

# Package: camalienr (via r-universe)

June 14, 2026

**Title** Functions Used in the Plant Module of the Biodiversa+ Pilot on Invasive Alien Species

**Version** 0.4.2

**Description** Tools used in the work with the vehicle mounted CamAlien system for monitoring plants. The package provides functions to interact with the CamAlien database.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**URL** <https://camalienr.z16.web.core.windows.net>

**BugReports** <https://gitlab.au.dk/ecos/biodiversa/camalien/camalienr/-/issues>

**Imports** DBI, dbplyr, dplyr, fs, glue, jsonlite, magick, purrr, RPostgres, sf, tibble, tidyr

**Config/Needs/website** rmarkdown, ggplot2, ggridges, mapview

**Depends** R (>= 3.5)

**LazyData** true

**Config/pak/sysreqs** libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libmagick++-dev gsfonts libicu-dev libuv1-dev libssl-dev libpq-dev libproj-dev libsqlite3-dev libudunits2-dev

**Repository** <https://ldalby.r-universe.dev>

**Date/Publication** 2025-05-02 09:24:40 UTC

**RemoteUrl** <https://gitlab.au.dk/ecos/biodiversa/camalien/camalienr>

**RemoteRef** HEAD

**RemoteSha** d5bbe3e002432880571273bb7773d65b201904e8

## Contents

ca_as_sf . . . . .	2
ca_connect . . . . .	3

ca_disconnect . . . . .	3
ca_get_image . . . . .	4
ca_get_join_tbl . . . . .	4
ca_get_polygon . . . . .	5
ca_img_detection . . . . .	5
ca_img_draw_polygon . . . . .	6
ca_img_grid . . . . .	7
ca_img_plot . . . . .	7
ca_img_read . . . . .	8
ca_json_read . . . . .	8
ca_parse_chunks . . . . .	9
ca_parse_job . . . . .	9
ca_stat_images . . . . .	10
ca_stat_jobsize . . . . .	10
ca_stat_tagged . . . . .	11
ias . . . . .	11
id_box . . . . .	12
images . . . . .	13
partners . . . . .	14
<b>Index</b>	<b>15</b>

---

ca_as_sf	<i>Convert data.frame to Simple Feature collection</i>
----------	--

---

## Description

Convert data.frame to Simple Feature collection by converting the position column (pg\_geometry) to sfc.

## Usage

```
ca_as_sf(tbl)
```

## Arguments

tbl	The data.frame to be converted
-----	--------------------------------

## Value

tbl as Simple Feature collection

---

ca_connect	<i>Connect to the camalien database</i>
------------	---

---

### Description

Establish a connection to the camalien database. This function is a wrapper around [DBI::dbConnect](#). The function will create a connection using [RPostgres::Postgres](#) as a driver. For the function to work you need to have the following environment variables set in your .REnvirion file:

- CAMALIEN\_USER
- CAMALIEN\_PASSWORD

### Usage

```
ca_connect()
```

### Details

To modify or create an .REnvirion file we suggest you use [usethis::edit\\_r\\_envirion](#). Calling this function from RStudio will open the file for you to edit. Simply insert the variables, save and restart R.

### Value

PqDriver

---

ca_disconnect	<i>Disconnect from the database</i>
---------------	-------------------------------------

---

### Description

This is a simple wrapper around [DBI::dbDisconnect](#) to use when disconnecting from a database. It is good practice to disconnect when no longer using the connection.

### Usage

```
ca_disconnect(con)
```

### Arguments

con	DBIConnection The connection object which in the case of camalien will be a PqConnection.
-----	---

### Value

TRUE, invisibly

---

ca_get_image	<i>Get tables from the camalien database</i>
--------------	--

---

### Description

Get tables from the data schema of the camalien database. This function only return a lazy table and will not download any data to your session. Use `dplyr::collect` when you actually want to download the data. Normally you would want to do some selection and filtering before calling `collect()`

### Usage

```
ca_get_image(con)
ca_get_job(con)
ca_get_plantnetcall(con)
ca_get_detection(con)
ca_get_detectionssummary(con)
ca_get_imagemeta(con)
ca_get_species(con)
ca_get_chunk(con)
ca_get_partner(con)
```

### Arguments

con	PqConnection	The connection to the camalien database
-----	--------------	---

### Value

A lazy table

---

ca_get_join_tbl	<i>Create and return a table for filtering rows to a partner</i>
-----------------	--

---

### Description

A common task is to filter e.g. detections made within one of the partner countries. This involves a number of joins to carry the partner name all the way through to the image or detection table. This function sets up the required joins and can filter the resulting table to a single partner by supplying the `partner_name` argument.

