
DATA EXCHANGE USING CAMERA TRAP DATA PACKAGE

PRODUCED BY: ALAN FERIA & DR. ALEXANDER WHITE '22

ALAN214@UCSB.EDU

Table of Contents

1	Camera Trap Data Package	3
1.1	deployments.csv	3
1.1.1	deploymentID*	4
1.1.2	locationID	4
1.1.3	locationName	4
1.1.4	longitude*	4
1.1.5	latitude*	4
1.1.6	coordinateUncertainty	5
1.1.7	start*	5
1.1.8	end*	5
1.1.9	setupBy	5
1.1.10	cameraModel	6
1.1.11	cameraInterval	6
1.1.12	cameraHeight	6
1.1.13	cameraTilt	6
1.1.14	cameraHeading	6
1.1.15	detectionDistance	7
1.1.16	timestampIssues	7
1.1.17	baitUse	7
1.1.18	session	7
1.1.19	array	7
1.1.20	featureType	8
1.1.21	habitat	8
1.1.22	tags	8
1.1.23	comments	8
1.1.24	_id	8
1.2	media.csv	8
1.2.1	mediaID*	9
1.2.2	deploymentID*	9
1.2.3	sequenceID	9
1.2.4	captureMethod	9
1.2.5	timeStamp*	9
1.2.6	filePath*	10
1.2.7	fileName	10
1.2.8	fileMediatype*	10
1.2.9	exifData	10
1.2.10	favourite	10
1.2.11	comments	10

1.2.12	_id	10
1.3	Observation.csv	10
1.3.1	observationID*	11
1.3.2	deploymentID*	11
1.3.3	sequenceID*	11
1.3.4	mediaID	11
1.3.5	timestamp*	11
1.3.6	observationType*	11
1.3.7	cameraSetup	12
1.3.8	taxonID	12
1.3.9	scientificName	12
1.3.10	count	12
1.3.11	countNew	12
1.3.12	lifeStage	12
1.3.13	sex	12
1.3.14	behavior	12
1.3.15	indavidualID	13
1.3.16	classificationMethod	13
1.3.17	classificationTimestamp	13
1.3.18	classificationConfidence	13
1.3.19	comments	13
1.3.20	_id	13

About This File

This file was created for the benefit of all teachers and students wanting to use the work located on ([link to SCI github](#))

Camera Trap Data Package

Camera trapping is a well-established method of collecting field data on animal abundance, distribution, behavior, temporal activity, and space use. However, a common data exchange format is needed to effectively exchange camera trap data between infrastructures and harmonize data into large-scale wildlife datasets (BISS). Camera Trap Data Package (Camtrap DP) is a data exchange format for camera trap data managed by the Machine Observations Interest Group of Biodiversity Information Standards (TDWG). Data in Camtrap DP are organized as three related resources (CSV files): deployments, media, and observations. These Tabular Data Resources are described as resources in the datapackage.json file ([Camera Trap DP](#)). This section documentation will cover the deployments, media and observations files.

1.1 deployments.csv

This section discusses what the deployments.csv contains and what each field in the file means. The deployments.csv file is a tabular data table that contains data for each camera trap that has been deployed contains the following columns: deploymentID*, locationID, locationName, longitude*, latitude*, coordinateUncertainty, start*, end*, setupBy, cameraID, cameraModel, cameraInterval, cameraHeight, cameraTilt, cameraHeading, detectionDistance, timestampIssues, baitUse, session, array, featureType, habitat, tags, comments, _id.¹

¹According to TDWG, column names marked with an asterisk are required to have an entry, while those that don't contain an asterisk aren't required to have an entry.

