

Behavioral Observation Research Interactive Software (BORIS) user guide



version 5.1.0

BORIS is an easy-to-use event logging software for video/audio coding and live observations. BORIS is a free and open-source software available for GNU/Linux, Windows and Mac OS X.

BORIS web site: www.boris.unito.it

BORIS GitHub repository

BORIS twitter feed

BORIS discussion group

BORIS Facebook page

BORIS Google+

Contents

| | |
|---|-----------|
| Legal | 3 |
| Install BORIS | 3 |
| Linux | 4 |
| Microsoft-Windows | 4 |
| Mac OS | 4 |
| VirtualBox virtual appliance | 4 |
| Launch BORIS | 4 |
| Create a new project with BORIS | 5 |
| Set an ethogram | 5 |
| Define the subjects | 11 |
| Define the Independent variables | 12 |
| Observations' tab | 13 |
| Open an existing project with BORIS | 13 |
| Observations | 13 |
| Create a new observation | 13 |
| Observations list | 16 |
| Delete observations | 20 |
| Import observations from a project file | 20 |
| Coding your media | 20 |
| Media controls are available in the toolbar | 20 |
| Ethogram widget in the main window | 21 |
| Subjects widget in the main window | 22 |
| Media player widget in the main window | 22 |
| Events widget in the main window | 23 |
| Record an event | 23 |
| Frame-by-frame mode | 25 |
| Exporting events data | 25 |
| Export events | 26 |
| Export aggregated events | 26 |
| Export events as behavioral strings | 27 |
| Export events as Praat TextGrid | 27 |
| Extract sequences from media files | 27 |
| Export transitions matrix | 28 |
| Playback menu | 29 |
| Jump | 29 |
| Zoom | 29 |

| | |
|---------------------------------------|-----------|
| Tools | 29 |
| Geometric measurements | 29 |
| Coding pad | 31 |
| Transitions flow diagram | 32 |
| Re-encoding and resizing a video file | 33 |
| Project server | 34 |
| Coding map | 34 |
| The Behaviors coding map | 34 |
| The Modifiers coding map | 38 |
| Analysis | 39 |
| Time budget | 39 |
| Synthetic time budget | 41 |
| Plot events | 42 |
| Inter-rater reliability | 44 |
| Preferences | 44 |
| General preferences | 45 |
| FFmpeg framework | 46 |
| Frame-by-frame mode | 46 |
| Spectrogram | 47 |
| Plot colors | 47 |
| Various | 48 |
| Citing BORIS | 48 |
| Bug reports and features request | 48 |
| Docking | 48 |
| Acknowledgement | 49 |
| Valid keys for triggering behavior | 49 |

Legal

Copyright 2012-2017 Olivier Friard - Marco Gamba

BORIS is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or any later version.

BORIS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the [GNU General Public License](#) for more details.

Install BORIS

BORIS can be installed following instructions on the BORIS site at [download page](#)

Linux

BORIS is available as a DEB package for DEB based distributions based (like Debian and Ubuntu). After installation you will find BORIS in the **Education** sub-menu.

You can also run BORIS from sources. See the download page for details about the dependencies to install.

Microsoft-Windows

BORIS is available for Windows in 32-bit and 64-bit versions. Check your OS before downloading.

Two versions are available: **BORIS Setup** and **BORIS Portable**

The **BORIS Setup** will install BORIS on your system (the default install path is located in your home directory, you do not need administration rights to install BORIS).

The **BORIS Portable** will NOT install BORIS on your system but will run from a directory (find and launch boris.exe). Use this version if you want to test a new version of BORIS for example.

BORIS debugging version

Use this version if you want to report bug. In this case you should include screenshots or copy of the terminal output. This version will NOT install BORIS in your system.

Mac OS

BORIS is available for Mac OS only in 64-bit version (from >= 10.8)

VirtualBox virtual appliance

If you want to try BORIS without installing it you can download the [VirtualBox virtual appliance](#) from the BORIS site. In this case you must first install [VirtualBox](#)

Launch BORIS

Start BORIS by clicking on its icon. For launching BORIS from sources see the BORIS web site. When launching BORIS for the first time it may take some time to show up. Please be patient!

WARNING

On Mac OS X 10.9 and above, launching can be stopped according to the `Security & Privacy` settings of your computer. They can be changed opening `System Preferences...` > `Security & Privacy` > `General` and selecting the option `Anywhere` in the frame `Allow apps downloaded from`. Alternatively you can repeat the operation right-clicking on the BORIS icon and then clicking `Open` in the following two dialog boxes.

WARNING

On Microsoft-Windows, launching can be stopped by a `Security warning: The publisher can not be verified. Are you sure you want to run this software?` Click the `Run` button to launch BORIS.

At the first launch, BORIS will ask you to allow the automatically check for new version. Internet access is required for this feature. If you choose **Yes** BORIS will check for new version about every 15 days. This option can be changed on the [Preferences](#) window (See [general preferences](#))

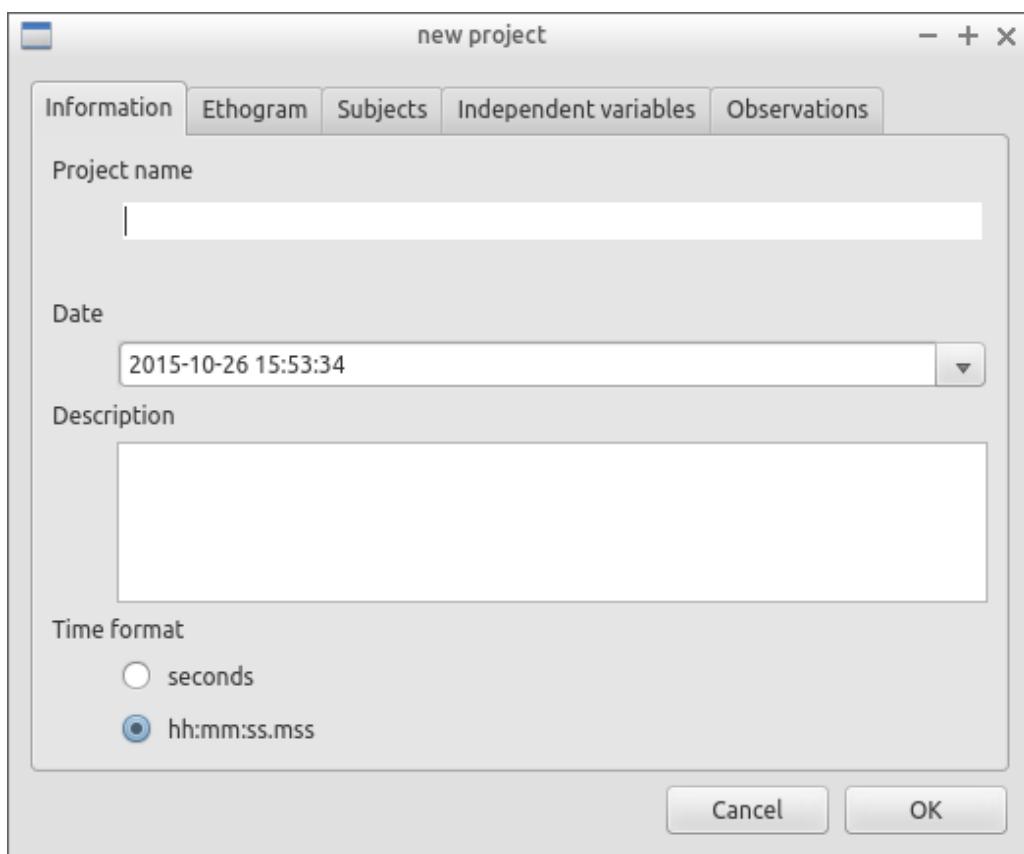
You can launch BORIS from command line by specifying the project file to open as the first argument.

Create a new project with BORIS

The BORIS project file is the container for all information related to the project. It contains the ethogram, the independent variables, the subjects and all observations data. The save menu option (**File > Save project**) will save all this information. You can also activate the [automatic backup](#) feature (see [Preferences](#)). BORIS allows creating 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 <https://www.youtube.com/watch?v=UGou4EVcLm0>

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** (for MAC users, **BORIS > Preferences**) > **Default project time format**.



Set an ethogram

See <https://en.wikipedia.org/wiki/Ethogram> for 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

Information Ethogram Subjects Independent variables Observations

| Behavior type | Key | Code | Description | Category |
|----------------|-----|------|--------------------------|--|
| 11 State event | ; | M | Manipulate | Otter handles an object in its environment (Hasenjager, 2011) |
| 12 State event | ; | F | Look for food | Otter looks for food in the environment |
| 13 State event | ; | S | Sniff | Otter moves the nose and head movement back and forth while not engaged in the rub behavior (Ethol, 2015) |
| 14 State event | ; | L | Locomotion | Otter moves from place to place |
| 15 State event | ; | I | Interact with enrichment | Otter interacts with enrichment |
| 16 State event | ; | E | Eat | Otter consumes food |
| 17 State event | ; | S | Swim | Otter swims in the water |
| 18 State event | ; | D | Drink | Otter consumes water found in its environment |
| 19 State event | ; | D | Dig | Otter uses front legs to move sand, stones on the ground (Ethol, 2015) |
| 20 State event | ; | D | Defecate | Otter eliminates fecal matter (Ethol, 2015) |
| 21 State event | ; | Z | Stomp | Otter stomps its rear legs alternately up and down several times before and after defecates/urinates (Ethol, 20... |
| 22 State event | ; | U | Urinate | Otter eliminates urine |
| 23 State event | ; | C | Chase | Otter chases other animals |
| 24 State event | ; | C | Carry objects | Otter carries objects or food by holding them against the cheek with one front paw while hobbling on three le... |
| 25 State event | ; | B | Breed | Otters are engaged in copulatory activities (Hasenjager, 2011) |
| 26 State event | ; | Q | Allogroom | Otter licks or scratches with forepaws or hind-paws another river otter's fur (Ethol, 2015) |
| 27 State event | ; | A | Alert | Otter is stationary and directs its attention towards something or someone (Hasenjager, 2011) |
| 28 Point event | ; | V | Vocalize | Otter vocalizes |
| 29 Point event | ; | Y | Yawn | Otter yawns |

Add behavior
Clone behavior
Remove behavior
Remove all behaviors
Behavioral categories

Exclusion matrix
Import behaviors from a BORIS project
Import from JWatcher
Import from text file

Cancel OK

Set your ethogram from scratch

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

Behavior types

2 types of behaviors can be defined:

- **Point event** behavior when the behavior has no duration. The behavior will be coded by pressing the defined keyboard key (see below) or by 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 clicking to the corresponding row in the Ethogram table. These behaviors **must** have a start event and a stop event.

You can switch between **State event** and **Point event** at your convenience from the **Behavior type** column. 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.

Behavioral codes (**Code** column) can be sorted alphabetically by checking the **Alphabetical order** checkbox. Alternatively they can be sorted manually by using the **Move up** and **Move down** buttons.

Categories of behaviors

Defining categories of behaviors can be usefull for the analysis of coded events (for example the time budget analysis). Click the **Behavioral categories** button and add a the categories of behaviors. Behaviors can then be included or not in a defined category.



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-insensitive.

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

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 following three columns (**Modifiers**, **Exclusion**, **Coding map**) cannot be edited from the *Ethogram* table.

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. "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.

3 types of modifiers are available: **Single selection**, **Multiple selection** and **Numeric**. **Single selection** modifiers allows the observer to select only one modifier in the list. **Multiple selection** allows to select more modifiers from the list. **Numeric** allows to input a numeric data.

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.

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



Set a name for the new modifiers set (new in v. 4) by typing it in the **Set name** edit box. Setting a modifiers' set name is not mandatory.

Select the modifier type using the **Modifier type** combo box. You can choose between **Single selection**, **Multiple selection** and **Numeric**

Within a set of modifiers, you can add and remove modifiers using the **Modifier** field and clicking on the **right-arrow** button (to add) and the **Remove modifier** button (to remove). The selected modifier can be edited using the **left-arrow** button. The **Key code** box can be used to set a shortcut key for the modifier (optionnal).

