

# Package: infinitylists (via r-universe)

December 8, 2024

**Title** A Shiny-Based Application for Generating Place-Based Species Lists for Australian Plants

**Version** 1.0.0

**Description** This shiny-based application allows users to extract plant occurrence data from the Atlas of Living Australia and generate a species list for any defined area in New South Wales. All records associated with either a physical voucher (stored in Australian herbaria) or a photographic voucher (stored in iNaturalist) are extracted. For each species within the defined area, the application will return voucher type, number of vouchers, date of the most recent voucher, spatial coordinates, voucher location, and the voucher collector. Records are displayed both in a table and on a map, and are downloadable as a CSV.

**License** CC BY 4.0

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Depends** R (>= 4.1)

**LazyData** true

**Imports** APCalign (>= 0.1.1), arrow, DT, dplyr, data.table, galah, httr, janitor, jsonlite, leaflet, lubridate, purrr, sf, shiny, shinybusy, stringr, shinythemes, tidyr

**Remotes** traitecoevo/APCalign

**Suggests** bsplus, here, job, knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**URL** <https://traitecoevo.github.io/infinitylists/>

**Config/pak/sysreqs** cmake libgdal-dev gdal-bin libgeos-dev make libicu-dev libpng-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev zlib1g-dev

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

**RemoteUrl** <https://github.com/traitecoevo/infinitylists>

**RemoteRef** HEAD

**RemoteSha** 009d4b90cca1bd3b2ee2ea1ed9547916cdb0e3e

## Contents

download_ala_obs . . . . .	2
download_gbif_obs . . . . .	3
infinitylistApp . . . . .	4
infinity_server . . . . .	5
query_gbif_global . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

download_ala_obs	<i>Download Observations for Infinity List</i>
------------------	--

---

## Description

This function downloads observation data for the given taxon and year range from the "Infinity List" source, processes and cleans the data, adds additional columns (e.g., establishment status), and optionally saves the raw and processed data.

## Usage

```
download_ala_obs(
  taxon,
  year_range = c(1923, as.numeric(format(Sys.Date(), "%Y"))),
  save_raw_data = FALSE,
  output_dir = file.path(system.file(package = "infinitylists"), "data/")
)
```

## Arguments

taxon	A character vector or string specifying the taxon (e.g., species, genus) for which observations are to be downloaded.
year_range	A numeric vector of length 2 indicating the start and end years for data retrieval. Default is from 1923 to the current year
save_raw_data	A logical value indicating whether to save the raw data. By default, raw data is not saved (FALSE).
output_dir	A character string specifying the directory where any saved data (raw or processed) will be stored. Default is <code>file.path(system.file(package = "infinitylists"), "data/")</code> .

**Details**

The function carries out the following steps:

1. Retrieve the data from the "Infinity List" source.
2. Process and clean the retrieved data to remove any inconsistencies.
3. Add additional columns to the cleaned data, such as the establishment status for the given taxon.
4. If `save_raw_data` is TRUE, save the processed data to the specified `output_dir`.

**Value**

This function saves the processed data and returns invisibly. The structure and content of the returned value (if any) is determined by the functions called within (e.g., `retrieve_data`, `process_data`).

---

<code>download_gbif_obs</code>	<i>Download and Process GBIF Observations</i>
--------------------------------	---

---

**Description**

This function retrieves, processes, and saves GBIF (Global Biodiversity Information Facility) observation data for a specified taxon.

**Usage**

```
download_gbif_obs(
  taxon,
  min_year = 1923,
  max_year = as.numeric(format(Sys.Date(), "%Y")),
  country_code = NULL,
  save_raw_data = FALSE,
  output_dir = file.path(system.file(package = "infinitylists"), "data/")
)
```

**Arguments**

<code>taxon</code>	Character. The taxon (species, genus, etc.) for which to retrieve GBIF data.
<code>min_year</code>	Numeric. The minimum year for the observations to be retrieved. Default is 1923.
<code>max_year</code>	Numeric. The maximum year for the observations to be retrieved. Default is the current year.
<code>country_code</code>	Character. The ISO 3166-1 alpha-2 country code to filter observations by country. Default is NULL (no country filter).
<code>save_raw_data</code>	Logical. Whether to save the raw data retrieved from GBIF. Default is FALSE.
<code>output_dir</code>	Character. The directory where the processed data will be saved. Default is a "data" directory within the "infinitylists" package.

**Value**

None. The function saves the processed data to the specified output directory.

---

`infinitylistApp`*Launch the Infinity List Application*

---

**Description**

This function initializes and runs the "Infinity List" application using Shiny. Before launching, it checks for any available updates and downloads them if necessary. The app provides a user interface (ui) and server-side logic (server) to interactively work with the Infinity List.

**Usage**

```
infinitylistApp(...)
```

**Arguments**

... Additional arguments passed to `shinyApp()`. This can include parameters like `port`, `launch.browser`, and others as described in the documentation for `shinyApp()`.

**Details**

The function carries out the following steps:

1. Checks and downloads any available updates using `check_and_download_update()`.
2. Initializes the user interface (ui) for the application.
3. Sets up the server-side logic (server) for the application.
4. Launches the Shiny application using `shinyApp()`.

**Value**

This function launches the Shiny application and does not return until the app is terminated by the user.

**Examples**

```
## Not run:  
# Launch the Infinity List App  
infinitylistApp()  
  
## End(Not run)
```

---

infinity_server	<i>Infinity List Server Function</i>
-----------------	--------------------------------------

---

**Description**

Initializes and sets up the server-side logic for the Infinity List Shiny application. This function manages the user interactions, data processing, and rendering of both UI elements and outputs (tables, maps, stats). It handles file uploads, data filtering, and interaction with map elements.

**Usage**

```
infinity_server(...)
```

**Arguments**

... Additional arguments that might be passed to server functions. These could be passed from a Shiny UI function or another server function.

**Details**

The server function handles several main tasks:

- Observes and processes KML file uploads.
- Reactively creates polygons based on user inputs.
- Filters and processes ALA data based on user-selected polygons.
- Computes and displays statistics about observed species.
- Updates select input choices based on the available data.
- Renders and displays a data table with specific details.
- Provides a CSV download handler for the displayed data.
- Renders a Leaflet map with markers, polygons, and buffers.

Key internal functions include:

- `load_place()`: For processing KML files.
- `create_circle_polygon()`: To create polygons from user inputs.
- `points_in_target()`: To determine if points lie within a target polygon.
- `points_in_buffer()`: To determine if points lie within a buffer around a polygon.
- `filter_by_taxon()`: Filters the ALA data by selected taxa.
- `add_buffer()`: Adds a buffer around a given polygon.

**Value**

This function sets up and returns a server function for the Shiny app. It does not have a direct return value, but rather, it sets up reactive outputs, observers, and expressions that the Shiny app will utilize.

**See Also**

[infinitylistApp](#): Main function that launches the Infinity List Shiny app.

---

query\_gbif\_global      *Generate galah query for Global GBIF Node*

---

**Description**

Generate galah query for Global GBIF Node

**Usage**

```
query_gbif_global(taxon, min_year, max_year, country_code = NULL)
```

**Arguments**

taxon	character, genus/family/kingdom
min_year	numeric, year cut off for query, only records where year $\geq$ min_year will be included
max_year	numeric, year cut off for query, only records where year $\leq$ max_year will be included
country_code	character, code for country

# Index

`download_ala_obs`, [2](#)  
`download_gbif_obs`, [3](#)

`infinity_server`, [5](#)  
`infinitylistApp`, [4](#), [6](#)

`query_gbif_global`, [6](#)