**Usage**

```
ca_get_join_tbl(con, partner_name = NULL)
```

**Arguments**

```
con          PqConnection The connection to the camalien database
partner_name chr Name of the partner
```

**Value**

A lazy table

---

ca_get_polygon	<i>Extract polygon information from Plantnet response</i>
----------------	---

---

**Description**

This function will extract the relevant information from a JSON file returned from the Plantnet API.

**Usage**

```
ca_get_polygon(pn_json)
```

**Arguments**

```
pn_json      The JSON object (read by e.g. jsonlite::read_json)
```

**Value**

list of center\_x, center\_y and size

---

ca_img_detection	<i>Plot image and polygon with detection</i>
------------------	--

---

**Description**

Draw an image and show the polygon where the detection was made as a polygon on top of the image.

**Usage**

```
ca_img_detection(data, height = 3000L, border = NA, col = NA, lwd = 4)
```

**Arguments**

data	data.frame Data.frame with imageurl, centerx, centery and size.
height	Integer specifying the height of the desired plot. Width is scaled proportionally.
border	chr Polygon outline color
col	chr Color to use if plotting filled polygons. The format is #RRGGBBAA where AA is the alpha channel for transparency
lwd	int Polygon outline line width

**Examples**

```
## Not run:
images[1,] |>
  ca_img_detection(height = 500, border = "red")

images |>
  ca_img_detection(height = 500, col = "#FF000040")

## End(Not run)
```

---

ca\_img\_draw\_polygon     *Draw polygons on top of images*

---

**Description**

Draw a polygon and annotate it with ID

**Usage**

```
ca_img_draw_polygon(
  center_x,
  center_y,
  size,
  border = "red",
  col,
  lwd = 4,
  scale
)
```

**Arguments**

center_x, center_y	int coordinates of the centre of the polygon
size	int size of the polygon
border	chr Polygon outline color
col	chr Color to use if plotting filled polygons. The format is #RRGGBBAA where AA is the alpha channel for transparency

lwd	int Polygon outline line width
scale	double Scaling factor if the image is not rendered full size

---

ca_img_grid	<i>Plot grid of images</i>
-------------	----------------------------

---

### Description

Plot a grid of multiple images.

### Usage

```
ca_img_grid(img_paths, ncol = 3, nrow = 3, height = 300L)
```

### Arguments

img_paths	Vector of paths (can be URLs). Should be of length nrow * ncol.
ncol, nrow	Int, the number of rows and columns
height	Integer specifying the height of the desired plot. Width is scaled proportionally.

### Examples

```
## Not run:
images$imageurl |>
ca_img_grid(ncol = 5, nrow = 1)

## End(Not run)
```

---

ca_img_plot	<i>Plot image</i>
-------------	-------------------

---

### Description

Plot and optionally scale image

### Usage

```
ca_img_plot(img, height = 3000L)
```

### Arguments

img	magick-image object read by <a href="#">ca_img_read()</a>
height	Integer specifying the height of the desired plot. Width is scaled proportionally.

ca\_img\_read                      *Read image from URL*

---

**Description**

Read image file for viewing or further manipulation

**Usage**

```
ca_img_read(url)
```

**Arguments**

url                      URL to the image to be read.

**Value**

magick-image object

---

ca\_json\_read                      *Read state files from CamAlien uploader*

---

**Description**

A simple wrapper around [jsonlite::read\\_json\(\)](#).

**Usage**

```
ca_json_read(path)
```

**Arguments**

path                      chr Path to the json file to read.

**Value**

list with content of json file being read

---

ca_parse_chunks	<i>Parse json files with info on chunks</i>
-----------------	---

---

**Description**

Parse json files with information about the individual chunks of images from the CamAlien uploader.

