

# Package: surimi (via r-universe)

June 4, 2026

**Title** Convert Acoustic Telemetry Data Between Institutional Formats

**Version** 0.0.0.1

**Description** Surimi takes as input data files representing acoustic telemetry, which may have column names and structures specific to particular institutions. It can convert, with minimal code on the user's part, data from one format to another, allowing data from one institution to be easily used across software packages that may expect different formats.

**License** GPL (>= 3)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Imports** dplyr, lubridate, tidyr, wrorms, nanoparquet, glatos

**Config/pak/sysreqs** libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libglpk-dev libavfilter-dev libicu-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

**Repository** <https://ocean-tracking-network.r-universe.dev>

**Date/Publication** 2026-06-04 18:33:49 UTC

**RemoteUrl** <https://github.com/ocean-tracking-network/surimi>

**RemoteRef** HEAD

**RemoteSha** 7a074267899446ac6f98c9c3cb5e6e598cc60df1

## Contents

get_aphiaid_from_lookup . . . . .	2
get_unique_aphiaids . . . . .	2
glatos_to_ato . . . . .	3
glatos_workbook_to_ato . . . . .	3
imos_otn_column_map . . . . .	4
imos_to_otn_detections . . . . .	4
imos_to_otn_receivers . . . . .	5
imos_to_otn_tags . . . . .	5

map_otn_file . . . . .	6
otn_imos_column_map . . . . .	7
otn_imos_new_style_column_map . . . . .	8
otn_to_ato . . . . .	9
rollup . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

get\_aphiaid\_from\_lookup  
*Consult a lookup table for the aphiaID.*

---

### Description

This is the helper function that we use in the sapply when mutating the WORMS\_species\_aphia\_id into existence.

### Usage

```
get_aphiaid_from_lookup(sciname, lookup)
```

### Arguments

sciname            A Scientific name as a string.  
lookup             The named list containing key-value pairs of scientific names and aphiaIDs.

### Value

Returns the appropriate aphiaID corresponding to the sciname.

---

get\_unique\_aphiaids    *Get AphiaIDs for scientific names*

---

### Description

Takes a column of scientific names and creates a lookup table (read: named list) of the unique scientific names against their aphia IDs. We can use worms to query the WORMS REST service for the aphiaIDs, but doing it for every row is time intensive in a way we don't want. This way, we can create the lookup client-side and then do all the querying only as we need to.

### Usage

```
get_unique_aphiaids(scinames)
```

**Arguments**

scinames            A vector (dataframe column) containing the list of scientific names from a detection extract dataframe in Surimi.

**Value**

Returns a named list with the scientific name as the key and the aphiaID as the value.

---

glatos\_to\_ato            *Convert GLATOS detection data to an ATO object.*

---

**Description**

Takes a GLATOS detection sheet and optionally receiver metadata and returns an ATO object.

**Usage**

```
glatos_to_ato(glatos_detections, glatos_receivers = "")
```

**Arguments**

glatos\_detections            The dataframe containing detection information.

glatos\_receivers            The dataframe containing receiver information.

**Value**

Returns an ATO object.

---

glatos\_workbook\_to\_ato            *Convert GLATOS workbook to an ATO object.*

---

**Description**

Takes a GLATOS workbook and returns an ATO object.

**Usage**

```
glatos_workbook_to_ato(glatos_workbook)
```

**Arguments**

glatos\_workbook            Path to the glatos workbook.

**Value**

Returns an ATO object.

---

imos\_otn\_column\_map     *Map IMOS receiver metadata to an OTN-like format*

---

**Description**

In the same way that `otn_imos_column_map` takes OTN data and messages it into an IMOS-like format for REMORA, this function and its ilk take IMOS data (in this case, receiver metadata) and message it into an OTN-like format, for the purposes of reporting and more general applicability within the OTN suite of programs.

**Usage**

```
imos_otn_column_map(
  det_dataframe,
  rcvr_dataframe = NULL,
  tag_dataframe = NULL,
  derive = TRUE
)
```

**Arguments**

`det_dataframe` ...  
`rcvr_dataframe` A dataframe containing IMOS receiver metadata.  
`tag_dataframe` ...  
`derive` ...

**Value**

A dataframe containing the above data in an OTN-like format.

---

imos\_to\_otn\_detections     *Map IMOS detection data to an OTN-like format*

---

**Description**

In the same way that `otn_imos_column_map` takes OTN data and messages it into an IMOS-like format for REMORA, this function and its ilk take IMOS data (in this case, animal detections) and messages it into an OTN-like format, for the purposes of reporting and more general applicability within the OTN suite of programs.

