M12 Input data streams as standardised open data and citable objects

2026-04-30

Author(s): Marina Golivets, Ahmed El-Gabbas

Funded by
the European Union

Prepared under contract from the European Commission
Grant agreement No. 101180559
EU Horizon Europe Innovation Action
European Research Executive Agency

Project acronym:	**OneSTOP**
Project full title:	**OneBiosecurity Systems and Technology for People, Places and Pathways**
Project duration:	01.01.2025 – 30.06.2028 (42 months)
Project coordinator:	Dr. Quentin Groom, Agentschap Plantentuin Meise (MeiseBG)
Call:	HORIZON-CL6-2024-BIODIV-01-1
Milestone title:	Input data streams as standardised open data and citable objects
Milestone №:	M12
WP responsible:	WP5
Means of verification:	Datasets and scripts available
Licence of use:	CC BY
Lead partner:	Helmholtz Centre for Environmental Research – UFZ
Recommended citation:	Golivets, M., El-Gabbas, A. ***Input data streams as standardised open data and citable objects.*** OneSTOP project M12.
Due date of milestone:	Month 18
Actual submission date:	Month 18

Milestone status:

Version	Status	Date	Author(s)
1.0	Draft	2026-04-29	Marina Golivets; UFZ

Table of contents

Summary

As part of the One Biosecurity approach to terrestrial invasive alien species (IAS) within the OneSTOP project, **Task 5.2** focuses on developing computational functionality for the automated modelling of IAS distributions across Europe under current and potential future environmental conditions. This will be achieved by developing an R package[1] that contains all the necessary functions to reliably model IAS distributions with minimal user input.

A key component of the automated species distribution modelling (SDM) workflow is the streaming and pre-processing of input data. Unlike traditional approaches, in which datasets are downloaded and prepared once, automated workflows rely on continuously updated, open data sources and tools to stream data as needed.

This milestone report outlines the approaches implemented in Task 5.2 to efficiently harvest, process, store, and prepare data for direct use in SDMs. It summarises the input datasets and the R functions developed to stream these data through an automated workflow, and highlights key differences from the approach used in Task 5.1 in how input data are handled.

List of abbreviations

EU	European Union
IAS	Invasive alien species
GBIF	Global Biodiversity Information Facility
EASIN	European Alien Species Information Network
OSF	Open Science Framework

[1] Currently developed and stored at https://github.com/MarinaGolivets/OneSDM

1. OneSDM R package

1.1. Overall approach

In Task 5.2, we are developing an R package, **OneSDM**[2], to enable the automated, seamless execution of species distribution models (SDMs) for IAS prioritisation and management planning. OneSDM is designed to cover the full SDM workflow, from data acquisition to model output post-processing and reporting. As part of this effort, dedicated data-streaming functionalities are being developed and integrated into the package.

The OneSDM package adopts a nested structure, combining a suite of fine-grained functions with a smaller set of higher-level, aggregated functions. This design will enable users to execute complete modelling workflows with only a few lines of code by relying on aggregated functions and their default settings. At the same time, it will preserve full flexibility for advanced users, who can access and customise individual components through lower-level functions, adjustable function arguments and global options.

1.2. Differences from WiSDM

Compared with Task 5.1, which applies the **WiSDM** framework (Davis et al., 2024) to generate distributions for selected IAS, OneSDM is designed to minimise reliance on hard-coded parameters and locally stored user inputs. For example, WiSDM requires all input data (>30 GB) to be downloaded and stored locally for processing, which can quickly exhaust available memory and significantly slow down analyses.

In contrast, OneSDM retrieves only the required data on demand from online repositories via APIs (described in more detail below). This approach reduces local storage requirements and improves computational efficiency. Additionally, OneSDM decreases processing time for data preparation, model execution, and output generation by using coarser spatial resolutions, chunked data retrieval, and more memory-efficient data formats.

Compared with WiSDM, OneSDM expands the range of supported species occurrence data sources to include, in addition to the Global Biodiversity Information Facility (GBIF[3]), data from the European Alien Species Information Network (EASIN[4]) and user-provided datasets. Integrating multiple data sources is particularly valuable for poorly studied or newly introduced alien species. Furthermore, occurrence records from multiple species can be combined to model the distributions of higher taxonomic groups (e.g. aggregates, or genera).