The modifier position into the modifiers' set can be manually set using the **Move modifier up** and **Move modifier down** buttons.

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:



Many values can be selected together.

Example of a 2 sets of modifiers:





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 **Behaviors 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 cell in the automatically-generated matrix. We suggest to work on the **Exclusion matrix** when all the behaviors have been added to your ethogram.

Example of an **exclusion matrix**:

In the previous example the Locomotion behavior will exclude **Alert**, **Allgroom**, **Breed**, **Carry objects** ...

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

Import an ethogram from an existing project

Behaviors within an ethogram can be imported from an existing BORIS project (.boris) using the **Import behaviours 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 JWatcher global definition file (.gdf)

Behaviors can be imported from a JWatcher global definition file (.gdf) using the **Import from JWatcher** button. BORIS will ask to select a JWWatcher 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 JWWatcher have to be redefined.

Import an ethogram from a plain text file

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

The fields will be interpreted as:

- field #1: event type (point or state)
- field #2: key (case insensitive)
- field #3: code (must be unique)
- field #4: behavior category (empty if no category)
- field #5: description (optional)

All fields after the 5th will be ignored.

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.

Export the ethogram

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

Define the subjects



BORIS allows coding behaviors for different subjects within a single observation. The **Subject** table allows specifying subjects using a **Key** (e.g. the "K" on your keyboard), **Subject name** (e.g. "Kanzi"), **Description** (e.g. male, born October 28 - 1980). In this case, pressing "N" will set "Nina" as the focal subject of the behavioural coding. Pressing "N" again will deselect "Nina" and set to "no focal 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). The subjects can also be imported from an existing BORIS project using the **Import Subjects from a BORIS project**.

Define the Independent variables

edit project

| Information | | Ethogram | | Subjects | | Independent variables | | Observations | |
|-------------|-------------|---------------------------------------|--|----------------|------------------------------|-----------------------|----------------|--------------|--|
| | Label | Description | | Type | Predefined value | | Set of values | | |
| 1 | Location | Location where observations were made | | text | 44° 55' 59" N - 7° 25' 18" E | | | | |
| 2 | Temperature | Average temperature of the day (°C) | | numeric | | | | | |
| 3 | Visitors | Visitors per day | | numeric | | | | | |
| 4 | Weather | Meteorological conditions | | value from set | sun | | sun,rain,cloud | | |

Add variable
Remove variable
Import variables from a BORIS project

Label: Location
Description: Location where observations were made
Type: text
Predefined value: 44° 55' 59" N - 7° 25' 18" E

Cancel OK

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 independent variable defined as "set of values"



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

Observations' tab

The **Observations** table in BORIS shows information about all the previous observations within a project. A selected "Observation" can be removed using the "Remove observation" button (you will be prompted for confirmation). This operation cannot be undone and deleted observations cannot be recovered once the project is saved. The **Observations** table shows four columns **id Date Description Media**.

Open an existing project with BORIS

IMPORTANT

Due to changes in modifiers format all the projects created with v. 4+ will not be suitable for previous version of BORIS. The v. 4 will open projects created with previous version and convert them. A copy of your old project will be kept.

To open an existing BORIS project, select the menu **File > Open project**. A BORIS project file is a container for all information related to a set of observations as the ethogram, the independent variables, and the subjects. BORIS allows creating an unlimited number of projects but only one project can be opened at a time.

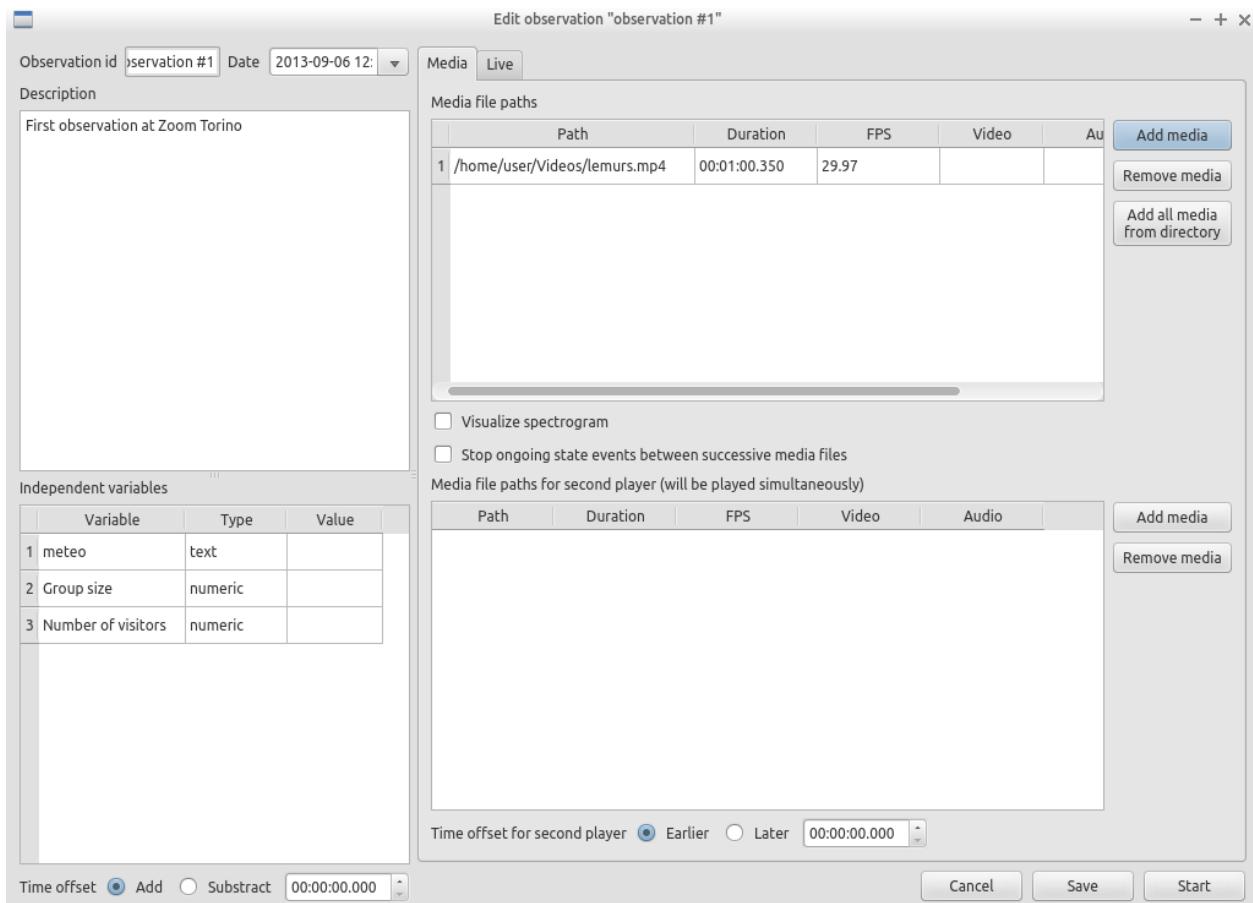
Observations

Create a new observation

A video tutorial about making an observation is available at <https://www.youtube.com/watch?v=ef-d6WEc0po>

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.



This window allow adding various observation data:

- a mandatory **Observation id** (must be unique across all observations);
- **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.

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

Live observation

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

Choose the **Live tab** to make a live observation.



In the above tab 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.

Media based observation

Choose the **Media** tab to make an obervation based on media.



In the **Media** tab there are two playlists. In the **Media file paths** playlist you can add one or more media files using the **Add media** button. Information about the selected media file will be extracted and displayed in the media list: media file path, media duration (in seconds), number of frames by second (FPS), video stream, audio stream .

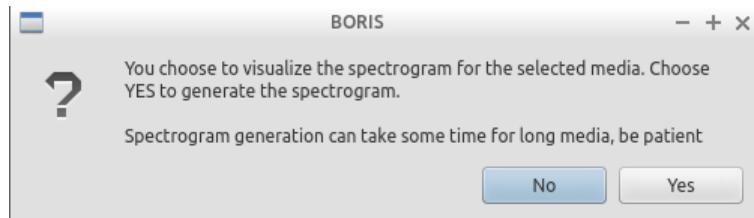
If you have to add various media files, you can use the **Add all media from directory** button, in this case all the media files found in the selected directory will be added in the playlist.

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

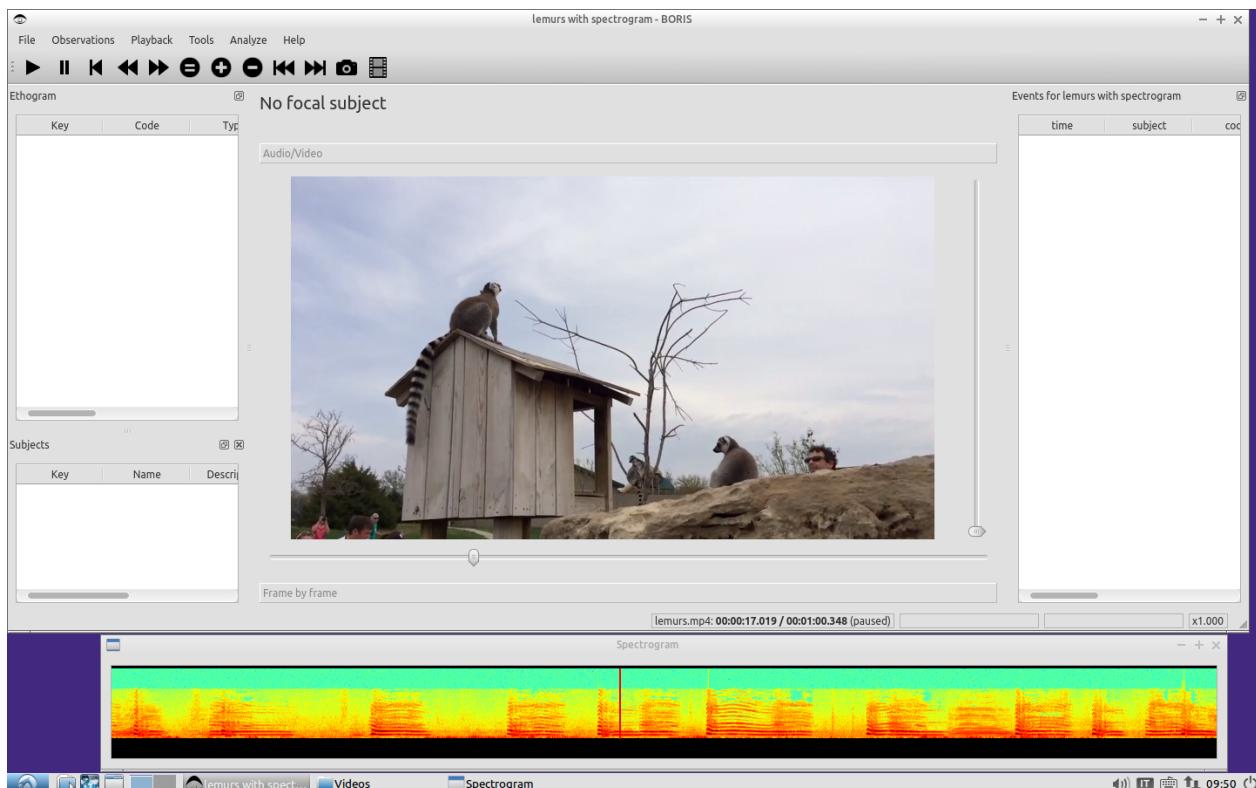
All the media types reported at <http://www.videolan.org/vlc/features.html> can be played in BORIS. The media queued in the *Media file paths* will be played sequentially. This means that an event occurring at time t_x in the media file queued as second (e.g. second_video.mp4) in the playlist will be 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).

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.