**Usage**

```
imos_to_otn_detections(detection_dataframe, coll_code = NULL)
```

**Arguments**

rcvr\_dataframe A dataframe containing IMOS receiver metadata.

**Value**

A dataframe containing the above data in an OTN-like format.

---

imos\_to\_otn\_receivers *Map IMOS receiver metadata to an OTN-like format*

---

**Description**

In the same way that `otn_imos_column_map` takes OTN data and messages it into an IMOS-like format for REMORA, this function and its ilk take IMOS data (in this case, receiver metadata) and message it into an OTN-like format, for the purposes of reporting and more general applicability within the OTN suite of programs.

**Usage**

```
imos_to_otn_receivers(rcvr_dataframe)
```

**Arguments**

rcvr\_dataframe A dataframe containing IMOS receiver metadata.

**Value**

A dataframe containing the above data in an OTN-like format.

---

imos\_to\_otn\_tags *Map IMOS tag metadata to an OTN-like format*

---

**Description**

In the same way that `otn_imos_column_map` takes OTN data and messages it into an IMOS-like format for REMORA, this function and its ilk take IMOS data (in this case, tag metadata and animal measurements data) and message it into an OTN-like format, for the purposes of reporting and more general applicability within the OTN suite of programs.

**Usage**

```
imos_to_otn_tags(tag_dataframe, animal_measurements_dataframe)
```

**Arguments**

tag\_dataframe A dataframe containing IMOS-formatted tag metadata.  
 animal\_measurements\_dataframe A dataframe containing IMOS-formatted animal measurements data.

**Value**

A single dataframe containing the tag and measurement data combined into an OTN-like format.

---

map_otn_file	<i>Determine whether an input file is CSV or Parquet and pipe it into the correct mapping function. Hopefully this is all a stopgap until the typical format of OTN -&gt; ATO and then ATO -&gt; IMOS gets done, at which point those pipeline pieces will connect together. But for now, this will keep dependent software like Remora running.</i>
--------------	--

---

**Description**

Determine whether an input file is CSV or Parquet and pipe it into the correct mapping function. Hopefully this is all a stopgap until the typical format of OTN -> ATO and then ATO -> IMOS gets done, at which point those pipeline pieces will connect together. But for now, this will keep dependent software like Remora running.

**Usage**

```
map_otn_file(filename, derive = TRUE, coll_code = NULL)
```

**Arguments**

filename The path to the file to be processed.  
 derive Passed through to the mapping functions; determines whether or not receiver and tag metadata will be derived from the detection extract or not.  
 coll\_code Passed through to the mapping functions; allows user to supply collectionCode if they are passing their own rcvr/tag metadata, which won't contain the collectionCode.

**Value**

The output of the appropriate mapping function.

---

otn\_imos\_column\_map    *Map OTN-formatted data to IMOS-format*

---

### Description

Takes three dataframes in the OTN format- one for a detection extract, one for receiver deployment metadata, and one for tag metadata- and rearranges, renames, and creates columns until they can pass for IMOS-format dataframes. This allows us to pass the data directly into Remora without making substantial changes to how that code runs or what it looks for.

### Usage

```
otn_imos_column_map(  
  det_dataframe,  
  rcvr_dataframe = NULL,  
  tag_dataframe = NULL,  
  derive = TRUE,  
  coll_code = NULL,  
  tagname_column = "tagname"  
)
```

### Arguments

det_dataframe	The dataframe containing detection information. Most likely a detection extract.
rcvr_dataframe	The dataframe containing receiver information.
tag_dataframe	The dataframe containing tag information.
derive	An optional flag that allows the user to pass in fewer than all three files. If given, the code will use the detection extract dataframe to generate dataframes for either or both of the receiver and tag dataframes, if they are not passed in. Although this will result in missing information, it does let the user supply only a detection extract file, which is a situation some may find themselves in.
coll_code	The user-supplied collectioncode, which we'll use to populate the receiver_project_name and tagging_project_name columns in the receiver and tag metadata files respectively. We don't have a good way to associate the relevant info from the det extract to the appropriate columns in the rcvr/tag metadata, but those datasets are restricted to one collectioncode each, so we can just take it from the user at the time they run the code.
tagname_column	The name of the column that's equivalent to 'tagname', if the tagname column isn't present. Should only be necessary if deriving.

