

# Package: ILORA (via r-universe)

May 12, 2026

**Title** Indian Alien Flora Information Database Retrieval

**Version** 0.1.2

**Description** The Indian Alien Flora Information (ILORA) database contains 14 invasion-relevant variables for 1388 alien plant species in India. The package enables exploration of the database using user-defined criteria. Using this package, users can retrieve variable-specific and species-level data from the database. The package also supports exploratory data analysis and visualization to give users an idea of the variables of interest. Further details about the database are available at <https://iloradb.wixsite.com/alienflora>.

**License** MIT + file LICENSE

**Depends** R (>= 4.0.0)

**Imports** DBI, RColorBrewer, units, RPostgres, dplyr, ggplot2, plotly, glue, rnatuarearth, rnatuarearthdata, sf, viridis, tidyr

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Ruqaiya Shaikh [aut, cre], Achyut Kumar Banerjee [aut],  
Abhishek Mukherjee [aut], Amiya Ranjan Bhowmick [aut]

**Maintainer** Ruqaiya Shaikh <ruqaiyashaikh41@gmail.com>

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

**Repository** <https://ruqaiya41.r-universe.dev>

**Date/Publication** 2026-05-12 12:22:12 UTC

**RemoteUrl** <https://github.com/cran/ILORA>

**RemoteRef** HEAD

**RemoteSha** 2a69c623bc1de9d64de2c3c677448b6829cf108b

## Contents

bar_plot . . . . .	2
calculate_market_metrics . . . . .	3
calculate_mrt . . . . .	4
calculate_total_uses . . . . .	5
EDA . . . . .	6
generate_palette . . . . .	7
get_data . . . . .	8
get_species_details . . . . .	8
get_species_names . . . . .	9
get_table_names . . . . .	10
get_variable_names . . . . .	10
ILORA . . . . .	11
ilora_variable_guide . . . . .	11
introduction_pathways_plot . . . . .	12
plot_first_record_year_histogram . . . . .	12
plot_species_on_map . . . . .	13
species_count_plot . . . . .	14
visualize_native_range . . . . .	15
visualize_naturalized_range . . . . .	16
<b>Index</b>	<b>17</b>

---

bar_plot	<i>Plot the Number of Species by Invasion Status</i>
----------	--

---

### Description

Retrieves species data from the `sp_categorization` table in the ILORA database and creates a bar plot showing the number of species in each invasion status category. Categories include Invasive (In), Naturalized (Nt), Casual Aliens (CA), Cryptogenic (CG), and Native (N).

### Usage

```
bar_plot()
```

### Value

A ggplot object representing the number of species in each invasion status category.

### Examples

```
# Generate a bar plot of species counts for each invasion status
bar_plot()
```

---

`calculate_market_metrics`*Calculate Market Metrics for Seed and Plant Prices*

---

### Description

Retrieves market data from the `market_dynamics` table in the ILORA database and computes key metrics for seed and plant prices. Metrics include average prices, and identification of species with the highest and lowest prices for seeds, nursery-live plants (NL), and plant-live plants (PL).

### Usage

```
calculate_market_metrics()
```

### Details

NL refers to Nursery-Live plants and PL refers to Plant-Live plants. The function calculates averages across all species recorded in the database and identifies extremes for each price category.

### Value

A named list containing:

**averages** A data frame with average seed, NL plant, and PL plant prices.

**highest\_seed** Data frame with species having the highest seed price.

**lowest\_seed** Data frame with species having the lowest seed price.

**highest\_nl\_plant** Data frame with species having the highest NL plant price.

**lowest\_nl\_plant** Data frame with species having the lowest NL plant price.

**highest\_pl\_plant** Data frame with species having the highest PL plant price.

**lowest\_pl\_plant** Data frame with species having the lowest PL plant price.

### Examples

```
# Compute market metrics
market_metrics <- calculate_market_metrics()

# View average prices
market_metrics$averages

# View species with highest seed price
market_metrics$highest_seed
```

---

`calculate_mrt`*Calculate Minimum Residence Time (MRT) for Species*

---

### Description

Computes the Minimum Residence Time (MRT) for species based on their first recorded year in the ILOGA database. MRT is defined as the difference between the current year and the species' first record year. Users can calculate MRT for specific species or all species.

### Usage

```
calculate_mrt(species = "ALL")
```

### Arguments

`species` Character vector or single string specifying species names to calculate MRT for. Use "ALL" to calculate MRT for all species in the database. Defaults to "ALL".

### Value

A data frame with columns:

- `acc_species_name`: The accepted species name.
- `first_year`: The first recorded year of the species.
- `mrt`: Minimum Residence Time (current year minus first recorded year).

If a specified species is not found in the database, a message is printed indicating the missing species. Returns NULL if no matching records are found.