1.1.1 deploymentID*

deploymentID should be a **unique identifier for each deployment** (camera trap) that has been set up, we are currently using a schema that concatenates the deployments locationName & locationID. This will be **auto filled** when executing deploymMediaObs.py

Example: UCSB02_2A

1.1.2 locationID

We have many camera locations throughout Santa Cruz Island that host up to 2 or 3 cameras. Each camera within each location has a unique ID made of a combination of letters and numbers. These unique IDs are referred to as locationID within the camera trap DP. This will be **input by user** when executing deploymMediaObs.py

Example: 2A

1.1.3 locationName

We have many camera locations throughout Santa Cruz Island that host up to 2 or 3 cameras. Each location has a unique ID, a combination of letters and numbers. These unique IDs are referred to as locationID within the camera trap DP. This will be **input by user** when executing deploymMediaObs.py

Example: UCSB02

1.1.4 longitude*

longitude should be the **longitude sampling location** in decimal degrees **using the World Geodetic System (WGS84) datum**. Longitude must have five decimal place accuracy, have a maximum of 180 degrees, and a minimum of -180 degrees. This will be **input by user** when executing deploymMediaObs.py

Example: 175.12345

1.1.5 latitude*

latitude should be the **latitude sampling location** in decimal degrees **using the World Geodetic System (WGS84) datum**. longitude must have five decimal place accuracy, have a maximum of

90 degrees, and a minimum of -90 degrees. This will be **input by user** when executing deploymMediaObs.py

Example: 23.12345

1.1.6 coordinateUncertainty

coordinateUncertainty refers to the **horizontal distance in meters from the given latitude and longitude** describing the smallest circle containing the location of the camera trap. This entry must be greater than one and is **not required by TWDG**. This will be **auto filled as empty** when executing deploymMediaObs.py

Example: 57

1.1.7 start*

Date and time when the deployment started, as an International Organization for Standardization 8601 (ISO 8601) formatted string with time zone designator YYYY-MM-DDThh:mm:ssZ or YYYY-MM-DDThh:mm:ss±hh:mm. This will be **auto filled** when executing deploymMediaObs.py

Example: 2022-03-01T22:00:00Z

1.1.8 end*

Date and time when the deployment ended, as an International Organization for Standardization 8601 (ISO 8601) formatted string with time zone designator YYYY-MM-DDThh:mm:ssZ or YYYY-MM-DDThh:mm:ss±hh:mm. This will be **auto filled** when executing deploymMediaObs.py

Example: 2020-04-01T22:00:00Z

1.1.9 setupBy

setupBy should be the name of the person who set up the camera for the deployment. This information can be found on the datasheet filled out while setting up the camera. This will be **input by user** when executing deploymMediaObs.py

Example: Alan Feria

1.1.10 cameraModel

Camera model of the camera used for the deployment. This will be **auto filled** when executing `deployMediaObs.py`

Example: HYPERFIRE 2 COVERT

1.1.11 cameraInterval

The time specified between shutter triggers when activity in the sensor will not trigger the shutter is expressed in seconds. This will be **auto filled as empty** when executing `deployMediaObs.py`

Example: 0

1.1.12 cameraHeight

Height at which the camera was deployed. Expressed in meters. This will be **auto filled as empty** when executing `deployMediaObs.py`

Example: 1.3

1.1.13 cameraTilt

The angle at which the camera trap was deployed in the vertical plane. Expressed in degrees, with -90 facing down, 0 horizontal, and 90 facing up. Will be **auto filled as empty** when executing `deployMediaObs.py`

Example: 45

1.1.14 cameraHeading

The angle at which the camera was deployed in the horizontal plane. Expressed in decimal degrees clockwise from north, with values ranging from 0 to 360: 0 = north, 90 = east, 180 = south, 270 = west. This will be **auto filled as empty** by `deployMediaObs.py`

Example: 115

1.1.15 detectionDistance

The furthest distance at which the camera trap detected a human during setup expressed in meters. This will be **auto filled as empty** when executing deployMediaObs.py

Example: 1.5

1.1.16 timestampIssues

If the camera trap timestamp was incorrectly set up or had unsolvable timestamp issues (e.g., unknown timezone, am/pm switch), this column will be marked true and otherwise false if there are no issues. This will be **user input** when executing deployMediaObs.py

Example: True

1.1.17 baitUse

Type of bait (if any) that was used for the deployment. If other, more info can be provided in the comments section of deployment....csv. This will be **auto filled as empty** when executing deployMediaObs.py

