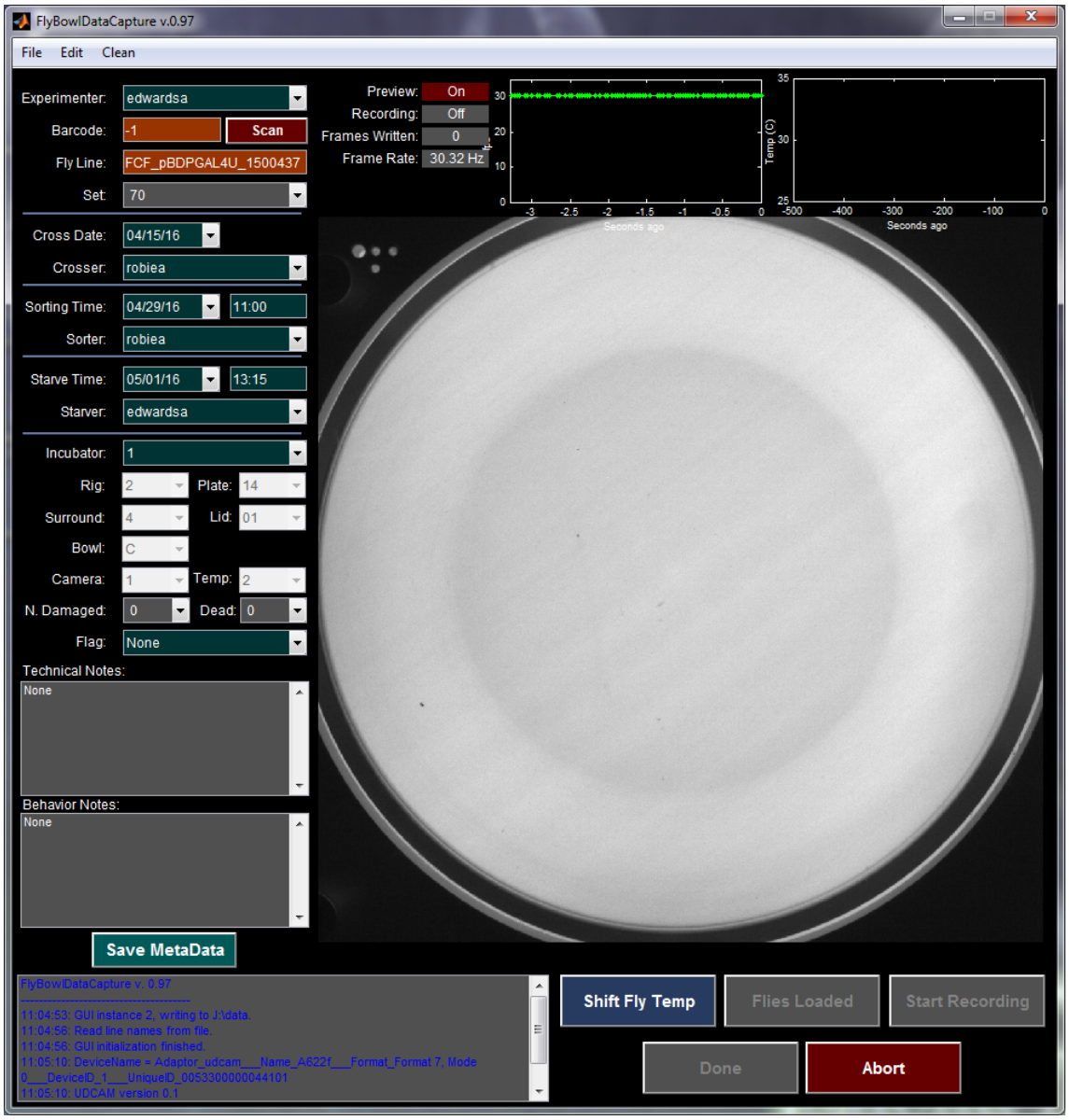
FlyBowlDataCapture Program

FlyBowlDataCapture

Overview

FlyBowlDataCapture is a MATLAB program for collecting video and experimental metadata for the FlyBowl Olympiad assay. The user selects which devices to record from, enters metadata about the flies in the left panel of the GUI, clicks to record the timestamps of events, and finally clicks to start recording. A preview of the frames recorded, the frame rate, and temperature stream is shown after the devices are initialized. The program creates a directory for each experiment, and the video, video diagnostics, video log, temperature log, experiment log, and metadata are written to this directory.

Program Usage



Run Command

To begin, run `FlyBowlDataCapture` in MATLAB in the FlyBowlDataCapture directory containing the relevant code.

GUI Initialization

The GUI will first prompt the user for the parameters file. Choose the parameters file tailored to your experiment. See #Parameters for information on how to create the parameters file.

The GUI will then appear. The layout has been optimized to have two GUIs displayed in a 1080x1920 monitor in portrait mode. Immediately, a log file will be created in the temporary data directory TmpOutputDirectory with the name "TmpLog_<datetime>.txt", for example `TmpLog_20100927T094126.txt`.