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



Close current behavior between videos

If this option is selected BORIS will close all ongoing State events between successive media files.

This option can be useful if you have to code various short media files enqueued in the first player (for example the output of a Camera trap).

Simultaneous play

BORIS also allows simultaneous playback of two media recorded independently (e.g. videos recorded from different points in a room; or a video and an audio recording of the same observation). The videos to be played simultaneously can be loaded in the **Media file paths for second player** playlist. In this case only one video per playlist is accepted. If the two media are not synchronised you can specify a time offset for the second player.

Click OK to start coding. The **Observation** window will be closed and you'll be transferred to the main **BORIS** window.

Observations list

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

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

Observations list - BORIS

1440 observations

| | id | date | description | subjects | |
|----|-----------|---------------------|----------------------------|-------------------------------|---------------|
| 1 | 0001 | 2016-05-17 00:00:31 | Vegetation | Nautilus, Himal | #1: C:/Users/ |
| 2 | 0002 | 2016-05-17 00:00:24 | Vegetation | Nautilus, Sharky, Himal | #1: C:/Users/ |
| 3 | 0003 | 2016-05-17 00:00:05 | Vegetation | Nina, Nautilus, Sharky, Himal | #1: C:/Users/ |
| 4 | 0004 | 2016-05-17 00:00:59 | Central trunks | Nina, Nautilus, Sharky, Himal | #1: C:/Users/ |
| 5 | 0005 | 2016-05-17 00:00:49 | In the pool | Nautilus, Sharky | #1: C:/Users/ |
| 6 | 0006 | 2016-05-17 00:00:42 | In the pool | Nautilus, Sharky, Himal | #1: C:/Users/ |
| 7 | 0007 | 2016-05-17 00:00:13 | In the pool | Nina, Sharky, Himal | #1: C:/Users/ |
| 8 | 0008 | 2016-05-17 00:00:17 | In the pool | Nautilus, Sharky, Himal | #1: C:/Users/ |
| 9 | 0009 | 2016-05-17 00:00:10 | In the pool | Nina, Sharky, Himal | #1: C:/Users/ |
| 10 | 0010 | 2016-05-17 00:00:57 | Area near the glass window | Nautilus, Sharky, Himal | #1: C:/Users/ |
| 11 | 0011 | 2016-05-17 00:00:50 | Area near the glass window | Nina, Nautilus, Sharky, Himal | #1: C:/Users/ |
| 12 | 0012 | 2016-05-17 00:00:45 | Area near the glass window | Nina, Sharky, Nautilus | #1: C:/Users/ |
| 13 | 0013 | 2016-05-17 00:00:25 | Central trunks | Nina, Nautilus, Sharky, Himal | #1: C:/Users/ |
| 14 | 0014 | 2016-05-17 00:00:52 | In the pool | Nina, Sharky, Nautilus, Himal | #1: C:/Users/ |
| 15 | 0015 | 2016-05-17 00:00:18 | Central trunks | Nina, Nautilus, Sharky, Himal | #1: C:/Users/ |

Cancel Open Edit

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 - BORIS

93 observations

| | id | date | description | subjects | |
|----|------|---------------------|-------------|-------------------------------|---------------------|
| 1 | 0005 | 2016-05-17 00:00:49 | In the pool | Nautilus, Sharky | #1: C:/Users/Eli... |
| 2 | 0006 | 2016-05-17 00:00:42 | In the pool | Nautilus, Sharky, Himal | #1: C:/Users/Eli... |
| 3 | 0007 | 2016-05-17 00:00:13 | In the pool | Nina, Sharky, Himal | #1: C:/Users/Eli... |
| 4 | 0008 | 2016-05-17 00:00:17 | In the pool | Nautilus, Sharky, Himal | #1: C:/Users/Eli... |
| 5 | 0009 | 2016-05-17 00:00:10 | In the pool | Nina, Sharky, Himal | #1: C:/Users/Eli... |
| 6 | 0014 | 2016-05-17 00:00:52 | In the pool | Nina, Sharky, Nautilus, Himal | #1: C:/Users/Eli... |
| 7 | 0185 | 2016-05-28 00:00:23 | In the pool | Nina, Nautilus, Sharky | #1: C:/Users/Eli... |
| 8 | 0212 | 2016-05-31 00:00:53 | In the pool | Sharky | #1: C:/Users/Eli... |
| 9 | 0217 | 2016-05-31 00:00:49 | In the pool | Nina, Himal | #1: C:/Users/Eli... |
| 10 | 0220 | 2016-05-31 00:00:36 | In the pool | Nina, Nautilus, Sharky, Himal | #1: C:/Users/Eli... |
| 11 | 0359 | 2016-06-06 00:00:54 | In the pool | Nina, Sharky, Nautilus, Himal | #1: C:/Users/Eli... |
| 12 | 0411 | 2016-06-07 00:00:53 | In the pool | Nina, Nautilus | #1: C:/Users/Eli... |
| 13 | 0412 | 2016-06-07 00:00:21 | In the pool | Nautilus | #1: C:/Users/Eli... |
| 14 | 0513 | 2016-06-10 00:00:48 | In the pool | Nina, Sharky, Nautilus, Himal | #1: C:/Users/Eli... |
| 15 | 0579 | 2016-06-13 00:00:32 | In the pool | Nina, Nautilus, Sharky, Himal | #1: C:/Users/Eli... |

Cancel Open Edit

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 :

Observations list - BORIS

148 observations

Weather does not contain Sunny

| | id | date | description | subjects | cati | Weather | Temperature | Visitors |
|----|------|---------------------|------------------|-------------------------------|------|---------|-------------|----------|
| 1 | 0197 | 2016-05-31 00:00:05 | Latrine | Nina, Sharky, Nautilus, Himal | ... | Cloudy | 16.6 | 215 |
| 2 | 0198 | 2016-05-31 00:00:39 | Latrine | Nina, Sharky | ... | Cloudy | 16.6 | 215 |
| 3 | 0199 | 2016-05-31 00:00:37 | Latrine | Nautilus, Himal | ... | Cloudy | 16.6 | 215 |
| 4 | 0200 | 2016-05-31 00:00:45 | Latrine | Nina, Nautilus, Sharky, Himal | ... | Cloudy | 16.6 | 215 |
| 5 | 0201 | 2016-05-31 00:00:48 | Latrine... | Nina, Nautilus, Sharky, Himal | ... | Cloudy | 16.6 | 215 |
| 6 | 0202 | 2016-05-31 00:00:12 | Latrine | Nina, Nautilus, Sharky, Himal | ... | Cloudy | 16.6 | 215 |
| 7 | 0203 | 2016-05-31 00:00:05 | Central trunks | Nina, Nautilus, Himal | ... | Cloudy | 16.6 | 215 |
| 8 | 0204 | 2016-05-31 00:00:04 | Central trunks | Nautilus, Himal | ... | Cloudy | 16.6 | 215 |
| 9 | 0205 | 2016-05-31 00:00:18 | Central trunks | Himal | ... | Cloudy | 16.6 | 215 |
| 10 | 0206 | 2016-05-31 00:00:55 | Central trunks | Nina, Nautilus, Sharky, Himal | ... | Cloudy | 16.6 | 215 |
| 11 | 0207 | 2016-05-31 00:00:55 | Central trunks | Nina, Nautilus, Sharky, Himal | ... | Rainy | 16.6 | 215 |
| 12 | 0208 | 2016-05-31 00:00:58 | Central trunks | Nina, Nautilus, Himal | ... | Cloudy | 16.6 | 215 |
| 13 | 0209 | 2016-05-31 00:00:00 | Central trunks | Nina, Nautilus, Sharky, Himal | ... | Cloudy | 16.6 | 215 |
| 14 | 0210 | 2016-05-31 00:00:19 | Central trunks | Nina, Sharky, Nautilus, Himal | ... | Cloudy | 16.6 | 215 |
| 15 | 0211 | 2016-05-31 00:00:56 | Bank of the pool | Nina, Nautilus, Himal | ... | Cloudy | 16.6 | 215 |

Cancel Open Edit

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

Observations list - BORIS

780 observations

Temperature between (use and to separate terms) 18 and 22

| | id | date | description | subjects | cati | Weather | Temperature | Visitors |
|----|------|---------------------|------------------|-------------------------------|------|---------|-------------|----------|
| 1 | 0065 | 2016-05-26 00:00:48 | Latrine | Nina | ... | Sunny | 20.1 | 1130 |
| 2 | 0066 | 2016-05-26 00:00:59 | Latrine | Nina, Sharky | ... | Sunny | 20.1 | 1130 |
| 3 | 0067 | 2016-05-26 00:00:49 | Latrine | Nautilus, Himal | ... | Sunny | 20.1 | 1130 |
| 4 | 0068 | 2016-05-26 00:00:54 | Latrine | Nina, Nautilus | ... | Sunny | 20.1 | 1130 |
| 5 | 0069 | 2016-05-26 00:00:22 | Latrine | Himal | ... | Sunny | 20.1 | 1130 |
| 6 | 0070 | 2016-05-26 00:00:42 | Latrine | Nina, Nautilus, Sharky, Himal | ... | Sunny | 20.1 | 1130 |
| 7 | 0071 | 2016-05-26 00:00:49 | Bank of the pool | Nina, Himal | ... | Sunny | 20.1 | 1130 |
| 8 | 0072 | 2016-05-26 00:00:12 | Central trunks | Nina, Sharky | ... | Sunny | 20.1 | 1130 |
| 9 | 0073 | 2016-05-26 00:00:27 | Central trunks | Sharky | ... | Sunny | 20.1 | 1130 |
| 10 | 0074 | 2016-05-26 00:00:02 | Central trunks | Sharky | ... | Sunny | 20.1 | 1130 |
| 11 | 0075 | 2016-05-26 00:00:20 | Central trunks | Nautilus, Himal | ... | Sunny | 20.1 | 1130 |
| 12 | 0076 | 2016-05-26 00:00:50 | Central trunks | Nina, Nautilus, Sharky, Himal | ... | Sunny | 20.1 | 1130 |
| 13 | 0077 | 2016-05-26 00:00:05 | Central trunks | Nina, Nautilus, Sharky, Himal | ... | Sunny | 20.1 | 1130 |
| 14 | 0078 | 2016-05-26 00:00:31 | Central trunks | Nina, Nautilus, Sharky, Himal | ... | Sunny | 20.1 | 1130 |
| 15 | 0079 | 2016-05-26 00:00:15 | Central trunks | Nina, Himal | ... | Sunny | 20.1 | 1130 |

Cancel Open Edit

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

| Observations list - BORIS | | | | | | | | | |
|---------------------------|------|---------------------|-------------------------|-------------------------------|------------|---------|-------------|----------|--|
| 824 observations | | | | | | | | | |
| Visitors | > | 1000 | | | | | | | |
| | id | date | description | subjects | ... ati | Weather | Temperature | Visitors | |
| 1 | 0001 | 2016-05-17 00:00:31 | Vegetation | Nautilus, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 2 | 0002 | 2016-05-17 00:00:24 | Vegetation | Nautilus, Sharky, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 3 | 0003 | 2016-05-17 00:00:05 | Vegetation | Nina, Nautilus, Sharky, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 4 | 0004 | 2016-05-17 00:00:59 | Central trunks | Nina, Nautilus, Sharky, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 5 | 0005 | 2016-05-17 00:00:49 | In the pool | Nautilus, Sharky | ... 4... | Sunny | 16.0 | 1046 | |
| 6 | 0006 | 2016-05-17 00:00:42 | In the pool | Nautilus, Sharky, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 7 | 0007 | 2016-05-17 00:00:13 | In the pool | Nina, Sharky, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 8 | 0008 | 2016-05-17 00:00:17 | In the pool | Nautilus, Sharky, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 9 | 0009 | 2016-05-17 00:00:10 | In the pool | Nina, Sharky, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 10 | 0010 | 2016-05-17 00:00:57 | Area near the glass ... | Nautilus, Sharky, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 11 | 0011 | 2016-05-17 00:00:50 | Area near the glass ... | Nina, Nautilus, Sharky, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 12 | 0012 | 2016-05-17 00:00:45 | Area near the glass ... | Nina, Sharky, Nautilus | ... 4... | Sunny | 16.0 | 1046 | |
| 13 | 0013 | 2016-05-17 00:00:25 | Central trunks | Nina, Nautilus, Sharky, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 14 | 0014 | 2016-05-17 00:00:52 | In the pool | Nina, Sharky, Nautilus, Himal | ... 4... | Sunny | 16.0 | 1046 | |
| 15 | 0015 | 2016-05-17 00:00:18 | Central trunks | Nina, Nautilus, Sharky, Himal | ... 4... | Sunny | 16.0 | 1046 | |

Delete observations

The observations can be deleted from the project using the following procedure: **File > Edit project > Observation's tab**

Select all observations to remove

Click the **Remove selected observations** button and confirm the deletion.