Example: canned chicken

1.1.18 session

Temporal deployment group. Common sessions are seasons (wet and dry), months, years, or other logical groupings when sampling occurred. For groupings without context, use tags. This will be **auto filled as empty** when executing deployMediaObs.py

Example: Winter 2022

1.1.19 array

Clusters or other logical groupings where sampling occurred. In our case, this would be the camera location as we host a cluster of 2-3 cameras. This will be **auto filled** when executing deployMediaObs.py

Example: UCSB04

1.1.20 featureType

Type of feature (if any) that camera deployment is associated with. If other, more info can be provided in the comments section within deployments....csv. This will be **auto filled** when executing deployMediaObs.py

Example: road bridge, overpass, water source, etc.

1.1.21 habitat

Short description of the habitat where the camera trap was set up at. This will be **user input** when executing deployMediaObs.py

Example: Grassy area with lots of trees and little sunlight.

1.1.22 tags

User-defined tags associated with the deployment as a pipe (|) separated list. This will be **auto filled as empty** when executing deployMediaObs.py

Example: NaN

1.1.23 comments

Comments or notes about the deployments. This will be **user input** when executing deployMediaObs.py

Example: Deployment was chewed up by animals.

1.1.24 _id

Unique identifier of the deployment as assigned by the data management system. This will be **auto filled as empty** when executing deployMediaObs.py

1.2 media.csv

Table with media files (images/videos) captured by the camera traps. Associated with deployments (deploymentID) and organized in sequences (sequenceID). Includes timestamp and file path.

1.2.1 mediaID*

unique ID within the package that can identify a media file (image file). This will be **auto filled** when executing deployMediaObs.py

Example: 2022_UCSB02_2B_img_00001

1.2.2 deploymentID*

Unique ID of the deployment the media file belongs to. This should be a Foreign key to deployments.deploymentID. This will be **auto filled** when executing deployMediaObs.py

Example: _UCSB02_2B

1.2.3 sequenceID

Unique ID of the sequence the media file (image) belongs to. Each image is its sequence and shall be named accordingly. This will be **auto filled** when executing deployMediaObs.py

Example: 2022_UCSB02_2B_img_00001_SEQ_1

1.2.4 camptureMethod

Short description of the setting that was applied to the camera to detect and capture images. In our case, it should be 'Motion detection' but can vary if the camera trap was set up incorrectly. This will be **auto filled** when executing deployMediaObs.py

Example: Motion detection

1.2.5 timeStamp*

Date and time the media file (image) was captured as an International Organization for Standardization 8601 (ISO 8601) formatted string with time zone designator YYYY-MM-DDThh:mm:ssZ or YYYY-MM-DDThh:mm:ss±hh:mm. This will be **auto filled** when executing deployMediaObs.py

Example: 2022-03-24T11:21:46Z

1.2.6 filePath*

URL or relative path to the media file, respectively, for externally hosted files or files that are part of the package. This will be **auto filled** when executing deployMediaObs.py

Example: /Volumes/Pi/*/DCIM/RCNX100/2022_UCSB02_2B_img_00001.jpg

1.2.7 fileName

Name of the media file. One can sort media chronologically within a deployment on timestamp (first) and fileName (second). This will be **auto filled** when executing deployMediaObs.py

1.2.8 fileMediatype*

The file media type for a specific file within the data package

Example: image/jpeg

1.2.9 exifData

Exif data dump of the media file (image) as a JSON object. Each entry will contain its own JSON object in its respective row. This will be **auto filled** when executing deployMediaObs.py

Example: {"EXIF":{"ISO":200,"Make":"RECONYX"}}

1.2.10 favourite

If a media file is deemed of interest, e.g., an exemplar photo of an individual, then that specific entry will be labeled as True. Otherwise, it will be labeled as False. This will be **auto filled as empty** when executing deployMediaObs.py

1.2.11 comments

Comments or notes about the media file. This will be **auto filled as empty** when executing deployMediaObs.py

1.2.12 _id

Until further changes are made, this will be auto-filled as empty when executing deployMediaObs.py

1.3 Observation.csv

descriptions of observations.csv

1.3.1 observationID*

Unique ID of the observation made throughout the data package. This will be **auto filled** when executing `deployMediaObs.py`