A few parameters, such as the OutputDirectory, depend on which instance of the FlyBowlDataCapture program this is. For instance, we will usually be running two instances per computer, and will want to write to two separate disks. To allow multiple instance of MATLAB and FlyBowlDataCapture to share resources and communicate, there are various semaphore files created. When the GUI is initialized, a semaphore file is created to communicate that a GUI instance is claimed. If this is the first GUI initialized, it will create a file `.GUIInstances/GUIInstance_1.mat` or in general `.GUIInstances/GUIInstance_<instance>.mat`

Select Camera

In the left panel, the user can also select which devices to record from. The **Device ID** is used to select which camera to record from, the number corresponding to the order that the computer discovers them. Once the camera device is selected, the camera can be initialized by clicking **Initialize Camera**, after which the video preview axes will be updated with a stream from the camera. This preview is of the raw video input from the camera. The same camera device **cannot** be selected in multiple GUIs running simultaneously on the same computer. This is ensured using another semaphore file.

If you plug in cameras after the GUI has started, you can attempt to detect new cameras by selecting the "Edit -> Detect Cameras" menu choice.

When the camera is set, a semaphore file `.DetectCameras/IsCameraRunning_<AdaptorName>_<DeviceID>.mat` is created. This file contains the output of `imaqhwinfo` because calling this command in one MATLAB will freeze the preview in another MATLAB. When one camera is already running, when `imaqhwinfo` needs to be called, we instead load it from this file.

We could not prevent all calls to `imaqhwinfo`. When the camera is set in a second instance of MATLAB after it is set in another instance, the first instance's preview will freeze temporarily, but should be restarted automatically.

Select Temperature Probe Channel

The **Temp Probe ID** is used to select which of the allowed channels of the Pico temperature probe to record from. Once the temperature channel is selected, the temperature probe can be initialized by clicking **Init Temp Probe**, after which the recorded temperature will be shown in the top right plot.

Because only one MATLAB instance can read from the Pico temperature probe at a time, if multiple MATLAB instances or running then one instance will be the master. It will record the last temperature readings for each channel to a file, and all other instances will read from these files. To indicate that there is a master instance controlling the temperature probe, a semaphore file `.TempRecordData/IsMaster.mat` is created. The temperature for each channel is recorded to the file `.TempRecordData/Channel_<Channel>` (e.g. `TempRecordData/Channel_01`.)

Experiment Events

To begin recording, three buttons must be pushed in order. First, the time that the flies are brought into the hot experiment room must be recorded by clicking the **Shift Fly Temp** button. Next, the time that the flies are put into the FlyBowl must be recorded by clicking the **Flies Loaded** button. Finally, the **Start Recording** button must be pushed to start recording data. The remaining record time is shown in the "Done" button.

Metadata Entry

The panel on the left allows the user to input experimental metadata manually or by using a barcode scanner. Metadata can be modified before, while, and after video is recorded. See #Metadata Fields for a description of each field. The controls will be blue when the control has not yet been used to set the metadata value, and the default value is used. The default value is chosen intelligently, usually using the data entered in the previous experiment. They will turn gray once data has successfully been entered. They will be orange if there is an error in data entry, for instance if the entered starvation time is before the sorting time.

Barcode Entry

To enter metadata using a barcode scanner, the user clicks the **Scan** button and uses a barcode scanner to scan the barcode associated with the current experiment. The metadata fields are filled automatically.

Manual Entry

Fly Line Edit Box

The **Fly Line** edit box always begins orange, as the user should enter this piece of data in every experiment. It is an auto-complete edit box: when the user begins typing in the line name, a drop-down menu will appear showing all possible valid line name completions, every line name that begins with the characters before the cursor. If no line names begin with the entered prefix, then all line names will be shown in the drop-down menu. The up and down arrows can be used to navigate the list, and hitting Enter will select the highlighted choice.

Save Metadata

Clicking the "Save MetaData" button will output the current metadata to the metadata file. Metadata is also saved when "Start Recording" is clicked, and when the "Done" button is clicked.

Saving metadata involves first creating a name for the experiment, with the template `<FlyLine>_<Effector>_Rig<Rig>Plate<Plate>Bowl<Bowl>_<StartRecordingTime>`

Example:

`GMR_22D03_AE_01_TrpA_Rig1Plate10BowlD_20110608T103942.`

If an experiment name is created (Save MetaData pressed) before recording has begun, the name will be `<FlyLine>_<Effector>_Rig<Rig>Plate<Plate>Bowl<Bowl>_notstarted_<GUIInitializationTime>.`

Example:

`GMR_22D03_AE_01_TrpA_Rig1Plate10BowlD_notstarted_20110608T103942.`

The Experiment Directory will be created with the experiment name in the OutputDirectory. If this is not the first time metadata has been saved, the old experiment directory will be moved to the new directory name. After the experiment is finished, all relevant data for this experiment should be within in this directory. When metadata is saved, the log file will be immediately placed in the experiment directory and renamed according to the LogFileName parameter ("Log.txt" by default). A metadata XML file will also be saved in the experiment directory with the name set by the MetaDataFileName parameter ("Metadata.xml" by default).

Experiment Start

When recording is started, besides files created while saving metadata, two files will be opened and placed in the TmpOutputDirectory to record streamed experiment data. These files are placed in the temporary directory and given temporary names so that the "Save MetaData" button can remain active during recording, and the experiment name can be changed.

Temporary Video File

The program will begin logging all collected video frames to the temporary video file in TmpOutputDirectory with the name template FBDC_movie_<StartRecordingTime>_<RandomNumber>.<MovieType> where <StartRecordingTime> is the time when the experiment started, <RandomNumber> is a random number associated with the current experiment, and <MovieType> is the extension for the type of video recorded.