Please note that the deletion is irreversible.

Import observations from a project file

The **Observations > Import observations** option allows to import observations from a BORIS project file into the current project. Choose the 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.

Coding your media

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.

Media controls are available in the toolbar



Key to the symbols:

- **Play**
- **Pause** (the SPACE bar can be used)
- **Rewind** reset your media at the beginning
- **Fast backward** jumps for n seconds backward in your media (See [general preferences](#) to set n)
- **Fast forward** jumps for n seconds forward in your media (See [general preferences](#) to set n)
- **Set the playback at speed 1x**
- **Increase playback speed** (See [general preferences](#) to set the step value)
- **Decrease playback speed** (See [general preferences](#) to set the step value)
- **Jump to the previous media file**
- **Jump to the next media file**
- **Take a snapshot** of current video or frame
- **Switch between VLC and frame-by-frame modes**

In frame-by-frame mode:

- **Move on frame back**
- **Move one frame forward**

The media can be controlled by special keyboard keys:

- **Page Up** key: switch to the next media
- **Page Down** key: switch to the previous media
- **Up arrow** key: jump forward in the current media
- **Down arrow** key: jump backward in the current media
- **ESC**: switch between VLC and frame-by-frame mode

Ethogram widget in the main window

| Key | Code | Type | Description | Modifiers |
|-------|---------------|-------------|---|--------------------------------|
| 1 J | jump | Point event | jumping | foo,bar foo,bar foo,... |
| 2 G | groom | State event | Animal engages in washing or smoothing | self (s),others (o) |
| 3 E | eat | State event | Animal consumes food | alone,in group orang... |
| 4 O | look for food | State event | Animal searches the environment for food | |
| 5 S | sleep | State event | Animal assumes position for sleep, stays in one place and is not alert... | light (l),deep (d) |
| 6 D | dream | State event | Animal dreams | mvt (m),no mvt (n) |
| 7 H | fight | State event | Animal engages a physical contact with another animal | |
| 8 P | play | State event | Animal engages in interactions with others | alone (a),with others (o) b... |
| 9 L | locomote | State event | Animal moves from place to place | |

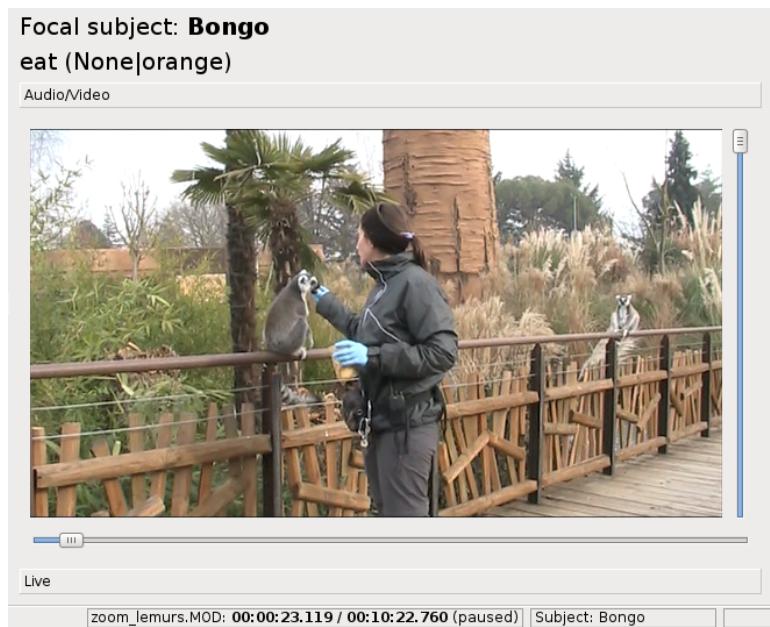
The **Ethogram** widget provide the user with the list of behaviors defined in the **Ethogram**. It can be used to record an event by double clicking on the corresponding row. The **Key** column indicates the keyboard key assigned to each behavior (if any). Pressing a key will record the corresponding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

Subjects widget in the main window

| Subjects | | | |
|----------|-------|------------------------------|------------------|
| Key | Name | Description | Current state(s) |
| 1 1 | Kanzi | Male, born October 28 - 1980 | |
| 2 2 | Bongo | Male, adult | |
| 3 3 | Marco | Orangutan | |

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).

Media player widget in the main window



The **Media player** widget has two controls: the media position (horizontal slide bar) and the audio volume (vertical slide bar) provide the user with the list of subjects defined in the **Subject** tab in the **Project** window. The horizontal slide bar can be used to navigate the media file.

Events widget in the main window

| Events for observation #4 | | | | | |
|---------------------------|--------------|---------|-------|-------|-----------------|
| | time | subject | code | type | modifier |
| 1 | 00:00:16.040 | Kanzi | eat | START | alone orange |
| 2 | 00:00:30.320 | Kanzi | eat | STOP | alone orange |
| 3 | 00:00:44.240 | Bongo | eat | START | in group banana |
| 4 | 00:00:53.740 | Bongo | eat | STOP | in group banana |
| 5 | 00:01:11.480 | Bongo | face | | orb occ |
| 6 | 00:01:21.920 | Bongo | groom | START | self |
| 7 | 00:01:31.240 | Bongo | groom | STOP | self |

The **Events** widget shows all the recorded behaviors with the following parameters (columns):

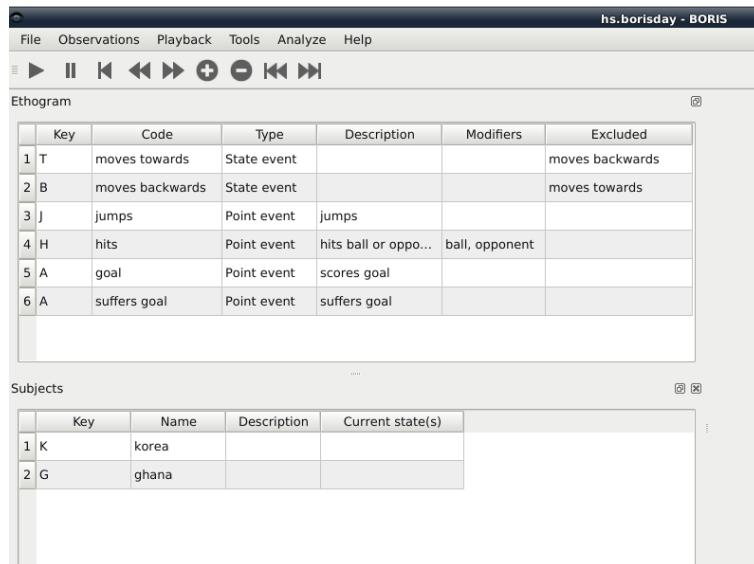
- **time**, the time at which the event occurred;
- **subject**, the focal subject (if any);
- **code**, the behavior code;
- **type**, in case of a state event indicates whether the time corresponds to the start or to the stop.
- **modifier**, indicates the modifier(s) that was(ere) selected (if any);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

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

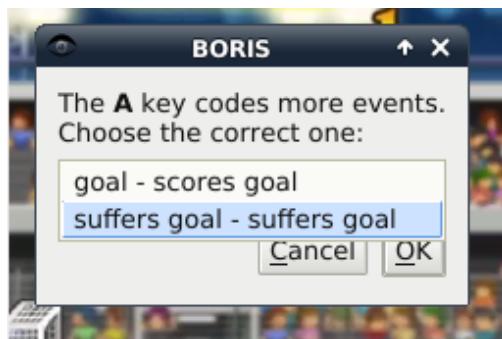
A double-click on a row will reposition the media player to the moment of the corresponding event. See [Time offset for media reposition](#) in Preferences window to customize the time offset for media repositioning.

Record an event

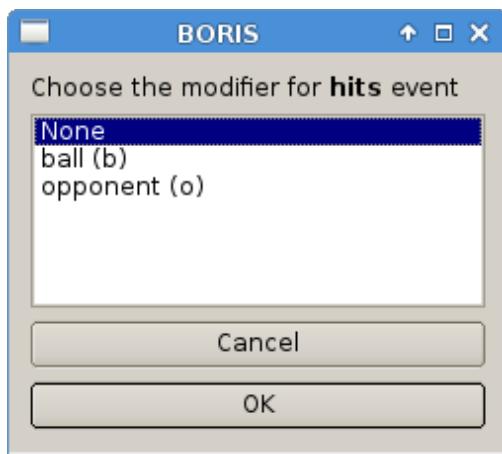
Once ready to begin your coding, you can start the media player using the **Play** button (or the Space bar). The behaviors can be recorded using the keyboard with the predefined keys, by double-clicking the corresponding row in the **Ethogram** table or by using the **Coding pad** (See [coding pad](#)).



If the pressed key defines a single event, the corresponding event will be recorded in the **Events** table. In the case you have specified the same key for two (or more) events (e.g. key A in the figure below), BORIS will prompt you for the desired 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 correct one using the mouse or the keyboard (**b** key or **o** key)



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.

Frame-by-frame mode

You can switch between the media player and the frame-by-frame mode using the dedicated button in the toolbar:



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

Warning

Please note that MTS video files should be re-encoded to be used in frame-by-frame mode. Otherwise the extracted frames are not reliable.

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

For the both modes (media player and frame-by-frame mode)

- **Page Up** key: switch to the next media
- **Page Down** key: switch to the previous media
- **Up arrow** key: jump forward in the current media
- **Down arrow** key: jump backward in the current media
- **ESC**: switch between VLC and frame-by-frame mode

Only for the frame-by-frame mode

- **Left arrow** key: go to the previous frame
- **Right arrow** key: go to the next frame

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 again the frame-by-frame button in the toolbar.

The frame can be resized before visualization. See the [frame resizing](#) option.

The frame viewer can be detached from the main window (See **File > Preferences**). In case on 2 simultaneous players the frame viewers will be automatically detached from the main window.

For every second of the media file the frames are extracted by BORIS with the embedded ffmpeg program and are saved in the [FFmpeg cache directory](#) specified in the [Preferences](#) window.

Exporting events data

The coded events can be exported in various format (**Observations > Export ?**):

Export events

This function will export the events of selected observations in TSV, XLS or ODS formats. These formats are suitable for further analysis.

| A | B | C | D | E | F |
|--------------------------|-------------------------|------|----------|---------|--------|
| 1 Observation id | demo #1 | | | | |
| 2 | | | | | |
| 3 Media file(s) | | | | | |
| 4 | | | | | |
| 5 Player #1 | /home/olivier/crop.avi | | | | |
| 6 Player #1 | /home/olivier/crop2.avi | | | | |
| 7 | | | | | |
| 8 Observation date | 2015-Nov-30 10:39:18 | | | | |
| 9 | | | | | |
| 10 Description | | | | | |
| 11 | | | | | |
| 12 Time offset (s) | 0 | | | | |
| 13 | | | | | |
| 14 independent variables | | | | | |
| 15 variable | value | | | | |
| 16 group size | 7 | | | | |
| 17 weather conditions | rain | | | | |
| 18 | | | | | |
| 19 time | subject | code | Modifier | comment | status |
| 20 | 4.3 Subject #1 | eat | salad | test | START |
| 21 | 10 Subject #1 | eat | salad | | STOP |
| 22 | 26.6 Subject #2 | eat | meat | | START |
| 23 | 113.988 Subject #2 | eat | meat | | STOP |
| 24 | 116.588 Subject #1 | jump | | | POINT |
| 25 | 118.988 Subject #1 | jump | | | POINT |
| 26 | 120.863 Subject #2 | jump | | | POINT |
| 27 | 122.438 Subject #2 | jump | | | POINT |
| 28 | | | | | |

Export aggregated events

This function will export the events of the selected observations in the following formats:

- **tabular format** (TSV, CSV, XLS, ODS, HTML)
- **SQL** format for populating a SQL database
- **SDIS** format for analysis with the GSEQ program available at <http://www2.gsu.edu/~psyrob/gseq>

The **State events** are paired and duration is available.

