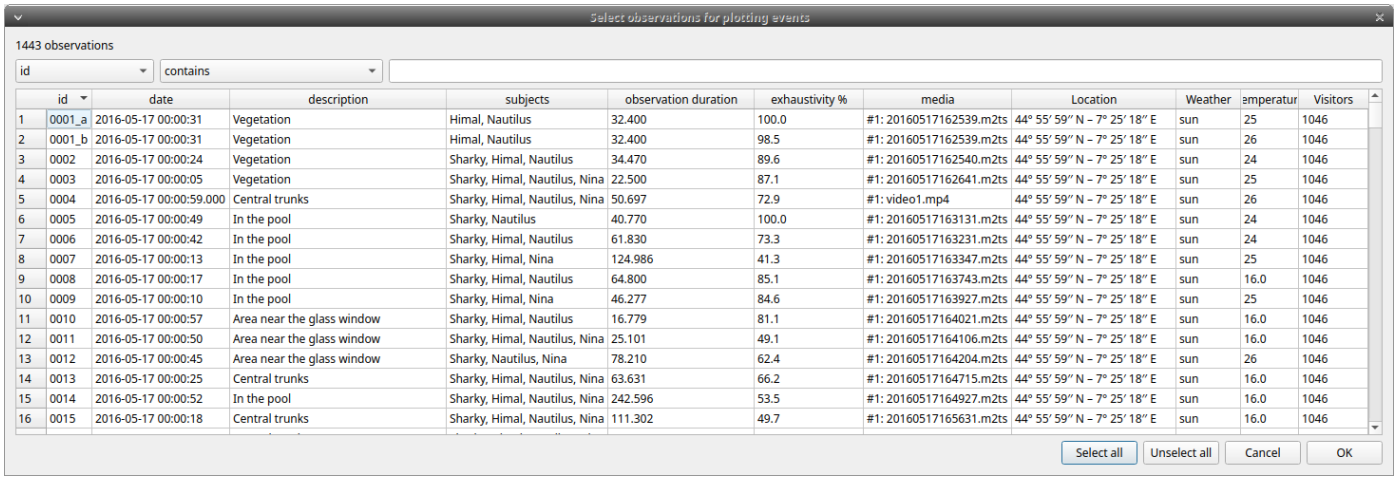
2.12 Plot

2.12.1 Plot events

The recorded events can be plotted along a time axis.

Analysis > **Plot** > **Plot events**

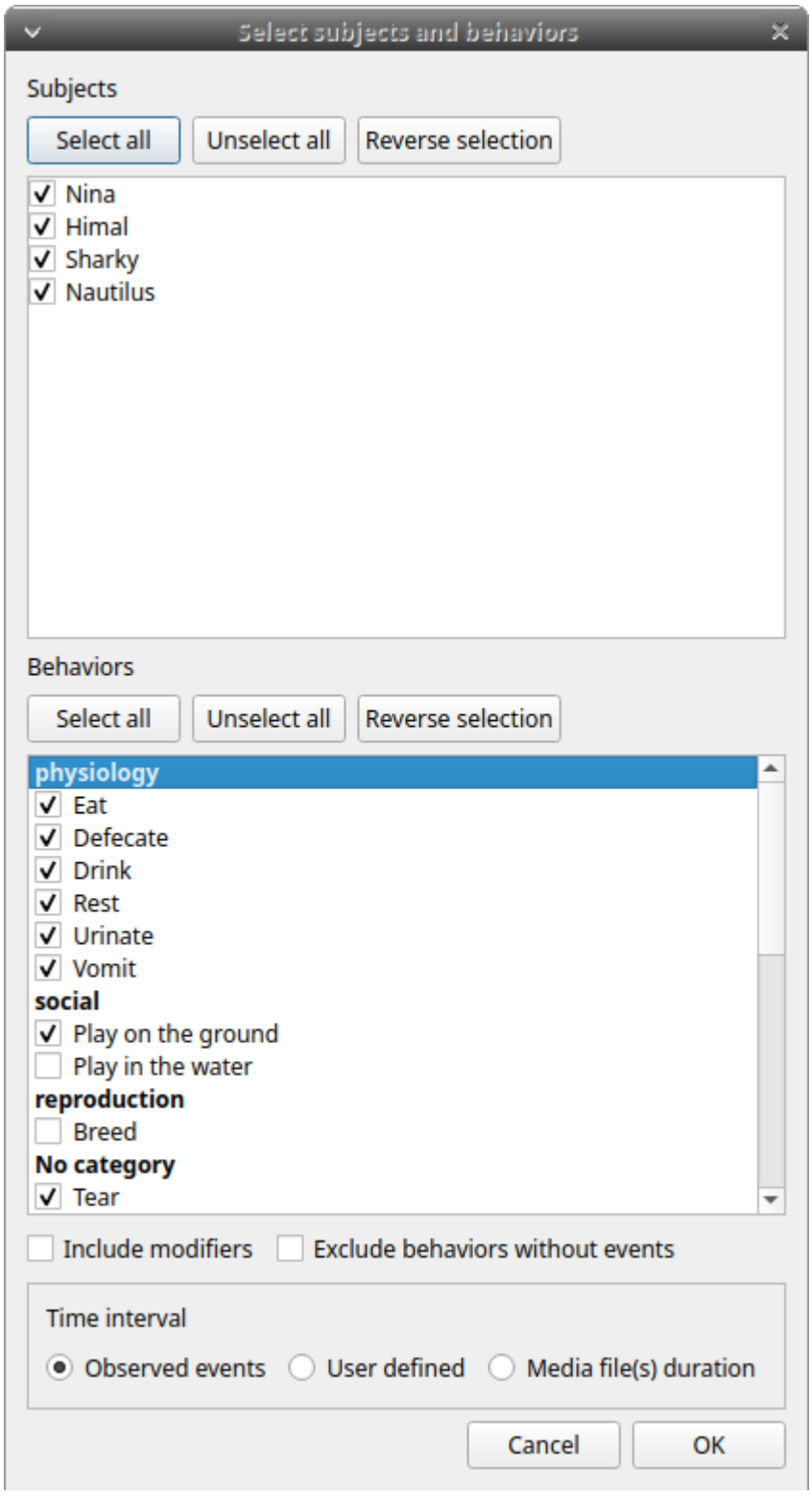
Select the **observations** you want to plot. If more than one observation are selected BORIS will ask you for a directory where to save the plots.



	id	date	description	subjects	observation duration	exhaustivity %	media	Location	Weather	emperatur	Visitors
1	0001_a	2016-05-17 00:00:31	Vegetation	Himal, Nautilus	32.400	100.0	#1: 20160517162539.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25	1046
2	0001_b	2016-05-17 00:00:31	Vegetation	Himal, Nautilus	32.400	98.5	#1: 20160517162539.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
3	0002	2016-05-17 00:00:24	Vegetation	Sharky, Himal, Nautilus	34.470	89.6	#1: 20160517162540.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
4	0003	2016-05-17 00:00:05	Vegetation	Sharky, Himal, Nautilus, Nina	22.500	87.1	#1: 20160517162641.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25	1046
5	0004	2016-05-17 00:00:59.000	Central trunks	Sharky, Himal, Nautilus, Nina	50.697	72.9	#1: video1.mp4	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
6	0005	2016-05-17 00:00:49	In the pool	Sharky, Nautilus	40.770	100.0	#1: 20160517163131.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
7	0006	2016-05-17 00:00:42	In the pool	Sharky, Himal, Nautilus	61.830	73.3	#1: 20160517163231.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	24	1046
8	0007	2016-05-17 00:00:13	In the pool	Sharky, Himal, Nina	124.986	41.3	#1: 20160517163347.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25	1046
9	0008	2016-05-17 00:00:17	In the pool	Sharky, Himal, Nautilus	64.800	85.1	#1: 20160517163743.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046
10	0009	2016-05-17 00:00:10	In the pool	Sharky, Himal, Nina	46.277	84.6	#1: 20160517163927.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	25	1046
11	0010	2016-05-17 00:00:57	Area near the glass window	Sharky, Himal, Nautilus	16.779	81.1	#1: 20160517164021.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046
12	0011	2016-05-17 00:00:50	Area near the glass window	Sharky, Himal, Nautilus, Nina	25.101	49.1	#1: 20160517164106.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046
13	0012	2016-05-17 00:00:45	Area near the glass window	Sharky, Nautilus, Nina	78.210	62.4	#1: 20160517164204.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	26	1046
14	0013	2016-05-17 00:00:25	Central trunks	Sharky, Himal, Nautilus, Nina	63.631	66.2	#1: 20160517164715.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046
15	0014	2016-05-17 00:00:52	In the pool	Sharky, Himal, Nautilus, Nina	242.596	53.5	#1: 20160517164927.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046
16	0015	2016-05-17 00:00:18	Central trunks	Sharky, Himal, Nautilus, Nina	111.302	49.7	#1: 20160517165631.m2ts	44° 55′ 59″ N – 7° 25′ 18″ E	sun	16.0	1046

Select the observations to plot

The **subjects** and **behaviors** you want to include in the plot can be selected in the following window:



You can choose to include or not the behavior modifiers (if any) and to exclude behaviors without coded events.