Example:

`FBDC_movie_20100925T124453_9571.ufmf`
When recording is finished, this file will be placed within the experiment directory, and renamed `movie.<MovieType>`.

Temporary Temperature File

Also when recording is started, the program will begin logging the temperature stream to a **temporary temperature file** in TmpOutputDirectory with the name template `FBDC_temperature_<StartRecordingTime>_<RandomNumber>.txt`. The temperature will be recorded every TempProbePeriod seconds. The format of this file will be one line per recording. Each line consists of the timestamp, a comma, then the temperature in degrees Celsius.

Precon Temperature and Humidity Reading

At the start of the experiment, NPreconSamples samples are taken of the temperature and humidity recorded by the Precon sensor. The serial port to be recorded from is set by the PreconSensorSerialPort parameter. The average readings are stored in the metadata file. Again, to prevent multiple Matlabs from requiring the resource simultaneously, a semaphore file `.PreconRecordData/InUse_<SerialPort>.txt` exists while samples are being recorded. If this instance finds this file, then it waits and tries to read later. In addition, to avoid many readings from the sensor, we save the average readings in `.PreconRecordData/PreconTempHumid_<SerialPort>.txt`. This file is read from instead of the sensor if it is less than 30 seconds old.

Abort Experiment

An experiment can be canceled at any time by clicking the **Abort** button. Data is *not* automatically deleted if an experiment is canceled. The experiment directory will remain. If currently recording, the program will stop recording immediately, and attempt to move the temporary video and temperature files to the experiment. This may take a few seconds, and occasionally may fail to rename because of timing issues between multiple threads. A file named "ABORTED" will be created in the experiment directory.

Status Log

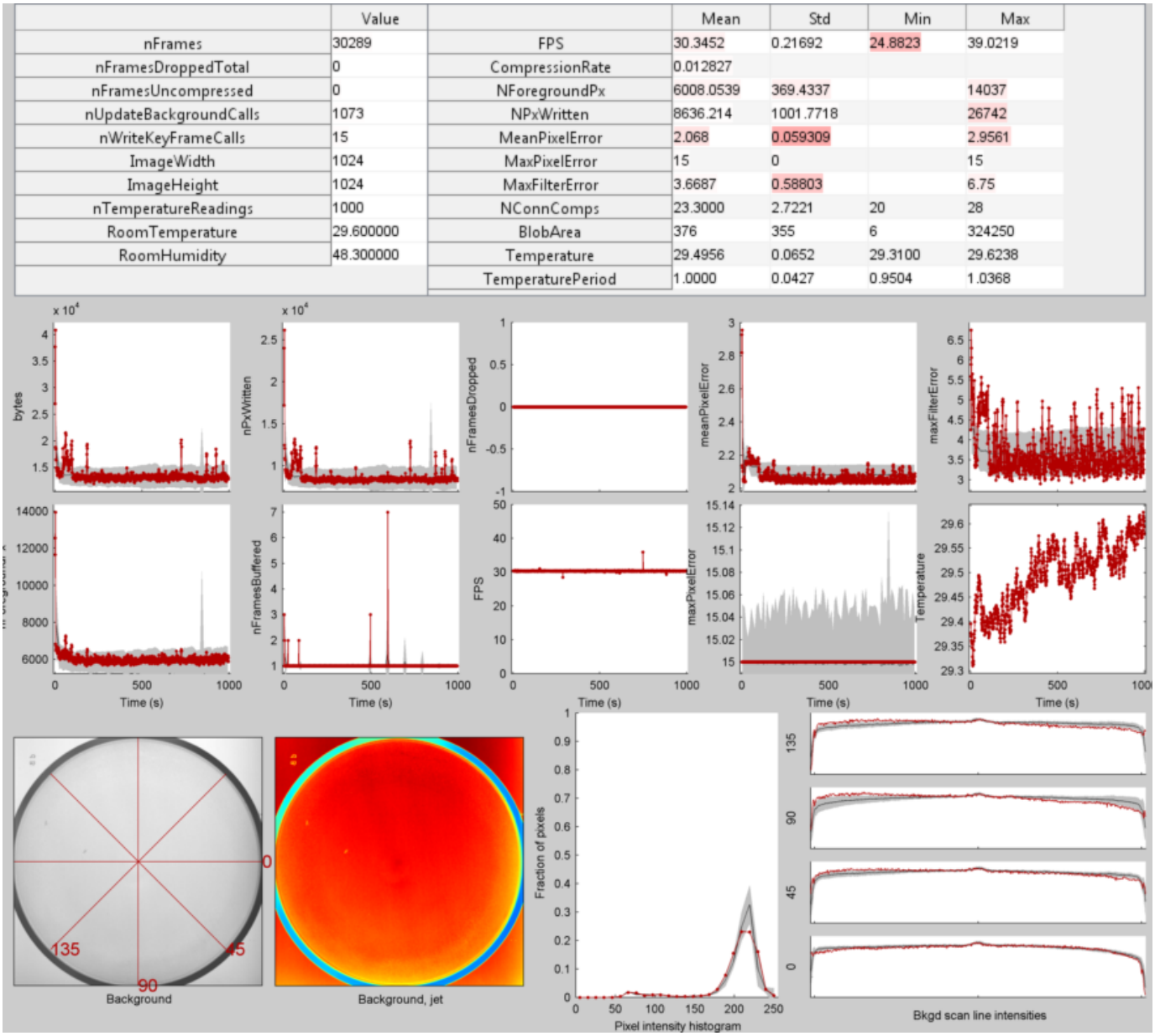
The log in the bottom left corner shows warning messages and status updates. Check here for information about errors caught during the experiment. The same information is written to the experiment log file. The color of the text in the log will be green for the first GUI opened, and blue for the second GUI opened.