Example of tabular export

| A | B | C | D | E | F | G | H | I | J |
|------------------|----------------------|------------|----------|-----------|---------------|---------|---------|---------------|--------------|
| 1 Observation id | Observation date | Subject | Behavior | Modifiers | Behavior type | Start | Stop | Comment start | Comment stop |
| 2 demo #1 | 2015-Nov-30 10:39:18 | Subject #1 | jump | | POINT | 116.588 | 0 | | |
| 3 demo #1 | 2015-Nov-30 10:39:18 | Subject #1 | jump | | POINT | 118.988 | 0 | | |
| 4 demo #1 | 2015-Nov-30 10:39:18 | Subject #1 | eat | salad | STATE | 4.3 | 10.0 | test | |
| 5 demo #1 | 2015-Nov-30 10:39:18 | Subject #2 | jump | | POINT | 120.863 | 0 | | |
| 6 demo #1 | 2015-Nov-30 10:39:18 | Subject #2 | jump | | POINT | 122.438 | 0 | | |
| 7 demo #1 | 2015-Nov-30 10:39:18 | Subject #2 | eat | meat | STATE | 26.6 | 113.988 | | |
| 8 | | | | | | | | | |
| 9 | | | | | | | | | |
| 10 | | | | | | | | | |

Example of SQL export:

```
CREATE TABLE events (id INTEGER PRIMARY KEY ASC, observation TEXT,
                     date DATE, subject TEXT, behavior TEXT,
                     modifiers TEXT, event_type TEXT, start FLOAT,
                     stop FLOAT, comment_start TEXT,
                     comment_stop TEXT);

INSERT INTO events (observation, date, subject, behavior, modifiers,
                    event_type, start, stop, comment_start, comment_stop) VALUES
('demo #1', '2015-11-30 10:39:18', 'Subj #1', 'jump', '', 'POINT', 116.588, 0, "", ""),
('demo #1', '2015-11-30 10:39:18', 'Subj #1', 'jump', '', 'POINT', 118.988, 0, "", ""),
('demo #1', '2015-11-30 10:39:18', 'Subj #1', 'eat', 'salad', 'STATE', 4.3, 10.0, 'vvv', ""),
('demo #1', '2015-11-30 10:39:18', 'Subj #2', 'jump', '', 'POINT', 120.863, 0, "", ""),
('demo #1', '2015-11-30 10:39:18', 'Subj #2', 'jump', '', 'POINT', 122.438, 0, "", ""),
('demo #1', '2015-11-30 10:39:18', 'Subj #2', 'eat', 'meat', 'STATE', 26.6, 113.988, "", "");
```

Export events as behavioral strings

Behavioral string can be used with the BSA service: Behavioral Strings Analysis (BSA)

Example:

```
# observation id: demo#1
# observation description:
# Media file name: crop.avi, crop2.avi

Subject #1:
eat|eat|jump|jump

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

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"
```

Extract sequences from media files

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

1. Click on **Observations > Extract events from media files** option.
2. Choose the observation(s).

3. Select the events to be extracted.
4. Select a destination directory that will contain the extracted sequences.
5. Select a time offset (in seconds, the default value is 0).

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

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

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 |

Playback menu

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.

Zoom

Allow to zoom into the current video file. The available zoom values are: 1:2, 1:4, 2:1 and 1:1. **Fit to window** adapts the zoom value to the current window size. In case of simultaneous playing you can set different values of zoom in player #1 and player #2.

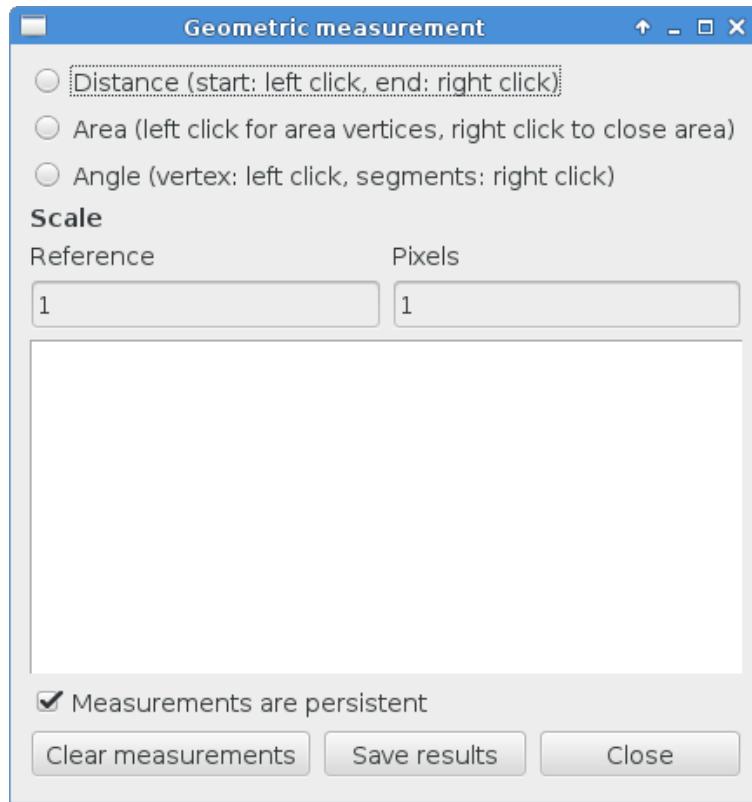
The Zoom option is not available in frame-by-frame mode.

The Zoom function is not working on Mac

Tools

Geometric measurements

Geometric measurements can only be made in frame-by-frame mode. Distances, areas and angles can be measured. Click on **Tools > Geometric measurements** to activate the measurements. The **Measurements window** will be shown:

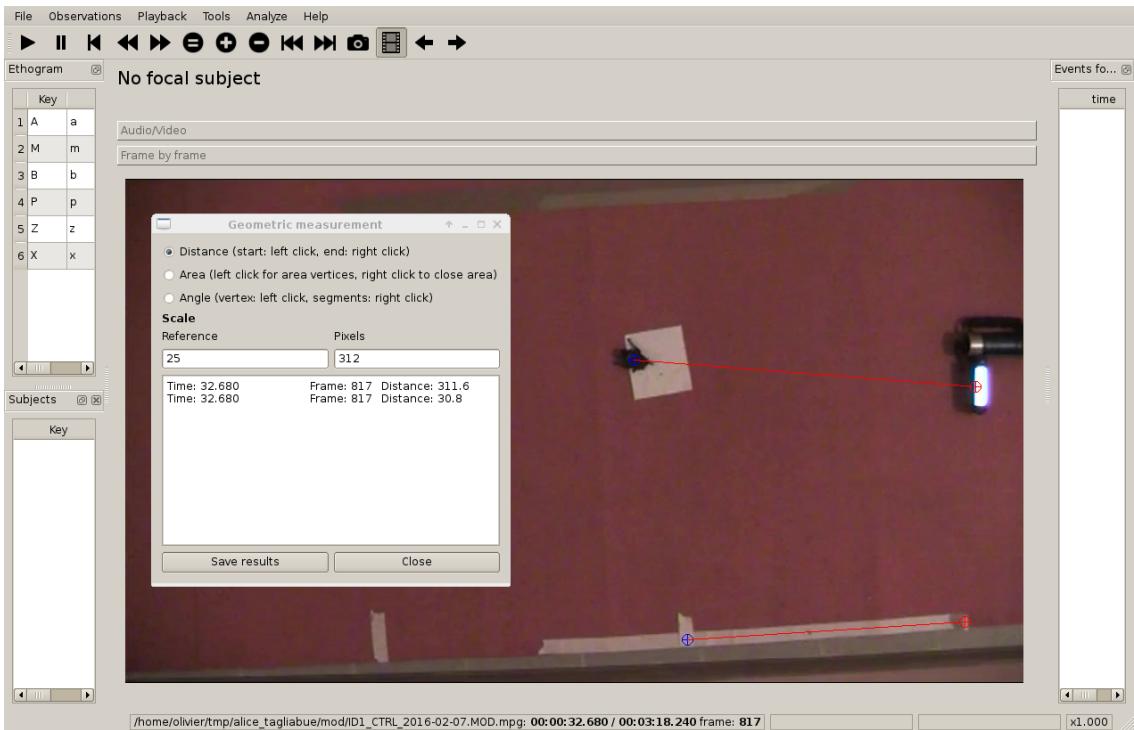


Setting the scale

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). 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).

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 blue 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. Blue 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

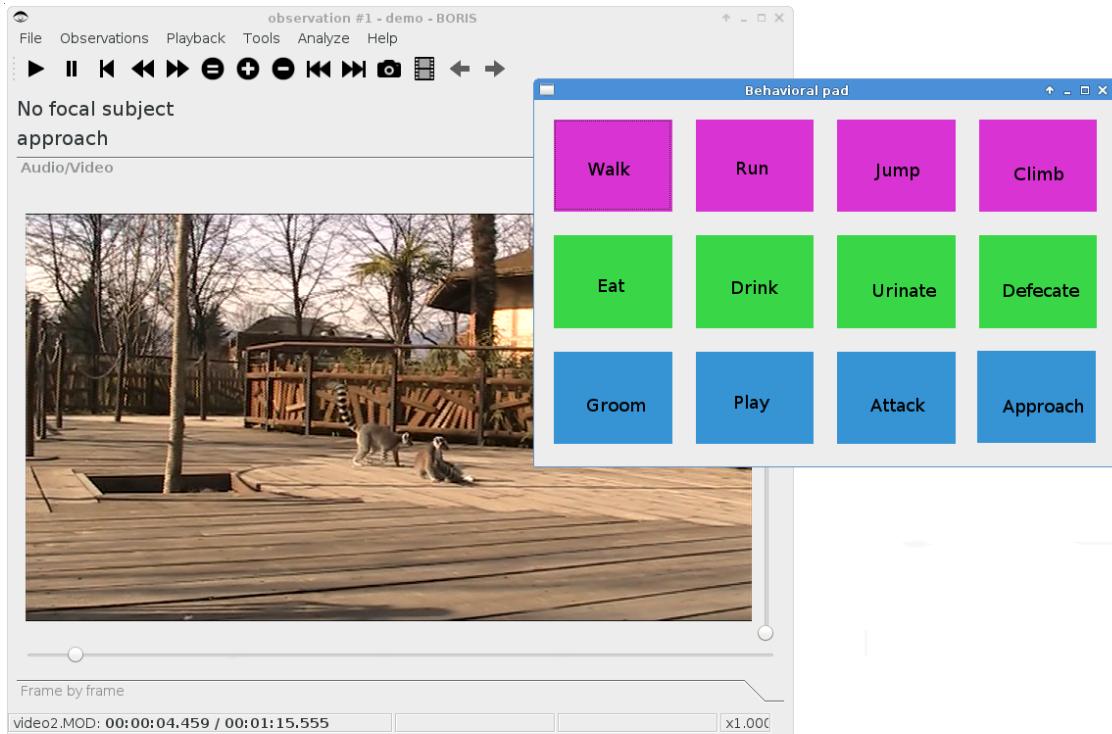
Select the **Angle** radio button. Click the left mouse button on the frame bitmap to set the angle vertex. A red circle with a cross will be drawn. Click the right mouse button to set the two segments. Blue circles with a cross will be drawn. The angle between the two drawn segments will be available 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.