The time interval can be selected (See time budget)

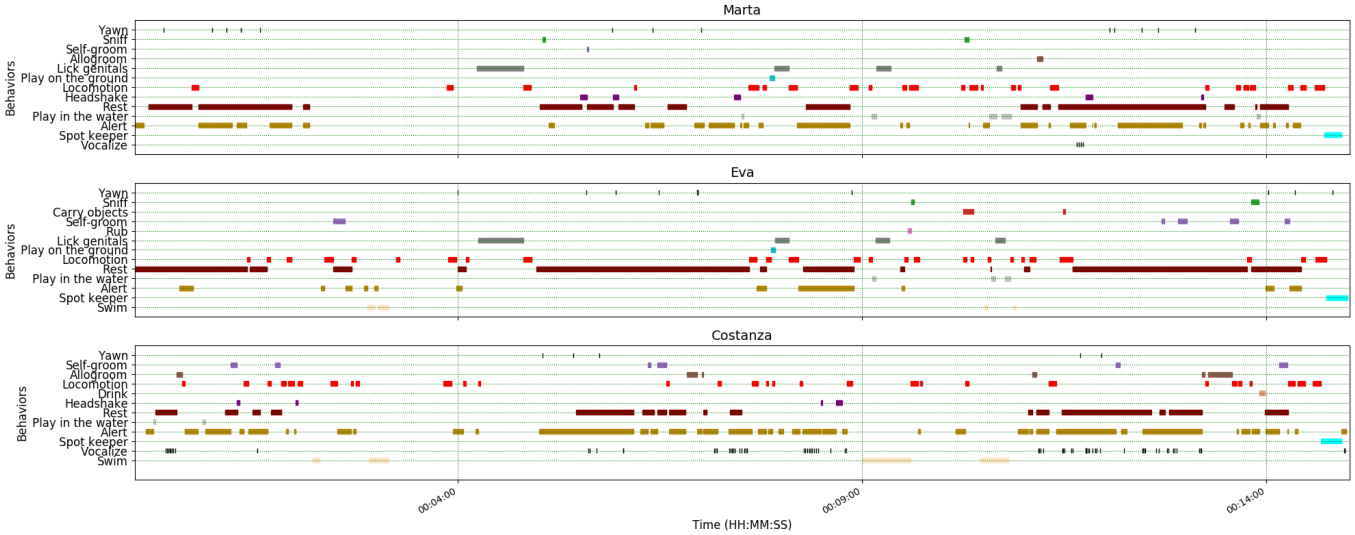
The plot can be exported in various formats like bitmap (PNG, JPG, TIFF) or vectorial graphic (SVG, PDF, EPS, PS). The SVG format can be further edited with the Inkscape vector graphics editor.

Important

If a STATE behavior has an odd number of coded events, you will see this error message: "The STATE behavior XXX is not paired"

This function creates one plot by subject on one figure.

The color of behaviors can be customized. See plot colors

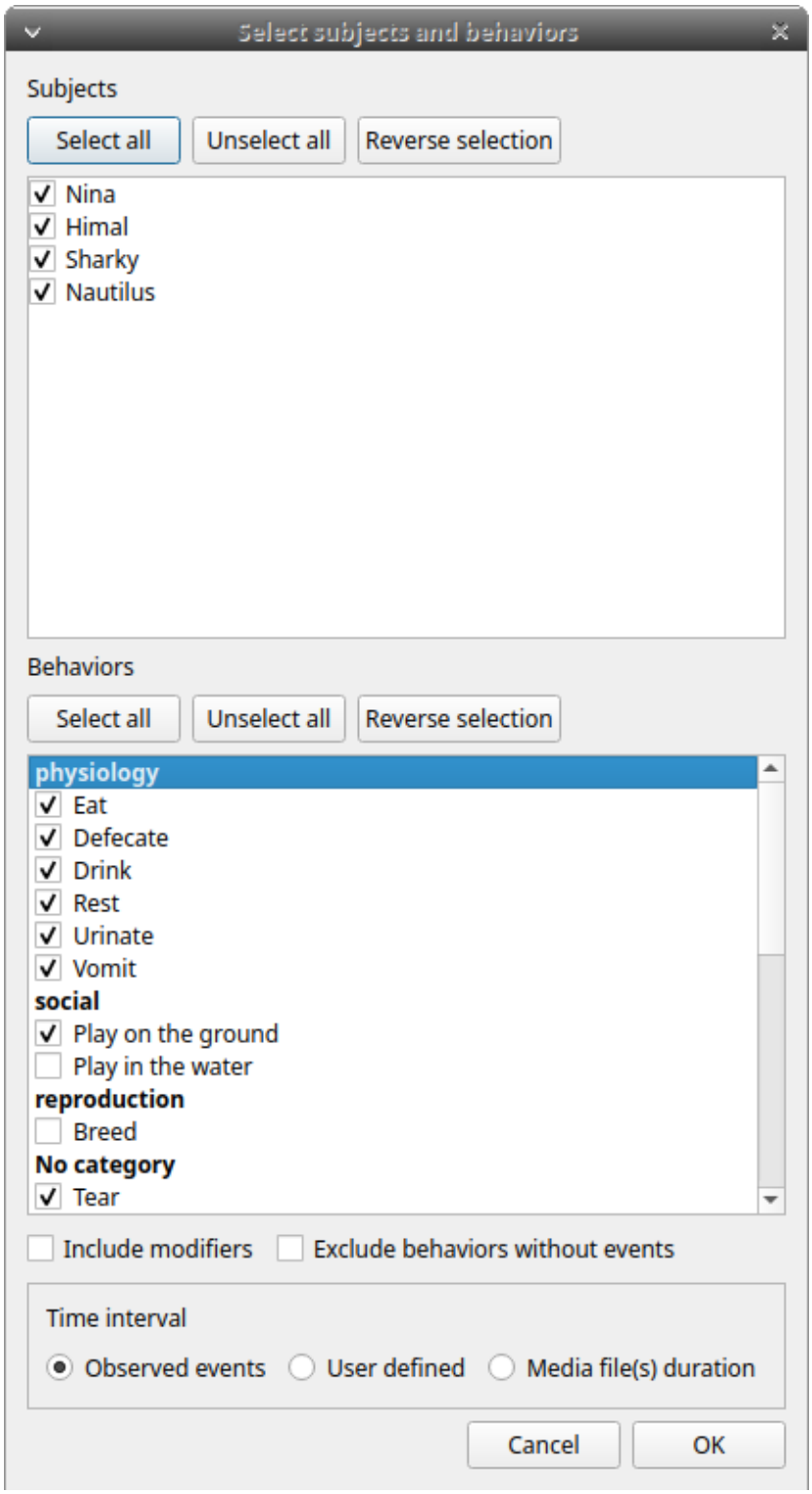


2.12.2 Plot time budget

The duration and number of occurences can be plotted for each subject and behavior.

Analysis > **Plot** > **Plot time budget**

The subjects and behaviors you want to include in the plot can be selected in the following window:



The behavior modifiers can not be included in the plot for now.

The time interval can be selected (See time budget)

The plot can be exported in various formats like bitmap (PNG, JPG, TIFF) or vectorial graphic (SVG, PDF, EPS, PS). The SVG format can be further edited with the Inkscape vector graphics editor.