Experiment Completion

Once an experiment is finished, click the **Done** button to finalize all entered and recorded data. Metadata will be saved one final time.

QuickStats

At the end of the experiment, two figures will pop up. The QuickStats figure contains diagnostic measurements of the video and temperature data collected.



The tables in the top of the figure contain statistics about data collection. Each cell is shaded red to indicate how different the measured value is from the standard set of values stored in a file called

`Quickstats_Stats_<timestamp>.mat. (ex: QuickStats_Stats_20110914T022555.mat).`

This file is specified in the #Parameters file.

The top left table contains the following scalar statistics of data capture:

- nFrames: The number of frames of video collected. This should be around 30,000 for 30 fps * 1000 s.
- nFramesDroppedTotal: The number of frames that were dropped because either compression or writing the video to disk was too slow. This should be 0.
- nFramesUncompressed: The number of frames that were written to file uncompressed because compression or writing is too slow, and

the frame memory buffer is getting full. This should be 0.
- nUpdateBackgroundCalls: The number of times the UFMF background model was updated. This should happen once a second during most of the video (more often at the very start), so it should be slightly more than 1,000.
- nWriteKeyFrameCalls: Number of times the UFMF background keyframes were written. This is currently set to happen once every 100 seconds in most of the movie, but more frequently in the beginning of the move; the number should be around 15.
- ImageWidth, ImageHeight: Size of the image.
- nTemperatureReadings: Number of recordings from the Pico temperature probe. This is currently set to read once a second, so this number should be around 1000.

The top-right table contains the mean, standard deviation, minimum, and maximum of several streams of measurements collected. The following measurement streams are represented:

- FPS: Frame rate of the camera, computed as the inverse of the time between each pair of consecutive frames.
- CompressionRate: The size of the compressed frame divided by the hypothetical size had it not been compressed, with overhead for headers, keyframes amortized over the entire video.
- NPxWritten: Number of foreground pixels written per frame – the summed area of all the foreground boxes stored.
- MeanPixelError: The average absolute error between each compressed frame and true frame.
- MaxPixelError: The maximum error between each compressed frame and true frame.
- MaxFilterError: We filter the absolute error image with a 10x10 uniform averaging box to determine if several pixels near each other have high error (indicative of a fly being missed, change in background model, as opposed to imaging noise). We compute the maximum filtered error per frame.
- NConnComps: The number of connected components of foreground, based on a few sampled frames of background subtraction.
- BlobArea: The average area of the connected components of foreground in a frame.
- Temperature: The temperature measured by the Pico probe.
- TemperaturePeriod: Time in between temperature readings.

In the next two rows of plots, some of these time series of data are shown.

The last row of plots illustrates the background model. The first two plots are of the median of the background key frames, first in gray then in jet colormaps. The next plot shows a histogram of the pixel intensities in this background image. The plots on the right show the pixel intensity across the four scan lines illustrated in the left plot.

Starting a New Experiment

To start a new experiment in the same GUI, use the **File -> New Experiment** menu choice.

The output files from an experiment are described in #Output Data.

Semaphores

As discussed above, there are four different resources that must be shared and semaphore files to allow multiple Matlabs to communicate. The semaphore directories are

- .DetectCamerasData
- .TempRecordData
- .GUIInstances
- .PreconRecordData

If the GUI crashes for some reason, some of the semaphore files may not be correctly cleared. The user can list all existing semaphores with the command `ListSemaphores`. The user can delete all semaphores with the command `ClearSemaphores`. Clearing these while another GUI is running will potentially cause bad problems.

Parameters

The following parameters must be set in the parameters file input during #GUI Initialization. Example values are shown in brackets. The parameter file is always copied to the experiment directory for future reference.

- `OutputDirectory`: List of directories to which experiments are saved. Which element of this list is selected based on which instance of the GUI this is [`D:\data, J:\data`].
- `TmpOutputDirectory`: List of directories to store temporary data within. Which element of this list is selected based on which instance of the GUI this is. These should be on the same disk as the corresponding `OutputDirectory` to avoid having to rewrite data when renaming from temporary to permanent file names [`D:\tmpdata,J:\tmpdata`].
- `HardDriveName`: List of hard drive names, corresponding to the ordered OutputDirectory list [`bransonlab-ww1:Internal_D`].
- `Imaq_Adaptor`: Camera adaptor name [`udcam`].
- `Imaq_VideoFormat`: Video format. For gdcam, udcam, this should be `Format 7, Mode 0`. For dcam, it should be `F7_Y8_1280x1024 [Format 7, Mode 0]`
- `FileType`: Extension of video file, such as fmf, ufmf, avi [`ufmf`].
- `RecordTime`: Number of seconds of video to record [`1000`].