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.



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.

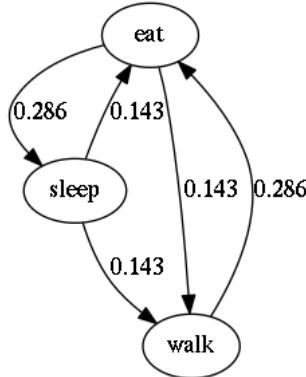
Flow diagram

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.

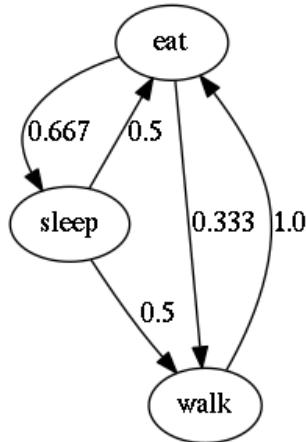
Tools > Transitions flow diagram > Create transitions flow diagram

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

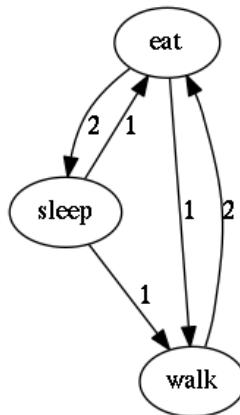
Flow diagram of frequencies of transitions



Flow diagram of frequencies of transitions after behavior



Flow diagram of number of transitions



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.

warning

The MTS video files should be re-encoded to be used in frame-by-frame mode. Otherwise the extracted frames are not reliable.

Project server

This function allow you to communicate with the **BORIS App** by sending project and receiving observations made with the **BORIS App**. For details read the **BORIS App user guide**

Starting the project server

- Open your project
- Select **File > Project server**

A window will open and show you the URL of the server that should be used on the **BORIS App**. The project server will serve the project for 30 minutes.

Stopping the project server

- Select **File > Stop serving project**

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

The Behaviors coding map

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

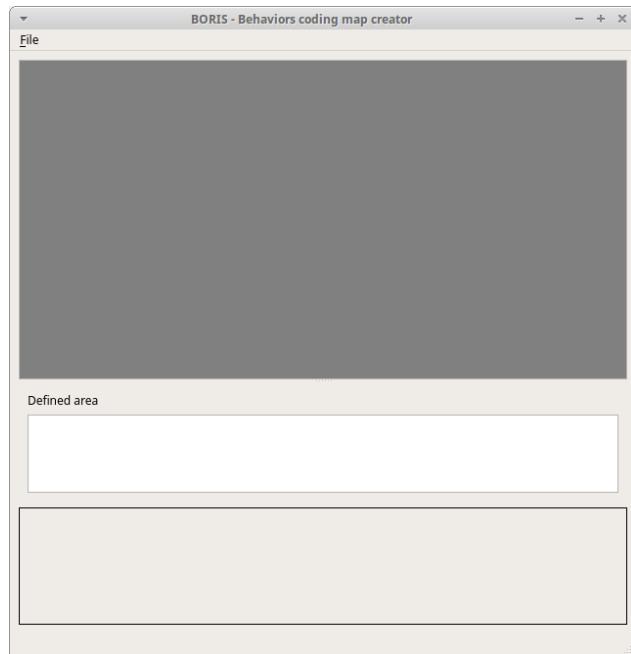
A **Behaviors coding map** can be created only if you have defined behaviors in your ethogram.

Creating a Behaviors coding map

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

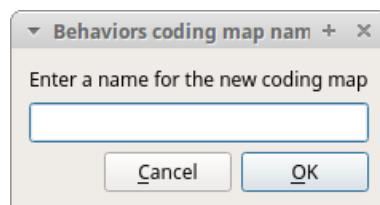
Tools > Create a coding map > for behaviors).

A new window will open



File > New behaviors coding map

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

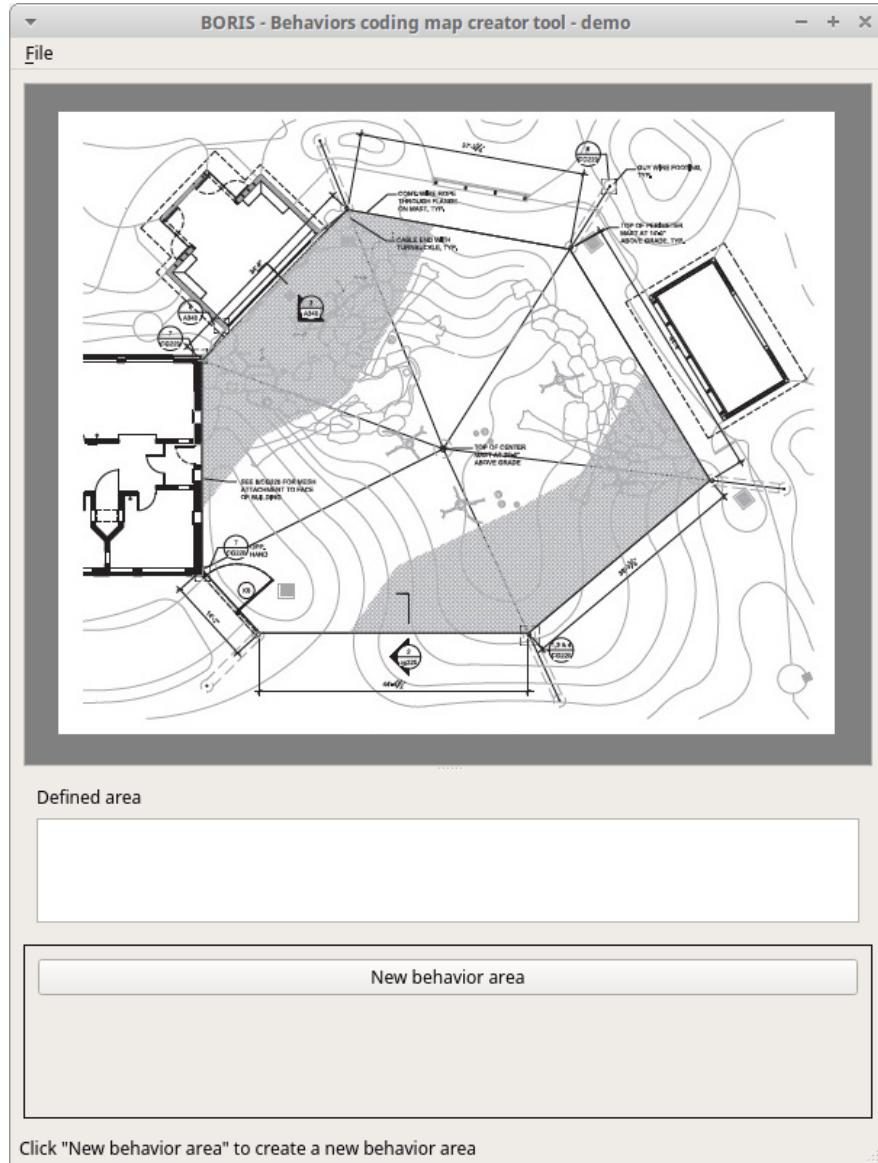


Loading a bitmap for a behaviors coding map

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

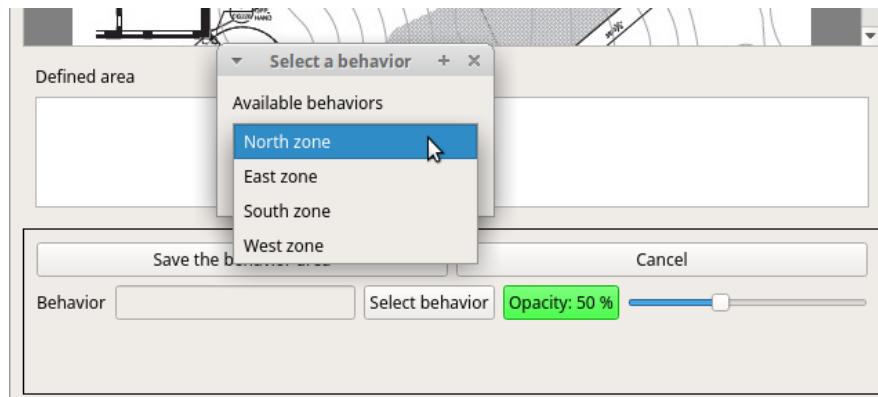
If the size of your bitmap image is bigger than 640 x 640 pixels BORIS will resize it to 640 x 640 pixels keeping the aspect ratio and store the resized version in the coding map file.

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**)

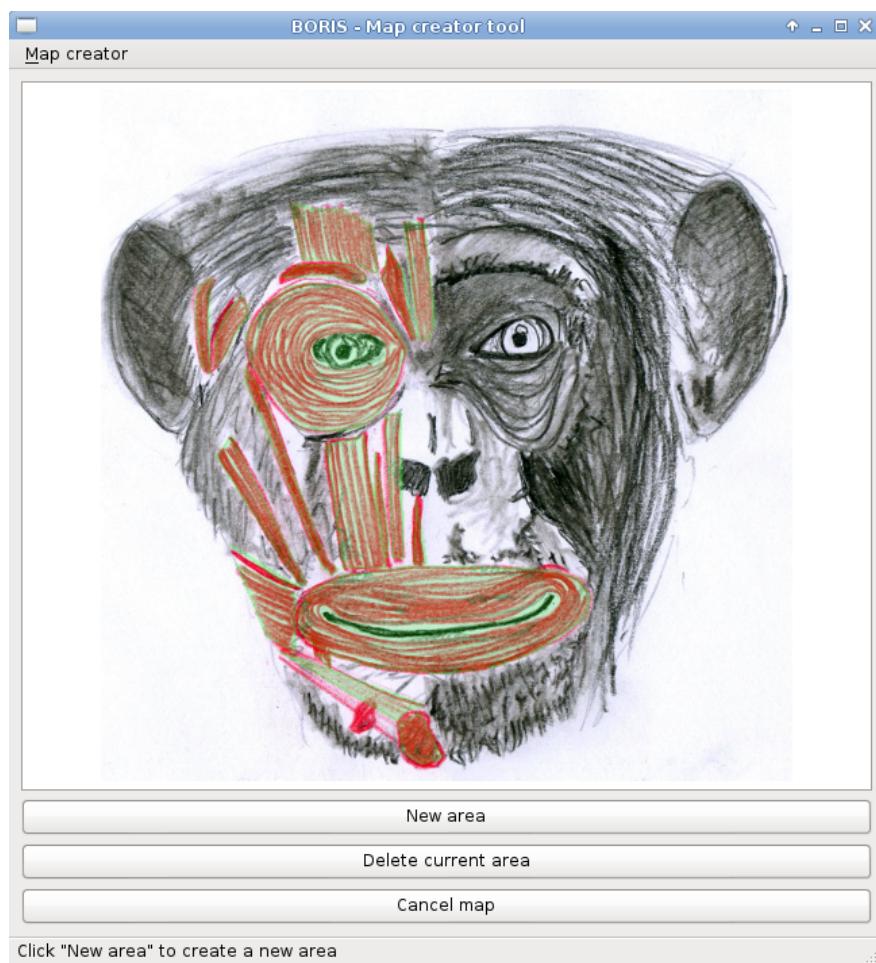
The Modifiers coding map

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.

NOTE: If you later modify your coding map you must reload the new version in your BORIS project.