---

[2] El-Gabbas, A. & Golivets, M. (2026). OneSDM: Automated workflows for species distribution modelling under environmental change scenarios. R package version 0.1.0, https://github.com/MarinaGolivets/OneSDM

[3] https://www.gbif.org/

[4] https://easin.jrc.ec.europa.eu/easin

Importantly, to maintain internal consistency within the project and ensure comparability of results, OneSDM adopts the methodological decisions implemented in Task 5.1 as default settings wherever possible. It also relies on the same raw environmental data sources to generate input datasets for modelling.

1.3. Data streams

1.3.1. Environmental data

OneSDM uses climate and land-use variables as model predictors. Climatic data were obtained from the CHELSA v2.1 database (Karger et al., 2021). Variables considered potentially relevant for modelling IAS distributions were processed at three spatial resolutions (5 km, 10 km, and 20 km) globally, under both current conditions and future scenarios. This resulted in 7,520 climate raster layers, all of which were exported as compressed GeoTIFF files for downstream tasks.

Land-use data were sourced from Chen et al. (2022) and processed in line with the approach adopted in OneSTOP Task 5.1. Specifically, the original 20 plant functional types (PFTs) were reclassified into 12 broader categories. Land-use layers were then aggregated to the same three spatial resolutions, and only time frames and scenarios overlapping with the CHELSA dataset were retained. Three types of GeoTIFF datasets were generated: (i) majority class per grid cell based on the original 20 PFT classes; (ii) percentage cover per grid cell for each of the 20 original PFT classes; and (iii) percentage cover per grid cell for the 12 aggregated PFT classes used in Task 5.1.

All environmental layers are hosted in a public Open Science Framework (OSF) project[5] for streaming and downstream processing. OSF was selected as the data repository due to its unlimited storage capacity, unlike alternatives such as Zenodo, and its support for direct data access via an API. This enables seamless integration with the R environment through the osfr package (Wollen et al., 2020).

To retrieve climate and land-use data, as well as mask layers from OSF, we wrote three R functions. Below, we describe each function in detail.

[5] El-Gabbas, A., & Golivets, M. (2026). Supporting data for the OneSTOP's OneSDM framework. Retrieved from osf.io/7mwxp

1.3.1.1. get_climate_data

Title: Download climate data files from the OneSDM OSF project

Description: Downloads selected climate raster files (GeoTIFF) from the OneSDM Open Science Framework (OSF) storage based on chosen spatial resolution, climate scenario, climate model, time period, and variable names. The function validates inputs, locates the corresponding files in the OSF project, creates local directories as needed, downloads the files, and verifies their integrity. Returns (invisibly) a tibble with metadata of the downloaded climate data files, including their local paths.

Function arguments:

```
get_climate_data(
  climate_dir = NULL,
  resolution = 10L,
  climate_scenario = "current",
  climate_model = "current",
  year = "1981_2010",
  var_names = NULL,
  verbose = TRUE,
  sleep_time = 1L
)
```

climate_dir	(character) Destination directory for climate and land-use data files. The same directory should be used when modelling multiple species to ensure consistency. Default is NULL.
resolution	(numeric) Spatial resolution. Valid values are 5, 10, or 20 for resolutions of approximately 5, 10, and 20 km (2.5, 5, and 10 arc-minutes, respectively). Default is 10L.
climate_scenario	(character). Climate scenario: current, ssp126, ssp370, ssp585. Default is "current".
climate_model	(character) Global Circulation Model: current, gfdl, ipsl, mpi, mri, ukesm1. Default is "current".
year	(character) Time period: 1981_2010, 2011_2040, 2041_2070, 2071_2100. Default is "1981_2010".
var_names	(character) A vector of climate variable codes to download. See climate_data for the list of valid climate variable names. Required.
verbose	(logical) If TRUE (default), prints progress messages during execution.
sleep_time	(numeric) Number of seconds to pause after downloading files to avoid overwhelming the server. Default is 1L.

1.3.1.2. get_landuse_data

Title: Download land use data files from the OneSDM OSF project

Description: Downloads selected land use data files from the OneSDM Open Science Framework (OSF) storage based on spatial resolution, climate scenario, climate model, time period, and Plant Functional Type (PFT) names. The function validates input parameters, checks for existing files, and downloads only those that are missing or corrupted. Returns (invisibly) a tibble containing metadata about the downloaded files, including local file paths, OSF paths, download links, and validation status.