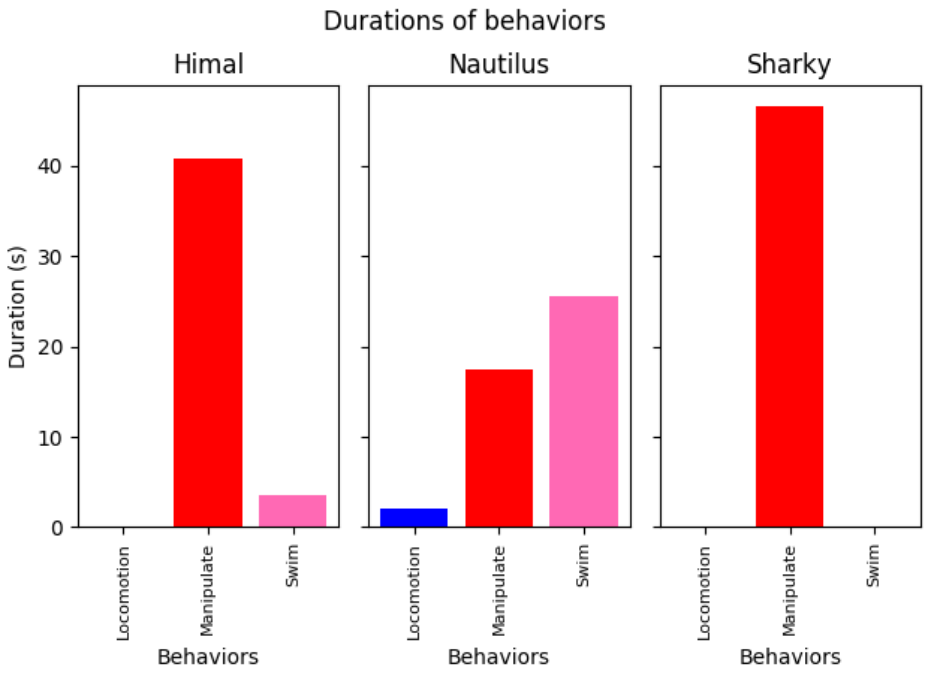
Important

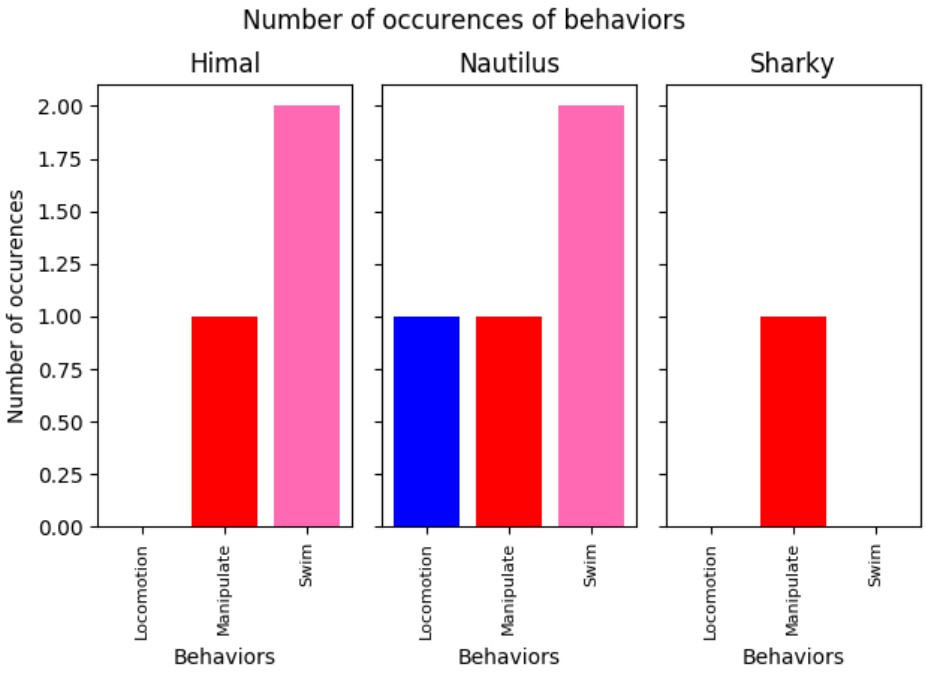
If a STATE behavior has an odd number of coded events, you will see this error message: "The STATE behavior XXX is not paired"

This function creates 2 plots with all subjects for each observation:

- a plot of the behavior durations for the behaviors defined as STATE event.
- a plot of the number of occurences for all the behaviors.

The color of behaviors can be customized. See plot colors

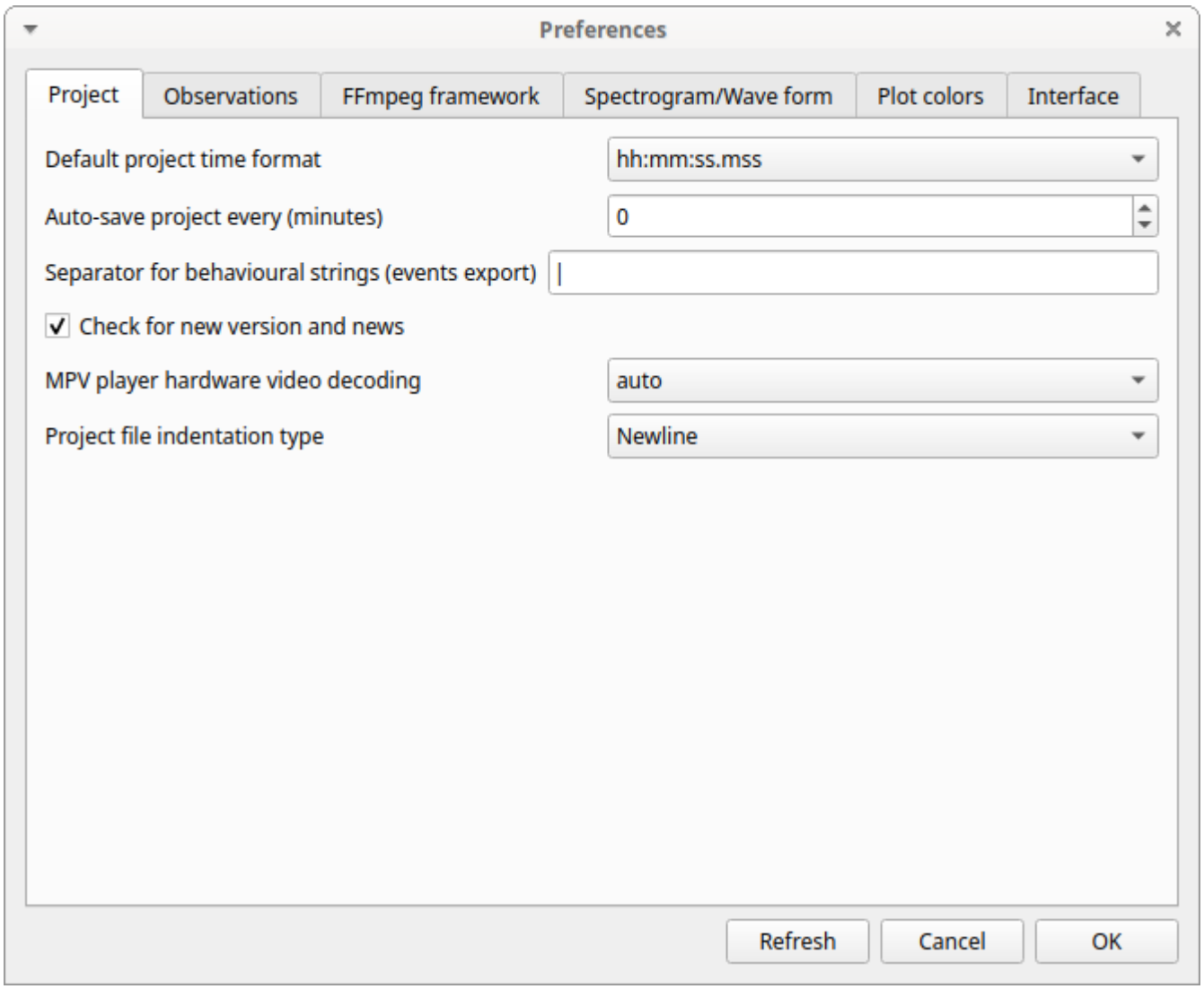




2.13 Preferences

You can customize BORIS using the Preferences window (**File** > **Preferences**)

2.13.1 Project preferences



Default project time format

This option allows the user to choose the format for displaying time in the project. Please note that time is internally always saved in seconds with a precision of 3 decimal digits

Auto-save project every (minutes)

if set BORIS will save your project automatically every n minutes. 0 indicate no automatic backup. The project will be saved if the project is already saved and an observation is open.