- `ColormapPreview`: Whether to show the video preview in jet colormap (1) or grayscale (0). [0]
- `Imaq_Shutter`: Shutter period [`100`].
- `Imaq_Gain`: Gain [`170`].
- `Imaq_Brightness`: Brightness [`25`].
- `MovieFilePrefix`: Prefix of movie file to store [`movie`].
- `Assay_Experimenters`: List of possible experimenters [`hirokawaj,robiea`].
- `NFlies`: Expected number of flies in the bowl [`20`].
- `Rearing_IncubatorIDs`: List of possible Incubator ID values. [`1,2,3,4`]
- `PreAssayHandling_CrossDate_Range`: Range of Cross Dates to show – minimum and maximum number of days before experiment day [`14,22`].
- `PreAssayHandling_SortingDate_Range`: Range of Sorting Dates to show – minimum and maximum number of days before experiment day [`1,8`].
- `PreAssayHandling_SortingHour_Range`: Range of Sorting Times to show – minimum and maximum in hour of the day [`8,19`].
- `PreAssayHandling_SortingDate_Interval`: Interval of Sorting Times to show in hours [`0.25`].
- `PreAssayHandling_SortingHandlers`: List of possible Sorting Handler values [`hirokawaj,robiea`].
- `PreAssayHandling_StarvationDate_Range`: Range of Starvation Dates to show – minimum and maximum number of days before experiment day [`0,1`].
- `PreAssayHandling_StarvationHour_Range`: Range of Starvation Times to show – minimum and maximum in hour of the day [`8,19`].
- `PreAssayHandling_StarvationDate_Interval`: Interval of Starvation Times to show in hours [`0.25`].
- `PreAssayHandling_StarvationHandlers`: List of possible Starvation Handler values [`hirokawaj,robiea`].
- `Assay_Rooms`: List of possible room identifiers names [`2C.310.1, 2C.310.2`].
- `Assay_Room`: Assay room number [`2C.310.1`].
- `Assay_Rigs`: List of possible Rig numbers [`1,2`].
- `Assay_Plates`: List of possible Plate numbers [`10,11,12`].
- `Assay_VisualSurrounds`: List of possible visual surround numbers [`1,2,3,4`].
- `Assay_Lids`: List of possible Lid numbers [`01,02,03`].
- `Assay_Bowls`: List of possible Bowl names [`A,B,C,D`].
- `RedoFlags`: List of possible Redo flags [`None,Rearing problem,Flies look sick,See behavioral notes,See technical notes`].
- `ReviewFlags`: List of possible Review flags [`None,Rearing problem,Flies look sick,See behavioral notes,See technical notes`].
- `Imaq_DeviceName`: Camera name [`A622f`].
- `Imaq_ROIPosition`: Video ROI Position: xmin, ymin, width, height [`128,0,1024,1024`].
- `Imaq_FrameRate`: Frame rate we expect [`30.4`].
- `Imaq_MaxFrameRate`: Maximum frame rate we expect to get [`31`].
- `PreviewUpdatePeriod`: Minimum time in between preview window updates [`0`].
- `gdcamPreviewFrameInterval`: For gdcam, number of frames between updates [`2`].
- `MetaDataFileName`: Name of file to store metadata to [`Metadata.xml`].
- `LogFileName`: Name of file to store log to [`Log.txt`].
- `MetaData_AssayName`: Assay name [`FlyBowl`].
- `MetaData_RearingProtocols`: Names of the rearing protocol files. Length of this list should be equal to length of list in `Rearing_IncubatorIDs` [`RP_Olympiad_v008p2.xls`].
- `MetaData_HandlingProtocols`: Name of handling protocol file [`HP_flybowl_v004p0.xls`].
- `MetaData_ExpProtocols`: Name of experiment protocol file [`EP_flybowl_v010p0.xls`].
- `MetaData_Effector`: Name of effector [`UAS_dTrpA1_2_0002`].
- `MetaData_Gender`: Gender of flies, one of {m,f,b} [b].
- `MetaData_RoomTemperatureSetPoint`: Set point temperature for the experiment [`29`].
- `MetaData_RoomHumiditySetPoint`: Set point humidity for the experiment [`60`].
- `FrameRatePlotYLim`: Limits of y-axis for the frame rate plot [`0,35`].
- `TempPlotYLim`: Limits of y-axis for temperature plot [`25,35`].
- `DoQuerySage`: Whether to try to connect over the network to Sage to query line names, 1 to query, 0 not to query [`0`].
- `TempProbePeriod`: Time in between temperature readings from the Pico temperature probe [`5`].
- `TempProbeChannels`: Which channels of the Pico probe can be recorded from [`1,2`].
- `TempProbeTypes`: Thermocouple types for each channel of the Pico probe. This should correspond to the ordered `TempProbeChannels` list [`T,T`].
- `TempProbeReject60Hz`: If 1, Pico temperature probe will reject 60 Hz, otherwise will reject 50 Hz [`0`].
- `DoRecordTemp`: Whether to record a temperature stream from the Pico probe. 1 means record, 0 means do not. [`1`].
- `NPreconSamples`: Number of temperature and humidity samples to read from the Precon probe to compute average start temperature and humidity. Set to 0 if no recording should be taken [5].
- `PreconSensorSerialPort`: Name of serial port to connect to [`COM1`].
- `ExtraLineNames`: Line Names to add to the list read from Sage [`DL-wildtype,pBDPGAL4U`].
- `UFMFLogFileName`: Log file for writing ufmf [`ufmf_log.txt`].
- `UFMFStatFileName`: Name of file to write UFMF diagnostic stats to [`ufmf_diagnostics.txt`]
- `UFMFPrintStats`: Whether to compute UFMF diagnostic statistics. `1` means compute, `0` means don't compute [`1`].
- `UFMFStatStreamPrintFreq` Number of frames between outputting per-frame compression statistics: 0 means don't print, 1 means every frame [`30`].
- `UFMFStatComputeFrameErrorFreq`: Number of frames between computing statistics of compression error. 0 means don't compute, 1