Function arguments:

```
get_landuse_data(
  climate_dir = NULL,
  resolution = 10L,
  climate_scenario = "current",
  year = "1981_2010",
  pft_type = "cross-walk",
  pft_id = NULL,
  verbose = TRUE,
  sleep_time = 1L
)
```

climate_dir	See get_climate_data().
resolution	See get_climate_data().
climate_scenario	See get_climate_data().
year	See get_climate_data().
pft_type	(character). Plant functional type category. Either "cross-walk" (default) or "original". See landuse_data for details.
pft_id	(numeric) A vector of plant functional type identifiers to download. Must be valid for the specified pft_type: 1-20 for pft_type == "original" and 1-12 for pft_type == "cross-walk". See landuse_data for details. Required.
verbose	See get_climate_data().
sleep_time	See get_climate_data().

1.3.1.3. get_mask_layer

Title: Download land use data files from the OneSDM OSF project

Description: Loads a mask layer (raster) at a specified model resolution from the OneSDM Open Science Framework (OSF) storage. Returns either a SpatRaster object or a path.

Function arguments:

```
get_mask_layer(
  resolution = NULL,
  climate_dir = NULL,
  verbose = FALSE,
  europe_only = FALSE,
  overwrite = FALSE,
  return_spatraster = TRUE,
  wrap = FALSE
)
```

resolution	See get_climate_data().
climate_dir	See get_climate_data().
verbose	See get_climate_data().
europe_only	(logical) If TRUE, downloads the Europe-only mask layer. Default is FALSE.
overwrite	(logical) Should existing files be overwritten? Default is FALSE.
return_spatraster	(logical) If TRUE (default), returns a SpatRaster object. If FALSE, saves the raster to a file and returns the file path (invisibly).
wrap	(logical) Should the resulting SpatRaster be wrapped using *terra::wrap()*? Default is FALSE.

1.3.2. Occurrence data

OneSDM prepares species (or other taxonomic level) occurrence data from one or more sources — GBIF, EASIN, and user-provided datasets. The functionality for this is wrapped in the R function *prepare_species_data()*. Internally, the function relies on *prepare_gbif_data()*, *prepare_easin_data()*, and *prepare_user_data()* to retrieve and process data from each respective source. Below, we describe all **four R functions**.

1.3.2.1. prepare_gbif_data

Title: Download and Process GBIF Occurrence Data

Description: Downloads, filters, and processes occurrence records from GBIF for specified taxon keys (GBIF IDs). The function handles data requests, downloads, cleaning, and conversion to an sf object, with optional spatial and temporal filters, coordinate-uncertainty thresholds, and geographic boundary constraints.

The function performs the following steps:

- Validates input parameters and environment.
- Checks for existing processed data to avoid redundant downloads/requests.

- Requests occurrence data from GBIF using the rgbif package, applying filters for coordinates, geospatial issues, year, occurrence status, basis of record, and spatial boundaries.
- Downloads and reads the raw data, applying further cleaning:
    - Removes records with missing or low-precision coordinates.
    - Filters by spatial uncertainty and accepted taxonomic ranks.
    - Applies additional cleaning using the CoordinateCleaner package.
- Converts the cleaned data to an sf object and saves it.
- If return_data is TRUE, return the cleaned GBIF data as an sf object. Otherwise, returns (invisibly) a named list with paths to the saved GBIF data files.

Function arguments:

```
prepare_gbif_data(
  gbif_ids = NULL,
  model_dir = NULL,
  r_environ = ".Renviron",
  start_year = 1981L,
  boundaries = c(-180L, 180L, -90L, 90L),
  max_uncertainty = 10L,
  overwrite = FALSE,
  return_data = FALSE,
  verbose = TRUE
)
```

gbif_ids	(character or numeric) A vector of one or more GBIF taxon keys. When multiple IDs are supplied, data are combined across all keys. If NULL (default), the function attempts to retrieve IDs from the onesdm_gbif_ids option. Required.
model_dir	(character) Path to the directory where model outputs will be saved. A subdirectory named data is automatically created within this directory to store processed species data. When modelling multiple species, it is recommended to use a separate directory for each run to avoid overwriting or mixing data files. Default is NULL. Required.
r_environ	(character) Path to an .Renviron file containing GBIF credentials (default: ".Renviron"). The function uses *ecokit::check_gbif()* to verify the presence of GBIF credentials. If the GBIF_USER, GBIF_PWD, and GBIF_EMAIL environment variables are not set in the current R session, check_gbif attempts to read them from the specified .Renviron file. If r_environ = NULL, the path to the .Renviron file is retrieved from the "onesdm_r_environ" option.
start_year	(integer) The starting year from which records are included. The default is 1981L, to match the temporal coverage of CHELSA climate data.

boundaries	(numeric) A vector of size 4 containing spatial boundaries as (left, right, bottom, top) in decimal degrees (default: c(-180L, 180L, -90L, 90L) for global extent).
max_uncertainty	(numeric) Maximum allowed spatial uncertainty in kilometres. Default is 10L.
overwrite	(logical) If TRUE, overwrites existing cleaned GBIF data file in the model directory. Default is FALSE.
return_data	(logical) If TRUE, returns the processed GBIF data as an sf object in addition to saving it. Default is FALSE.
verbose	(logical) If TRUE (default), prints progress messages during execution.