Separator for behavioural strings

Character (or string) used to separate behaviors when exporting events as behavioural strings. See also Behatrix

Check for new version

Check for new version on BORIS web site every 15 days (internet access required)

MPV hardware video decoding

if you experiment some problems with the mpv embedded player try to change this value

Project file indentation type

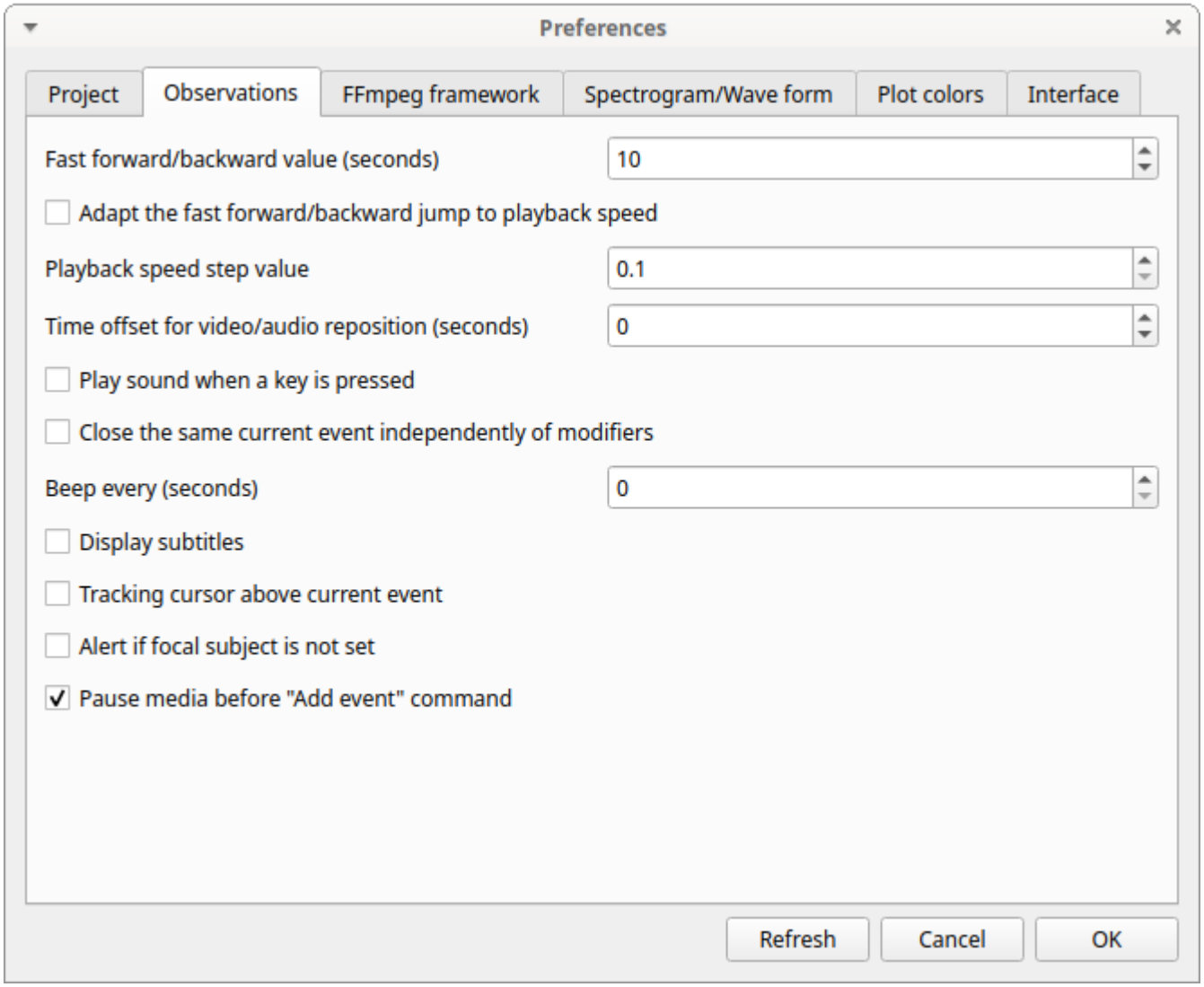
The BORIS project file is encoded in JSON format. Choose the indentation style for the project file between:

- None
- Newline
- Tab
- 2 spaces
- 4 spaces

Refresh button

Option to reinitialize the configuration to default. BORIS will be closed.

2.13.2 Observations



Fast forward/backward value (seconds)

This option allows the user to customize the amount of time for "jumping" forward or backward in media.

Adapt the fast for/backward jump to playback speed

The jump value will be adapted to the playback speed.

Playback speed step value

This value indicate how much the speed will be increased or decreased after pressing the *change playback speed* buttons.

Time offset for media reposition (seconds)

This value indicates the time offset for repositioning the media after double-click on a row event of the *Events* table. 'for example -4 seconds indicates that after a double-click the media will be repositioned 4 seconds before the recorded event.'

Play sound when a key is pressed

Activate a sound signal after every keypress event

Close the same current event independently of modifiers

Option used to STOP the current behavior without regarding the modifiers

Display subtitles

Option to display or hide the visualization of subtitles. In case of separate file, the file containing subtitles must have the same base name than the video files with a .srt extension.

Tracking cursor above current event

Check this box to position the tracking cursor above the current event in events list table.

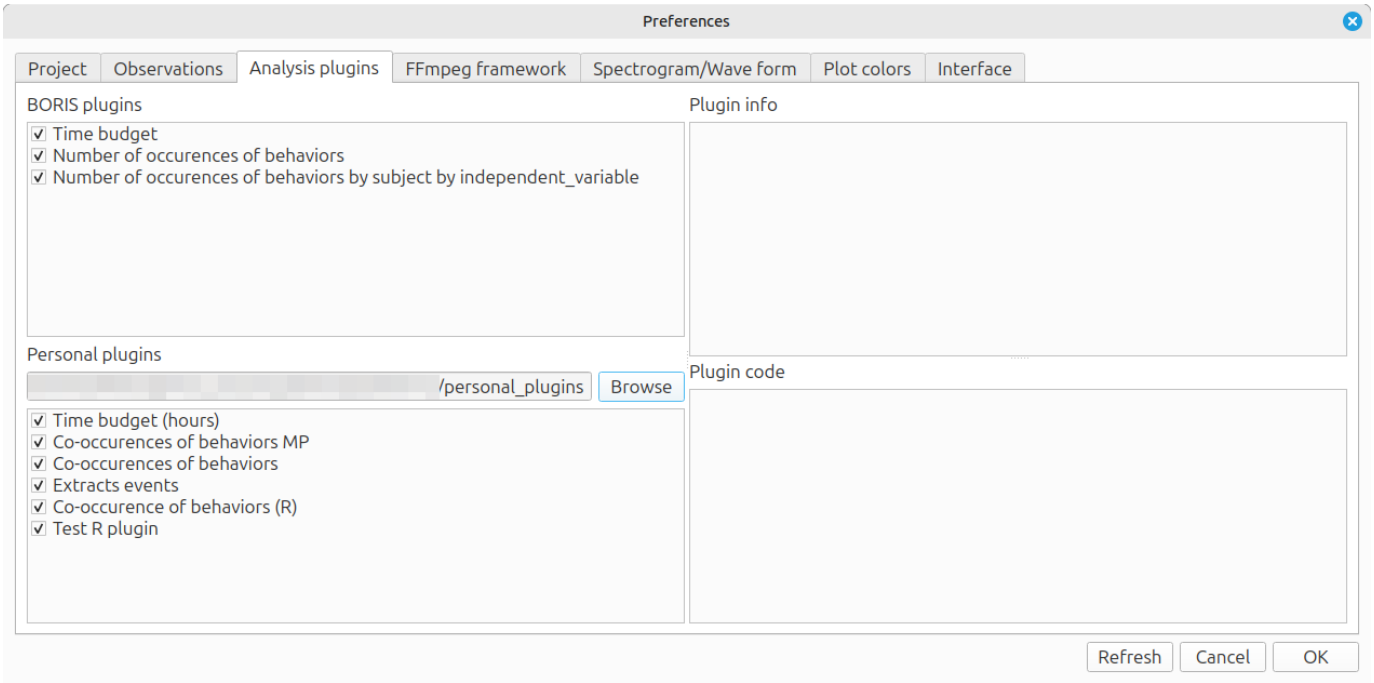
Alert if focal subject is not set

If this option is activated BORIS will show an alert box if no focal subject is selected

Pause media before "Add event" command

Option to pause the media before manually adding an event.