### Examples

```
# Calculate MRT for all species
all_species_mrt <- calculate_mrt(species = "ALL")

# Calculate MRT for specific species
species <- c("Species A", "Species B", "Species C")
specific_species_mrt <- calculate_mrt(species)
```

---

calculate\_total\_uses *Calculate Total Number of Uses for Each Species*

---

### Description

Computes the total number of uses for species based on the economic\_uses table in the ILORA database. It sums across all use-related columns for each species.

### Usage

```
calculate_total_uses(species = NULL)
```

### Arguments

species            Optional character vector of species names to filter results. If NULL (default), results for all species are returned.

### Value

A data frame with columns:

- acc\_species\_name: The accepted species name.
- acc\_species\_id: Internal species ID.
- total\_uses: Sum of all recorded uses across relevant economic use columns.

If no matching records are found, a message is printed and NULL is returned.

### Examples

```
# Calculate total uses for all species
all_species_uses <- calculate_total_uses()

# Calculate total uses for specific species
species_list <- c("Quercus robur L.", "Azadirachta indica")
specific_species_uses <- calculate_total_uses(species_list)
```

## Description

This function acts as a dynamic interface to call any available EDA function from the ILORA package. Users specify which function to execute and pass any required arguments to that function.

## Usage

```
EDA(func, ...)
```

## Arguments

<code>func</code>	Character string specifying the name of the function to call.
<code>...</code>	Additional arguments to pass to the specified function.

## Details

Available functions that can be called through `EDA()` include:

**plot\_species\_on\_map** Plots specified species on a map of India.

**bar\_plot** Bar plot showing counts of species across 5 invasion status categories.

**species\_count\_plot** Returns a list of bar plots for growth habit and duration, plus summary tables.

**introduction\_pathways\_plot** Pie chart showing species counts by introduction pathways.

**plot\_first\_record\_year\_histogram** Histogram of species first record years within a given interval.

**calculate\_mrt** Calculates Minimum Residence Time (MRT) for species as current year minus first record year.

**calculate\_total\_uses** Calculates total number of uses for each species.

**calculate\_market\_metrics** Returns metrics for average seed/plant prices and species with highest/lowest prices.

**visualize\_native\_range** Visualizes the native range of species on a global map.

**visualize\_naturalized\_range** Visualizes the naturalized range of species on a global map.

Use `?function_name` for detailed help on each individual function.

## Value

The output of the specified EDA function. This could be a ggplot object, a Plotly object, a list of plots and summaries, or a data frame, depending on the function.

## Examples

```
# Call the bar_plot function
EDA("bar_plot")

# Call plot_species_on_map with additional arguments
EDA("plot_species_on_map",
    species_names = c("Quercus robur L."),
    palette = "inferno",
    opacity = 0.9,
    highlight = FALSE)
```

---

generate_palette	<i>Generate a Color Palette</i>
------------------	---------------------------------

---

## Description

Returns a vector of colors based on a specified palette name. Supports built-in R palettes, RColorBrewer palettes, viridis palettes, and custom diverging palettes.

## Usage

```
generate_palette(n, palette)
```

## Arguments

n	Integer. Number of colors to generate.
palette	Character. Name of the palette. Supported values: "rainbow", "heat", "terrain", "topo", "cm", "viridis", "plasma", "inferno", "magma", "cividis", "green", "red", "purple", "orange", "yellow", "brown", "diverging1", "diverging2", "diverging3", "diverging4".

## Value

Character vector of color hex codes.

## Examples

```
# Generate 5 colors from rainbow palette
generate_palette(5, "rainbow")

# Generate 10 colors from viridis palette
generate_palette(10, "viridis")
```

---

get\_data

*Retrieve Data for One or More Species from ILORA Database*


---

### Description

Fetches either specific variables across tables or an entire table for one or more species from the ILORA database.

### Usage

```
get_data(species_name, variables = NULL, table_name = NULL)
```

### Arguments

species_name	Character vector of species names to fetch data for. Must match acc_species_name in the database.
variables	Character vector, optional. Specific variables to retrieve. Should match variable names returned by get_variable_names().
table_name	Character, optional. Name of the table to retrieve data from. Either variables or table_name must be provided, not both.

### Value

A data frame containing the requested species data. Returns an error if species or variables/table are not found.

### Examples

```
# Retrieve specific variables
get_data("Quercus robur L.", variables = c("Longitude", "Latitude"))

# Retrieve all columns from a specific table
get_data("Quercus robur L.", table_name = "habitat")
```

---

get\_species\_details

*Retrieve Comprehensive Species Information from ILORA Database*


---

### Description

This function fetches detailed data for one or more species from multiple ILORA database tables. It retrieves taxonomy, invasion status, general information, native range, introduction pathways, uses, market data, habitat, naturalized range, occurrence and distribution, geography, and climate.