Example: 2022_UCSB02_2B_img_00001_OBS_1

1.3.2 deploymentID*

Unique identifier of the deployment the observation belongs to. Foreign key to `deployments.deploymentID`. This will be **auto filled** when executing `deployMediaObs.py`

Example: _UCSB02_2B

1.3.3 sequenceID*

Unique identifier of the sequence that is the source of the observation. Foreign key to `media.sequenceID`. This will be **auto filled** when executing `deployMediaObs.py`

Example: 2022_UCSB02_2B_img_00001_SEQ_1

1.3.4 mediaID

Unique identifier of the media file (image) that is the source of the observation. Foreign key to `media.mediaID`. This will be **auto filled** when executing `deployMediaObs.py`

Example: 2022_UCSB02_2B_img_00001

1.3.5 timestamp*

Date and time the media file (image) was captured as an International Organization for Standardization 8601 (ISO 8601) formatted string with time zone designator `YYYY-MM-DDThh:mm:ssZ` or `YYYY-MM-DDThh:mm:ss±hh:mm`. This will be **auto filled** when executing `deployMediaObs.py`

Example: 2022-03-01T22:00:00Z

1.3.6 observationType*

Classification of the media file (image) is done by the custom-trained tensor flow model. All categories in this vocabulary must be understandable from an AI point of view. Unknown describes

classifications with a confidence level below some predefined threshold, i.e., neither humans nor AI can say what was recorded. This will be **auto filled** when executing `deploymMediaObs.py`

Example: skunk

1.3.7 cameraSetup

Until further changes are made to the custom-trained tensor flow model, this will be **auto filled as empty** when executing `deploymMediaObs.py`

1.3.8 taxonID

Until further changes are made to the custom-trained tensor flow model, this will be **auto filled as empty** when executing `deploymMediaObs.py`

1.3.9 scientificName

Until further changes are made to the custom-trained tensor flow model, this will be **auto filled as empty** when executing `deploymMediaObs.py`

1.3.10 count

Until further changes are made to the custom-trained tensor flow model, this will be **auto filled as empty** when executing `deploymMediaObs.py`

1.3.11 countNew

Until further changes are made to the custom-trained tensor flow model, this will be **auto filled as empty** when executing `deploymMediaObs.py`

1.3.12 lifeStage

Until further changes are made to the custom-trained tensor flow model, this will be **auto filled as empty** when executing `deploymMediaObs.py`

1.3.13 sex

Until further changes are made to the custom-trained tensor flow model, this will be **auto filled as empty** when executing `deploymMediaObs.py`

1.3.14 behavior

Until further changes are made to the custom-trained tensor flow model, this will be **auto filled as empty** when executing `deploymMediaObs.py`

1.3.15 individualID

Until further changes are made to the custom-trained tensor flow model, this will be **auto filled as empty** when executing `deployMediaObs.py`

1.3.16 classificationMethod

Our classificationMethod is a custom-trained image classification architecture. Architectures may vary depending on who created and trained the model. This will be **auto filled** when executing `deployMediaObs.py`

Example: `custom_train_efficientnet_lite3`

1.3.17 classificationTimestamp

classificationTimestamp should be the time when the media file (image) was processed using the previously mentioned classificationMethod. The date and time of the classification should be in the form of an ISO 8601 formatted string with a timezone designator (YYYY-MM-DDThh:mm:ssZ or YYYY-MM-DDThh:mm:ss±hh:mm). This will be **auto filled** when executing `deployMediaObs.py`

Example: `2022-08-22T10:25:19Z`

1.3.18 classificationConfidence

classificationConfidence should be the accuracy or confidence that the classificationMethod had when classifying the media file (image). It should be a decimal with two significant figures, maximum confidence of 1 and minimum confidence of 0. This will be **auto filled** when executing `deployMediaObs.py`

Example: `2022-08-22T10:25:19Z`

1.3.19 comments

Comments or notes about the observation. This will be **autofilled as empty** when executing `deployMediaObs.py`

1.3.20 _id

Until further changes are made this will be **auto filled as empty** when executing `deployMediaObs.py`