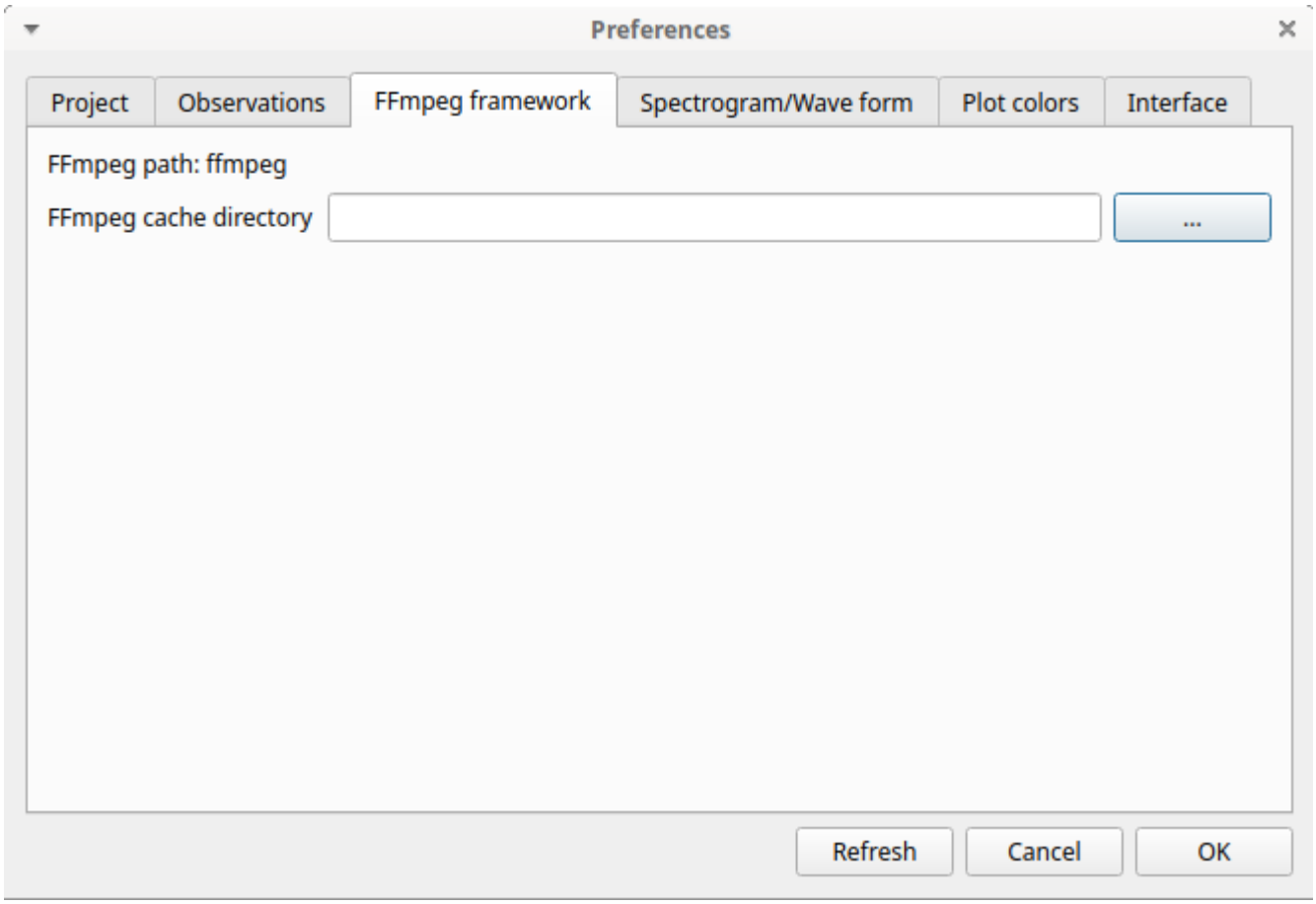
2.13.3 Analysis plugins



Personal plugins directory

Select the directory that contains the plugins to be loaded.

2.13.4 FFmpeg framework

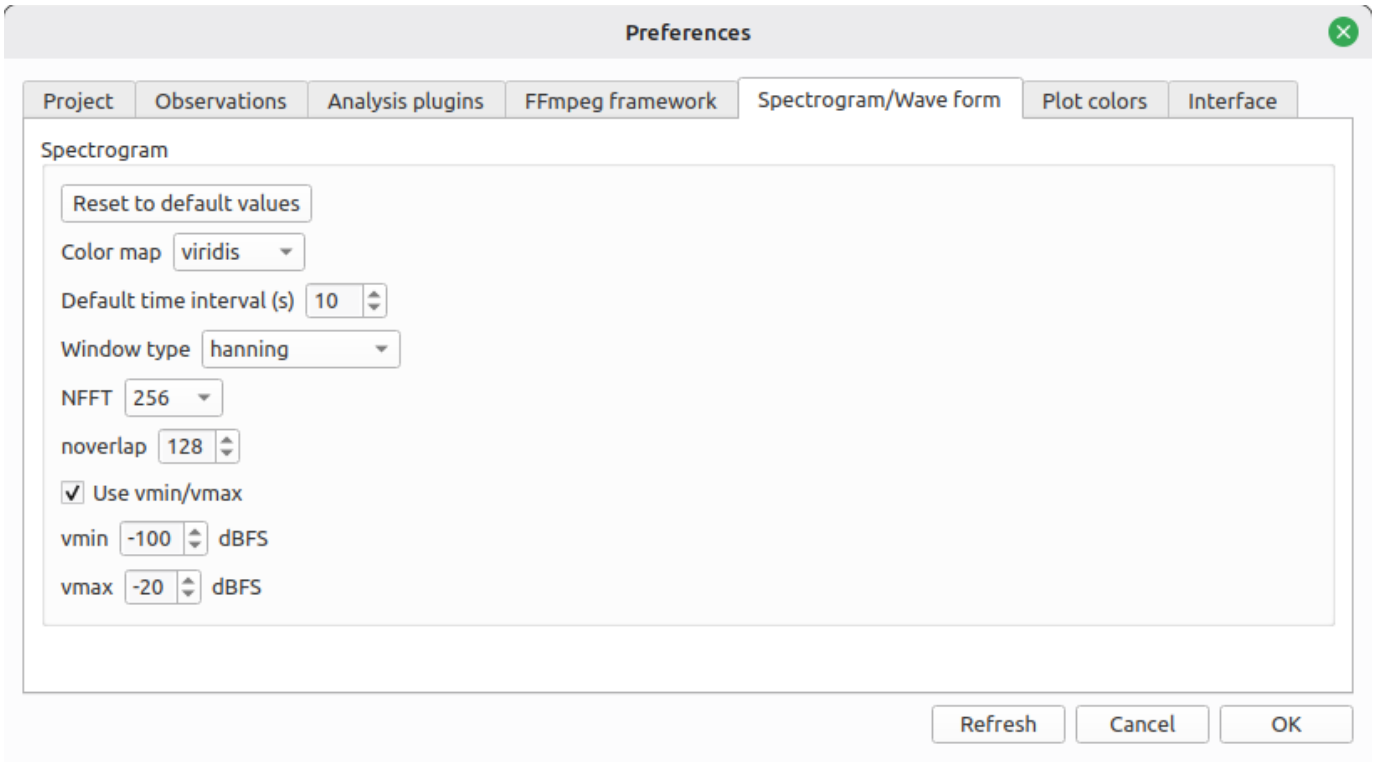


The path for the ffmpeg executable program is displayed. The FFmpeg executable is included with BORIS for Windows. The FFmpeg framework is required to run BORIS.

FFmpeg cache directory

This indicates the directory that will be used as image cache for frame-by-frame mode and spectrogram visualization. If you do not specify a path, BORIS will use the default temporary directory of your system.

2.13.5 Spectrogram / wave form



Color map

Select the color map for displaying the generated spectrogram. See Matplotlib colormaps for details.