**Usage**

```
get_species_details(species_name)
```

**Arguments**

**species\_name** Character vector of species names to query. Must be non-empty.

**Value**

A structured list containing data frames for each category:

**taxonomy** Taxonomic classification.

**invasion\_status** Invasion status of the species.

**general\_info** General species information.

**native\_range** Native range at TDWG level 2.

**introduction** Introduction pathway and first record date in India.

**uses** Economic uses across different sectors.

**market\_info** Market metrics such as plant prices.

**habitat** Habitat information.

**naturalized\_range** Naturalized range at TDWG level 2.

**occurrence\_distribution** List containing occurrence and distribution data frames.

**geography** List containing LULC, anthrome, and ecoregion data frames.

**climate** Climate-related data.

**Examples**

```
species_details <- get_species_details(c("Azadirachta indica", "Quercus robur L."))
species_details$taxonomy
species_details$market_info
```

---

get\_species\_names      *Retrieve Species Names from ILORA Database*

---

**Description**

This function connects to the ILORA database and retrieves all species names available in the species\_name table. It returns them as a character vector in alphabetical order.

**Usage**

```
get_species_names()
```

**Value**

A character vector containing all species names in the ILORA database. If no species are found, returns an empty character vector.

**Examples**

```
species_names <- get_species_names()
head(species_names)
```

---

get\_table\_names      *Retrieve Table Names from ILORA Database*

---

**Description**

This function connects to the ILORA database and returns all table names in the public schema, excluding system tables.

**Usage**

```
get_table_names()
```

**Value**

A character vector of table names. Returns an empty vector if no tables are found.

**Examples**

```
# Retrieve all user tables
get_table_names()
```

---

get\_variable\_names      *Retrieve Column Names by Table from ILORA Database*

---

**Description**

This function connects to the ILORA database and retrieves all column names for tables in the public schema. Columns are grouped by table name.

**Usage**

```
get_variable_names(include_id = FALSE)
```

**Arguments**

`include_id` Logical; if FALSE (default), identifier columns such as `acc_species_id` are excluded. If TRUE, all columns are included.

**Value**

A named list. Each element is a character vector of column names for a table. If no tables/columns are found, returns an empty list.

**Examples**

```
# Retrieve variables excluding ID columns
get_variable_names()

# Retrieve variables including ID columns
get_variable_names(include_id = TRUE)
```

---

ILORA

*ILORA: A package for retrieving species data from ILORA database*

---

**Description**

Provides tools to access, query, and visualize species occurrence data from the Indian Alien Flora Information (ILORA) database. The package enables users to retrieve species names, explore available variables, and extract species-level data based on user-defined criteria. It also supports exploratory data analysis and visualization to facilitate ecological and biogeographical research. Further details about the database are available at <https://ilora2020.wixsite.com/ilora2020>.

---

`ilora_variable_guide` *Open ILORA Variable Description PDF*

---

**Description**

Opens the ILORA variable description PDF file (`ILORA_HowToRead.pdf`) in the system's default PDF viewer. This guide helps users understand the variables and structure of the ILORA dataset.

**Usage**

```
ilora_variable_guide()
```

**Details**

This function uses `utils::browseURL()` to open the PDF file. Depending on the user's system configuration, the file may open in a web browser, external PDF viewer, or within RStudio's viewer pane.

**Value**

No return value. This function is called for its side effect of opening the PDF file.

**Examples**

```
ilora_variable_guide()
```

---

```
introduction_pathways_plot
```

*Species Count Plot by Introduction Pathways*

---

**Description**

Retrieves species data from the introduction table in the ILORA database, counts distinct species for each reported introduction pathway, and generates a 3D pie chart using Plotly. The summary table of counts is also printed.

**Usage**

```
introduction_pathways_plot()
```

**Value**

A plotly object representing a 3D pie chart of species counts by introduction pathway. The summary table of top pathways is printed as a message.

**Examples**

```
# Generate pie chart for species introduction pathways
plot <- introduction_pathways_plot()
plot # Displays the Plotly pie chart
```

---

```
plot_first_record_year_histogram
```

*Histogram of Species First Record Year*

---

**Description**

Retrieves first record years of species from the introduction table in the ILORA database within a specified year interval and plots a histogram using ggplot2.

**Usage**

```
plot_first_record_year_histogram(start_year, end_year, binwidth = NULL)
```

**Arguments**

start_year	Numeric. Start year of the interval.
end_year	Numeric. End year of the interval.
binwidth	Numeric, optional. Width of histogram bins. If NULL (default), it is automatically determined based on the year range.

**Value**

A ggplot object representing the histogram of species first record years.