1.3.2.2. prepare_easin_data

Title: Download and Clean EASIN Occurrence Data

Description: Downloads, combines, cleans, and saves occurrence data for given EASIN species IDs from the EASIN API.

The function performs the following steps:

- Supports chunked downloads, multiple attempts, and extensive data cleaning, including coordinate-precision checks and spatial filtering with the CoordinateCleaner package. The cleaned data is saved as an .RData file in the specified model directory.
- Checks for an existing cleaned EASIN data file and skips download if found.
- Applies several cleaning steps, e.g., filtering out records with low coordinate precision, duplicate records, or records at equal longitude/latitude, as well as records near centroids, capitals, biodiversity institutions, and GBIF headquarters.
- Invisibly returns the file path to the saved, cleaned EASIN data easin_data.RData file in the data subdirectory of model_dir, unless return_data is TRUE, in which case it returns the cleaned EASIN data as an sf object.

Function arguments:

```
prepare_easin_data(
  easin_ids = NULL,
  model_dir = NULL,
  timeout = 600L,
  n_search = 1000L,
  n_attempts = 10L,
  sleep_time = 5L,
  exclude_gbif = TRUE,
  verbose = TRUE,
  start_year = 1981L,
  overwrite = FALSE,
  return_data = FALSE
```

)

easin_ids	(character) A vector of one or more EASIN species IDs, each starting with "R" followed by five digits (e.g., "R00544"). Species IDs can be obtained from the EASIN website by searching for a species and checking its EASIN ID in the species details section. When multiple IDs are provided, data are collated across all IDs. If NULL(default), the function attempts to retrieve IDs from the "onesdm_easin_ids" option and skips the EASIN download if no IDs are found. Required.
model_dir	See prepare_gbif_data().
timeout	(integer) Timeout (in seconds) for each download attempt. Default is 600L. Can also be set via the "onesdm_easin_timeout" option. *Note: This timeout applies to each download chunk separately and does not limit the total duration of the full download process.*
n_search	(integer) Number of records requested per API call (chunk size). Default is 1000L, which is the maximum allowed by the EASIN API.
n_attempts	(integer) Maximum number of download attempts per chunk. Default is 10L.
sleep_time	(integer) Time to wait (in seconds) between successive chunk downloads. This delay helps prevent overloading the EASIN server. Default is 5L.
exclude_gbif	(logical) If TRUE (default), GBIF records are omitted from the download.
start_year	See prepare_gbif_data().
overwrite	(logical) If TRUE, overwrites existing cleaned EASIN data in the modelling directory. Default is FALSE.
return_data	(logical) If TRUE, returns the processed EASIN data as an sf object in addition to saving it. Default is FALSE.
verbose	(logical) If TRUE (default), progress and information messages are printed, including the URL of the currently processed chunk.

1.3.2.3. prepare_use_data

Title: Process and Validate User Observation Coordinates

Description: Processes a set of user-provided geographic coordinates, validates their structure and values, converts them into a spatial object, and saves the result to a specified model directory.

The function performs the following steps:

- Validates that coordinates have at least one row and exactly two columns.
- Ensures that the columns represent numeric longitude and latitude values within valid ranges (longitude: [-180, 180], latitude: [-90, 90]).
- Converts the coordinates to a tibble with standardised column names.
- Removes rows with missing or out-of-range values.
- Converts the cleaned coordinates to an sf spatial object (WGS84, EPSG:4326).
- Saves the resulting spatial object as data/user_coordinates.RData in model_dir.
- If return_data is TRUE, the function returns the processed spatial object of class sf. If FALSE, it returns the file path where the data is saved (invisible).

Function arguments:

```
prepare_user_data(
  coordinates = NULL,
  model_dir = NULL,
  return_data = FALSE
)
```

coordinates	A data frame or matrix containing longitude and latitude values. Must have exactly two columns. If not provided directly, the function will attempt to retrieve it from the "onesdm_coordinates" option.
model_dir	See prepare_gbif_data().
return_data	Logical. If TRUE, the function returns the processed spatial object instead of the file path. Default is FALSE.