**Usage**

```
ca_parse_chunks(path)
```

**Arguments**

path                    chr Path to the json file to read.

**Details**

The function will read all the chunk files in the directory at path.

**Value**

data.frame with content of json files being read

---

ca_parse_job	<i>Parse json files with info on the job</i>
--------------	--

---

**Description**

Parse json files with information about the job from the CamAlien uploader.

**Usage**

```
ca_parse_job(path)
```

**Arguments**

path                    chr Path to the json file to read.

**Value**

data.frame with content of json file being read

---

ca_stat_images	<i>Get statistics on images</i>
----------------	---------------------------------

---

**Description**

Calculate statistics on the images uploaded by each partner

**Usage**

```
ca_stat_images(con)
```

**Arguments**

con	PqConnection	The connection to the camalien database
-----	--------------	---

**Value**

lazy\_tbl

---

ca_stat_jobsize	<i>Get statistics on upload jobs</i>
-----------------	--------------------------------------

---

**Description**

Calculate sizes on the upload jobs that each partner has started. There is some inaccuracies relating to this as we can't see if images in one job are also part of other jobs.

**Usage**

```
ca_stat_jobsize(con)
```

**Arguments**

con	PqConnection	The connection to the camalien database
-----	--------------	---

**Details**

The Danish images were uploaded with an early version of the uploader which did not report the size of the job and therefore total\_size is NA here.

**Value**

lazy\_tbl

---

ca_stat_tagged	<i>Get statistics on tagged images</i>
----------------	--

---

**Description**

Calculate the number and ratio of tagged images.

**Usage**

```
ca_stat_tagged(con)
```

**Arguments**

con                   PqConnection The connection to the camalien database

**Value**

lazy\_tbl with partner, n\_tagged, n and prop

---

ias	<i>List of target species for all partners</i>
-----	--

---

**Description**

Each partner has their own list of invasive alien species which they targeted during the field campaign. The same species can be on the list of multiple partners. Target species can be an entire genus. The scientific and the canonical name are from a lookup on [GBIF](#).

**Usage**

```
ias
```

**Format**

Data.frame with gbifid and name

**partner** chr, The partner name

**canonical\_name** chr, The canonical name from GBIF with the modification that for genera we add .sp

**scientificname** chr, The scientific taxon name

**gbifid** int, The gbifid used by Plantnet and in the camalien database

**pnid** int, The ID used by Plantnet (hence pnid) and in the camalien database

**taxon\_name\_key** chr, The GBIF key for the taxon name

**taxon\_rank** chr, The taxon rank, either species or genus. Derived from GBIF rankMarker

---

id\_box

*Draw an ID box on an image*


---

### Description

Modified [graphics::legend](#) function to draw small boxes with ID information on top of images

### Usage

```
id_box(
  x,
  y = NULL,
  legend,
  col = par("col"),
  border = "black",
  angle = 45,
  bty = "o",
  bg = "white",
  box.lwd = par("lwd"),
  box.lty = par("lty"),
  box.col = par("fg"),
  cex = 1,
  xjust = 0,
  yjust = 1,
  adj = c(0.25, 0),
  text.width = NULL,
  text.col = par("col"),
  text.font = NULL,
  ncol = 1,
  horiz = FALSE,
  inset = 0,
  xpd,
  seg.len = 1
)
```

### Arguments

x, y	the x and y co-ordinates to be used to position the legend. They can be specified by keyword or in any way which is accepted by <a href="#">xy.coords</a> : See ‘Details’.
legend	a character or <a href="#">expression</a> vector of length $\geq 1$ to appear in the legend. Other objects will be coerced by <a href="#">as.graphicsAnnot</a> .
col	the color of points or lines appearing in the legend.
border	the border color for the boxes (used only if <code>fill</code> is specified).
angle	angle of shading lines.
bty	the type of box to be drawn around the legend. The allowed values are "o" (the default) and "n".