**Examples**

```
# Plot histogram for species first records between 1990 and 2000
plot_first_record_year_histogram(start_year = 1990, end_year = 2000)
```

---

plot\_species\_on\_map *Plot Latitude and Longitude of Species on a Map of India*

---

**Description**

Visualizes occurrences of one or more species on a map of India using their latitude and longitude data from the ILORA database.

**Usage**

```
plot_species_on_map(
  species_names,
  custom_colors = NULL,
  palette = "viridis",
  opacity = 0.7,
  highlight = FALSE
)
```

**Arguments**

species_names	Character vector. Names of species to plot. Must match acc_species_name in the database.
custom_colors	Optional. Named vector of colors for each species. If provided, length must match species_names.

palette	Optional. Character string specifying a palette for colors. Options include "rainbow", "heat", "terrain", "topo", "cm", "viridis", "plasma", "inferno", "magma", "cividis", "green", "red", "purple", "orange", "yellow", "brown", "grey", "diverging1", "diverging2", "diverging3", "diverging4".
opacity	Numeric. Opacity of the plotted points (0 to 1). Default 0.7.
highlight	Logical. If TRUE, highlights regions containing species occurrences. Default FALSE.

### Details

Plots multiple species occurrences with either a specified palette or custom colors. Ensures spatial coordinates align with India's shapefile CRS.

### Value

A ggplot object displaying the species occurrences on a map of India.

### Examples

```
# Plot multiple species with custom colors
plot_species_on_map(
  c("Cyanus segetum Hill", "Cuphea llavea Lex."),
  custom_colors = c("orange", "blue")
)

# Plot a single species using the heat palette and reduced opacity
plot_species_on_map("Quercus robur L.", palette = "heat", opacity = 0.6)

# Plot a species using custom colors and viridis palette
plot_species_on_map("Avena sativa L.", custom_colors = c("orange"), palette = "viridis")
```

---

species\_count\_plot      *Species Count Plot by Growth Habit and Duration*

---

### Description

Retrieves species data from the general\_information table in the ILORA database, calculates the number of distinct species for each growth habit and duration category, and generates bar plots to visualize these counts. Summary tables are also included.

### Usage

```
species_count_plot()
```

**Value**

A list containing:

**plot\_growthhabit** A ggplot object showing species counts by growth habit.

**plot\_duration** A ggplot object showing species counts by duration.

**growthhabit\_summary** A data frame summarizing species counts by growth habit.

**duration\_summary** A data frame summarizing species counts by duration.

**Examples**

```
# Generate and display plots for growth habit and duration
plots <- species_count_plot()
print(plots$plot_growthhabit)
print(plots$plot_duration)
```

---

visualize\_native\_range

*Visualize Native Range of Species on a Global Map*

---

**Description**

This function retrieves the native range of one or more species from the ILORA database and plots their distribution on a global map using TDWG Level 2 regions.

**Usage**

```
visualize_native_range(species_name)
```

**Arguments**

**species\_name** Character vector of species names to plot. Use "ALL" to plot all species in the database.

**Details**

The function joins the native range data from the ILORA database with a TDWG Level 2 shapefile to display species' native regions. Each species is color-coded, and overlapping ranges are handled to avoid duplicate polygons.

**Value**

A ggplot2 object showing the native range of the specified species.

**Examples**

```
# Visualize native range of specific species
visualize_native_range(c("Cyanus segetum Hill", "Avena sativa L."))
```

---

```
visualize_naturalized_range
```

*Visualize Naturalized Range of Species on a Global Map*

---

**Description**

This function retrieves the naturalized range of one or more species from the IORA database and plots their distribution on a global map using TDWG Level 2 regions.

**Usage**

```
visualize_naturalized_range(species_name)
```

**Arguments**

`species_name` Character vector of species names to plot. Use "ALL" to plot all species in the database.

**Details**

The function joins the naturalized range data from the IORA database with a TDWG Level 2 shapefile to display species' naturalized regions. Each species is color-coded.

**Value**

A ggplot2 object showing the naturalized range of the specified species.

**Examples**

```
# Visualize naturalized range of specific species
visualize_naturalized_range(c("Cyanus segetum Hill", "Avena sativa L."))
```

# Index

`bar_plot`, 2

`calculate_market_metrics`, 3

`calculate_mrt`, 4

`calculate_total_uses`, 5

EDA, 6

`generate_palette`, 7

`get_data`, 8

`get_species_details`, 8

`get_species_names`, 9

`get_table_names`, 10

`get_variable_names`, 10

ILORA, 11

`ilora_variable_guide`, 11

`introduction_pathways_plot`, 12

`plot_first_record_year_histogram`, 12

`plot_species_on_map`, 13

`species_count_plot`, 14

`visualize_native_range`, 15

`visualize_naturalized_range`, 16