means every frame [30].
- UFMFStatPrintTimings: Whether to print information about the time each part of the computation takes. 1 means do print, 0 means don't print [1].
- UFMFMaxFracFgCompress: Maximum fraction of pixels that can be foreground to try compressing frame. Otherwise, the raw frame will be output [.2].
- UFMFMaxBGNFrames: Approximate number of frames the background model should be based on. This means the background model is based on a window of size UFMFMaxBGNFrames [100]
- UFMFBGUpdatePeriod: Number of seconds between updates to the background model [1].
- UFMFBGKeyFramePeriod: Number of seconds between spitting out a new background model [100].
- UFMFMaxBoxLength: Maximum box width and height stored during compression [5].
- UFMFBackSubThresh: Threshold for background subtraction [15].
- UFMFNFramesInit: First nFramesInit we always update the background model [100].
- UFMFBGKeyFramePeriodInit: While ramping up the background model, use this list of keyframe periods. There can be at most 5 numbers in this list [1,10,25,50,75].
- DoRotatePreviewImage: The first number describes whether to flip the x-axis, the second whether to flip the y-axis. [(1,A,1),(1,B,1)]
- QuickStatsStatsFileName: Name of mat file containing Quickstats stats [QuickStats_Stats_20110914T022555.mat]
- ScanLineYLim: Y-axis limits for scan lines [60,250]
- MinFliesLoadedTime: Minimum time allowed between loading flies and starting recording [25].
- MaxFliesLoadedTime: Maximum time allowed between loading flies and starting recording [225].
- PreAssayHandling_CrossHandlers: List of possible Cross Handler values [hirokawaj,robiea].
- PreAssayHandling_FlipUsed: Indicates which flip was used [1].
- WishListRange: Range of possible wish_list values [-1,125].
- DoSyncBarcode: Indicates whether to connect to outside database to sync barcode with line name [1].
- ScreenType: Indicates which type of screen this experiment is associates with [primary].
- ScreenReason: Indicates screen reason [standard].
- ControlLineNames: Control fly line names [pBDPGAL4U].
- flip_days: Number of days between cross and flip [4].
- CheckBarCode: Whether to check that the barcode >= 0 [0,1].
- CoupleCameraTempProbeStart: Whether to couple camera and temperature probe start [0,1].

Metadata Fields

The following metadata must be entered using the left panel in the GUI:

- **Experimenter**: LDAP name of person conducting the experiment. Possible values are set by Assay_Experimenters parameter.
- **Fly Line**: Name of fly line registered in Sage.
- **Incubator ID**: ID of the incubator the flies were raised in. Possible values for this are set by the Rearing_IncubatorIDs parameter.
- **Cross Date**: The date the cross was performed on to generate these flies. Possible values are set by the range of dates set in the PreAssayHandling_CrossDate_Range parameter.
- **Sorting Time**: Date and time that the files were sorted on. The range of possible dates is set by PreAssayHandling_SortingDate_Range.
- **Sorter**: LDAP name of person who sorted the files. Possible values are set by the PreAssayHandling_SortingHandlers parameter.
- **Starvation Time**: Date and time that the flies were moved to starvation material. The range of possible dates is set by PreAssayHandling_StarvationDate_Range.
- **Starver**: LDAP name of person who moved the files to starvation material. Possible values are set by the PreAssayHandling_StarvationHandlers parameter.
- **Rig**: ID of the rig (which cart) the current bowl is on. Possible values are set by the Assay_Rigs parameter.
- **Plate**: ID of the plate the current bowl is in. Possible values are set by the Assay_Plates parameter.
- **Bowl**: ID of current bowl. Possible values are set by the Assay_Bowls parameter.
- **Device ID**: Which camera device we should record from. Possible values are autodetected using imaqhwinfo.
- **Temp Probe ID**: Channel of the Pico temperature probe to record the temperature stream from. Possible values are set by the TempProbeChannels parameter.
- **N. Dead Flies**: Number of flies that are dead in the bowl.
- **Redo Flag**: Reason the experiment should be redone.
- **Review Flag**: Reason the experiment should be reviewed.
- **Technical Notes**: Notes from the experimenter about possible technical problems during the experiment.
- **Behavior Notes**: Notes from the experimenter about possible behavior problems or observations during the experiment.