### Value

Returns a list containing three approximately IMOS-formatted dataframes.

---

```
otn_imos_new_style_column_map
```

*Map OTN-formatted data from our new Parquet detection extracts to IMOS-format*

---

### Description

Takes three dataframes in the OTN format- one for a detection extract, one for receiver deployment metadata, and one for tag metadata- and rearranges, renames, and creates columns until they can pass for IMOS-format dataframes. This allows us to pass the data directly into Remora without making substantial changes to how that code runs or what it looks for. This is functionally identical to `otn_imos_column_map()` except that the column names on the OTN side reflect that the detection dataframe came from our new parquet format rather than our old CSV format.

### Usage

```
otn_imos_new_style_column_map(
  det_dataframe,
  rcvr_dataframe = NULL,
  tag_dataframe = NULL,
  derive = TRUE,
  coll_code = NULL,
  tagname_column = "tagName",
  format = "parquet"
)
```

### Arguments

<code>det_dataframe</code>	The dataframe containing detection information.
<code>rcvr_dataframe</code>	The dataframe containing receiver information.
<code>tag_dataframe</code>	The dataframe containing tag information.
<code>derive</code>	An optional flag that allows the user to pass in fewer than all three files. If given, the code will use the detection extract dataframe to generate dataframes for either or both of the receiver and tag dataframes, if they are not passed in. Although this will result in missing information, it does let the user supply only a detection extract file, which is a situation some may find themselves in.
<code>coll_code</code>	The user-supplied collectioncode, which we'll use to populate the <code>receiver_project_name</code> and <code>tagging_project_name</code> columns in the receiver and tag metadata files respectively. We don't have a good way to associate the relevant info from the det extract to the appropriate columns in the rcvr/tag metadata, but those datasets are restricted to one collectioncode each, so we can just take it from the user at the time they run the code.
<code>tagname_column</code>	The name of the column that's equivalent to 'tagName', if the tagName column isn't present. Should only be necessary if deriving.

**format** Defaults to parquet. Since the column names are the same across parquet files and new-style CSV files, this function can handle both as long as it knows what it's getting. Calling `map_otn_file` will handle all the checking for the user, though.

### Value

Returns a list containing three approximately IMOS-formatted dataframes.

---

`otn_to_ato` *Convert OTN detection data to an ATO object.*

---

### Description

Takes an OTN detection extract and optionally receiver/tag metadata and returns an ATO object.

### Usage

```
otn_to_ato(otn_detections, otn_receivers = "", otn_tags = "")
```

### Arguments

`otn_detections` The dataframe containing detection information.  
`otn_receivers` The dataframe containing receiver information.  
`otn_tags` The dataframe containing tag information.

### Value

Returns an ATO object.

---

`rollup` *Join output from Remora back onto its OTN detection extract.*

---

### Description

Take two parameters- an OTN detection extract and the output created by Remora on parsing that detection extract- and merge them back together such that the Remora QC columns are appended to the OTN extract, preserving appropriate ordering and getting back all the OTN data. This function exists because, to get OTN data into Remora, we have to cut it up until it looks like IMOS data (this problem was the genesis of Surimi, in fact). But that means the output from Remora has all IMOS-formatted columns and is missing some information, because we either had to discard it to get into IMOS format or because we can't re-synthesize it from what's in the IMOS files. However, we do have enough information to join the two tables, thereby obviating the data loss problem by taking us all the way back to the original data, with a little something extra attached.

**Usage**

```
rollup(detection_extract, remora_output)
```

**Arguments**

detection\_extract

Path to an OTN detection extract corresponding to the remora output in the second parameter.

remora\_output

Path to Remora's QC output corresponding to the OTN detection extract in the first parameter.

**Value**

The OTN detection extract, but with the remora QC attached as appropriate.

# Index

[get\\_aphiaid\\_from\\_lookup](#), 2  
[get\\_unique\\_aphiaids](#), 2  
[glatos\\_to\\_ato](#), 3  
[glatos\\_workbook\\_to\\_ato](#), 3

[imos\\_otn\\_column\\_map](#), 4  
[imos\\_to\\_otn\\_detections](#), 4  
[imos\\_to\\_otn\\_receivers](#), 5  
[imos\\_to\\_otn\\_tags](#), 5

[map\\_otn\\_file](#), 6

[otn\\_imos\\_column\\_map](#), 7  
[otn\\_imos\\_new\\_style\\_column\\_map](#), 8  
[otn\\_to\\_ato](#), 9

[rollup](#), 9