Default time interval

Select the time interval (in seconds) for displaying the spectrogram and waveform

Window type

Select the window tyope between Hanning, Hamming and Blackmanharris

NFFT size

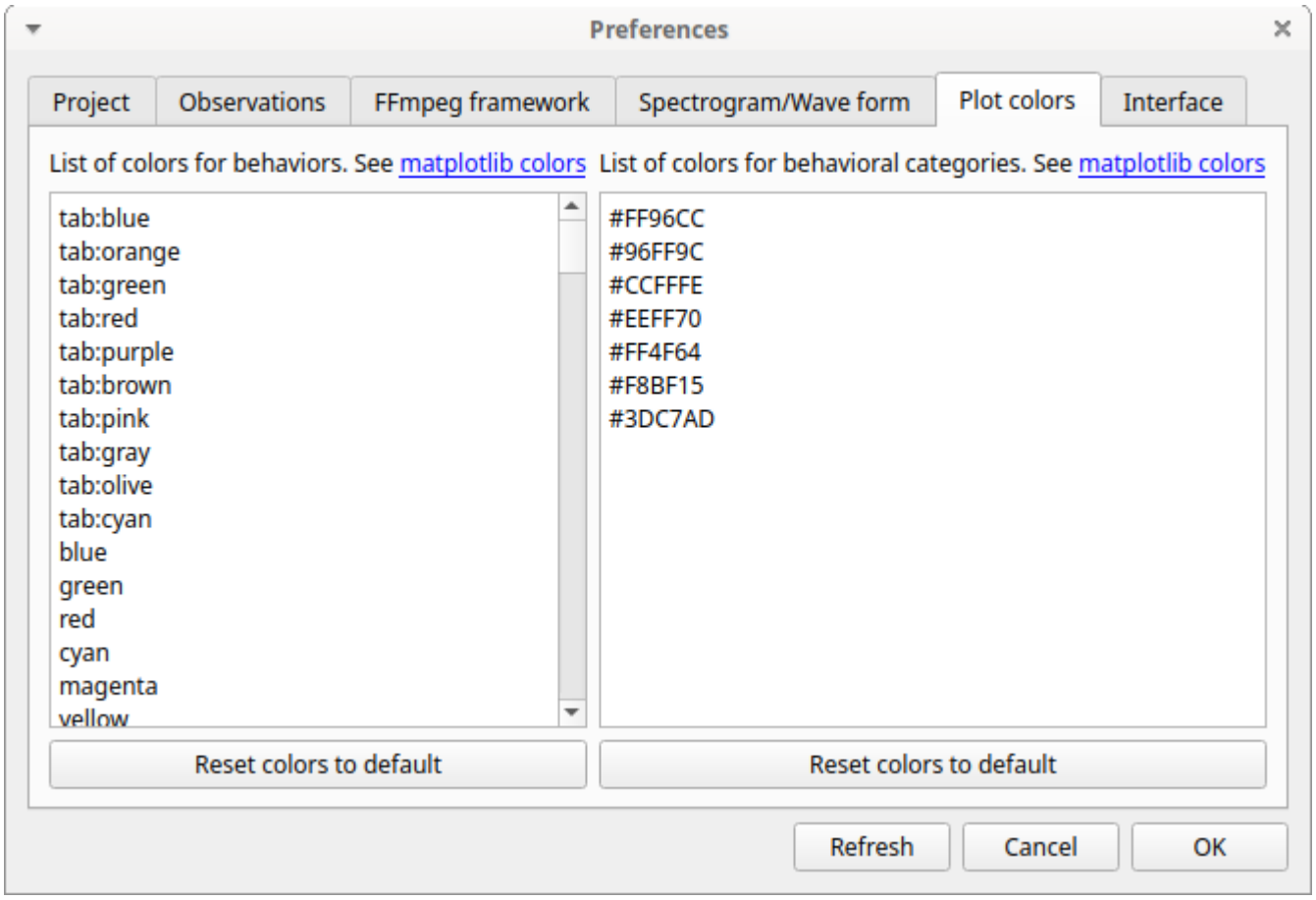
noverlap

Use vmin/vmax

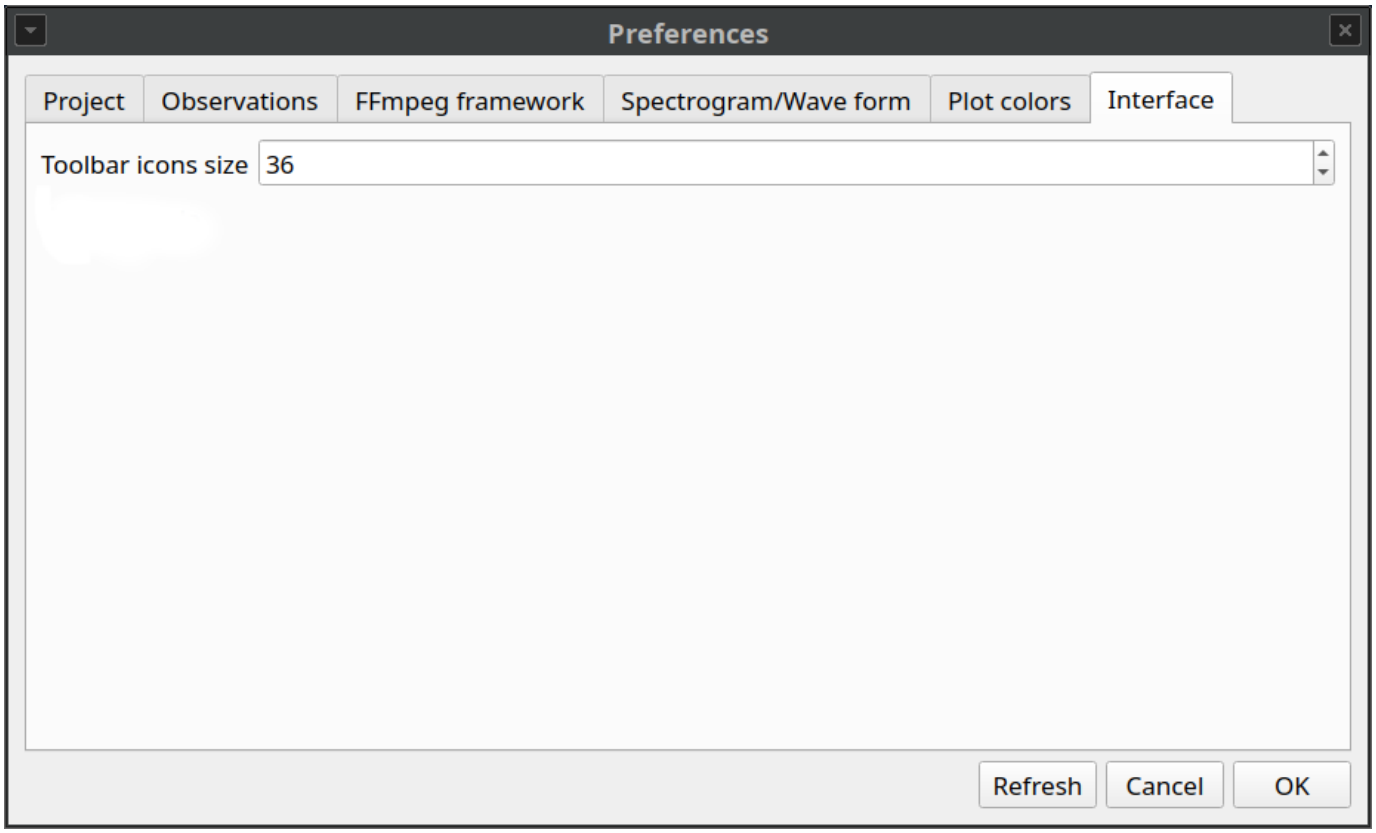
2.13.6 Plot colors

The color of behaviors in the plot events functions can be customized. The first color will be associated to the first behavior in your ethogram, the second color to the second behavior and so on. Various color formats can be used to specify a color: **named color** or **hex RGB** (like #0F0F0F). See https://matplotlib.org/api/colors_api.html and https://matplotlib.org/examples/color/named_colors.html for details

The **reset colors to default** button will reload the default colors.



2.13.7 Interface



Toolbar icons size

Set the size of the icons in the toolbar (in pixels)

2.14 Various

2.14.1 Removing path of media files

Using BORIS you can choose to store the full path of the media/data files into the file project (for example: `/home/user/Video/video_n1.mp4` or `c:\Users\user\Documents\video1.avi`).

If you want to move your project on a different computer or if you want to move your media/data files you may want to do not store the full path. For this you can choose to add media/data files with relative path (See **Add media files** section). You can also remove the full path of your media/data files from all observations of the current project (**File** > **Remove path from media files**. Please note that this operation is irreversible. After remotion the full path of your media will be lost and will not be recoverable.

