Data requirements of the R package glatos

Updated: 2019-02-08

Contents

1 Overview

This vignette describes minimum data requirements of the R package *glatos* to inform loading of data that is not in standard file formats of the Great Lakes Acoustic Telemetry Observation System (GLATOS) or the Ocean Tracking Network (OTN). Strictly speaking, there are no requirements of the *glatos* package as a whole, but input data are checked within each individual function to determine if requirements are met. The set of data requirements described in this vignette, if followed, will ensure compatibility with all *glatos* functions.

For data in standard GLATOS and OTN formats, use of built-in data loading functions (see the Data Loading vignette for details) will ensure that resulting data objects meet the requirements of *glatos* functions. For reference, the appendix provides data field definitions (data dictionary) of standard data files obtained from the GLATOS Data Portal.

1.1 Data requirements

1.1.1 Detection data

glatos functions that accept detection data as input will typically require a *data.frame* with one or more of the following columns, named and defined exactly as described below:

- **detection_timestamp_utc** *A* POSIXct *object with detection timestamps (e.g., "2012-04-29 01:48:37").*

- **receiver_sn** *A character vector with unique receiver identifier. This is needed to associate each record with a specific receiver (a physical instrument).*

- **deploy_lat** *A numeric vector with latitude (decimal degrees, NAD83) of geographic location where receiver was deployed. Must be negative for locations in the southern hemishere and positive for locations in the northern hemisphere (e.g.,* 43.39165*).*

- **deploy_long** *A numeric vector with longitude (decimal degrees, NAD83) of geographic location where receiver was deployed. Must be negative for locations in the western hemishere and positive for locations in the eastern hemisphere (e.g.,* -83.99264*).*

- **transmitter_codespace** *A character string with transmitter code space (e.g., "A69-1061" for Vemco PPM coding). In combination with* transmiter_id *, this is needed to associate each record with a specific transmitter (a physical instrument).*

- **transmitter_id** *A character string with transmitter ID code (e.g., "1363" for Vemco PPM coding). In combination with* transmitter_codespace*, this is needed to associate each record with a specific transmitter (a physical instrument).*

- **sensor_value** *A numeric sensor measurement (e.g., an integer for 'raw' Vemco sensor tags).*

- **sensor_unit** *A character string with sensor_value units (e.g., "ADC"* for 'raw' Vemco sensor tag detections).**

- **animal_id** *A character string with individual animal identifier. This is used to associate each record with an individual animal.*

Additionally, some functions will require at least one categorical column to identify location (or group of locations). These can be specified by the user, but examples of such columns in a GLATOS standard detection file are:

- **Examples of columns that identify receiver locations (or groups)**
  - *glatos_array*
  - *station*
  - *glatos_project_receiver*

Any *data.frame* that contains the above columns should be compatible with all *glatos* functions that accept detection data as input. Use of the data loading functions *read_glatos_detections* and *read_otn_detections* will ensure that these columns are present, but can only be used on data in GLATOS and OTN formats. Data in other formats will need to be loaded using other functions (e.g., *read.csv*, *fread*, etc.) and compatibility with *glatos* functions will need to be carefully checked. For data loading examples, see the Data Loading vignette.

1.1.2   Receiver location data

glatos functions that accept receiver location data as input will typically require a *data.frame* with one or more of the following columns, named and defined exactly as described below:

- **deploy_lat** *A numeric vector with latitude (decimal degrees, NAD83) of geographic location where receiver was deployed. Must be negative for locations in the southern hemishere and positive for locations in the northern hemisphere (e.g.,* 43.39165*).*

- **deploy_long** *A numeric vector with longitude (decimal degrees, NAD83) of geographic location where receiver was deployed. Must be negative for locations in the western hemishere and positive for locations in the eastern hemisphere (e.g.,* -83.99264*).*

- **deploy_date_time** *A* POSIXct *object with timestamp when receiver was deployed (e.g.,* "2012-04-29 01:48:37"*).*

- **recover_date_time** *A* POSIXct *object with timestamp when receiver was recovered (e.g.,* "2012-04-29 01:48:37"*).*

Additionally, some functions will require at least one categorical column to identify location (or group of locations). These can be specified by the user, but examples of such columns in a GLATOS standard receiver locations file are:

- **Examples of columns that identify receiver locations (or groups)**
    - *glatos_array*
    - *station*
    - *glatos_project_receiver*

Use of the data loading function *read_glatos_receivers* will ensure that these columns are present, but can only be used on data in GLATOS format. Data in other formats will need to be loaded using other functions (e.g., *read.csv*, *fread*, etc.) and compatibility with *glatos* functions will need to be carefully checked. For data loading examples, see the Data Loading vignette.

1.1.3   Animal tagging and biological data

There are currently no *glatos* functions that require animal tagging and biological data other than those columns present in the required *detection* data. Therefore, there are no formal requirements of such data in the package. Nonetheless, the *read_glatos_workbook* function can be used to facilitate loading animal tagging and biological data from a standard GLATOS project workbook (*.xlsm file) into an R session.