Analysis

Time budget

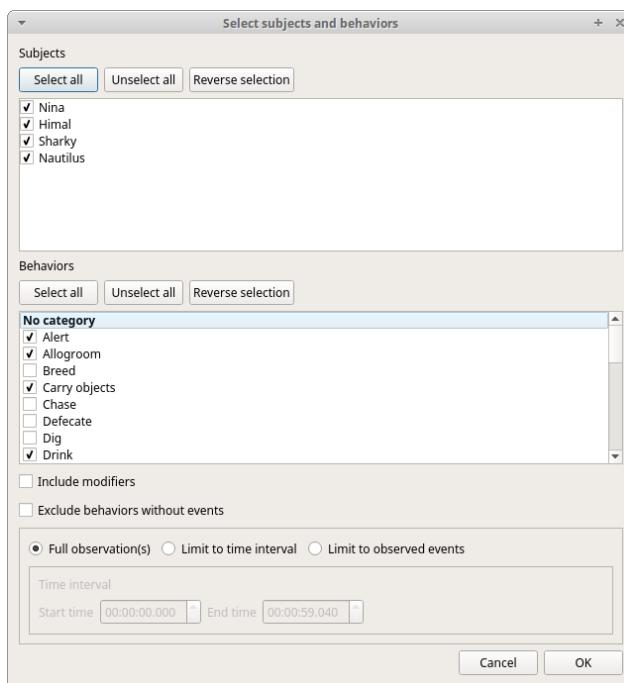
The **Time budget** analysis can be done by behavior or by category of behaviors. Choose the option from the **Analysis** menu.

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 on 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 results contain for each subject and behavior the **total number 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**, the **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).

Time budget

Selected observations

| | Subject | Behavior | Modifiers | Total number | Total duration (s) | Duration mean (s) | Duration std dev | inter-event intervals mean (s) | inter-event intervals std dev | % of total length |
|----|---------|--------------------|-----------|--------------|--------------------|-------------------|------------------|--------------------------------|-------------------------------|-------------------|
| 1 | Nina | Alert | - | 1 | 1.882 | 1.882 | NA | NA | NA | 0.4 |
| 2 | Nina | Locomotion | - | 2 | 2.913 | 1.457 | 0.824 | NA | NA | 0.7 |
| 3 | Nina | Manipulate | - | 2 | 65.048 | 32.524 | 15.414 | NA | NA | 14.6 |
| 4 | Nina | Rest | - | 1 | 25.549 | 25.549 | NA | NA | NA | 5.7 |
| 5 | Himal | Alert | - | 4 | 18.131 | 4.533 | 2.542 | 10.95 | NA | 4.1 |
| 6 | Himal | Allogroom | - | 1 | 3.603 | 3.603 | NA | NA | NA | 0.8 |
| 7 | Himal | Drink | - | 1 | 1.133 | 1.133 | NA | NA | NA | 0.3 |
| 8 | Himal | Locomotion | - | 10 | 36.633 | 3.663 | 5.777 | 6.124 | 5.88 | 8.2 |
| 9 | Himal | Manipulate | - | 3 | 66.702 | 22.234 | 17.18 | 4.723 | NA | 15.0 |
| 10 | Himal | Play on the ground | - | 1 | 16.46 | 16.46 | NA | NA | NA | 3.7 |
| 11 | Himal | Rub | - | 1 | 11.031 | 11.031 | NA | NA | NA | 2.5 |
| 12 | Himal | Self-groom | - | 1 | 3.571 | 3.571 | NA | NA | NA | 0.8 |

Total observation length: 00:07:25.680

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

- * Plain text files: TSV, CSV or HTML
- * Spreadsheets: Open Document (ODS), Microsoft Excel (XLSX) and Legacy Microsoft (XLS)

Note

If a STATE behavior has an odd number of coded events, BORIS will report "UNPAIRED" instead of results"

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 Alert | Nina Alert | Nina Allogroom | Nina Allogroom | Nina Breed | Nina Breed | Nina Carry objects | Nina Carry objects | Nina Chas |
| 2 | | Total length (s) | Total duration | Number of occurrences | Total duration | Number of occurrences | Total duration | Number of occurrences | Total duration | Number of occurrences |
| 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 files: TSV, CSV or HTML * Spreadsheets: Open Document (ODS), Microsoft Excel (XLSX) and Legacy Microsoft (XLS)

Plot events

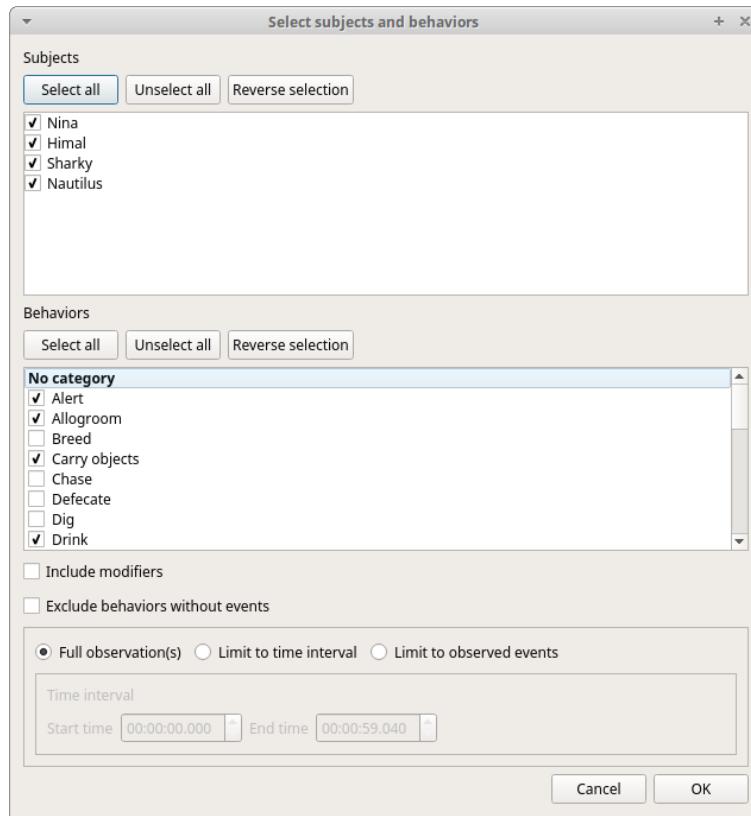
The events from a selected observation can be plotted along a time axis. Two types of plot are available.

Plot events (Back compatibility)

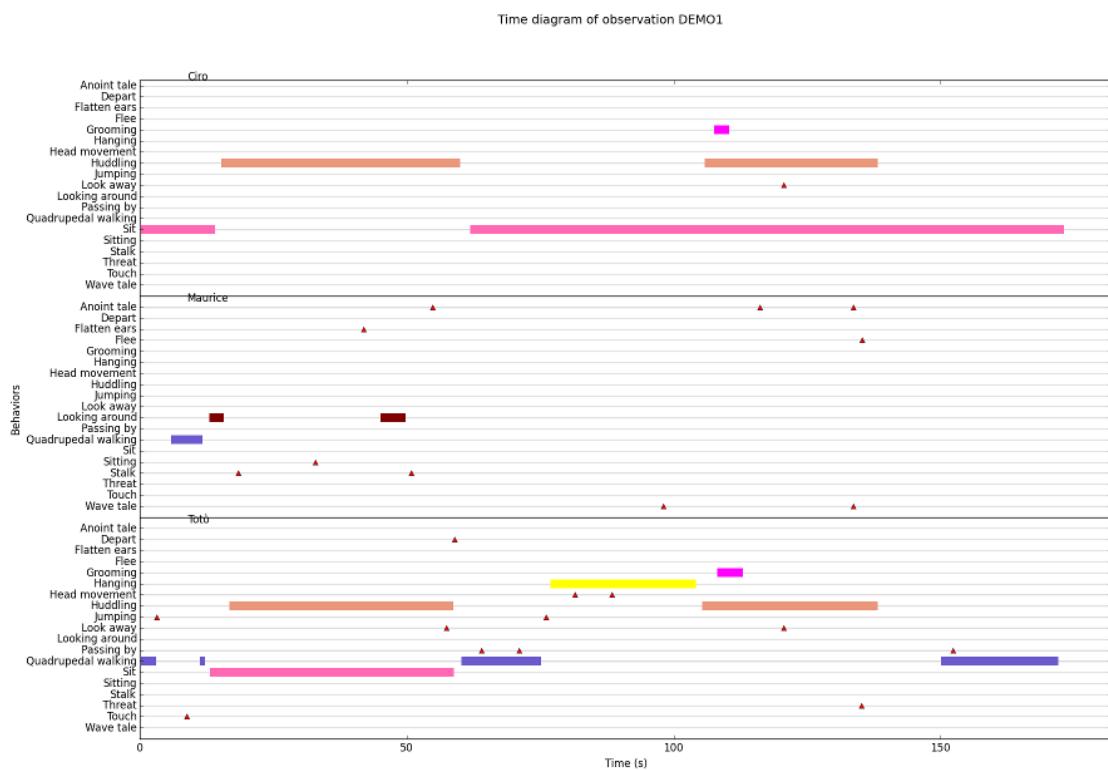
Analysis > Plot events (Back compatibility)

The function allows to plot only one observation.

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



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



The color of behaviors can be customized. See [plot colors](#)

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](#).

Note

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

Plot events

Analysis > Plot events

With this function you can select many observations to be plotted. This function creates one plot by subject on one figure.

The time interval can be selected (See time budget)

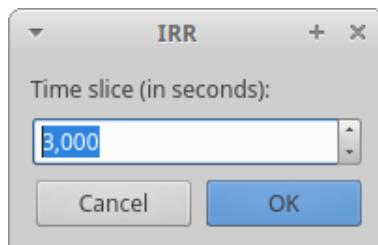
The color of behaviors can be customized. See [plot colors](#)

Inter-rater reliability

The Cohen's kappa coefficient can be calculated ([Analysis > Inter-rater reliability > Cohen's kappa](#)).

[Cohen's kappa on Wikipedia](#)

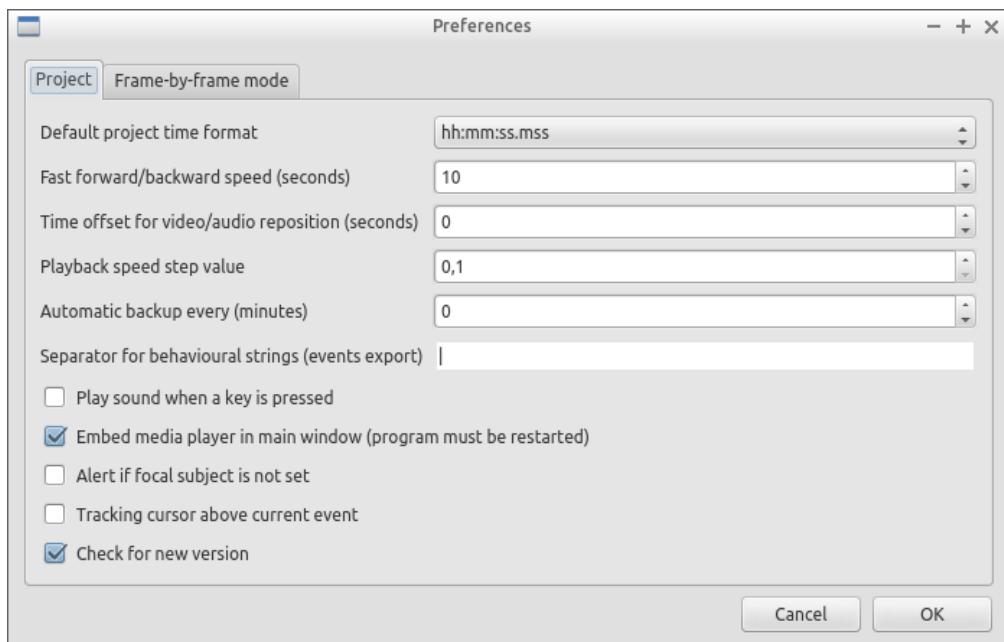
After selecting 2 observations and a time window (in seconds) for the analysis (the default value is 10 seconds) the Cohen's kappa will be displayed in the results window.