<code>bg</code>	the background color for the legend box. (Note that this is only used if <code>bty != "n"</code> .)
<code>box.lty, box.lwd, box.col</code>	the line type, width and color for the legend box (if <code>bty = "o"</code> ).
<code>cex</code>	character expansion factor <b>relative</b> to current <code>par("cex")</code> . Used for text, and provides the default for <code>pt.cex</code> .
<code>xjust</code>	how the legend is to be justified relative to the legend x location. A value of 0 means left justified, 0.5 means centered and 1 means right justified.
<code>yjust</code>	the same as <code>xjust</code> for the legend y location.
<code>adj</code>	numeric of length 1 or 2; the string adjustment for legend text. Useful for y-adjustment when labels are <a href="#">plotmath</a> expressions.
<code>text.width</code>	the width of the legend text in x ("user") coordinates. (Should be positive even for a reversed x axis.) Can be a single positive numeric value (same width for each column of the legend), a vector (one element for each column of the legend), NULL (default) for computing a proper maximum value of <code>strwidth(legend)</code> , or NA for computing a proper column wise maximum value of <code>strwidth(legend)</code> .
<code>text.col</code>	the color used for the legend text.
<code>text.font</code>	the font used for the legend text, see <a href="#">text</a> .
<code>ncol</code>	the number of columns in which to set the legend items (default is 1, a vertical legend).
<code>horiz</code>	logical; if TRUE, set the legend horizontally rather than vertically (specifying <code>horiz</code> overrides the <code>ncol</code> specification).
<code>inset</code>	inset distance(s) from the margins as a fraction of the plot region when legend is placed by keyword.
<code>xpd</code>	if supplied, a value of the <a href="#">graphical parameter</a> <code>xpd</code> to be used while the legend is being drawn.
<code>seg.len</code>	the length of lines drawn to illustrate <code>lty</code> and/or <code>lwd</code> (in units of character widths).

---

 images

---

*Extract of detections and image metadata*


---

### Description

Small example dataset with scores, boxes and urls.

### Usage

images

**Format**

Data.frame with the following columns:

**callid** chr, ID of the Plantnet call

**score** float, The model confidence score from Plantnet

**boxsize** int, Size of the polygon where the detection was made

**centerx** int, The center x coordinate of the polygon

**centery** int, The center y coordinate of the polygon

**speciesid** int, The plantnet species ID

**imageid** int, The image ID

**imageurl** int, The public URL to the image file

---

partners

*List of partners and their acronyms*

---

**Description**

List of partners and their acronyms

**Usage**

partners

**Format**

Data.frame with partner and partner\_acronym

**partner** chr, Name of the partner

**partner\_acronym** chr, The acronym used in the Biodiversa+ project

# Index

## \* datasets

ias, 11  
images, 13  
partners, 14

as.graphicsAnnot, 12

ca\_as\_sf, 2  
ca\_connect, 3  
ca\_disconnect, 3  
ca\_get\_chunk (ca\_get\_image), 4  
ca\_get\_detection (ca\_get\_image), 4  
ca\_get\_detectionssummary (ca\_get\_image),  
4  
ca\_get\_image, 4  
ca\_get\_imagemeta (ca\_get\_image), 4  
ca\_get\_job (ca\_get\_image), 4  
ca\_get\_join\_tbl, 4  
ca\_get\_partner (ca\_get\_image), 4  
ca\_get\_plantnetcall (ca\_get\_image), 4  
ca\_get\_polygon, 5  
ca\_get\_species (ca\_get\_image), 4  
ca\_img\_detection, 5  
ca\_img\_draw\_polygon, 6  
ca\_img\_grid, 7  
ca\_img\_plot, 7  
ca\_img\_read, 8  
ca\_img\_read(), 7  
ca\_json\_read, 8  
ca\_parse\_chunks, 9  
ca\_parse\_job, 9  
ca\_stat\_images, 10  
ca\_stat\_jobsize, 10  
ca\_stat\_tagged, 11

DBI::dbConnect, 3  
DBI::dbDisconnect, 3  
dplyr::collect, 4

expression, 12

graphical parameter, 13  
graphics::legend, 12

ias, 11  
id\_box, 12  
images, 13

jsonlite::read\_json(), 8

partners, 14  
plotmath, 13

RPostgres::Postgres, 3

strwidth, 13

text, 13

usethis::edit\_r\_environ, 3

xy.coords, 12