Output Data

All data is output to the Experiment Directory within the OutputDirectory. The following files will be in this directory:

- #QuickStats: QuickStats.png and QuickStats.txt
- #Log File: Log.txt

- #Metadata File: Metadata.xml
- #UFMF Video: movie.ufmf
- #UFMF Diagnostics File: ufmf_diagnostics.txt
- #UFMF Log File: ufmf_log.txt
- FlyBowlDataCapture parameters file
- SUCCESS/FAILED file

Log File

The log file `Log.txt` contains a record of all events during the experiment, as well as any errors or warnings. It is the same as the information output to the status window. Here is an example log file:

```
FlyBowlDataCapture v. 0.96
-------------------------------------
14:37:05: GUI instance 1, writing to D:\data.
14:37:07: Read line names from file.
14:37:07: Another camera is in operation, so reading adaptorinfo from
.DetectCamerasData\IsCameraRunning_udcam_0.mat. If this seems wrong, exit
all
FlyBowlDataCapture GUIs and run "CleanSemaphores"
14:37:07: GUI initialization finished.
14:37:09: DeviceName = Adaptor_udcam___Name_A622f___Format_Format 7, Mode
0___DeviceID_1___UniqueID_005330006FC74001
14:37:09: UDCAM version 0.1
14:37:10: Video preview started.
14:37:11: Initialized temperature probe timer
14:37:35: Shifted fly temperature.
14:38:27: Flies loaded.
14:39:48: Read start temperature 29.800000 and humidity 47.600000 from
Precon sensor
14:39:48: Recorded start temp 29.800000, humid 47.600000
14:39:48: Experiment name initialized to
GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946
14:39:48: Creating experiment directory
D:\data\GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946
14:39:48: Saved MetaData to file
D:\data\GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\Metadata.xml.
14:39:48: Opening D:\tmpdata\FBDC_temperature_20120217T143946832.txt to
write temperature
stream
14:39:46: Started recording to file
D:\tmpdata\FBDC_movie_20120217T143946832.ufmf.
14:39:52: Created temperature file
D:\tmpdata\FBDC_temperature_20120217T143946832.txt
14:56:32: Renaming experiment...
14:56:32: Renaming movie file from
D:\tmpdata\FBDC_movie_20120217T143946832.ufmf to
D:\data\GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\movie.ufmf
14:56:32: Renaming temperature stream file from
D:\tmpdata\FBDC_temperature_20120217T143946832.txt to
D:\data\GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\temperature.
txt
14:56:32: Renaming UFMF log file from
D:\tmpdata\FBDC_UFMF_Log_20120217T143709736.txt to
D:\data\GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\ufmf_log.txt
```

```
14:56:32: Renaming UFMF stats file from
D:\tmpdata\FBDC_UFMF_Diagnostics_20120217T143709736.txt to
D:\data\GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\ufmf_diagnos
tics.txt
14:56:33: Finished recording. Video file moved from
D:\tmpdata\FBDC_movie_20120217T143946832.ufmf to
D:\data\GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\movie.ufmf.
14:56:57: Copying metadata file to backup
D:\data\GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\Metadata.xml
.bak
14:56:57: Saved MetaData to file
D:\data\GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\Metadata.xml.
15:00:07: Data read from barcode:
Line_Name = 'GMR_22D03_AE_01'
Date_Crossed = '20120126T000000'
Effector = 'UAS_dTrpA1_2_0002'
Set_Number = '90'
Handler_Cross = 'mcmahonj'
Handler_Sorting = 'hibbardk'
Sorting_DateTime = '20120213T142000'
Handler_Starvation = 'hirokawaj'
Starvation_DateTime = '20120216T143000'
15:00:10: Renaming experiment...
15:00:10: Experiment renamed from
GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946
to GMR_22D03_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946
15:00:10: Trying to rename experiment directory from
D:\data\GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946 to
D:\data\GMR_22D03_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946
15:00:10: Renaming completed by changing directory name.
15:00:10: Renamed
D:\data\GMR_22A02_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\movie.ufmf ->
D:\data\GMR_22D03_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\movie.ufmf.
15:00:11: Copying metadata file to backup
D:\data\GMR_22D03_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\Metadata.xml
.bak
```

```
15:00:11: Saved MetaData to file
D:\data\GMR_22D03_AE_01_TrpA_Rig1Plate15BowlB_20120217T143946\Metadata.xml.
```

The metadata file `Metadata.xml` is an XML file containing all the metadata about the current experiment. It contains the following information:

- `experiment`
  - `assay`: Name of assay, from `MetaData_AssayName` parameter.
  - `exp_datetime`: Date and time recording was started.
  - `experimenter`: LDAP name of person conducting the experiment, set by the experimenter metadata field.
  - `protocol`: Name of protocol file, from `MetaData_ExpProtocols` parameter.
  - `screen_type`: Screen type identifier, from `MetaData_ExpProtocols` parameter.
  - `screen_reason`: Screen reason descriptor, from `MetaData_ExpProtocols` parameter.
  - `session`
    - `id`
    - `apparatus`
      - `apparatus_id`: Unique name specifying all pieces of the apparatus
      - `room`: Room number.
      - `rig`: Name of rig, set by the rig metadata field.
      - `top_plate`: Top plate number, set by the top plate metadata field.
      - `visual_surround`: Visual surround number, set by the visual surround metadata field.
      - `bowl`: Name of the bowl, set by the bowl metadata field.
      - `camera`: Unique ID for this camera, queried from device.
      - `computer`: Name of computer data was recorded on.
      - `harddrive`: Name of harddrive the data was stored on, set by GUI instance and `HardDriveName` parameter.
    - `flies`
      - `line`: Line name, set by fly line metadata field.
      - `effector`: Name of effector, set by `MetaData_Effector` parameter.
      - `gender`: Gender of flies, set by `MetaData_Gender` parameter. One of {m,f,b}.
      - `cross_date`: Date the cross for these flies was performed, set by the cross date metadata field.
      - `flip_date`: Date the parent flies were removed from the cross vials, .
      - `hours_starved`: Hours the flies have been starved, computed from Starvation Date and Time controls.
      - `num_flies`: Number of flies
      - `rearing`
        - `rearing_protocol`: Name of rearing protocol file, set by Incubator ID and `MetaData_RearingProtocols` parameter.
        - `rearing_incubator`: ID of the incubator the flies were raised in, set by Incubator ID control.
      - `handling`
        - `handling_protocol`: Name of handling protocol file, set by .
        - `handler_cross`: ID of person who sorted the flies, set by the Crosser metadata field.
        - `handler_sorting`: ID of person who sorted the flies, set by the Sorter metadata field.
        - `hours_sorted`: Hours between recording start and sorting time, computed from the sorting date and experiment date-time metadata fields.
        - `datetime_sorting`: Date and time the flies were sorted, from sorting date and time metadata fields.
        - `handler_starvation`: ID of person who flipped flies to starvation media, set by Starver metadata field.
        - `datetime_starvation`: Date and time the flies were moved to starvation material, computed from sorting date and experiment date-time metadata fields.
        - `seconds_shiftflytemp`: Seconds between bringing the flies into the hot room and starting recording. Set by clicking the Shift Fly Temp button.
        - `seconds_fliesloaded`: Seconds between loading the flies into the bowl and starting recording. Set by clicking the Flies Loaded button.
        - `num_flies_dead`: Number of dead flies in the bowl, set by the Dead metadata field.
        - `num_flies_damaged`: Number of damaged flies in the bowl, set by the N. Damaged metadata field.
  - `environment`
    - `temperature`: Temperature recorded from the Precon sensor at the start of the experiment, in degrees Celsius.
    - `humidity`: Relative humidity recorded from the Precon sensor at the start of the experiment, in percent.
  - `note_behavioral`: Observations about the flies' behavior, entered by experimenter during the experiment.
  - `note_technical`: Observations about technical aspects of the experiment, entered by the experimenter during the experiment.
  - `notes_keyword`: Field containing only legal keywords in SAGE
  - `flag_review`: Whether the experimenter flagged the experiment to be reviewed (1) or not (0).
  - `flag_redo`: Whether the experimenter flagged the experiment to be redone (1) or not (0).
  - `flag_aborted`: Whether the experiment was aborted early (1) or not (0).
  - `flag_legacy`: Wether the experiment took place before Metadata XML standardization (1) or not (0).

Here is an example metadata file:

```
<?xml version="1.0"?>
<experiment assay="FlyBowl" exp_datetime="20120217T143946"
experimenter="hirokawaj" protocol="EP_flybowl_v010p0.xls"
screen_type="primary" screen_reason="standard" data_capture_version="0.96"
>
  <session id="1">
    <apparatus
apparatus_id="Rig1__Plate15__Lid06__BowlB__Camera005330006FC74001__Compute
rbransonlab-ww1__HardDriveInternal_D" room="2C.310.1" rig="1" plate="15"
top_plate="06" visual_surround="4" bowl="B" camera="005330006FC74001"
computer="bransonlab-ww1" harddrive="Internal_D"/>
    <flies line="GMR_22D03_AE_01" effector="UAS_dTrpA1_2_0002" gender="b"
cross_date="20120126T000000" flip_date="20120130T000000"
hours_starved="24.163009" cross_barcode="41566" flip_used="1"
wish_list="90" robot_stock_copy="unknown" num_flies="0">
      <rearing rearing_protocol="RP_Olympiad_v008p2.xls"
rearing_incubator="2" />
      <handling handling_protocol="HP_flybowl_v004p0.xls"
handler_cross="mcmahonj" handler_sorting="hibbardk"
hours_sorted="96.329676" datetime_sorting="20120213T142000"
handler_starvation="hirokawaj" datetime_starvation="20120216T143000"
seconds_shiftflytemp="131.591994" seconds_fliesloaded="79.108996"
num_flies_dead="0" num_flies_damaged="0" />
    </flies>
  </session>
  <environment temperature="29.800000" humidity="47.600000" />
  <notes_behavioral>increased courtship</notes_behavioral>
  <notes_technical>None</notes_technical>
  <notes_keyword></notes_keyword>
  <flag_review>0</flag_review>
  <flag_redo>0</flag_redo>
  <flag_aborted>0</flag_aborted>
  <flag_legacy>0</flag_legacy>
</experiment>
```

UFMF Video

The primary piece of data output is the video record. The format is described at UFMF File Description. See udcam for information on how udcam creates this video. The video can be previewed with the `playfmf` or `showufmf` functions in JCtrax.

UFMF Diagnostics File

udcam outputs diagnostics about the video compression performed to the file `ufmf_diagnostics.txt`. See udcam for information about these diagnostics.

UFMF Log File

udcam outputs status, warning, and error messages to the log file `ufmf_log.txt`. See udcam for information about this file.

Effector abbreviations

UAS_dTrpA1_2_0002 = TrpA
EXT_CantonS_1101243 = CS
NoEffector_0_9999 = None
GuestEffector_0_9998 = Guest
CTRL_CSMH_1500154_0030 = CSMH