Use of the data loading function *read_glatos_workbook* will ensure that animal data are loaded efficiently and consistently among users, but can only be used on data in GLATOS format. Data in other formats will need to be loaded using other functions (e.g., *read.csv*, *fread*, etc.). Although there are currently no *glatos* requirements of animal data, any future requirements might be expected to be consistent with the *glatos_animals* class.

1.1.4   Transmitter specification data

There are currently no *glatos* functions that require transmitter specification data. Therefore, there are no formal requirements of such data in the package. Nonetheless, the *read_vemco_tag_specs* function can be used to facilitate loading transmitter specification data from a standard VEMCO tag spec (*.xls) file provided to tag purchasers from VEMCO.

Use of the data loading function *read_vemco_tag_specs* will ensure that transmitter specification data are loaded efficiently and consistently among users, but can only be used on data in VEMCO standard format. Data in other formats will need to be loaded using other functions (e.g., *read.csv*, *fread*, etc.). Although there are currently no *glatos* requirements of transmitter specification data, any future requirements might be expected to be consistent with the output of *read_vemco_tag_specs*.

1.2 Data objects and classes

Most *glatos* data loading functions return an object with a *glatos*-specific S3 class name (e.g., *glatos_detections*) in addition to a more general class (e.g., *data.frame*). Currently, no methods exist for *glatos* classes and such classes are not explicitly required by any function, so *glatos* classes can merely be thought of as labels showing that the objects were produced by a *glatos* function and will therefore be compatible with other *glatos* functions. Beware, as with any S3 class, that it is possible to modify a *glatos* object to the point that is will no longer be compatible with *glatos* functions.

2 Appendix: GLATOS network standard data files

Detection data from the GLATOS network are queried for each individual project and made available through the project-specific GLATOS Data Portal. Each detection export is a zipped folder that contains multiple files. This appendix describes the structure of two of the comma-separated-value files (.csv) contained in the standard export: *detections* and *receiver locations*. These files can be identified by file name. The file that contains "detectionsWithLocs" (e.g., "HECWL_detectionsWithLocs_20180627_172857.csv") contains detections from acoustic receivers in the GLATOS network for all tags in a project. This file contains columns that identify when and where a fish was released and detected, some biological attributes of tagged fish, and tag model and specs. The .csv file with "receiverLocations" in the file name contains deployment and recovery operating schedules for all receivers in the GLATOS network. By combining the information in these two files, researchers are able determine when and where a tagged fish was detected in the GLATOS network and also locations where receivers were deployed that did not detect their tagged fish. Fields (columns) in both files are described below.

2.1 Detection file

A comma-separated-values text file with the following columns:

- **animal_id** *A field that uniquely identifies tagged animal. This field is used to associate each record with an individual animal.*

- **detection_timestamp_utc** *An alpha-numeric field in the standard POSIXct format (YYYY-MM-DD HH:MM:SS, i.e., "2012-04-29 01:48:37") that represents the date and time when a tag was detected in UTC timezone.*

- **glatos_array** *Character field identifies subgroup of receivers, usually in close spatial proximity.*

- **station_no** *Character field that identifies receiver within glatos_array.*

- **transmitter_codespace** *A character field with transmitter code space (e.g., "A69-1061" for Vemco PPM coding). In combination with transmiter_id, this is needed to associate each record with a specific transmitter (a physical instrument).*

- **transmitter_id** *A character field with transmitter ID code (e.g., 1363 for Vemco PPM coding). In combination with transmitter_codespace, this is needed to associate each record with a specific transmitter (a physical instrument).*

- **sensor_value** *A numeric sensor measurement (e.g., an integer for 'raw' Vemco sensor tags).*

- **sensor_unit** *A character field with sensor_value units (e.g., ADC for 'raw' Vemco sensor tag detections).*

- **deploy_lat** *A numeric field with latitude (decimal degrees, NAD83) of geographic location where receiver was deployed. Must be negative for locations in the southern hemishere and positive for locations in the northern hemisphere (e.g., 43.39165).*

- **deploy_long** *A numeric field with longitude (decimal degrees, NAD83) of geographic location where receiver was deployed. Must be negative for locations in the western hemishere and positive for locations in the eastern hemisphere (e.g., -83.99264).*

- **receiver_sn** *A character field that uniquely identifies each receiver. This is needed to associate each detection with a specific receiver (a physical instrument).*

- **tag_type** *Character field that identifies type of tag used.*

- **tag_model** *Character field that identifies tag model used.*

- **tag_serial_number** *Character field that identifies tag serial number.*

- **common_name_e** *Character field with fish species common name in english.*

- **capture_location** *Character field with description of where fish were captured (capture_location).*

- **length** *Numeric field of length of tagged fish.*

- **weight** *Numeric field of weight of tagged fish.*

- **sex** *Character field that identifies sex of fish released.*

- **release_group** *Character field that identifies group of released fish (release_group)*