Preferences

You can customize BORIS using the Preferences window ([File > Preferences](#))

General 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

Fast forward/backward speed (seconds)

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

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.'

Playback speed step value

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

Automatic backup every (minutes)

If set BORIS will save your project automatically every n minutes. 0 indicate no automatic backup.

Play sound when a key is pressed

Activate a sound signal after every keypress event

Embed media player

This option allows the user to detach from the main window or embed the media player in the main window. **On Mac OS the media player can not be detached from main window.**

Alert if focal subject is not set

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

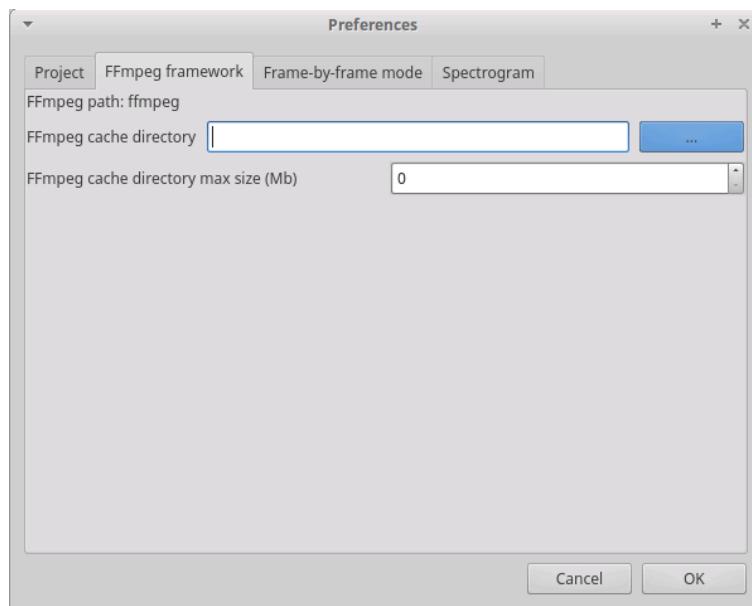
Tracking cursor above current event

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

Check for new version

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

FFmpeg framework



The path for the ffmpeg executable program is displayed. The FFmpeg executable is now included with BORIS in Windows and Mac OS versions.

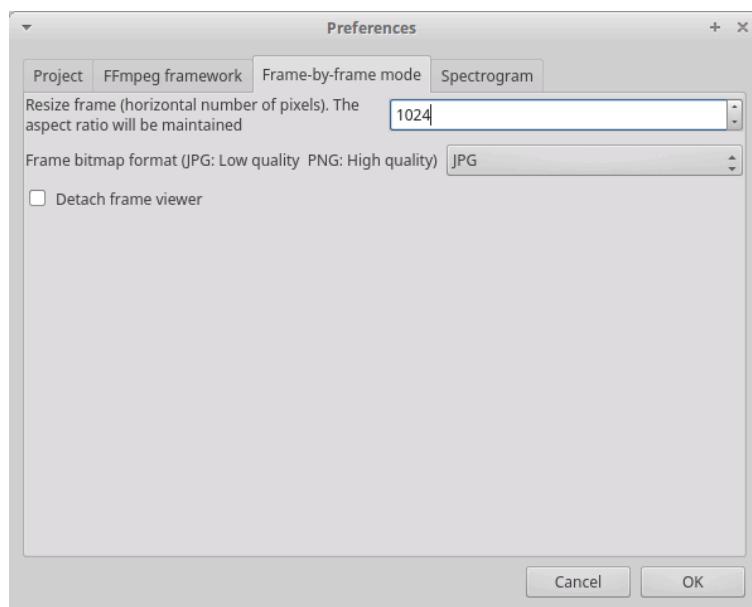
FFmpeg cache directory

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.

FFmpeg cache directory max size

Indicate a size limit (in Mb) for the image caching. 0 indicates no limit

Frame-by-frame mode



Resize frame

In case of high resolution video (for example 4K 1920x1080) the extracted frames can be resized before visualization to improve the speed. Select the width in pixel for the extracted frames. The aspect ratio will be maintained. 0 will indicate no frame resizing.

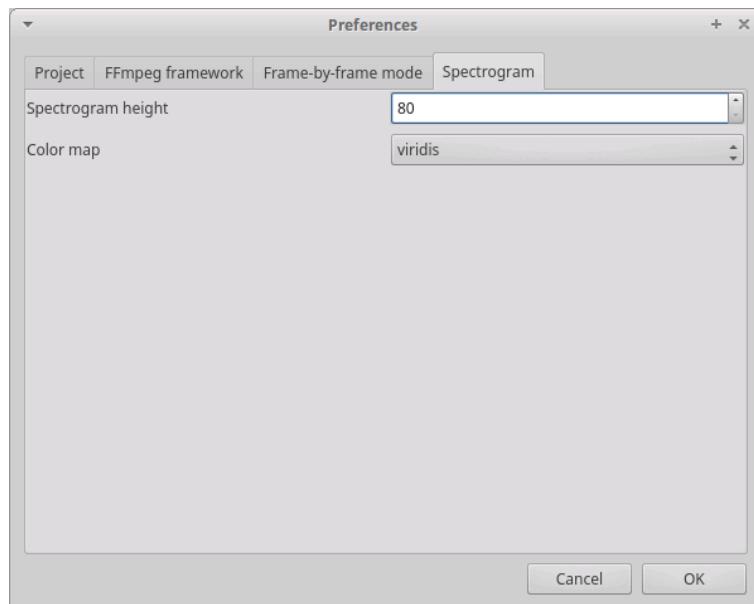
Frame bitmap format

A bitmap format can be selected between JPG (JPEG low quality image) and PNG (Portable Network Graphic - high quality image). The extracted frames will be saved in the directory defined in previous tab (FFmpeg cache directory).

Detaching the frame viewer

The frame viewer can be detached from the main window. Please note that in case of two simultaneous media files the frames viewers will be detached.

Spectrogram



Spectrogram height

Select the height of generated spectrogram (in pixels). You will need to restart the current observation to apply changes.

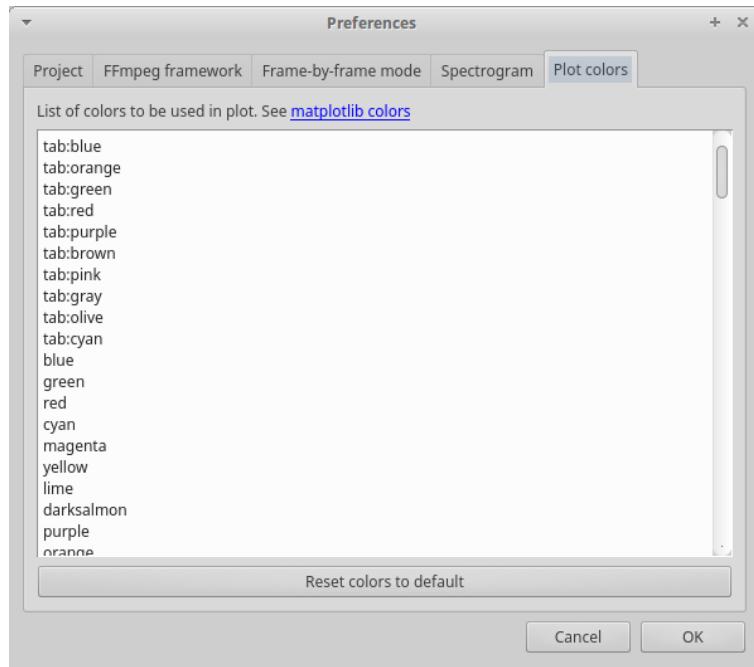
Color map

Select the color map for displaying the generated spectrogram. See [Matplotlib colormaps](#) for details.

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.



Various

Citing BORIS

If you have used BORIS for publications, please cite:

Olivier Friard and Marco Gamba. (2016) BORIS: a free, versatile open-source event-logging software for video/audio coding and live observations. *Methods in Ecology and Evolution*, 7(11), 1324–1330
DOI: 10.1111/2041-210X.12584

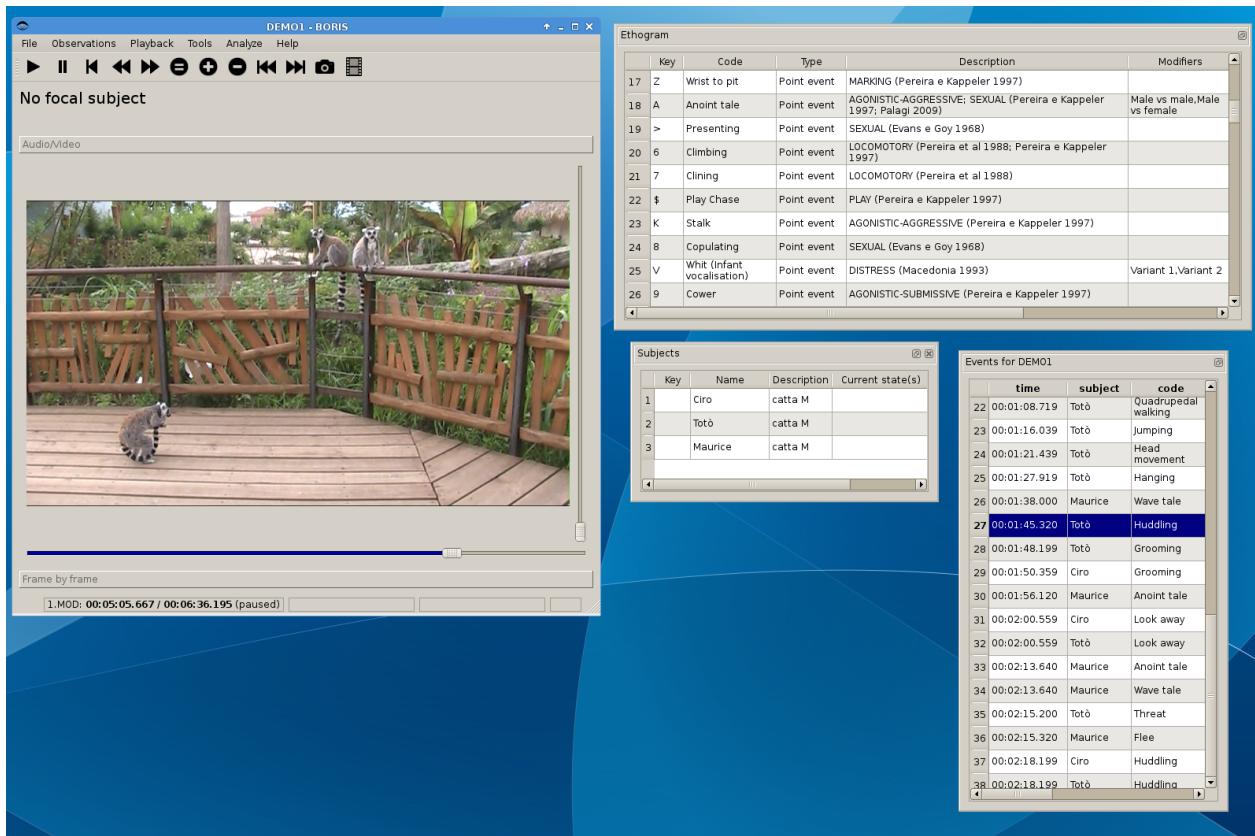
Bug reports and features request

Please send bug reports and features request using the BORIS GitHub repository (<https://github.com/olivierfriard/BORIS>) or by e-mail in english, french or italian (see web site <http://www.boris.unito.it/pages/about.html>).

In case of bug report please verify that you are using the last version of BORIS and indicate your operating system, its version and the CPU architecture (32/64 bits). You may also include the BORIS project that gave you an error. Any information you will provide will not be disclosed to any third party.

Docking

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



Acknowledgement

The authors would like to acknowledge Sergio Castellano, Valentina Matteucci and Laura Ozella for their precious help.

Valid keys for triggering behavior

BORIS will not make difference between lower case and upper case characters

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