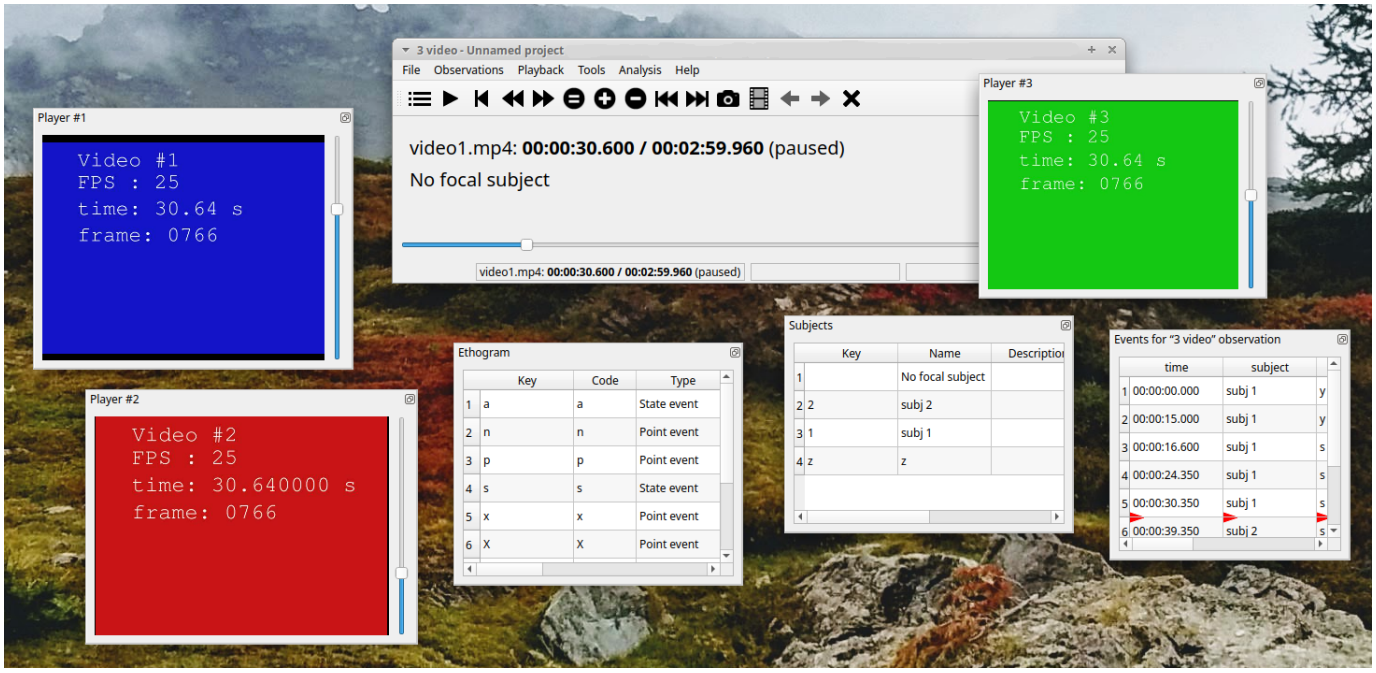
If you choose to do not store the full path of media/data files the path of the media/data files must contain the path of your BORIS project file.

Example: if you BORIS project file is saved in `/home/user/projects/test.project` your media/data files can be saved in the `/home/user/projects/videos` directory but **NOT** in the [/home/user/videos]{.title-ref}` directory.

2.14.2 Docking / undocking graphical elements

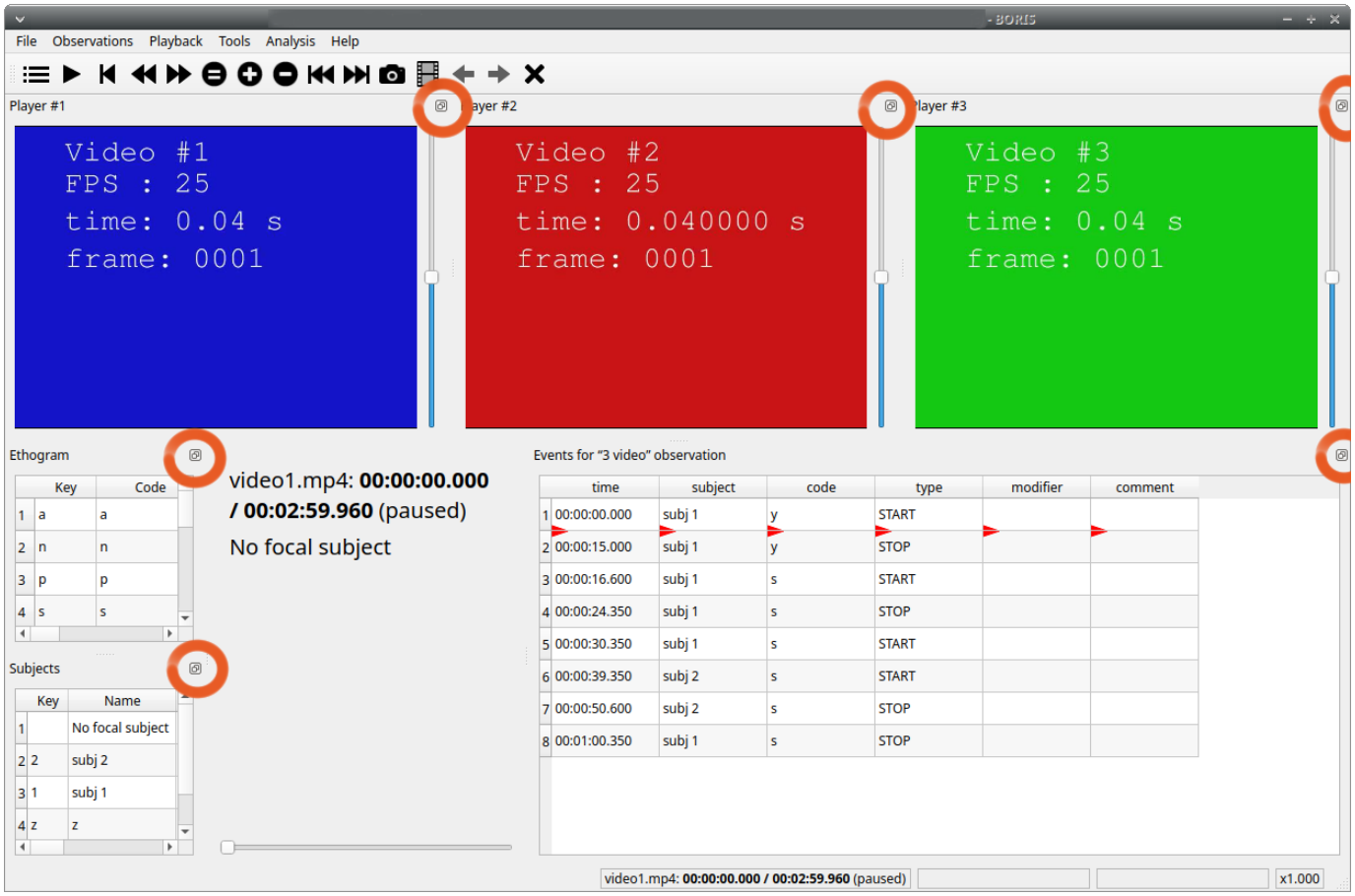
All elements, including all the media players can be undocked from the main window and positioned where you prefer (e.g. they can be on the same desktop over one or many screens).

The position of the various widgets is saved in the configuration file at the end of the work session.