- **release_location** *Character field that describes where fish was released.*

- **release_latitude** *A numeric field with latitude (decimal degrees, NAD83) of geographic location where tag was deployed. Must be negative for locations in the southern hemishere and positive for locations in the northern hemisphere (e.g., 43.39165).*

- **release_longitude** *A numeric field with longitude (decimal degrees, NAD83) of geographic location where tag was deployed. Must be negative for locations in the western hemishere and positive for locations in the eastern hemisphere (e.g., 43.39165).*

- **utc_release_date_time** *An alpha-numeric field in the standard POSIXct format (YYYY-MM-DD HH:MM:SS, i.e., "2012-04-29 01:48:37") that represents the date and time when a tag was released in UTC timezone.*

- **glatos_project_transmitter** *GLATOS five-character code of project associated with transmitter.*

- **glatos_project_receiver** *GLATOS five-character code of project associated with receiver.*

- **glatos_tag_recovered** *The values of the glatos_tag_recovered field is either "yes", or "no". A "yes" value indicates tag was recovered after deployment (e.g., tagged fish caught by angler) and "no" means the tag is still at large.*

- **glatos_caught_date** *If tag is recovered, the date (in YYYY-MM-DD format) is in the "glatos_caught_date" field.*

- **station** *Character field that combines glatos_array and station_no to identify receiver within project.*

- **min_lag** *minimum time interval (seconds) between current detection and another detection from the same tag on the same receiver. Field is used to identify possible false detections.*

2.2   Receiver location file

A comma-separated-values text file with the following columns:

- **station** *Character field that combines glatos_array and station_no to identify receiver within project.*

- **glatos_array** *Character field identifies subgroup of receivers, usually in close spatial proximity.*

- **station_no** *Character field that identifies receiver within glatos_array.*

- **consecutive_deploy_no** *Integer field of the number of times a receiver was deployed at a location.*

- **intend_lat** *Numeric field with intended geographic deployment latitude (decimal degrees, NAD83). Must be negative for locations in southern hemisphere and positive for locations in northern hemisphere.*

- **intend_long** *Numeric field with intended geographic deployment longitude (decimal degrees, NAD83). Must be positive for values in the eastern hemisphere and negative for locations in the western hemisphere.*

- **deploy_lat** *Numeric field of actual geographic deployment latitude (decimal degrees, NAD83). Must be negative for locations in southern hemisphere and positive for locations in northern hemisphere.*

- **deploy_long** *Numeric fields of actual geographic deployment longitude (decimal degrees, NAD83) must be positive for values in the eastern hemisphere and positive for locations in the western hemisphere.*

- **recover_lat** *Numeric fields of geographic latitude where receivers were recovered (decimal degrees, NAD83). Must be negative for locations in southern hemisphere and positive for locations in northern hemisphere.*

- **recover_long** *Numeric fields of geographic longitude where receivers were recovered (decimal degrees, NAD83). Must be negative for locations in western hemisphere and positive for locations in eastern hemisphere.*

- **deploy_date_time** *Alpha-numeric fields in standard POSIX format (YYYY-MM-DD HH:MM:SS, i.e., "2012-04-29 01:43:43") that represents the date and time in UTC timezone when a receiver was deployed.*

- **recover_date_time** *Alpha-numeric fields in standard POSIX format (YYYY-MM-DD HH:MM:SS, i.e., "2012-04-29 01:43:43") that represents the date and time in UTC timezone when a receiver was recovered (recover_date_time).*

- **bottom_depth** *Numeric field that represents the water depth (m; between bottom and surface) at location where receiver was deployed.*

- **riser_length** *Numeric field that represents the height of mooring equipment off of bottom (m).*

- **instrument_depth** *Numeric field that represents the distance (m) between the equipment and the water surface.*

- **ins_model_no** *Character field with user-defined model name of instrument deployed.*

- **glatos_ins_frequency** *Integer field with instrument operating frequency.*

- **ins_serial_no** *Character field of instrument model number.*

- **deployed_by** *Character field that contain name of person who deployed receiver.*

- **comments** *Character fields that contain comments recorded during deployment.*

- **glatos_seasonal** *Character field containing either "yes" or "no". Identifies seasonal ("yes") and annual receiver deployments ("no").*

- **glatos_project** *Character field containing unique 5-letter project identification code. This code is created by user when the project is initiated and submitted to GLATOS.*

- **glatos_vps** *Character field containing either "yes" or "no". If a receiver is part of a Vemco Positioning Study (VPS) then value equals "yes", otherwise field value is "no".*

2.3   Project workbook file

A macro-enabled Microsoft Excel workbook (*.xlsm) file containing the following worksheets:

- **Locations** *Contains descriptive data associated with receiver locations.*

- **Deployment** *Contains data associated with deployment of telemetry receivers.*

- **Recovery** *Contains data associated with recovery of telemetry receivers.*

- **Tagging** *Contains data associated with tagging of animals, including data associated with the animal and transmitter.*

For more details about the structure of this file, see the Data Submission Package in the GLATOS Data Portal.