1.3.2.4. prepare_species_data

Title: Prepare Species Occurrence Data for Species Distribution Modelling using OneSDM

Description: Prepares species occurrence data from one or more sources — GBIF, EASIN, and user-provided datasets — for species distribution modelling. It handles downloading, validating, integrating, and rasterising the data. Optional steps include excluding specified geographic areas and filtering spatial outliers. Internally, the function relies on *prepare_gbif_data()*, *prepare_easin_data()*, and *prepare_user_data()* to retrieve and process data from each respective source. Creates a data subdirectory within model_dir and saves multiple outputs, including raw, processed, and rasterised data, as well as optional visualisations.

The function performs the following steps:

- Validates input parameters.
- Retrieves species data from GBIF, EASIN, and/or user-provided coordinates.
- Merges data from all sources.
- Optionally removes spatial outliers based on nearest neighbour distances.

- Rasterises occurrence data to match the specified resolution.
- Excludes geographic extents if specified.
- Saves processed data and generates visualisations if requested.
    - The map the rasterised species distribution overlaid on a world map.
    - Excluded areas (from exclude_extents) are highlighted on the map in red.
    - If a specific extent is used for GBIF data retrieval, it is highlighted in green.

Function arguments:

```
prepare_species_data(
  gbif_ids = NULL,
  easin_ids = NULL,
  coordinates = NULL,
  model_dir = NULL,
  climate_dir = NULL,
  resolution = NULL,
  verbose = TRUE,
  exclude_extents = list(),
  species_name = "species",
  outlier_dist_km = 0L,
  outlier_resolution = 0.125,
  outlier_n_cores = 6L,
  plot_distribution = TRUE
)
```

gbif_ids	See prepare_gbif_data(). Optional.
easin_ids	See prepare_easin_data(). Optional.
coordinates	See prepare_user_data(). Optional.
model_dir	See prepare_gbif_data(). Required.
mask_dir	Character. This directory is used to load mask layers matching the specified resolution. The same directory should be used when modelling multiple species to ensure consistency. Default is NULL.
resolution	(numeric). Spatial resolution used to prepare data for analysis. Acceptable values are 5, 10, or 20, corresponding to approximate spatial resolutions of 5 km, 10 km, and 20 km (2.5, 5, and 10 arc-minutes), respectively. Default is NULL. Required.
verbose	(logical). If TRUE (default), prints progress messages during execution.
exclude_extents	(list). A list of SpatExtent objects (created with *terra::ext()* function) defining geographic areas to exclude from the data. An empty list (default) indicates that no areas are excluded. Optional.

species_label	(character). Name of the species used for labelling outputs, such as maps. This argument does not affect data retrieval from GBIF or EASIN. Default is "species".
outlier_dist_km	(numeric). Distance threshold (in kilometres) used to identify spatial outliers. If set to 0L (default), no outlier detection or filtering is applied. Optional.
outlier_resolution	(numeric). Spatial resolution (in degrees) used for outlier detection calculations. This parameter is only applied when outlier_dist_km is greater than 0. A coarser resolution can speed up computation at the cost of spatial precision. Default is 0.125.
outlier_n_cores	(integer). Number of CPU cores to use for parallel outlier detection. This parameter is only applied when outlier_dist_km is greater than 0. Default is 6L.
plot_distribution	logical). If TRUE (default), generates and saves a JPEG map of the final species distribution data.

2. References

Chen, G., Li, X. & Liu, X. (2022). Global land projection based on plant functional types with a 1-km resolution under socio-climatic scenarios. Sci Data 9, 125. https://doi.org/10.1038/s41597-022-01208-6.

Davis A.J.S., Groom Q., Adriaens T., Vanderhoeven S., De Troch R., Oldoni D., Desmet P., Reyserhove L., Lens L., Strubbe D. (2024) Reproducible WiSDM: a workflow for reproducible invasive alien species risk maps under climate change scenarios using standardized open data. Front. Ecol. Evol. 12:1148895. https://www.doi.org/10.3389/fevo.2024.1148895.

Karger, D. N., Conrad, O., Böhner, J., Kawohl, T., Kreft, H., Soria-Auza, R. W., Zimmermann, N. E., Linder, H. P., Kessler, M. (2021). Climatologies at high resolution for the earth's land surface areas. EnviDat. https://www.doi.org/10.16904/envidat.228.

Wolen, A.R., Hartgerink, C.H., Hafen, R., Richards, B.G., Soderberg, C.K., York, T.P. (2020). osfr: An R Interface to the Open Science Framework. Journal of Open Source Software, 5(46), 2071, https://doi.org/10.21105/joss.02071.