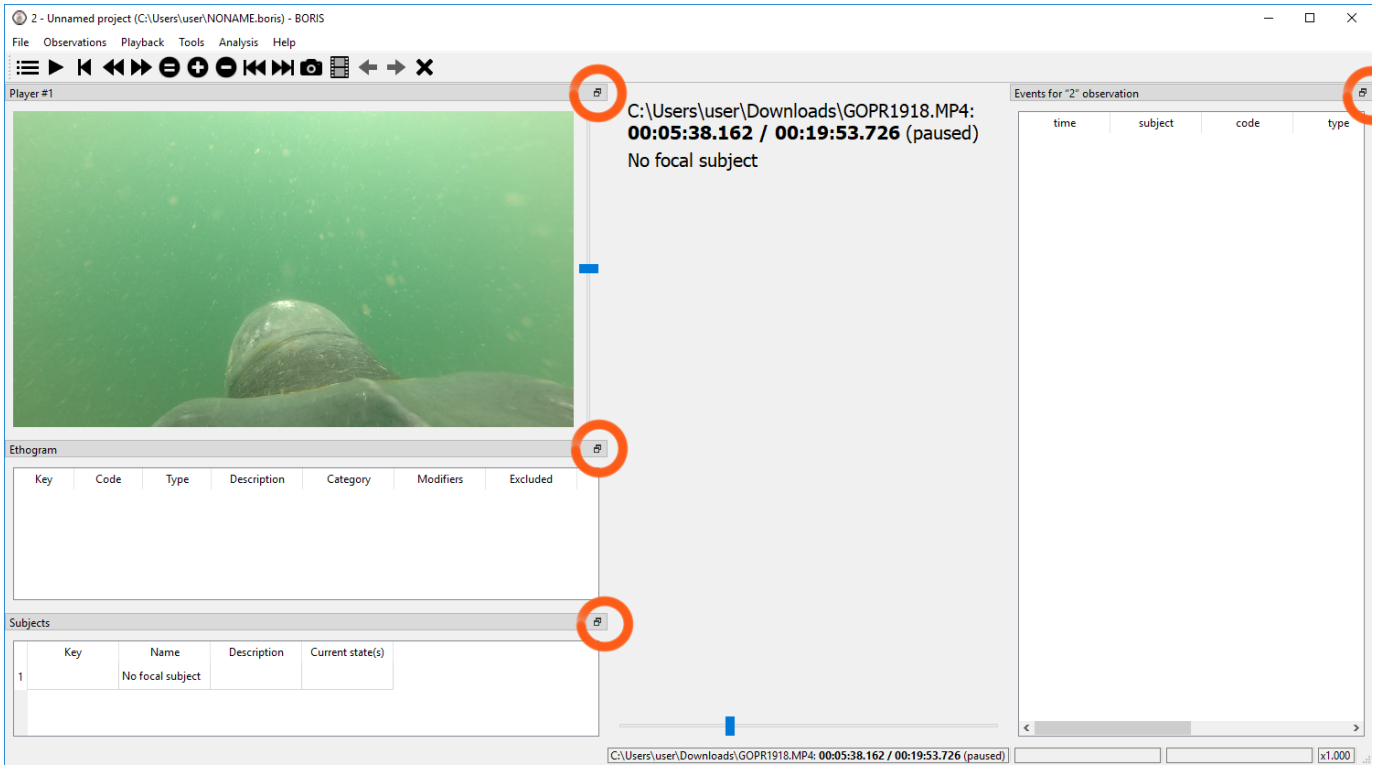


Click the icon present at the top-right corner of the widget (for MacOS the icon is located at the left-top corner) will undock the widgets that can be repositioned on another docking area or moved out of the main window. A double-click on the top bar of the widget will reposition it on the main window.

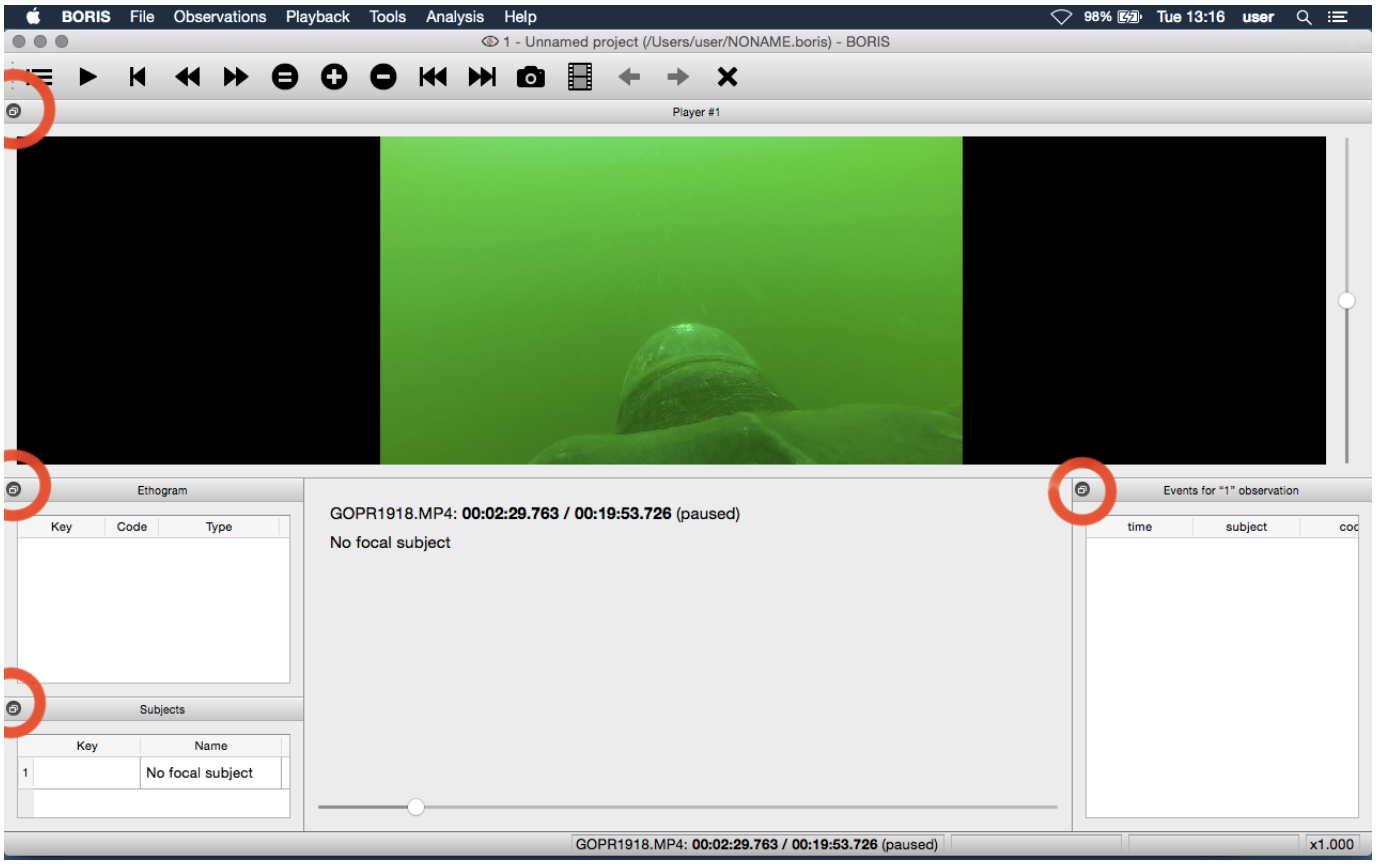
For Linux:



For Microsoft-Windows:



For MacOS:



If you fill uncomfortable with the dockwidgets you can block them on the main window by checking the **Lock dockwidgets** option (see **Tools** > **Lock dockwidgets**). All the dockwidgets will be docked on the main window and locked on it except the player dockwidgets.

2.14.3 Configuration files

BORIS saves the configuration (user preferences, windows position, widgets position) in a configuration file. This file is named **.boris** and it is saved on the home directory of the current user:

```
for Linux:
/home/USERNAME/.boris

for Microsoft-Windows:
C:\Users\USERNAME\.boris

for MacOS:
/Users/USERNAME/.boris
```

If you have some trouble using BORIS try to close the program, delete this file and relaunch BORIS.

The **recent projects list** is saved on the **.boris_recent_projects** file in the home directory of the current user.

2.14.4 Lock the dockwidgets

The dockwidgets (except the player dockwidgets) can be locked on the main window (See **Tools** > **Lock dockwidgets**).

2.14.5 Valid keys for triggering behavior

BORIS makes difference between lower case and upper case characters

- keys from a to z
- keys from A to Z
- keys from 0 to 9
- function keys from F1 to F12
- à é è ù ì ç
- ! " £ $ % & / ( ) = ? ^ [ ] { } @ | § ° #

3. Community

3.1 Acknowledgement

The authors would like to acknowledge all the users that report bugs and/or request features for their precious help.

3.2 Citing BORIS

If you have used BORIS for publications, please cite:

```
Friard, O. and Gamba, M.,
BORIS: a free, versatile open-source event-logging software for video/audio coding and live observations.
(2016) Methods Ecol Evol, 7: 1325–1330.
```

DOI: 10.1111/2041-210X.12584

You can also send us a nice postcard.

Please consider to give a star to the BORIS GitHub repository.

3.3 Bug reports and features request

Please report any bug you will find in the latest BORIS version using the GitHub repository.

Prior to report a bug please:

- Check the Frequent Asked Question (FAQ) section
- Check if the issue was not already reported (GitHub repository)
- Delete the configuration file and try again (see configuration file).

Remember to indicate:

- your operating system
- the version of your operating system
- the computer you are using (model, RAM ...)
- the version of BORIS you are using
- Information on the mediafile you are coding (if any) See Tools > Media file information

Give all the information that will allow to reproduce the bug: a detailed procedure, a screen recording, etc.

In case of crash please send me the file boris_error.log generated in your home directory just after the crash (before relaunching BORIS):

```
Linux:
/home/YOUR_PROFILE_NAME/boris_error.log

Microsoft-Windows:
c:\Users\YOUR_PROFILE_NAME\boris_error.log
```

Note

If the bug you have reported is fixed remember to close the issue.